

BOUNDED VARIABLES NONLINEAR MULTIPLE CRITERIA OPTIMIZATION USING SCATTER SEARCH

RICARDO P. BEAUSOLEIL*

Recibido/Received: 14 May 2002

Abstract

This paper introduces an adaptation of multiple criteria scatter search to deal with nonlinear continuous vector optimization problems on bounded variables, applying Tabu Search approach as diversification generator method. Frequency memory and another escape mechanism are used to diversify the search. A relation Pareto is apply in order to designate a subset of the best generated solutions to be reference solutions. A choice function called Kramer Selection is used to divide the reference solution in two subsets. The Euclidean distance is used as a measure of dissimilarity in order to find diverse solutions to complement the subsets of high quality current Pareto solutions to be combined. Convex combination is used as a combined method. The performance of this approach is evaluated on several test problems taken from the literature.

Keywords: Tabu Search, Scatter Search, Nonlinear Optimization.

Resumen

El artículo presenta una adaptación del algoritmo de Búsqueda Dispersa Multi-objetivo para la solucionar problemas de optimización vectorial no lineales continuos, empleando un enfoque de Búsqueda Tabú como un método generador de soluciones diversas. Memoria de Frecuencias y otros mecanismos de escapes son utilizados para diversificar la búsqueda. La relación Pareto es aplicada para designar un subconjunto de las mejores soluciones generadas a ser soluciones de referencias. Una función de selección denominada selección de Kramer se utiliza para dividir al conjunto de referencia en dos subconjuntos. La distancia Euclideana es usada como una medida de disimilaridad a modo de hallar soluciones diversas que complementen los subconjuntos de soluciones potencialmente Pareto de alta calidad a ser combinadas. Como método de combinación usamos la combinación convexa. El desempeño de este enfoque es evaluado con diferentes problemas de pruebas tomados de la literatura.

*Centro de Matemática y Física Teórica, Calle E # 309 esq. a 15, Vedado, Ciudad de La Habana, C.P. 10400, Cuba. Fax: +(537) 33 33 73. E-Mail: rbeausol@cidet.icmf.inf.cu

Palabras clave: Búsqueda Tabú, Búsqueda Dispersa, Optimización No Lineal.

Mathematics Subject Classification: 90C29, 90B50.

1 Introduction

This paper presents one way to apply Tabu Search and Scatter Search in nonlinear multiobjective optimization, using a search followed by multiple criteria decision making approach, that is, generating a set of non-inferior solutions and then select one or more of these on the basis of multiple criteria decision making.

Following our methodology [3], we use a Pareto-Based Approach, that uses a Choice Function to rank solutions.

An adaptation of tabu search for nonlinear optimization to a multiobjective environment is used to generate an initial set of diverse non-inferior solution. Also this adaptation is extended to incorporate a scatter search approach to improve these solutions toward the Pareto frontier.

To apply our strategy, we consider two approaches: (1) directional search, where a transition from one point to another occurs by reference to feasible directions; (2) scatter search, where successive collections of points are generated from weighted combinations of reference points.

The organization of the paper is as follows. Notation and general methods are treated in section 2. Section 3 presents a nonlinear multiobjective tabu search approach. Section 4 is devoted to scatter search applied to nonlinear optimization. Section 5 gives some test problems for nonlinear multiobjective problems. Section 6 contains conclusions.

2 Notation and general methods

Formally, we can state a quantitative decision making as follows. Decisions have a quantitative character, a decision(or solution) $x \in E_x$, where E_x denotes a decision space, and $X \subseteq E_x$ is a set of admissible decisions. We have functions f_1, f_2, \dots, f_r , defined over a set of situations $X \times V$, where V is a finite set of uncertain factors values. Then for each situation (x, v) , where $x \in X$ and $v \in V$, we have a vector function $F(x, v) = (f_1(x, v), f_2(x, v), \dots, f_r(x, v))$. For deterministic problems the vector function $F(x)$ determines the quality of the decision x .

2.1 Non-dominance

We will refer to an objective function vector as a point. The point $F(x)$ dominates the point $F(x')$ if and only if $F(x) \geq F(x')$ and $F(x) \neq F(x')$ (i.e. if $\forall i : f_i(x) \geq f_i(x')$ and $f_i(x) > f_i(x')$ for at least one objective i). The point $F(x)$ is dominated by the point $F(x')$, if the point $F(x')$ dominates the point $F(x)$. If a point is not dominated by another point, it is called a non-dominated point.

Solution x is superior to solution x' if the point $F(x)$ dominates the point $F(x')$.

Solution x is inferior to solution x' if the point $F(x')$ dominates the point $F(x)$. If the point $F(x)$ is non-dominated, then x is non-inferior.

The set of all non-inferior solutions is sometimes referred to as the Pareto Optimal Set. The set of all non-dominated points in the objective space is referred to as the Pareto Frontier.

2.2 Choice function

In order to obtain a reference set of solutions that encourages the search toward the Pareto frontier, an optimality principle is used: “Selection by a number of dominant criteria” [19].

For all $x, x' \in X$, let $q(x, x')$ be the number of criteria for which the decision variable x' improves the decision variable x , then $Q_X = \max_{x' \in X} q(x, x'), x \in X$ can be seen as a discordance index if x is assumed to be preferred to x' . Then the Kramer Choice function is defined as follows: $C^K(X) = \{x' \in X | Q_X(x') = \min_{x \in X} Q_X(x)\}$.

3 Nonlinear multiobjective tabu search

Tabu Search TS is a strategy based on the use of prohibition-based techniques and “intelligent” schemes as a complement of basic heuristic algorithms like local search, with the purpose of guiding the basic heuristic beyond local optimality.

In a nonlinear optimization context, a standard move is $m(x) = x + hd$, where d is a specified direction vector, such as a generalized gradient, and h is a scalar step size. In our approach we use a similar strategy. By the standard TS approach, a move is classified tabu and excluded from consideration if it reverses a recent previous move. An adaptation of Tabu search strategies for preventing move reversals in nonlinear context is applied, see [11]. A sequential fan strategy is used to create our neighborhood. Also, a TS strategy to guide the search toward the Pareto frontier is introduced. A weighted sum approach is used as a decision rule to transit from one solution to another. Recency and frequency based memory for diversification are used in our approach.

3.1 Directional search strategy

3.1.1 Neighborhood (move description)

In the following we suppress reference to the indexes of the different variables for notational convenience, and use the following move: $m(x) = x \oplus h^t d$ where we have h^t as the step sizes at iteration t , h^t positive. Let $\Delta x^t = h^t d$, and assume the condition $[x - \Delta x^t, x + \Delta x^t] \cap [a, b] \neq \emptyset$ holds. Then, $\Delta x^t \leq |x - a|$ if the operation in the move is “-”, in this case $d = |x - a|$, and $\Delta x^t \leq |b - x|$ if the operation in the move is “+”, then d take the value $|b - x|$, in both cases, x ranges over $[a, b]$, $h^t \in A = \{r/fan : r \text{ is a random number in a set } \{2, 3, \dots, fan\}\}$, where fan is the number of trial solutions created, as explained above. In our implementation we propose to move as maximum four variables, in the case of more than four variables we can use random controlled and frequency memory to select the move-variables.

