NEW RESULTS WITH SCATTER SEARCH APPLIED TO MULTIOBJECTIVE COMBINATORIAL AND NONLINEAR OPTIMIZATION PROBLEMS

Ricardo P. Beausoleil[*]

Abstract

This paper introduces two variants of a multiple criteria scatter search to deal with nonlinear continuous and combinatorial problems, applying a tabu search approach as a diversification generator method. Frequency memory and another escape mechanism are used to diversify the search. A Pareto relation is applied in order to designate a subset of the best generated solutions to be reference solutions. A choice function called Kramer Choice is used to divide the reference solution in two subsets. Euclidean and Hamming distances are used as measures of dissimilarity in order to find diverse solutions to complement the subsets of high quality current Pareto solutions to be combined. Linear combination and path relinking are used as a combination methods. The performance of these approaches are evaluated on several test problems taken from the literature.

Resumen

Este artículo introduce dos variantes de búsqueda dispersa multiobjetivo para problemas continuos y combinatorios, aplicando un enfoque de búsqueda tabú como un método generador de diversificación. Una memoria de frecuencia y otros mecanismos de escape para diversificar la búsqueda son utilizados. La relación Pareto es aplicada para designar un subconjunto de las mejores soluciones como conjunto de soluciones de referencia. Una función de selección llamada selección de Kramer es usada para dividir las soluciones de referencia en dos subconjuntos. Las distancias Euclidianas y Hamming son utilizadas como medida de desemejanza para hallar soluciones diversas

[*]Instituto de Cibernética Matemática y Física, Departamento de Optimización, Ciudad Habana,Cuba. E-Mail: `rbeausol@icmf.inf.cu`

como complemento de las soluciones actualmente Pareto a ser combinadas. Combinaciones lineales y reencadenamiento de trayectorias son usadas como métodos de combinaciones. El desempeño de estos enfoques es evaluado sobre varios problemas de prueba tomados de la literatura.

Palabras clave: Objetivos múltiples, metaheurísticas, búsqueda tabú, búsqueda dispersa, optimización no lineal.

Mathematics Subject Classification: 90C29, 90B50.

1 Introduction

The solving of multiobjective problems has been a continuing effort by researchers of different disciplines. Many powerful techniques for solving difficult single objective problems have been developed. Their fundamental algorithmic structures can also be applied to solving many multiobjective problems. Among these algorithms raise the multiobjective metaheuristics.Different techniques appears in the literature, for example Vector Evaluated Genetic Algorithm (VEGA) (Schaffer,1985), Multiple Objective Genetic Local Search (MOGLS) (Jaszkiewicz,1998),Micro Genetic Algorithm (Coello, 2001), NSGA-II (Deb et al., 2000), SPEA2 (Zitzler, 2001). This paper presents two variants of a multiobjective scatter search, as extensions of (Beausoleil, 2001, 2002, 2003) applied to a multiobjective job shop problem and nonlinear continuous problems. An adaptation of tabu search for nonlinear optimization to a multiobjective environment is used to generate an initial set of diverse non-inferior solutions. Also this adaptation is extended to incorporate a scatter search approach in order to improve these solutions toward the Pareto frontier.The proposed algorithms incorporates strategies for diversification in order to obtain a widely spread of solutions in the objective space and choice function to split the reference set in two subset of Pareto solutions.

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

2 Notation and general methods

Formally, we can state a quantitative decision making problem 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 $f_1, f_2, \ldots, f_r$, defined over a set of situations $X \times \Re$, where $\Re$ is a finite set of uncertain factors values. Then for each situation (x, σ), where $x \in X$ and $\sigma \in \Re$, we have a vector function $F(x, \sigma) = (f_1(x, \sigma), f_2(x, \sigma), \ldots, f_r(x, \sigma))$. 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 $f_k(x) \geq f_k(x'), \forall k$ and $f_k(x) > f_k(x')$ for at least one objective k). 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 refereed to as the Pareto Optimal Set. The set of all non-dominated points in the objective space is refereed 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" [13].

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

2.3   Referenced sets

We define the following sets:

S a set of trial solutions, from which all others sets derive.

P a Pareto set, constituted of non-inferior solutions of S.

R_1 a set of high-quality non-inferior solutions subset of P.

R_2 a difference set between P and R_1.

R a set of current reference solutions, constituted by the union of R_1 and R_2.

T an ordered set of tabu solutions, composing a subset of R excluded from consideration to be combined during t generation.

T_1 an ordered set of tabu solutions, composing a subset of R excluded from consideration to be combined during $t1$ generation .

Y a diverse subset of the set R.

$\Omega(Y)$ a set of combined solutions, created from a given set Y.

Also, C^K is the Kramer choice function and $C^K(P)$ is the set of selected elements of P. The reference set is constructed with the union of R_1 and R_2, where $R_1 = C^K(P)$ and $R_2 = P \setminus C^K(P)$. Let b be the larger size of the trial solutions set, b_1 the larger size of the reference set, and b_2 the larger size of Pareto-know set.

2.4   Search by goals

Our implementations use as move attributes the variables that change their values as result of the move. We represent change represented by a difference of values $f_k(x') - z_k^*$, $\forall k = 1..r, x' \in X$ where x' was generated from x by a recent move, x is a current solution and Z^* is a reference point, $Z^* = (z_1^*, \ldots, z_r^*)$.

A thresholding aspiration is used to obtain an initial set of trial solutions S as follows: without loss of generality, assume that every criteria is maximized. Notationally, let $\Delta f(x') = (\Delta f_1(x'), \ldots, \Delta f_r(x'))$ where $\Delta f_k(x') = f_k(x') - z_k^*, k \in \{1, \ldots, r\}$.

Let

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

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

The point Z^* is updated, in each iteration of tabu search, by $z_k^* = \max f_k(x')$, $\forall k \in \{1, \ldots, r\}, x' \in S$.

2.5   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 weights λ_k ($\lambda_k \geq 0$), representing the relative importance of the objectives. We want to set the weights $(\lambda_k, k = 1..r)$ so that the solution selected is the closest to the new aspiration threshold. Therefore each component in the weights vector is set according to the objective function values. We would give more importance to those objectives 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 $\exp(-s_k)$, where s_k is obtained as follows

$$s_i = \frac{|f_k(x') - z_k^*|}{|z_k^*|}$$

$$\lambda_k = 2 - \exp(-s_k)$$

$$afv(x') = \sum_{k=1,r} \lambda_k \Delta f_k(x')$$

if $z_k^* = 0$ then we take only the absolute value.

Note, that the weight grows while the distance between $f_k(x')$ and z_k^* grows, then if the difference $\Delta f_k(x') \geq 0$; $\lambda_k \Delta f_k(x')$ yields an incentive, otherwise $\lambda_k \Delta f_k(x')$ creates a penalty, producing a bias to those solutions that have more incentive and less penalized.

