Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework

David Hadka[*] dmh309@psu.edu
Department of Computer Science and Engineering, The Pennsylvania State University, University Park, Pennsylvania 16802

Patrick Reed pmr11@engr.psu.edu
Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, Pennsylvania 16802

Abstract

This study introduces the Borg multi-objective evolutionary algorithm (MOEA) for many-objective, multimodal optimization. The Borg MOEA combines ϵ-dominance, a measure of convergence speed named ϵ-progress, randomized restarts, and auto-adaptive multioperator recombination into a unified optimization framework. A comparative study on 33 instances of 18 test problems from the DTLZ, WFG, and CEC 2009 test suites demonstrates Borg meets or exceeds six state of the art MOEAs on the majority of the tested problems. The performance for each test problem is evaluated using a 1,000 point Latin hypercube sampling of each algorithm's feasible parameterization space. The statistical performance of every sampled MOEA parameterization is evaluated using 50 replicate random seed trials. The Borg MOEA is not a single algorithm; instead it represents a class of algorithms whose operators are adaptively selected based on the problem. The adaptive discovery of key operators is of particular importance for benchmarking how variation operators enhance search for complex many-objective problems.

Keywords

Evolutionary algorithm, multi-objective optimization, many-objective optimization, multimodal problems, ϵ-dominance.

1 Introduction

Multi-objective evolutionary algorithms (MOEAs) are a class of optimization algorithms inspired by the processes of natural evolution (Holland, 1975). Over the past 20 years, researchers have successfully applied MOEAs to a large array of problems from industrial, electrical, computer, civil, and environmental engineering; aeronautics; finance; chemistry; medicine; physics; and computer science (Coello Coello et al., 2007). However, such studies have traditionally concentrated on problems involving two or three objectives. With burgeoning computing power and an increasing acceptance of MOEAs as multi-objective optimizers, MOEAs are beginning to address many-objective problems with four or more objectives (Fleming et al., 2005; Coello Coello et al., 2007; Ferringer et al., 2009; Kasprzyk et al., 2009).

While many-objective applications are growing in their success, there exists strong theoretical and experimental evidence suggesting that existing approaches are

[*] Corresponding Author.

insufficient for many-objective problems. Farina and Amato (2004), Fleming et al. (2005), and Purshouse and Fleming (2007) observe that the proportion of locally Pareto nondominated solutions tends to become large as the number of objectives increases. This is a direct result of Pareto dominance and its aim to capture, without preference, the entire trade-off surface between two or more objectives. This leads to difficulties in producing offspring that dominate poorly performing, but still nondominated, members in the population—a phenomenon termed dominance resistance (Hanne, 2001; Ikeda et al., 2001; Purshouse and Fleming, 2007).

This increasing proportion of locally Pareto nondominated solutions and the phenomenon of dominance resistance can impact the performance of MOEAs in several ways. First, these conditions may limit the ability of dominance relations in differentiating high-quality and low-quality solutions. Several researchers have proposed alternate dominance relations to provide more stringent dominance criteria, including the preferability (Fonseca and Fleming, 1998), preferred (Drechsler et al., 2001), ϵ-preferred (Sülflow et al., 2007), k-optimality (Farina and Amato, 2004), and preference order ranking (di Pierro et al., 2007) dominance relations. One must, however, be aware of the impact of selecting a different dominance relation, as they may focus search toward a subspace and fail to produce solutions along the entire extent of the trade-off surface (Coello Coello et al., 2007).

Second, as the proportion of locally Pareto nondominated solutions increases and the offspring are likely to also be nondominated as a result of dominance resistance, it is often difficult for an MOEA to identify which offspring should survive and replace existing members in the population. In such scenarios, the diversity operator, such as crowding, is often the primary mechanism for determining survival. This phenomenon is termed active diversity maintenance (Purshouse and Fleming, 2007).

Third, Hanne (1999) observed that active diversity maintenance can cause deterioration. Deterioration occurs whenever the solution set discovered by an MOEA at time i contains one or more solutions dominated by a solution discovered at some earlier point in time $j < i$. In the extreme, deterioration can cause an MOEA to diverge away from the Pareto front. Laumanns et al. (2002) effectively eliminate deterioration with the ϵ-dominance relation; however, at present, most state of the art MOEAs in use today have yet to adopt mechanisms for avoiding deterioration.

Lastly, Hadka and Reed (2012) show empirically on several MOEAs that parameterization can greatly impact the performance of an MOEA, and for many top-performing algorithms, this issue becomes severely challenging as the number of objectives increases. In addition, they demonstrate that most modern MOEAs can fail in terms of both convergence and reliability on test problems with as few as four objectives. These results are backed by the theoretical work of Teytaud (2006, 2007), which show that dominance resistance can cause the convergence rate of MOEAs to degrade to be no better than random search for problems with 10 or more objectives, and the experimental work of Ishibuchi et al. (2008), where it is also demonstrated that several state of the art MOEAs fail on problems with as few as four objectives.

This study introduces a novel search framework, called the Borg MOEA, designed to operate on many-objective, multimodal problems. Borg features an ϵ-dominance archive with auto-adaptive operators that detect search stagnation, exploit randomized restarts to escape local optima, and select recombination operators based on their success in generating high quality solutions. Using a suite of many-objective test problems, we demonstrate the ability of Borg to match or outperform six state of the art MOEAs. These top-ranked MOEAs, which are introduced in detail in Section 2, provide a rigorous performance baseline for distinguishing Borg's contributions.

The remainder of this paper is organized as follows. Section 2 provides the background material and definitions used in the remainder of this paper. Section 3 presents the Borg MOEA and discusses its theoretical convergence behavior. The results of a comparative study between Borg and other top-performing MOEAs is presented and analyzed in Section 4. Lastly, the impact of the Borg MOEA and future work are discussed in Section 5.

2 Background

A multi-objective problem with M objectives is defined as

$$\begin{aligned} \underset{\mathbf{x}\in\Omega}{\text{minimize}} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0,\ \forall i \in \mathcal{E}, \\ & c_j(\mathbf{x}) \leq 0,\ \forall j \in \mathcal{I}. \end{aligned} \tag{1}$$

We call $\mathbf{x}$ the decision variables and Ω the decision space. In this study, we consider only real-valued decision variables $\mathbf{x} = (x_1, x_2, \ldots, x_L)$, $x_i \in \Re$, of fixed length L. The sets $\mathcal{E}$ and $\mathcal{I}$ contain the indices for all equality and inequality constraints, respectively. The feasible region, Λ, is the set of all decision variables in Ω that satisfy all constraints. In this study, we consider only unconstrained problems ($\mathcal{E} = \mathcal{I} = \emptyset$ and $\Lambda = \Omega$).

In this study, we consider the form of optimization known as a posteriori optimization, in which search precedes the decision making process (Coello Coello et al., 2007). The search algorithm generates a set of potential solutions allowing a decision maker to explore the various trade-offs and identify the preferred solution(s). This form of optimization is particularly useful when weights or preferences are not known a priori. The notion of optimality when trade-offs exist between solutions is captured by Pareto dominance and the Pareto optimal set.

DEFINITION 1: *A vector* $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ *Pareto dominates another vector* $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ *if and only if* $\forall i \in \{1, 2, \ldots, M\}, u_i \leq v_i$ *and* $\exists j \in \{1, 2, \ldots, M\}, u_j < v_j$. *This is denoted by* $\mathbf{u} \prec \mathbf{v}$.

DEFINITION 2: *For a given multi-objective problem, the Pareto optimal set is defined by*

$$\mathcal{P}^* = \{\mathbf{x} \in \Lambda \mid \neg\exists \mathbf{x}' \in \Lambda, F(\mathbf{x}') \prec F(\mathbf{x})\}$$

DEFINITION 3: *For a given multi-objective problem with Pareto optimal set* $\mathcal{P}^*$, *the Pareto front is defined by*

$$\mathcal{PF}^* = \{F(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}^*\}$$

In MOEAs, the Pareto dominance relation is applied to the objectives. For convenience, we use $\mathbf{x} \prec \mathbf{y}$ interchangeably with $F(\mathbf{x}) \prec F(\mathbf{y})$. Two solutions are nondominated if neither Pareto dominates the other. Using this terminology, the goal of a posteriori optimization is to capture or closely approximate the Pareto front.

Schaffer (1984) introduced the vector evaluated genetic algorithm (VEGA), which is generally considered the first MOEA to search for multiple Pareto-optimal solutions in a single run. VEGA was found to have problems similar to aggregation-based approaches, such as an inability to generate concave regions of the Pareto front. Goldberg (1989a) suggested the use of Pareto-based selection, but this concept was not applied until 1993 in the multi-objective genetic algorithm (MOGA) by Fonseca and Fleming (1993).

In subsequent years, several popular MOEAs with Pareto-based selection were published, including the niched-Pareto genetic algorithm (NPGA) by Horn and Nafpliotis (1993) and the nondominated sorting genetic algorithm (NSGA) by Srinivas and Deb (1994). These foundational algorithms established the utility of MOEAs for solving multi-objective problems. Between 1993 and 2003, the importance of elitism, diversity maintenance, and external archiving was demonstrated in various studies through the introduction of new MOEAs. Algorithms developed during this time are commonly referred to as first-generation. Notable first-generation algorithms include the strength Pareto evolutionary algorithm (SPEA) by Zitzler and Thiele (1999), the Pareto-envelope based selection algorithm (PESA) by Corne and Knowles (2000), and the Pareto archived evolution strategy (PAES) by Knowles and Corne (1999). For a more comprehensive overview of the historical development of MOEAs, please refer to the text by Coello Coello et al. (2007).

Since 2003, a large number of MOEAs have been developed in the literature. Particular interest has been devoted to researching and developing MOEAs for addressing many-objective problems. The following sections provide a broad overview of the various strategies proposed in the literature, and also highlight the many-objective algorithms selected for comparison in this study.

