Research Journal of Recent Sciences ISSN 2277-2502
Vol. **2(ISC-2012),** 12-16 **(2012)** *Res.J.Recent.Sci.*

Review Paper

Selection Based Efficient Algorithm for Finding Non Dominated Set in Multi Objective Optimization

Sharma Shailja[1], Singh Garima[2] and Singh Krishna Vir[3]
[1,2]Department of CS / IT, S.I.T, Mathura, INDIA
[3]Department of CS / IT, IIMT, Gr. Noida, INDIA

Available online at: www.isca.in
Received 29th November 2012, revised 30th December 2012, accepted 28th January 2012

Abstract

Non Dominated Sets always plays vital role in solution strategies for multi objective optimization, as the appropriateness of the solution is dependent on the selection of the sets hence efficient search for the optimal solution is dependent on the Non Dominated Sets. Finding Non Dominated set from the set of solutions is very time consuming so to increase the overall performance of the solution strategy an efficient approach is highly in demand. In this paper we have proposed a Selection Based Algorithm which finds effective Non Dominated sets among the set of solutions by establishing dominance among solutions in very less time as compared to the previous approaches.

Keywords: Non Dominated Sorting, Multi Objective Optimization, Non Dominated Set, Selection Based Approach, Non Dominance.

Introduction

All chalks of human life is full of optimizations, we do optimizations in real life unknowingly, since optimizing a single objective is far more simple as compared to optimizing multiple objectives, because they have different solution best suited for different objectives but finding a single solution which will improve overall solution and making every objective function to produce effective values is a kind of NP Hard problem. For optimizing multiple objectives[1] in a single solution we use Multi Objective Optimization Strategies[2-5]. Among various available strategies an important approach is Non Dominated Sorting Multi Objective Optimization[6]. This approach finds effective solutions in very less time as compared to other available approaches. To establish dominance among the set of solutions, the Pareto Optimal approach is widely accepted, Strength Pareto Evolutionary algorithm II (SPEAII)[7,8] and Non-dominated Sorting Genetic Algorithm II (NSGA II)[9] are the proof. The efficiency of these approaches are dependent on the effectiveness of Non dominated Set of solutions, various approaches have been proposed like naïve and slow method[10], fast algorithm[11], Arenas principal[12] , Jun Du and et al[13], Novel Algorithm[14]. Although by applying objective wise sorting Jun du[13] has improved the best case complexity of the algorithm and in Novel Algorithm[14] the early separation of non dominated points and dominated sets improves the worst case complexity of the algorithm, but improvement is still required in the overall performance of the algorithm for optimization[15,16]. In our approach we have proposed selection criteria for comparison of solution sets based on the sorted merit of the solution, our algorithm comprises of simple steps and improves the overall performance of the algorithm.

This paper comprises of five sections namely Introduction, Background, Proposed Algorithm, Experimental results and Complexity analysis, Conclusion. Introduction is all about the requirement of such kind of strategies, Background is the discussion about the previous approaches, Proposed Algorithm gives the deep insight about the new proposed approach step wise and with the help of an example, Experimental results and Complexity analysis is all about comparative analysis of performance and complexity of the algorithm, Conclusion concludes the paper.

Background

Dominance and Pareto optimality[9]: Most MOO algorithms use the concept of dominance. In these algorithms, two solutions are compared on the basis of whether one dominates the other solution or not.

Definition1: A solution X^1 is said to dominate other solution X^2 if both condition 1 and 2 are true:-
i. The solution X^1 is no worse than X^2 in all objectives or $f_j(X1) \leq f_j(X2)$ for all j=1,2,….m.
ii. The solution X^1 is strictly better than X^2 in at least one objective, or $f_j(X^1) \prec f_j(X^2)$ for at least one j={1,2,3….m}.

Definition 2: (Non-dominated set): Among a set of solutions P, the non dominated set of solutions P' are those that are not dominated by any other member of the set P.

Definition 3: (Globally Pareto Optimal Set): the non-dominated set of the entire feasible search space S of globally Pareto optimal set.

Previous Approaches: Before discussing the proposed algorithm, let us have a glance of previous algorithms.