3   Multiobjective non-linear scatter search approach

3.1   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.

By the standard TS approach, a move is classified tabu and excluded from consideration if it reverses a recent previous move. We apply an adaptation of tabu search strategies for preventing move reversals in nonlinear context (see [10]). 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. Our approach also uses recency and frequency based memory for diversification [11].

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 [10].

3.1.1   Neighborhood (move description)

In our implementation we propose to select moves that consist of changing at most five variables. We use a controlled randomization and frequency memory to select the move-variables if the problem has more than five variables. In a phase of intensification we move only two selected variables. The range of variables are split into subranges, frequency memory is used to control the random selection of the subranges. In our approach a standard move is defined as follows:

First we divide the range of each variable $U_i - L_i$ into $\hat{b}$ sub-range $a_{ij} - b_{ij}$ of equal size (see [12]), then the components of the trial solution $x' = m(x)$ are as follows:

$$x'_i = \left\{ \begin{array}{ll} r_i & \text{if } i \in V, \\ x_i & \text{otherwise} \end{array} \right.$$

$$r_i \in [a_{ij}, b_{ij}], i \in V, j \in B$$

where V is the index set of the variables randomly selected to move and B is the index set of the randomly selected sub-ranges where the variables take their values, r_i is a random number.

3.1.2   Tabu status

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 of values in going from one solution to another [10]. 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 when the forbidden moves grow to the point 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.

We use a tenure of 10 for the tabu memory structure and one tabu list for each variable.

3.1.3 Candidate list strategy

We use a simplified version of a sequential fan strategy as a candidate list strategy. The sequential fan generates p best alternative moves at a given step, and then creates 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 one stream and several points to consider.

3.1.4 Frequency-based memory and diversification strategy

Our diversification method employs frequency memories to encourage the search move into unvisited regions or less visited regions. We accomplish this by dividing the range of variables $b_{ij} - a_{ij}$ into sub-ranges of equal size as in [12]. The threshold T_i determines the number of times that one sub-range can be visited without penalizing. A diversifying move is executed when $freq(ij)$ is greater than T_i, where $freq(ij)$ is the number of time that the variable x'_i takes value in the sub-range j. 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 $freq(ij)$ associated to the variables and subranges that hold the condition (a): $freq(ij) > T_i$, and $freqtotal = \sum_i \sum_j freq(ij)$ for each variable that hold the condition (a) and for all sub-range j. We have for each variable a threshold T_i, where

$$T_i = \max\{Round(\sum_{j=1}^{subranges} freq(ij)/subranges), 1\}$$

and *Round* is the closest integer. The frequency memory is maintained over all iterations and in the two phases of this algorithm. Also, this frequency memory is used to control the probability of selecting the sub-ranges, this probability is inversely proportional to its frequency count.

Similar frequency memory is used to control the probability of selection of the move-variables.

3.1.5   Pseudo–code of our TS approach

Our procedure may be described as follows: 1) select an initial reference point, 2) apply a diversification and intensification strategies inside a "for loop" that controls the maximum number of iterations to generate new solutions and to chose the best of them.

What means the best solution in our approach?. We denote E a set of efficient moves, where a efficient move is a move that satisfies the aspiration level, otherwise the move is deficient. Then, we would define the best move as $[m \in E(x) : afv(x^*) = \max\{afv(x'), x' = m(x)\}]$ if $E(x) \neq \emptyset$, in the case where $E(x) = \emptyset$ we take randomly one solution of S. The best solution consists of the solution obtained by application of the best move to x.

Skeleton of our Tabu

Procedure Taboo
{   *(Initialization step)*
    Set a and b equal to the lower and upper bounds
    respectively of each variables
    Generate a feasible solution x *(the midpoint of each interval)*
    $z^* = F(x)$ *(Setting the reference point)*
    $S = x$
    newelement=True
    (newelement indicates the introduction of a new solution in S)
    for 1 to numiter *(the maximum number of iterations)*
        if newelement then intensification
        else diversification
        for 1 to fan*(the number of candidate solutions)*
            x'=Candidate(x)*(the fan strategy)*
            Make_tabudist(x, x')
            $x = x'$
            Update the reference point z^*
        endfor
    endfor }

4   Scatter search for non-linear problems

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.

4.1   Diversification generation method

An initial set of solutions is generated by means of a diversification generator. The generator that we implemented is based on tabu search approach explain above. The goal of the approach is obtain an initial good-diverse Pareto-know set of solutions.

4.2   Choosing subset of reference points

Our approach is organized to generate three different collections of diverse subsets of R, which we refer to as subset1 and subset2, subset3 (see [9]). The type of subsets we consider are as follows:

subset1: 3-elements subsets D_1, where the first element pertains to $R_1 - T$, the second element pertains to $R_1 - T_1$ the most dissimile to the first, and third element belongs to $R_1 - T_1$ selected to be the most dissimile to the selected elements

subset2: 3-elements subsets D_2, where the first element pertains to $R_2 - T$, the second element belongs to $R_2 - T_1$ the most dissimile to the first, and the third element pertains to $R_2 - T_1$ selected to be the most dissimile to the selected elements .

subset3: 3-elements subsets D_3, where the first element belongs to $R_2 - T$, the second element pertains to $R_1 - T_1$ the most dissimile to the first , and a third element selected to be the most dissimile to the selected elements belongs to $R_1 - T_1$.

If $R_1 - Z$ or $R_2 - Z$, for $Z = T$ and $Z = T_1$, are empty then, we take a random solution of R_1 or R_2 respectively.

The most dissimile solution 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.

In order to avoid duplicate the diverse subsets to be combined and to obtain more diversity in the search we use two different tabu sets. Once selected the first solution to be consider in the diverse subset, this solution is included in the tabu set T. The second and the third elements to be consider in the diverse subset are included in the tabu set T_1. A simple dynamic rules to create a tabu tenure t as tenure of the set T is used, chosen t to vary randomly between $t_{\min} = 1$ and $t_{\max} = 7$, and a static rules for t_1 as tenure of the set T_1 was selected, where $t_1 = 10$.

4.3   Combining strategies

As a basis for creating combined solutions we generate subsets $Y \subseteq R$ and for each subset use a solution combination method, generating solutions on line and using weights to sample points from the line.

Let $\Omega(Y) = x + \varpi(y - x)$, for $\varpi = 1/2, 1/3, 2/3, -1/3, -2/3, 4/3, 5/3$, and $x, y \in Y, \forall Y \in \{D_1, D_2, D_3\}$.

The strategy to create the $\Omega(Y)$ set is as follows: for each diverse subsets,

1. Generate new trial points on lines between x and y.

2. Generate new trial points on lines between x and z.

3. Generate new trial points on lines between y and z.

4. Generate one solution by applying $y + \frac{x-y}{2}$.

5. Generate one solution by applying $z + \frac{x-z}{2}$.

6. Generate one solution by applying $z + \frac{y-z}{2}$.

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 releases searches star from middle trial points.

