International Journal of Mathematical Archive-4(5), 2013, 16-25
Available online through www.ijma.info ISSN 2229 – 5046

SOLVING CONSTRAINED MULTIOBJECTIVE OPTIMIZATION USING RP-TR/PSO

Ahmed A. El-Sawy[a], Zeinab M. Hendawy[b] and M. A. El-Shorbagy[b*]

[a]Department of Mathematics, Faculty of Science, Qassim University, Saudi Arabia
[b]Department of Basic Engineering Science, Faculty of Engineering, Menoufiya University, Egypt

(Received on: 20-03-12; Revised & Accepted on: 16-04-13)

ABSTRACT

In this paper, a hybrid approach combining trust region (TR) algorithm and particle swarm optimization (PSO) is proposed for solving constrained multiobjective optimization problems (MOOPs). The proposed approach integrates the merits of both TR and PSO. the constrained MOOP is handled by reference point (RP) Interactive Approach and some of the points in the search space are generated. Secondly, For each point the TR algorithm is used to obtain a point on the Pareto frontier. Finally, All the points that have been obtained on the Pareto frontier are used as particles position for PSO; where homogeneous PSO is applied to get all the points on the Pareto frontier. To restrict velocity of the particles and control it, a dynamic constriction factor is presented. The algorithm is coded in MATLAB 7 using 3 GHz PC. Various kinds of multiobjective (MO) benchmark problems have been tested to illustrate the successful result in finding a Pareto-optimal set.

Keywords: *Multiobjective optimization; trust region algorithm; particle swarm optimization, Pareto-optimal solution; reference point method.*

INTRODUCTION

In the real world, there are many problems involving multiple objectives which should be optimized simultaneously. Thus multiobjective optimization problem (MOOPs) is a very important research topic for both scientists and engineers and there are still many open questions in this area [3].

For MOOPs, objective functions may be optimized separately from each other and the best solution can be found for each objective dimension. However, suitable solutions for all functions can seldom be found. This because in most cases the objective functions are in conflict with each other. It results in there being a group of alternative solutions (Pareto-optimal solutions) which must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others. i.e., there is no single optimal value as in single objective optimization.

Numerical optimization techniques (Traditional methods) such as the gradient-based methods are single objective optimization methods that optimize only one objective (not designed to deal with multiple optimal solutions). These methods usually starts with a single baseline and use local gradient information of the objective function with respect to changes in decision variable to calculate a search direction. When these methods are applied to a MOOP, the problem is transformed into a single objective optimization problem by combining multiple objectives into a single objective typically using a weighted sum method [14].

Therefore, one must run many optimizations by trial and error adjusting the weights to obtain Pareto-optimal solutions. This is considerably time consuming. What is more, there is no guarantee that uniform Pareto-optimal solutions can be obtained. For example, when this approach is applied to a MOOP that has a concave tradeoff surface, it converges to two extreme optimums without showing any tradeoff information between the objectives.

TR is a term used in numerical optimization to denote the subset of the region of the objective function to be optimized that is approximated using a model function (often a quadratic). If an adequate model of the objective function is found within the TR then the region is expanded; conversely, if the approximation is poor then the region is contracted. TR methods are also known as restricted step methods.

The fit is evaluated by comparing the ratio of expected improvement from the model approximation with the actual improvement observed in the objective function. Simple thresholding of the ratio is used as the criteria for expansion and contraction (a model function is "trusted" only in the region where it provides a reasonable approximation).

Corresponding author: *M. A. El-Shorbagy[b*]*
[b]Department of Basic Engineering Science, Faculty of Engineering, Menoufiya University, Egypt

TR methods are in some sense dual to line search methods: TR methods first choose a step size (the size of the TR) and then a step direction while line search methods first choose a step direction and then a step size [19].

Evolutionary Algorithms (EAs) [5], on the other hand, are particularly suited for MOOPs. By maintaining a population of design candidates and using a fitness assignment method based on the Pareto-optimal concept, they can uniformly sample various Pareto-optimal solutions in one optimization without converting a MOOP into a single objective problem. In addition, EAs have other advantages such as robustness, efficiency, as well as suitability for parallel computing. Due to these advantages, EAs are unique and attractive approach to real-world optimization problems.

PSO also is an evolutionary computational model which is based on swarm intelligence. PSO is developed by Kennedy and Elberhart [11] who have been inspired by the research of the artificial livings. Similar to EAs, PSO is also an optimizer based on population. The system is initialized firstly in a set of randomly generated potential solutions, and then performs the search for the optimum one iteratively. Whereas the PSO does not possess the crossover and mutation processes used in EAs, it finds the optimum solution by swarms following the best particle. Compared to EAs, the PSO has much more profound intelligent background and could be performed more easily. Based on its advantages, the PSO is not only suitable for science research, but also engineering applications, in the fields of evolutionary computing, optimization and many others.

