Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2018, 5(4):238-246

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

An Efficient Simultaneous Modelling-to-Generate-Alternatives Procedure

Julian Scott Yeomans

OMIS Area, Schulich School of Business, York University, Toronto, ON, M3J 1P3 Canada

Abstract "Real world" decision-making typically contains complex performance requirements riddled with incongruent performance conditions. This situation arises because most decision-making is characterized by complex problems possessing incompatible performance objectives together with opposing design requirements which are very problematic – if not impossible – to quantify and capture when the supporting decision models are actually formulated. There are invariably unmodelled components, not evident during model construction, that can significantly influence the relevance of the model's solutions. Accordingly, it is generally desirable to produce numerous, disparate alternatives that provide multiple, distinct perspectives to the problem. These alternatives should be near-optimal for all known objective(s), but be maximally different from each other when characterized by the solution structure of their decision variables. Such a maximally different solution construction method is referred to as modelling-to-generate-alternatives (MGA). This paper outlines an efficient optimization approach that can simultaneously create multiple, maximally different alternatives by employing the Firefly Algorithm. The efficiency of this metaheuristic mathematical programming approach is illustrated using a commonly-tested engineering optimization benchmark problem.

Keywords Firefly Algorithm, Modelling-to-generate-alternatives, Nature-inspired Metaheuristics

Introduction

Decision-making in the "real world" typically involves multifaceted problems possessing design requirements which are very difficult to formulate into an underlying mathematical programming model and tend to be inundated by numerous unquantifiable components [1-5]. While mathematically "best" answers provide optimal solutions to the modelled formulations, these answers are generally not the best solutions to the fundamental real problems as there are invariably unmodelled components not readily apparent when the model was constructed [1, 2, 6]. Therefore, it is frequently considered more desirable to create a reasonable number of very different alternatives that provide multiple, distinct perspectives for the specified problem [3, 7]. These alternatives should preferably all possess near-optimal objective values for all modelled objective(s), but be fundamentally dissimilar from each other when characterized by the system structures of their decision variables. Numerous methods collectively referred to as *modelling-to-generate-alternatives* (MGA) have been constructed in response to this multi-perspective, creation requirement [6, 7, 8].

The principal impetus underlying MGA is to produce a tractable set of alternatives that are good with respect to all measured objective(s) yet are fundamentally dissimilar from each other within the prescribed decision space. This resulting solution set should provide numerous perspectives that all perform comparably with respect to the modelled objectives, yet very differently with respect to any unmodelled components [5]. Clearly the decision-makers would have to conduct a subsequent comprehensive assessment of these alternatives to ascertain which option(s) most closely satisfies their very specific situations. Necessarily, MGA approaches are classified as a decision support processes rather than as explicit solution determination methods generally assumed for optimization.

Preceding MGA algorithms have employed direct processes for generating their alternatives by iteratively re-running their solution procedures whenever new alternatives must be produced [6-10]. These iterative methods follow the seminal MGA approach of Brill *et al*. [8] in which, once an initial problem formulation has been optimized, the supplementary alternatives are created one-by-one. Consequently, these incremental algorithms all require $n+1$ runnings of their respective approaches to optimize the initial problem and to then create their subsequent n alternatives [7, 11-13].

For optimization and calculation purposes, Yang [14-15] has established that the nature-inspired Firefly Algorithm (FA) is more computationally efficient than other commonly-used metaheuristic procedures such as enhanced particle swarm optimization, genetic algorithms, and simulated annealing [16-17]. However, what differentiates an FA from other population-based metaheuristics is that it has been explicitly designed to converge simultaneously into a pre-specified number of local (including global) optima within highly non-linear mathematical programming problems. Imanirad & Yeomans [12] have subsequently shown how the FA's functional optimization capabilities for determining numerous local optima can be adapted to concurrently generate all n maximally different alternatives required in an MGA approach after an initial optimal solution has been determined.