In the special case where $C^k(P) = 1$ the approach begins by selecting the first element pertaining to R_1, denoting it by x, the second element is selected from $R_2 - T_2$, denoting it by y, if the cardinal of R_2 is greater than 1 then, the third element is selected from $R_2 - T_2$ to maximize its distance from the first and second elements, denoting it by z, in other case the third element is created by a linear combination of the two elements selected above. Then, apply the above scheme of combination explained.

If all subsets have been examined then, the algorithm subtract the Pareto set $\Omega^P \subseteq S\backslash C$, where $S = \cup\Omega(Y)$ and C is the critical set. New elements are incorporated into R if $|P \cap R| < |P|$ where, $P \subseteq \Omega^P \cup R$.

If $|P| > b_1$, then use the maxmin criteria to obtain a diversified collection of solutions R, that is, $x \in P$ maximizes the minimum distance $d(F(x), F(y(i)))$ for $i \leq |R|$, $y(i)$ pertaining to non-empty set R.

4.4   Critical event design

Avoiding the duplicated points already generated can be a significant factor in producing an effective overall procedure. The control is limited to these solutions that satisfy the condition of being Pareto. Our algorithm is based on a "critical event design" that monitors the current solutions in R and in the combined set $\Omega(Y)$. The elements considered in the critical events design are the values of the objectives, and the decision variables of the current Pareto solutions generated. We consider that a critical event has taken place if one trial solution is too close to another solution pertaining to the generated solution set or to the reference set taking account the distance between these solutions in the objective space and the decision space. Then, a critical event has occurred if a trial solution satisfies a "full critical condition". A "full critical condition" is satisfied if the "critical condition" is satisfied and the trial solution is inside the ball of radius δ defined on $\Omega(Y) \cup R$. The "critical condition" is satisfied if the image of the trial solution falls within the ball of radius ρ defined on $F[\Omega(Y) \cup R]$. The balls are defined for each solution generated so far. The distances that define each ball are adaptive parameters and their values depend of the number of solutions that historically have fallen within the balls.

The parameters ρ and δ range in the interval $[\min_r, \max_r]$ initially $\rho = \max_r$ and $\delta = \min_r$, then if the current number of points that have fallen inside the ball is greater than the average of the solutions that historically have fallen within the ball, then the current radius is halved, otherwise the value of the radius is twice. In our current implementation $\max_r = 1$ and $\min_r = 0.0625$.

The new solutions are put in the set $\Omega(Y)$ if they do not belong to the set of critical events.

4.5   Rebuild the reference set

In order to rebuild the reference set, we keep the solutions obtained so far and use tabu search method to obtain new trials solutions taking as starting solution a randomly solution of R. The trial solutions are introduced in R using the Pareto relation.

4.6   Pseudo–code scatter search

Procedure Scatter
```
{   Initialization
    For 1 to maxiter (the maximum number of scatter iteration)
    Taboo (Generate an initial reference set of trial points)
    if | P |> 1 then
        R₁ = Cᴷ(P)
        R₂ = P\Cᴷ(P)
        R = R₁ ∪ R₂
        while(0 < maxgeneration) or (ratio ≥ β)
            For all new subset Y from R
                Apply line search approach to obtain a set Ω(Y)
                of new solutions
            EndFor
            Obtain P ⊆ Ωᴾ ∪ R
            If |P| = 1 then Break
            ratio =| P ∩ R | / | P |
            R₁ = Cᴷ(P)
            R₂ = P \ Cᴷ(P)
            R = R₁ ∪ R₂
            maxgeneration = maxgeneration − 1
        endwhile
    endif
    endmaxiter }
```

5   Scatter search for a multiobjective combinatorial problem

In this part we examine an extension of our work ([2],[3]) applied to a job shop scheduling problem with multiple objectives using a framework scatter search that uses as a diversification generator a tabu search approach. Several cases are studied to demonstrate the ability of our algorithm to find good and diverse approximation solutions to Pareto-optimal front.

5.1   Job shop scheduling problem

Scheduling has been defined as the "allocation of resource over time to perform a collection of activities". Scheduling is important in the design and management of a variety of

systems, including production systems, construction projects, education, health delivery systems, transportation systems, and so on.

A job shop has a finite set of machines, each of which can do several types of operations but can process only one activity at a time. In a finite set of jobs, each job is split into a number of operations to be processed by a fixed order and on a specific machine for a specified duration.

In the standard model of job shop scheduling several assumptions concerning the way of processing the activities by machines are made (e.g., once a job initiates processing on a given machine it must complete processing on that machine uninterrupted), a schedule has to be found that optimizes several measures of performance. If release times of jobs are assumed, the problem is called a dynamic job shop, otherwise all release times are set to zero and it is called static.

Formally, the job shop scheduling problem can be stated as follows. A set M of $m(k = 1, \ldots, m)$ machines and a set J of $n(j = 1, \ldots, n)$ jobs are released at predetermined points in time r_j. Job j consists of $q_j(i = 1, \ldots, q_j)$ operations in series. Each operation i of a given job j is processed on a machine k_{j_i} for a given uninterrupted processing time p_{j_i} where $k_{j_i} \neq k_{j_{i+1}}$ for $i = 1, \ldots, q_j, j = 1, \ldots, n$. The j_i represents operation i of job j, and thus represents a particular activity $h \in \{1, 2, \ldots, L\}$, where $L = \sum_{j=1}^{n} q_j$ be the number of the different activities. Each machine M_k, can process at most one activity at a time. The problem is to find a table of starting times st_h of all activities with respect to setup times of machines, release time of jobs and technological constraints.

From the viewpoints of combinatorics, the question of how to sequence and schedule activities in such a system looks rather complex and is know to be NP-hard to almost state.

5.2   Job shop scheduling problem as a disjunctive graphs

A more useful representation of the job shop scheduling is provided by the disjunctive graph model that can be represented as follows: $G = (N; C, D)$, where N is the set of nodes, C the set of conjunctive arcs, and D the set of disjunctive arcs [1].

For each job shop scheduling problem (JSP) one can define a (directed) disjunctive graph $G = (N; C, D)$ by associating (α) a node $h \in N$ with each activity, including two dummies node s (start) to be the source of G, and node e (end) to be the sink of G; (β) a conjunctive arc $(h, h+1) \in C$ with each pair of operations pertaining to the same job and adjacent in the technological sequence; also, an arc $(s, h) \in C$ for each h, that is the first activity (operation to be performed on some job), and an arc $(h', e) \in C$ for each h', last activity (operation pertaining to a job); (γ): a disjunctive pair of arcs $(h, h') \in D$, with each pair of activities to be performed on different jobs but on the same machine; (δ): nodes are weighted. Nodes s and e have weight zero, while the weight of nodes $h \in N$, is the processing time p_h. In a disjunctive graph C corresponds to the job precedence constraints and D corresponds to the machine capacity constraints.

