SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 7 No 4
ISSN 2054-7390

A Population-Based Multicriteria Algorithm for Alternative Generation

[1]Julian Scott Yeomans
[1] OMIS Area, Schulich School of Business, York University, Toronto, ON, M3J 1P3 Canada;
syeomans@schulich.yorku.ca

ABSTRACT

Complex problems are frequently overwhelmed by inconsistent performance requirements and incompatible specifications that can be difficult to identify at the time of problem formulation. Consequently, it is often beneficial to construct a set of different options that provide distinct approaches to the problem. These alternatives need to be close-to-optimal with respect to the specified objective(s), but be maximally different from each other in the solution domain. The approach for creating maximally different solution sets is referred to as modelling-to-generate-alternatives (MGA). This paper introduces a computationally efficient, population-based multicriteria MGA algorithm for generating sets of maximally different alternatives.

Keywords: Multicriteria Objectives, Population-based algorithms, Modelling-to-generate-alternatives.

1 Introduction

Complex problems frequently contain inconsistent and incompatible design specifications that can be difficult to incorporate into supporting mathematical formulations [1]-[5]. While "optimal" solutions can be calculated for the mathematical models, they may not provide a truly best solution to the "real" problem as there are usually unmodeled components not apparent when the initial mathematical models are formulated [1][2][6]. Generally, it is better to construct a small number of dissimilar alternatives that provide distinct viewpoints for the particular problem [3][7]. These distinct solutions should be close-to-optimal with respect to the specified objective(s), but be maximally different from each other within the solution domain. Numerous approaches collectively referred to as *modelling-to-generate-alternatives* (MGA) have been proposed to satisfy this multi-solution requirement [6]-[8]. The principal motivation behind MGA is to construct a set of options that are "good" with respect to all specified objective(s), but are fundamentally different from each other in the decision space. Decision-makers have to conduct a subsequent assessment of this set of alternatives to determine which specific alternative(s) most closely achieve their specific requirements. Consequently, MGA approaches are considered as decision support methods rather than as solution determination processes as assumed in explicit optimization.

The initial MGA algorithms used straightforward, iterative methods for constructing alternatives by incrementally re-solving their procedures whenever a new solution needed to be generated [6]-[10]. These iterative approaches imitated the seminal MGA method of Brill *et al*. [8] where, after the initial mathematical model had been optimized, all supplementary alternatives were produced one-at-a-time.

DOI: 10.14738/tmlai.74.6733
Publication Date: 14th July, 2019
URL: http://dx.doi.org/10.14738/tmlai.74.6733

Consequently, these incremental approaches all required *n*+1 iterations of their respective algorithms – initially to optimize the original problem, then to produce each of the subsequent *n* alternatives [7][11]-[18].

Subsequently, it was demonstrated how a set of maximally different alternatives could be generated using *any* population-based algorithm by permitting the generation of the overall optimal solution together with *n* distinct alternatives in a single computational run irrespective of the value of *n* [19]-[23]. In [24] a new data structure was created that permits alternatives to be *simultaneously* constructed by population-based solution methods and this was incorporated into a dual-criterion procedure in [25].

In this paper, a multicriteria, objective is introduced that combines the data structure into a simultaneous solution approach to create a new stochastic MGA algorithm. The max-sum components of the objective produce a maximum distance between alternatives by ensuring that the total deviation between all of the variables in all of the alternatives is collectively large. This does not, however, preclude the occurrence of relatively small (or zero) deviations between certain individual variables within certain solutions. In contrast, max-min objectives force a maximum distance between every variable over all solutions. By considering the multiple objectives simultaneously, the alternatives created can be forced as far apart as possible for all variables in general and the closest distance in the worst case between any solutions will never be less than the value obtained for the max-min objective. This stochastic MGA algorithmic approach proves to be extremely computationally efficient.