In this paper, it is shown how to *simultaneously* generate sets of maximally different solution alternatives by implementing a modified version of the nature-inspired FA [14-15] by extending the previous concurrent MGA approaches of Yeomans [18], Imanirad & Yeomans [12] and Imanirad *et al*. [13, 19-22]. Remarkably, this new MGA procedure extends the previous concurrent approaches of Imanirad *et al*. [13, 19-22] to permit the simultaneous generation of the globally optimal solution together with n locally optimal, maximally different alternatives in a single computational run. Explicitly, to generate the additional n maximally different solution alternatives, the new simultaneous MGA algorithm needs to run exactly the same number of times that an FA needs to run for function optimization purposes alone (namely once) irrespective of the value of n [23-24]. Therefore, this new, innovative simultaneous FA procedure is extremely computationally efficient for MGA purposes. This study illustrates the efficacy of the approach for simultaneously constructing multiple, good-but-very-different solution alternatives on a 100-peak multimodal optimization test problem [6, 18].

2. Firefly Algorithm for Function Optimization

While this section provides only a relatively brief synopsis of the FA procedure, more detailed explanations can be accessed in [12-15, 17]. The FA is a biologically-inspired, population-based metaheuristic. Each firefly in the population represents one potential solution to a problem and the initial population of fireflies should be distributed uniformly and randomly throughout the solution space. The solution approach employs the following three idealized rules: (i) All fireflies within the population are considered essentially unisex, so that any one firefly could potentially be attracted to any other firefly irrespective of their sex; (ii) The relative attractiveness between any two fireflies is directly proportional to their respective brightness. This implies that for any two flashing fireflies, the less bright firefly will always be inclined to move towards the brighter one. However, attractiveness and brightness both decrease as the relative distance between the fireflies increases. If there is no brighter firefly within its visible neighborhood, then the particular firefly will move about randomly; and, (iii). The brightness of a firefly is determined by the overall landscape of the objective function. Namely, for a maximization problem, the brightness can simply be considered proportional to the value of the objective function. Based upon these three rules, the basic operational steps of the FA can be summarized within the following pseudo-code.

Objective Function $F(\boldsymbol{X})$, $\boldsymbol{X} = (x_1, x_2, \ldots x_d)$

Generate the initial population of n fireflies, $\boldsymbol{X}_i$, $i = 1, 2, \ldots, n$

Light intensity I_i at $\boldsymbol{X}_i$ is determined by $F(\boldsymbol{X}_i)$

Define the light absorption coefficient γ

while (t < MaxGeneration)

for $i = 1: n$, all n fireflies

for $j = 1: n$, all n fireflies (inner loop)

if (I_i<I_j), Move firefly i towards j; **end if**
Vary attractiveness with distance r via $e^{-\gamma r}$
endfor*j*
end for*i*
Rank the fireflies and find the current global best solution $\boldsymbol{G}^*$
end while
Postprocess the results

In the FA, there are two important issues to resolve: the formulation of attractiveness and the variation of light intensity. For simplicity, it can always be assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with its encoded objective function value. 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 needs to vary with the degree of absorption. Consequently, the overall attractiveness of a firefly can be defined as

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

where β_0 is the attractiveness at distance r = 0 and γ is the fixed light absorption coefficient for the 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

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

In this expression of movement, the second term is due to the relative attraction and the third term is a randomization component. Yang [15] 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 explicitly noted that this expression represents a random walk biased toward brighter fireflies and if β_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 [15, 17]. 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 begin to converge into all of these local optima (including the global ones). Hence, by comparing the best solutions among all these optima, the global optima can easily be determined. Yang [15] proves 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 with n set in the range 20 to 50 [14, 15, 17].

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 to every other firefly anywhere within the solution domain. Hence, a single (usually global) optima can easily be reached. If the inner loop for j in the pseudo-code is removed and $\boldsymbol{X}_j$ is replaced by the current global best $\boldsymbol{G}^*$, then this implies that the FA reverts to a special case of the accelerated particle swarm optimization (PSO) algorithm. Subsequently, the computational efficiency of this special FA case is equivalent to that of enhanced PSO. Conversely, when $\gamma\rightarrow\infty$, the attractiveness is essentially zero along the sightline of all other fireflies. This is equivalent to the case where the fireflies randomly roam throughout a very thick foggy region with 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 asymptotic extremes, it is possible to adjust the parameters α and γ so that the FA can outperform both a random search and the enhanced PSO algorithms [17].

