RESEARCH ARTICLE OPEN ACCESS

Simultaneous Computing of Sets of Maximally Different Alternatives to Optimal Solutions

Julian Scott Yeomans*
** (OMIS Area, Schulich School of Business, York University, 4700 Keele Street, Toronto, ON, M3J 1P3 Canada*
Corresponding author: Julian Scott Yeomans

ABSTRACT
In solving many "real world" engineering optimization applications, it is generally preferable to formulate several quantifiably good alternatives that provide distinct perspectives to the particular problem. This is because decision-making typically involves complex problems that are riddled with incompatible performance objectives and contain competing design requirements which are very difficult – if not impossible – to capture and quantify at the time that the supporting mathematical programming models are actually constructed. There are invariably unmodelled design issues, not apparent at the time of model construction, which can greatly impact the acceptability of the model's solutions. Consequently, it is preferable to generate several, distinct alternatives that provide multiple, disparate perspectives to the problem. These alternatives should possess near-optimal objective measures with respect to all known modelled objective(s), but be fundamentally different from each other in terms of their decision variables. This solution approach is referred to as modelling to generate-alternatives (MGA). This paper provides an efficient computational procedure for simultaneously generating multiple different alternatives to optimal solutions that employs the Firefly Algorithm. The efficacy of this approach will be illustrated using a well-known engineering optimization benchmark problem.
***Keywords*:** Biologically-inspired Metaheuristic Algorithms, Firefly Algorithm, Modelling-to-generate-alternatives

Date of Submission: 18-09-2017 Date of acceptance: 28-09-2017

I. INTRODUCTION

Typical "real world" decision-making involves complex problems that possess design requirements which are frequently very difficult to incorporate into their supporting mathematical programming formulations and tend to be plagued by numerous unquantifiable components [1][2][3]. While mathematically optimal solutions provide the best answers to these modelled formulations, they are generally not the best solutions to the underlying real problems as there are invariably unmodelled aspects not apparent during the model construction phase [1][2]. Hence, it is generally considered desirable to generate a reasonable number of very different alternatives that provide multiple, contrasting perspectives to the specified problem [4]. These alternatives should preferably all possess near-optimal measures with respect to all of the modelled objective(s), but be as different as possible from each other in terms of the system structures characterized by their decision variables. Several approaches collectively referred to as modelling-to-generate-alternatives (MGA) have been developed in response to this multi-solution creation requirement [4]-[9].

The primary motivation behind MGA is to construct a manageably small set of alternatives that are good with respect to all measured objective(s) yet are fundamentally dissimilar within the prescribed decision space. The resulting set of alternatives should provide diverse approaches that all perform similarly with respect to the known modelled objectives, yet very differently with respect to any unmodelled issues [3][10]. Clearly the decision-makers must then conduct a sub-sequent comprehensive comparison of these alternatives to determine which options would most closely satisfy their very specific circumstances. Therefore, MGA methods must necessarily be classified as decision support processes in contrast to the explicit solution determination methods assumed, in general, for optimization.

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 Firefly Algorithm (FA) [10][11] by extending previous concurrent MGA approaches [6]-[9][12]-[14]. For calculation and optimization, it has been demonstrated that the FA is more computationally efficient than such commonly-employed metaheuristics as enhanced particle swarm optimization, simulated annealing, and genetic algorithms [11][15]. The MGA procedure extends the

earlier approaches of Imanirad et al. [6]-[9][12][13] to now permit the simultaneous generation of the desired number of alternatives in a single computational run. This new simultaneous FA-based MGA procedure is extremely computationally efficient. This paper illustrates the efficacy of the new FA approach for simultaneously constructing multiple, good-but-very-different solution alternatives on a commonly evaluated benchmark engineering optimization test problem [16].

II. FIREFLY ALGORITHM FOR OPTIMIZATION

While this section provides only a relatively brief synopsis of the FA procedure [5][12][13], more comprehensive explanations appear in [10][11]. The FA is a biologically-inspired, population-based metaheuristic. Each firefly in the population represents one potential solution to a problem and the population of fireflies should initially be distributed uniformly and randomly throughout the solution space. The solution approach employs three idealized rules. (i) The brightness of a firefly is determined by the overall landscape of the objective function. Namely, for a maximization problem, the brightness is simply considered to be proportional to the value of the objective function. (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. (iii) All fireflies within the population are considered unisex, so that any one firefly could potentially be attracted to any other firefly irrespective of their sex. Based upon these three rules, the basic operational steps of the FA can be summarized within the following pseudo-code [11].

