

A Co-evolutionary, Nature-Inspired Algorithm for the Concurrent Generation of Alternatives

Raha Imanirad, Xin-She Yang, and Julian S. Yeomans

Abstract— Engineering optimization problems usually contain multifaceted performance requirements that can be riddled with unquantifiable specifications and incompatible performance objectives. Such problems typically possess competing design requirements which are very difficult – if not impossible – to quantify and capture at the time of model formulation. 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, when solving many “real life” mathematical programming applications, it is generally preferable to formulate several quantifiably good alternatives that provide very different 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 the system structures characterized by their decision variables. This solution approach is referred to as modelling-to-generate-alternatives (MGA). This study demonstrates how the nature-inspired, Firefly Algorithm can be used to concurrently create multiple solution alternatives that both satisfy required system performance criteria and yet are maximally different in their decision spaces. This new co-evolutionary approach is very computationally efficient, since it permits the concurrent generation of multiple, good solution alternatives in a single computational run rather than the multiple implementations required in previous MGA procedures.

Index Terms— Firefly Algorithm, Modelling-to-generate-alternatives, Nature-inspired Algorithms

I. INTRODUCTION

Engineering optimization problems typically involve complex problems that possess design requirements which are very difficult to incorporate into supporting mathematical programming models and tend to be riddled with unquantifiable specifications [1]-[4]. While mathematically optimal solutions provide the best answers to these modelled problems, they are generally not the best solutions to the underlying real problems as there are invariably unquantified issues and unmodelled objectives not apparent at the time of model construction [1][2][5]. Consequently, it is generally

preferable to generate a reasonable number of very different alternatives that provide multiple, contrasting perspectives to the specified problem [6][7]. These alternatives should preferably all possess near-optimal objective measures with respect to all of the modelled objective(s), but be as fundamentally different from each other as possible 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 [5][7] in response to this option creation requirement. The primary motive behind MGA is to produce a manageably small set of alternatives that are good with respect to all modelled objective(s) yet are as different as possible from each other within the decision space. The resulting set of alternatives should provide solutions that all perform somewhat similarly with respect to the modelled objectives, yet very differently with respect to the unmodelled issues [4].

In this paper, it is shown how to efficiently generate a set of maximally different solution alternatives by implementing a modified version of the biologically-inspired Firefly Algorithm (FA) of Yang [8][9] in conjunction with a new co-evolutionary MGA procedure. For calculation and optimization purposes, Yang [9] has demonstrated that the FA is more computationally efficient than such commonly-used metaheuristics as genetic algorithms, simulated annealing, and enhanced particle swarm optimization [10]. The MGA procedure provided in this study extends the earlier approach of Imanirad *et al.* [11] and uses the concept of co-evolution to concurrently generate the desired number of solution alternatives in a single computational run of the algorithm. Hence, this innovative concurrent co-evolutionary FA procedure is extremely computationally efficient for MGA purposes. This study illustrates the efficacy of this new approach on a highly non-linear, 100-peak multimodal optimization test problem [5].

II. FIREFLY ALGORITHM FOR FUNCTION OPTIMIZATION

While this section provides a brief synopsis of the FA procedure, more specific details can be found in [8][9][11]. The FA is a nature-inspired, population-based metaheuristic. Each firefly in the population represents one potential solution to the problem. The initial firefly population is distributed randomly and uniformly throughout the solution space. The solution procedure 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

Manuscript received August 30, 2012.

R. Imanirad is with the OMIS Area, Schulich School of Business, York University, Toronto, ON, M3J 1P3 Canada (e-mail: Rimanirad09@schulich.yorku.ca).

X-S. Yang is with the Mathematics & Scientific Computing Area, National Physical Laboratory, Teddington, Middlesex TW11 0LW, UK (e-mail: xy227@cam.ac.uk).

J.S.. Yeomans is with the OMIS Area, Schulich School of Business, York University, Toronto, ON, M3J 1P3 Canada (corresponding author. phone: 416-736-5074; fax: 416-736-5687; e-mail: syeomans@schulich.yorku.ca).

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. Based upon these three rules, the basic operational steps of the FA are summarized within the pseudo-code of Figure 1.