2.1 Indicator-Based Methods

Indicator-based methods replace the Pareto dominance relation with an indicator function intended to guide search toward regions of interest (Ishibuchi et al., 2010). The hypervolume measure is often used as the indicator function due to its theoretical characteristics (Ishibuchi et al., 2010). Hypervolume-based methods avoid active diversity maintenance by not using an explicit diversity-preserving mechanism, and instead promote diversity through the hypervolume measure itself (Wagner et al., 2007). One potential downfall to hypervolume-based methods is the computational complexity of calculating the hypervolume measure on high-dimensional problems, but Ishibuchi et al. (2010) have proposed an approximation method to reduce the computational complexity. The indicator-based evolutionary algorithm (IBEA; Zitzler and Künzli, 2004) is a popular implementation of a hypervolume-based MOEA analyzed in this study.

2.2 Pareto Front Approximation

Issues like deterioration arise when finite population sizes force an MOEA to remove Pareto nondominated solutions (Laumanns et al., 2002). As the proportion of Pareto nondominated solutions increases as the number of objectives increases, the occurrence of deterioration increases. Laumanns et al. (2002) introduced the ϵ-dominance relation as a way to eliminate deterioration by approximating the Pareto front, and also provided theoretical proofs of convergence and diversity for algorithms using this relation. ϵ-MOEA (Deb, Thiele, et al., 2002) and ϵ-NSGA-II (Kollat and Reed, 2006) are two popular algorithms using ϵ-dominance included in this study. ϵ-MOEA is included in this study as it serves as the underlying algorithm for the Borg MOEA, which was selected for its demonstrated success on a number of scalable test problems (Hadka and Reed, 2012). ϵ-NSGA-II has been applied successfully to several real-world many-objective problems, and its use of adaptive population sizing and time continuation inspired several features included in the Borg MOEA (Kollat and Reed, 2006, 2007; Kasprzyk et al., 2009; Ferringer et al., 2009; Kasprzyk et al., 2011; Kollat et al., 2011).

2.3 Space Partitioning and Dimensionality Reduction

Both space partitioning and dimensionality reduction methods attempt to convert many-objective problems into lower-dimensional instances that can be solved effectively using existing MOEAs. Space partitioning methods attempt to emphasize search in lower-dimensional objective spaces by partitioning the M-objective space of the original problem into many disjoint lower-dimensional subspaces, each of which is searched independently (Aguirre and Tanaka, 2009). On the other hand, dimensionality reduction methods attempt to convert the higher-dimensional objective space into a lower-dimensional representation using methods such as principal component analysis (PCA; Saxena and Deb, 2008). While these methods are areas of active research, limitations in the availability of source codes and their emphasis on searching subspaces in Pareto optimal sets eliminated them from consideration in this study.

2.4 Aggregate Functions

Using aggregation functions to convert a multi-objective problem into a single-objective problem have remained popular, but special care must be taken when designing the aggregation function to avoid its potential pitfalls (Coello Coello et al., 2007; Wagner et al., 2007). MOEA/D (Zhang, Liu, et al., 2009) is a recently introduced MOEA that uses aggregate functions, but attempts to avoid such pitfalls by simultaneously solving many single-objective Chebyshev decompositions of the many-objective problem in a single run. Since its introduction, MOEA/D has established itself as a benchmark for new MOEAs by winning the IEEE Congress on Evolutionary Computation (CEC) competition held in 2009 (Zhang and Suganthan, 2009). For this reason, MOEA/D is included in this analysis.

2.5 Other Paradigms

Differential evolution (DE) and particle swarm optimization (PSO) are two nature-inspired metaheuristics whose use in many-objective optimization has been explored in the literature. In particular, GDE3 (Kukkonen and Lampinen, 2005) and OMOPSO (Reyes Sierra and Coello Coello, 2005) are two prominent algorithms from their respective fields and are included in this study. GDE3 (and DE in general) is notable for rotationally invariant operators—they produce offspring independent of the orientation of the fitness landscape—which is important for problems with high degrees of conditional dependence among its decision variables (Iorio and Li, 2008). As the Borg MOEA includes the DE operator, GDE3 provides a good baseline comparison. OMOPSO is notable for being the first multi-objective PSO algorithm to include ϵ-dominance as a means to solve many-objective problems. OMOPSO thus provides a representative baseline from the PSO class of algorithms.

3 The Borg MOEA

As a result of the shortcomings of MOEAs discussed in Section 1, this study introduces the Borg MOEA. Borg is designed specifically for handling many-objective, multimodal problems where our primary future focus will be on advancing severely challenging real-world applications. In order to facilitate these design goals, Borg assimilates several design principles from existing MOEAs and introduces several novel components. These components include the following:

- An ϵ-box dominance archive for maintaining convergence and diversity throughout search.
- ϵ-progress, which is a computationally efficient measure of search progression and stagnation introduced in this study.

- An adaptive population sizing operator based on ϵ-NSGA-II's (Kollat and Reed, 2006) use of time continuation to maintain search diversity and to facilitate escape from local optima.
- Multiple recombination operators to enhance search in a wide assortment of problem domains.
- The steady-state, elitist model of ϵ-MOEA (Deb et al., 2003), which can be easily extended for use on parallel architectures.

Each of these components is discussed individually in Sections 3.1–3.4. Section 3.5 discusses how these individual components are combined to form the Borg MOEA. Section 3.6 analyzes the runtime complexity of the Borg MOEA, Section 3.7 provides a proof of convergence, and Section 3.8 provides recommended parameters settings.

3.1 ϵ-Dominance Archive

As discussed in Section 1, deterioration is a fundamental issue encountered by MOEAs. The dominance resistance encountered in many-objective optimization only serves to exacerbate deterioration. Rudolph (1998) and Rudolph and Agapie (2000) presented a selection strategy for a fixed-size archive that avoids deterioration. However, Laumanns et al. (2002) noted that while their selection strategy guarantees convergence to the true Pareto-optimal front, their approach was unable to guarantee a diverse set of Pareto-optimal solutions. As a result of these observations, Laumanns et al. (2002) developed the ϵ-dominance archive in order to guarantee simultaneous convergence and diversity in MOEAs.

Definition 4: *For a given $\epsilon > 0$, a vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ ϵ-dominates another vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if $\forall i \in \{1, 2, \ldots, M\}, u_i \leq v_i + \epsilon$ and $\exists j \in \{1, 2, \ldots, M\}, u_j < v_j + \epsilon$.*

In addition to the theoretical benefits of guaranteed convergence and diversity, ϵ-dominance provides a minimum resolution which effectively bounds the archive size. This is of practical importance to decision makers, who are able to define ϵ using domain-specific knowledge of their precision goals or computational limits (Kollat and Reed, 2007; Kasprzyk et al., 2009). In practice, it is useful to specify different ϵ values for each objective; however, without loss of generality, we use a single ϵ value to improve the clarity of this paper.

A variant called the ϵ-box dominance archive is used in the ϵ-MOEA and ϵ-NSGA-II algorithms by Deb et al. (2003) and Kollat and Reed (2007), respectively. The ϵ-box dominance relation is defined below and the archive update procedure is outlined in Algorithm 1. The archive update procedure is executed once for every solution generated by the MOEA.

Definition 5: *For a given $\epsilon > 0$, a vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ ϵ-box dominates another vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if one of the following occurs*

1. $\left\lfloor \frac{\mathbf{u}}{\epsilon} \right\rfloor \prec \left\lfloor \frac{\mathbf{v}}{\epsilon} \right\rfloor$, *or*
2. $\left\lfloor \frac{\mathbf{u}}{\epsilon} \right\rfloor = \left\lfloor \frac{\mathbf{v}}{\epsilon} \right\rfloor$ *and* $\left\| \mathbf{u} - \epsilon \left\lfloor \frac{\mathbf{u}}{\epsilon} \right\rfloor \right\| < \left\| \mathbf{v} - \epsilon \left\lfloor \frac{\mathbf{v}}{\epsilon} \right\rfloor \right\|$.

This is denoted by $\mathbf{u} \prec_\epsilon \mathbf{v}$.

Algorithm 1 ϵ-Box dominance archive update method

```
Input: The new solution $\mathbf{x}$ being added to the archive.
Output: true if $\mathbf{x}$ is added to the archive; false otherwise.
foreach solution $\mathbf{y}$ in the archive do
    if $\mathbf{x} \prec_\epsilon \mathbf{y}$ then
        remove $\mathbf{y}$ from the archive;
    else if $\mathbf{y} \prec_\epsilon \mathbf{x}$ then
        return false;
add $\mathbf{x}$ to the archive;
return true;
```

Conceptually, the ϵ-box dominance archive divides the objective space into hyperboxes with side length ϵ, called ϵ-boxes. The ϵ-box in which a solution resides is determined using the ϵ-box index vector. We use the notation $\lfloor \frac{\mathbf{u}}{\epsilon} \rfloor = \left(\lfloor \frac{u_1}{\epsilon} \rfloor, \lfloor \frac{u_2}{\epsilon} \rfloor, \ldots, \lfloor \frac{u_M}{\epsilon} \rfloor \right)$ for computing the ϵ-box index vector, where $\lfloor \cdot \rfloor$ is the floor function. As seen in Definition 5, dominance is determined using this index vector rather than the objective values. Case 2 in Definition 5 covers the situation in which two or more solutions reside in the same ϵ-box. In this situation, the solution nearest the lower-left (minimized) corner of the ϵ-box dominates any other solutions in the same ϵ-box.

3.2 ϵ-Progress

While the ϵ-box dominance archive guarantees convergence and diversity, this guarantee is subject to the solutions produced by the MOEA. MOEAs tend to fail on multi-modal problems due to preconvergence to local optima causing search to stagnate. In this section, we introduce a computationally efficient extension to the ϵ-box dominance archive for measuring search progression called ϵ-progress. Consequently, the inability of an MOEA to maintain ϵ-progress indicates search stagnation, which can subsequently trigger routines for reviving search.