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,…, 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 [11] 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 $\beta_0 = 0$, 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 [11,15]. 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 the local optima (including the global ones). Hence, by comparing the best solutions among all these optima, the global optima can easily be determined. Yang (2010) 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 [10,15].

III. MODELLING TO GENERATE ALTERNATIVES

Most optimization methods appearing in the mathematical programming 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][3][5][12][13]. 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 questionable [1][2][5]. In most "real world" decision environments, there are innumerable system requirements and objectives that are never included or apparent in the decision formulation stage [1][3]. Furthermore, it may never be possible to explicitly incorporate all of the subjective components because there are frequently many incompatible, competing, design interpretations and, perhaps, adversarial stakeholders involved. Therefore, most of the subjective aspects of a problem necessarily remain unquantified and unmodelled in the construction of the resultant decision models. This is a common occurrence in situations where final decisions are constructed based not only upon clearly stated and modelled objectives, but also upon more fundamentally subjective socio-political-economic goals and stakeholder preferences [4]. Numerous "real world" examples describing these types of incongruent modelling dualities are discussed in [5][17][18].

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 components must focus not only on the non-inferior solution set, but also necessarily on an explicit exploration of the problem's inferior decision space.

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 "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 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 solution methods such as the FA permit concurrent searches throughout a feasible region and thus prove to be particularly adept procedures for searching through 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 the known modelled objectives yet are as different as possible from each other in the decision space. The resulting alternatives are likely to provide truly different choices that all perform somewhat similarly with respect to the modelled objective(s) yet very differently with respect to any unknown unmodelled issues. By generating a set of 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.

$$\text{Maximize } \Delta(\mathbf{X}, \mathbf{X}^*) = \sum_i |X_i - X_i^*| \qquad \text{[P1]}$$

$$\text{Subject to: } \boldsymbol{X} \in D$$
$$|F(\boldsymbol{X}) - \boldsymbol{Z}^*| \leq T$$

where Δ represents some difference function (for clarity, shown as an absolute difference in this instance and T is a targeted tolerance value specified relative to the problem's original optimal objective $\boldsymbol{Z}^*$. 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.

IV. FA-DRIVEN 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 toward 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 [5] 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 [4][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 [6][8][12][13]. 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 [4].

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 [6][8].

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 simultaneous co-evolutionary alternative generation algorithm are as follows:

Initialization Step. In this preliminary step, solve the original optimization problem to determine the optimal solution, X*. As with prior solution approaches [6]-[9][12][13] and without loss of generality, it is entirely possible to forego this step and construct the algorithm to find $\boldsymbol{X^*}$ 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 $\boldsymbol{X^*}$ while the other elements of each population solution remain essentially "computational overhead". Based upon the objective value F(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 $\boldsymbol{X}^*$.

This step simultaneously selects a set of alternatives that respectively satisfy different values of the target T while being as far apart as possible (i.e. maximally different as defined in the maximal difference model) from the other solutions generated. By the co-evolutionary nature of the FA, the alternatives are simultaneously generated in one pass of the procedure rather than the p implementations suggested by the necessary increments to T in the maximal difference problem.

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

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

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 2.

V. COMPUTATIONAL TESTING OF THE FIREFLY ALGORITHM USED FOR MGA

As described earlier, "real world" decision-makers generally 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. The ability of the FA MGA procedure to simultaneously produce such maximally different alternatives will be demonstrated using an optimization problem that has frequently been employed as a standard benchmark test problem for constrained, non-linear engineering optimization algorithms [16]. The mathematical formulation for this problem is:

$$\text{Min } F(\boldsymbol{X}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

Subject to:

$$g_1(\boldsymbol{X}) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \le 0$$

$$g_2(\boldsymbol{X}) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0$$

$$g_3(\boldsymbol{X}) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0$$

$$g_4(\boldsymbol{X}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 12x_7 \le 0$$

$$-10 \le x_i \le 10, \quad i = 1, 2, 3, 4, 5, 6, 7$$

The optimal solution for the specific design parameters employed within this formulation F(X*) = 680.6300573 with decision variable values of $\boldsymbol{X}^*$ = (2.330499, 1.951372, -0.4775414, 4.365726, 0.6244870, 1.038131, 1.594227) [16].

In order to create the set of different alternatives,

extra target constraints that varied the value of *T* by up to 1% 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%, 2%, 3%, etc. for the respective alternatives). The MGA maximal difference algorithm described in the previous section was run to produce the optimal solution and the 10 maximally different solutions shown in Table 1. Subsequently, target constraints that varied T by up to 2.5% between successive alternatives were employed. The MGA algorithm was run again to produce the optimal solution and the 10 maximally different solutions shown in Table 2.

