Multicriteria Generation of Alternatives for Engineering Optimization Problems Using Population-Based Metaheuristics: A Computational Test

YAVUZ GUNALAY
Faculty of Economic and Administrative Sciences
Bahcesehir University
34353 Besiktas Istanbul, Turkey
TURKEY
yavuz.gunalay@bahcesehir.edu.tr

JULIAN SCOTT YEOMANS
OMIS Area, Schulich School of Business
York University
4700 Keele Street, Toronto, ON, M3J 1P3
CANADA
syeomans@schulich.yorku.ca

Abstract: - Engineering optimization problems can be dominated by inconsistent performance requirements and incompatible specifications that can be difficult to detect when supporting mathematical programming models are formulated. Thus, it often proves advantageous to construct a set of options that provide dissimilar approaches to such problems. These alternatives should satisfy the required performance criteria, but be maximally different from each other in their decision spaces. The method for constructing maximally different sets of solutions is referred to as modelling-to-generate-alternatives (MGA). This paper considers a multicriteria method that can generate sets of maximally different alternatives using any population-based solution algorithm. This MGA approach is both computationally efficient and simultaneously produces the prescribed number of maximally different solution alternatives in a single computational run of the procedure. The computational efficacy of this multicriteria MGA approach is demonstrated on two commonly-tested engineering optimization problems using the population-based Firefly Algorithm metaheuristic.

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

1 Introduction

Complex engineering optimization problems frequently contain inconsistent and incompatible design specifications that can be difficult to incorporate into mathematical decision-models [1]-[5]. While "optimal" solutions can be calculated for the mathematical formulations, whether these answers produce best outcomes for the original "real" system is far less certain [1]-[6]. To improve decision-making under such ambiguities, it is often preferable to construct a limited number of dissimilar options that provide very different perspectives [3][7]. To be beneficial, these distinct options should be close-to-optimal with respect to all mathematically modelled objective(s), but be *maximally different* from each other within the solution space [6]-[8]. Numerous methods collectively referred to as *modelling-to-generate-alternatives* (MGA) have been created to achieve this multi-solution requirement [6]-[8]. The primary motivation behind MGA is to construct a set of alternatives that are all "good" with respect to the specified objective(s), but fundamentally distinct from each other in the decision space. Decision-makers must conduct a subsequent assessment of the alternatives to ascertain which specific option(s) most closely satisfies their underlying circumstances. Consequently, MGA is generally considered a decision support method rather than as an explicit solution creation process assumed in "normal" mathematical optimization.

The earliest MGA approaches employed a straightforward process in which each alternative was incrementally formulated by re-running the solution generation algorithm whenever a new option had to be produced [6]-[10]. These iterative procedures mimicked the seminal approach in [8] where, once the initial mathematical model had been

optimized, all supplementary alternatives were produced one-at-a-time. These incremental approaches all required n+1 computational iterations – 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 efficiently 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 [23] a novel data structure was introduced that permits alternatives to be constructed *simultaneously* by population-based solution techniques.

In this paper, it is demonstrated how a set of maximally different engineering design options can be produced by extending several earlier optimization techniques [12]-[18]. A multicriteria, max-sum, max-min MGA process provides a method that can be deployed using *any* population-based solution algorithm. This approach advances earlier procedures [13][15]-[18] by permitting the simultaneous generation of n distinct alternatives in a single computational run. Namely, to construct the requisite n maximally different design options, the algorithm has to run only once irrespective of the value of n [19]-[23]. The multicriteria objective combines the data structure into the simultaneous solution approach in the 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. The computational efficacy of this approach for creating alternatives is demonstrated by applying the multicriteria MGA procedure to two highly non-linear, engineering optimization benchmark problems using the population-based Firefly Algorithm (FA) metaheuristic.