3.1.2 Status tabu

We focus our attention on tabu conditions based on move reversals. For our purpose, we will select variables as a basis for defining move attributes, identifying the change values in going from one solution to another [11]. Tabu restrictions are imposed to prevent moves that bring the values of variables “too close” to values they held previously. Specifically a move is tabu if it creates a solution x which lies closer than a specified tabu distance $dist$ to any solution visited during the preceding t iterations.

The implementation of this rule is as follows: the variable x' is excluded from falling inside the line interval bounded by $x - w(x' - x)$ and $x + w(x' - x)$, where $1 \geq w > 0$, when a move from x to x' is executed. In our approach we use a tenure = 7 for the tabu memory structure and one tabu list for each variable.

3.1.3 Memory structures

We briefly illustrate the meaning of the variables and memory structures used. An elementary move is indexed by the variable h^t , the step size. The tabu distances are saved in records that contain the lower and the upper bounds of these distances, `tabu[current].lower` and `tabu[current].upper`, where `current` is a pointer to tabu list, `tabu[current].IX` and `tabu[current].IY` stores the “from” and “to” variables, we have a similar tabu list for each variable. Our tabu lists are a circular one.

Our approach uses a frequency-based memory denoted by residence, this is a record that has two entries, `residence[j].range` contains the upper bound of a sub-range of $[a, b]$, and `residence[j].freq` contains the number of times that the associated sub-ranges have been visited, j is an index. A similar record is associated to each variable.

3.1.4 Candidate list strategy

We use a simplified version of a sequential *fan* strategy as candidate list strategy. The sequential *fan* generates p best alternative moves at a given step, and then to create a *fan* of solution streams, one for each alternative. The best available moves for each stream are again examined, and only the p best moves overall provide the p new streams at the next step. In our case, taking $p = 1$, we have in each step for $h^t \in A$ one stream and several points equal to *fan*.

3.1.5 Search by goals

Our implementation uses as move attribute, variables that change their values as result of the move. We represent change represented by a difference of values $f_i(x') - z_i^* \forall i = 1..r$, $x' \in X$ where x' was generated from x by a recent move, x is a current solution and Z^* is a reference solution, $Z^* = (z_1^*, \dots, z_r^*)$.

A thresholding aspiration is used to obtain an initial set of solutions as follows: without lost generality, assume that every criteria is maximized. Notationally, let $\Delta f(x') = (\Delta f_1(x'), \dots, \Delta f_r(x'))$ where $\Delta f_i(x') = f_i(x') - z_i^*$, $i \in \{1, \dots, r\}$.

Let

$$\Delta f_i(x') = \begin{cases} \text{preference} & \text{if } \Delta f_i(x') > 0, \\ \text{indifference} & \text{if } \Delta f_i(x') = 0, \\ \text{nonpreference} & \text{if } \Delta f_i(x') < 0, \end{cases}$$

A goal is satisfied, allowing x' to be accepted and introduced in S if $(\exists \Delta f_i(x') = \text{preference})$ or $(\forall i \in \{1, \dots, r\} [\Delta f_i(x') = \text{indifference}])$, otherwise it is rejected.

The point Z^* is updated by $z_i^* = \max f_i(x'), \forall i \in \{1, \dots, r\}, x' \in S$.

3.1.6 Weighted sum approach

In order to measure the quality of the solution we propose to use in our tabu search approach an Additive Function Value AFV with weighting coefficients $\lambda_i (\lambda_i \geq 0)$, representing the relative importance of the objectives. We want to set the weights $(\lambda_i, i = 1, \dots, r)$ so that the solution selected is closest to the new aspiration threshold. Therefore each component in the weight vector is set according to the objective function values. We would give more importance to those objective that have greater differences between the quality of the trial solution and the quality of the reference solution. The influence is given by an exponential function $\exp(-s_i)$, where s_i is obtained as follow

$$\begin{aligned} s_i &= \frac{|f_i(x') - z_i^*|}{|z_i^*|} \\ \lambda_i &= 2 - \exp(-s_i) \\ AFV(x') &= \sum_{i=1, r} \lambda_i \Delta f_i(x') \end{aligned}$$

Note that in the cases of greater differences the value of the AFV is less modified that in the cases of lesser differences, making that the chosen solution is close to the new aspiration threshold.

3.1.7 Frequency-based memory and diversification strategy

Our diversification method employs a frequency memory to encourage the search to unvisited regions or less visited regions. We accomplish this by dividing the range of variables $b - a$ into sub-ranges of equal size as in [14]. The threshold T determines the number of times that one sub-range can be visited without penalizing. A diversifying movement is executed when `residence.freq[j]` is greater than T . We modify the value of $AFV(x')$ as follows: $AFV(x') = AFV(x') - \frac{freq}{freqtotal} \times AFV(x')$, where $freq$ is an addition of the entries of type `residence.freq` associated to the variables and subranges that hold the condition $freq(x') > T$, and $freqtotal = \sum_i freq_i$, we would have for each variable a threshold T . T is equal to $\max\{Round(\sum_{i=1}^{subranges} freq_i / subranges), 1\}$, where $Round$ is the closest integer. The frequency memory is maintained over all iterations and in the two phases of this algorithm.

3.1.8 Pseudo code for our TS approach

Skeleton of our Tabu

Procedure Taboo

```

{  (Initialization step)
  Set  $a$  and  $b$  equal to the lower and upper bounds respectively of the variables
  Generate a feasible solution  $x$  (the midpoint of each interval)
   $z^* = F(x)$  (Setting the reference point)
   $S = x$ 
  for  $i = 1$  to numiter
    newelement = FALSE
    UpdateThreshold
     $x' = \text{Candidate}(x)$ 
    Make_tabudist( $x, x'$ )
     $x = x'$ 
     $z^* = \text{next}z^*$ 
  endfor }

```

Procedure Make_tabudist(x, y)

```

{  (Record the lower and the upper bound of the interval where the variable is excluded)
  tabu[current].lower= $x - w(y - x)$ 
  tabu[current].upper= $x + w(y - x)$ 
  tabu[current].IX= $x$ 
  tabu[current].IY= $y$ 
  (update the current point of tabu list)
  if current = tenure then
    current = 1
  else current = current+1 }

```

Procedure Update_Criterion

```

{  if  $AFV(\text{best}x') < AFV(x')$  then
     $AFV(\text{best}x') = AFV(x')$ 
     $\text{best}x' = x'$  }

```

Procedure Candidate(x)

```

{  tabuflag=TRUE
   $AFV(\text{best}x') = -\infty$ 
  num_tabu=0
  repetition=0
  while tabuflag
    tabuflag = FALSE
    for  $i = 1$  to fan
       $x' = \text{Move}(x)$ 
      if not Is_tabu( $x'$ ) or Aspiration( $x'$ ) then

```

```

        if Aspiration( $x'$ ) then Update_Criterion
        else if  $D(x') \neq \emptyset$  and  $E(x') = 0$  then Update_Criterion
        else
            num_tabu=num_tabu+1
        endfor
    if num_tabu = fan then
        tabuflag=TRUE
        Reduction_tabu_distance
         $x' = \text{Candidate}(x)$ 
    endwhile }

```

We denote E as the set of efficient moves and D the set of deficient moves, where a deficient move is a move that not satisfies the aspiration level, in otherwise the move is efficient. Then, we define the best move as $[m \in E(x') : c(x') = \max\{c(x'), x' = m(x)\}]$ if $E(x') \neq \emptyset$, in the case where $E(x') = \emptyset$ and $D(x') \neq \emptyset$ then, select $[m \in D(x') : c(x') = \max\{c(x'), x' = m(x)\}]$.