This paper intends to present a hybrid approach (RP-TR/PSO) to solving constrained MOOPs. It combines the two optimization techniques TR and PSO. It is a new algorithm that performs random searching (PSO algorithm) and deterministic searching (TR algorithm) for solving constrained MOOPs. Various kinds of MO benchmark problems have been tested to illustrate the successful result in finding a Pareto-optimal set.

MULTIOBJECTIVE OPTIMIZATION (MOO)

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term “optimize” means finding such a solution which would give the values of all the objective functions acceptable to the designer [4,16]. The general minimization problem of q objectives can be mathematically stated as:

$$\left.\begin{array}{ll}\text{Minimize :} & f(x)=\left[f_j(x),\ j=1,2,\dots,q\right] \\ \text{Subject to the constraints :} & Ci(x)\le 0,\quad \mathrm{i}=1,2,\dots,p, \\ & Ce(x)=0,\quad e=1,2,\dots,m,\end{array}\right\} \tag{1}$$

where $f_j(x)$ is the *j-th* objective function ,$Ci(x)$ is the *i-th* inequality constraint, $C_e(x)$ is the e-th equality constraint and $x=[x_1,x_2,\dots,x_n]$ is the vector of optimization or decision variables; where n the dimension of the decision variable space. The MOO problem then reduces to finding an x such that $f_j(x)$ is optimized. Since the notion of an optimum solution in MOOP is different compared to the SOOP, the concept of Pareto dominance is used for the evaluation of the solutions. This concept formulated by Vilfredo Pareto is defined as [17]:

Definition 1: (Dominance Criteria [13]). For a problem having more than one objective function (say, f_j, $j=1,\dots,q$, $q>1$), any two solution x_a and x_b can have one of two possibilities, one dominates the other or none dominates the other. A solution x_a is said to dominate the other solution x_b, if both the following condition are true:

The solution x_a is no worse (say the operator $\prec$ denotes worse and $\succ$ denotes better) than x_b in all objectives, or $f_j(x_a)\not\prec f_j(x_b)$ for all $j=1,\dots,q$ objectives.

The solution x_a is strictly better than x_b in at least one objective, or $f_j(x_a)\succ f_j(x_b)$ for at least one $j\in\{1,\dots,q\}$.

If any of the above condition is violated, the solution x_a dose not dominates the solution x_b.

Definition 2: (Pareto-optimal solution). x^* is said to be a Pareto-optimal solution of MOOP if there exists no other feasible x such that, $f_j(x)\le f_j(x^*)$ for all $j=1,\dots,q$ and $f_j(x)<f_j(x^*)$ for at least one objective function f_j.

RP-TR/PSO APPROACH

In this section, the proposed algorithm is presented. The proposed algorithm contains three stages RP stage, TR stage (used to obtain a point on the Pareto frontier), and PSO stage (is applied to get all the points on the Pareto frontier). The mechanism of the proposed algorithm in the objective space is shown in Figure 1.

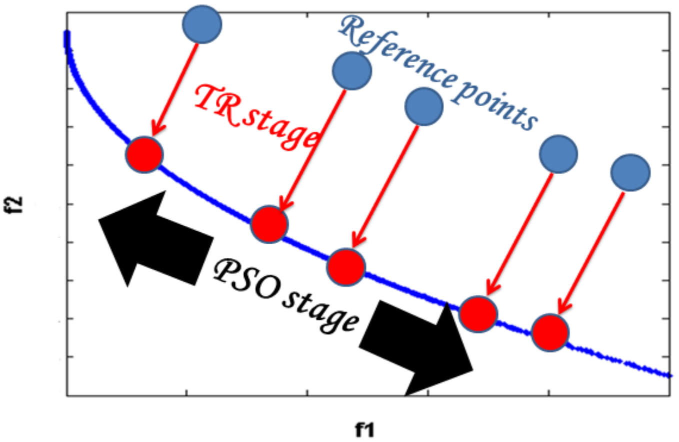


Fig. 1: The mechanism of RP-TR/PSO approach in the objective space.

RP stage

Initialization

Initialize N reference points ($\bar{z}$) in the search space.