2 Modelling to Generate Alternatives

Mathematical programming has focused almost exclusively on finding single optimal solutions to single-objective problems or, equivalently, producing noninferior solutions to multi-objective formulations [2][5][8]. While these approaches may solve the formulations as constructed mathematically, whether these solutions are truly "best" for the original "real world" applications remains less certain [1][2][6][8]. In most "real world" systems, there are countless system specifications that can never be incorporated into the mathematical problem formulation [1][5]. Unavoidably, the majority of the subjective aspects remain unmodelled and unquantified in the mathematical system formulations. This frequently occurs when final outcomes are decided upon based not only on modelled objectives, but also on more subjective socio-political-economic preferences and stakeholder goals [7]. When unmodelled components are suspected to exist, non-traditional solution approaches are needed for searching the decision space not only for noninferior solutions, but also for *sub-optimal* possibilities. Specifically, any search for alternatives to problems suspected to possess unmodelled components must concentrate not only on a non-inferior set of solutions, but also necessarily on an explicit exploration of the problem's *inferior* solution space. Numerous "real life" instances of these types of modelling situations are illustrated in [6][8]-[10].

To demonstrate the impact of unmodelled objectives on a solution search, assume that the optimal solution to a maximization problem is $\boldsymbol{X}^*$ with objective value $Z1^*$ [24]. Suppose a second, unmodelled, maximization objective $Z2$ exists that represents some "politically acceptable" feature. 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 actually have been simultaneously considered. While $\boldsymbol{X}^{a}$ would be the best solution to the real problem, in the actual mathematical formulation it would seem inferior to solution $\boldsymbol{X}^*$, since $Z1^{a} \leq Z1^*$. Thus, when unmodelled components are included in the decision-making process, inferior decisions to the mathematically modelled system could actually be optimal to the fundamental "real" problem. If unmodelled aspects and unquantified objectives might exist, alternative solution procedures are essential to not only explore the decision region for noninferior solutions to the modelled problem, but also to concurrently search the decision space for explicitly inferior solutions.

Necessarily, then, in these situations, the aim is to create a workable set of options that are quantifiably good with respect to the modelled objectives yet are as different as possible from each other within the solution space. By satisfying this maximal difference condition, the resulting set of alternatives is able to supply truly different perspectives that all perform similarly with respect to the known modelled objective(s) yet very differently with respect to various potentially unmodelled aspects. By creating good-but-different options, the system designers are the able to consider potentially desirable qualities within the alternatives that might be able to satisfy the unmodelled objectives to varying degrees of stakeholder acceptability.

To motivate the process, it is necessary to formally characterize the mathematical definition of maximal difference [6][7]. Assume that the optimal solution to an original mathematical programming formulation is $\boldsymbol{X}^*$ with corresponding objective value $\boldsymbol{Z}^* = F(\boldsymbol{X}^*)$. An ensuing difference model can then be solved to produce an alternative solution, $\boldsymbol{X}$, that is maximally different from $\boldsymbol{X}^*$:

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

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

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

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 $\boldsymbol{Z}^*$. T is a user-specified limit that determines what proportion of the inferior region ought to be explored for acceptable alternatives. This difference function concept can be extended into a difference measure between any *set of alternatives* by replacing $\boldsymbol{X}^*$ 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 procedure that is subsequently employed is designed to generate a fixed, pre-determined 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 structures of the optimization 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 Multicriteria MGA Procedure

In this section, a data structure is employed that permits a multicriteria MGA approach to be used for creating system options using any population-based solution algorithm [24]-[28]. Suppose that it is desired to be able to produce P alternatives that each possess n decision variables and that the population algorithm is to possess K solutions in total. Namely, each solution in the population contains one complete set of P maximally different alternatives. Let $\boldsymbol{Y}_k$, $k = 1,\ldots, K$, represent the k^{th} solution consisting of one complete set of P different alternatives. Specifically, if $\boldsymbol{X}_{kp}$ corresponds to the p^{th} alternative, $p = 1,\ldots, P$, of solution k, $k = 1,\ldots, K$, then $\boldsymbol{Y}_k$ can be represented as

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

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

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

Accordingly, the entire population, $\boldsymbol{Y}$, comprised of K different sets of P alternatives can be expressed in vectorized format as,