In the disjunctive graph, the basic scheduling decision corresponds to orienting each disjunctive arc in one direction or the other. A schedule is obtained by orienting all disjunctive arcs resulting in a conjunctive graphs. The orientation is feasible if the resulting

directed graph is circuit-free, and one can compute the starting time of each activity by computing the longest (weighted) path, from s to e.

A subset of D containing at most one arc of each disjunctive pair is called a selection. A selection containing exactly one arc of each disjunctive pair is called "complete". Let, $S = \{S_1, \ldots, S_r\}$ be the set of all complete selections. Each selection $S_l \in S$ generates a conjunctive graph of the form $G_l = (N; C \cup S_l) = (N, C_l)$.

5.3 Tabu search for a job shop

5.3.1 Neighborhood structures

The basic neighborhood and some properties have been defined by Van Laarhoven [8]. Let X be the set of all feasible schedules to a problem instance, then (1) if $x \in X$ is a feasible schedule to a problem instance, then using the proposition 1 and 2 explained above will obtain a feasible solution. We define the neighborhood as follows: N_1 produces a new feasible solution by complementing one arc $(h, h') \in S_l \cap CP_l(x)$.

5.3.2 A starting solution

As in [1] we define a direction for each disjunctive pair of arcs in D by calling the arc (h, h') normal if $h > h'$. D^+ and D^- will denote the set of all normal arcs and the set of all reverse arcs respectively. Then the graphs $(N, C \cup D^+)$ and $(N, C \cup D^-)$ have no circuits.

Starting with the graph $(N, C \cup D^+)$ in which all disjunctive arcs are normal, we generate a sequence of graphs $G_l = (N, C \cup S_l) \in Q'$. Each G'_l is obtained from the current G_l by applying N_1.

5.3.3 Tabu list

The fundamental memory structure in tabu search is the so-called tabu list. We associate with each leading from the current solution x to a solution $x' \in N(x)$ a set of attributes. We memorize in the tabu list the attributes of the applied moves and, at each iteration we select the best move among the set of candidates whose attributes do not belong to the tabu list or satisfies the aspiration condition. With the neighborhood N_1 a move consists in swapping one arc of the critical path then, given a candidate arc (h, h') and the associated move, we memorize as forbidden the reversal of the arc (h', h).

A candidate move is considered forbidden, if the move pertains to the tabu list and not satisfies the aspiration level.

The data structure is conceptually a square matrix T with dimensions equal to the number of activities. The element $T(h, h')$ contains the count of the iteration. In our case we can represent (h, h') or (h', h) by the pair (h, h') for $h < h'$. Thus, is possible use the data structure T as recency and frequency memory.

In our implementation we use a simple dynamic rule to vary our tabu time (*tenure*) randomly between $[.5\sqrt{n}, 2\sqrt{n}]$.

5.3.4   Diversification strategy

Frequency-based memory provides information to diversify the search. We present one forms to diversify the search using frequency counts. The frequency is obtained counting the number of times that each disjunctive arc appears in the selection $S_l(x)$ corresponding to solution x pertains to GS.

For our purposes, we use a frequency measure consisting of ratios, whose numerator is a count explained above and the denominator is the maximum numerator value.

The diversification is obtained modifying the value afv as follows:

$$afv = afv * (1 - \frac{freq(h, h')}{\max\{freq(h, h') \mid (h, h') \in S_l(x)\}})$$

where $freq(h, h')$ is the number of times that the disjunctive arc (h, h') has been used as move attribute in the considered set of solutions.

5.3.5   TS algorithm

Following the approach presented in [1] we start with the graph $(N, C \cup D^+)$ in which all disjunctive arcs are normal. The basis of our algorithm is to select at each step a non scheduled activity and to schedule this activity as early as possible on its machine. We generate a trial solution, by applying the move N_1. If the solution satisfies the aspiration level then, introduce it in the set GS. Choice the best solution that satisfies the aspiration level or is not tabu. If during a number of iteration equal to $MaxNonEfficientMove$ new elements are not introduced in GS then, we use a restart approach chosen one randomly solution of GS, resetting the tabu list and the aspiration level. The process is repeated during a fixed number of iteration and while the set GS is not full.

$MaxNonEfficientMove$ is an adaptive parameter defined as follows:

$$MaxNonEfficientMove = \lfloor \frac{10 \times num\_efficient\_move}{last\_efficient\_move} \rfloor$$

where $num\_efficient\_move$ is the number of efficient moves and $last\_efficient\_move$ is the number of the iteration of the last efficient move.

What means an efficient move in our tabu approach? An efficient move is a move that satisfies the aspiration level.

5.4   Scatter search for a job shop

5.4.1   Overall view of our procedure

An overall view of our procedure is given in this subsection.

The procedure starts with the generation of $1 <\mid GS \mid\leq b$ disperse solutions. These solutions are generated by tabu search explained above (I phase). The reference set R is the union $R_1 \cup R_2$, $R_1 = C^K(P)$ and $R_2 = P \backslash C^K(P)$ where C^K is the Kramer choice function and P is the current Pareto set extracted from GS. If $|P| > b_1$, then use the max-min criterion to obtain a diversified collection of solutions P', that is, set one first element

of P into P'then, let $x \in P$ maximizes the minimum distance $d(F(x), F(y(i)))$ for $i \leq |P'|$, $y(i)$ pertaining to non-empty set P' when $P' = b_1$ then, set $P = P'$. Diverse subsets DS are generated from the reference set as basis for creating combined solutions, then for each subset DS, to use a Solution Combined Method. We propose in our algorithm an structured weighted combination, to produce a set $\Omega(DS)$ of combined solutions. The new solutions are put in the set Ω^P if they do not belong to the set of critical events C. If all subsets have been examined then, the algorithm subtract the Pareto set $\Omega^P \subseteq GS \backslash C$, where $GS = \cup \Omega(DS)$. Let $P \subseteq \Omega^P \cup R$.

5.4.2   Dissimilarity measure H

As dissimilarity measure, we use the Hamming distance between two complete selection associated to the conjunctive graphics representing theirs corresponding job shop schedule solutions. Let $S_l =$ the complete selection associated to the graphic G_l then, $S_l(pq) = 1/0$ if the disjunctive arc (p, q) is present or absent in a graph G_l. For obtaining the distance between schedules x_1 and x_2 the logical xor() operator is applied to a complete selection associated to the graphics G_1, G_2 corresponding to the solutions x_1 and x_2.

$$H(x_1, x_2) = \sum_{(p,q) \in S_1, S_2} \mathrm{xor}(S_1, S_2)$$

where S_1 and S_2 are the complete selections to a corresponding solutions.

