

# Improved Multi-Objective Particle Swarm Optimization Applied to Design Problem

Kapse Swapnil, K. Shankar

**Abstract**—Aiming at optimizing the weight and deflection of cantilever beam subjected to maximum stress and maximum deflection, Multi-objective Particle Swarm Optimization (MOPSO) with Utopia Point based local search is implemented. Utopia point is used to govern the search towards the Pareto Optimal set. The elite candidates obtained during the iterations are stored in an archive according to non-dominated sorting and also the archive is truncated based on least crowding distance. Local search is also performed on elite candidates and the most diverse particle is selected as the global best. This method is implemented on standard test functions and it is observed that the improved algorithm gives better convergence and diversity as compared to NSGA-II in fewer iterations. Implementation on practical structural problem shows that in 5 to 6 iterations, the improved algorithm converges with better diversity as evident by the improvement of cantilever beam on an average of 0.78% and 9.28% in the weight and deflection respectively compared to NSGA-II.

**Keywords**—Utopia point, multi-objective particle swarm optimization, local search, cantilever beam.

## I. INTRODUCTION

MULTI-OBJECTIVE optimization has to deal with conflicting nature of objectives. There are various methods of optimization and particle swarm optimization (PSO) is one of them. PSO is a population based meta-heuristic search technique which is known for its simplicity and quick convergence [1]. Reference [2] shows that the introduction of inertia weight in PSO improved the ability of its global convergence. In the comparative study among algorithms like genetic algorithm, memetic algorithm, PSO, ant-colony system and shuffled frog leaping, PSO performed better in terms of success rate and solution quality [3]. Due to these qualities of PSO, it has attracted many researchers to extend it to multi-objective problems. Multi-objective optimization has to deal with the conflicting nature of objectives and there are specifically two goals to achieve [4]. They are:

- Finding a set of solutions closer to the optimal Pareto-front. Optimal Pareto front is a set of non-dominated particles
- Spreading the solution set uniformly over the Pareto-front known as diversity.

MOPSO was first extended by Moore and Chapman [5]. MOPSO was developed further with novel strategies for obtaining Pareto-front and a global best/leader and many

Swapnil Kapse is a MS research scholar at Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai 600032 India (phone: +918124189471; e-mail: swapnilk121@gmail.com).

Dr. Shankar Krishnapillai is a Professor at Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai 600032 India (e-mail: skris@iitm.ac.in).

algorithms were introduced which has extensive applications. Reference [6] shows a new local search technique to improve diversity and could solve a variety of unconstrained multi-objective problem, but got trapped into local optimal in some benchmark functions. Local search is a heuristic method to solve hard computational problems [7]. Improvement in PSO are of three categories: Extending the search space, hybridizing with other optimization technique and adjusting the parameter [8].

The paper presents an algorithm which has the implementation of a local search based on Utopia point. The algorithm is examined by testing it on standard test functions and also it is applied to a standard cantilever problem whose objective is to minimize weight and deflection.

## II. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

PSO comprises of two main operations: 1. Swarm initialization and 2. Updating swarm with the velocity parameter obtained by particle best and global best. An individual and the swarm move to the optimal solution by gathering information by its own intelligence as well as intelligence of the swarm. The velocity of the particle is for updating the design variable every generation. Equation (1) is the mathematical expression of velocity of the particle. Equation (2) is defining updated particle.

$$v(i+1) = \Psi \cdot v(i) + c_1 \cdot r_1 \cdot (P_{best}(i) - x(i)) + c_2 \cdot r_2 \cdot (G_{best}(i) - x(i)); \quad (1)$$

$$x(i+1) = x(i) + v(i) \quad (2)$$

The current position is denoted by  $x(i)$  and  $x(i+1)$  represents the updated position of the particle.  $v$  is the velocity of the particle and  $\Psi$  denotes the inertia weight coefficient.  $c_1$  and  $c_2$  represent the acceleration components;  $P_{best}$  is the best position an individual has obtained till then and  $G_{best}$  is the best particle encountered so far by the swarm.  $r_1$  and  $r_2$  are the random numbers between 0 and 1.