DEFINITION 6: *ϵ-Progress occurs when a solution $\mathbf{x}$ passed to the update procedure outlined in Algorithm 1 is accepted into the archive such that no existing member of the archive existed with the same ϵ-box index vector.*

ϵ-Progress supplements the use of ϵ as the problem resolution by mandating ϵ as the minimum threshold for improvement. An MOEA must periodically produce at least one solution whose improvement exceeds this threshold to avoid stagnation. If stagnation is detected, appropriate action can be taken to either revive search or terminate the algorithm.

Figure 1 demonstrates ϵ-progress on a 2D example. Existing archive members are indicated by $\bullet$, and the ϵ-boxes dominated by these members are shaded gray. New solutions being added to the archive are indicated by $\times$. Cases (1) and (2) depict occurrences of ϵ-progress. The new solutions reside in previously unoccupied ϵ-boxes. Case (3) shows the situation in which the new solution is accepted into the archive, but since it resides in an occupied ϵ-box it does not count toward ϵ-progress—the improvement is below the threshold ϵ.

Extending the ϵ-box dominance archive in Algorithm 1 to include ϵ-progress is straightforward. In this study, the ϵ-box dominance archive increments a counter

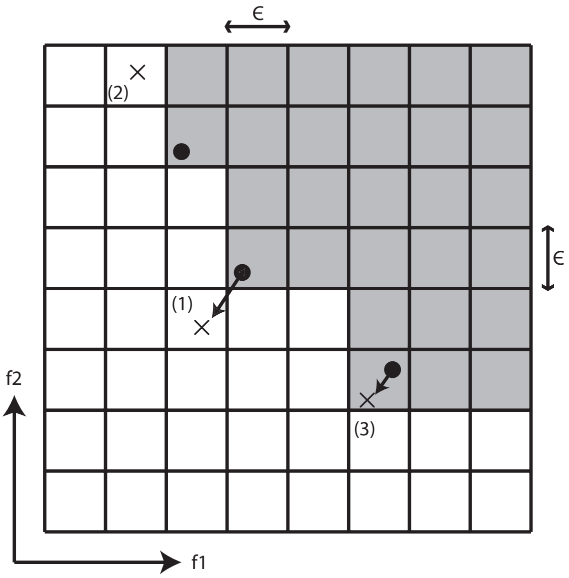


Figure 1: 2D example depicting how ϵ-progress is measured. Existing archive members are indicated by $\bullet$, and the ϵ-boxes dominated by these members are shaded gray. New solutions being added to the archive are indicated by $\times$. Cases (1) and (2) depict occurrences of ϵ-progress. The new solutions reside in previously unoccupied ϵ-boxes. Case (3) shows the situation in which the new solution is accepted into the archive, but since it resides in an occupied ϵ-box it does not count toward ϵ-progress—the improvement is below the threshold ϵ.

every time ϵ-progress occurs. This counter is periodically checked after a user-specified number of evaluations. If the counter is unchanged from the previous check, then the MOEA failed to produce significant improvements and the restart mechanism discussed in Section 3.3 is triggered.

3.3 Restarts

Restarts are a mechanism for reviving search after stagnation is detected using ϵ-progress. In Borg, a restart consists of three actions:

1. The search population size is adapted to remain proportional to the archive size.
2. The tournament selection size is adapted to maintain elitist selection.
3. The population is emptied and repopulated with solutions from the archive, with any remaining slots filled by mutated archive solutions.

Each of these three functions utilized in Borg restarts are described in more detail below.

3.3.1 Adaptive Population Sizing

Tang et al. (2006) observed that maintaining a population size proportional to the archive size helped escape local optima on a highly multimodal real-world problem. This mechanism of adapting the population size is built into the ϵ-NSGA-II algorithm by Kollat and Reed (2006) via the use of the population-to-archive ratio γ (ϵ-NSGA-II

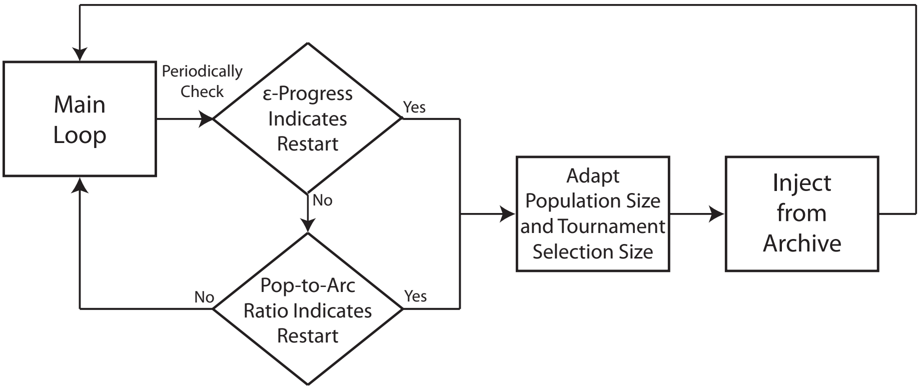

Main Loop
Periodically Check
ε-Progress Indicates Restart
Yes
No
Pop-to-Arc Ratio Indicates Restart
No
Yes
Adapt Population Size and Tournament Selection Size
Inject from Archive

Figure 2: Flowchart of the Borg restart logic. After a certain number of evaluations, the MOEA breaks out of its main loop to check if ϵ-progress or the population-to-archive ratio indicate a restart is required. If a restart is required, the population is resized and filled with all members of the archive. Any remaining population slots are filled with solutions selected randomly from the archive and mutated using uniform mutation applied with probability $1/L$. In addition, the tournament selection size is adjusted to account for the new population size. Finally, the MOEA's main loop is resumed.

literature refers to this ratio as the injection rate). The population-to-archive ratio specifies the ratio of the population size to the archive size:

$$\gamma = \frac{\text{population size}}{\text{archive size}} \geq 1. \tag{2}$$

Borg utilizes the same adaptive population sizing strategy as ϵ-NSGA-II, except that the population-to-archive ratio is maintained throughout the run. At any point during the execution of the algorithm, if the population-to-archive ratio differs from γ by more than 25%, the population size is adapted. Figure 2 outlines the logic of triggering restarts by ϵ-progress and the population-to-archive ratio.

This strategy ensures the population size remains commensurate with the Pareto front discovered by the MOEA. By using the archive size as a proxy for problem difficulty, we assume the population should grow proportionally with problem difficulty based on the theoretical recommendations of Horn (1995) and Mahfoud (1995).

3.3.2 Adaptive Tournament Size

Borg is designed such that it maintains tournament sizes to be τ, a fixed percentage of the population size, after every restart:

$$\text{tournament size} = \max\left(2, \lfloor \tau(\gamma A) \rfloor\right), \tag{3}$$

where A is the size of the archive. As Deb (2001) discusses, the concept of selection pressure is important in understanding the convergence behavior of EAs, but its formulation is not readily applicable to multi-objective optimization. Whereas selection pressure originally measured the probability of selecting the ith best individual from a population (Bäck, 1994), the multi-objective equivalent can be formulated as the probability of selecting a solution from the ith best rank. If we assume that the proportion of nondominated solutions in the population is approximately $1/\gamma$ after a restart, the probability of binary tournament selection choosing a nondominated member when

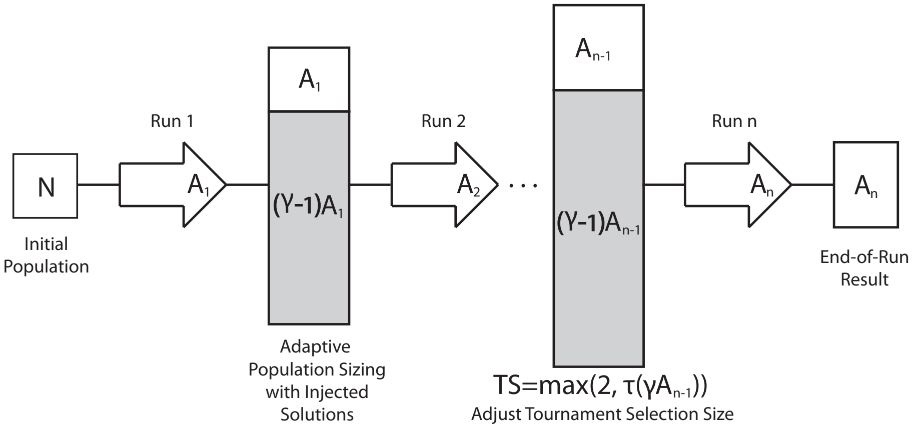


Figure 3: Illustration of how a population evolves from multiple restarts, forming what is known as "connected runs." With an initial population of size N, the MOEA is run until the first restart is triggered. At this point, the population is emptied and filled with the current archive, A_1. Next, the remaining slots in the resized population, shown in gray, are filled with solutions selected randomly from A_1 and mutated using uniform mutation applied with probability $1/L$. Lastly, the tournament size is adjusted to account for the new population size. This process repeats until termination.

$\gamma = 4$ is $1 - (1 - 1/\gamma)^2 = 1 - (\frac{3}{4})^2 = 0.44$. If instead $\gamma = 8$, this probability decreases to $1 - (\frac{7}{8})^2 = 0.23$, or roughly half what it was before. In order to maintain the same multi-objective selection pressure, the tournament size must be increased to 4, resulting in a selection probability of $1 - (\frac{7}{8})^4 = 0.41$. In this manner, τ governs the tournament size as the population dynamics increase the population size beyond the initial minimum value. Note that $\tau = 0$ can be used to enforce binary tournaments regardless of the population size.