In the above algorithm we have the following escape mechanism: when the forbidden moves grow so much that all movements become tabu and none satisfies the aspiration level, a reduction mechanism is activated and the tabu distance in each list is reduced, then the number of forbidden moves is reduced.

Function Variation(x)
 { $UB = \min\{\frac{b-x}{d}, \frac{(x-a)}{d}\}$
 $h = \frac{2UB}{s}$
 variation = $h * d$ }

Function Move(x)
 { Step_size
 move = $x + \text{Variation}$ }

Function Is_tabu(x)
 { (return the boolean value TRUE if, after the movement, the value x
 fall inside one interval bounded by the value lower and upper contained
 in the tabu list. The boolean value is FALSE otherwise)
 Is_tabu = FALSE
 if iter \leq list_size
 then point = 1
 else point = current+1
 while point \neq current
 (x can be a portion of the decision variable)
 if tabu[point].lower $< x <$ tabu[point].upper then
 Is_tabu=TRUE
 break
 if point = list_size then

```

    point = 1
    else point = point+1
endwhile }

```

Procedure Aspiration(x)

(Return the boolean value TRUE if the function values, after the movement, permit to be accepted the point as probably non-dominated point)

```

newelement = FALSE
for  $i = 1$  to  $r$ 
    delta[i] =  $f_i[x_i] - z_i^*$ 
if ( $\exists$  delta[i] = preference  $i \in \{1, \dots, r\}$ ) or
( $\forall i \in \{1, \dots, r\}$  [delta[i] = indifference])
    then
         $X = X + x'$ 
         $Obj = F(X)$ 
        newelement = TRUE
        Update_freq(x)
        if  $\text{freq}(x') - T > 0$  then  $AFV(x') = \text{Penalize}(AFV(x'))$ 
        next_ $z^* = \max\{f_i(x')\}; x' \in X$ 
    endif
endif }

```

Procedure Reduction_tabu_distance(x)

{(Change the bounds in the tabu list)

```

w=w/2
for all  $i$  belonging to tabu list
    tabu[i].lower=tabu[i].IX - w(tabu[i].IY-tabu[i].IX)
    tabu[i].upper=tabu[i].IX + w(tabu[i].IY-tabu[i].IX) }

```

4 Multiobjective scatter search “MOSS”

4.1 Overall view

Scatter Search operates on a set of solutions, the reference set, by combining these solutions to create new ones. In our approach the mechanism for combining solutions is such that a new solution is created from a linear combination of two other solutions.

We define the following sets:

X : a set of trial points, from which all others sets derive.

R : a set of current reference solutions, constituted by non-dominated elements of S .

$R1$: a set of high-quality non-inferior solutions, composing a subset of R .

$R2$: a difference set between R and $R1$.

- T : a set of tabu solutions, composing a subset of R excluded from consideration to be combined as first solution in a convex combination.
- $T1$: a set of tabu solutions, composing a subset of R excluded from consideration to be combined during t generation as second and third elements in the subset combined.
- P : a pareto set, constituted of non-inferior solutions of S .

The reference set is constructed with the union of $R1$ and $R2$, where $R1 = C^k$ and $R2 = P \setminus C^k$.

Scatter Search has three main loops: 1) a “for loop” that controls the maximum number of iterations, 2) a “while loop” that monitors the presence of new elements in the reference set and 3) a “for loop” that controls the examination of all subsets that hold the tabu restriction imposed.

As a basis for creating combined solutions we generate subsets x' and for each x' use a solution combined method, generating solution in a line. We use weights to sample points from the line.

Avoiding the duplicated strategies already generated can be a significant factor in producing an effective overall procedure. The control is limited to these solutions that hold the condition of being Pareto. Our algorithm is provided by “critical event design” that monitors the current solutions in R and in the combined set $\Omega(x')$. The elements considered in the critical events design are the values of the objectives, and the decision variables of the current Pareto solutions generated. The new solutions are put in the set $\Omega(x')$ if they do not belong to the set of critical events. If all subsets have been examined then, the algorithm subtracts the non-dominated solutions $\Omega^P \subseteq \cup \Omega(x') \setminus C$, where C is a critical event set. New elements are incorporated into R if $|P \cap R| < |P|$ where, $P \subseteq \Omega^P \cup R$.

4.2 Choosing subset of reference points

Our approach is organized to generate two different collections of subsets of R , which we refer to as subset1 and subset2, see [12].

The type of subsets we consider are as follow:

subset1: 3-elements subsets of $R1$ where the second and third elements are the most dissimile to the first.

subset2: 3-elements subsets derived from the two first elements of a subset1 by augmenting it to include the most diverse solution relative to it from $R2$.

4.3 Memory structures

Let `S_size` denote the current number of solutions in the trial point set. `S_size` begins at 0 in an initialization step and is increased each time a new solution is added to the trial points set, until reaching the value (`max_size - cardR`) or all subsets of reference points were examined.

At each step, S stores the current trial solution in `X[trial[i]].x`, the values of the objective functions are recorded in `X[trial[i]].F` where `F` contains several fields (f_1, \dots, f_r).

$X[\text{trial}[i]].T$ is a boolean variable identifying that the solution was combined in a previous iterations as first element in the subset combined. $X[\text{trial}[i]].T2$ records the time that this solution will be forbidden as second and third solution in a subset combined . A solution x is not permitted to be recorded if it duplicates another solution already in R . We speed the check to see if one solution duplicates another by keeping information about the objective functions and the decision variables. If one solution is duplicated, it is eliminated.

Several associated location are defined: $\text{pareto}[i]$, $i = 1$ to psize , indicates the position in S of non-inferior current solutions, that is, $X[\text{pareto}[i]]$ is a current pareto solution, $\text{kramer}[i]$, indicates the best non-inferior solutions, $\text{refset1}[i]$ $i = 1$ to cardR1 , the high quality current Pareto solutions and $\text{refset2}[i]$ $i = 1$ to cardR2 , that take a role as a diverse subset of current non-inferior solutions.