The inspiration behind the implementation of local search in MOPSO is to achieve improved convergence and diversity to obtain effective results of practical engineering problems. The presented algorithm is comprised of the following steps:

### A. Initialization

$N$  particles are initialized randomly within the decision variable space and the respective fitness values are evaluated.

### B. Personal Best Operator ( $P_{best}$ )

$P_{best}$  operator is used to update the personal best particle after every iteration. If the previous  $P_{best}$  is dominated by the current particle, the  $P_{best}$  is replaced by the current particle

### C. Elite Particles Operator

According to constrained Pareto dominance criteria, the operator finds the non-dominated particles among the swarm [4]. Consider P, Q, R and S as the solution coordinate in the function space. Let  $f_{Px}, f_{Rx}, f_{Sx}$  and  $f_{Py}, f_{Qy}, f_{Ry}, f_{Sy}$  be the fitness values of objective function 1 and objective function 2 respectively as shown in Fig. 1. Solution P is said to be dominating Q if both the conditions are true [4]:

- Solution P is no worse than Q in all the objectives
- Solution P is strictly better than Q in at least one objective.

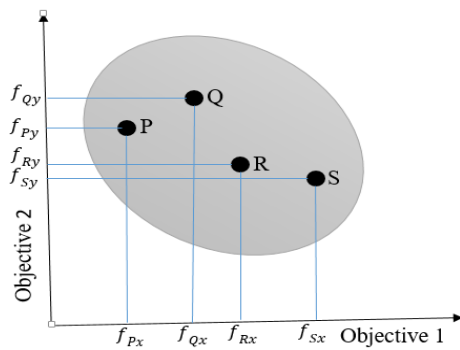


Fig. 1 Concept of dominance for Multi-objective problems

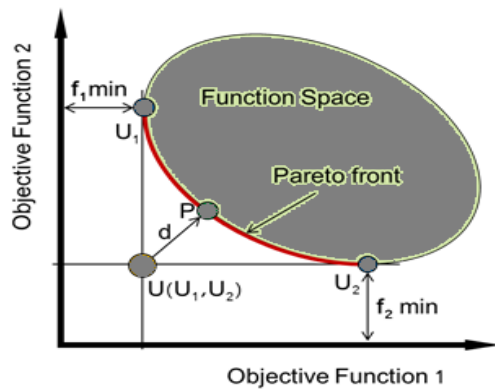


Fig. 2 Utopia point representation for bi-objective problem

### D. Mutation Operator

To enhance the global search ability of the algorithm, polynomial mutation [8] is introduced in the evolutionary cycle.

### E. Local Search Operator

The presented algorithm uses the Utopia point for guiding the swarm towards the True Pareto-front. Utopia point is the optimal solution of the objective function. Let the utopia point of objective function 1 ( $f_1$ ) on the function space be  $U_1$  and the

corresponding design variables be  $x_1^{U_1}, x_2^{U_1}, x_3^{U_1} \dots x_n^{U_1}$  where  $n$  is the number of decision variable. Similarly, Utopia point of objective function 2 ( $f_2$ ) be  $U_2$  and the corresponding design variables be  $x_1^{U_2}, x_2^{U_2}, x_3^{U_2} \dots x_n^{U_2}$ . Fig. 2 shows the schematic diagram of the Utopia point of bi-objective problem with coordinates  $U_1, U_2$ .

After updating the position of the particles, the local search is implemented as follows.

