World Academy of Science, Engineering and Technology
International Journal of Physical and Mathematical Sciences
Vol:1, No:12, 2007

Interactive Concept-based Search using MOEA: The Hierarchical Preferences Case

Gideon Avigad, Amiram Moshaiov, and Neima Brauner

***Abstract*—**An IEC technique is described for a multi-objective search of conceptual solutions. The survivability of solutions is influenced by both model-based fitness and subjective human preferences. The concepts' preferences are articulated via a hierarchy of sub-concepts. The suggested method produces an objective-subjective front. Academic example is employed to demonstrate the proposed approach.

***Keywords*—**Conceptual solution, engineering design, hierarchical planning, multi-objective search, problem reduction.

I. Introduction

RECENTLY, a concept-based Interactive Evolutionary Computation (IEC) approach has been introduced for the exploration of conceptual solutions in multi-objective engineering design problems [1]. The use of conceptual solutions improves human-machine interface, and enables evaluating concepts, rather then just specific solutions, while taking into account human perceptions and subjective preferences. Dealing with engineering design problems, a progressive goal approach has been taken, where solution concepts are evolved around a dynamic target in a multi-objective space [1]. In [2], an algorithm has been presented that interactively evolves conceptual solutions for a multi-objective path-planning problem, by a Pareto-directed, rather then a progressive goal approach. This allows a non-localized inspection of solutions with respect to the objective space. Here, the technique is extended to deal with conceptual solutions that can be represented by a hierarchical tree of sub-concepts (sub-solutions). Such a situation may occur in problem reduction and in application areas such as engineering design (e.g., sub-systems), and hierarchical planning.

Pareto-based approaches are between the most popular MOEA solution techniques [3]. Surveys and descriptions of such algorithms can be found in several references (e.g., [3-5]). The use of MOEA in conjunction with concepts, which are represented as sets of particular solutions, is not common. Andersson employed MOEA to separately evolve concepts by way of their particular solutions [6]. The novelty of our approach is in the ability to simultaneously evolve several concepts. Moreover, our approach provides a method to combine human preferences and takes into account the hierarchy of sub-conceptual solutions.

Manuscript received July 15, 2005.

G. Avigad, A. Moshaiov, and N. Brauner are with the School of Mechanical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel (e-mails: gideon@ eng.tau.ac.il; moshaiov@eng.tau.ac.il).

II. Methodology

A. The Concept-based Search Problem

In the common multi-objective search problem, such as described in [3,4], the set of Pareto optimal solutions is sought from the set of all possible particular solutions. Any particular solution is characterized by specific values of the problem variables representing a point in the problem variable space. The set of Pareto optimal solutions is found by comparing the performances of all particular solutions in the objective space for non-dominancy. The representation, in the objective space, of the set of non-dominated solutions is known as the Pareto front. Finding the performances of particular solutions is usually done by the use of models.

Commonly the notion of a conceptual solution is associated with abstractive ideas generated by humans. It describes a generic solution to a problem. During a conceptual solution stage, such as initial planning and conceptual design, no particular solution is stated. By a particular solution we mean a fully detailed solution, such that it has a one-to-one relationship with a point in the objective space. Multiple particular solutions might be associated with a conceptual solution, constituting a one-to-many relationship between the conceptual solution and the objective space.

In 'real-life' situations humans rely on their experiences and preferences in choosing a conceptual solution, and eventually they translate the chosen concept into a chosen particular solution. This is usually done with or without the ability to explicitly evaluate the merits of the chosen particular situation. We are interested in supporting humans, such as planners or designers, by computers, while performing a concept-based multi-objective search. In the proposed concept-based search, and in similar investigations, such as in [6-7], the interest is not on the performances of particular solutions, but rather on the performances of conceptual solutions. This is in contrast to the common multi-objective search problem. The proposed search concerns conceptual solutions that can be represented by sub-sets of the set of particular solutions of the problem (see examples in [1,2]). It is further assumed that the performances of each particular solution are computable via models (e.g., tables, parametric models). Each conceptual solution, and its associated particular solutions, may be characterized by different models and/or range of variables, and consequently may possess different performances. Each conceptual solution, may be characterized by a different performance model and/or a range of variables, in comparison with other concepts.