The computational efficiencies of the FA will be exploited in the subsequent MGA solution approach. As noted, within the two asymptotic extremes, the population in the FA can determine both the global optima as well as the local optima concurrently. The concurrency of population-based solution procedures holds huge computational and efficiency advantages for MGA purposes [7]. An additional advantage of the FA for MGA implementation is that the different fireflies essentially work independently of each other, implying that FA procedures are better than genetic algorithms and PSO for MGA because the fireflies will tend to aggregate more closely around each local optimum [15-17]. Consequently, with a judicious selection of parameter settings, the FA will simultaneously converge extremely quickly into both local and global optima [14, 15, 17].

3. Modelling to Generate Alternatives

Most mathematical programming approaches arising in the optimization literature have concentrated almost exclusively upon producing single optimal solutions to single-objective problem instances or, equivalently, generating noninferior solution sets to multi-objective formulations [2, 5, 8]. While such algorithms may efficiently generate solutions to the derived complex mathematical models, whether these outputs actually establish "best" approaches to the underlying real problems is certainly questionable [1-2, 6, 8]. In most "real world" decision environments, there are innumerable system objectives and requirements that are never explicitly apparent or included in the decision formulation stage [1, 5]. Furthermore, it may never be possible to explicitly express all of the subjective components because there are frequently numerous incompatible, competing, design requirements and, perhaps, adversarial stakeholder groups involved. Therefore, most subjective aspects of a problem necessarily must remain unquantified and unmodelled in the construction of the resultant decision models. This is a common occurrence in situations where the final decisions are constructed based not only upon clearly stated and modelled objectives, but also upon fundamentally subjective socio-political-economic goals and stakeholder preferences [7]. Numerous "real world" examples describing these types of incongruent modelling dualities appear in [6, 8-10].

When unquantified issues and unmodelled objectives exist, non-conventional approaches are required that not only search the decision space for noninferior sets of solutions, but must also explore the decision space for discernibly *inferior* alternatives to the modelled problem. In particular, any search for good alternatives to problems known or suspected to contain unmodelled objectives must focus not only on the non-inferior solution set, but also necessarily on an explicit exploration of the problem's inferior region.

To illustrate the implications of an unmodelled objective on a decision search, assume that the optimal solution for a quantified, single-objective, maximization decision problem is $\boldsymbol{X}^*$ with corresponding objective value $Z1^*$. Now suppose that there exists a second, unmodelled, maximization objective $Z2$ that subjectively reflects some unquantifiable "environmental/political acceptability" component. Let the solution $\boldsymbol{X}^a$, belonging to the noninferior, 2-objective set, represent a potential best compromise solution if both objectives could somehow have been simultaneously evaluated by the decision-maker. While $\boldsymbol{X}^a$ might be viewed as the best compromise solution to the real problem, it would appear inferior to the solution $\boldsymbol{X}^*$ in the quantified mathematical model, since it evidently must be the case that $Z1^a \leq Z1^*$. Consequently, when unmodelled objectives are factored into the decision-making process, mathematically inferior solutions for the modelled problem can prove optimal to the underlying real problem. Therefore, when unmodelled objectives and unquantified issues might exist, different solution approaches are needed in order to not only search the decision space for the noninferior set of solutions, but also to simultaneously explore the decision space for inferior alternative solutions to the modelled problem. Population-based methods such as the FA permit concurrent searches throughout a feasible region and thus prove to be particularly adept solution procedures for searching through such a problem's decision space.

The primary motivation behind MGA is to produce a manageably small set of alternatives that are quantifiably good with respect to modelled objectives yet are as different as possible from each other in the decision space. In doing this, the resulting alternative solution set is likely to provide truly different choices that all perform somewhat similarly with respect to the known modelled objective(s) yet very differently with respect to any unmodelled issues. By generating these good-but-different solutions, the decision-makers can explore desirable qualities within the alternatives that may prove to satisfactorily address the various unmodelled objectives to varying degrees of stakeholder acceptability.