Equations (3) and (4) show the mathematical representation of the Local search scheme:

$$X_{ijG}^p = x_{ijG} + r * (x_{ijG} - x_{ij}^U); \quad (3)$$

$$X_{ijG}^q = x_{ijG} - r * (x_{ijG} - x_{ij}^U); \quad (4)$$

where  $X_{ijG}^p$  and  $X_{ijG}^q$  are the obtained values on the opposite direction of the particle.  $r$  is the random number between 0 and 1. Then, the operator checks for the best particle among the particle and two neighbor particle, according to constrained dominance criteria and the best particle is updated. This local search operator is also implemented on the archive where the elite candidates are stored. This helps in choosing the  $G_{best}$  more efficiently.

### F. Evolutionary Update Operator

After every iteration, the elite particles are stored in the archive. The archive is truncated with the strategy of implementing crowding distance of the particle. The non-dominated elite particles are arranged in the descending order and the Pareto-front is obtained with user defined number of solutions.

### G. $G_{best}$ Operator

Among the non-dominated particles stored in the archive, the  $G_{best}$  operator chooses the candidate as global best which lie in the least crowded region.

## III. SIMULATION

The performance of the presented algorithm was compared with NSGA- II and RMOPSO [9] by rigorous examination of standard test functions available from the literature.

### A. Standard Test Functions

Four standard test functions of the different nature of Pareto-fronts are adopted from the literature, namely: ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 [10]. All these standard test functions are bi-objective unconstrained problems consisting of either 10 or 30 design variables. The ZDT1 function has a convex true Pareto-front, whereas ZDT2 has non-convex. Pareto-front of ZDT3 is discrete and ZDT4 has 21 local Pareto-fronts [11].

### B. Performance Parameters

There are various ways to examine the performance of the algorithm by checking its convergence and diversity of solutions. These are generational distance (GD), inverted generational distance (IGD) and Diversity metric ( $\Delta$ ).

Inverted generational distance (IGD): IGD is the parameter which is used to examine both convergence and diversity of the algorithm. It is the distance of each elite candidate obtained by the algorithm from the True Pareto-front. Equation (5) shows the mathematical expression of IGD metric [12]. It is understood that IGD cannot have absolute value, but it can be utilized for comparing the performance of different algorithms. The lower the value of IGD, the better is the convergence and diversity achieved by the algorithm.

$$IGD(P^*, P) = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \tag{5}$$

C. Selection Parameters

The presented algorithm is compared with NSGA-II and RMOPSO. The value for comparison of IGD metric is adopted from [9]. Table I shows that the parameter selection of NSGA-II and SMPSO.

TABLE I  
INPUT PARAMETERS

Algorithms	Parameter settings [9]
NSGA-II	$N=100, p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20$
SMPSO	$N=100, \omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.0], p_m = 1/n,$

As NSGA-II and SMPSO have been tested with population size of 100, the population size for the presented algorithm is taken as 100 and the inertia weight  $\omega$  is chosen according to [13] which is varied from 0.7 to 0.4 over every iteration. To overcome premature convergence of the algorithm, velocity limits [14] are set as  $v \in [-v_{max}, v_{max}]$ ,  $v_{max} = 0.3 * (x_{max} - x_{min})$ . Acceleration coefficients are chosen randomly between 0 and 1. Each test function is independently tested for 30 runs with 10 iterations in every run.

IV. RESULTS

The simulation results obtained by the algorithm are shown in the IMOPSO column of Table II. As stated above, the IGD values of NSGA-II and SMOPSO are adopted from [9] and tabulated in 1<sup>st</sup> and 2<sup>nd</sup> column respectively in Table II. The maximum number of iterations performed by the presented algorithm named as ‘Improved multi-objective particle swarm optimization’ (IMOPSO) in every run is 10.