Given the above assumption on the existence of models, a Pareto front can be found for each concept, independently, by the associated sub-set of particular solutions. Eventually, all such Pareto fronts can be further organized to produce one final front (e.g., [6]). Our approach differs from such a sequential approach by two major aspects. First, the concepts are evolved simultaneously, which is far more efficient. Second, their survivability depends not only on computable models, but also on human subjective preferences of sub-concepts.

The simultaneous approach is motivated by its computational efficiency. This stems from the fact that no full development of fronts of bad concepts is expected. In contrast, the sequential technique inherently involves such a development.

The interactivity element is significant for dealing with complex real-life situations. In many such problems the available models are limited. The computation of merits might only partially reflect all issues that are involved in selecting a concept. Difficulties in realizing solutions associated with a particular concept, might not be modeled (e.g., difficulties of manufacturing in design problems, un-modeled hazards associated with plans, which constitute a conceptual plan). The interactivity allows overcoming the lack of such models by employing human preferences. In the following section the foundations for the interactive concept-based search problem are outlined.

B. Hierarchical Representation of Concepts

We use a hierarchical ‘AND/OR’ tree to represent the set of all conceptual solutions of the problem (conceptual solution space). Each concept is represented as a hierarchical ‘AND’ tree, which is extracted from the ‘AND/OR’ tree, by decisions taken at the ‘OR’ nodes. The ‘OR’ nodes are termed D-N (decision node), indicating that a decision on selecting a branch at that node has to be taken for the extraction of a concept. All other nodes on the ‘AND/OR’ tree can be viewed as sub-conceptual solutions (S-Cs) of the problem. Figure 1 depicts such a tree. It includes 10 S-Cs, which are used to express 8 different concepts as further detailed in section III. It is important to note that the current representation of the conceptual solution space is assumed to be a-priori to the search. This means that in the current implementation, the tree is not generated, while searching, as in common AI problems, and the search is limited for innovative rather then creative concepts.

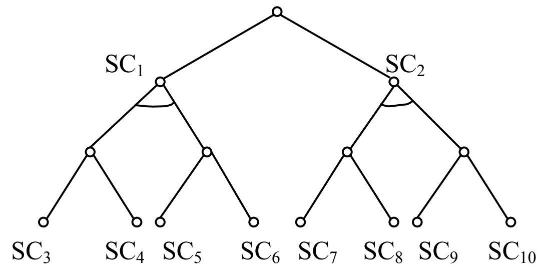

Fig. 1 Conceptual solution space representation

C. Evaluating Concepts and Particular Solutions

Our explicit interest is in the performances of conceptual solutions. The performances of a concept are examined via its associated particular solutions. Each particular solution has its associated point at the objective space. A parallel search process is performed, where a finite set, S, of particular solutions, is examined simultaneously for their relative performances. The resultant Pareto–optimal set may consist of clusters of particular solutions, each associated with one of the concepts.

Machine based fitness - Evaluating a particular solution of S is done by a multi-stage process, which is described below. First we employ a sorting and ranking procedure, following [8], to obtain ranked sets of non-dominated solutions out of S. This is done purely by a model-based evaluation. Next we assign ‘dummy fitness,’ fit_U^r , to each of the particular solutions, according to their rank, and further correct it by two penalty functions (see appendix, and [2] for further details). A machine-based fitness MBF of the i-th individual of the r-th rank, which is based on models, is calculated by

$$MBF_i = fit_U^r - \phi_i^{r,m} - m_i^{r,m} \qquad (1)$$

where, the right hand side of equation 1 is described in the appendix.

Hierarchy-based S-Cs’ weighting **-** The process of evaluating a particular solution is not completed without the insertion of the affect of humans on the evaluation. The fitness of a particular solution should be corrected according to the human preference of the associated concept. The simplest case would be when humans assign preferences weights to each one of the concepts. In [2], we dealt with the less trivial case, where each sub-concept is assigned with a preference weight. Here we deal with a modification to include the affect of hierarchy on the fitness.