3.3.3 Injection

The idea of injection is derived from the work of Goldberg (1989b) and Srivastava (2002) exploiting time continuation. Time continuation uses multiple-epoch runs instead of the single-epoch run typically employed by MOEAs. Multiple-epoch runs are characterized by periodically emptying the population, retaining the best solution(s), and repopulating with new randomly generated solutions. For multi-objective problems, Kollat and Reed (2006) introduced injection, which involves refilling the population with all members of the archive. Any remaining slots in the population are filled with new randomly generated solutions.

After some experimentation on the DTLZ (Deb et al., 2001), WFG (Huband et al., 2006), and CEC 2009 (Zhang, Zhou, et al., 2009) test suites, we observed that filling the remaining slots with solutions selected randomly from the archive and mutated using uniform mutation applied with probability $1/L$ achieved significantly better results. This is supported by the work of Schaffer et al. (1989) and others showing the dependence of effective mutation rates upon the number of decision variables L.

Figure 3 illustrates how a population evolves throughout the execution of the Borg MOEA as a result of the restart mechanism. Pseudocode for the restart mechanism is presented in Algorithm 2.

Algorithm 2 Random restart

```
Input: The current archive, the population-to-archive ratio $\gamma$, and the selection
       ratio $\tau$
Output: The population after random restart
Empty the population;
Fill population with all solutions in the archive;
// Compute the size of the new population
new_size $\leftarrow \gamma *$ size(archive);
// Inject mutated archive members into the new population
while size(population) < new_size do
    new_solution $\leftarrow$ select randomly one solution from archive;
    Mutate new_solution using uniform mutation applied with probability $1/L$;
    Add new_solution to population;
    Update archive with new_solution;
// Adjust tournament size to account for the new
   population size
Set the tournament size to max(2, floor($\tau *$ new_size));
```

3.4 Auto-Adaptive Multi-Operator Recombination

One of the problems encountered when using MOEAs in real-world contexts is the inability to know a priori which recombination operator performs best on a given problem. Vrugt and Robinson (2007) and Vrugt et al. (2009) address this issue by introducing an adaptive multi-operator hybrid called AMALGAM. The adaptability and reliability of AMALGAM was demonstrated on 10 multi-objective test problems in Vrugt and Robinson (2007) and a complex hydrologic model calibration problem in Zhang et al. (2010).

The idea is to establish a feedback loop in which operators that produce more successful offspring are rewarded by increasing the number of offspring produced by that operator. Given $K > 1$ operators, we maintain the probabilities $\{Q_1, Q_2, \ldots, Q_K\}$, $Q_i \in [0, 1]$, of applying each operator to produce the next offspring. These probabilities are initialized to $Q_i = 1/K$. Periodically, these probabilities are updated by first counting the number of solutions in the ϵ-box dominance archive that were produced by each operator, $\{C_1, C_2, \ldots, C_K\}$, and updating each Q_i by

$$Q_i = \frac{C_i + \varsigma}{\sum_{j=1}^{K}(C_j + \varsigma)}. \tag{4}$$

The constant $\varsigma > 0$ prevents the operator probabilities from reaching 0, thus ensuring no operators are "lost" during the execution of the algorithm. In this study, we use $\varsigma = 1$.

This approach differs from AMALGAM primarily in how the probabilities are updated. Our feedback loop updates the probabilities by counting the number of solutions produced by each operator in the ϵ-box dominance archive. Since AMALGAM is based on NSGA-II, which does not use an archive, it instead counts solutions in the population. This lack of an ϵ-dominance archive makes AMALGAM prone to deterioration on many-objective problems (Laumanns et al., 2002). In addition, since the ϵ-box

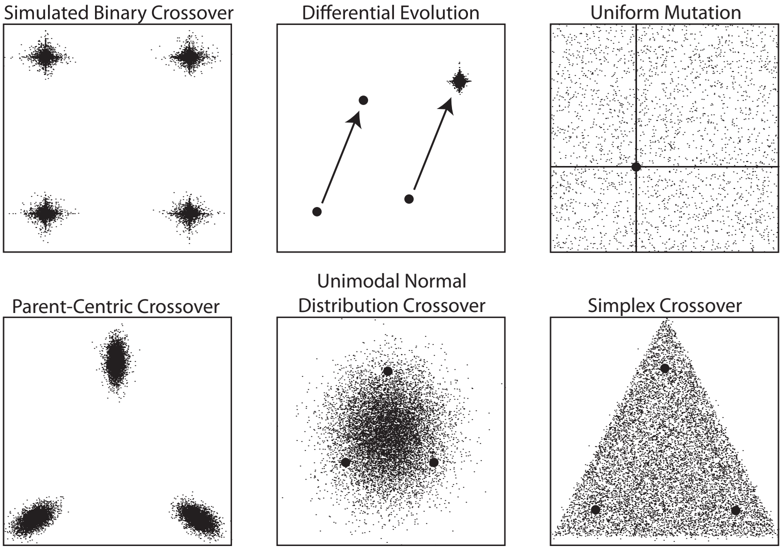


Figure 4: Examples showing the offspring distribution of the operators used in this study. Parents are indicated by •. The DE plot depicts the difference vector with arrows.

dominance archive maintains the best solutions in terms of both convergence and diversity, our approach favors operators producing offspring with both of these qualities.

As a result, the Borg MOEA is not a single algorithm but a class of algorithms whose operators are adaptively selected based on the problem and the decision variable encoding. The discovery of key operators is of particular importance to real-world problems where such information is unknown a priori. In addition, this is an ideal platform for benchmarking how new variation operators enhance search on complex many-objective problems. Since this study is considering only real-valued test problems, we have selected the following parent-centric, mean-centric, uniformly distributed, and self-adaptive real-valued operators.

- Simulated binary crossover (SBX; Deb and Agrawal, 1994).
- Differential evolution (DE; Storn and Price, 1997).
- Parent-centric crossover (PCX; Deb, Joshi et al., 2002).
- Unimodal normal distribution crossover (UNDX; Kita et al., 1999).
- Simplex crossover (SPX; Tsutsui et al., 1999).
- Uniform mutation (UM) applied with probability $1/L$

In addition, offspring produced by SBX, DE, PCX, UNDX, and SPX are mutated using polynomial mutation (PM; Deb and Agrawal, 1994). Figure 4 provides examples showing the offspring distribution generated by each of these operators. These figures clearly show the tendency of SBX, UM, and PM to generate solutions along a single axis, which degrades their efficacy on problems with conditional dependencies among

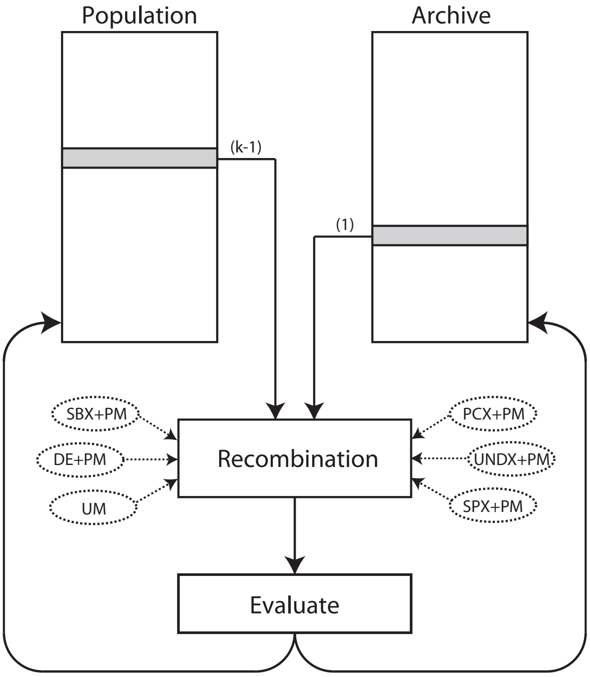


Figure 5: Flowchart of the Borg MOEA main loop. First, one of the recombination operators is selected using the adaptive multi-operator procedure described in Section 3.4. For a recombination operator requiring k parents, one parent is selected uniformly at random from the archive. The remaining $k-1$ parents are selected from the population using tournament selection. The offspring resulting from this operator are evaluated and then considered for inclusion in the population and archive.

its decision variables. DE, PCX, UNDX, and SPX do not exhibit this tendency; one can expect these four operators to perform better on rotated, epistatic problems.

3.5 The Algorithm

The Borg MOEA combines the components discussed in the previous sections within the ϵ-MOEA algorithm introduced by Deb et al. (2003). The rationale behind selecting ϵ-MOEA is its highly efficient steady-state model. Selection and replacement in ϵ-MOEA is based solely on the dominance relation and requires no expensive ranking, sorting, or truncation. In addition, the steady-state model will support parallelization in future studies without the need for synchronization between generations.

Figure 5 is a flowchart of the Borg MOEA main loop. First, one of the recombination operators is selected using the adaptive multi-operator procedure described in Section 3.4. For a recombination operator requiring k parents, one parent is selected uniformly at random from the archive. The remaining $k-1$ parents are selected from the population using tournament selection. The resulting offspring are evaluated and considered for inclusion in the population and archive.

If the offspring dominates one or more population members, the offspring replaces one of these dominated members randomly. If the offspring is dominated by at least one population member, the offspring is not added to the population. Otherwise, the offspring is nondominated and replaces a randomly selected member of the population.

Inclusion in the archive is determined with the archive update procedure outlined in Section 3.1.

Each iteration of this main loop produces one offspring. After a certain number of iterations of this main loop, ϵ-progress and the population-to-archive ratio are checked as described in Section 3.3. If a restart is required, the main loop halts and the restart procedure is invoked. Once the restart has completed, the main loop resumes and this process repeats until termination.

For the comparative analysis in this study, the Borg MOEA terminates after a fixed number of function evaluations. However, in practice, ϵ-progress can be used to terminate the algorithm if no improvements are detected after a specified number of function evaluations.

3.6 Runtime Analysis

Consider the runtime computational complexity of the Borg MOEA. For each offspring, dominance checks against the population and archive of sizes P and A, respectively, take time $O(M(P + A))$. However, since the population size is a constant multiple of the archive size, this simplifies to $O(MA)$. For η evaluations, the total runtime of the Borg MOEA is $O(\eta MA)$. Note that we simplified these expressions by assuming that selection and recombination take constant time.