Kung's Algorithm[15]: Kung algorithm is the most efficient and widely used one. In this approach Kung's first sort the population in descending order in accordance to first objective function. Thereafter, the population is recursively halved as top (T) = Front (P (1)-P (|P/2||)) and bottom (B) =Front (P (|P|/2+1)-P (P)) subpopulations. As top half (T) is better in objectives, so we check the bottom half for domination with top half. The solution of B which is not dominated by solution of T is merged with members of T to from merged population M.

Sorting based algorithm (Jun Du Algorithm)[7]: Jun Du has given a sorting based algorithm for finding non dominating sets. The sorting technique is applied in various steps of the algorithm.

The population of solution is sorted according to the descending order to every objective functions. Score all the solution according to the positions of solution in original problem sets. Take the score summation of every solution and sort it to get one new solution summation sequence. The new summation sequence could be used for finding non dominated set by deleting all the dominated solutions in side. Than JUN Du divide the summation sequences in two parts – the bottom summation sequences are used as start of compared solution and top summation sequences are used as start of the comparing one. If compared solution is dominated by comparing summation sequences than it is deleted and then finally get the set of non dominated solutions as a compared solution.

Proposed Algorithm

In our approach we have used the fundamental definitions for Pareto Optimality from the Definition 1 and 2 we can conclude following points: i. If in the solution set for any objective the best value achieved by any solution, this solution is strictly non dominated solution, because for the particular objective it has the best value hence could not be dominated by any other solution, such sets are primary non dominated points. ii. If any solution has worst value for any objective function, such solution can not dominate any other solution in the set. iii. If the solutions to be compared for dominance are better than each other for different objectives then both the solutions are Non Dominated with respect to each other. Such a case if a solution is not dominated by any other solution will be considered as Non Dominated Point. iv. On the basis of Property[1], we have formed set S1, if any solution fulfills property[2] and the solution is not part of set S1, such solutions will not be considered for further comparison. v. If any solution which is still remaining in the original set and not belongs to S1 and S2, we will select the second best solutions from the objective function wise sorted lists and compare it with the solutions of the set S1 only, as the solution under comparison is second best in the particular objective so it could be only dominated by primary dominated points only, if the solution gets dominated then delete it else add it to S1, further third best solutions from the sorted list will be compared with updated S1, and the process moves on till all the sets existing in the remaining original set get compared at least once.

Proposed algorithm will comprise of following steps: i. Sort all solution sets in decreasing order for every objective function and form sorted lists ($O_1 \ldots O_m$) 1 to m in parallel. ii. Select all the solution sets comprising O_{i1} elements from the sorted list and form set S1. iii. Select all the solution sets comprising O_{in} elements and if the solution is not part of set S1, form set S2 and put all such elements in it.

iv. For (j =(2 to n)of all objective functions(i)) 2 to n in parallel { Compare Solution set comprising O_{ij} with Set S1
If ($S_j \prec$ S1) Then Delete Else S1=S1 $\cup$ S_j}
Repeat step 4 till all the solution sets get compared with S1 at least once.

v. Print final Set S1.

Detail of algorithm

To make the algorithm more clear, let us explain the example taken from Jun Du's[7] paper, we illustrate the working of above steps on this example. First, let's take out an MOO example with[10] solutions to[4] objectives the following table 1 presents the[4] objectives (O1-O4) function values for[10] solutions (P1-P10).

Table-1
Objective Function Values

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
O1	0.94	0.35	0.76	0.88	0.39	0.86	0.27	0.91	0.73	0.53
O2	2934	3599	2780	1998	3476	3331	2597	2318	3273	4055
O3	5.3	6.6	5.4	8.0	8.7	7.9	9.1	2.1	4.9	7.7
O4	289	45	23	598	444	99	188	239	177	328

According to step1 P1….P10 are sorted in descending order to O1…
O4, therefore the following Table2 is presented.