The team’s preferences are accounted for in accordance with their location in the hierarchical tree. This is carried out by the following procedure, which makes sure that human preferences of S-Cs are not contradicting to the hierarchy of the S-Cs. Preferences of S-Cs are not to be considered whenever preferences exist at ancestors’ nodes. For example if designers of an airplane reject using a ‘delta’ wing then a high preference towards a bolt arrangement to mount the wing is irrelevant. The team may assign weights to some S-Cs of the problem, with values in the interval [-1, 1], where -1 designates pure dislike, and 1 stands for highest preference. Each S-C$_g$ of the k-th hierarchy (the root has the lowest k), might be interactively assigned with weight, which is designated as w_g^{*k}. Each extracted 'AND' tree, representing a concept, contains n_A paths. For each path j, the highest node with an assigned weight is marked, and it's weight is termed $w_g^{*k(j)}$. Branches, below such nodes are pruned. S-Cs, with no preference, are automatically assigned with zero weights. The weights, of the resulting pruned tree, are used to obtain the m-th concept-weight, H_m, representing the concept preference. Starting from the leaves of the pruned tree, the weight, w(pr), of each parent node, is calculated by averaging the weights of its children, w(ch). The weight of a parent node is:

$$w(pr) = \frac{1}{n_L}\sum_{n=1}^{n_L} w(ch)$$

$$\text{and } H_m = w(root) \tag{2}$$

where, n_L is the number of the node's children. The calculation of the weight of the root, $H_m = w(root)$, is obtained by calculating the weights of the ancestors up to the root node of the 'AND' tree of the m-th concept.

Human-Machine fitness - The human preferences should be combined with the MBF so that evolution is affected by both the objective performances and the subjective designers' preferences. Thus a fitness of a design solution is influenced by its performances and by the preferences of the designers towards the S-Cs, as related to the concept to which the design belongs. The fitness, which results from considering both influences, is termed Human Machine Fitness (HMF), HMF = f (MBF, H). The results presented in section III, were obtained using the following HMF function.

$$HMF = \begin{cases} fit_i \cdot (H+1) & \text{for } -1 \le H \le 0 \\ fit_i + (fit_{max} - fit_i^{max,\, m}) \cdot (H) & \text{for } 0 < H \le 1 \end{cases} \tag{3}$$

where, fit_{max} is the maximal machine fitness over all individuals within the generation, and $fit_i^{max,m}$ is the maximal fitness of an individual belonging to a concept m of the generation. Thus, the fitness of an individual is scaled according to the team preferences.

Objective subjective front – We term the presentation of solutions obtained by the HMF, in the objective space, as the objective-subjective front. It should be noted that this 'front,' might not necessarily possess the non-dominancy characteristics in the objective space. This is due to the fact that it is not based on performances alone, but also on the human preferences.

D. MOEA Implementation

A Compound-Individual, (C-I) holds a genetic code, as described in [1]. The code consists of all S-Cs and all the problem variables, in an 'AND/OR' structure. It enables the evolution of concepts, and the problem variables, simultaneously. For each 'OR' node, a genetic code is used for the competing S-Cs. Decoding the S-Cs' competition code points at the winning S-C of the node, and its related problem variables. The pseudo MOEA for the interactive concept-based multi-objective search is outlined below.

***INITIALIZE**: C-I= compound individuals*
While *team discussion continues*
  Insert w_g^{*k} *- team preferences, interactively*
  While *generation ≤ final generation*
    While *not all individuals' performances computed*
    Decode C-I for extracting winning S-Cs and values of the problem variables
    Compute Performances
  End
    Compute MBF (see appendix and eq. 1)
    Compute H (eq. 2)
    Compute HMF= Scaled human fitness (eq. 3)
      C-I = Reproduce C-I
    C-I = recombine C-I with pc probability
    C-I = Mutate C-I with pm probability
    End
Introduce fronts to the team
End

It is noted that due to the possible shuffling of the initially assigned ranks, by the human intervention, the HMF of a large portion of the population may rise rapidly. This can cause exploitation at a too early stage of the search. Therefore a re-ranking procedure is implemented. The outlined MOEA is used in the following example.

III. Case Studies

The purpose of this bi-objective academic example is to demonstrate the ideas presented in this paper and in particular to show the affect of the hierarchical preferences on the resulting front. The genetic algorithm parameters are detailed in table 1. Where pc, and pm, are the probabilities for crossover and mutation, respectively.