Thus, the Borg MOEA is an efficient algorithm that scales linearly with the archive size. Recall from Section 3.1 how the archive size is controlled by the value of ϵ. By scaling ϵ, the algorithm can be made to run more efficiently at the cost of producing more approximate representations of the Pareto front. The determination of ϵ is left to the decision maker, who may use domain-specific knowledge of their significant precision goals or computational limits (Kollat and Reed, 2007).

3.7 Proof of Convergence

Exploring the limit behavior of an algorithm as the runtime goes to infinity, $t \to \infty$, is important from a theoretical view. It is not necessary for an algorithm to have guaranteed convergence to be practically useful, but issues like preconvergence and deterioration that arise in many-objective optimization make such results informative. In fact, most MOEAs do not have guaranteed convergence (Laumanns et al., 2002). The main crux of such convergence proofs is the assumption that there exists a nonzero probability of generating Pareto optimal solutions. Using the terminology of Rudolph (1998) and Rudolph and Agapie (2000), the recombination operators must have diagonal-positive transition matrices. Since tournament selection operates with replacement and all recombination operators used in this study have a form of mutation in which the entire decision space is reachable, the conditions outlined by Rudolph and Agapie (2000) for diagonal-positive transition matrices are satisfied.

The second necessary condition for guaranteed convergence on a multi-objective problem is elite preservation (Rudolph, 1998). As proved by Laumanns et al. (2002), the ϵ-dominance archive satisfies elite preservation. The ϵ-box dominance archive used in this study also satisfies elite preservation using the same logic—a solution in the archive at time t, $\mathbf{x} \in A_t$, is not contained in A_{t+1} if and only if there exists a solution $\mathbf{y} \in A_{t+1}$ with $F(\mathbf{y}) \prec_\epsilon F(\mathbf{x})$—thus proving the sequence of solutions generated by the Borg MOEA converges completely and in the mean to the set of minimal elements (the Pareto optimal set) as $t \to \infty$. In addition, Laumanns et al. (2002) proved the ϵ-box dominance archive preserves the diversity of solutions.

Table 1: The problems used in the comparative study along with key properties.

Problem	M	L	Properties	ϵ
UF1	2	30	Complicated Pareto set	0.001
UF2	2	30	Complicated Pareto set	0.005
UF3	2	30	Complicated Pareto set	0.0008
UF4	2	30	Complicated Pareto set	0.005
UF5	2	30	Complicated Pareto set, discontinuous	0.000001
UF6	2	30	Complicated Pareto set, discontinuous	0.000001
UF7	2	30	Complicated Pareto set	0.005
UF8	3	30	Complicated Pareto set	0.0045
UF9	3	30	Complicated Pareto set, discontinuous	0.008
UF10	3	30	Complicated Pareto set	0.001
UF11	5	30	DTLZ2 5D rotated	0.2
UF12	5	30	DTLZ3 5D rotated	0.2
UF13	5	30	WFG1 5D	0.2
DTLZ1	2–8	M+4	Multimodal, separable	0.01–0.35
DTLZ2	2–8	M+9	Concave, separable	0.01–0.35
DTLZ3	2–8	M+9	Multimodal, concave, separable	0.01–0.35
DTLZ4	2–8	M+9	Concave, separable	0.01–0.35
DTLZ7	2–8	M+19	Discontinuous, separable	0.01–0.35

3.8 Recommended Parameter Values

Appropriate parameterization of the algorithm and operators is important for its efficiency and effectiveness. The following parameterization guidelines are derived from the Latin hypercube sampling performed in Section 4 and the suggested operator parameterizations from the literature. Refer to the cited papers for the meaning and usage of the parameters.

For the Borg algorithm itself, it is recommended to use an initial population size of 100, a population-to-archive ratio of $\gamma = 4$ and a selection ratio of $\tau = 0.02$. On the problems tested, the SBX and PM operators performed best with distribution indices less than 100 with SBX applied with probability greater than 0.8. Both PM and UM should be applied with probability $1/L$. DE performed best with a crossover rate and step size of 0.6. For the multiparent operators, Deb, Joshi, et al. (2002) suggests using three parents for PCX and UNDX and $L + 1$ parents for SPX. For PCX, the σ_η and σ_ζ parameters controlling the variance of the resulting distribution should be set to 0.1 (Deb, Joshi, et al., 2002). For UNDX, use $\sigma_\xi = 0.5$ and $\sigma_\eta = 0.35/\sqrt{L}$ to preserve the mean vector and covariance matrix (Kita et al., 1999). For SPX, the expansion rate should be $\sqrt{P + 1}$, where P is the number of parents, to preserve the covariance matrix of the population (Tsutsui et al., 1999).

4 Comparative Study

To test the performance of the Borg MOEA, a comparative study between Borg, ϵ-MOEA, MOEA/D, GDE3, OMOPSO, IBEA, and ϵ-NSGA-II was undertaken using several many-objective test problems from the DTLZ (Deb et al., 2001), WFG (Huband et al., 2006), and CEC 2009 (Zhang, Zhou, et al., 2009) test problem suites. These top-ranked MOEAs provide a rigorous performance baseline for distinguishing Borg's contributions on a set of problems widely accepted in the community for benchmarking performance (Zhang and Suganthan, 2009). Table 1 lists the problems explored in this study along with their key properties.

Table 2: Statistical comparison of algorithms based on the 75% quantile of the hypervolume, generational distance and ϵ_+-indicator metrics. $+$, $=$, and $-$ indicate Borg's 75% quantile was superior, statistically indifferent from, or inferior to the competing algorithm, respectively.

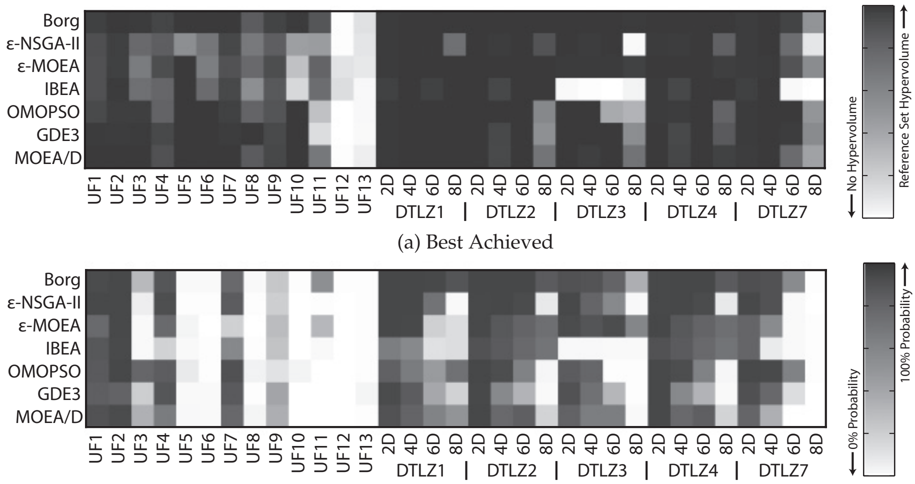
	Hypervolume			Generational distance			ϵ_+-Indicator		
Algorithm	$+$	$=$	$-$	$+$	$=$	$-$	$+$	$=$	$-$
ϵ-NSGA-II	15	8	10	17	4	12	15	4	14
ϵ-MOEA	16	9	8	24	3	6	17	3	13
IBEA	23	7	3	18	1	14	24	2	7
OMOPSO	24	4	5	25	3	5	22	4	7
GDE3	25	2	6	29	3	1	24	2	7
MOEA/D	25	3	5	27	3	3	24	4	5

Unlike single objective optimization, the result of multi-objective optimization is a nondominated set of points approximating the Pareto optimal set. Knowles and Corne (2002) provide a detailed discussion of the methods and issues for comparing nondominated sets. As a result of their analysis, they suggest the hypervolume metric since it can differentiate between degrees of complete outperformance of two sets, is independent of scaling, and has an intuitive meaning. Hypervolume measures the volume of objective space dominated by a nondominated set, thus capturing both convergence and diversity in a single metric. The major disadvantage of the hypervolume metric is its runtime complexity of $O(n^{M-1})$, where n is the size of the nondominated set. However, Beume and Rudolph (2006) provide an implementation with runtime $O(n \log n + n^{M/2})$ based on the Klee's measure algorithm by Overmars and Yap. This implementation permits computing the hypervolume metric on moderately sized nondominated sets up to $M = 8$ objectives in a reasonable amount of time.

For each problem instance, a reference set was generated using the known analytical solution to the problem. The reference point for calculating the hypervolume metric is based on the extent of each reference set plus a small increment. Without the small increment, the extremal points would register no volume and not contribute to the hypervolume value. Results with 0 hypervolume indicate the algorithm was unable to generate any solutions exceeding this reference point in any objective.

While the figures in this section only show the hypervolume metric, Table 2 does include summary results with generational distance and the additive ϵ-indicator (ϵ_+). Generational distance directly measures convergence whereas the ϵ_+-indicator provides a better measure of diversity and consistency (Coello Coello et al., 2007). We defer to (Hadka and Reed, 2012) for a more detailed discussion and analysis using these additional metrics.

Each algorithm was executed 1,000 times using parameters produced by a Latin hypercube sampling (LHS; McKay et al., 1979) across each algorithm's feasible parameter range. Each execution of a sampled parameter set was replicated 50 times with different randomly generated initial populations. The parameters analyzed include the population size, maximum number of objective function evaluations, and the parameters controlling selection and recombination operators. Since certain parameterizations can result in poor performance, the worst performing half of all parameterizations was eliminated from the remainder of this analysis. By analyzing the set of best performing parameters, we measure the performance of an algorithm in terms of solution quality as well as its reliability and controllability across a range of parameterizations.