In order to properly motivate an MGA search procedure, it is necessary to provide a more mathematically formal definition to the goals of the MGA process [6], [7].Suppose the optimal solution to an original mathematical model is $\boldsymbol{X^*}$ with objective value $Z^* = F(\boldsymbol{X^*})$. The following maximal difference model can then be solved to generate an alternative solution that is maximally different from $\boldsymbol{X}^*$:

$$\text{Max } \Delta = \sum_i |X_i - X_i^*|$$

$$\text{Subject to: } \quad \boldsymbol{X} \in D$$

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

where Δ represents some difference function (for clarity, shown as absolute in this instance) and T is a targeted tolerance value specified relative to the original optimal function value Z^*. T is a user-supplied value that determines how much of the inferior region is to be explored in the search for acceptable alternative solutions. This difference function concept can be extended into a measure of difference between any set of alternatives by replacing $\boldsymbol{X}^*$ in the objective of the maximal difference model and calculating the overall sum (or some other function) of the differences of the pairwise comparisons between each pair of alternatives – subject to the condition that each alternative is feasible and falls within the specified tolerance constraint.

The FA-based 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 using the FA to solve the corresponding maximal difference problem instance. By exploiting the population structure of the FA, the Fireflies collectively evolve toward different local optima within the solution space. 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.

4. FA-based Simultaneous MGA Computational Algorithm

The MGA method to be introduced 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 an FA to solve the corresponding, maximal difference problem. Each solution within the FA's population contains one potential set of p different alternatives. By exploiting the co-evolutionary solution structure within the population of the algorithm, the Fireflies collectively evolve each solution to-ward sets of different local optima within the solution space. In this process, each desired solution alternative undergoes the common search procedure of the FA. However, the survival of solutions depends both 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 generated in the decision space.

A direct process for generating alternatives with the FA would be to iteratively solve the maximum difference model by incrementally updating the target T whenever a new alternative needs to be produced and then re-running the algorithm. This iterative approach would parallel the original Hop, Skip, and Jump (HSJ) MGA algorithm of Brill *et al*. [8] in which, once an initial problem formulation has been optimized, supplementary alternatives are systematically created one-by-one through an incremental adjustment of the target constraint to force the sequential generation of the suboptimal solutions. While this approach is straightforward, it requires a repeated execution of the optimization algorithm [7, 12, 13].

To improve upon the stepwise alternative approach of the HSJ algorithm, a concurrent MGA technique was subsequently designed based upon the concept of co-evolution Imanirad *et al*. [13, 19, 21]. In the co-evolutionary approach, pre-specified stratified subpopulation ranges within the algorithm's overall population were established that collectively evolved the search toward the creation of the specified number of maximally different alternatives. Each desired solution alternative was represented by each respective subpopulation and each subpopulation underwent the common processing operations of the FA. The survival of solutions in each subpopulation depended 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].

By employing this co-evolutionary concept, it becomes possible to implement an FA-based MGA procedure that concurrently produces alternatives which possess objective function bounds that are somewhat analogous to those created by the sequential, iterative HSJ-styled solution generation approach. While each alternative produced by an HSJ procedure is maximally different only from the overall optimal solution (together with its bound on the objective value which is at least x% different from the best objective (i.e. x = 1%, 2%, etc.)), a concurrent procedure is able to generate alternatives that are no more than x% different from the overall optimal solution but with each one of these solutions being as maximally different as possible from every other generated alternative that was produced. Co-evolution is also much more efficient than the sequential HSJ-style approach in that it exploits the inherent population-based searches of FA procedures to concurrently generate the entire set of maximally different solutions using only a single population [12, 21].

While a concurrent approach exploits the population-based nature of the FA's solution approach, the co-evolution process occurs within each of the stratified subpopulations. The maximal differences between solutions in different subpopulations is 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 FA 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 circumvented.