5.4.3   Choosing subsets of reference points

Our approach is organized to generate different collections of divers subsets of R.

Suppose $R_1 \neq \emptyset$ and $R_2 \neq \emptyset$ then, the type of subsets we consider are as follows:

- 3-elements subset $D1$, where the first element x pertains to R_1, the second element y, belongs to $R_1 - T$ the most diverse relative to x, and a third element selected to be the most dissimile to the selected elements belongs to $R_1 - T$.

- 3-element subsets $D2$, where the first element x pertains to R_1, the second element y belongs to $R_2 - T$ the most diverse relative to x, and the third element pertains to $R_2 - T$ selected to be the most dissimile to the former two.

- 3-element subsets $D3$, where the first element x pertains to R_2, the second element y belongs to $R_2 - T$ the most diverse relative to x, and a third element chosen from $R_2 - T$ to be the most dissimile to the selected elements.

The most dissimile solution from any given solution is chosen using the following function $\max\{d_{\min}(x, y)\}$, where $d_{\min}(x, y)$ is the Hamming distance.
If $R_1 - T$ or $R_2 - T$, is empty then, we take a random solution of R_1 or R_2 respectively.

5.4.4   Combination method

Let three-element subsets $DS \in \{D1, D2, D3\}$. Create the pairs $(x, y) \in A = \{(\binom{|DS|}{2}) \mid x, y \in DS\}$, and for all (x, y) apply the structured weighted combination method using different weights associated to the combined solutions.

To apply the structured combination to the solutions, we need obtain a sequence of the activities for each machine then, a voting mechanism for precedence relationships where a vector assigns vote of 1 for h to precede h' if this precedence order occurs in the sequence and 0 if not occurs (for more details of this method see [10]). If the solution generated is infeasible, this solution is ignored working only with feasible solutions.

Our system of weights gives more preference to solutions pertain to R_1.

5.5   Improving method

As improving method we use a simple local search, using the neighborhood N_1 defined bellow. The trial solution is incorporated to GS if it is non-dominated by the its "donors".

5.6   Illustrative examples

Example 1

The processing times (*proctime*) are fixed in the following table and the machine routing (*machrout*) in the table bellow

Job	proctime			machrout		
1	4	3	2	1	2	3
2	1	4	4	2	1	3
3	3	2	3	3	2	1
4	3	3	1	2	3	1

For this problem the optimal solution for makespan objective is 13 and the optimal solution for earliness objective is 0. The 1 shows the approximation to Pareto front achieved by our algorithm, observe that the algorithm achieved the optimum.
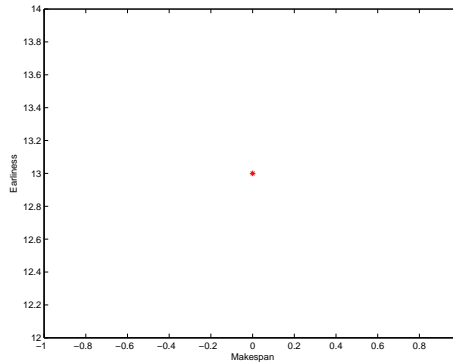

Figure 1: Pareto frontier.

Example 2

We illustrate the case where more than one operation associated to one job use the same machine. We have 3 jobs, 4 machines and not all jobs have the same number of operations. Processing time table is represented below and the machine routing table as follows:

Job	proctime					machrout				
1	4	3	2	4	6	1	2	3	4	-
2	1	4	4	5	8	1	2	4	4	-
3	3	2	3	4	7	1	2	3	2	4
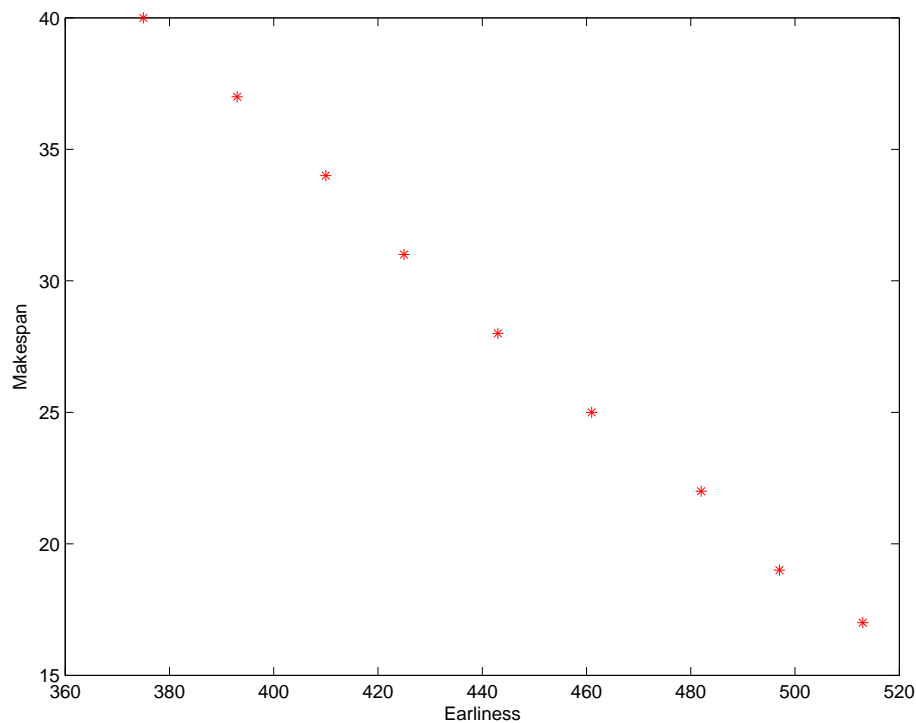
Figure 2: Pareto front achieved.

6 Computational experiments

6.1 Non-linear problems

Several test problems of were used to assess our approach [7]. Now we show the performance of our improved algorithm with new test functions.

6.2   3-objectives test problems

In this section we will examine the performance of our algorithm on several test problems taken from the literature [Deb and others, [7]].

Our experiments were run with the following parameters: $maxiter = 30$, $maxgeneration = 25$ for iteration, the numbers of iteration in tabu was 500, $fan = 25$, $\beta = 0.5$, $b = 500$, $b_1 = 100$ with $b_2 = b + b_1$. In this case our approach uses at most 750 000 objective evaluations.

6.2.1   Problem DTLZ1

The following problem is a 3-objective problem with a linear Pareto-true front. The Pareto front corresponds to $X_3 = 0$ and the objective function values lie on the linear hyper-plane:$\sum_{m=1}^{3} f_m = 0.5$.