RP method

The RP interactive approach of Wierzbickiis [13] is very simple and practical for MOOPs. Before the solution process starts, some information is given to the decision maker (DM) about the problem. The goal is to achieve Pareto-optimal solution closest to a supplied RP. Given a RP $\bar{z}$ for an q-objective optimization problem (1), the following SOOP is solved for this purpose [13]:

$$\left.\begin{aligned} &\text{minimize : } \; f(x)=\left(\sum_{j=1}^{q}\left|f_j(x)-\bar{z}_j\right|^p\right)^{1/p} \\ &\text{subject to: } \; Ci(x)\le 0, \qquad \mathrm{i}=1,2,\dots,l, \\ &\qquad\qquad\qquad Ce(x)=0, \quad e=1,2,\dots,m, \end{aligned}\right\} \tag{2}$$

where the parameter p can take any value between 1 and ∞. When $p=2$ is used, an Euclidean distance of any point in the objective space from the RP $\bar{z}$ is minimized. For a chosen RP, the closest Pareto-optimal solution is the target solution to the RP method.

The method proceeds as follows: The DM specifies a RP $\bar{z}$ and a solution with equal proportional achievements is generated. Then the DM specifies a new RP and the iteration continues until the DM satisfied with the solution produced.

TR stage

This section is devoted to presenting the detailed description of TR algorithm for solving problem (2). The TR algorithm combines ideas from Byrd [2], Omojokun [15], El Alem [8].

Following Dennis et al. [6], we define the indicator matrix $W(x)\in\mathbb{R}^{p\times p}$, whose diagonal entries are

$$w_i(x)=\begin{cases}1 & \text{if } \; Ci(x)\ge 0,\\ 0 & \text{if } \; Ci(x)<0.\end{cases} \tag{3}$$

Using this matrix, the Problem defined in Eq. (2) can be transformed to the following equality constrained optimization problem:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && 1/2Ci(x)^T W(x) Ci(x)=0, \\ & && Ce(x)=0. \end{aligned} \tag{4}$$

The above problem can be rewritten as:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0, \end{array} \tag{5}$$

where $h(x) = [Ce(x)\ \ 1/2Ci(x)^T W(x) Ci(x) = 0]$.

The Lagrangian function associated with problem defined in (5) is given by

$$L(x_k, \lambda_k) = f(x_k) + \lambda_k^T h(x_k) \tag{6}$$

where $\lambda_k \in \mathbb{R}$ is the Lagrange multiplier vector associated with equality constraint $h(x_k) \in \mathbb{R}$.

The augmented Lagrangian is the function

$$\Phi(x, \lambda; r) = L(x, \lambda) + r\|h(x_k)\|^2, \tag{7}$$

where r > 0 is a parameter usually called the penalty parameter.

The reduced Hessian approach is used to compute a trial step d_k. In this approach, the trial step d_k is decomposed into two orthogonal components; the normal component d_k^n and the tangential component d_k^t. The trial step d_k has the form $d_k = d_k^n + Z_k \bar{d}_k^t$, where Z_k is a matrix whose columns form an orthonormal basis for the null space of $\nabla h(x_k)^T$.

We obtain the normal component d_k^n by solving the following TR sub-problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|h(x_k) + \nabla h(x_k)^T d^n\right\|^2 \\ \text{subject to} & \|d^n\| \le \xi\Delta_k, \end{array} \tag{8}$$

for some $\xi \in (0,1)$ and trust region radius (Δ_k).

Given the normal component d_k^n, we compute the tangential component $d_k^t = Z_k \bar{d}_k^t$ by solving the following TR sub-problem:

$$\begin{array}{ll} \text{minimize} & \left[Z_k^T\left(\nabla_x L(x_k, \lambda_k) + H_k d_k^n\right)\right]^T \bar{d}^t + \frac{1}{2}\bar{d}^{t^T} Z_k^T H_k Z_k \bar{d}^t \\ \text{subject to} & \|Z_k \bar{d}^t\| \le \sqrt{\Delta_k^2 - \|d_k^n\|^2}, \end{array} \tag{9}$$

Once the trial step is computed, it needs to be tested to determine whether it will be accepted or not. To do that, a merit function is needed. We use the augmented Lagrangian function (7) as a merit function. To test the step, we compare the actual reduction in the merit function in moving from x_k to $x_k + d_k$ versus the predicted reduction.