Figure 1: Pseudo Code of the Firefly Algorithm

```

Objective Function  $F(\mathbf{X})$ ,  $\mathbf{X} = (x_1, x_2, \dots, x_d)$ 
Generate the initial population of  $n$  fireflies,  $\mathbf{X}_i$ ,  $i = 1, 2, \dots, n$ 
Light intensity  $I_i$  at  $\mathbf{X}_i$  is determined by  $F(\mathbf{X}_i)$ 
Define the light absorption coefficient  $\gamma$ 
while ( $t < \text{MaxGeneration}$ )
  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  $e^{-\gamma r}$ 
    end for  $j$ 
  end for  $i$ 
  Rank the fireflies and find the current global best solution  $\mathbf{G}^*$ 
end while
Postprocess the results
  
```

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 \mathbf{X} would be its calculated objective value $F(\mathbf{X})$. 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

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

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 \mathbf{X}_i and \mathbf{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

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

In this expression of movement, the second term is due to the relative attraction and the third term is a randomization component. Yang [9] 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 [9]. In any given optimization problem, for a very large number of fireflies $n \gg k$ 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 [9] demonstrates that the FA will approach the global optima when $n \rightarrow \infty$ and the number of iterations t , is set so that $t \gg 1$. 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 the inner loop for j in Figure 1 is removed and \mathbf{X}_j is replaced by the current global best \mathbf{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 the 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]. 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. Depending upon the parameter settings, the FA will converge extremely quickly into both local and global optima [8].

III. MODELLING TO GENERATE ALTERNATIVES WITH THE FIREFLY ALGORITHM

Notwithstanding their fundamental limitations, most mathematical programming approaches emanating from the engineering optimization literature have focused almost exclusively upon producing optimal solutions to single-objective problem instances or, equivalently, generating noninferior solution sets to multi-objective problem formulations. While such algorithms may efficiently generate solutions to the derived complex mathematical models, whether their results actually establish "best" approaches for

providing appropriate decisions to the underlying real problems is certainly questionable. In most “real world” decision problems, there are numerous system objectives and requirements that are never explicitly apparent or included in the decision formulation stage [1][4]. Furthermore, it may never be possible to explicitly express all of the subjective considerations because there are frequently numerous incompatible, competing, design requirements and, perhaps, adversarial stakeholder groups. Therefore most subjective aspects of a problem 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, political and socio-economic goals and stakeholder preferences [7]. Numerous “real world” examples of this type of incongruent modelling duality are described in [5] and [12]-[14].

When unmodelled objectives and unquantified issues exist, different approaches are required in order to not only search the decision space for the noninferior set of solutions, but also to explore the decision space for *inferior* alternative solutions 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 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 X^* with corresponding objective value ZI^* . Now suppose that there exists a second, unmodelled, maximization objective $Z2$ that subjectively reflects environmental/political acceptability. Let the solution 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 X^a might be viewed as the best compromise solution to the real problem, it would clearly appear inferior to the solution X^* in the quantified model, since it must be the case that $ZI^a \leq ZI^*$. Consequently, when unmodelled objectives are factored into the decision making process, mathematically inferior solutions for the modelled problem can be optimal for the real problem. Therefore, when unmodelled objectives and unquantified issues might exist, different approaches are required 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 procedures such as the FA permit concurrent searches throughout a feasible region and thus prove to be particularly adept methods 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 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 alternatives that may satisfy the unmodelled objectives to varying degrees of stakeholder acceptability. Obviously the solution-setters must then conduct a subsequent comprehensive comparison of the alternatives to determine which options would most closely satisfy their very specific circumstances. Thus, an MGA approach should necessarily be considered as one of decision support rather than of explicit solution determination.

In order to properly motivate an MGA search procedure, it is necessary to provide a more formal definition of the goals of the MGA process [5][7]. Suppose the optimal solution to an original mathematical model is X^* with objective value $Z^* = F(X^*)$. The following model can then be solved to generate an alternative solution that is maximally different from X^* :

$$\begin{aligned} \text{Max } \Delta &= \sum_i | X_i - X_i^* | \\ \text{Subject to: } & X \in D \\ & / F(X) - Z^* / \leq T \end{aligned}$$

where Δ represents some difference function (shown as an absolute difference in this instance) and T is a tolerance target specified in relation 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 for alternative solutions. The 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. In this approach, subpopulations within the algorithm’s overall population are established as the Fireflies collectively evolve toward different local optima within the solution space. Each desired solution alternative undergoes the common search procedure of the FA. 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.

IV. CONCURRENT CO-EVOLUTIONARY COMPUTATIONAL ALGORITHM FOR MGA

A direct approach to generate alternatives with the SO algorithm would be to iteratively solve the maximum difference model by incrementally updating the target T whenever a new alternative must be produced. This approach would be somewhat similar in scope to the original Hop, Skip, and Jump (HSJ) MGA algorithm [13] in which an initial problem formulation is optimized and then supplementary alternatives are generated by systematically adjusting the target constraint to force the creation of suboptimal solutions. While this approach is straightforward, it would require repeated execution of the optimization algorithm, which could prove computationally expensive [7].