Table 1. Objective Values and Solutions for the 11 Maximally Different Alternativeswith 1% Increment.

Increment	**1% Increment Between Alternatives**							
	F(*X*)	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Optimal	680.630	2.3304	1.9513	-0.4775	4.3657	0.6244	1.0381	1.5942
Alternative 1	683.917	2.3025	1.9353	-0.4881	4.3333	-0.6169	1.0355	1.5889
Alternative 2	687.580	2.2892	1.8985	-0.4605	4.3364	-0.5962	1.0208	1.5782
Alternative 3	696.899	2.2934	1.9096	-0.4397	4.3369	-0.6616	1.0331	1.6176
Alternative 4	705.926	2.3080	1.9171	-0.4724	4.3343	-0.6578	1.053	1.6078
Alternative 5	706.837	2.2913	1.9003	-0.3965	4.3548	-0.6388	1.0796	1.6023
Alternative 6	718.478	2.2904	1.9037	-0.427	4.3637	-0.5871	0.9955	1.6230
Alternative 7	725.652	2.3428	1.9158	-0.4459	4.3929	-0.6672	1.0382	1.6129
Alternative 8	730.091	2.2892	1.8985	-0.4605	4.3364	-0.5962	1.0208	1.5782
Alternative 9	741.897	2.2904	1.9037	-0.427	4.3637	-0.5871	0.9955	1.6230
Alternative 10	747.925	2.3577	1.9121	-0.4395	4.3314	-0.5869	1.0038	1.6148

Table 2. Objective Values and Solutions for the 11 Maximally Different Alternativeswith 2.5% Increment

Increment	**2.5% Increment Between Alternatives**							
	F(*X*)	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Optimal	680.630	2.3304	1.9513	-0.4775	4.3657	0.6244	1.0381	1.5942
Alternative 1	687.022	2.3056	1.9076	-0.4245	4.3256	-0.6184	1.0388	1.6067
Alternative 2	711.793	2.3174	1.9111	-0.4084	4.3668	-0.6166	1.0759	1.6116
Alternative 3	730.671	2.2916	1.9496	-0.4442	4.3474	-0.6154	1.0555	1.5864
Alternative 4	744.901	2.3468	1.9118	-0.4087	4.3557	-0.6283	0.9899	1.6024
Alternative 5	756.260	2.2985	1.9019	-0.4452	4.3577	-0.5927	1.0022	1.5770
Alternative 6	779.735	2.3463	1.9397	-0.4338	4.3425	-0.5867	1.0457	1.6301
Alternative 7	796.641	2.3011	1.9128	-0.4282	4.3386	-0.5758	1.0035	1.6227
Alternative 8	811.767	2.3539	1.9579	-0.4543	4.3338	-0.6425	1.0413	1.6155
Alternative 9	832.123	2.3690	1.9208	-0.4181	4.4001	-0.617	1.0411	1.6374
Alternative 10	846.019	2.2967	1.897	-0.4684	4.3467	-0.6374	1.0252	1.5721

As described earlier, most "real world" optimization applications tend to be riddled with incongruent performance requirements that are exceedingly difficult to quantify. Consequently, it is preferable to create a set of quantifiably good alternatives that 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, hopefully thereby addressing some of the unmodelled issues into the actual solution process.

The example in this section underscores how a co-evolutionary MGA modelling perspective can be used to simultaneously generate multiple alternatives that satisfy known system performance criteria according to the prespecified bounds and yet remain as maximally different from each other as possible in the decision space. In addition to its alternative generating capabilities, the FA component of the MGA approach simultaneously performs extremely well with respect to its role in function optimization. It should be explicitly noted that the cost of the overall best solution produced by the MGA procedure is indistinguishable from the one determined in [16].

VI. CONCLUSIONS

"Real world" engineering optimizationproblems generally possess multidimensional performance specifications that are compounded by incompatible performance objectives and unquantifiable modelling features. These problems usually contain incongruent design requirements which are very difficult – if not impossible – to capture at the time that supporting decision models are formulated. Consequently, there are invariably unmodelled problem facets, not apparent during the model construction, that can greatly impact the acceptability of the model's solutions to those end users that must actually implement the solution. These uncertain and competing dimensions force decision-makers to integrate many conflicting sources into their decision process prior to final solution construction. Faced with such incongruencies, it is unlikely that any single solution could ever be constructed that simultaneously satisfies all of the ambiguous system requirements without some significant counterbalancing involving numerous tradeoffs. Therefore, any ancillary modelling techniques used to support decision formulation have to somehow simultaneously account for all of these features while being flexible enough to encapsulate the impacts from the inherent planning uncertainties.