Table-2
Sorted Solution Sequence

Sorted List	Sorted solution sequence (Objective wise)
O1	{P1,P8,P4,P6,P3,P9,P10,P5,P2,P7}
O2	{P10,P2,P5,P6,P9,P1,P3,P7,P8,P4}
O3	{P7,P5,P4,P6,P10,P2,P3,P1,P9,P8}
O4	{P4,P5,P10,P1,P8,P7,P9,P6,P2,P3}

In this procedure we will use two main properties of non dominated points. First according to properties of dominated point, it is clear a solution will be dominated only if all its objective functions value is worse than other solution. With this definition we can infer that if a solution has any of its objective value good in comparison with other solution, then both solution will be non dominated.

Let us explain this, for example in list O1 Solution P1 is non dominated because it is best in one objective, Second Solution P8 is better from solutions {P4, P6, P3, P9, P10, P5, P2, P7}, so these solution can not dominate P8.only P1 can dominate this solution. Similarly P6 can be dominated by {P1, P8, P4} no other solution can dominate this solution. This property holds true for every solution.

According to step 2, we will create set S1 .Move all first element of the list in S1 so set S1 will contains following points {P1, P10, P7, P4}.

According to step 3 create S2 which will contain all last elements of sorted solution of objects.

According to step 4 compare elements of S1 with the elements of second column of solution set. First element of second column is P8 (which does not belong to S1 and S2) with the contents of S1 (column wise). Here we compare whole set of P8 {.091, 2318, 2.1, 239} to P1 {.94, 2934, 5.3, 289}. P8 is dominated by set P1 so we delete P8 and move further to P2 repeat same procedure and compare with the elements of S1. This is also dominated by P10 so we delete this. Now we compare P5 to contents of S1. And it is not dominated by all the elements of S1 so P5 is added in S1.

Now move to 3rd column of solution set and compare all the elements of S1 with the elements of this column. If any solution already presented in S1 and S2 then no need to compare it again. If it is dominated by any element of S1 then delete it. And if any element of this column is not dominated by S1 then add it to S1.Repeat this procedure until all the columns have not been compared. S1 will contain all the non dominated sets finally. Similarly we move third best solution and follow same procedure. And continuously update set S1. Repeat this process for all columns. Finally our non dominated set will be S1 {P1, P10, P6, P9, P5, P7, P4}.

Experimental Results and Complexity

Experimental Results and Performance Graph: Comparison of Jun du's algorithm and proposed algorithm are performed on a computer with Intel core™2 Duo 2.10G Hz CPU and 3GB memory. Running time of the algorithms is taken as the criterion to evaluate the efficiency. Various objectives number and solutions number are set for test.

The objective function values are chosen randomly. To remove experimental error, comparisons are performed more than once. Tables 3 and 4 contain the experiments results, where M is objective number, N is the no of solution set for the problem and I is the size of non dominated set. From the table, it is clear that our algorithm is more efficient than Jun du's algorithm.

Table-3
Running Time analysis of Novel algorithm and Proposed Algorithm for 3 objective functions

M(No. of objective functions)	N (Population Size)	Running Time		I(No. of Non-dominated solutions)
		Proposed Algorithm	Novel Algorithm	
3	100	0.117	0.1760	10
		0.106	0.1860	19
		0.1200	0.1800	17
3	500	0.6622	1.1590	15
		0.9360	1.4040	34
		0.6891	1.2060	21
3	1000	1.8725	3.2770	35
		1.6866	2.5300	20
		2.0754	3.6320	20

Table-4
Running Time analysis of Novel algorithm and Proposed Algorithm for 3 objective functions

M(No. of objective functions)	N(Population Size)	Running Time		I(No. of Non-dominated solutions)
		Proposed Algorithm	Novel Algorithm	
4	100	0.1780	0.2670	25
		0.1291	0.2260	22
		0.1513	0.2270	23
4	500	1.4373	2.1560	52
		0.9708	1.6990	31
		1.742	2.6130	51
4	1000	3.7766	5.6550	71
		2.9331	5.1330	62
		4.9960	7.4940	98

In step 1 the time taken by quick sort will be N (log N) per list. So overall complexity of all sorting algorithm is O (M.N (log N)).

Complexity of step 4 is O (M I (N - M)) where I is the no of set of Non-Dominated solution sets, M is the no of objective functions and N is the total no of solution sets. This is the worst case complexity when no element in set S2.

