ROBUST DECISION SUPPORT METHODS

A COMPARATIVE ANALYSIS

by

ERIN BARTHOLOMEW

Student number:	4627237	
Project duration:	9 February 2018 – 23 August 2018	
Thesis committee:	Prof. dr. A. Verbraeck,	TU Delft, Chair
	Dr. J. Kwakkel,	TU Delft, first supervisor
	Dr. M. Warnier,	TuU Delft, second supervisor

3.1 Multi-Objective Evolutionary Algorithms (MOEAs)

In policy analysis, there are many problems with multiple conflicting objectives and behavior that is complex and powered by deep uncertainty. Because of this, there is no single optimal solution to these problems; instead, decision makers must look for a set of alternatives that are defined as Pareto-optimal, where each solution in a set is not dominated by any other solution in the set. To find the Pareto-optimal set of solutions, analysis requires a search algorithm that is capable of handling these issues. The most common solution to this is what's known as a multi-objective evolutionary algorithm (Maier et al., 2014; Reed, Hadka, Herman, Kasprzyk, & Kollat, 2013). A multi-objective evolutionary algorithm (MOEA) is one that takes a generic approach to optimizing, in which a population of potentially optimal items is generated through an iterative process of the following steps:

Begin:

0. A set of options is randomly determined, the fitness of of which is determined based on the case-specific definition provided.

Iterative Steps:

1. The most fit options are used to generate new options for consideration - the next generation of alternatives.
2. Fitness of those new options is determined using the same definition as in 0.
3. The options with the lowest fitness in the existing set are replaced with items of higher fitness in the newly discovered set.

Several MOEAs have been developed over the years, each of which aims to discover potential solutions to multi-objective problems. One of the earlier instances, NSGAII, is a generational algorithm that was developed by Deb, Pratap, Agarwal, and Meyarivan (2002). NSGAII uses a constant population size for each generation, and was one of the earliest to use Pareto dominance to search for and rank alternative solutions to a stated problem, resulting in a set of non-dominated alternatives (Reed et al., 2013). The goal of this search process is to find a set of non-dominated solutions that make up the Pareto optimal front, which would precisely describe the complex tradeoffs between conflicting objectives. Because of uncertain behavior and conflicting objectives, though, it is impossible to reach the optimal front precisely. Therefore, the search determines what is known as an approximation of the Pareto front. To facilitate the search process, NSGAII makes use of a single operator that is responsible for maintaining diversity of solutions (Reed et al., 2013; Ward, Singh, Reed, & Keller, 2015). The NSGAII algorithm has had solid performance in many optimization problems and is therefore still commonly used today (Zheng, Zecchin, Maier, & Simpson, 2016).

The NSGAII algorithm was extended to create ϵ-NSGAII by incorporating epsilon dominance in the sorting process and by using adaptive population sizing across generations. Together, these two characteristics have shown to reduce the need for extensive calibration of input parameters and to include more efficient search process (Ward et al., 2015). Epsilon dominance enables the decision maker to specify her desired level of precision for a policy to be considered dominant for each objective value under consideration. With this property, solutions are only considered dominant if they fall outside the space defined by the epsilon value of an objective (Horoba & Neumann, 2008). This allows for the elimination of alternatives that are considered similar and encourages diversity in the final set of recommended options (Reed et al., 2013). Along with epsilon dominance, the adaptive population sizing ability allows for a more efficient search to be completed: the number of alternatives being tested begin small to reduce computational cost, until fit solutions have been discovered; at that time, population size increases to put more pressure on the fitness and selection process, which ensures that the most fit solutions are being found within a specific generation (Ward et al., 2015). Other traditionally structured MOEA algorithms include SPEA2, ϵ-MOEA, OMOPSO, MOEA/D, and GDE3 (Reed et al., 2013; Ward et al., 2015; Zheng et al., 2016). Each of these algorithms use a constant operator and some form of Pareto dominance to find solutions and maintain diversity among those solutions.