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

The multicriteria method that follows can produce a pre-determined number of close-to-optimal, maximally different system options, by modifying the value of T in the maximal difference model and using any population-based optimization algorithm to solve the corresponding, maximal difference problem. Each solution in the population is composed of one complete set of P different possible system options. By exploiting the co-evolutionary aspects of the algorithm, the procedure evolves each solution (i.e. set of alternatives) 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 both upon 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 seminal Hop, Skip, and Jump (HSJ) approach [8] in which the alternatives are constructed one-at-a-time through an incremental adjustment of the target constraint. Although the HSJ is straightforward, it necessitates

a repetitive execution of the optimization algorithm [7][12][13]. To improve upon the stepwise HSJ approach, a concurrent 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 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 more computationally efficient than the sequential HSJ-style approach by exploiting inherent population-based searches to concurrently generate the entire set of maximally different solutions using only a single population [12][17].

While concurrent approaches possess the ability to 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 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 calculated *simultaneously* for the specific set of alternatives considered within each specific solution – and the need for concurrent subpopulation aggregation measures is circumvented.

Using the data structure terminology, the steps for the multicriteria MGA algorithm are as follows [14][19]- [28]. The stratification approach employed by this method can be easily modified for solution via any population-based optimization algorithm.

Preliminary 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 must be fixed *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 is divided into 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, $p = 1,\ldots,P$, in solution $\boldsymbol{Y}_k$, $k = 1,\ldots,K$.

Step 2. In each of the K solutions, evaluate each $\boldsymbol{X}_{kp}$, $p = 1,\ldots,P$, with respect to the modelled objective. Alternatives meeting both their target constraint and all the other problem constraints are designated as *feasible*, while all other alternatives are designated as *infeasible*. An individual solution can only be designated as feasible 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 remain 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 $\boldsymbol{Y}_k$, $k = 1,\ldots, K$, calculate R Max-Min and/or Max-Sum distance measures, D^r_k, $r = 1,\ldots, R$, between all of the alternatives contained within the solution.

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

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

$$D^2_k = \Delta^2(\boldsymbol{X}_{ka}, \boldsymbol{X}_{kb})$$

$$= \sum_{a=1toP} \sum_{b=1toP} \sum_{q=1...n} |X_{kaq} - X_{kbq}|. \quad (8)$$

and

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

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, distance function could be calculated using 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 appropriate metaheuristic "change operations" to each solution within the population and return to Step 2.

4 Firefly Algorithm Optimization

While this section provides a brief synopsis of the steps involved in the FA process, more extensive details can be found in [29]. The FA is a nature-inspired, population-based metaheuristic that employs the following three idealized rules: (i) All fireflies within a population are unisex, so that one firefly will be attracted to other fireflies irrespective of their sex; (ii) Attractiveness between fireflies is proportional to their brightness, implying that for any two flashing fireflies, the less bright one will move towards the brighter one. Attractiveness and brightness both decrease as the distance between fireflies increases. If there is no brighter firefly within its visible vicinity, then a particular firefly will move randomly; and (iii) The brightness of a firefly is determined by the landscape of the objective function. Namely, for a maximization problem, the brightness can simply be considered proportional to the value of the objective function.

In the FA, there are two important issues to resolve: the variation of light intensity and the formulation of attractiveness. For simplicity, it can always be assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function. In the simplest case, the brightness of a firefly at a particular location $\boldsymbol{X}$ would be its calculated objective value $F(\boldsymbol{X})$. However, the attractiveness, β, between fireflies is relative and will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so the attractiveness should be allowed to vary with the degree of absorption. Consequently, the overall attractiveness of a firefly can be defined as

$$\beta = \beta_0 \exp(-\gamma r^2) \quad (10)$$

where β_0 is the attractiveness at distance $r = 0$ and γ is the fixed light absorption coefficient for a specific medium. If the distance r_{ij} between any two fireflies i and j located at $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, respectively, is calculated using the Euclidean norm, then the movement of a firefly i that is attracted to another more attractive (i.e. brighter) firefly j is determined by

$$\boldsymbol{X}_i = \boldsymbol{X}_i + \beta_0 \exp(-\gamma(r_{ij})^2)(\boldsymbol{X}_i - \boldsymbol{X}_j) + \alpha\boldsymbol{\varepsilon}_i . \quad (11)$$