Minimize the following functions:

$$f_1(x) = \frac{1}{2}x_1 x_2 (1 + g(X_3))$$

$$f_2(x) = \frac{1}{2}x_1 (1 - x_2)(1 + g(X_3))$$

$$f_3(x) = \frac{1}{2}(1 - x_1)(1 + g(X_3))$$

$$0 \le x_i \le 1, i = 1, 2, \ldots, 7$$

$$g(X_3) = 100[|X_3| + \sum_{x_i \in X_3} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))]$$

6.2.2   Problem DTLZ2

The following test problem has an objective search space where the Pareto front must lie inside of the unit sphere.

Minimize the following functions:

$$f_1(x) = (1 + g(X_3))cos(x_1\pi/2)cos(x_2\pi/2)$$

$$f_2(x) = (1 + g(X_3))cos(x_1\pi/2)sin(x_2\pi/2)$$

$$f_3(x) = (1 + g(X_3))sin(x_1\pi/2)$$

$$0 \le x_i \le 1, i = 1, 2, \ldots, 12$$

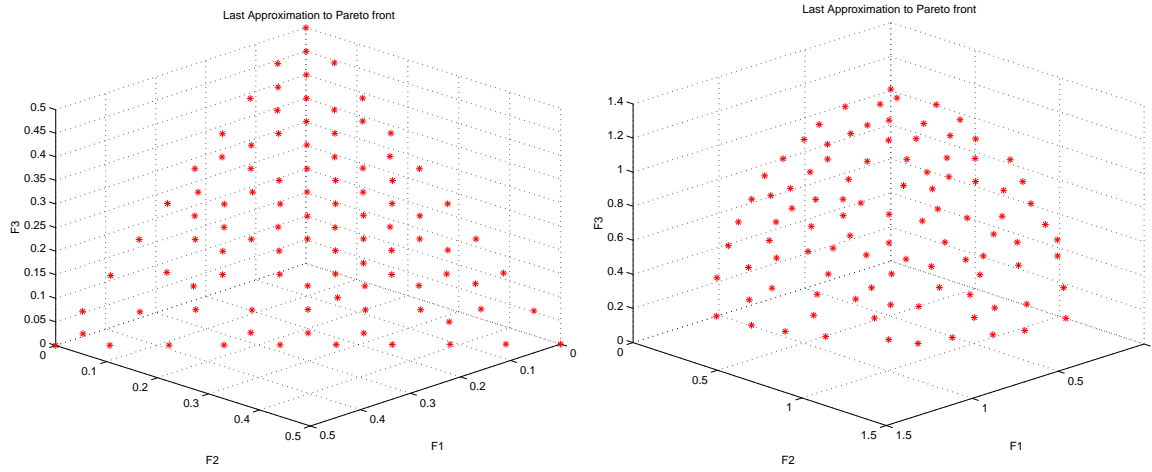$$g(X_3) = 100[|X_3| + \sum_{x_i \in X_3} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))]$$

Figure 3: Pareto-true front vs Pareto front achieved by MOSS on DTLZ1 and DTLZ2 problems.

6.2.3   Problem DTLZ3

The following test problem prove the ability to converge to the global Pareto front, $(3^{10}-1)$ local Pareto front, and one global Pareto front.

Minimize the following functions:

$$f_1(x) = (1 + g(X_3))cos(x_1\pi/2)cos(x_2\pi/2)$$

$$f_2(x) = (1 + g(X_3))cos(x_1\pi/2)sin(x_2\pi/2)$$

$$f_3(x) = (1 + g(X_3))sin(x_1\pi/2)$$

$$0 \le x_i \le 1, i = 1, 2, \dots, 12$$

$$g(X_3) = \sum_{x_i \in X_3} (x_i - 0.5)^2$$

6.2.4   Problem DTLZ7

This problem has a disconnected set within the Pareto-true regions.

Minimize the following functions:

$$f_1(x_1) = x_1,$$

$$f_2(x_2) = x_2,$$

$$f_3(X_3) = (1 + g(X_3))h(f_1, f_2, g),$$

$$g_(X_3) = 1 + \frac{9}{X_3} \sum_{x_i \in X_3} x_i,$$

$$h(f_1, f_2, g) = 3 - \sum_{i=1}^{2} [\frac{f_i}{1+g}(1 + sin(3\pi f_i))],$$

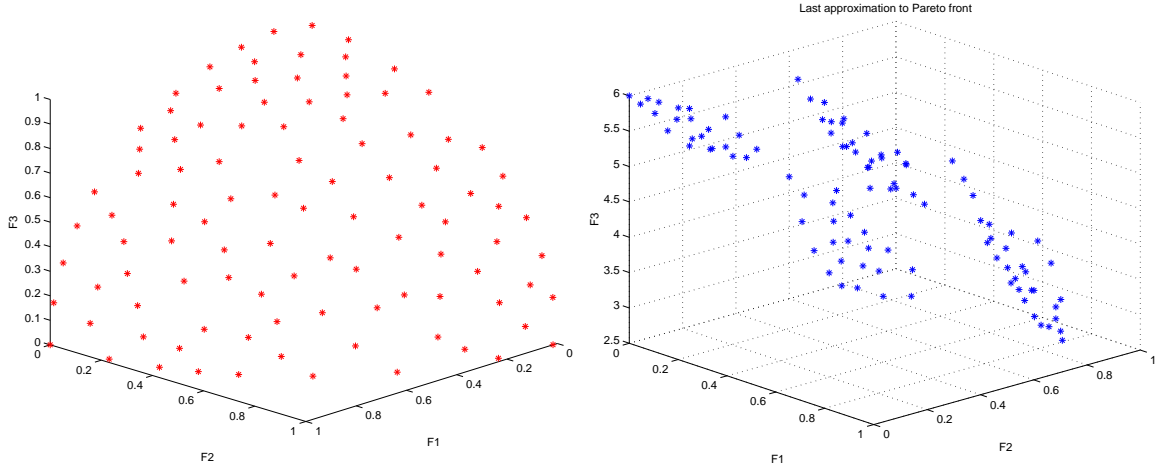$$0 \le x_i \le 1, i = 1, 2, \dots, 22$$

Figure 4: Pareto-true front vs Pareto front achieved by MOSS on DTLZ3 and DTLZ4 problems.

This test problem has 4 disconnected Pareto-true regions in the objective space. The functional g has 20 decision variables and the total number of variables was 22. The Pareto-true solutions correspond to $X_3 = 0$. This tests an algorithm's ability to maintain subsets in different Pareto-true regions.

The results are shown in the above figures, and discloses that the algorithm was able to find stable and distributed subsets in all Pareto-true regions.

In addition to the graphical presentation, the algorithm was assessed in pairs using the Coverage Set metric (CS), Generational Distance metric (GD) [20], the Spacing metric (S) [18], the Maximum Extension metric (MEF) [21], and the Time (T) to evaluate the distribution of the solutions of each Pareto-achieved front. For an ordered pair ($PF_{optimal}, PF_{achieved}$), the shortcut PF stands for Pareto front.