4.4 Pseudo code scatter search

Procedure Scatter

```

{ Set  $R = \emptyset$ 
  Set  $C = \emptyset$ 
  generation=0
  For 1 to MaxIter
    (Generate an initial reference set of trial points)
    Taboo
    (SS Phase)
     $P = \text{ParetoSet}(S)$ 
    if  $|P| > 1$  then
       $R1 = \text{KramerChoice}(P)$ 
       $R2 = P \setminus \text{KramerChoice}(P)$ 
       $R = R1 \cup R2$ 
    iter=0;
    (In our implementation we take MaxGeneration = 2)
    while( $0 < \text{MaxGeneration}$ )or( $ratio \geq \beta$ )
      Calculate the number of MaxSubset that includes at least
      one new element
      For  $counter = 1$  to MaxSubset
        Generate the new subset  $x'$  from  $R$  with a number of
        elements ranging from 2 to 3(optional)
        Apply line search approach to obtain a set  $\Omega(x')$ 
        of new solutions
      EndForMaxsubset
      Obtain  $\Omega^P \subseteq \Omega(x') \setminus C$ 
      ( $\Omega^P$  is the current Pareto set)
      If  $|P| = 1$  then Break
       $ratio = |P \cap R| / |P|$ 
       $R1 = \text{KramerChoice}(P)$ 

```

```

    R2 = P \ KramerChoice(P)
    R = R1 ∪ R2
    generation = generation + 1
    MaxGeneration = MaxGeneration - 1
  endwhile
endif
endMaxiter }

```

Procedure Sigma

```

{   for i = 1 to S_size
    X[trial[p]].sum =  $\sum_{j=1..r} X[\text{trial}[p]].f[j]$  }

```

Procedure Is_Pareto

```

{   if  $\exists i = 1..psize, j = 1..r : X[\text{trial}[p]].f[j] < X[\text{pareto}[i]].f[j]$ 
    then Is_Pareto=true
    else Is_Pareto=false }

```

Procedure Critical_Condition

```

{   if X[trial[p]].F = X[pareto[i]].F then
    Critical_Condition=true
    else Critical_Condition=false }

```

Procedure Is_Critical

```

{   Is_Critical=false
    for i=1 to psize
      if X[pareto[i]].sum = X[trial[i]].sum
        if Critical_Condition then
          Is_Critical=true
          break }

```

Procedure Non_Dominated

```

{   for p=2 to S_size
    if not Is_critical(trial[p] and Is_Pareto(trial[p])) then
      psize=psize+1
      pareto[psize]=trial[p] }

```

Procedure Pareto

```

{   Sigma
    Sort
    X[pareto[1]].F = X[trial[1]].F
    psize=1
    Non_dominated }

```

Procedure KramerChoice

```

{ (Records the index of the selected solution)

```

array and the cardinal of the choice set in size)

```

Q = r + 1
i = 1
while (i ≤ cardR)
  QI[i] = 0
  j = 1
  while (j ≤ cardR)
    d = Discordance(R[i].F(x), R[j].F(x))
    QI[i] = max(d, QI[i])
    j = j + 1
  if QI[i] < Q then
    Q = QI
    size = 1
    selection[size] = i
  else
    size = size + 1
    selection[size] = i
  endwhile
endwhile
KramerChoice = selection }

```

Combining Strategy

Let $\Omega(X) = x + \varpi(x' - x)$, for $\varpi = 1/2, 1/3, 2/3, -1/3, -2/3, 4/3, 5/3$, and $x, x' \in X$. The most dissimilar solution relative to x' is measured with the max-min criterion, that is, maximize the minimum value of a dissimilarity measure. We take as dissimilarity measure the Euclidean measure.

The strategy to create the Ω set is as follows:

Let x be the first element choice, x' the second and x'' the third, the selection of each element is made with the max-min criterion then, we apply the following rules:

1. Generate new trial points on lines between x and x' .
2. Generate new trial points on lines between x'' and x .
3. Generate new trial points on lines between x'' and x' .
4. Generate one solution by applying $x' + \frac{x-x'}{2}$.
5. Generate one solution by applying $x'' + \frac{x-x''}{2}$.
6. Generate one solution by applying $x'' + \frac{x'-x''}{2}$.

In the cases 4, 5, and 6 generate new trial points on lines that join each reference point with one of the trial points generated using convex combination methods. In this case the search relies searches star from middle trial points. If we have only two points initially then, the third point is obtained by linear combination of this two points.

In the special case where $C^k = 1$ then, the approach begins by selecting the first element pertaining to $R1$, denoting it by x , the second element is selected from $R2$, denoting it by x' , if the cardinal of $R2$ is greater than 1 then, the third element is selected from $R2$ to maximize its distance from the first and second elements, denoting it by x'' , in other case the third element is created by a linear combination of the two elements selected above, and then to apply the above scheme of combination explained.

Procedure Combine

```
{ (Create subset type 1 and 2, and combine)
  for  $i = 1$  to cardR1
    if not R[refset1[i]].T then
      Let R[refset1[p]].x =  $x \in x'$ 
      Let the most dissimile R[refset1[p]].x the second element in  $x'$ 
      Set R[refset1[p]].T1=generation+tenure
      Put the most dissimile R[refset1[p]].x relative to  $x'$  the third element  $z$ 
      Create  $\Omega(x' + z)$ 
      Set R[refset1[p]].T1=generation+tenure
      Set R[refset1[i]].T=TRUE
      if cardR2  $\neq 0$  then Combine2
    endfor }
```

Procedure Combine2

```
{ (Create subset type 2)
Put the most dissimile R[refset2[p]].x relative to  $x' \in x'$ 
Create  $\Omega(x')$ 
Set R[refset2[p]].T1=generation+tenure }
```

4.5 Restarting solutions

In order to build a new set S in the next iteration, we generate a new seed point using the history of the search looking for the subranges unless visited taking the middle points of these subranges as components of the new seed point, and then use the diversification generator. Initialise the generation process with the solutions currently in R .

5 Computational experiments

In this section we will examine the performance of our algorithm on several test problems with one, two and three variables since these may be plotted.

5.1 2-objectives test problems

5.1.1 Problem 1 Schaffer's F2

First we examine the algorithm convergence properties and the diversity of the solutions in the frontier using Schaffer's F2 function [21].

$$f_1(x) = x^2 \quad \text{and} \quad f_2(x) = (x - 2)^2$$

This is a simple function with a single decision variable and two objectives to be minimized. The Pareto front is where the tradeoff exits between the two functions. That is, for $0 \leq x \leq 2.00$, one of the functions is decreasing while the other is increasing.

The problem was solved on the domain $-100 \leq x \leq 100$. Figure 1(a) shows the first approximation to the Pareto set in the first generation obtained from tabu search, figure 1(b) shows that the points were well distributed over all Pareto frontier in this generation.

F2 is an easy problem for our MOSS, the initial reference solutions contains several solutions on the frontier. The solutions quickly converge to Pareto set and distributed over the Pareto frontier very well.

Figure 1(a) shows the last approximation to Pareto set in the fourth generation obtained from Scatter search, figure 1(b) shows that the points were maintained over all Pareto Front in all generations.