2 Modelling to Generate Alternatives

Mathematical optimization focuses almost exclusively on producing single optimal solutions to single-objective problems or, equivalently, constructing sets of noninferior solutions to multi-objective formulations [2][5][8]. While these conventions create solutions to the mathematical formulations as derived, whether the outputs provide "best" solutions to the original "real" problems remains less convincing [1][2][6][8]. Most "real world" decision situations possess numerous system conditions that can never be completely accounted for in mathematical constructions [1], [5]. In addition, it may not be possible to account for all of the subjective requirements as there are frequently numerous adversarial stakeholders and incompatible components to incorporate. Most subjective aspects remain unquantified and unmodelled in the mathematical formulations. This frequently happens when conclusions must be based not only on modelled objectives, but also upon more incongruent stakeholder predilections and socio-political-economic aspects [7]. Several "real life" instances of these idiosyncratic modelling features are described in [6][8]-[10].

When potentially unaccounted objectives and unmodelled components exist, non-traditional techniques are needed to scour the decision region for not only noninferior sets of alternatives, but also for solutions that are clearly *sub-optimal* to the modelled problem. Specifically, any search for alternatives to problems known or suspected to contain unmodelled components must concentrate not only on non-inferior sets of solutions, but also necessarily on explicit explorations of the problem's inferior solution domain.

To illustrate the consequences of unmodelled objectives on a decision search, assume that the optimal solution for a maximization problem is $\boldsymbol{X}^*$ with objective value $Z1^*$ [24]. Suppose a second, unquantified, maximization objective $Z2$ exists that represents some "politically acceptable" factor. Assume that the solution, $\boldsymbol{X}^a$, belonging to the 2-objective noninferior set, exists that corresponds to a best compromise solution if both objectives could have been simultaneously considered. Although $\boldsymbol{X}^a$ would be considered as the best solution to the real problem, in the actual mathematical model it would appear inferior to solution ***X****, since $Z1^a \leq Z1^*$. Therefore, when unquantified components are included in the decision-making process, inferior decisions to the mathematically modelled problem could be optimal to the underlying “real” problem. Thus, when unquantified issues and unmodelled objectives could exist, alternative solution procedures are required to not only explore the solution domain for noninferior solutions to the modelled problem, but also to concurrently search the solution domain for inferior solutions. Population-based algorithms prove to be proficient solution methods for concurrent searches throughout a decision space.

The prime directive for MGA is the construction of a practicable set of options that are quantifiably good when evaluated with respect to the modelled objectives, yet remain as different as possible from each other within the decision space. By achieving this task, the resultant set of alternatives can supply quite different perspectives with respect to the modelled objective(s) yet very differently with respect to the potentially unmodelled aspects. By creating these good-but-different options, the decision-makers can then identify specific desirable qualities within the alternatives that might satisfy the unmodelled objectives to varying degrees of stakeholder tolerability.

To motivate the MGA process, it is necessary to more formally characterize the mathematical definition of its goals [6][7]. Assume that the optimal solution to the original mathematical formulation is ***X**** producing a corresponding objective value of $\boldsymbol{Z}^* = F(\boldsymbol{X}^*)$. The resulting model can then be solved to produce an alternative solution, ***X***, that is maximally different from ***X****:

$$\text{Maximize} \quad \Delta(\boldsymbol{X}, \boldsymbol{X}^*) = \sum_i (X_i - X_i^*)^2 \qquad (1)$$

$$\text{Subject to:} \quad \boldsymbol{X} \in D \qquad (2)$$

$$| F(\boldsymbol{X}) - \boldsymbol{Z}^* | \leq T \qquad (3)$$

where Δ represents an appropriate difference function (shown in (1) as an absolute difference) and *T* is a tolerance target relative to the original optimal objective value ***Z****. *T* is a user-specified limit that determines what proportion of the inferior region needs to be explored for acceptable alternatives. This difference function concept can be extended into a difference measure between any *set of alternatives* by replacing ***X**** in the objective of the maximal difference model and calculating the overall minimum absolute difference (or some other function) of the pairwise comparisons between corresponding variables in each pair of alternatives – subject to the condition that each alternative is feasible and falls within the specified tolerance constraint.