TABLE II  
IGD VALUES OF ALGORITHMS ON ZDT FAMILY FUNCTION

		NSGA-II[9]	SMPSO[9]	IMOPSO
ZDT1	Mean	0.0051	0.0071	0.0009
	SD	0.000233	0.000783	0.000191
ZDT2	Mean	0.009	0.0061	0.001459
	SD	0.28	0.0182	0.000457
ZDT3	Mean	0.0075	0.023	0.006303
	SD	0.00271	0.00599	0.002727
ZDT4	Mean	0.29	0.56	0.2821
	SD	0.4	0.0412	0.147
ZDT6	Mean	0.0062	0.0046	0.001445
	SD	0.000702	0.000336	0.001017

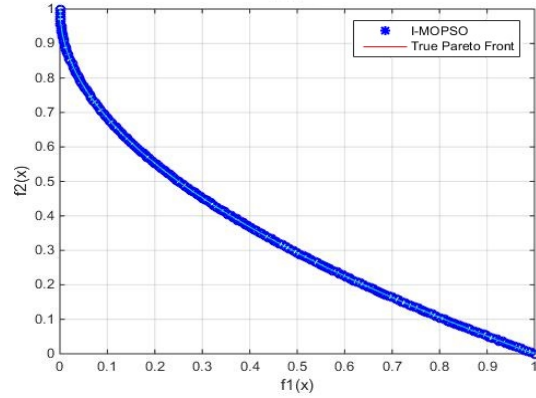


Fig. 3 Pareto-front of ZDT1

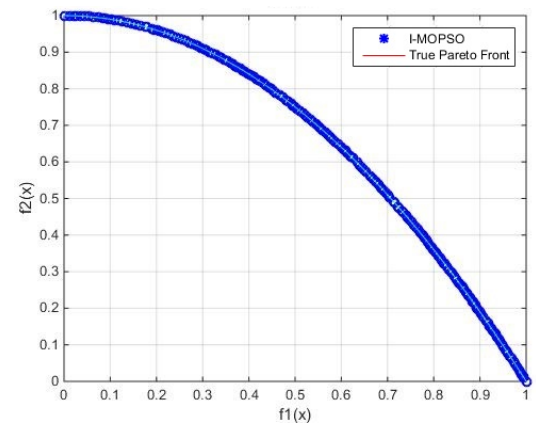


Fig. 4 Pareto-front of ZDT2

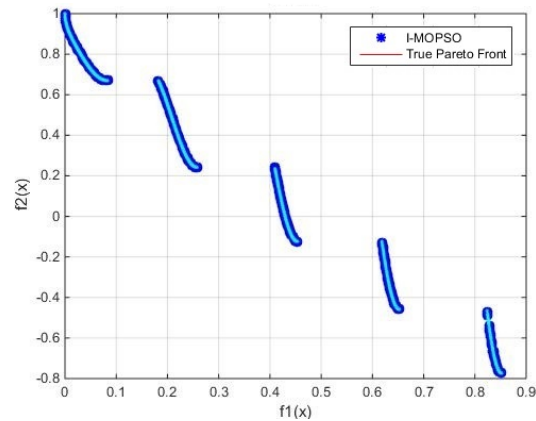


Fig. 5 Pareto-front of ZDT3

According to IGD values shown in Table II, it is observed that IMOPSO significantly outperforms NSGA-II and SMPSO for ZDT1, ZDT2, ZDT3 and ZDT6 test function. NSGA-II and IMOPSO have performed comparatively equal in case of ZDT4. Figs. 3-6 show the obtained Pareto-front of ZDT1, ZDT2, ZDT3 and ZDT6 respectively obtained by IMOPSO which signifies the closeness and uniformity achieved by the proposed algorithm. The light color line in Fig. 3 represents the true Pareto-front and the dark dots represents the obtained

Pareto-front of ZDT1 by the algorithm.

