A Weighted multi-objective evolutionary algorithm optimization

Amit Kumar Sinha

[1] Department of Mechanical Engineering, SMVD University Katra, Jammu, India-182320

email: amitsinha5050@gmail.com

Abstract

Weight Based Genetic Algorithms (WBGA) have a computational efficacy and non-cumbersome for multi-objective optimization. The solutions obtained in the converged region do not always produce maximum optimization for all the objective functions simultaneously. But, the combination of the solutions from different iteration may yield optimized values for all the objective functions to a satisfactory level. The paper attempts to find a method which keeps the simplicity and computational efficiency of WBGA intact, but at the same time counters the problem of inferior pareto-optimal solutions. This is done by finding such a combinational set of solutions which yields strong values for all the objectives. The paper proposes neutrosophic logic (NL) as a postprocessor to the outcome of the WBGA. The NL assigns a percentage of truth, false and indeterminant value to the obtained solutions. The proposed postprocessor operation has been demonstrated with hand calculations on a test problem, and a complex practical example. The results obtained as compared to WBGA show the emergence of a superior solution-set and reaches in close agreement with NSGA-II, while maintaining the computational efficiency.

Keywords: Genetic Algorithm; Optimization; Local Search Algorithm; Algorithm.

Introduction

In the multi-objective optimization (MOP) problems, the WBGA's are exclusively used. One side effect of this is

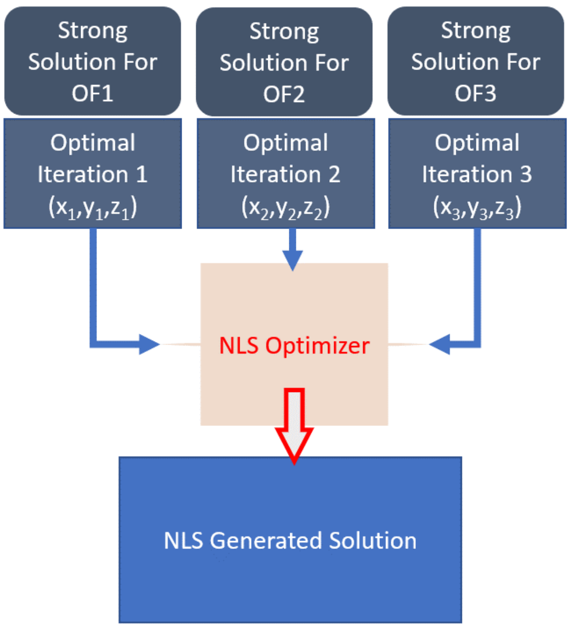

Figure.1. Concept of Neutrosophic Local optimizer

loss of pareto-optimal solutions, when solution space is uniformly dispersed over a non-convex trade-off surface [1,2]. vector evaluated genetic algorithms (VEGA) are exclusively used to deal with this issue, which maintain the use of random-concept of genetic algorithms, which are based on Darwinian theory of natural selection.