The population-based multicriteria MGA procedure to be introduced is designed to generate a pre-determined small number of close-to-optimal, but maximally different alternatives, by adjusting the value of *T* and solving the corresponding maximal difference problem instance by exploiting the population structure of the algorithm. The survival of solutions depends upon how well the solutions perform with respect to the problem’s originally modelled objective(s) and simultaneously by how far away they are from all of the other alternatives generated in the decision space.

3 Population-based, Multicriteria MGA Algorithm

In this section, a data structure is employed that enables a multicriteria MGA solution approach via any population-based algorithm [24]. Suppose that it is desired to produce *P* alternatives that each possess *n* decision variables and that the population algorithm is to possess K solutions in total. That is, each solution contains one set of P maximally different alternatives. Let $\boldsymbol{Y}_k$, k = 1,…, K, represent the k^{th} solution in the population which is comprised of one complete set of P different alternatives. Namely, if $\boldsymbol{X}_{kp}$ is the p^{th} alternative, p = 1,…, P, of solution k, k = 1,…, K, then $\boldsymbol{Y}_k$ can be represented as

$$\boldsymbol{Y}_k = [\boldsymbol{X}_{k1}, \boldsymbol{X}_{k2},\ldots, \boldsymbol{X}_{kP}] \ . \tag{4}$$

If X_{kjq}, q = 1,…, n, is the q^{th} variable in the j^{th} alternative of solution k, then

$$\boldsymbol{X}_{kj} = (X_{kj1}, X_{kj2},\ldots, X_{kjn}) \ . \tag{5}$$

Consequently, the entire population, $\boldsymbol{Y}$, consisting of K different sets of P alternatives can be expressed in vectorized form as,

$$\boldsymbol{Y'} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2,\ldots, \boldsymbol{Y}_K] \ . \tag{6}$$

The following population-based MGA method produces a pre-determined number of close-to-optimal, but maximally different alternatives, by modifying the value of the bound T in the maximal difference model and using any population-based method to solve the corresponding, maximal difference problem. The multicriteria MGA algorithm that follows constructs a pre-determined number of maximally different, close-to-optimal alternatives, by modifying the bound value T in the maximal difference model and using any population-based technique to solve the corresponding maximal difference problem. Each solution in the population comprises one set of p different alternatives. By exploiting the co-evolutionary aspects of the algorithm, the algorithm evolves each solution toward sets of dissimilar local optima within the solution domain. In this processing, each solution alternative mutually experiences the search steps of the algorithm. Solution survival depends upon both how well the solutions perform with respect to the modelled objective(s) and by how far apart they are from every other alternative in the decision space.

A straightforward process for generating alternatives solves the maximum difference model iteratively by incrementally updating the target T whenever a new alternative needs to be produced and then re-solving the resulting model [24]. This iterative approach parallels the original Hop, Skip, and Jump (HSJ) MGA algorithm of Brill *et al*. [8] in which the alternatives are created one-by-one through an incremental adjustment of the target constraint. While this approach is straightforward, it entails a repetitive execution of the optimization algorithm [7][12][13]. To improve upon the stepwise HSJ approach, a concurrent MGA technique was subsequently designed based upon co-evolution [13][15][17]. In a co-evolutionary approach, pre-specified stratified subpopulation ranges within an algorithm's overall population are established that collectively evolve the search toward the specified number of maximally different alternatives. Each desired solution alternative is represented by each respective subpopulation and each subpopulation undergoes the common processing operations of the procedure. The survival of solutions in each subpopulation depends simultaneously upon how well the solutions perform with respect to the modelled objective(s) and by how far away they are from all of the other alternatives. Consequently, the evolution of solutions in each subpopulation toward local optima is directly influenced by those solutions contained in all of the other subpopulations, which forces the concurrent co-evolution of each subpopulation towards good but maximally distant regions within the decision space according to the maximal difference model [7]. Co-evolution is also much more efficient than a sequential HSJ-style approach in that it exploits the inherent population-based searches to concurrently generate the entire set of maximally different solutions using only a single population [12][17].

