Chapter 2
Simultaneous Modelling-to-Generate-Alternatives Procedure Employing the Firefly Algorithm

Julian Scott Yeomans
York University, Canada

ABSTRACT

"Real-world" decision-making applications generally contain multifaceted performance requirements riddled with incongruent performance specifications. 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 decision models are actually constructed. 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 several distinct alternatives that provide multiple disparate perspectives 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 chapter provides an efficient optimization algorithm that simultaneously generates multiple, maximally different alternatives by employing the metaheuristic firefly algorithm. The efficacy of this mathematical programming approach is demonstrated on a commonly tested engineering optimization benchmark problem.

DOI: 10.4018/978-1-5225-6164-4.ch002

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 (Belarbi *et al*., 2017; Matallah *et al*., 2017; Brugnach *et al*., 2007; Janssen *et al*., 2010; Junejah *et al*., 2017; Matthies *et al*., 2007; Mowrer, 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 (Acharjya & Anitha, 2017; 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 (Matthies *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 (Arrais-Castro *et al*., 2015). 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 (see, also: Benatia *et al*., 2016; Sharma & Virmani, 2017; Strand *et al*., 2017).

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, 2013; Imanirad *et al*., 2012a; Yeomans & Gunalay, 2011).

For calculation and optimization purposes, Yang (2009, 2010) has demonstrated that the nature-inspired Firefly Algorithm (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 (see, also: Arun *et al*., 2017; Dekhici *et al*., 2015; Dey *et al*., 2014; Jagatheesan *et al*., 2017; Kumar *et al*., 2017). Imanirad & Yeomans (2013) have demonstrated how the FA's functional optimization capabilities to determine multiple local optima can be modified