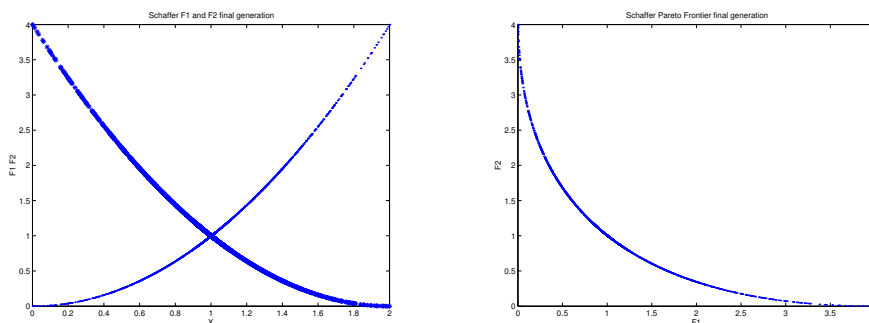


Figure 1: (a) Final approximation to Pareto Set. (b) Final approximation to Pareto Frontier.

5.1.2 Problem 2 Schaffer's F3

$$f_{31} = \begin{cases} -x & \text{if } x \leq 1, \\ -2 + x & \text{if } 1 < x \leq 3, \\ 4 - x & \text{if } 3 < x \leq 4, \\ -4 + x & \text{if } 4 < x. \end{cases} \quad f_{32} = (x - 5)^2$$

This problem has two separate Pareto-Optimal set in the ranges $[1, 2]$ and $[4, 5]$. Our MOSS obtain exactly either ranges and a widely distribution points in the Pareto set. The result is shown in the figure 2.

5.1.3 Problem 3 Multiple Local-Optimal

Next we compare our approach using the following function below [20].

$$f_1(x) = \sin(x) * (1 + x/20)$$

$$f_2(x) = \cos(x) * (1 + x/20)$$

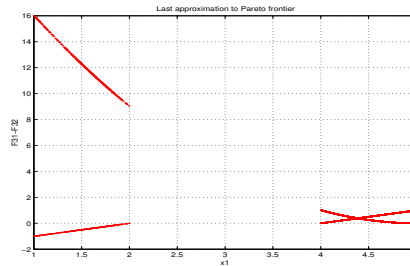


Figure 2: Final approximation to Pareto Set.

This is a function with a single decision variable and two objectives, to be minimized. It has three local Pareto optimal regions in the domain $0 \leq x \leq 20$, the right most completely dominating the other two. Our approach obtained the global Pareto-optimal. The final Pareto set is shown in figure 3(a), the figure 3(b) shows the final Pareto frontier. More than 2000 solutions were obtained very quickly obtaining a well defined Pareto frontier.

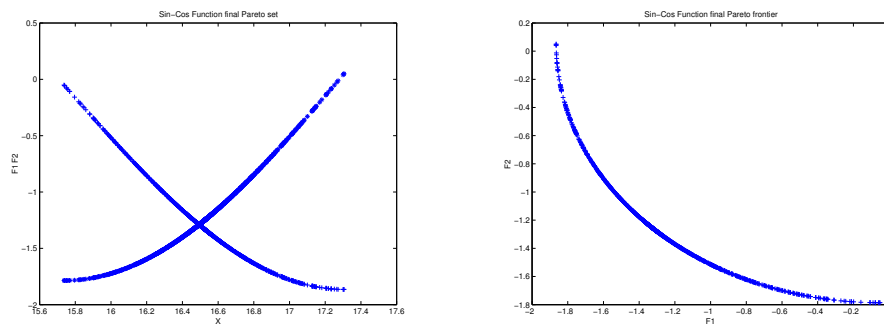


Figure 3: (a) Final approximation to Pareto set. (b) Final approximation to Pareto Frontier.

5.1.4 Problem 4 Bi-Objectives with two variables

Now, let's consider the next simple bi-objective problem of simultaneously minimizing [4]:

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 + x_2^2 \\ f_2(x_1, x_2) &= (x_1 - 5)^2 + (x_2 - 5)^2 \end{aligned}$$

in a region of the objective space, defined by $-5 \leq x_1 \leq 10$ and $-5 \leq x_2 \leq 10$. The objective function has its minimal points at $(0, 0)$ and $(5, 5)$ respectively. The figure 4 shows the Pareto-optimal frontier. Observe that was obtained a widely Pareto frontier.

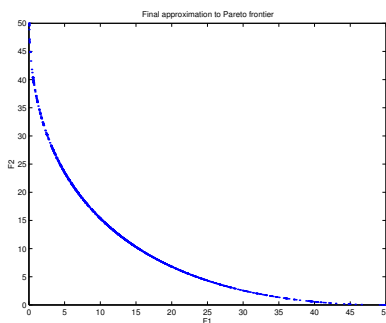


Figure 4: Bi-objectives with two variables final approximation to Pareto Frontier.

5.1.5 Problem 5 Objective with one and three variables

We studied two new examples [15]. The first is the following function:

$$f_1(x) = 3 \cos(x)$$

$$f_2(x) = 2 \sin(x)$$

The second is the following:

$$f_1(x_1, x_2, x_3) = 3 \cos(x_1) \prod_{i=2,3} x_i$$

$$f_2(x_1, x_2, x_3) = 2 \sin(x_1) \prod_{i=2,3} x_i$$

The range of $x_1 \in (-\pi, \pi]$ and the range of all the other variables are $[0, 1]$. The goal of the process is to maximize the objectives and to obtain a set of points spread along the Pareto set. Figure 5(a) shows the last approximation to Pareto frontier in the case of one variable. The figure 5(b) shows the last approximations to Pareto frontier with three variables. Observe that if $x_2 = 1$ and $x_3 = 1$ the Pareto frontier will be the same as in the case of one variable. In the last case with three variables, our Pareto frontier was approximately the same as in the case of one variable showing that was obtained a good converge to the Pareto frontier.

5.1.6 Problem 6 Himmelbau's Function

The function has four equal height peaks, this function has been used by several authors [20].

$$f_1(x, y) = 5 - ((x^2 + y - 11)^2 + (x + y^2 - 7)^2)/200$$

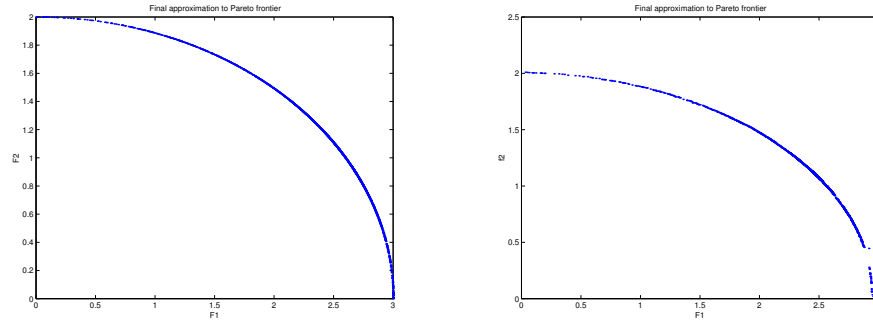


Figure 5: (a) Last approximation to Pareto set. (b) Final approximation to Pareto Frontier.

$$f_2(\alpha, \beta) = f_1, \text{ where } \alpha = 2x, \beta = 2y$$

The decision variables x_1 and x_2 ranges in the interval $[-5, 5]$. In this case, our current approach obtained a good distributed Pareto frontier but only one local Pareto-optimal set. The figures 6(a) and 6(b) show the final approximation to Pareto set and Pareto frontier.