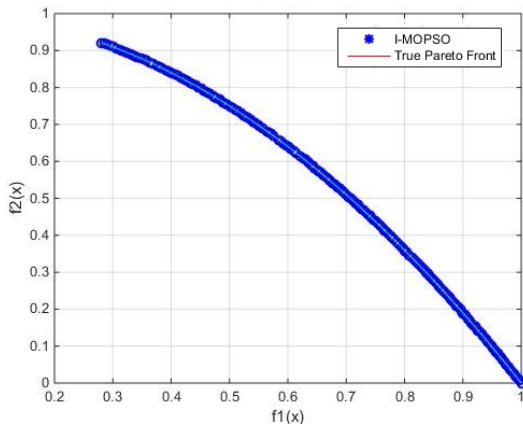


Fig. 6 Pareto-front of ZDT6

V. ENGINEERING DESIGN PROBLEM

To examine the applicability and efficiency of the algorithm, engineering design problem is very useful. In this paper, the detail analysis of the cantilever beam problem is taken into consideration.

A. Cantilever Beam Design

The objective of the cantilever beam design is to minimize the weight and deflection of the beam subjected with the constraint of maximum developed stress and end deflection of the beam. Fig. 7 shows the cantilever beam with  $d$  as the diameter and  $l$  as the length of the beam with the vertical load of  $P$  as the tip. The young modulus ( $E$ ) and density ( $\rho$ ) of the material of the beam are  $207\text{ GPa}$  and  $78000\text{ kgf/m}^3$  respectively. The value of load  $P$  is  $1000\text{ N}$ . The limit of maximum stress ( $S$ ) is  $300\text{ MPa}$  and maximum end deflection ( $\delta_{\max}$ ) is  $5\text{ mm}$ . Table III shows the upper and lower bounds of the diameter and length of cantilever beam which are the design variables.

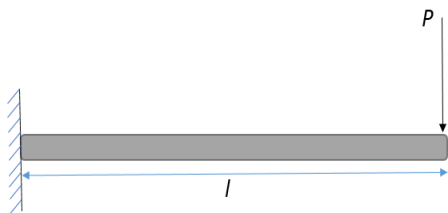


Fig. 7 Cantilever beam

$$\text{Objective functions: Min} \begin{cases} \text{Weight} = f_1 = \frac{\rho \pi d^2 l}{4} \\ \text{Deflection} (\delta) = f_2 = \frac{64Fl^3}{3E\pi d^4}; \end{cases} \quad (6)$$

Constraint: 1. Maximum developed stress must be lower than the allowable stress  $S$ .

$$\text{i.e. } \frac{32Fl}{\pi d^3} - S \leq 0$$

2. The actual deflection at the tip of the cantilever must be less than the maximum deflection ( $\delta_{\max}$ )

$$\delta - \delta_{\max} \leq 0$$

TABLE III  
DESIGN VARIABLES UPPER AND LOWER LIMITS

Sr. No	Variables	Bounds (mm)
1	Diameter of the beam	$10 \leq d \leq 50$
2	Length of the beam	$200 \leq l \leq 1000$

B. Results

The results of IMOPSO are compared with NSGA-II, taking into consideration the four best solutions from Pareto-front of the proposed algorithm and NSGA-II. Table IV shows improvement in the weight and table V shows for improvement in the deflection of the cantilever beam.

TABLE IV  
COMPARISON OF WEIGHT

S. No	Weight $f_1 = \frac{\rho \pi d^2 l}{4}$	% improvement in weight	
	NSGA-II [4]	IMOPSO	
1	0.44	0.44	0
2	0.58	0.57	1.72
3	1.43	1.41	1.39
4	3.06	3.06	0

TABLE V  
COMPARISON OF DEFLECTION

S. No	Deflection $f_2 = \frac{64Fl^3}{3E\pi d^4}$	% improvement in weight	
	NSGA-II [4]	IMOPSO	
1	2.04	1.98	2.94
2	1.18	1.17	0.85
3	0.19	0.19	0
4	0.06	0.04	33.33

The average improvement in the weight of the cantilever beam is obtained as 0.78% and the average improvement in the deflection is 9.28%. Fig. 8 shows the 200 well-distributed solution of cantilever beam design for weight vs deflection.