Average case complexity of the proposed algorithm is O (M I (N – (M+L))). Such that there will be L no of solutions which would qualify for the set S2.

The Best Case Complexity will remain O M (N Log N) as this is the complexity for sorting M lists.

Conclusion

The algorithm proposed in the paper is finding non-dominated set efficiently by three steps: sorting, deleting and selection Step. The time complexity analysis shows that this algorithm is better than any other algorithm in its average and worst case analysis. Also in best case its complexity is same as of Jun Du's algorithm which has better complexity in comparison of other traditional algorithm and Kung's algorithm, It has already been proved in the paper[1].

Acknowledgement

First and foremost we bear our sincere thanks to Mr. K.K. Mishra who guided us in each and every work related to our research and helped throughout in overcoming the obstacles we encountered. Second we are very much thankful to Mr. Deepak Kumar singh for vital encouragement and support. He kindly offered invaluable detailed advices.

References

1. Sharifi M. and Shahriari B., Pareto Optimization of Vehicle Suspension Vibration for a Nonlinear Half-car Model Using a Multi-objective Genetic Algorithm, *Res.J.Recent Sci.,* **1(8),** 17-22**(2012)**

2. Fonseca C. and Fleming P. J., Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, In S. Forrest (Ed.), Proceedings of the *5th International Conference on Genetic Algorithms, San Mateo, Californi*, Morgan Kaufmann.**,** 416C423 **(2003)**

3. Horn J., and Nafpliotis N., Multiobjective optimization using the niched Pareto genetic algorithm, *IlliGAL Report 93005*, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign, **(1993)**

4. Horn J., Nafpliotis N., and Goldberg D. E., A niched Pareto genetic algorithm for multiobjective optimization, In Proceedings of the *1st IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Computation, Volume 1, Piscataway, NJ, *IEEE***,** 82C87 **(1994**)

5. Freitas A., A critical review of multi-objective optimization in data mining: a position paper, *SIGKDD Explorations***, 6(2),** 77-86 **(2004)**

6. Srinivas N. and Deb K., Multiobjective optimization using non-dominated sorting in genetic algorithms, *Evolutionary Computation,* **2(3),** 221C248 **(1985)**

7. Zitzler E.,Laumanns M. and Thiele L., SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *TIK-Report 103. ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switzerland* **(1999)**

8. Zitzler E., Laumanns M., and Thiele L., SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, In Proceedings of the *Evolutionary Methods for Design, Optimization, and Control, Barcelona, Spain*, 19-26 **(2002)**

9. Deb K., Agrawal S., Pratap A. and Meyarivan T., A Fast Elitist non-dominated sorting genetic algorithm for multi-objective optimization:NSGA-II, In proceeding of the *6th International Conference on Parallel Problem Solving from Nature*, 849-858 **(2000)**

10. Deb K., Multi-Objective Optimization Using Evolutionary Algorithms, JOHN WILEY and SONS, LTD, 2001 33-43 **(2000)**

11. Ding L., Zeng S. and Kang L., A fast algorithm on finding the non-dominated set in multi-objectve optimization, In Proceedings of *International Conference on Evolutionary Computation*, 2565-2571 **(2003)**

12. Suqin Tang, Zixing Cai, Jinhua Zeng, A Fast method of Constructing the Non-Dominated Set: Arena`s Principle, *4th international Conference on Natural Computation*, 391-395, ICNC-**(2008)**

13. Du Jun, Cai Zhihua and Chen Yunliang, A Sorting Based Algorithm for finding Non Dominated Set in multi-Objective Optimization, *Third International Conference on natural Computation* **(2007)**

14. Mishra Krishn, Verma Manoj, Singh Krishnavir, A novel algorithm for finding non-dominated set in multi objective optimization, 542-547, IC-AI **(2009)**

15. Kung H., Luccio F., and Preparata F.,On finding the maxima of a set of vectors, *Journal of the Association Computing Machinery*, **22(4),** 469-476 **(1975)**

16. Deepak S.S.K., Applications of Different Optimization Methods for Metal Cutting Operation–A Review , *Research J. Engineering Sci.,***1(3),** 52-58 **(2012)**