A Concurrent Modelling to Generate Alternatives Approach Using the Firefly Algorithm

Raha Imanirad, Department of Operations Management and Information Systems (OMIS) Area, Schulich School of Business, York University, Toronto, ON, Canada

Xin-She Yang, Department of Mathematics & Scientific Computing, National Physical Laboratory, Teddington, UK

Julian Scott Yeomans, Department of Operations Management and Information Systems (OMIS) Area, Schulich School of Business, York University, Toronto, ON, Canada

ABSTRACT

Real world” decision-making applications generally contain multifaceted performance requirements riddled with incongruent performance specifications. There are invariably unmodelled elements, not apparent during model construction, which can greatly impact the acceptability of the model’s solutions. Consequently, it is preferable to generate numerous alternatives that provide dissimilar approaches to the problem. These alternatives should possess near-optimal objective measures with respect to all known objective(s), but be maximally different from each other in terms of their decision variables. This maximally different solution creation approach is referred to as modelling-to-generate-alternatives (MGA). This study demonstrates how the Firefly Algorithm can concurrently create multiple solution alternatives that both satisfy required system performance criteria and yet are maximally different in their decision spaces. This new approach is 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.

Keywords: Biologically-Inspired Metaheuristic Algorithms, Firefly Algorithm, Modelling-to-Generate-Alternatives (MGA), Real World Decision-Making, System Performance

1. INTRODUCTION

Typical “real world” decision-making situations involve complex problems that possess design requirements which are very difficult to incorporate into their supporting mathematical programming formulations and tend to be plagued by numerous unquantifiable components (Brugnach et al., 2007; Janssen et al., 2010; Mower, 2000; Walker et al., 2003). 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 (Brugnach et al.,...
2007; Janssen et al., 2010; Loughlin et al., 2001). Hence, it is generally considered desirable to generate a reasonable number of very different alternatives that provide multiple, contrasting perspectives to the specified problem (Mathies et al., 2007; Yeomans & Gunalay, 2011). 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 in response to this multi-solution creation requirement (Brill et al., 1982; Loughlin et al., 2001; Yeomans & Gunalay, 2011).

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 as fundamentally different as possible from each other within the prescribed decision space. The resulting set of alternatives should provide numerous solutions that all perform somewhat similarly with respect to the modelled objectives, yet very differently with respect to the unmodelled issues (Walker et al., 2003). Obviously the decision-makers must then conduct a subsequent comprehensive comparison of these alternatives to determine which options would most closely satisfy their very specific circumstances. Consequently, MGA approaches should necessarily be classified as a decision support processes rather than the role of explicit solution determination methods assumed, in general, for optimization.

Previous MGA methods have employed direct, iterative processes for generating alternatives by incrementally re-running their solution algorithms whenever new alternatives must be produced (Baugh et al., 1997; Brill et al., 1982; Loughlin et al., 2001; Yeomans & Gunalay, 2011; Zechman & Ranjithan, 2004). These iterative approaches follow the seminal MGA approach of Brill et al. (1982) in which, once an initial problem formulation has been optimized, the supplementary alternatives are created one-by-one. Consequently, these iterative approaches all require $n+1$ runnings of their respective algorithms to optimize the initial problem and to subsequently create their $n$ alternatives (Imanirad & Yeomans, 2012; Imanirad et al., 2012; Yeomans & Gunalay, 2011).

In this paper, it is shown how to efficiently generate a set of maximally different solution alternatives by implementing a modified version of the nature-inspired Firefly Algorithm (FA) of Yang (2009, 2010) combined with a new concurrent, co-evolutionary MGA approach. For calculation and optimization purposes, Yang (2010) has demonstrated that the FA is more computationally efficient than such commonly-used metaheuristic procedures as genetic algorithms, simulated annealing, and enhanced particle swarm optimization (Cagnina et al., 2008; Gandomi et al., 2011). However, what differentiates the FA from other population-based metaheuristics for functional optimization purposes, is that it has been specifically designed to simultaneously converge into a specified number of local optima (including the global ones) in highly non-linear mathematical programming problems. Recently, Imanirad and Yeomans (2012) have demonstrated how the FA’s functional optimization capabilities to determine multiple local optima can be modified to produce the $n$ maximally different alternatives required in an MGA approach after the initial optimal solution has been determined.

The MGA procedure provided in this study extends the earlier approaches of Imanirad et al. (2012) and Imanirad and Yeomans (2012) byexploiting the concept of co-evolution within the FA’s solution approach in order to concurrently generate the optimal solution together the desired number of maximally different alternatives to it. Remarkably, this novel approach simultaneously produces the overall best solution together with $n$ locally optimal, maximally different alternatives in a single computational run. Namely, to generate the additional $n$ maximally different solution alternatives, the MGA algorithm would need to run exactly the same number of times that the FA would need to be run for function optimization purposes alone (i.e. once) irrespective
A Logit Model for Budget Allocation Subject to Multi Budget Sources
www.igi-global.com/chapter/logit-model-budget-allocation-subject/70964?camid=4v1a