(a) Best Achieved

(b) Attainment

Figure 6: Best achieved and 75% attainment results from the comparative study. (a) The best value achieved by the MOEA across all seeds, where black indicates values near the reference set hypervolume. (b) The probability of attaining at least 75% of the reference set hypervolume for each problem. Black indicates 100% probability; white indicates 0% probability.

The ranges from which the parameters were sampled is as follows. The number of fitness evaluations was sampled between [10,000, 1,000,000] in order to permit tractable execution times while providing meaningful results. The population size, offspring size, and archive sizes are all sampled between [10, 1,000]. This range was chosen to encompass the commonly employed rule-of-thumb population sizes in MOEA parameterization recommendations. Mutation rate, crossover rate, and step size encompass their entire feasible ranges of [0, 1]. Distribution indices for SBX and PM range between [0, 500], which is based on the sweet spot identified by Purshouse and Fleming (2007). The ϵ values used by the Borg MOEA, ϵ-MOEA, ϵ-NSGA-II, and OMOPSO are shown in Table 1.

Table 2 provides a summary of the results from this analysis. The Kruskal-Wallis one-way analysis of variance and Mann-Whitney U tests were used to compare the algorithms using the 75% quantile of the hypervolume, generational distance and ϵ_+-indicator metrics with 95% confidence intervals (Sheskin, 2004). These tests help guarantee any difference in the observed value is statistically significant and not a result of random chance. Table 2 records the number of problems in which the Borg MOEA outperformed, underperformed, or was statistically indifferent from each competing algorithm with respect to the 75% quantile of each metric. The 75% quantile was selected to compare the algorithms at a moderate level of success. As shown, Borg outperformed the competing algorithms on a majority of problem instances, but ϵ-NSGA-II and ϵ-MOEA were strong competitors.

For a more detailed view of the results, we compare the algorithms using their best achieved value and the probability of attaining at least 75% of the reference set hypervolume. The best achieved value, shown in Figure 6(a), presents the best achieved hypervolume for each algorithm across all seeds and parameters. Figure 6(b), which shows the probability of attaining at least 75% of the reference set hypervolume, indicates for each algorithm the percentage of its parameters and seeds that reached a moderate level of success (i.e., 75% of the reference set hypervolume). For completeness, we have also included 50% and 90% attainment plots in Figure A1 in the appendix. We distinguish between these two measurements since the best achieved value may be a needle in a haystack, where only a small number of parameters or seeds were successful. In this scenario, reporting only the best achieved value hides the fact that the likelihood of producing the best achieved value is low. The attainment measurement distinguishes these cases. All shaded figures in this paper, such as Figures 6(a) and 6(b), use a linear scale.

Figure 6(a) shows that across the majority of the tested problem instances, the Borg MOEA is able to produce approximation sets matching or exceeding the quality achieved by the competing algorithms. Only in UF1, UF8, UF12, and DTLZ7 8D is the Borg MOEA slightly outperformed. As GDE3 is the only algorithm outperforming the Borg MOEA on all such cases, this suggests the rotationally invariant DE operator may prove useful on these instances and consequently an optimal operator choice would be expected to provide some advantage relative to learning. MOEA/D and OMOPSO also show an advantage over the UF1 and 6D DTLZ7, respectively.

Figure 6(a) also shows several algorithms failing on UF12, UF13, and DTLZ3 at higher dimensions. UF12 and UF13 are rotated instances of the 5D DTLZ3 and 5D WFG1 problems. As unrotated DTLZ3 instances cause many MOEAs to fail (Hadka and Reed, 2012), it is not surprising that UF12 is difficult. What is surprising, however, is that the MOEAs tested in this study with rotationally invariant operators (e.g., GDE3 and Borg) struggled on UF12, given their good performance on the 6D DTLZ3. In addition, IBEA seems to completely fail on DTLZ3. As IBEA uses SBX and PM, which are the variation operators used by a number of the MOEAs tested in this study, this suggests the hypervolume indicator fails to guide search on this problem. Further investigation of this disparity should be undertaken.

While the majority of the algorithms produce at least one good approximation set on UF3, UF5, UF6, UF8, and UF10, Figure 6(b) shows that the probability of doing so is very low. This demonstrates how reporting only the best attained value may be misleading, as the likelihood of attaining good quality solutions may be extremely low.

Identifying and understanding the root causes of these failures is necessary to improve the reliability of MOEAs. UF5 and UF6 both consist of small, disjoint, finitely sized Pareto sets (Zhang, Zhou, et al., 2009). These sparse Pareto optimal solutions are separated by large gaps, which appear to cause significant problems for the variation operators, many of which, such as SBX, PCX, and PM, favor producing offspring near the parents. It is not immediately obvious which properties of UF3, UF8, and UF10 are causing all tested MOEAs to fail. UF8 and UF10 do share identical Pareto sets and Pareto fronts, which suggests the construction of the Pareto sets and Pareto fronts for these two problems may be the source of such failures.

In summary, the Borg MOEA showed superior performance in both the best attained value and the probability of attaining at least 75% of the reference set hypervolume. This is initial evidence that the Borg MOEA provides superior performance and reliability when compared to other state of the art MOEAs. However, there is still room for improvement on several of the UF test problems for all algorithms, as seen in the attainment results. The difficulties exhibited by UF3, UF5, UF6, UF8, and UF10 should prompt further investigation and influence the development of additional test problems.

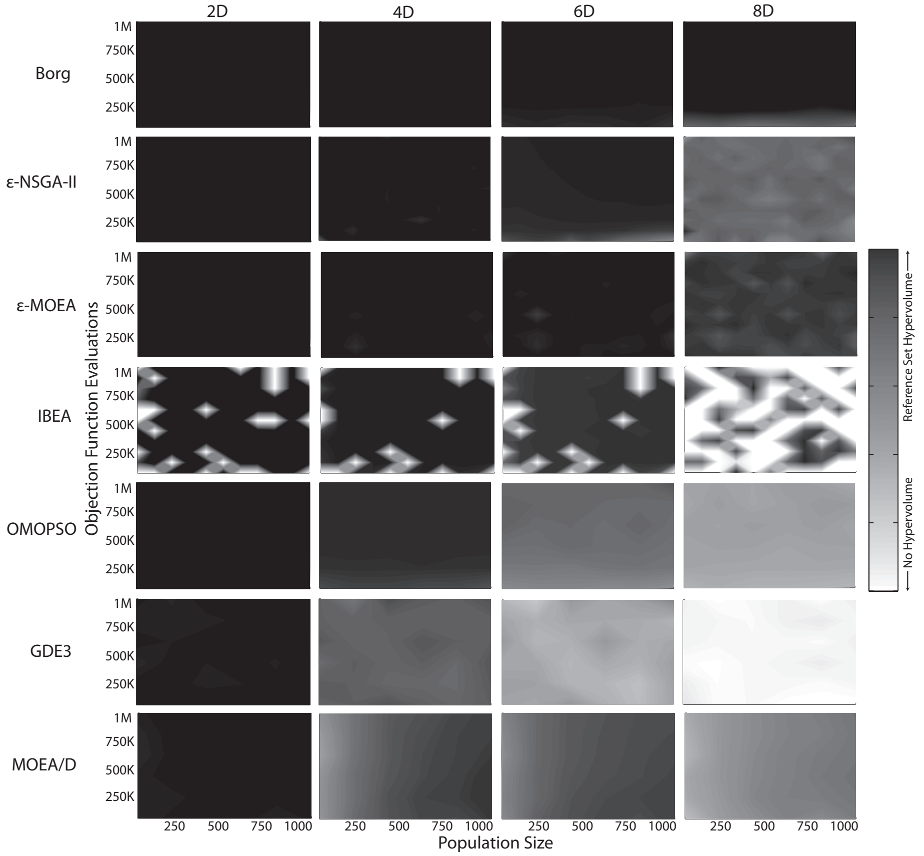


Figure 7: Control map showing the relation between population size and number of objective function evaluations on the DTLZ2 problem from two to eight objectives.

4.1 Control Maps

Figures 7 and 8 provide a more detailed exploration of the algorithms' performance on two specific problem instances, DTLZ2 and DTLZ1, by showing their control maps. These two problem instances are selected since DTLZ2 is one of the easiest problems tested in this study, whereas DTLZ1 is multimodal and challenging for all of the algorithms. Control maps highlight regions in parameter space whose parameterizations produce approximation sets with hypervolume values near the reference set hypervolume (black regions), and parameterizations that produce poor approximation sets (white regions). In this case, we plot population size versus the number of objective function evaluations.