The new MGA procedure is designed to concurrently generate a small number of good but maximally different alternatives in a single run of the FA procedure (i.e. the same

number of runs as if FA were used solely for function optimization purposes) and its efficiency is based upon the concept of co-evolution. In this co-evolutionary approach, pre-specified stratified subpopulation ranges within the FA algorithm's overall population are established that collectively evolve the search toward the formation of the stipulated number of very different solution alternatives. Each desired solution alternative is represented by each respective subpopulation and each subpopulation undergoes the common operations of the FA. This approach can be structured upon any standard FA solution procedure containing appropriate encodings and operators that best correspond to the problem. However, the survival of solutions in each subpopulation depends upon how well the solutions perform with respect to both the modelled objective(s) and by how far away they are from all of the other solutions in the decision space. Thus, the evolution of solutions in each subpopulation is directly influenced by those solutions contained in all of the other subpopulations, which forces the co-evolution of each subpopulation towards good but maximally distant regions of the decision space. This co-evolutionary concept enables the simultaneous search for, and production of, a set of quantifiably good solutions that are maximally different from each other [7].

By using the 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, but superior, to those created by an iterative HSJ-styled approach. While each alternative produced by an HSJ procedure is maximally different only from the single, overall optimal solution together with an objective value which is at least $x\%$ different from the best objective (i.e. $x = 1\%, 2\%$, etc.), the new co-evolutionary 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 is produced in terms of the solution structure of their decision variables. Co-evolution is also a much more efficient procedure than HSJ in that it exploits the population-based searches of FA algorithms in order to generate the multiple maximally different solution alternatives concurrently. Namely, while an HSJ-styled approach would be required to run n different times in order to generate n different alternatives, the new algorithm need be run only a single time to produce its entire set of alternatives irrespective of the value of n . Hence, it is a much more computationally efficient process.

The steps in the co-evolutionary algorithm are as follows:

1. Create an initial population stratified into P equally-sized subpopulations. The value for P must be established *a priori* by the decision-maker. P represents the desired number of maximally different alternative solutions within a prescribed target deviation from the optimal to be generated. S_p represents the p^{th} subpopulation set of solutions, $p = 1, \dots, P$ and there are K solutions contained within each S_p .

2. Evaluate all solutions in S_p , $p = 1, \dots, P$, with respect to the modelled objective. Solutions meeting the target constraint and all other problem constraints are designated as *feasible*, while all other solutions are designated as *infeasible*.

3. Apply an appropriate elitism operator to each S_p to preserve the best individual in each subpopulation. In S_p , $p = 1, \dots, P$, the best solution is the feasible solution most distant in decision space from all of the other subpopulations (the distance measure is defined in Step 6). Note: Because the best solution to date is always placed into each subpopulation, at least one solution in S_p will always be feasible.

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.

5. Identify the decision space centroid, C_{ip} , for each of the $K' \leq K$ feasible solutions within $k = 1, \dots, K$ of S_p , for each of the N decision variables X_{ikp} , $i = 1, \dots, N$. Each centroid represents the N -dimensional centre of mass for the solutions in each of the respective subpopulations, p . As an illustrative example for determining a centroid, calculate $C_{ip} = (1/K') * \sum_k X_{ikp}$. In this calculation, each dimension of each centroid is computed as the straightforward average value of that decision variable over all of the values for that variable within the feasible solutions of the respective subpopulation. Alternatively, a centroid could be calculated as some fitness-weighted average or by some other appropriately defined measure.

6. For each solution $k = 1, \dots, K$, in each S_q , calculate D_{kq} , a distance measure between that solution and all other subpopulations. As an illustrative example for determining a distance measure, calculate $D_{kq} = \text{Min} \{ \sum_i |X_{ikp} - C_{ip}| ; p = 1, \dots, P, p \neq q \}$. This distance represents the minimum distance between solution k in subpopulation q and the centroids of all other subpopulations. Alternatively, the distance measure could be calculated by some other appropriately defined measure.

7. Rank the solutions within each S_p according to the distance measure D_{kq} objective – appropriately adjusted to incorporate any constraint violation penalties. The goal of maximal difference is to force solutions from one subpopulation to be as far apart as possible in the decision space from the solutions of each of the other subpopulations. This step orders the specific solutions in each subpopulation by those solutions which are most distant from the solutions in all of the other subpopulations.