The following objectives are used:

$$\begin{aligned} f_1 &= x^2 + b \cdot c \\ f_2 &= (x-2)^2 + b \cdot d \end{aligned} \tag{4}$$

World Academy of Science, Engineering and Technology
International Journal of Physical and Mathematical Sciences
Vol:1, No:12, 2007

TABLE I
ALGORITHM'S PARAMETERS

parameter	value
q_r	8
δ	40
ε	10
n_p	250
pc	0.7
pm	0.03

The parameter, x, is the problem parameter searched within the interval [-10, +10]. Eight concepts are evolved based on ten S-Cs. The S-Cs, within the hierarchy arrangement of the ‘AND/OR’ tree, are depicted in figure 1. Table 2 provides a list of concepts and their associated symbols (legend) as used in the following figures.

TABLE II
SUMMARY OF CONCEPTS AND THEIR LEGEND

Concept	S-Cs	b	c	d	Legend
1	1, 3, 6	1.0	2.0	1.0	○
2	1, 4, 5	1.0	3.0	2.0	▷
3	1, 3, 5	1.0	2.0	2.0	✳
4	1, 4, 6	1.0	3.0	1.0	□
5	2, 7, 9	1.5	1.0	0.5	+
6	2, 8, 10	1.5	2.0	1.5	◆
7	2, 7, 10	1.5	1.0	1.5	◁
8	2, 8, 9	1.5	2.0	0.5	▼

The parameters b, c, and d, which are associated with the S-Cs, dictate different performance models, (equation 4), for the various concepts. It should be noted that the S-C_1 to S-C_{10} are related to b=1.0, b=1.5, c=2.0, c=3.0, d=2.0, d=1.0, c=1.0, c=2.0, d=0.5, and d=1.5, respectively. It is also noted that the equal values of c=2.0 for S-C_3 and S-C_8 mean that these S-Cs are actually the same, and the different indices is a result of the representation. The problem parameter 'b' characterizes the highest S-Cs of the hierarchy, and the rest of the parameters are associated with the S-Cs of the lower hierarchy. Deciding on the branch at each D-N, leads to an ‘AND’ tree of S-Cs, which corresponds to a concept. Each concept of this example has its unique model resulting from determining the values of the parameters b, c, and d.

Figure 2 shows a part of the initial population. The eight concepts are distributed in the objective space according to their performances, as calculated by equation. 4.

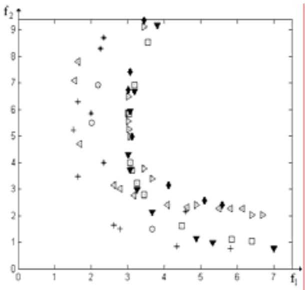

Fig. 2 Initial concepts distribution

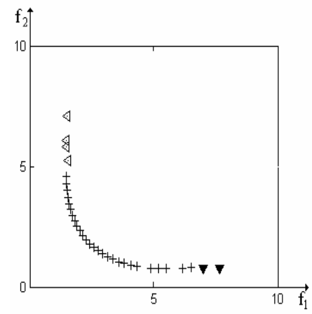

Fig. 3 Machine-based front

Figure 3 depicts the resulting front. It shows that three concepts survived (concepts # 5,7,8). The winning concepts are associated with the S-C of b=1.5. Figure 4 shows the resulted front with preference assignment. A weight of w_1^{*1} = 0.6 for the S-C_6 (associated with b=1), is used. Clearly, this causes a change from the front of figure 3. In addition to concept # 5, which belongs to the branch of b=1.5, a second concept survived belonging to the branch with b=1(concept # 1). It is noted that any assigned weights for S-C_3 $\div$ S-C_6, will not be accounted due to the pruning procedure.

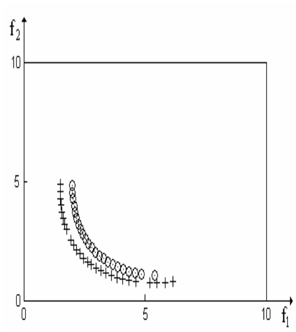

Fig . 4 Obj-Sub front w_1^{*1} =0.6

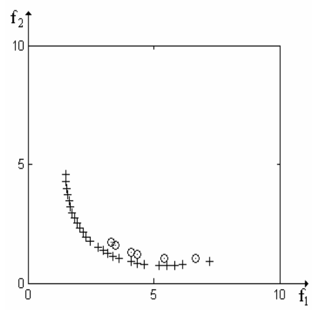

Fig. 5 Obj-Sub front $w_2^{*2} = 0.6$

Figure 5 shows the results, with a change of preference of a S-C, which belongs to a lower hierarchy. S-C_6 (associated with d=1) is assigned a preference weight, w_6^{*2} = 0.6. This assignment causes the survival of concept # 1, yet the resulting front is not as full, in comparison with that of figure 4. This is due to the lower location of the preferred S–C within the hierarchy. When the S-Cs, associated with c=2 and d=1, are both assigned with weights ($w_3^{*2} = w_8^{*2}$, w_6^{*2}) of 0.6, the result is similar to the one depicted in figure 4. Increasing weight w_1^{*1} (the b=1 branch) to 1.0, a further increase in the survivability of concepts, which include S-C_6, takes place, as shown in figure 6.