Identifying so-called sweet spots is of particular interest, which are large regions of high-performing parameterizations (Goldberg, 1998). In Figure 7, all algorithms excluding IBEA show reliable parameterization on the 2D DTLZ2 instance. However, as the number of objectives is increased, MOEA/D, GDE3, OMOPSO, and IBEA show significant declines in performance. Borg, ϵ-MOEA, and ϵ-NSGA-II retain a large sweet spot on DTLZ2 instances with up to eight dimensions, but a small decline in performance is observed on ϵ-MOEA and ϵ-NSGA-II on the 8D DTLZ2 problem. In Figure 8,

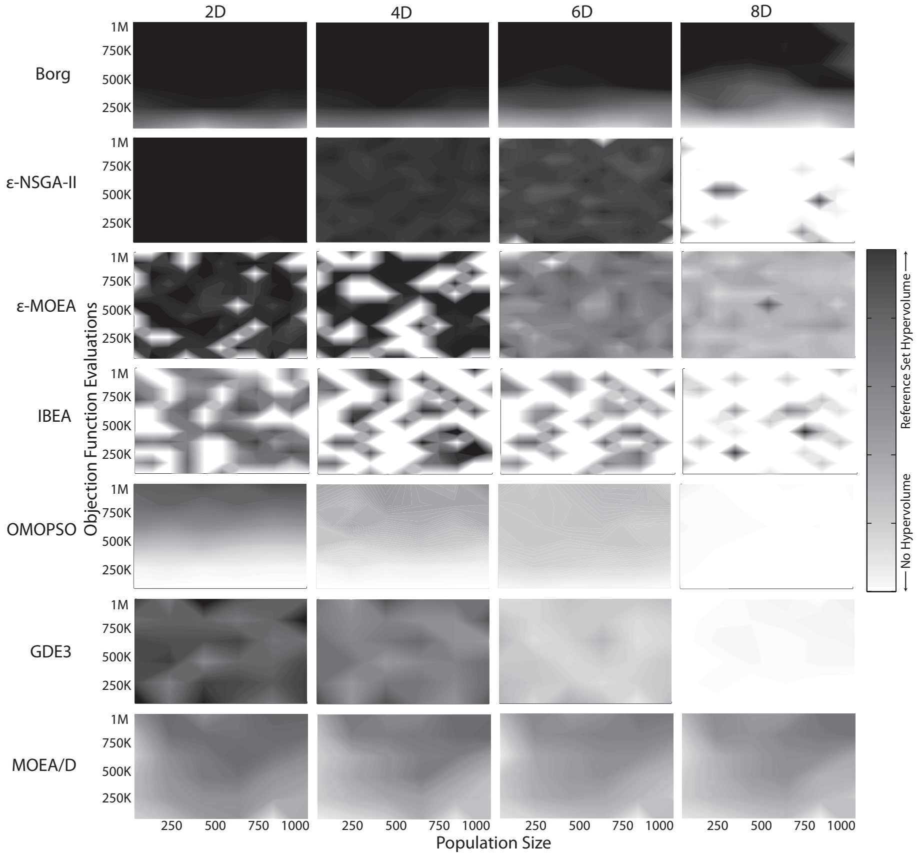
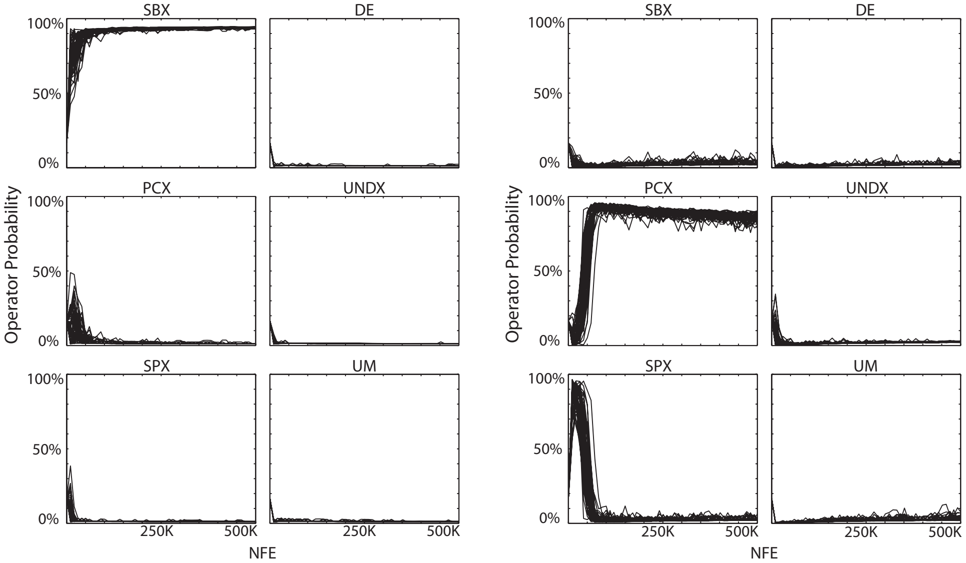


Figure 8: Control map showing the relation between population size and number of objective function evaluations on the DTLZ1 problem from two to eight objectives.

we observe that Borg and ϵ-NSGA-II are the only algorithms showing large sweet spots on DTLZ1, even on the 2D instance. Borg is the only tested algorithm with a sweet spot on the 8D DTLZ1 instance.

ϵ-MOEA and IBEA have chaotic control maps, with patches of light and dark regions, indicating that specific parameters or parameter combinations result in poor performance. Algorithms whose performance is highly dependent on parameter selection are expected to be difficult to use on real-world problems, where expensive objective evaluation costs prohibit experimentation to discover correct parameter settings. Utilizing MOEAs with large sweet spots is therefore desirable in real-world settings.

For algorithms that do not exhibit large sweet spots, trends can often be observed to guide better parameter selection. As an example, Figures 7 and 8 show MOEA/D has a strong dependency on population size. These results suggest that running MOEA/D with larger population sizes will tend to improve its resulting approximation sets. However, since MOEA/D's neighborhood scheme severely increases its runtime as the population size grows, increasing the population size may not be a feasible option. Borg is expected to be insensitive to the initial population size due to its adaptive population sizing scheme. Figures 7 and 8 confirm this hypothesis. For the Borg MOEA, the number



(a) DTLZ2 5D Unrotated

(b) DTLZ2 5D Rotated

Figure 9: The effect of epistasis on success of operators in Borg's auto-adaptive multi-operator recombination on the DTLZ2 problem. (a) The unrotated version from the DTLZ test suite. (b) The rotated version from the CEC 2009 competition.

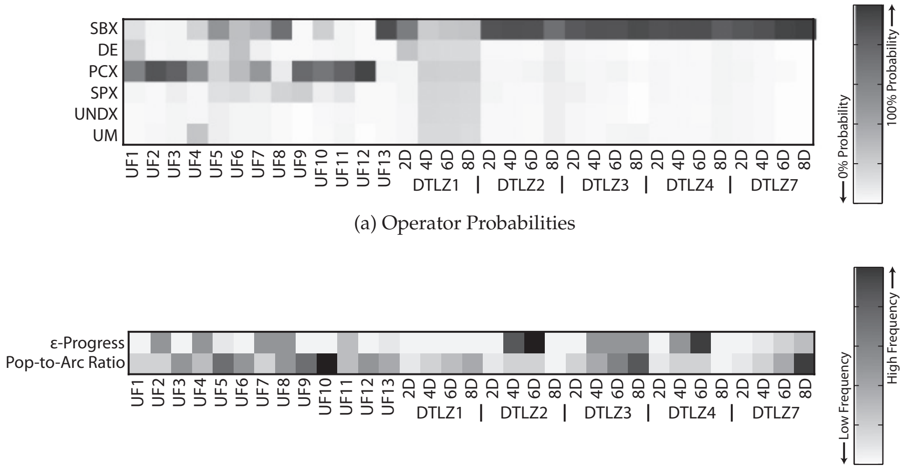
of objective function evaluations is key to improving its performance, suggesting the Borg MOEA will benefit from parallelization. The study of parameterization trends and their impact on controllability is discussed in detail in (Hadka and Reed, 2012).

4.2 Auto-Adaptive Multi-Operator Behavior

Next we demonstrate the ability of the auto-adaptive multi-operator recombination to adapt to a specific problem. Several of the tested problems can be classified into unrotated and rotated instances. Rotated instances have high degrees of conditional dependence between decision variables. Such conditional dependencies can degrade the performance of recombination operators, but we claim the auto-adaptive multi-operator procedure is capable of identifying and exploiting rotationally invariant operators on such problems. Figure 9 shows the operator probabilities as discussed in Section 3.4 throughout the execution of the Borg MOEA on an unrotated and rotated instance of the DTLZ2 problem. The plots show 50 replicates of Borg executed with the recommended parameters from Section 3.8. As expected, Borg correctly selects rotationally invariant operators to maximize performance on the rotated problems. It is interesting to note in Figure 9 that multiple operators work cooperatively during search and that their emphasis in search is highly dynamic (e.g., see SPX and PCX in Figure 9(b)).

A more comprehensive view of operator probabilities is given in Figure 10(a), which shows the percentage of operator usage throughout an entire run across all tested problem instances. Each cell in the figure is colored according to the percentage of operator use by calculating the "area under the curve" in the plots in Figure 9:

$$\text{Percentage}_i = \frac{\sum_j O(i, j)}{\sum_i \sum_j O(i, j)}, \tag{5}$$



(a) Operator Probabilities

(b) Restart Frequencies

Figure 10: (a) The percentage of operator usage throughout an entire run across all tested problems using a set of fixed parameters. Black cells indicate 100% usage and white cells indicate 0% usage of each operator. SBX and PCX are the two dominant operators on unrotated and rotated problems, respectively, while the other operators show moderate influence on several problems. (b) The restart frequencies due to ϵ-progress and the population-to-archive ratio. ϵ-Progress is scaled so that black cells indicate the maximum of 826 restarts observed during any run; the population-to-archive ratio is scaled so that black cells indicate the maximum of 14 observed restarts.

where $O(i, j)$ is the probability of applying operator i after j objective function evaluations. Using a similar setup as above, the results are averaged over 50 replicates of Borg executed with the recommended parameters from Section 3.8. Figure 10(a) shows that SBX is dominant on the unrotated DTLZ problems, whereas PCX, SBX, and several other operators have significant contributions on the UF problems. This confirms that a priori operator selection is nontrivial, especially on real-world problems where the problem characteristics are most likely unknown. Analysis of both Figure 6(b) and Figure 10(a) show that in UF1, UF4, UF7, UF9, UF11, and DTLZ1, Borg's high attainment benefits from the cooperative utilization of several variational operators. These results are corroborated by the findings of Vrugt and Robinson (2007) and of Vrugt et al. (2009), who also show that even operators that provide minor contributions can critically influence the quality of final results.

Figure 10(b) shows the frequency that ϵ-progress and population-to-archive ratio triggered restarts across all tested problem instances. On the DTLZ problem instances, we observe higher frequencies of both ϵ-progress and population-to-archive ratio restarts as the problem dimension is increased. As increasing the problem dimension generally results in proportionally larger nondominated Pareto sets, the population-to-archive ratio should be triggered more frequently with the increasing archive size. Overall, Figure 10 demonstrates that the auto-adaptive multioperator component and the two restart triggers are actively used across a variety of problems. In the following section, we extend this analysis to show that a combination of all three components is necessary for the performance and reliability of the Borg MOEA.