The actual reduction in the merit function is defined as:

$$Ared_k = L(x_k, \lambda_k) - L(x_{k+1}, \lambda_{k+1}) + r_k\left[\|h(x_k)\|^2 - \|h(x_{k+1})\|^2\right],. \tag{10}$$

The predicted reduction in the merit function is defined as:

$$Pred_k = -\nabla_x L(x_k, \lambda_k)^T d_k - \frac{1}{2}d_k^T H_k d_k - \Delta\lambda_k^T\left(h(x_k) + \nabla h(x_k)^T d_k\right) + r_k\left[\|h(x_k)\|^2 - \left\|h(x_k) + \nabla h(x_k)^T d_k\right\|^2\right]; \tag{11}$$

where $\Delta\lambda_k = (\lambda_{k+1} - \lambda_k)$.

If $(Ared_k / Pred_k) < \tau_0$, where $\tau_0 \in (0,1)$ is a small fixed constant, then the step is rejected. In this case, the radius of the TR Δ_k is decreased by setting $\Delta_k = \tau_3\|d_k\|$, where $\tau_3 \in (0,1)$, and another trial step is computed using the new trust-region radius. If $(Ared_k / Pred_k) \ge \tau_2$, where $\tau_2 > 0$, then the step is accepted and set the TR as $\Delta_{k+1} = \min\{\Delta_{\max},\ \max\{\Delta_{\min}, \tau_1\Delta_k\}\}$. If $\tau_0 \le (Ared_k / Pred_k) < \tau_2$, then the step is accepted and set the TR as $\Delta_{k+1} = \max(\Delta_k, \Delta_{\min})$. Finally, the algorithm is terminated when either $\|d_k\| \le \varepsilon_1$ or $\|Z_k^T \nabla_x L_k\| + \|h_k\| \le \varepsilon_2$, for some $\varepsilon_1, \varepsilon_2 > 0$.

PSO stage
In this stage a homogeneous PSO for MOOP (see [10]) is proposed with a decreasing constriction factor to restrict velocity of the particles and control it [1]. In homogeneous PSO one global repository concept is proposed for choosing *pbest* and *gbest*, this means that each particle has lost its own identity and treated simply as a member of social group. The procedure of the PSO stage is as follows.

Step 1: Initialization
All non-dominated points (which obtained by applying TR stage) chosen as particles position x_i^t.

Store non-dominated particles in Pareto repository. If the specific constraint doesn't exist for a repository, the size of the repository is unlimited.

Step 2: Evaluation
Evaluate the MO fitness value of each particle and save it in a vector form.

Step 3: Floating
Two optimal solutions are chosen randomly for *pbest* (P_i) and *gbest* (P_g) from the repository.

Determine the new position of each particle in the following manner:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{12}$$

with the velocity v_i^{t+1} calculated as follows:

$$v_i^{t+1} = wv_i^t + c_1 r_1 \left(p_i - x_i^t\right) + c_2 r_2 \left(p_g - x_i^t\right). \tag{13}$$

Here, subscript *t* indicates an pseudo-time increment, *w* is the inertia weight, c_1 and c_2 are two positive constants, called the cognitive and social parameter respectively, and r_1 and r_2 are random numbers uniformly distributed with in the range [0,1].

Step 4: Repairing of particles:
Where the particle *i* start at the position x_i^t with velocity v_i^t in the feasible space, the new position x_i^{t+1} (see Figure 2) depends on velocity v_i^{t+1}.

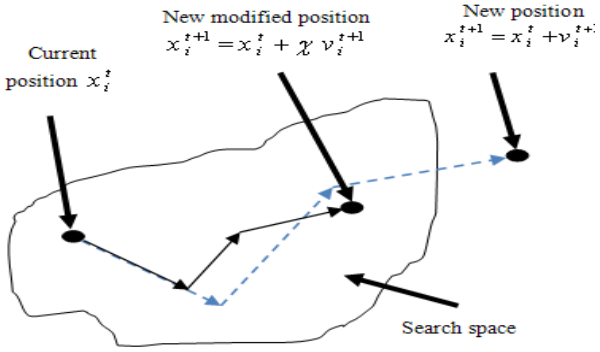


Fig. 2: The movement of **the** particle *i* through search space

To restrict (control) the particle's velocity v_i^t, a modified constriction factor (i.e., dynamic constriction factor) is presented to keep the feasibility of the particles. E.g., Figure 2 shows the movement of the particle *i* through the search space without any control factor (dashed line) also with a modified constriction factor (solid line). Where the particle *i* starts at position x_i^t with velocity v_i^t in the feasible space, the new position x_i^{t+1} depends on velocity v_i^{t+1} making the particle lose its feasibility, so we introduce a modified constriction factor

$$\chi = \frac{2}{\left|-2-\tau-\sqrt{\tau^2+\tau}\right|}; \tag{14}$$

where, τ is the age of the infeasible particle (i.e., how long it is still infeasible) and it is increased with the number of failed trials to keep the feasibility of the particle.