The steps in the co-evolutionary alternative generation algorithm are as follows [18, 23, 24]:

Initialization Step. In this preliminary step, solve the original optimization problem to determine the optimal solution, ***X****. As with prior solution approaches Imanirad *et al*. [13, 19-22] and without loss of generality, it is entirely possible to forego this step and construct the algorithm to find ***X**** as part of its solution processing. However, such a requirement increases the number of computational iterations of the overall procedure and the initial stages of the processing focus upon finding ***X**** while the other elements of each population solution remain essentially "computational overhead". 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 ***X****. Note: The value for P has to have been set *a priori* by the decision-maker.

Step 1. Create the initial population of size K in which each solution is divided into P equally-sized partitions. The size of each partition corresponds to the number of variables for the original optimization problem. $\boldsymbol{A}_p$ represents the p^{th} alternative, $p = 1,\ldots,P$, in each solution.

Step 2. In each of the K solutions, evaluate each $\boldsymbol{A}_p$, $p = 1,\ldots,P$, 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*. A 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 measure is defined in Step 5). Note: Because the best solution to date is always retained in the population throughout each iteration of the FA, at least one solution will always be feasible. A feasible solution for the first step can always consists of P repetitions of ***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 $k = 1,\ldots, K$, calculate D_k, a distance measure between all of the alternatives contained within solution k.

As an illustrative example for determining a distance measure, calculate

$$D_k = \sum_{i=1toP} \sum_{j=1toP} \Delta (A_i,A_j).$$

This represents the total distance between all of the alternatives contained within solution k. Alternatively, the distance measure could be calculated by some other appropriately defined function.

Step 6. 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 FA "change operations" to the each of the solutions and return to Step.

It should be apparent that the stratification approach outlined in this algorithm could be readily modified to accommodate any of the population-based solution procedures. However, as noted in Section1, the FA is an algorithm specifically designed to simultaneously converge into numerous local optima which provides distinct computational advantages over other population-based metaheuristics. A disadvantage to the FA-based procedure in comparison to iterative MGA approaches lies in the extra computational overhead required to store the expanded population size for all of the alternatives and the additional solution time required to generate both the overall optimal solution together with the set of maximally different alternatives. Based upon preliminary testing and significant experimentation in the subsequent section, it seems that the additional storage requirements can be considered essentially negligible for all practical purposes and that the extra computational effort is virtually undetectable. However, these disadvantages could potentially become more pronounced for other problem instances that might be identified in subsequent extensions to this paper.

5. Computational Testing of Simultaneous MGA Algorithm

As stated previously, "real world" decision-makers often prefer to be able to select from a set of "near-optimal" alternatives that differ significantly from each other in terms of their system structures. The efficacy of the co-evolutionary MGA procedure to simultaneously produce maximally different alternatives will be demonstrated using a non-linear optimization problem taken from [6] and [18].

The mathematical formulation for this multimodal problem can be summarized as:

$$\text{Maximize } F(x,y) = \text{Sin}(19\pi x) + \frac{x}{1.7} + \text{Sin}(19\pi y) + \frac{y}{1.7} + 2$$

$$0.0 \le x \le 1.0 \qquad\qquad 0.0 \le y \le 1.0$$

The non-linear, feasible region contains 100 peaks separated by valleys in which the amplitudes of both the peaks and valleys increase as the values of the decision variables increase from the (0,0) toward (1,1). For the design parameters employed in this formulation, the best solution of $F(x,y) = 5.146$ occurs at point $(x,y) = (0.974, 0.974)$ [6, 18].

In order to create the set of different alternatives, extra target constraints that varied the value of T by up to 1.5% between successive alternatives were placed into the original formulation in order to force the generation of solutions maximally different from the initial optimal solution (i.e. the values of the bound were set at 1.5%, 3%, 4.5%, etc. for the respective alternatives). The MGA procedure was used to create the optimal solution and the 10 maximally different solutions shown in Table 1.