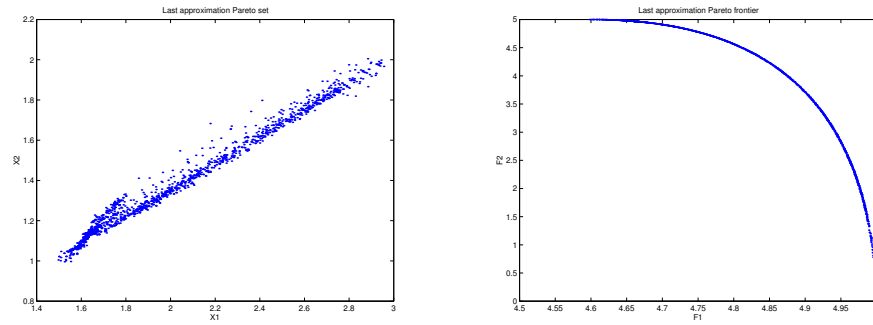


Figure 6: (a) Final approximation to Pareto set. (b) Final approximation to Pareto Frontier.

5.1.7 Problem 7 Sin-Sin Function

Now we study the following function described in [6]:

$$f_1(x_1, x_2) = \sin(x_1^2 + x_2^2 - 1)$$

$$f_2(x_1, x_2) = \sin(x_1^2 + x_2^2 + 1)$$

where x_1 and x_2 ranges in the interval $[0, 3\pi/4]$. Here, the function f_1 take your maximum value in a point $x_1^2 + x_2^2 = \pi/2 + 1$ and f_2 in the point $x_1^2 + x_2^2 = \pi/2 - 1$. Figures 7(a) and 7(b) show the final approximation to a widely diversity Pareto set and a well defined Pareto frontier respectively.

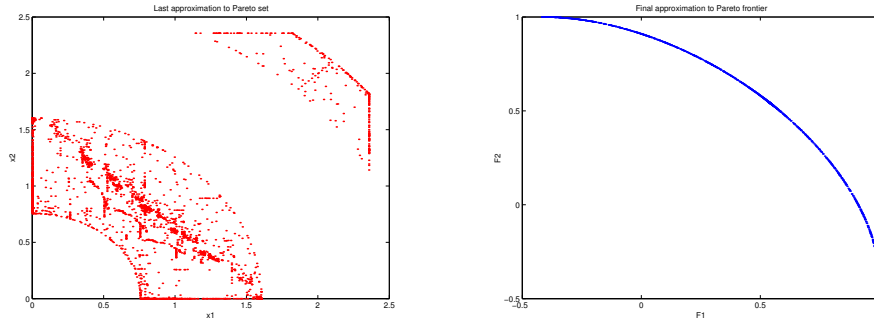


Figure 7: (a) Final approximation to Pareto set. (b) Final approximation to Pareto Frontier.

5.1.8 Problem 8 n-Objective FON Function

The following function shown below has the Pareto set constituted with all solutions in $x_i^* \in [-1, 1]$ for all i (see figure 8(a)).

$$\begin{aligned} f_1(x) &= 1 - \exp(-(x_1 - 1)^2) - (x_2 + 1)^2 \\ f_2(x) &= 1 - \exp(-(x_1 + 1)^2) - (x_2 - 1)^2 \\ -2 \leq x_i \leq 2, \quad i = 1, 2 \end{aligned}$$

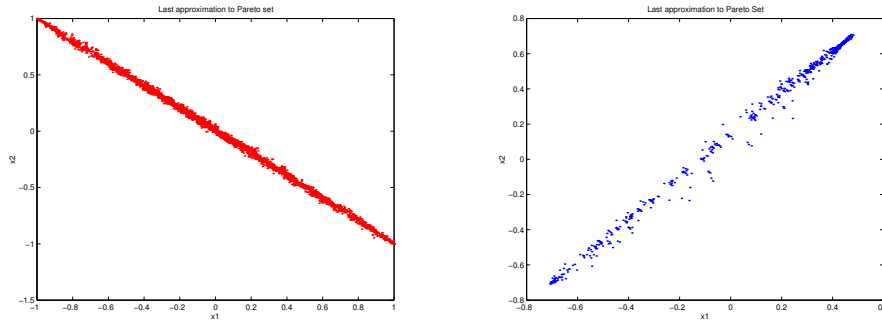


Figure 8: (a) Final approximation to Pareto Frontier for Problem 8. (b) Final approximation to Pareto set for Problem 9.

5.1.9 Problem 9(a) n-Objective FON1 Function

The following function shown below has $x_i^* = -1/\sqrt[n]{n}$ for all i as the minimum solution for f_1 and $x_i^* = 1/\sqrt[n]{n}$ for all i as the minimum solution for f_2 . The Pareto set is constituted with all solutions in $x_i^* \in [-1/\sqrt[n]{n}, 1/\sqrt[n]{n}]$ for all i .

$$f_1(x) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i - \frac{1}{\sqrt[n]{n}}\right)^2\right)$$

$$f_2(x) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i + \frac{1}{\sqrt[2]{n}}\right)^2\right)$$

$$-4 \leq x_i \leq 4 \quad i = 1, 2, \dots, n$$

The Pareto set in the above problem is constituted with all solutions in $x_i^* = [-1/\sqrt{n}, 1/\sqrt{n}]$, in our case our approach obtained a well distributed set of points in this range for $n = 2$. Figure 8(b) shows the last approximation to Pareto Set with two variables.

5.2 3-objectives test problems

5.2.1 Problem 9(b) 3-Objective Ellipsoid Function

This is a 3-objective function of two variables to be maximize. The representation of this multiobjective problem in the objective space is an ellipsoid, in this case the Pareto set in the objective space covers 1/8th part of the total space size, see[15].

$$f_1(x_1, x_2) = 3 \cos(x_1) \cos(x_2)$$

$$f_2(x_1, x_2) = 2 \cos(x_1) \sin(x_2)$$

$$f_3(x_1, x_2) = 2 \sin(x_1)$$

with $x_1 \in [-\pi/2, \pi/2]$ and $x_2 \in (-\pi, \pi]$.

The following application is an ellipsoid function with 4 variables, where the variables $x_i \in [0, 1]$ for $i = 3, 4$.

$$f_1(X) = 3 \cos(x_1) \cos(x_2) \prod_{i=3,4} x_i$$

$$f_2(X) = 2 \cos(x_1) \sin(x_2) \prod_{i=3,4} x_i$$

$$f_3(X) = 2 \sin(x_1) \prod_{i=3,4} x_i$$

Figure 9(a) shows last approximation to Pareto frontier, obtaining a widely Pareto frontier. Figure 9(b) shows the last approximation to Pareto frontier with four variables. Observe that in this case the Pareto frontier is the same as in the above case when $x_3 = x_4 = 1$. In our case either frontier obtained are approximately the same.