The new modified positions of the particles are computed as:

$$x_i^{t+1} = x_i^t + \chi\, v_i^{t+1}. \tag{15}$$

For each particle, the feasibility is checked, if it is infeasible, the χ parameter is implemented to control its position and velocity. The relation between the modified constriction factor and the age of the infeasible particle is shown in Figure 3.

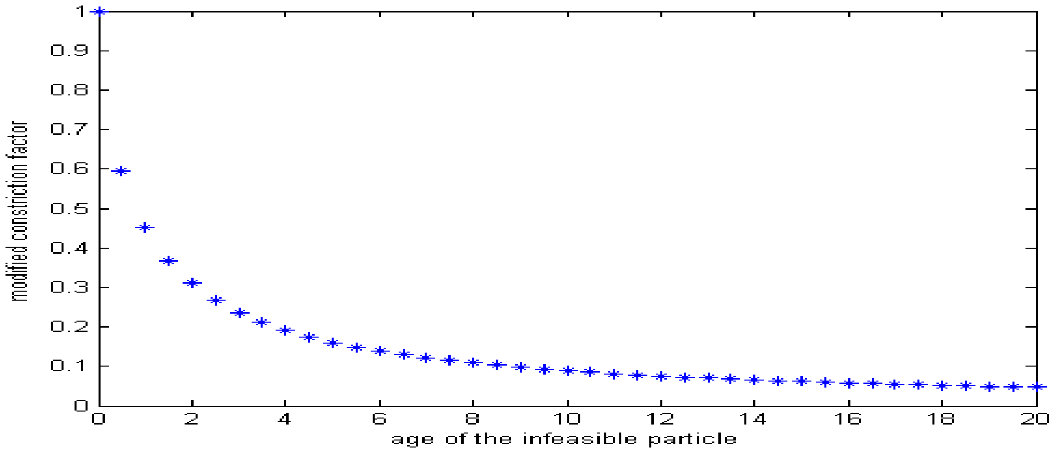


Fig. 3: Relation between the modified **constriction** factor and the age of the infeasible particle.

Step 5: Selection and update the repository
Check the Pareto optimality of each particle. If the fitness value of the particle is non-dominated when it compared to the Pareto-optimal set in a repository, save it into the Pareto repository.
In the Pareto repository, if a particle is dominated from new one, then discard it.

Step 6: Repeat
Repeat again step 2 to step 5 until the number of generation reaches to given t .

The PSO stage algorithm needs at least two Pareto solutions in the first generation to avoid premature convergence.

The pseudo code of the proposed algorithm showing in Figure 4.

Initialize parameters for TR and PSO
TR parameters $(\varepsilon_1,\ \varepsilon_2,\ \tau_0,\ \tau_1,\ \tau_2,\ \tau_3, \Delta_0, \Delta_{\max}, \Delta_{\min})$
PSO parameters $(v_i,\ w,\ c_1,\ c_2)$
Construct the RP model of MOOP
TR algorithm
While the stopping criterion is not met
Compute the normal component of the trial step
The tangential component of the trial step is calculated
Compute the trial step (the sum of normal and tangential component)
Test the new trial step
End while
PSO algorithm.
While number of iterations not met
Set the solution (non-dominated solution) obtained by TR in a repository (particles positions)
Chosen randomly *pbest* and *gbest* from the repository.
Update particles velocity and position
Repair the unfeasible particle
Evaluate fitness of particle swarm
Selection and update the repository
End while

Fig. 4: The pseudo code of the proposed algorithm

NUMERICAL RESULTS
This section is devoted to the discussion of the experimental results. In order to validate the proposed algorithm, several benchmark problems are solved which are reported in the literature [5]. The parameters adopted in the implementation of the proposed algorithm are listed in Table 1 (see [1, 7, 9]).