In this expression of movement, the second term is due to the relative attraction and the third term is a randomization component. Yang [29] indicates that α is a randomization parameter normally selected within the range [0,1] and $\boldsymbol{\varepsilon}_i$ is a vector of random numbers drawn from either a Gaussian or uniform (generally [-0.5,0.5]) distribution. It should be pointed out that this expression is a random walk biased toward brighter fireflies and if $\beta_0 = 0$, it becomes a simple random walk. The parameter γ characterizes the variation of the attractiveness and its value determines the speed of the algorithm's convergence. For most applications, γ is typically set between 0.1 to 10 [29]. In any given optimization problem, for a very large number of fireflies $n >> k$ where k is the number of local optima, the initial locations of the n fireflies should be distributed relatively uniformly throughout the entire search space. As the FA proceeds, the fireflies would converge into all of these local optima (including the global ones). By comparing the best solutions among all these optima, the global optima can easily be determined. Yang [29] demonstrated that the FA will approach the global optima when n $\rightarrow \infty$ and the number of iterations t, is set so that $t >> 1$. In reality, the FA has been found to converge extremely quickly.

Two important limiting or asymptotic cases occur when $\gamma \rightarrow 0$ and when $\gamma \rightarrow \infty$. For $\gamma \rightarrow 0$, the attractiveness is constant $\beta = \beta_0$, which is equivalent to having a light intensity that does not decrease.

Thus, a firefly would be visible anywhere within the solution domain. Hence, a single (usually global) optima can easily be reached. If $\boldsymbol{X}_j$ is replaced by the current global best $\boldsymbol{G^*}$, then this implies that the FA becomes a special case of the accelerated particle swarm optimization (PSO) algorithm. Subsequently, the computational efficiency of this special case of the FA is equivalent to that of enhanced PSO. Conversely, when $\gamma \rightarrow \infty$, the attractiveness is essentially zero in the sightline of other fireflies. This is equivalent to the case where the fireflies randomly roam throughout a very thick foggy region. No other fireflies are visible and each firefly roams in a completely random fashion. This case corresponds to a completely random search method. As the FA operates between these two extremes, it is possible to adjust the parameters α and γ so that an FA can outperform both the random search and the enhanced PSO algorithms. Furthermore, the FA can find both the global optima as well as the local optima concurrently which holds huge computational and efficiency advantages for MGA purposes [7][16][17]. Another additional advantage of the FA for MGA implementation is that different fireflies essentially work independently of each other and FA are thus better than genetic algorithms and PSO for MGA because the fireflies can aggregate more closely around each local optimum [16][17].

An FA-based MGA procedure is designed to generate a small number of good but maximally different alternatives by adjusting the value of T and using the FA to solve the corresponding, new maximal difference problem instance [16][17]. In this approach, subpopulations within the algorithm's overall population are established as the Fireflies collectively evolve toward different local optima in the solution space. Each desired solution alternative undergoes the common search procedure of the FA. As required, the survival of solutions depends 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 previously generated alternatives in the decision space.

5 Computational Testing of the MGA

As outlined earlier, in engineering optimization, decision-makers frequently prefer to be able to select from a set of "near-optimal" alternatives that significantly differ from each other in terms of the system structures characterized by their decision variables. In order to create this set of alternative planning options, a computational testing of the efficacy of employing the multicriteria MGA algorithm in conjunction with the population-based FA metaheuristic will be illustrated using (i) a well-studied, standard, constrained engineering optimization problem [30] and (ii) a 100-peak multimodal optimization problem taken from [6].