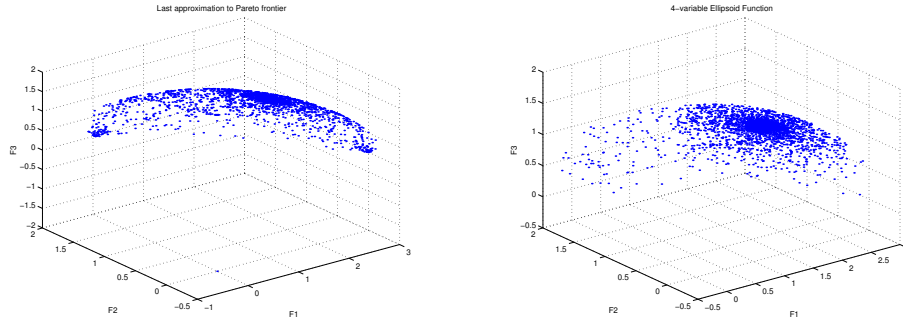


Figure 9: Final approximation to Pareto Frontiers.

5.2.2 Problem 10 3-Objective Sphere Function

The following test problem have an objective search space where the Pareto frontier is constituted by an spherical surface of unit radio.

$$\begin{aligned}
 f_1(x_1, x_2, x_3) &= (1 + x_3) \cos(x_1) \cos(x_2 + \pi/4) \\
 f_2(x_1, x_2, x_3) &= (1 + x_3) \cos(x_1) \sin(x_2 + \pi/4) \\
 f_3(x_1, x_2, x_3) &= (1 + x_3) \sin(x_1) \\
 0 &\leq x_1 \leq \pi/2 \\
 -\pi/4 &\leq x_2 \leq \pi/4 \\
 0 &\leq x_3 \leq 1
 \end{aligned}$$

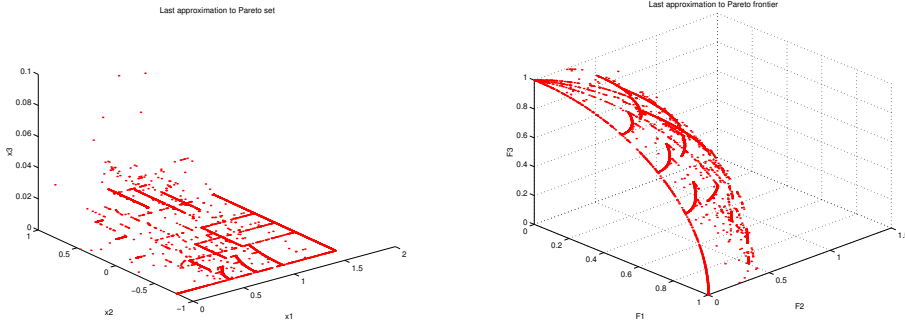


Figure 10: (a) Final approximation to Pareto set. (b) Final approximation to Pareto frontier.

Figures 10(a) and 10(b) show the last approximation to Pareto set and Pareto frontier respectively. The real Pareto set for the above problem is as follows [7]: $0 \leq x_1 \leq \pi/2$, $-\pi/4 \leq x_2 \leq \pi/4$, $x_3 = 0$, and the Pareto frontier is the unit sphere in the first quadrant.

Introducing difficulties across the Pareto frontier

The following is the sphere function, but note that the variables are mapped to the meta-variable vector θ .

$$\begin{aligned}
 f_1(\theta, r) &= (1 + r) \cos(\theta_1) \cos(\theta_2) \\
 f_2(\theta, r) &= (1 + r) \cos(\theta_1) \sin(\theta_2) \\
 f_3(\theta, r) &= (1 + r) \sin(\theta_1) \\
 \theta_i &= \frac{\pi}{2} x_i, \quad i = 1, 2 \\
 0 &\leq x_i \leq 1, \quad i = 1, 2 \\
 r &= x_3^2 \text{ and } -1 \leq x_3 \leq 1
 \end{aligned}$$

The Pareto frontier occurs at $x_3^* = 0$ and the function values satisfy the following condition: $\sum (f_i^*)^2 = 1$, [7]. Figures 11(a) and 11(b) show the approximation to Pareto set and Pareto frontier respectively.

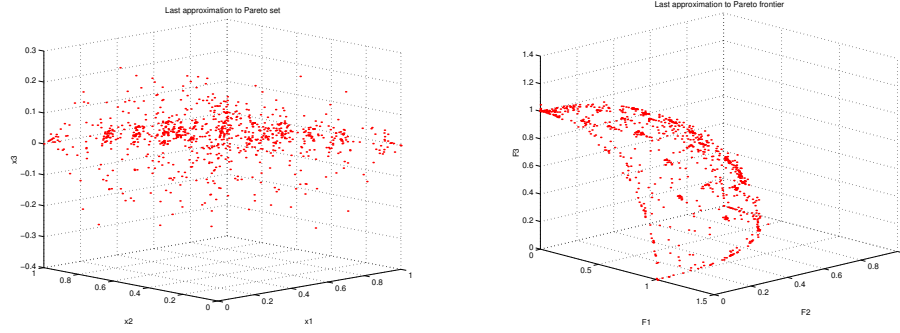


Figure 11: (a) Final approximation to Pareto set. (b) Final approximation to Pareto frontier.

5.2.3 Problem 11 3-Objective DTLZ7 Function

This problem has a disconnected set of Pareto-optimal regions. Minimize the following function:

$$\begin{aligned} f_1(x_1) &= x_1, \\ f_2(x_2) &= x_2, \\ f_3(X_3) &= (1 + g(X_3))h(f_1, f_2, g), \end{aligned}$$

where $g(X_3) = 1 + \frac{9}{2} \sum_{i=3,4} x_i$, $h(f_1, f_2, g) = 3 - \sum_{i=1}^2 [\frac{f_i}{1+g}(1 + \sin(3\pi f_i))]$, subject to $0 \leq x_i \leq 1$, for $i = 1, 2, 3, 4$.

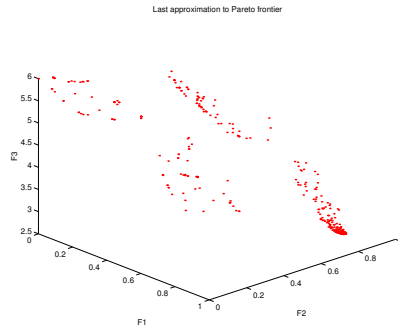


Figure 12: Final approximation to Pareto frontier.

This test problem has 4 disconnected Pareto-optimal regions in the objective space. The functional g has in our prove 2 decision variables and the total number of variables was 4. The Pareto-optimal solutions corresponds to $X_3 = 0$. This test an algorithm's ability to maintain subsets in different Pareto-optimal regions. The result is show in the figure 12, it shows that the algorithm was able to find stable and distributed subsets in all four disconnected Pareto-optimal regions.

5.2.4 Problem 12 3-Objective Comet Function

This function has the comet-like Pareto frontier. Starting from a widely spread region, the Pareto frontier continuously reduces to thinner region [7]. This is a hard problem for any multiobjective evolutionary algorithm.