Table - 1: The parameter adopted in the implementation of the proposed algorithm

Parameter	Value	Parameter	Value
N	20-50	$\Delta_{\max}$	$105\,\Delta_0$
$\varepsilon_1,\ \varepsilon_2$	10-7	$\Delta_{\min}$	10-3
τ_0	0	PSO iteration	300
τ_1	2	w	0.6
τ_2	0.25	c_1	2.8
τ_3	0.25	c_2	1.3
Δ_0	$(1,1.5)\times\Delta_{\min}$	τ	15

Test problems

For evaluating the performance of the proposed approach nine well-known MO benchmark problems are used. Each test problem consists of two objective functions with constraints and has continuous/discrete with Pareto front. The following test problems for study are considered [5]:

BNH Problem

Minimize $f_1(x) = 4x_1^2 + 4x_2^2$

Minimize $f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$

Subject to

$C_1(x) = (x_1 - 5)^2 + x_2^2 \le 25$ $\qquad x_1 \in [0,5]$

$C_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \ge 7.7$ $\qquad x_2 \in [0,3]$

SRN Problem

Minimize $f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2$

Minimize $f_2(x) = 9x_1 - (x_2 - 1)^2$

Subject to

$C_1(x) = x_1^2 + x_2^2 \le 225$ $\qquad x_1 \in [-20,20]$

$C_2(x) = x_1 - 3x_2 + 10 \le 0$ $\qquad x_2 \in [-20,20]$

TNK Problem

Minimize $f_1(x) = x_1$

Minimize $f_2(x) = x_2$

Subject to

$C_1(x) = x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\left(x_1/x_2\right)\right) \ge 0$ $\qquad x_1 \in [0,\pi]$

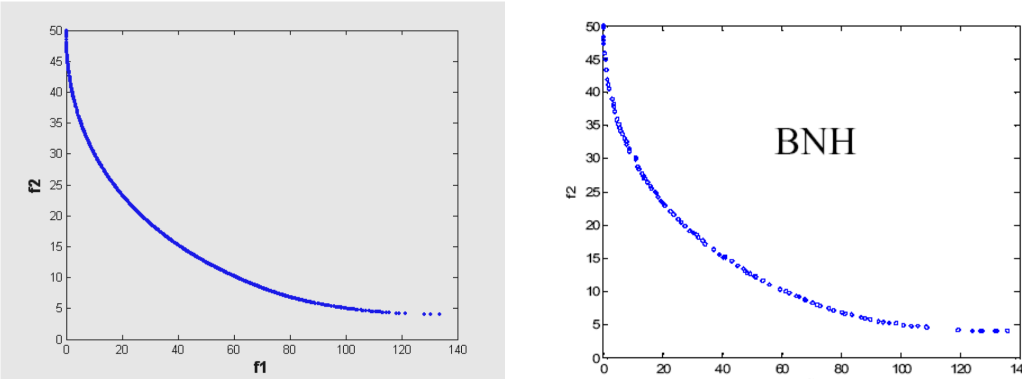
$C_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \le 0.5$ $\qquad x_2 \in [0,\pi]$

RESULTS & DISCUSSION

BNH test problem has two second degree nonlinear objective functions and two nonlinear second order inequality constraints. Its Pareto front generated by the proposed algorithm and Woldesenbet et al. algorithm [20] is given in Figure 5. As it can be seen from the figure, the proposed approach provides feasible optimal solutions that are diversity distributed on the true Pareto front. Also, our approach and Woldesenbet et al. algorithm [20] displayed a better distribution of the Pareto-optimal points but in Woldesenbet et al. algorithm [20] There are gaps between the non-dominated solutions.

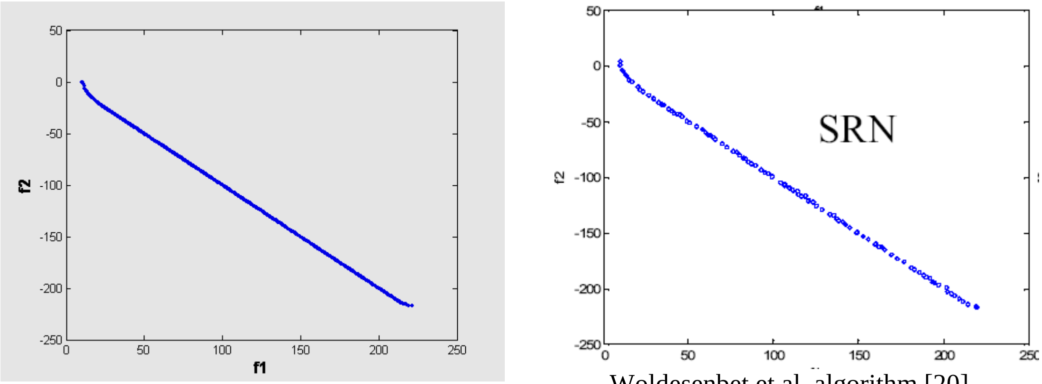
SRN test problem has two second order nonlinear objective functions, one linear inequality constraint and one nonlinear second order inequality constraint. The resulted Pareto front obtained by our approach and Woldesenbet et al. algorithm [20] displayed in Figure 6. From the figure, we can see that our approach able to generate a good approximation of the true Pareto set whish is better than that obtained by Woldesenbet et al. algorithm [20].