The first computational test of the MGA procedure will consider the engineering spring design benchmark problem from [30]. The design of a tension and compression spring has frequently been employed as a benchmark problem for constrained engineering optimization problems [30]. The problem contains three design variables: (i) x_1, the wire diameter, (ii) x_2, the coil diameter, and (iii) x_3, the length of the coil. The aim is to essentially minimize the weight subject to constraints on deflection, stress, surge frequency and geometry. The mathematical formulation for this test problem can be summarized as:

$$\text{Minimize } F(\boldsymbol{X}) = x_1^2 x_2 \left(2 + x_3\right) \tag{14}$$

Subject to:

$$g_1(\boldsymbol{X}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \tag{14}$$

$$g_2(\boldsymbol{X}) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_1^3 x_2 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \tag{16}$$

$$g_3(\boldsymbol{X}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \tag{17}$$

$$g_4(\boldsymbol{X}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \tag{18}$$

$$0.05 \leq x_1 \leq 2,\ 0.25 \leq x_2 \leq 1.3,\ 2 \leq x_3 \leq 15 \tag{19}$$

The optimal solution for the specific design parameters employed within this formulation is $F(\boldsymbol{X^*}) = 0.0127$ with decision variable values of $\boldsymbol{X^*} = (0.051690, 0.356750, 11.287126)$ [30]. The FA-driven multicriteria MGA algorithm was run to create the 11 maximally different solutions shown in Table 1.

Alternatives	F(*X*)	x_1	x_2	x_3
Best Found	0.0127	0.05	0.3174	14.0324
1	0.0128	0.05	0.3164	14.1754
2	0.0128	0.0514	0.3472	12.0089
3	0.0129	0.0529	0.3862	9.9684
4	0.0130	0.0521	0.3656	11.0667
5	0.0131	0.0527	0.3766	10.5179
6	0.0134	0.05	0.3157	14.978
7	0.0135	0.0524	0.3597	11.6966
8	0.0137	0.052	0.3629	12.1615
9	0.0138	0.0523	0.348	13.3247
10	0.0140	0.0535	0.3857	14.162

Table 1. Objective Values and Solutions for the 11 Maximally Different Alternatives

The second computational test will be on the highly non-linear, engineering optimization benchmark problem from [6]. The mathematical formulation for this multimodal test problem is:

Maximize $F(x, y)$ =

$$\text{Sin}(19\pi x) + \frac{x}{1.7} + \text{Sin}(19\pi y) + \frac{y}{1.7} + 2 \quad (12)$$

$$0.0 \leq x \leq 1.0,\ 0.0 \leq y \leq 1.0 \quad (13)$$

The corresponding feasible region for this problem contains 100 peaks separated by valleys with the amplitudes of both the peaks and valleys increasing as the values of the decision variables increase from their lower bounds of (0,0) toward their upper limits at (1,1). For the design parameters employed in this specific problem formulation, the mathematically optimal solution of $F(x, y) = 5.146$ occurs at point $(x, y) = (0.974, 0.974)$ [6]. The FA metaheuristic was used in the multicriteria MGA procedure to generate the 11 maximally different solutions shown in Table 2.

Alternatives	***F*(*x*,*y*)**	*x*	*y*
Best Found	5.14	0.97	0.97
1	5.10	0.98	0.97
2	5.05	0.87	0.98
3	5.00	0.76	0.98
4	4.99	0.98	0.87
5	4.91	0.98	0.76
6	4.89	0.55	0.97
7	4.89	0.98	0.55
8	4.74	0.34	0.98
9	4.69	0.98	0.24
10	4.64	0.13	0.98

Table 2. Objective Values and Solutions for the 11 Maximally Different Alternatives

The two computational examples have demonstrated how a multicriteria MGA modelling perspective can be effectively used to generate multiple, good solution alternatives by employing the very computationally efficient, population-based FA metaheuristic. By following this process, the alternatives produced all satisfy the required system criteria, yet remain as maximally different from each other as possible within the decision space. Furthermore, the multicriteria MGA procedure has simultaneously performed exceedingly well with respect to its role in function optimization. It can be noted explicitly that the overall best solutions calculated in the MGA procedure are identical to the optimal solutions determined in [6] and [30].