While concurrent approaches can exploit population-based algorithms, co-evolution can only occur within each of the stratified subpopulations. Consequently, the maximal differences between solutions in different subpopulations can only be based upon aggregate subpopulation measures. Conversely, in the following simultaneous MGA algorithm, each solution in the population contains exactly one entire set of alternatives and the maximal difference is calculated only for that particular solution (i.e. the specific alternative set contained within that solution in the population). Hence, by the evolutionary nature of the population-based search procedure, in the subsequent approach, the maximal difference is simultaneously calculated for the specific set of alternatives considered within each specific solution – and the need for concurrent subpopulation aggregation measures is avoided.

Using the data structure terminology, the steps for the multicriteria MGA algorithm are as follows [14][19]-[24]. It should be readily apparent that the stratification approach employed by this method can be easily modified for any population-based algorithm.

Initialization Step. Solve the original optimization problem to find its optimal solution, $\boldsymbol{X^*}$. Based upon the objective value $F(\boldsymbol{X^*})$, establish P target values. P represents the desired number of maximally different alternatives to be generated within prescribed target deviations from the $\boldsymbol{X^*}$. Note: The value for P has to have been set *a priori* by the decision-maker.

Without loss of generality, it is possible to forego this step and to use the algorithm to find $\boldsymbol{X^*}$ as part of its solution processing in the subsequent steps. However, this significantly increases the number of iterations of the computational procedure and the initial stages of the processing become devoted to finding $\boldsymbol{X^*}$ while the other elements of each population solution are retained as essentially “computational overhead”.

Step 1. Create an initial population of size K where each solution contains P equally-sized partitions. The partition size corresponds to the number of decision variables in the original optimization problem. $\boldsymbol{X_{kp}}$ represents the p^{th} alternative, $p = 1,…,P$, in solution $\boldsymbol{Y_k}$, $k = 1,…,K$.

Step 2. In each of the K solutions, evaluate each $\boldsymbol{X_{kp}}$, $p = 1,…,P$, with respect to the modelled objective. Alternatives meeting their target constraint and all other problem constraints are designated as *feasible*, while all other alternatives are designated as *infeasible*.

Note: A solution can be designated as feasible only if all of the alternatives contained within it are feasible.

Step 3. Apply an appropriate elitism operator to each solution to rank order the best individuals in the population. The best solution is the feasible solution containing the most distant set of alternatives in the decision space (the distance measures are defined in Step 5).

Note: Because the best-solution-to-date is always retained in the population throughout each iteration, at least one solution will always be feasible. Furthermore, a feasible solution based on the initialization step can be constructed using P repetitions of $\boldsymbol{X^*}$.

Step 4. Stop the algorithm if the termination criteria (such as maximum number of iterations or some measure of solution convergence) are met. Otherwise, proceed to Step 5.

Step 5. For each solution $\boldsymbol{Y_k}$, $k = 1,…, K$, calculate R Max-Min and/or Max-Sum distance measures, $D^r_{k,}$ $r = 1,…, R$, between all of the alternatives contained within the solution.

As an illustrative example for calculating the multicriteria distance measures, compute:

$$D^1_k = \Delta^1 (\boldsymbol{X}_{ka}, \boldsymbol{X}_{kb}) = \min_{a,b,q} \mid X_{kaq} - X_{kbq} \mid , \qquad a = 1,\ldots,P,\ b = 1,\ldots,P,\ q = 1,\ldots,n, \tag{7}$$

$$D^2_k = \Delta^2 (\boldsymbol{X}_{ka}, \boldsymbol{X}_{kb}) = \sum_{a=1 to P} \sum_{b=1 to P} \sum_{q=1\ldots n} \mid X_{kaq} - X_{kbq} \mid \tag{8}$$

and

$$D^3_k = \Delta^3 (\boldsymbol{X}_{ka}, \boldsymbol{X}_{kb}) = \sum_{a=1 to P} \sum_{b=1 to P} \sum_{q=1\ldots n} (X_{kaq} - X_{kbq})^2. \tag{9}$$

D^1_k denotes the minimum absolute distance, D^2_k represents the overall absolute deviation, and D^3_k determines the overall quadratic deviation between all of the alternatives contained within solution k.