TNK test problem has two linear objective functions and two nonlinear second order inequality constraint. Figure 7 present the Pareto front obtained by our approach and Woldesenbet et al. algorithm [20]. We find that the Pareto fronts generated using our approach is well distributed, very closed in the shape of the Pareto fronts and able to produce a continuous Pareto set than Woldesenbet et al. algorithm [20].



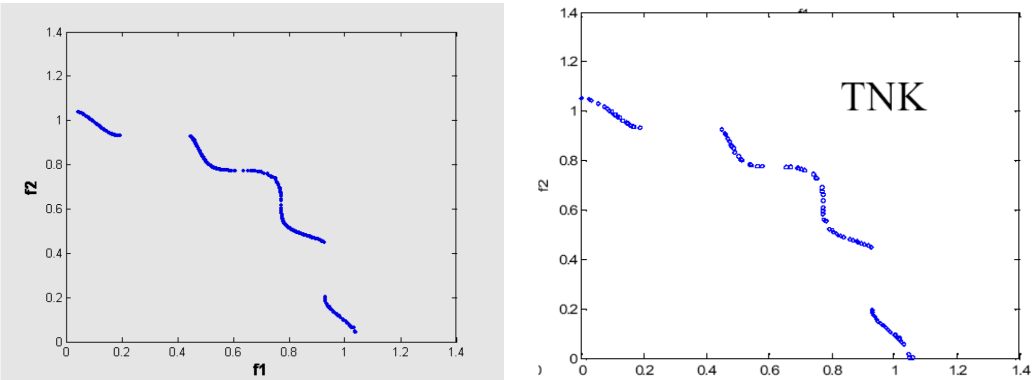
RP-TR/PSO Woldesenbet et al. algorithm [20]

Fig. 5. Pareto **front** for BNH problem.



RP-TR/PSO Woldesenbet et al. algorithm [20]

Fig. 6. Pareto front for SRN problem.



RP-TR/PSO Woldesenbet et al. algorithm [20]

Fig. 7. Pareto front for TNK problem.

PERFORMANCE ASSESSMENTS

There are usually two important aspects of MOO performance. One is the spread across the Pareto-optimal front and the other is the ability to attain the global optimum or final tradeoffs [12]. Every MO optimizer should have the ability of exploration and exploitation to achieve these two goal simultaneously. There are several metrics to express these two aspects with a quantitative assessment.

To evaluate the proposed algorithm, the generational distance (GD) criterion is used [18]. when the optimal Pareto set is known, GD is a way of estimating how far are the elements in the set of non-dominated vectors found so far from those in the Pareto-optimal set and is defined as follows.

$$GD = \frac{\sqrt{\sum_{i=1}^{Nv} d_i^{Nv}}}{Nv}; \tag{21}$$

where Nv is the number of vectors in the set of non-dominated solutions found so far and d_i is the Euclidean distance between each of these and the nearest member of the Pareto-optimal set. If all the solution candidates are in the Pareto-optimal set, then the value of GD is 0.

Table 2 shows the GD criterion for the three test problems. In Table 2, we can see that GD is very small that mean the approximate Pareto obtained by the proposed approach is very near to the true Pareto solution.

Table – 2: The GD criterion for test problems

Test problem	Generational Distance(GD)
Problem (1)	0.00019916
Problem (2)	0.00345000
Problem (3)	0.00224500

This result can approve that the proposed algorithm is able to find well distribution of the Pareto-optimal curve in the objective space. Also, it is observed that the resulting Pareto front is smooth, uniformly distributed, and it achieves very good solutions at the two ends of the curve. In addition, The results have demonstrated that the proposed algorithm can successfully find the Pareto-optimal for all the test problems.

On the other hand, classical techniques aim to give a single point (solution) at each run of problem solving. On the contrary, the proposed approach generates the set of Pareto-optimal solution, which provides the facility to save computing time.

CONCLUSION

This paper presents a hybrid algorithm combining TR and PSO for solving constrained MOOPs. It is a new algorithm that performs random searching with deterministic searching and integrates the merits of both TR and PSO. In the proposed algorithm, MOOP is handled by point RP Interactive Approach, TR is used to obtain a point on the Pareto frontier and homogeneous PSO with a dynamic constriction factor is applied to get all the points on the Pareto frontier. Various kinds of MO benchmark problems showed the effectiveness of the new algorithm and illustrate the successful result in finding a Pareto-optimal set. The following are the significant contributions.