8. In each S_p , apply appropriate FA “change operations” to the solutions and return to Step 2.

V. COMPUTATIONAL TESTING OF THE FIREFLY ALGORITHM USED FOR MGA

As described previously, planners 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 efficacy of the co-evolutionary MGA procedure to concurrently produce such maximally different alternatives will be illustrated using a two-dimensional, multimodal optimization problem taken from [5]. The mathematical formulation for the multimodal test problem is:

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

$$0.0 \leq x \leq 1.0$$

$$0.0 \leq y \leq 1.0$$

The 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 $(x, y) = (0.974, 0.974)$ [5].

In order to create the alternatives, it would be possible to incrementally place extra target constraints into the original mathematical formulation which would force the generation of solutions that were structurally different from the initial optimal solution. Suppose for example that ten additional solution options were to be created through the inclusion of a technical constraint that decreased the objective of the original optimal model from 1% up to 10% in increments of 1%. By adding these incremental target constraints to the original model and sequentially resolving the problem 10 times, it would be possible to create the prescribed number of maximally different alternatives. However, to improve upon the process of running ten separate additional instances of the algorithm, the co-evolutionary MGA procedure could be run exactly once to concurrently produce all of the desired alternatives. By employing the co-evolutionary MGA algorithm, the optimal solution and the 10 maximally different solutions shown in Table 1 were generated.

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

Increment	1% Increment Between Alternatives			2.5% Increment Between Alternatives		
	$F(x,y)$	x	y	$F(x,y)$	x	y
Optimal	5.14	0.98	0.97	5.14	0.98	0.97
Alternative 1	5.10	0.97	0.97	5.01	0.87	0.87
Alternative 2	5.05	0.87	0.98	4.89	0.66	0.87
Alternative 3	5.00	0.76	0.98	4.77	0.87	0.45
Alternative 4	4.99	0.98	0.87	4.65	0.33	0.97
Alternative 5	4.91	0.98	0.76	4.50	0.98	0.02
Alternative 6	4.89	0.55	0.97	4.43	0.02	0.98
Alternative 7	4.89	0.98	0.55	4.29	0.98	0.02
Alternative 8	4.74	0.34	0.98	4.13	0.02	0.99
Alternative 9	4.69	0.98	0.24	4.02	0.99	0.02
Altern. 10	4.64	0.13	0.98	3.87	0.01	0.98

In totality, this section underscore several important findings with respect to the concurrent co-evolutionary MGA procedure: (i) Co-evolution can be used 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) The alternatives generated are good for planning purposes since all of their structures are as mutually and maximally different from each

other as possible (i.e. these differences are not just simply different from the overall optimal solution as in the HSJ-style approach to MGA); (iv) The MGA procedure is computationally very efficient since it need only be run once to generate its entire set of multiple, good solution alternatives (i.e. to generate n solution alternatives, the MGA algorithm needs to run exactly the same number of times that the FA would need to be run for function optimization purposes alone – namely once – irrespective of the value of n); and, (v) The best overall solutions produced by the MGA procedure will be very similar, if not identical, to the best overall solutions that would be produced by the FA for function optimization alone.

As described earlier, many engineering optimization applications tend to be riddled with incongruent performance requirements that contain significant performance requirements that are exceedingly difficult to quantify. Consequently, it is preferable to create several 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, thereby incorporating the unmodelled issues into the actual solution process. The example demonstrated how the co-evolutionary MGA modelling perspective could be used to concurrently generate multiple alternatives via the very computationally efficient FA that satisfy required system performance criteria according to 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 MGA approach simultaneously performs extremely well with respect to its role in function optimization. It should be explicitly noted that the overall best solutions produced by the MGA procedure for the test problem are indistinguishable from the one determined in [5].

VI. CONCLUSIONS

Engineering optimization problems and “real world” engineering decision-making usually contain multifaceted performance requirements that can be riddled with unquantifiable specifications and incompatible performance objectives. Such problems typically possess competing design requirements which are very difficult – if not impossible – to quantify and capture at the time that any supporting decision models are 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. This multitude of uncertain and competing dimensions forces decision-makers to integrate many conflicting sources into their decision process prior to final solution construction. Under the presence of so much uncertainty, it becomes unlikely that any single solution could ever be constructed that simultaneously satisfies all of the incongruent system requirements without a significant counterbalancing of the numerous tradeoffs involved. Therefore, any ancillary modelling techniques used to support the decision formulation process must somehow simultaneously account for all of these features while being

flexible enough to encapsulate the impacts from the inherent planning uncertainty.