In this case were generated 100 points of the true Pareto front for each test problems, the 5 and 6, show the distribution of coverage set and spacing of 21 runs for each test problems. The results indicate, that the convergence to the true Pareto front in all cases was good and in general a well distribution of points was obtained. The box plot representing the coverage set indicates that the approximation to the Pareto-optimal front is good (close to cero), and also the distribution of the solution is acceptable. The S and MEF metrics show a right performance of our performance.
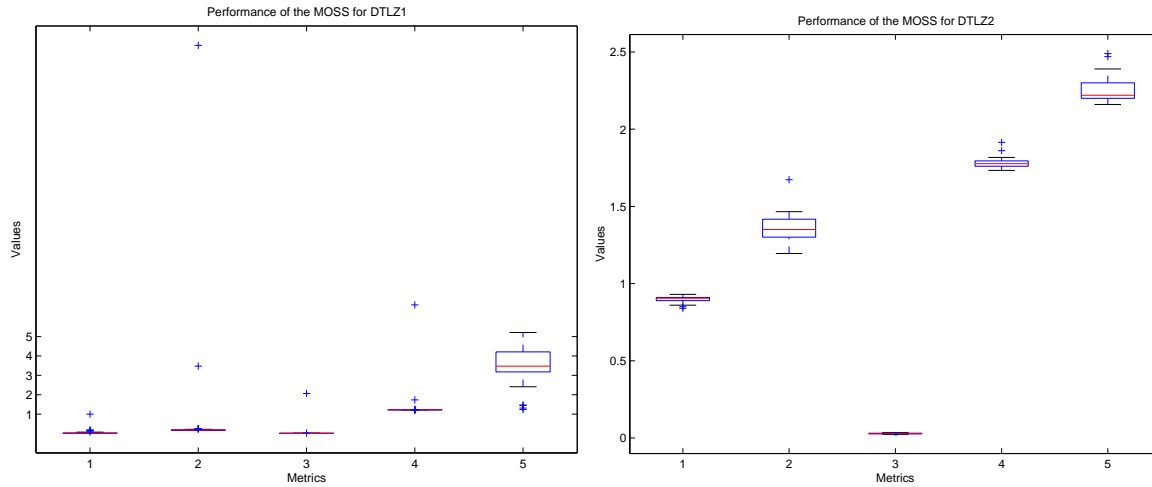
Figure 5: Boxplot representing the distribution of the metrics on DTLZ1 and DTLZ2 test problems ($CS = 1, GD = 2, S = 3, MEF = 4$ and $T = 5$.)

6.2.5 Combinatorial problem

We conducted preliminary experiments on different generated problems, coded in Delphi on a Pentium II with a clock of 350 MHz. We solved two 2-objective and one 3-objective instances with ($n = 5, m = 30$), with objectives functions (earliness-makespan), (makespan-averagetime), and (makespan-averagetime-earliness), taking as a benchmark ABZ test problems with due date incorporated. The problems were generated as follows:

a) Choose a number of machines m and jobs n from the following cases: $m = 30; n = 5$.

b) All jobs must be processed on all machines.

c) The next machines is chosen uniformly from those remaining.

d) Operation processing times are all drawn uniformly from the interval $[5, 99]$. There are static arrivals and static machine availability.

Following Rubin and Ragatz[14], the due date was generated as follows: the mean of the due dates was set equal to $(1 - TF) * n * ($mean of processing time$)$, and its range was set equal $(RD) * n * ($mean of processing time$)$, where RD is the relative range of due dates and n is the number of jobs. The objective functions were (earliness-makespan), and (earliness-average time). This instance was ran with the following parameters $\rho = 0$ and $\xi(i) = 1$ for $i \in \{1, 2, 3\}$, $CutoffLimit = 5$, $MaxIter = 1$, $b = 500$, $b_1 = b$, and $b_2 = 100$.

7 Conclusions

In this paper we have presented an adaptation of the tabu/scatter Search methods to multi-objective optimization. Several test problems to demonstrate the ability of our approach
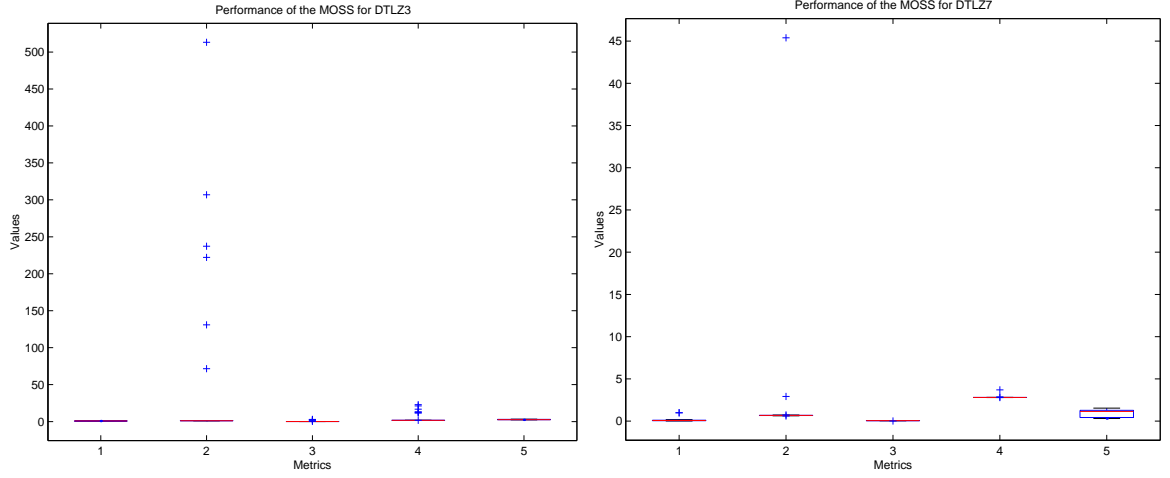
Figure 6: Boxplot representing the distribution of the metrics on DTLZ3 and DTLZ7 problems ($CS = 1, GD = 2, S = 3, MEF = 4$ and $T = 5$.)

were run. Convergence to the Pareto frontier was good, and a widely Pareto frontier was obtained. We can conclude that the application of a based-memory diversification strategy, the use of a choice function to separate the reference set points and convex combination seems to be a good approach in finding a good approximation and widely distribution of the Pareto frontier. A new approach for job shop scheduling problem is proposed. Different cases (static, dynamic, dependent-setup times, operations pertaining to the same job partially ordered) can be solved with our algorithm in a multiobjective environment.

The use of tabu search as generation method of diverse initial solutions, different strategies of memories to diversify the search, the use of nondominated solutions as reference set permit a better control to explore the solution space to achieve new potentially Pareto solutions, the partition of this set using choice functions as the Kramer choice function and structured combination produce very quickly new potentially Pareto solutions.