As described earlier, many "real world" engineering optimization applications can be riddled with inconsistent performance specifications that can be very difficult to capture and quantify. Consequently, it is frequently preferable to create several quantifiably good alternatives that concurrently provide very different perspectives to the potentially unmodelled performance design issues during the policy formulation stage. The unique performance features captured within these dissimilar alternatives can result in very different system performance with respect to the unmodelled issues, thereby incorporating the unmodelled issues into the actual solution process. The computational results from the two benchmark optimization problems demonstrate that the FA-driven multicriteria MGA algorithm provides a suitable approach for producing not only a requisite set of maximally different alternatives but also the overall single optimal solution of the problem formulation if that result, alone, is desired.

6 Conclusion

Complex engineering problem-solving inherently involves complicated performance components that can be confounded by incongruent requirements and inconsistent performance objectives. These decision environments frequently contain incompatible design specifications that are problematic – if not impossible – to incorporate when ancillary decision support models are constructed. Invariably, there are unmodelled elements, not apparent during model formulation, that can significantly affect solution adequacy. These confounding features require the decision-makers to integrate numerous discrepancies into their solution processes before a definitive solution can be determined. Faced with such inconsistencies, it is unlikely that any single solution can simultaneously satisfy all ambiguous system requirements without significant

compromises. Therefore, any decision support approach must somehow address these complicating features in some way, while simultaneously being flexible enough to condense the potential effects within the intrinsic planning incongruities.

This paper has employed a multicriteria procedure in conjunction with a population-based FA metaheuristic to direct this MGA search processes. The 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 ensure that the distances between the alternatives created by this algorithm are good in general, while the max-min criteria ensure that the distances between the alternatives are good in the worst case. The computational efficacy of employing the multicriteria MGA algorithm in conjunction with the population-based FA metaheuristic was demonstrated on two well-known, engineering optimization benchmark problems. The computational procedure not only produced sets of high-quality, maximally different solution alternatives, but also simultaneously found the optimal solutions to the two benchmark engineering problems examined. The practicality of this multicriteria MGA solution approach can clearly be extended to wide range of "real world" engineering and scientific applications. Such extensions will be examined in future research.

References:

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

[2] J.A.E.B. Janssen, 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*, Vol. 13, No. 3, 2010, pp. 229-238.

[3] M. Matthies, C. Giupponi, and B. Ostendorf, Environmental decision support systems: Current issues, methods and tools, *Environmental Modelling and Software*, Vol. 22, No. 2, 2007, pp. 123-127.

[4] H.T. Mowrer, Uncertainty in natural resource decision support systems: Sources, interpretation, and importance, *Computers and Electronics in Agriculture*, Vol. 27, No. 1-3, 2000, pp. 139-154.

[5] W.E. Walker, P. Harremoes, J. Rotmans, J.P. Van der Sluis, M.B.A.P. Van Asselt, Janssen, and M.P. Krayer von Krauss, Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support, *Integrated Assessment*, Vol. 4, No. 1, 2003, pp. 5-17.

[6] D.H. Loughlin, S.R. Ranjithan, E.D. Brill, and J.W. Baugh, Genetic algorithm approaches for addressing unmodelled objectives in optimization problems, *Engineering Optimization*, Vol. 33, No. 5, 2001, pp. 549-569.

[7] J.S. Yeomans, and Y. Gunalay, Simulation-optimization techniques for modelling to generate alternatives in waste management planning, *Journal of Applied Operational Research*, Vol. 3, No. 1, 2011, pp. 23-35.

[8] E.D. Brill, 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*, Vol. 28, No. 3, 1982, pp. 221-235.

[9] J.W. Baugh, S.C. Caldwell, and E.D. Brill, A mathematical programming approach for generating alternatives in discrete structural optimization, *Engineering Optimization*, Vol. 28, No. 1, 1997, pp. 1-31.