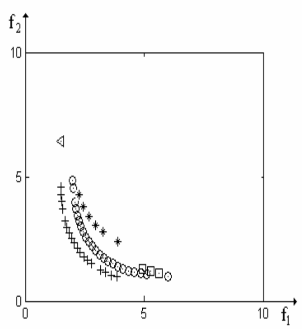
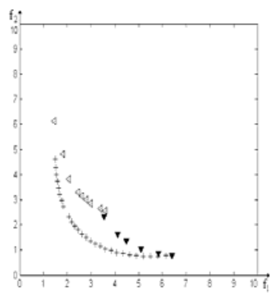

Fig. 6 Subjective front w_1^{*1} =1.0 Fig. 7 Subjective front w_2^{*1} =0.6

Three out of the four concepts, belonging to that branch (b=1), appear in the resulting objective-subjective front. The forth concept did not appear, due to its low performance.

If the branch of b=1.5 is given a higher preference (w_2^{*1} = 0.6), with no preference to S-Cs associated with b=1, a second front is surviving along with a front similar to the initial one. This can be seen by comparing figure 7 with figure 3. Any concept can be retained in the evolution by changing the preferences of its' relevant S-Cs', and those of the competing concepts. For example, concept #6, which has not survived so far, can be elevated by a subjective decision. This can be done by assigning preference weights of to $w_8^{*2} = w_{10}^{*2} = 0.6$ as well as assigning $w_2^{*1} = w_7^{*2} = w_9^{*2} = -0.4$. The resulting front of these assignments, which contains concept # 6 alone, is depicted in figure 8.

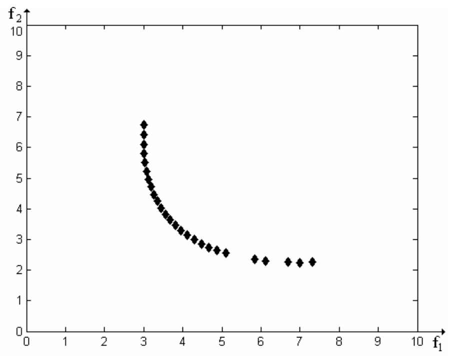

Fig. 8 Concept elevation

IV. Summary and Conclusions

A. Figures and Tables

A concept-based MOEA, which strengthens symbiosis between computers and humans, in exploring conceptual solutions to multi-objective search problems, is presented. In particular the hierarchical preferences of S-Cs is dealt with. The algorithm allows simultaneous evolution of concepts. This is in contrast with methods that evolve each concept at a time, and use a post-evolution creation of a mixed front. In contrast to a theoretical Pareto front, the methodology uses ranked non-dominated sets, which are obtained by objective model-based design performance evaluations. These are further manipulated by subjective evaluations to produce objective-subjective fronts.

An example is given, which demonstrates the performance of our concept-based method.

Appendix

A-1 Rank Assignment based on Non-dominancy

The following algorithm assigns a level of non-dominancy (herby termed rank), r, to each individual. The predefined number of ranks is q_r. The individuals of the first rank are assigned with an initial upper (U) fitness bound, as large as the population size, n_p

$$\mathrm{fit}_U^r = n_p, \quad \text{for} \quad r=1 \tag{A1}$$

For subsequent ranks upper fitness bounds are calculated as follows:

$$\mathrm{fit}_U^{r+1} = \mathrm{fit}_U^r - \delta_0, \quad \text{for } r=1,\ldots, q_r\text{-}1 \tag{A2}$$

Similarly, a lower (L) bound is assigned for each rank according to:

$$\mathrm{fit}_L^r = \mathrm{fit}_U^{r+1} + \varepsilon, \text{ for } r=1,\ldots,q_r \tag{A3}$$

where $\varepsilon << \mathrm{fit}_U^{r+1}$ is a constant that separates between adjacent ranks. As a result, each rank has an available fitness span, δ, where,

$$\delta = \delta_0 - \varepsilon \tag{A4}$$

The available span is reserved for distributing the fitness of the individuals, of the rank, according to front-based concept sharing, and in-concept niching, as explained in the following.