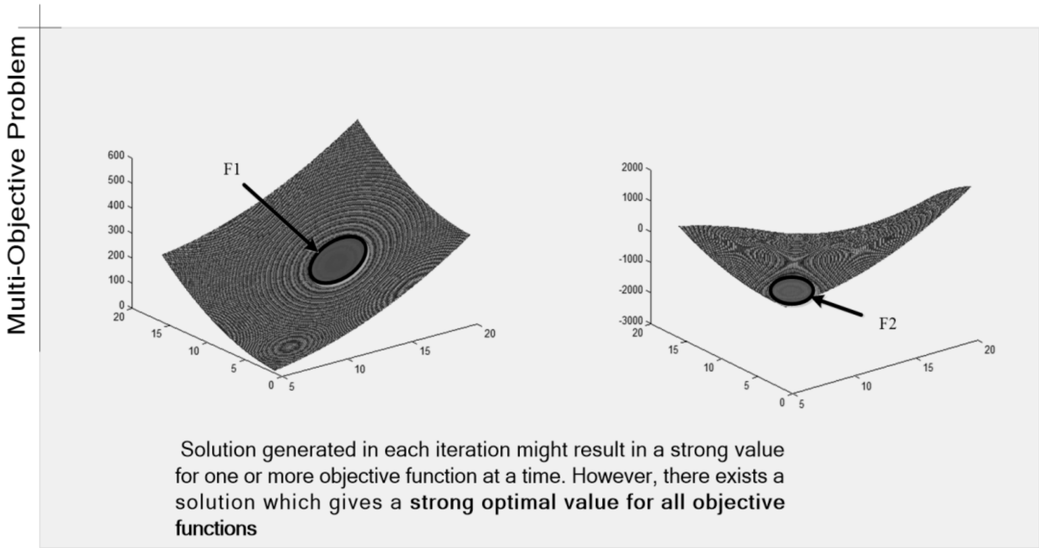

Figure.2. Common Strong Solution Search in a multi-objective function

However, VEGA are computationally cumbersome to implement. Here we show a method to maintain the Darwinian principles of GA for MOP and yet are computationally light. The method does a local search by "fine tuning" of solutions within the obtained optimal set of solutions. This is done by applying NL- which applies a degree of Truth (T), False (F) and Indeterminant (I) to the optimized solutions.

Figure 1 shows the concept of the neutrosophic local search optimizer (NLS), which generates a combination of strong solutions set from different optimal iterations generated by a generalized WBGA scheme.

For an MOP, one set of solution can exhibit strong candidature of optimum solution in one of the objective functions. At the same time, it can be a relatively weak contender in the case of the other objective function. The solution obtained can result into a strong maximized or minimized value in one of the objective functions, while it might vary for the rest of the objective functions and vice versa. Hence, it is safe to assume that there exists a degree of **truth**, **false** and **indeterminance** attached to each solution set. The essence is, one event might be true in one scenario. But the same event might not be true in another scenario. Each solution set obtained is assigned a set of truth, false and indeterminant values. The solution set are divided into **STRONG**, **MEDIUM** and **WEAK** sets.

The need is to find a combination set of solutions, obtained from more than one-iteration, in the near-optimal region to satisfy all the objective functions. the Figure 2 the weighted multi-objective genetic algorithm generates the near optimal solutions in the convergence region. Objective function optimizations are composed of several parameters. However, in the case of multi-objective optimizations utilizing the GA, the obtained values of parameters from iteration (convergence-region) might not produce the optimum value for all the objective functions. It is commonly observed that the combination of parameters obtained in n-th iteration and m-th iteration might produce an optimum value for all the objective-functions.

Problem Statement

When multi-objective optimization problem is solved using the genetic algorithm, in the convergence region a set of near optimal solutions is generated. The solution set and multi-objective functions can mathematically be expressed as:

$$f_1 = f_1(x_1, x_2)$$
$$f_2 = f_2(x_1, x_2)$$
$$s.t.$$
$$l_1 < x_1 < U_1$$
$$l_2 < x_2 < U_2$$

(1)

Where, L and U are upper and lower bounds of the constraints for parameters x_1 and x_2; f_1 and f_2 are objective functions. The solutions generated by GA:

$$x_1 = \{x_{1,1}, x_{1,2}, .., x_{1,i}\}$$

$$x_2 = \{x_{2,1}, x_{2,2}, .., x_{2,i}\}$$

(2)

Where, '*i*' is the iteration number or generation number. While the solution set is (x_1, x_2) and, is represented as:

$$(x_1, x_2)_i = \{(x_{1,1}, x_{2,1}), (x_{1,2}, x_{2,2}) \dots . (x_{1,i}, x_{2,i})\}$$

(3)

From the solution sets generated in (equation 3), the objective functions are evaluated as:

$$\{f_1(x_1, x_2)\} = \{f_{1,1}(x_{1,1}, x_{2,1}), f_{1,2}(x_{1,2}, x_{2,2}) \dots . f_{1,i}(x_{1,i}, x_{2,i})\}$$

(4)

$$\{f_2(x_1, x_2)\} = \{f_{2,1}(x_{1,1}, x_{2,1}), f_{2,2}(x_{1,2}, x_{2,2}) \dots . f_{2,i}(x_{1,i}, x_{2,i})\}$$

(5)

Where,

$x_{1,1}, x_{2,1}, \dots, x_{1,i}, x_{2,i} \in \{x_1, x_2\}$

The evaluation of objective functions due to strong and weak solution sets can be expressed as:

$$\{f_1(x_1, x_2)\}_{strong} = f_{1,1}(x_{1,1}, x_{2,1}) \quad \{f_2(x_1, x_2)\}_{strong} = f_{2,1}(x_{1,1}, x_{2,2})$$
$$\{f_2(x_1, x_2)\}_{weak} = f_{2,1}(x_{1,1}, x_{2,1}) \quad \{f_1(x_1, x_2)\}_{weak} = f_{1,1}(x_{1,2}, x_{2,2})$$

(6)

While, the solution set ($x_{1,1}$, $x_{2,1}$) might result in a strong value of f_1, it need not be so in the case of f_2. Likewise, other solution sets may result in weak or medium values as compared to the desired optimal values of f1 and f2. Hence, when weight functions are used in multi-objective GA for finding the optimized solution, commonly encountered problem is loss of strong solutions.