In this paper, an MGA procedure was presented that demonstrated how the population structures of a computationally efficient FA could be exploited to simultaneously generate multiple, maximally different, near-best alternatives. In this MGA capacity, the approach produces numerous solutions possessing the requisite structural characteristics, with each generated alternative guaranteeing a very different perspective to the problem. The computational example has demonstrated several important findings with respect to the simultaneous FA-based MGA method: (i) The co-evolutionary capabilities within the FA can be exploited to generate more good alternatives than planners would be able to create using other MGA approaches because of the evolving nature of its population-based solution searches; (ii) By the design of the MGA algorithm, the alternatives generated are good for planning purposes since all of their structures will be maximally different from each other (i.e. these differences are not just simply different from the overall optimal solution as in an HSJ-style approach to MGA); and, (iv) The approach is computationally efficient since it need only be run a single time in order to generate its entire set of multiple, good solution alternatives (i.e. to generate n solution alternatives, the MGA algorithm needs to run exactly once irrespective of the value of n). Since FA techniques can be modified to solve a wide variety of problem types, the practicality of this MGA approach can clearly be extended into numerous disparate planning applications. These extensions will be studied in future research.

REFERENCES

[1] Brugnach, M., Tagg, A., Keil, F., and De Lange W.J., Uncertainty matters: computer models at the science-policy interface, *Water Resources Management21*, 2007, 1075-1090.

[2] Janssen, J.A.E.B., Krol, M.S., Schielen, R.M.J., and Hoekstra, A.Y., The effect of modelling quantified expert knowledge and uncertainty information on model based decision making,*Environmental Science and Policy13(3)*,2010, 229-238.

[3] Walker, W.E., Harremoes, P., Rotmans, J., Van der Sluis, J.P., Van Asselt, M.B.A., Janssen, P., and Krayer von Krauss, M.P., Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support,*Integrated Assessment4(1)*,2003, 5-17.

[4] Yeomans, J.S., and Gunalay, Y., Simulation-Optimization Techniques for Modelling to Generate Alternatives in Waste Management Planning,*Journal of Applied Operational Research3(1)*, 2011, 23-35.

[5] Brill, E.D., Chang, S.Y., and Hopkins L.D., Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning, *Management Science28(3)*, 1982, 221-235.

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

[7] Imanirad, R., Yang, X.S., and Yeomans, J.S.,Modelling-to-Generate-Alternatives Via the Firefly Algorithm,*Journal of Applied Operational Research5(1)*, 2013, 14-21.

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

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

[10] Yang, X.S., Firefly Algorithms for Multimodal Optimization,*Lecture Notes in Computer Science5792*, 2009, 169-178.

[11] Yang, X.S., *Nature-Inspired metaheuristic*

algorithms 2nd Ed (UK: Luniver Press, Frome, 2010).

[12] Imanirad, R., Yang, X.S., and Yeomans, J.S., A Computationally Efficient, Biologically-Inspired Modelling-to-Generate-Alternatives Method,*Journal on Computing2(2)*, 2012, 43-47.

[13] Imanirad, R., Yang, X.S., and Yeomans, J.S., A Co-evolutionary, Nature-Inspired Algorithm for the Concurrent Generation of Alternatives,*Journal on Computing2(3)*,2012,101-106.

[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*, X.S. Yang, Editor. Heidelberg: Springer (Germany),2018.

[15] Gandomi, A.H., Yang X.S., and Alavi, A.H., Mixed Variable Structural Optimization Using Firefly Algorithm,*Computers and Structures, 89(23-24)*,2011, 2325-2336.

[16] Aragon, V.S.,Esquivel, S.C., and Coello, C.C.A., A Modified Version of a T-Cell Algorithm for Constrained Optimization Problems,*International Journal for Numerical Methods In Engineering84*, 2010, 51–378.

[17] Baugh, J.W., Caldwell, S.C., and Brill E.D., A Mathematical Programming Approach for Generating Alternatives in Discrete Structural Optimization,*Engineering Optimization28(1)*,1997, 1-31.

[18] Zechman, E.M., and Ranjithan, S.R., An Evolutionary Algorithm to Generate Alternatives (EAGA) for Engineering Optimization Problems,*Engineering Optimization36(5)*,2004, 539-553.

Julian Scott Yeomans. “Simultaneous Computing of Sets of Maximally Different Alternatives to Optimal Solutions.” International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 9, 2017, pp. 21–28.