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

Increment	**1.5% Increment Between Alternatives**		
	***F*(*x*,*y*)**	***x***	***y***
Optimal	5.14	0.97	0.97
Alternative 1	5.11	0.97	0.98
Alternative 2	5.06	0.98	0.87
Alternative 3	5.01	0.87	0.76
Alternative 4	4.98	0.87	0.98
Alternative 5	4.92	0.76	0.98
Alternative 6	4.90	0.87	0.66
Alternative 7	4.77	0.45	0.87
Alternative 8	4.73	0.98	0.34
Alternative 9	4.66	0.13	0.97
Alternative 10	4.65	0.98	0.13

As detailed earlier, most "real world" optimization problems tend to be riddled with incongruent performance specifications that are exceedingly difficult to quantify. Therefore, it is frequently desirable to generate a set of quantifiably good alternatives that deliver very different perspectives to any potentially unmodelled performance design matters during the solution formulation stage. Any unique performance characteristics contained within these divergent alternatives can produce very different system performance with respect to unmodelled issues, thereby hopefully capturing some of the previously unmodelled issues into the actual solution process.

The benchmark test problem demonstrates how the co-evolutionary MGA modelling algorithm can simultaneously generate multiple alternatives via the computationally efficient FA that satisfy all known system performance criteria to within a prespecified bounds and while remaining as maximally different from each other as possible in the solution space. In addition to its alternative generating capabilities, the FA aspect of the MGA approach concurrently performs soundly with respect to function optimization. Namely, for the test problem, the best overall solution determined in the FA-based MGA procedure is identical to the optimal solution found in [6].

This section highlights several noteworthy discoveries with respect to the simultaneous FA-based MGA procedure: (i) Due to the evolving nature of its population-based searches, the co-evolutionary capabilities within an FA can generate more good alternatives than planners could generate using other MGA approaches; (ii) By the nature of the MGA procedure, the alternatives created are good for planning, since all of their structures will be maximally different from each other (specifically, these differences are not just simply different from the overall optimal solution as in an HSJ-style approach to MGA); and, (iv) The algorithm is very computationally efficient, since it need run only a single time to generate its entire set of multiple, good alternatives (explicitly, to generate *n* solution alternatives, the MGA algorithm needs to run exactly once, irrespective of value of *n*).

6. Conclusions

In general, "real world" decision-making entails multifaceted performance requirements further complicated by discordant performance specifications and unquantifiable objectives. These problems often contain incongruent design issues which are very problematic – if not impossible – to capture when the supporting decision models need to be formulated. Thus, there are invariably unmodelled design requirements, not apparent during the model formulation, that can critically affect the adequacy of the model's solutions. These ambiguous and conflicting dimensions force the decision-makers into integrating many in congruencies into their decision process prior to final solution construction. In the face of such inconsistencies, it is improbable that any single solution could ever be constructed to simultaneously satisfy all of the contradictory system requirements without a substantial rebalancing of the various tradeoffs. Consequently, any auxiliary modelling techniques used in the decision formulation must somehow simultaneously account for all of these features while being flexible enough to capture the inherent planning uncertainties.

In this paper, an MGA procedure was constructed that illustrated how the computationally efficient, population-based FA could be exploited to simultaneously produce numerous near-best, maximally different alternatives. In this MGA role, the simultaneous procedure creates numerous alternatives possessing the essential problem characteristics, with each generated solution providing a very different perspective. Since FA approaches have been employed in solving a diverse spectrum of problem types, the practicability of this simultaneous MGA procedure can clearly be extended to numerous disparate applications. These extensions will become the focus in future studies.

References

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

[2]. Janssen, J.A.E.B., Krol, M.S., Schielen, R.M.J., & Hoekstra, A.Y. (2010). The Effect of Modelling Quantified Expert Knowledge and Uncertainty Information on Model Based Decision Making. *Environmental Science and Policy*, 13(3): 229-238.