Flow Chart and Steps

The neutrosophic logic post-processor (NLPP) identifies the near optimal solution candidates which offer the "common -strong" values in all the objective functions under question. The existence of such a solution can also be concluded from the ***Lemma 1*** and ***2***(The supplementary material shows the proof).

Lemma.1. There exists atleast one set of solutions generated at two different iterations of an evolutionary algorithm in the near optimal zone namely, n-th and m-th iteration such that, the objective functions f1 and f2 attain a strong value. The plot is shown in Figure 3.

$$\{f_1(x_1,x_2)\}_{strong} = \{f_1(x_{1,n},x_{2,n})\}$$
$$\{f_2(x_1,x_2)\}_{strong} = \{f_2(x_{1,m},x_{2,m})\} \quad (7)$$

Lemma.2. There exists, atleast one combination set of solutions

$$\begin{cases}(x_{1n},x_{2n})|(\{f_1\}_{strong} \wedge \{f_2\}_{near\_strong}) \\ \quad \vee (\{f_1\}_{strong} \wedge \{f_2\}_{medium})\end{cases}$$

$$\begin{cases}(x_{1m},x_{2m})|(\{f_1\}_{near\_strong} \wedge \{f_2\}_{strong}) \\ \quad \vee (\{f_1\}_{medium} \wedge \{f_2\}_{strong})\end{cases}$$

(8)

by applying T, I, F values to the evaluation of objective functions obtained from each solution set as represented in equation (4) and (5).From the NL theory the percentage of T, I, F for objective function f1, is obtained from the solution set generated by the first iteration of the converged region and is , expressed as:

$$T_{f_{1\_x_{1,1}\_x_{2,1}}} = \{t \pm \delta\}_{f_{1\_x_{1,1}\_x_{2,1}}} = \{t-\delta, t, t+\delta\}_{f_{1\_x_{1,1}\_x_{2,1}}}$$
$$F_{f_{1\_x_{1,1}\_x_{2,1}}} = \{f \pm \delta\}_{f_{1\_x_{1,1}\_x_{2,1}}} = \{f-\delta, f, f+\delta\}_{f_{1\_x_{1,1}\_x_{2,1}}}$$
$$I_{f_{1\_x_{1,1}\_x_{2,1}}} = \{i \pm \delta\}_{f_{1\_x_{1,1}\_x_{2,1}}} = \{i-\delta, i, i+\delta\}_{f_{1\_x_{1,1}\_x_{2,1}}}$$

(9)

Similarly, for objective-function, f_2:

$$T_{f_{2\_x_{1,1}\_x_{2,1}}} = \{t \pm \delta\}_{f_{2\_x_{1,1}\_x_{2,1}}} = \{t-\delta, t, t+\delta\}_{f_{2\_x_{1,1}\_x_{2,1}}}$$
$$F_{f_{2\_x_{1,1}\_x_{2,1}}} = \{f \pm \delta\}_{f_{2\_x_{1,1}\_x_{2,1}}} = \{f-\delta, f, f+\delta\}_{f_{2\_x_{1,1}\_x_{2,1}}}$$
$$I_{f_{2\_x_{1,1}\_x_{2,1}}} = \{i \pm \delta\}_{f_{2\_x_{1,1}\_x_{2,1}}} = \{i-\delta, i, i+\delta\}_{f_{2\_x_{1,1}\_x_{2,1}}}$$

(10)

Let the set of T, I, F values be represented as {t},{i},{f}. The matrix NL, represents the solution space and it's T, I, F values corresponding to its objective functions.

$$\{x_{1i}x_{2i}\} = [NL_{f_1}, [NL_{f_2}]_i \quad (11)$$

The details of the matrix NL are shown in tabular format in the TABLE 1.