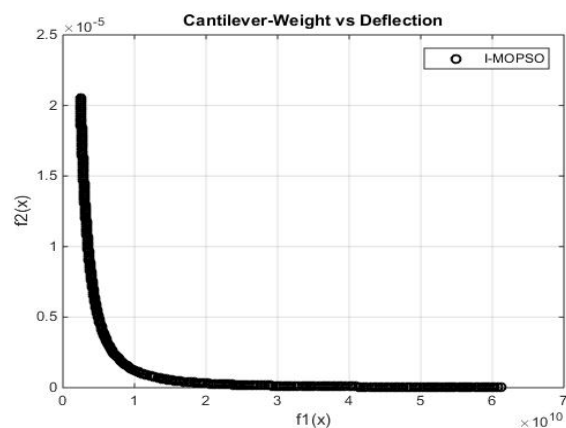


Fig. 8 Pareto-front of cantilever beam

## VI. CONCLUSIONS

The presented algorithm effectively evaluates the Pareto-fronts of the standard test functions. The implementation of local search improved the rate of convergence and thus improved the performance of the presented algorithm. The IGD metric results showed that the presented algorithm outperformed NSGA-II and SMPSO in achieving the convergence and diversity of Pareto-front.

## ACKNOWLEDGMENT

The authors gratefully and sincerely thank the academic community of IIT Madras for providing all the resources and support.

## REFERENCES

- [1] J. Kennedy & R. C. Eberhart, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.
- [2] Y. Shi and R. Eberhart, Parameter selection in particle swarm optimization. In *Evolutionary Programming VIZ: Proc. EP98*, New York: Springer Verlag, 1998, pp. 591-600.
- [3] E. Elbeltagi, T. Hegazy and D. Grierson. Comparison among five evolutionary-based optimization algorithms, *Advanced Engineering Informatics* Volume 19 Issue 1, Pages 43-53
- [4] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Chichester, UK. ISBN: 047187339X, 2001
- [5] J. Moore & R. Chapman. *Application of Particle Swarm to Multi-objective Optimization*. Department of Computer Science and Software Engineering, Auburn University, 1999.
- [6] Q. Lin, J. Li, Z. Du, J. Chen, Z. Ming. A novel multi-objective particle swarm optimization with multiple search strategies. *European Journal of Operational Research* 247, 2015, 732–744
- [7] K. Harada, K. Ikeda & S. Kobayashi. Hybridization of Genetic Algorithm and Local Search in Multi- objective Function Optimization: Recommendation of GA then LS *GECCO'06*, July 8–12, 2006, Seattle, Washington, USA
- [8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197. 2002, Doi: 10.1109/4235.996017
- [9] M. L. Zheng, Q. Wang, S. N. Zhang & S. Y. Zheng. Population recombination strategies for multi-objective particle swarm optimization. *Soft Computing*, 2016. DOI 10.1007/s00500-016-2078
- [10] E. Zitzler, K. Deb, L. Thiele. Comparison of multi- objective evolutionary algorithms: empirical results, *Evolutionary Computation*. 2000, 8 (2) 173–195.
- [11] J.A. Adeyemo and F.A. O. Otieno. Multi-Objective Differential Evolution Algorithm for Solving Engineering Problems. *Journal of Applied Sciences*, 2009, 9: 3652-3661.
- [12] M. R. Sierra & C. A. Coello Coello. Improving PSO-based multi-objective optimization using crowding, mutation and epsilon-dominance. *Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science, 2005, 3410*, 505–519
- [13] K. P. Tripathi., S. Bandyopadhyay, S. Pal. Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients. *Information Sciences* 177 (2007) 5033–5049
- [14] Kiranyaz, Serkan, Ince, Turker, Gabbouj & Moncef, *Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition*, Springer-Verlag Berlin Heidelberg, 2014, pp. 48-49