Alternatively, the distance functions could be calculated using some other appropriately defined measures.

Step 6. Let $D_k = G(D^1_k, D^2_k, D^3_k, \ldots, D^R_k)$ represent the multicriteria objective for solution k. Rank the solutions according to the distance measure D_k objective – appropriately adjusted to incorporate any constraint violation penalties for infeasible solutions. The goal of maximal difference is to force alternatives to be as far apart as possible in the decision space from the alternatives of each of the partitions within each solution This step orders the specific solutions by those solutions which contain the set of alternatives which are most distant from each other.

Step 7. Apply applicable algorithmic "change operations" to each solution within the population and return to Step 2.

4 Conclusion

Complex problem solving inherently involves incongruent features and indeterminate performance specifications. These situations commonly possess inconsistent structural components that are difficult to incorporate into supporting decision systems. There are always unmodelled features, not apparent during model formulation, that can significantly impact the adequacy of its solutions. These components force decision-makers to combine uncertainties into their solution process prior to any problem resolution. When faced with these inconsistencies, the likelihood that any single solution can concurrently satisfy all of the ambiguous system requirements to "optimum" is quite low. Therefore, any decision support approach must somehow address these complicating aspects in some way, while simultaneously being flexible enough to include the intrinsic planning uncertainties.

This paper has provided a new multicriteria approach and an updated MGA procedure. This new computationally efficient MGA method establishes how population-based algorithms can simultaneously construct entire sets of close-to-optimal, maximally different alternatives by exploiting the evolutionary characteristics of any population-based solution approach. In this MGA role, the multicriteria objective can efficiently generate the requisite set of dissimilar alternatives, with each generated solution providing an entirely different outlook to the problem. The max-sum criteria ensures that the distances between the alternatives created by this algorithm are good in general, while the max-min criteria ensures that the distances between the alternatives are good in the worst case. The value of an absolute-type function delivers a physical interpretation to its measure of distance. Since population-based procedures can be applied to a wide spectrum of problem types, the practicality of the multicriteria algorithm can be extended to many "real world" applications. These extensions will be considered in future studies.

REFERENCES

[1]. Brugnach, M., A. Tagg, F. Keil, and W.J. De Lange, *Uncertainty matters: computer models at the science-policy interface*. Water Resources Management, 2007. 21: p. 1075-1090.

[2]. Janssen, J.A.E.B., M.S. Krol, R.M.J. Schielen, and A.Y Hoekstra, *The effect of modelling quantified expert knowledge and uncertainty information on model based decision making*. Environmental Science and Policy, 2010. 13(3): p. 229-238.

[3]. Matthies, M., C. Giupponi, and B. Ostendorf, *Environmental decision support systems: Current issues, methods and tools*. Environmental Modelling and Software, 2007. *22(2)*: p. 123-127.

[4]. Mowrer, H.T., *Uncertainty in natural resource decision support systems: Sources, interpretation, and importance*. Computers and Electronics in Agriculture, 2000. *27(1-3)*: p. 139-154.

[5]. Walker, W.E., P. Harremoes, J. Rotmans, J.P. Van der Sluis, M.B.A. Van Asselt, P. Janssen, and M.P. Krayer von Krauss, *Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support*. Integrated Assessment, 2003. 4(1): p. 5-17.

[6]. Loughlin, D.H., S.R. Ranjithan, E.D. Brill, and J.W. Baugh, *Genetic algorithm approaches for addressing unmodelled objectives in optimization problems*. Engineering Optimization, 2001. *33(5)*: p. 549-569.

[7]. Yeomans, J.S., and Y Gunalay, *Simulation-Optimization Techniques for Modelling to Generate Alternatives in Waste Management Planning*. Journal of Applied Operational Research, 2011. 3(1): p. 23-35.