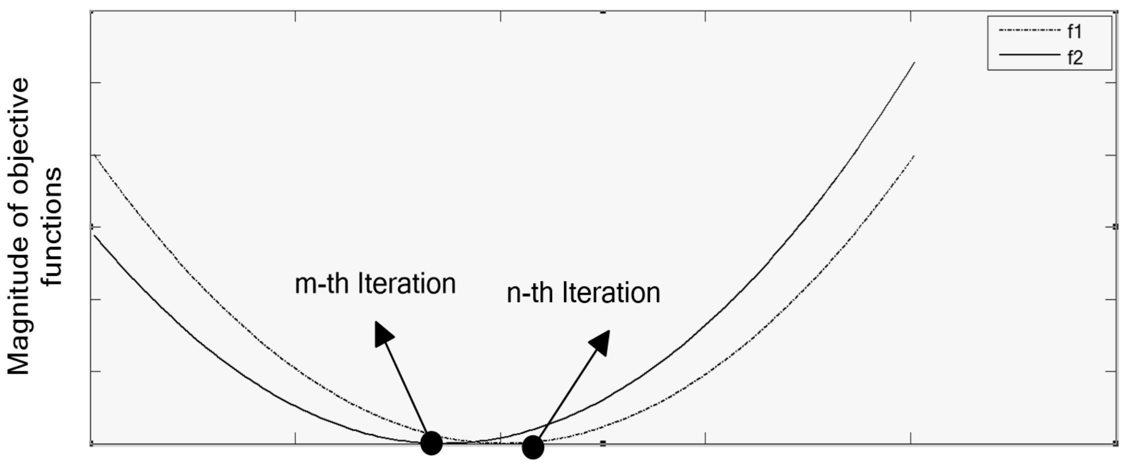

Figure.3. Plot of objective functions

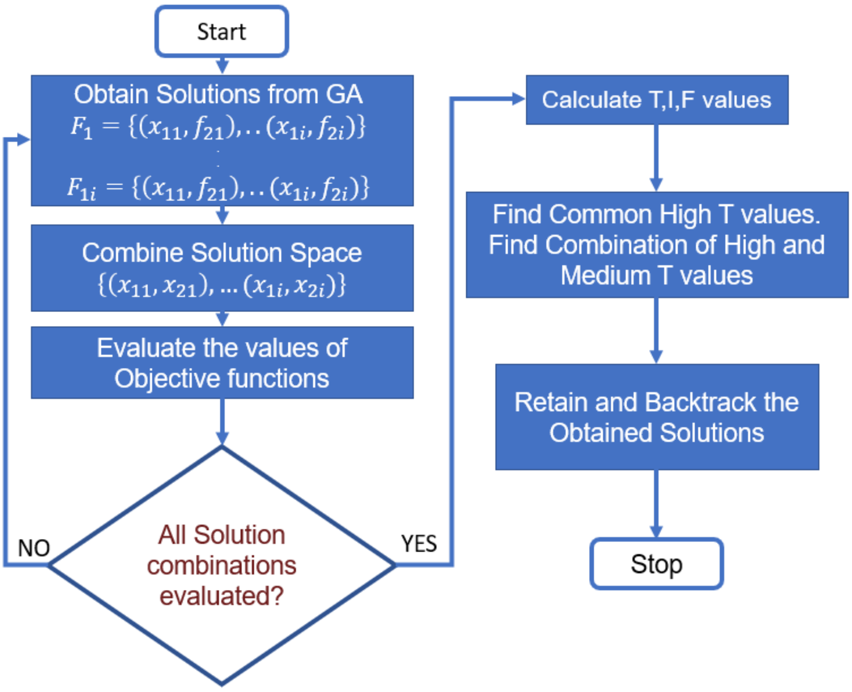

Figure.4. Flow Chart of the NLS post multi objective GA optimization

The Figure 4 shows the flow chart of the NLS post multi-objective GA. The Corresponding example problem is

TABLE 1

NL Matrix

	x_{11}	x_{12}	-	x_{1i}
x_{21}	$[NL_{f_1}, NL_{f_2}]_{11}^{21}$	$[NL_{f_1}, NL_{f_2}]_{12}^{21}$		$[NL_{f_1}, NL_{f_2}]_{1i}^{21}$
x_{22}	$[NL_{f_1}, NL_{f_2}]_{11}^{22}$	$[NL_{f_1}, NL_{f_2}]_{12}^{22}$		$[NL_{f_1}, NL_{f_2}]_{1i}^{22}$
-	-	-	-	-
x_{2i}	$[NL_{f_1}, NL_{f_2}]_{11}^{2i}$	$[NL_{f_1}, NL_{f_2}]_{12}^{2i}$		$[NL_{f_1}, NL_{f_2}]_{1i}^{2i}$