[10] E.M. Zechman, and S.R. Ranjithan, Generating alternatives using evolutionary algorithms for water resources and environmental management problems, *Journal of Water Resources Planning and Management*, Vol. 133, No. 2, 2007, pp. 156-165.

[11] Y. Gunalay, 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*, Vol. 19, No. 2, 2012, pp. 58-69.

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

[13] R. Imanirad, X.S. Yang, and J.S. Yeomans, A computationally efficient, biologically-inspired modelling-to-generate-alternatives method,

Journal on Computing, Vol. 2, No. 2, 2012, pp. 43-47.

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

[15] R. Imanirad, X.S. Yang, and J.S. Yeomans, A co-evolutionary, nature-inspired algorithm for the concurrent generation of alternatives, *Journal on Computing*, Vol. 2, No. 3, 2012, pp. 101-106.

[16] R. Imanirad, X.S. Yang, and J.S. Yeomans, Modelling-to-generate-alternatives via the firefly algorithm, *Journal of Applied Operational Research*, Vol. 5, No. 1, 2013, pp. 14-21.

[17] R. Imanirad, 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*, Vol. 5, No. 2, 2013, pp. 33-45.

[18] R. Imanirad, X.S. Yang, and J.S. Yeomans, A biologically-inspired metaheuristic procedure for modelling-to-generate-alternatives, *International Journal of Engineering Research and Applications*, Vol. 3, No. 2, 2013, pp. 1677-1686.

[19] J.S. Yeomans, Simultaneous Computing of Sets of Maximally Different Alternatives to Optimal Solutions, *International Journal of Engineering Research and Applications*, Vol. 7, No. 9, 2017, pp. 21-28.

[20] J.S. Yeomans, An Optimization Algorithm that Simultaneously Calculates Maximally Different Alternatives, *International Journal of Computational Engineering Research*, Vol. 7, No. 10, 2017, pp. 45-50.

[21] J.S. Yeomans, Computationally Testing the Efficacy of a Modelling-to-Generate-Alternatives Procedure for Simultaneously Creating Solutions, *Journal of Computer Engineering*, Vol. 20, No. 1, 2018, pp. 38-45.

[22] J.S. Yeomans, A Computational Algorithm for Creating Alternatives to Optimal Solutions, *Transactions on Machine Learning and Artificial Intelligence*, Vol. 5, No. 5, 2017, pp. 58-68.

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

[24] J.S. Yeomans, An Algorithm for Generating Sets of Maximally Different Alternatives Using Population-Based Metaheuristic Procedures, *Transactions on Machine Learning and Artificial Intelligence*, Vol. 6, No. 5, 2018, pp. 1-9.

[25] J.S. Yeomans, A Bicriterion Approach for Generating Alternatives Using Population-Based Algorithms, *WSEAS Transactions on Systems*, Vol. 18, No. 4, 2019, pp. 29-34.

[26] J.S. Yeomans, A Simulation-Optimization Algorithm for Generating Sets of Alternatives Using Population-Based Metaheuristic Procedures, *Journal of Software Engineering and Simulation*, Vol. 5, No. 2, 2019, pp. 1-6.

[27] J.S. Yeomans, A Stochastic, Dual-Criterion, Simulation-Optimization Algorithm for Generating Alternatives, *Journal of Computer Science Engineering*, Vol. 5, No. 6, 2019, pp. 1-10.

[28] J.S. Yeomans, A Multicriteria Simulation-Optimization Algorithm for Generating Sets of Alternatives Using Population-Based Metaheuristics, *WSEAS Transactions on Computers*, Vol. 18, No. 9, 2019, pp. 74-81.

[29] X.S. Yang, Firefly Algorithms for Multimodal Optimization, *Lecture Notes in Computer Science*, 5792, 2009, pp. 169-178.

[30] L.C. Cagnina, C.A. Esquivel, and C.A. Coello, Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer, *Informatica*, Vol. 32, 2008, pp. 319-326.