References

[1] Balas, E. "Machine sequencing via disjunctive graphs: an implicit enumeration algorithm", *Ops.Res* **17** (6): 927–1136.

[2] Beausoleil, R.P. (2001) "Multiple criteria scatter search", *MIC'2001-4th Metaheuristics International Conference*. Porto, Portugal.

[3] Beausoleil, R.P. (2002) "A tabu search approach for weighted tardiness with sequence-dependent setups in one-machine problem", *Revista de Matemática: Teoría y Aplicaciones* **9**(1): 35–46.
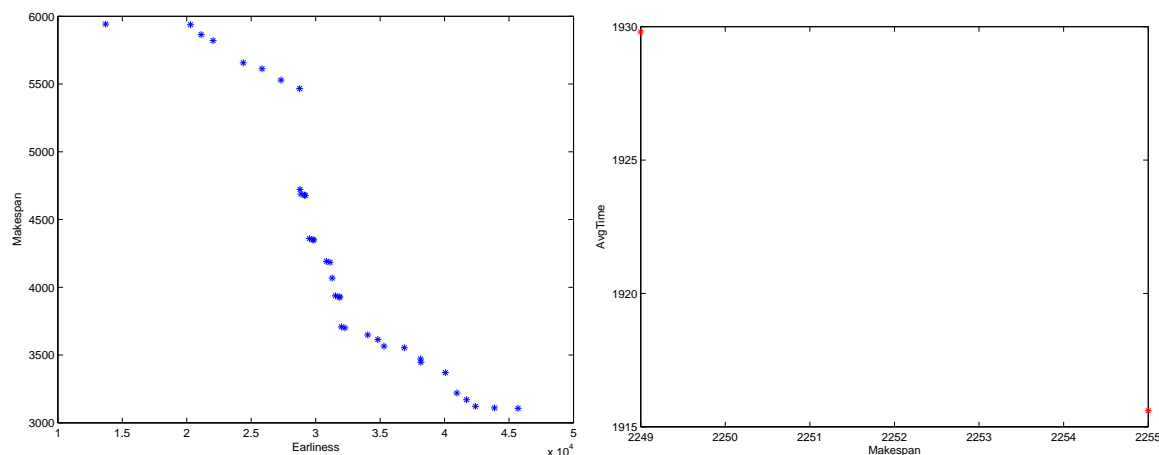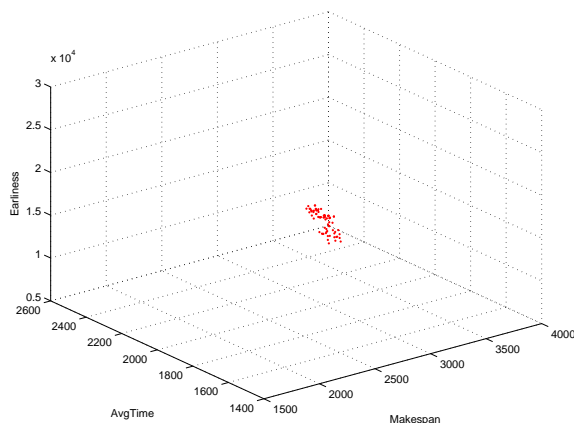
Figure 7: Evolution to Pareto frontier (*)

Figure 8: Evolution to Pareto frontier (*)

[4] Beausoleil, R. (2004) "Bounded Variables Nonlinear Multiple Criteria Optimization using Scatter Search", *Revista de Matemática: Teoría y Aplicaciones* **11** (1): 17–40.

[5] Coello, C.A.; Toscano P.G. (2001) "A micro-genetic algorithm for multiobjective optimization", In: E. Zitzler, K. Deb, L. Thiele, C.A. Coello & D. Corne (Eds.), *First International Conference on Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Sciences No.1993, Springer-Verlag: 216–140

[6] Deb, K.; Agrawal, S.; Pratab, A.; Mayariban, T. (2000) "A fast elitist nonsorting genetic algorithms for multi-objective optimization", NSGA-II, KanGal, report 200001, Indian Institute of Technology, Kanpur, India.

[7] Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. (2001) "Scalable test problems for evolutionary multi-objective optimization", *TIK-Tecnical Report* (112).

[8] Laarhoven P.J.; Aarts E.H.; Lenstra J.K. (1998) "Job shop scheduling by simulated anneling", Report OS, Centre for Mathématiques, Ecole Polytechnique Fédérale de Lausanne.

[9] Glover, F. (1998) "A template for scatter search and path relinking in artificial evolution", *Lecture Notes in Computer Science* (1363): 13–54. J.-K,Hao, E.Lutton, E.Ronald, M.Schoenauer and D.Snyers (Eds), Springer.

[10] Glover, F. (1994) "Tabu search for nonlinear and parametric optimization (with links to genetic algorithms)", *Discrete Applied Mathematics* (40): 231–255.

[11] Glover, F.; Laguna M. (1993) "Modern Heuristic Techniques for Combinatorial Problems", Halsted Press, John Wiley & Sons, Inc.: 70–147.

[12] Glover, F.; Laguna, M.; Martí, R. "Scatter Search Tutorial for a class of non-linear optimization problem on bounded variables".

[13] Makarov, I.M.; Vinogradskaia, T.M.; Rubinski, A.A; Sokolov, V.B. (1982) *Choice Theory and Decision Making.* Nauka, Moscu.

[14] Rubin, P.A.; Ragatz, G.L. (1995) "Scheduling in a sequence dependent setup environment with genetic search", *Computers and Operation Research* (22): 85–99.

[15] Pratyush, S.; Jian-Bo, Y. (1998) *Multiple criteria decision support in engineering design*, Springer-Verlag.

[16] Schaffer, J.D. (1985) "Multiple objective optimization with vector evaluated genetic algorithms", Unpublished doctoral dissertation, Vanderbilt University.

[17] Knowles, J.; Corne, D. (1999) *The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimization.* University of Reading, UK.

[18] Schott, J.R. (1995) *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization.* Master's thesis, Departament of Aeronautics, Massachusetts Institute of Technology, Cambrige, Massachusetts.

[19] Schaffer, J.D. (1985) "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", In: *Genetic Algorithms and their Applications*, Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum, Hillsdale, New Jersey: 93–100

[20] Van, V.; Lamont, G.B. (1998) "Evolutionary computation and convergence to a Pareto front", In: J.R. Koza (Ed.), Late Breaking Papers at the *Genetic Programming 1998 Conference*, Stanford University, CA: 221–228

[21] Zitzler, E.; Kalyanmoy, D.; Thiele, L. TIK-Report, No.70, December 22, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

[22] Zitzler, E.; Laumanns, M.; Thiele, L. (2001) "SPEA2: Improving the strength Pareto evolutionary algorithm", Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.