The Supplementary Material shows the implementation of the algorithm for two test cases. In order to evaluate the performance measure of the proposed method, the investigation of the proposed method was done on MATLAB environment. The CPU speed was 2.0 GHz, Intel i3 processor. The method was compared with regular NSGA-II algorithm for the given example II. And, example II demonstrates the validity of the NLPP model for complex problems ranging from decision making to VLSI circuits. The TABLE 2 shows the comparison of WBGA along with its refinement after NLPP is applied. TABLE 3 compares the proposed method with the outcome of NSGA-II algorithm, when applied to example II. The NLPP when applied as a post-processor to WBGA, retains the simplicity of and ease of implementation along with its computational efficiency. It has been observed that the NLPP application to WBGA, eliminates the need to compromise on the pareto-optimal fronts. A generalized theory on the NLPP for the GA has been proposed and presented. The existence of relative percentage of truth, false and indeterminant state in the neutrosophic logic has been exploited to find the best possible solution candidates. The model has demonstrated the evaluation of the common-strong, common-weak and common-medium solution sets for a multi-objective optimization problem. The validity of the model has been demonstrated by the practical examples. They show the results to be in agreement with the proposed NLPP

TABLE 2.

Comparison of results of WBGA and NLPP applied to WBGA.

	Near Optimal Iteration Results			**NL-PP Post-processing results**	
	Iteration1	**Iteration2**	**Iteration3**	**Strong**	Parameters as shown in Table S
WBGA	28.78	30.02	12.28	30.02	
	18.29	21.81	28.825	18.29	
	12.106	12.48	13.76	12.106	
	31.95	53.18	83.25	31.95	
	84.68	55.53	69.01	84.68	
	1.55	2.59	0.92	0.92	
	90.87	69.82	43.04	90.87	
	9.53	8.88	7.7	8.88	
	87.98	74.32	89.245	89.245	
	15.62	26.47	46.83	46.83	

TABLE 3.

Proposed method compared to NSGA-II

Parameters	**NSGA-II**		**WBGA-NLPP**	
Solutions	(W/L)1	30.89	(W/L)1	30.02
	(W/L)2,3	18.01	(W/L)2,3	18.29
	(W/L)4,5	12.07	(W/L)4,5	12.106
	(W/L)6	31.79	(W/L)6	31.95
	(W/L)7,8	84.25	(W/L)7,8	84.68
	(W/L)9	0.98	(W/L)9	0.92
	(W/L)10	91	(W/L)10	90.87
	(W/L)11	9.3	(W/L)11	8.88
	Cm1	89.10	Cm1	89.245
	Cm2	46.27	Cm2	46.83
Computational Time	1008 seconds		703 seconds	
Number of Iterations	25		25	
Population	600		600	

Conclusions

In this paper we have presented an NL post processor for WBGA

Retains Non-Dominant Solutions: NLPP can be an alternative to NSGA-II, it retains strong solution sets.

Computational Efficiency: The performance of WBGA-NLPP is very similar to NSGA-II, however it maintains the simplicity and computational efficiency of WBGA when compared to NSGA-II.

Flexibility: NLPP can be extended to a large number of multi-objective optimization problems. Namely- decision making, artificial intelligence, supply chain management, analog IC design etc.

Random nature of GA intact: The research also shows that the random-operation of principle of evolutionary algorithm has not been hampered by the application of NLPP.

Author Contributions

AS conceived the idea and he was responsible for implementing the algorithm and matlab code.

References

1. A. Konak, D.W. Coit, A.E. Smith, Reliability Engineering & System Safety, 91, 992(2006). https://doi.org/10.1016/j.ress.2005.11.018
2. K.Deb et al. *IEEE Transactions on Evolutionary Computation*, 6, 182(2002). https://doi.org/10.1109/4235.996017
3. H. Eskandari, et al. *Journal of Heuristics*, 14, 203(2008). https://doi.org/10.1007/s10732-007-9037-z
4. A.K. Sinha, A. Anand, *Applied Soft Computing*, 86,105921(2020) https://doi.org/10.1016/j.asoc.2019.105921
5. A.K. Sinha, D.Y. Kim, D. Ceglarek, *Optics and Lasers in Engineering*, 51, 1143(2013). https://doi.org/10.1016/j.optlaseng.2013.04.012
6. J.Wang and X. Zhang, *Symmetry* (MDPI publishers) 11, 1074(2019) https://doi.org/10.3390/sym11091074
7. F. Smarandache and L. Vlădăreanu, *IEEE International Conference on Granular Computing, Kaohsiung,* 607(2011). https://doi.org/10.1109/GRC.2011.6122666