In this paper, a co-evolutionary MGA procedure was introduced that demonstrated how the very computationally efficient FA could be used to concurrently generate multiple, maximally different, near-best alternatives via co-evolution. In this MGA capacity, the co-evolutionary procedure produces numerous solutions possessing the requisite problem characteristics, with each generated alternative providing a very different perspective. Since FA techniques can be adapted to solve a wide variety of problem types, the practicality of this co-evolutionary MGA approach can clearly be extended into numerous disparate engineering optimization and design applications. These extensions will be the subject of future studies.

REFERENCES

- [1] M. Brugnach, A. Tagg, F. Keil, and W.J. De Lange, "Uncertainty matters: computer models at the science-policy interface," *Water Resour Manag*, vol. 21, pp. 1075-1090, 2007.
- [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," *Environ. Sci. Policy*, vol.13, no. 3, pp. 229-238, 2010.
- [3] H.T. Mowrer, "Uncertainty in natural resource decision support systems: Sources, interpretation and importance," *Comput. Electron. in Agric.*, vol. 27, no. 1-3, pp. 139-154, 2000.
- [4] W.E. Walker, P. Harremoes, J. Rotmans, J.P. Van der Sluis, M.B.A. Van Asselt, P. Janssen, and M.P. Krayer von Krauss, "Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support", *Integrated Assessment*, vol. 4, no. 1, pp. 5-17, 2003.
- [5] D.H. Loughlin, S.R. Ranjithan, E.D. Brill, and J.W. Baugh, "Genetic Algorithm Approaches for Addressing Unmodeled Objectives in Optimization Problems," *Eng. Optimiz.* vol. 33, no. 5, pp. 549-569, 2001.
- [6] M. Matthies, C. Giupponi, and B. Ostendorf. "Environmental decision support systems: Current issues, methods and tools," *Environ. Modell. Softw*, vol. 22, no. 2, pp. 123-127, 2007.
- [7] J.S. Yeomans, and Y. Gunalay, "Simulation-Optimization Techniques for Modelling to Generate Alternatives in Waste Management Planning," *J. Appl. Oper. Res.*, vol. 3, no. 1, pp. 23-35, 2011.
- [8] X.S. Yang, "Firefly Algorithms for Multimodal Optimization," *Lect. Notes Comp. Sci.*, vol. 5792, pp. 169-178, 2009.
- [9] X.S. Yang, *Nature-Inspired Metaheuristic Algorithms 2nd Ed.*, Frome (UK): Luniver Press, 2010.
- [10] L.C. Cagnina, C.A. Esquivel, and C.A. Coello, "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer," *Informatica*, vol. 32, pp. 319-326, 2008.
- [11] R. Imanirad, X.S. Yang, and J.S. Yeomans, "A Computationally Efficient, Biologically-Inspired Modelling-to-Generate-Alternatives Method," *J. on Comput.*, vol. 2, no. 2, pp. 43-47, 2012.
- [12] J.W. Baugh, S.C. Caldwell, and E.D. Brill, "A Mathematical Programming Approach for Generating Alternatives in Discrete Structural Optimization," *Eng. Optimiz.*, vol. 28, no. 1, pp. 1-31, 1997.
- [13] 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," *Manage. Sci.*, vol. 28, no. 3, pp. 221-235, 1982.
- [14] E.M. Zechman, and S.R. Ranjithan, "An Evolutionary Algorithm to Generate Alternatives (EAGA) for Engineering Optimization Problems," *Eng. Optimiz.*, vol. 36, no. 5, pp. 539-553, 2004.



Raha Imanirad is a Research Scientist in the Operations Management and Information Systems Area of the Schulich School of Business at York University, Toronto. She holds degrees in Computer Science and Business from York University.



Xin-She Yang received his DPhil in Applied Mathematics from the University of Oxford, worked at Cambridge University for several years, and is currently a Senior Research Scientist at National Physical Laboratory. He is also a Distinguished Professor of Shaanxi Province at Xi'an Engineering University, and was the recipient of 1996 Garside Scholar Award at Oxford. He has authored/edited a dozen books and published more than 140 papers. Now he is the Editor-in-Chief of International Journal of Mathematical Modelling and Numerical Optimisation.



Julian Scott Yeomans is a Professor in the Operations Management and Information Systems Area at the Schulich School of Business at York University, Toronto. He holds graduate degrees in Management Science & Information Systems from McMaster University and in Environmental Engineering from the University of Toronto together with undergraduate degrees in Mathematics and in Business from the University of Regina.