- The present work addressed an important task of combining TR with PSO to not find a single optimal solution, but to find a set of nondominated solutions.
- The integration of TR and PSO has improved the quality of the founded solutions, also it guarantee the faster converge to the Pareto-optimal solution. TR has provided the initial set (close to the Pareto set as possible) followed by PSO to improve the quality of the solutions and get all the points on the Pareto frontier.
- The proposed algorithm does not have any restrictions on the number of the Pareto-optimal solutions found; where it keeps track of all the feasible solutions found during the optimization.
- The numerical results reveal that approach finds a front better than that found by other approach, can generate well-distributed sets of Pareto points very efficiently and is thus very suitable for engineering MOOPs and has good application value.
- Using the GD criterion show that the proposed algorithm give good approximation of the Pareto-optimal solution.

REFERENCES

[1] Abd-El-Wahed W. F., Mousa A. A., El-Shorbagy M. A., Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems, J. Computational Applied Mathematics 235 (2011) 1446-1453.

[2] Byrd R., Robust Trust Region methods for Nonlinearly Constrained Optimization, A talk presented at the second SIAM conference on optimization, Houston, TX (1987).

[3] Coello C.A.C., Veldhuizen D.A.V., Lamount G.B., Evolutionary Algorithms for Solving Multiobjective Problems, Kluwer Academic Publishers, (2001).

[4] C. C. Coello, An Updated Survey of GA-Based Multiobjective Optimization Techniques, ACM Computing Surveys, ACM Press 32 (2000) 109-143.

[5] Deb K., Multi-objective optimization using evolutionary algorithms, NY, USA: Wiley (2001).

[6] Dennis J., El-Alem M., and Williamson K., A trust-region approach to nonlinear systems of equalities and inequalities", SIAM J Optimization 9 (1999) 291-315.

[7] Duff I. S., Nocedal J., and Reid J.K., The use of linear programming for the solution of sparse sets of nonlinear equations, SIAM J. Sci. Stat. Comput. 8 (1987) 99-108.

[8] El-Alem M., A robust trust-region algorithm with a non-monotonic penalty parameter scheme for constrained optimization, SIAM J. Optim. 5 (1995) 348-378.

[9] Fletcher R., Practical Methods of Optimization, second edition, JohnWiley and Sons, Chichester (1987).

[10] Hwang S. K., Koo K., and Lee J. S., Homogeneous Particle Swarm Optimizer for Multi-objective Optimization Problem, ICGST International Journal on Artificial Intelligence and Machine Learning, AIML (2006).

[11] Kennedy J., and Eberhart R. C., Particle swarm optimization, Proceedings of the 1995 IEEE International Conference on Neural Networks, Vol.4, pp.1942–1948, IEEE Press (1995).

[12] Lee T. H., Tan K. C., and Khor E. F., Evolutionary algorithms for multi-objective optimization: Performance assessment and comparisons. Proceedings of the 2001 IEEE Congress on Evolutionary Computation Seoul,Korea (2001) 979–986.

[13] Miettinen K., Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Boston (1999).

[14] Nocedal J. and Wright J. S., Numerical Optimization, Springer-Verlag New York, Inc. (1999).

[15] Omojokun E., Trust-region strategies for optimization with nonlinear equality and inequality constraints, PhD thesis, Department of Computer Science, University of Colorado, Boulder, Colorado, Spares Matrices and Thier Uses, Academic Press, New York (1989) 57-87.

[16] Osyczka A., Multicriteria optimization for engineering design, In John S. Gero, editor, Design Optimization, Academic Press (1985) 193–227.

[17] Pareto V., Cours d'Économie politique, volume I and II. F. Rouge, Lausanne, (1896-97).

[18] Pulido G. T. and Coello C. A. V., Using Clustering Techniques to Improve the Performance of a Multi-Objective Particle Swarm Optimizer, Proc. of the Genetic and Evolutionary Computation Conference, Springer-Verlag, Lecture Notes in Computer Science, 3102: 700-712, Seattle, Washington, USA, June (2004).

[19] Wang F., Zhang K., Wang C., Wang L., A variant of trust-region methods for unconstrained optimization. Applied Mathematics and Computation, 203 (2008) 297-307.

[20] Woldesenbet Y. G., Tessema B. G., Gary G. Yen, Constraint Handling in Multiobjective Evolutionary Optimization Evolutionary Computation, IEEE Transactions on Evolutionary Computation 13 (2009) 514-525.

Source of support: Nil, Conflict of interest: None Declared