A-2 Front-based concept sharing

The goal of concept sharing is to preserve concept diversity and prevent a good concept from hindering the evolution of other potential concepts. Concept sharing is implemented within each rank. A sharing penalty function for the i-th C-I, belonging to the m-th concept, within the r-th rank is defined as:

$$\phi_i^{r,m} = \frac{0.5\,\delta}{n^r} n^{r,m} \tag{A5}$$

where, $n^{r,m}$ is the total number of C-Is belonging to the m-th concept within rank r, and n^r is the size of the population belonging to rank r.

A-3 In-concept front niching

In our approach, fitness sharing is practiced within each concept, rather than within all the population. This preserves diversity within each concept belonging to a particular rank. A normalized Euclidean distance-measure, following [9], for the i-th and the j-th individuals, belonging to the r-th rank and m-th concept, is computed as follows:

$$d_{ij}^{r,m} = \sqrt{\sum_{n=1}^{n_o} \left(\frac{f_i^n - f_j^n}{f_{best}^n - f_{worst}^n} \right)^2} \qquad \text{(A6)}$$

where, n_o is the number of objectives to be optimized, and f_i^n is the performance of the particular solution of the i-th individual with respect to the n-th objective. Also, $f_{best}^n = \max(f_i^n)$ and $f_{worst}^n = \min(f_i^n)$, are the best and worst performances within objective n, of the individuals, in rank r and concept m. A sharing function, [8], for the i-th individual with respect to the j-th individual, of the r-th rank and the m-th concept, is computed as:

$$sh_{d_{ij}^{r,m}}^{r,m} = \begin{cases} 1-(d_{ij}^{r,m}/\sigma_{share})^2, & if \ d_{ij}^{r,m} \le \sigma_{share} \\ 0, & otherwise \end{cases} \qquad \text{(A7)}$$

where,

$$\sigma_{share} = \frac{0.5}{\sqrt[p]{q}} \qquad \text{(A8)}$$

where q is the desired number of niches and p is the number of the problem variables. The niche count for each individual i, belonging to the m-th concept, and the r-th rank, is computed by:

$$m_i^{r,m} = \frac{0.5\,\delta}{n^r} \sum_{j=1}^{n^{r,m}} sh_{d_{ij}^{r,m}}^{r,m} \qquad \text{(A9)}$$

REFERENCES

[1] G. Avigad, A. Moshaiov, and N. Brauner , "Concept-based interactive brainstorming in engineering design," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 8(5), 2004a.

[2] A. Moshaiov, G. Avigad, and N. Brauner, "Multi-objective path planning by the concept-based IEC method," *Proceedings of the 2004 IEEE Int. Conference on Computational Cybernetics, ICCC 2004*. Vienna, Austria.

[3] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. J. Wiley & Sons, Ltd., 2001.

[4] D. Van Veldhuizen, and G. B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evolutionary Computation*, 8(2), 2000, pp. 125-147.

[5] C.A.C. Coello, "A comprehensive survey of evolutionary-based Multiobjective optimization techniques, Knowledge and Information techniques." *Knowladge and Information Systems*, 1(3), 1999, pp. 269-308.

[6] J. Andersson, "Multiobjective optimisation in engineering design applications to fluid power systems," *Thesis submitted to Linkoping University*, Linkoping.2001.

[7] G. Avigad, A. Moshaiov, and N.Brauner, "MOEA-based Approach to Delayed Decisions for Robust Conceptual Design," *In Applications of Evolutionary Computation, Lecture Notes in Computer Science*, LCNS 3449, pp: 584-589, Springer, 2005.

[8] N. Srinivas, and K. Deb, "Multi-Objective Function Optimization using Non-Dominated Sorting Genetic Algorithms," *Evolutionary Computation* 2(3), pp: 221- 248, 1994.

[9] C.M. Fonseca and P.J. Fleming, "Genetic algorithms for multi-objective Optimization: formulation, discussion and generalization." *Proc Fifth Int Conf Genetic Algorithms*; 416–23, 1993.