[8]. Brill, E.D., S.Y. Chang, and L.D Hopkins, *Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning*. Management Science. 1982. 28(3): p. 221-235.

[9]. Baugh, J.W., S.C. Caldwell, and E.D Brill, *A Mathematical Programming Approach for Generating Alternatives in Discrete Structural Optimization*. Engineering Optimization. 1997, 28(1): p. 1-31.

[10]. Zechman, E.M., and S.R. Ranjithan, *An Evolutionary Algorithm to Generate Alternatives (EAGA) for Engineering Optimization Problems*. Engineering Optimization. 2004, 36(5): p. 539-553.

[11]. Gunalay, Y., J.S. Yeomans, and G.H. Huang, *Modelling to generate alternative policies in highly uncertain environments: An application to municipal solid waste management planning*. Journal of Environmental Informatics, 2012. *19(2)*: p. 58-69.

[12]. Imanirad, R., and J.S. Yeomans, *Modelling to Generate Alternatives Using Biologically Inspired Algorithms*. in *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*, X.S. Yang, Editor 2013. Amsterdam: Elsevier (Netherlands). p. 313-333.

[13]. Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Computationally Efficient, Biologically-Inspired Modelling-to-Generate-Alternatives Method*. Journal on Computing. 2012, 2(2): p. 43-47.

[14]. Yeomans, J.S., *An Efficient Computational Procedure for Simultaneously Generating Alternatives to an Optimal Solution Using the Firefly Algorithm*, in *Nature-Inspired Algorithms and Applied Optimization*, Yang, X.S. Editor 2018. Heidelberg (Springer), Germany. p. 261-273.

[15]. Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Co-evolutionary, Nature-Inspired Algorithm for the Concurrent Generation of Alternatives*. Journal on Computing. 2012, 2(3): p. 101-106.

[16]. Imanirad, R., X.S. Yang, and J.S. Yeomans, *Modelling-to-Generate-Alternatives Via the Firefly Algorithm*. Journal of Applied Operational Research. 2013. 5(1): p. 14-21.

[17]. Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Concurrent Modelling to Generate Alternatives Approach Using the Firefly Algorithm*. International Journal of Decision Support System Technology. 2013, 5(2): p. 33-45.

[18]. Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Biologically-Inspired Metaheuristic Procedure for Modelling-to-Generate-Alternatives*. International Journal of Engineering Research and Applications. 2013, 3(2): p. 1677-1686.

[19]. Yeomans, J.S., *Simultaneous Computing of Sets of Maximally Different Alternatives to Optimal Solutions*. International Journal of Engineering Research and Applications, 2017. *7(9)*: p. 21-28.

[20]. Yeomans, J.S., *An Optimization Algorithm that Simultaneously Calculates Maximally Different Alternatives*. International Journal of Computational Engineering Research, 2017. *7(10)*: p. 45-50.

[21]. Yeomans, J.S., *Computationally Testing the Efficacy of a Modelling-to-Generate-Alternatives Procedure for Simultaneously Creating Solutions*. Journal of Computer Engineering, 2018. *20(1)*: p. 38-45.

[22]. Yeomans, J.S., *A Computational Algorithm for Creating Alternatives to Optimal Solutions*. Transactions on Machine Learning and Artificial Intelligence, 2017. 5(5): p. 58-68.

[23]. Yeomans, J.S., *A Simultaneous Modelling-to-Generate-Alternatives Procedure Employing the Firefly Algorithm*, in *Technological Innovations in Knowledge Management and Decision Support*, Dey, N. Editor, 2019. Hershey, Pennsylvania (IGI Global), USA. p. 19-33.

[24]. Yeomans, J.S., *An Algorithm for Generating Sets of Maximally Different Alternatives Using Population-Based Metaheuristic Procedures*. Transactions on Machine Learning and Artificial Intelligence, 2018. 6(5): p. 1-9.

[25]. Yeomans, J.S., *A Bicriterion Approach for Generating Alternatives Using Population-Based Algorithms*. WSEAS Transactions on Systems, 2019. 18(4): p. 29-34.