An alternative to these traditional algorithms is a hybrid-MOEA known as Borg, which is built on the ϵ-MOEA algorithm, pairs the ideas of adaptive population sizing and epsilon dominance with auto-adaptive operator selection and randomized algorithm restarts to efficiently find well-performing solution alternatives (Hadka & Reed, 2013). By using auto-adaptive operator selection, as opposed to the single set of operators used in the previously described methods, Borg is able to select operators based on their ability to select highly fit policies, leading to a more efficient and effective search process. Because it is built on the ϵ-MOEA algorithm, Borg is a steady-state and not a generational MOEA. This means that a newly discovered alternative must only compete with and potentially replace one specific alternative in that set. This is in contrast to generational algorithms like NSGAII, which build an entirely new set of non-dominated alternatives is constructed after each iteration by comparing the existing set of alternatives with the newly generated set as a whole. The randomized restart function enables Borg, upon detecting that the search has stalled, to inject new and diverse alternatives into the search process, ensuring the best chance for a diverse and fit set of alternatives once the algorithm completes its run (Reed et al., 2013). The random restart function, along with adaptive population sizing and ϵ-dominance sorting, were inspired by the work done to develop the ϵ-NSGAII algorithm.

Multiple studies have been completed that compare the success or failure of these MOEAs given different policy problems. These studies have generally concluded that the auto-adaptive operator selection and adaptive population sizing together allow Borg to discover a more robust and diverse set of alternatives than the other methods (Reed et al., 2013; Ward et al., 2015; Zheng et al., 2016). Other algorithms that achieved at least some success are ϵ-MOEA, NSGAII, and ϵ-NSGAII, but none of these algorithms provide results at the level of Borg.

3.1.1 Auto-adaptive NSGAII

While Borg has been discovered to be a successful algorithm to find a strong approximation of a problem's Pareto optimal front, it does so by leveraging a steady-state genetic algorithm. In this type of algorithm, only one new alternative is generated and folded into the existing non-dominated set at a time. This requires strategies to both select the desired parents from the existing set of alternatives and to prescriptively replace solutions in the existing set, which can significantly impact the outcome of the search process (Vavak & Fogarty, 1996), leading to the potential for additional uncertainty in analysis of a problem already fraught with uncertainty. Slow replacement of the non-dominated set can also contribute to slower convergence to the Pareto-optimal front (Vavak & Fogarty, 1996). In contrast, a genetic algorithm uses a much larger portion of the existing set of non-dominated alternatives in each iteration to generate a new set of alternatives. These two sets are then compared to build an entirely new set of alternatives at the end of each iteration. This can lead to a faster convergence toward the Pareto optimal front (Vavak & Fogarty, 1996).

This research proposes an advancement of ϵ-NSGAII to combine the strongest attributes of Borg: the auto-adaptive operator selection and adaptive population sizing, with the best attributes of ϵ-NSGAII, epsilon-dominance and a generational algorithm with fast convergence to the Pareto optimal front. Instead of using a single operator as in the traditional ϵ-NSGAII algorithm, auto-adaptive NSGAII uses a selection process similar to Borg to select an operator out of the same group of operators that Borg has implemented. Each operator is assigned a probability of selection, based on the number of solutions that an operator has produced that ends up in the non-dominated set of alternatives. Operator elements and their parameter settings are indicated in Table 3.1, and the complete operator definitions are listed below.

- Simulated Binary Crossover (SBX) variator + Polynomial Mutation (PM) mutator
- Parent-centric Crossover (PCX) variator + PM mutator
- Differential Evolution (DE) variator + PM mutator
- Unimodal Normal Distribution Crossover (UNDX) variator + PM mutator
- Simplex Crossover (SPX) variator + PM mutator

Table 3.1: A sumary of the configuration details for the variant and mutator elements of the operators used in the proposed auto-adaptive NSGAII algorithm (Hadka & Reed, 2013)

Name	Property name	Value
PM	probability	1.0 / L
	distribution index	20
SBX	probability	1.0
	distribution index	15
PCX	nparents	3
	noffspring	2
	eta	0.1
	Zeta	0.1
DE	crossover rate	0.1
	step size	0.6
UNDX	nparents	3
	noffspring	2
	zeta	0.5
	eta	0.35/sqrt(L)
SPX	nparents	L+1
	noffspring	L+1
	expansion	sqrt((L+1)+1)
UM	probability	1.0/L

– Uniform mutation (UM) mutator

This algorithm is implemented using the existing functions from the Python-based Platypus optimization package, which is a clone of the Java-based MOEAFramework library. In addition to having auto-adaptive operator selection in a generational algorithm, this generational algorithm is easily parallelizable, as each iteration of the algorithm produces an entirely new population with which to compare. In addition, because the auto-adaptive NSGAII algorithm leverages existing open-source code, it is freely available for anyone to leverage in their own analysis. As of the time of this thesis, Borg requires a paid license or special dispensation to use.

3.2 Nondominated Selection of Alternatives