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

[4]. Mowrer, H.T. (2000). Uncertainty in Natural Resource Decision Support Systems: Sources, Interpretation and Importance. *Computers and Electronics in Agriculture*, 27(1-3): 139-154.
[5]. Walker, W.E., Harremoes, P., Rotmans, J., Van der Sluis, J.P., Van Asselt, M.B.A., Janssen, P.,& Krayer von Krauss, M.P. (2003). Defining Uncertainty – A Conceptual Basis for Uncertainty Management in Model-Based Decision Support. Integrated Assessment, 4(1): 5-17.
[6]. Loughlin, D.H., Ranjithan, S.R, Brill, E.D., & Baugh, J.W. (2001). Genetic Algorithm Approaches for Addressing Unmodeled Objectives in Optimization Problems. *Engineering Optimization*,33(5):549-569.
[7]. Yeomans, J.S., & Gunalay, Y. (2011). Simulation-Optimization Techniques for Modelling to Generate Alternatives in Waste Management Planning.*Journalof Applied Operational Research*, 3(1):23-35.
[8]. Brill, E.D., Chang, S.Y., & Hopkins L.D. (1982). Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning. *Management Science*, 28(3), 221-235.
[9]. Baugh, J.W., Caldwell, S.C., & Brill E.D. (1997). A Mathematical Programming Approach for Generating Alternatives in Discrete Structural Optimization. *Engineering Optimization*, 28(1), 1-31.
[10]. Zechman, E.M., & Ranjithan, S.R. (2004). An Evolutionary Algorithm to Generate Alternatives (EAGA) for Engineering Optimization Problems. *Engineering Optimization*, 36(5), 539-553.
[11]. Gunalay, Y., Yeomans, J.S., & Huang, G.H. (2012). Modelling to Generate Alternative Policies in Highly Uncertain Environments: An Application to Municipal Solid Waste Management Planning. *Journal on Environmental Informatics*, 19(2), 58-69.
[12]. Imanirad, R., & Yeomans, J.S. (2013). Modelling to Generate Alternatives Using Biologically Inspired Algorithms. In: X.S. Yang, Ed. Swarm Intelligence and Bio-Inspired Computation: Theory and Applications, Amsterdam (Netherlands): Elsevier, 313-333.
[13]. Imanirad, R., Yang, X.S., & Yeomans, J.S. (2012a). A Computationally Efficient, Biologically-Inspired Modelling-to-Generate-Alternatives Method. *Journal on Computing*, 2(2), 43-47.
[14]. Yang, X.S. (2009). Firefly Algorithms for Multimodal Optimization. *Lecture Notes in Computer Science*, 5792, 169-178.
[15]. Yang, X.S. (2010). *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, UK,2nd Ed.
[16]. Cagnina, L.C., Esquivel, C.A., & Coello, C.A. (2008). Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer. *Informatica*, 32, 319-326.
[17]. Gandomi, A.H., Yang X.S., & Alavi, A.H. (2011). Mixed Variable Structural Optimization Using Firefly Algorithm. *Computers and Structures*, 89(23-24), 2325-2336.
[18]. Yeomans, J.S. (2018). 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*, Cham (Switzerland): Springer, 261-273.
[19]. Imanirad,R., Yang, X.S., & Yeomans, J.S. (2012). A Co-evolutionary, Nature-Inspired Algorithm for the Concurrent Generation of Alternatives. *Journal on Computing*, 2(3): 101-106.
[20]. Imanirad, R., Yang, X.S., & Yeomans, J.S. (2013). Modelling-to-Generate-Alternatives Via the Firefly Algorithm. *Journal of Applied Operational Research*, 5(1), 14-21.
[21]. Imanirad, R., Yang, X.S., & Yeomans, J.S. (2013). A Concurrent Modelling to Generate Alternatives Approach Using the Firefly Algorithm. *International Journal of Decision Support System Technology*, 5(2), 33-45.
[22]. Imanirad, R., Yang, X.S., & Yeomans, J.S. (2013). A Biologically-Inspired Metaheuristic Procedure for Modelling-to-Generate-Alternatives. *International Journal of Engineering Re-search and Applications*, 3(2), 1677-1686.
[23]. Yeomans, J.S. (2017). Simultaneous Computing of Sets of Maximally Different Alternatives to Optimal Solutions. *International Journal of Engineering Research and Applications*, 7(9), 21-28.
[24]. Yeomans, J.S. (2017). An Optimization Algorithm that Simultaneously Calculates Maximally Different Alternatives. *International Journal of Computational Engineering Research*. 7(10), 45-50.