Minimize the following function:

$$\begin{aligned} f_1[x_1, x_2, x_3] &= (1 + x_3)(x_1^3 x_2^2 - 10x_1 - 4x_2) \\ f_2[x_1, x_2, x_3] &= (1 + x_3)(x_1^3 x_2^2 - 10x_1 - 4x_2) \\ f_3[x_1, x_2, x_3] &= 3(1 + x_3)x_1^2 \\ &1 \leq x_1 \leq 3.5 \\ &-2 \leq x_2 \leq 2 \\ &0 \leq x_3 \leq 1 \end{aligned}$$

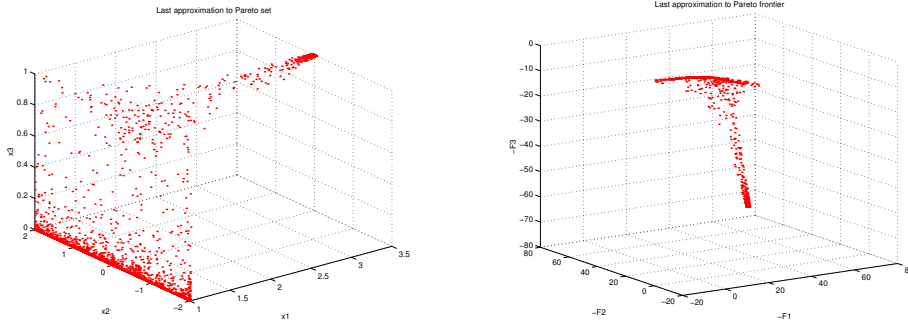


Figure 13: (a) Final approximation to Pareto set. (b) Final approximation to Pareto frontier.

The Pareto-optimal surface corresponds to $x_3^* = 0$ and for $-2 \leq x_1^{*3} x_2^{*2} \leq 2$ with $1 \leq x_1^* \leq 3.5$. The figures ?? and 13 show the last approximation to Pareto set and Pareto frontier respectively. It was obtained a distributed approximation to Pareto frontier but the approximation to Pareto set still not satisfy our expectations.

Our experiments were run with the following parameters: maxiter=100, maxgeneration=2, the numbers of iteration in Tabu was 1000, fan=30, $\beta = 0.5$, but in fewer iterations (less than 40) our approach obtain a well defined Pareto frontier.

6 Conclusions

In this paper we have presented an adaptation of the Scatter Search method to continuous vectorial optimization using an adaptation of Tabu Search to continuous variables as a diversification generator. Several test problems to demonstrate the ability of our approach were run, in the first 10 problems the convergence to the Pareto frontier was good, and was obtained a widely Pareto set and Pareto frontier, in the Comet problem the algorithm had trouble, finding few Pareto optimal solutions. We can conclude that the application of

a diversification strategy, the use of a choice function to separate the reference set points and an strategy to combine solutions seems to be a good approach in finding in a few generation a good approximation to Pareto frontier. In a future work we will incorporate new memories structures inside our current approach and another escape mechanisms trying to improve the current methodology, and to apply it to more complex test problems.

Acknowledgements

Thanks are due to Prof. Fred Glover and Dr. Longina Castellano for useful remarks on the first draft of the manuscript.

References

- [1] Battiti, R.; Tecchiolli, G. (1994) “The reactive tabu search”, *ORSA Journal on Computing* **6**(2): 126–140.
- [2] Battiti, R. (1996) “Reactive search: toward self-tuning heuristics”, in: *Modern Heuristic Search Methods*, chapter 4, V.J. Rayward-Smith, I.H. Osman, C.R. Reeves & G.D. Smith (Eds.) John Wiley and Sons: 61-83.
- [3] Beausoleil, R. (2001) “Multiple criteria scatter search”, *Proceedings of 4th Meta-heuristics International Conference*, Jorge Pinho de Sousa (Ed.), Porto, Portugal: 534–539
- [4] Binh, T.T.; Korn, U. (1996) “An evolution strategy for the multiobjective optimization”, Institute for Automation, Magdeburg.
- [5] Coello Coello, C.A. (1999) “A comprehensive survey of evolutionary-based multiobjective optimization techniques”, *Knowledge and Information Systems* **1**(3): 269–308.
- [6] Cvetkovic, D.; Parmee, I.C. (1999) “Use of preference for GA-based multi-objective optimization”, PEDC, University of Plymouth.
- [7] Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. (2001) “Scalable test problems for evolutionary multi-objective optimization”, TIK-Technical Report No. 112.
- [8] Glover, F. (1989) “Tabu search”, part I, *ORSA Journal of Computing* **1**(3): 190–206.
- [9] Glover, F. (1993) “Tabu thresholding: improved search by nonmonotonic trajectories”, *ORSA Journal on Computing*.
- [10] Glover, F.; Laguna, M. (1993) “Modern heuristic techniques for combinatorial problems”, John Wiley & Sons, Inc., chapter 3: 70–147.
- [11] Glover, F. (1994) “Tabu Search for nonlinear and parametric optimization (with links to genetic algorithms)”, *Discrete Applied Mathematics* **40**: 231–255.

- [12] Glover, F. (1998) “A template for scatter search and path relinking”, in: *Artificial Evolution*, Lecture Notes in Computer Science 1363, J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer & D. Snyers (Eds), Springer: 13–54.
- [13] Glover, F. (1999) “Scatter search and path relinking”, in: *New Methods in Optimization*, D. Corne, M. Dorigo & F. Glover (Eds.), McGraw-Hill.
- [14] Glover, F.; Laguna, M.; Martí, R. (s.f.) “Scatter search tutorial for a class of non-linear optimization problem on bounded variables”.
- [15] Grignon, P.; Fadel, G.M. (1999) “Multiobjective optimization by iterative genetic algorithm”, *Proceedings of DETC 99, ASME Design Engineering Technical Conferences*, Las Vegas, Nevada.
- [16] Horn, J.; Nafpliotis, N.; Goldberg, D.E. (1994) “A niched Pareto genetic algorithm for multiobjective optimization”, *ICEC'94, IEEE*: 82–87.
- [17] Keeney, R.L.; Raiffa H. (1976) *Decision with Multiple Objectives: Preference and Value Tradeoffs*. Wiley, New York.
- [18] Knowles, J.; Corne, D. (1999) “The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimization”, University of Reading, UK.
- [19] Makarov, I. M.; Vinogradskaja, T. M.; Rubinski, A.A; Sokolov. V. B. (1982) *Choice Theory and Decision Making*. Nauka, Moscow (in Russian).
- [20] Molyneaux, A. K.; Leyland, G.B.; Favrat, D. (2001) “A new clustering evolutionary multi-objective optimisation technique”, *Proceeding ISAS2001, Evolutionary Computation and Probabilistic Graphical Models*, Havana, Cuba: 41–43.
- [21] Schaffer, J.D. (s.f.) *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Unpublished doctoral dissertation, Vanderbilt University.
- [22] Sen, P; Yang, J.B. (1998) *Multiple Criteria Decision Support in Engineering Design*. Springer-Verlag.
- [23] Vol'ski V. I. (1982) “Application of the Kramer method to identifying part of a Pareto set in multi-criteria optimization”, *Automation and Remote Control* **12** (in Russian): 111–119.
- [24] “Genetic algorithms with gender for multi-function optimization”, EPCC-SS92-01.