Table 3: Statistical comparison of the critical components of the Borg MOEA based on the 75% quantile of the hypervolume, generational distance, and ϵ_+-indicator metrics. $+$, $=$, and $-$ indicate the full Borg MOEA's 75% quantile was superior, statistically indifferent from, or inferior to the competing variant, respectively. The enabled components in each variant are identified with letters (A) population-to-archive ratio triggered restarts with adaptive population sizing; (B) ϵ-progress; and (C) auto-adaptive multioperator recombination.

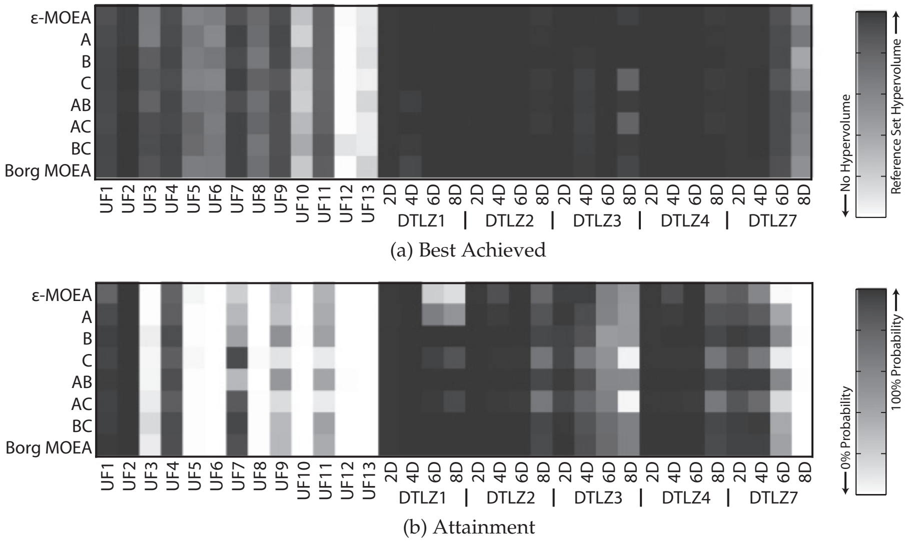
	Hypervolume			Generational Distance			ϵ_+-Indicator		
Variant	$+$	$=$	$-$	$+$	$=$	$-$	$+$	$=$	$-$
A	17	6	10	22	4	7	16	4	13
B	11	10	12	11	4	18	9	8	18
C	21	6	6	24	4	5	15	7	11
AB	11	13	9	11	4	18	9	7	17
AC	20	6	7	23	7	3	14	8	11
BC	5	20	8	2	28	3	4	22	7

4.3 Critical Components of Borg

We conclude this analysis by determining how critical each of the individual constituent components of the Borg MOEA is to the Borg MOEA's overall performance and reliability. The components analyzed are (A) population-to-archive ratio triggered restarts with adaptive population sizing; (B) ϵ-progress triggered restarts; and (C) the auto-adaptive multioperator recombination operator. We repeated the analyses based on 1,000 LHS parameterizations, where each parameterization is run for 50 random seed replicates, as was done earlier, but with the individual components enabled or disabled to run all six potential variants. These variants are compared against the baseline ϵ-MOEA and the full Borg MOEA.

Table 3 shows the statistical comparison of the different combinations. For the majority of the tested cases, the full Borg variant is equivalent or superior. Given that the test problem suite used in this study is biased toward a few operators (i.e., SBX on DTLZ), it is expected that the variants B and AB are competitive. Since the Borg MOEA must expend objective function evaluations learning the dominant operator(s), the variants using the dominant operator by default have a competitive advantage. The full potential of auto-adaptive multioperator variation is on real-world applications, where the dominant operators are not known a priori and are likely to vary. Simply having the ability to discover this operator dependency is a significant contribution and strength of the Borg MOEA.

Figure 11 shows the best achieved value and the probability of attaining at least 75% of the reference set hypervolume for the different variants. As expected, ϵ-progress has a very strong impact on proximity, but requires the other operators for diversity, as seen on the darker shadings for variants B, AB, and BC in Figure 11(b). The effects of the auto-adaptive multioperator variation operator can be seen on a number of problems, and is very pronounced on UF7. The variants C, AC, BC, and the full Borg MOEA show significant improvements on UF7. From Figure 10(a), we see this was achieved by identifying SBX and PCX as dominant operators. Figure 11 does verify that the variants without the multioperator learning curve do have an advantage on the DTLZ test problems in which SBX is a dominant operator. The population-to-archive ratio triggered restarts with adaptive population sizing appear to have a more pronounced effect on the higher dimensional DTLZ instances. This distinction is clearly visible when



(a) Best Achieved

(b) Attainment

Figure 11: Best achieved and 75% attainment results from the analysis of the critical components of the Borg MOEA. (a) The best value achieved by the configuration across all seeds, where black indicates values near the reference set hypervolume. (b) The probability of attaining at least 75% of the reference set hypervolume for each problem. Black indicates 100% probability; white indicates 0% probability. The enabled components in each variant are identified with letters: (A) population-to-archive ratio triggered restarts with adaptive population sizing; (B) ϵ-progress; and (C) auto-adaptive multioperator recombination.

comparing ϵ-MOEA with variant A. The earlier results seen on ϵ-NSGA-II also support the positive impacts of adaptive population sizing as captured in Table 3, which shows ϵ-NSGA-II as one of the top performers overall. It can therefore be concluded that all three constituent components of the Borg MOEA contribute to its observed success and its intended use in many-objective real-world applications.

5 Conclusion

The Borg MOEA provides robust optimization by assimilating design components from other MOEAs and introduces several novel features. ϵ-Dominance archives and ϵ-progress maintain a well-spread Pareto front and monitor convergence speed. If the convergence speed declines and search stagnates, randomized restarts are triggered which revive search by resizing and diversifying the population while carefully maintaining selection pressure. Lastly, Borg provides a facility to incorporate multiple recombination operators and automatically adapts its use of these operators based on their relative performance.

Our comparative study demonstrates the efficacy of Borg on 33 test problem instances ranging from two to eight objectives. Using a large-scale Latin hypercube sampling of each algorithm's parameters, we observed that the Borg MOEA outperformed the competing algorithms on the majority of the test problems. Borg reliably and consistently produced Pareto sets matching or exceeding the best-known algorithms in terms of hypervolume, generational distance, and ϵ_+-indicator.

In particular, Borg showed significant advantages over competing algorithms on many-objective, multimodal problems. On such problems, Borg produced results with significantly better hypervolume and achieved such results with higher probability. However, all of the tested algorithms showed serious reliability issues on several UF problems, an issue which should elicit further investigations.

While Borg's use of multiple recombination operators requires users to set more parameters, our control map results highlight that Borg's auto-adaptive search features strongly reduce parameterization challenges and provide large sweet spots, even on problems with eight objectives. Nonetheless, operator selection and parameterization are important considerations when maximizing the performance of any MOEA. In Hadka and Reed (2012), we detail the dependencies among search operators, their parameters, and many-objective performance for a broad range of MOEAs, including the Borg framework introduced in this study.

Acknowledgments

This work was supported in part through instrumentation funded by the National Science Foundation through grant OCI-0821527. Patent protection for the Borg MOEA is being pursued by the Pennsylvania State University including, but not limited to, United States patent application number 13/356, 391 (pending). Interested readers can contact Patrick Reed (preed@engr.psu.edu) for access to the Borg MOEA's source code.

References

Aguirre, H., and Tanaka, K. (2009). Space partitioning with adaptive epsilon-ranking and substitute distance assignments: A comparative study on many-objective MNK-landscapes. In *2009 Genetic and Evolutionary Computation Conference*, pp. 547–554.

Bäck, T. (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp. 57–62.

Beume, N., and Rudolph, G. (2006). Faster S-metric calculation by considering dominated hypervolume as Klee's measure problem. In *Proceedings of the Second IASTED Conference on Computational Intelligence*, pp. 231–236.

Coello Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. New York: Springer Science.

Corne, D. W., and Knowles, J. D. (2000). The Pareto-envelope based selection algorithm for multiobjective optimization. In *Parallel problem solving from nature (PPSN VI)*, pp. 839–848.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York: Wiley.

Deb, K., and Agrawal, R. B. (1994). Simulated binary crossover for continuous search space. Technical Report IITK/ME/SMD-94027, Indian Institute of Technology, Kanpur, UP, India.

Deb, K., Joshi, D., and Anand, A. (2002). Real-coded evolutionary algorithms with parent-centric recombination. In *Proceedings of the World Congress on Computational Intelligence*, pp. 61–66.

Deb, K., Mohan, M., and Mishra, S. (2003). A fast multi-objective evolutionary algorithm for finding well-spread Pareto-optimal solutions. KanGAL Report No. 2003002, Kanpur Genetic Algorithms Laboratory (KanGAL).

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2001). Scalable test problems for evolutionary multi-objective optimization. TIK-Technical Report No. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH).

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC 2002)*, pp. 825–830.

di Pierro, F., Khu, S.-T., and Savic, D. A. (2007). An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation,* 11(1):17–45.

Drechsler, N., Drechsler, R., and Becker, B. (2001). Multi-objective optimisation based on relation favour. In *Evolutionary multi-criterion optimization* (pp. 154–166). Berlin: Springer.

Farina, M., and Amato, P. (2004). A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* 34(3):315–326.

Ferringer, M. P., Spencer, D. B., and Reed, P. (2009). Many-objective reconfiguration of operational satellite constellations with the large-cluster epsilon non-dominated sorting genetic algorithm-II. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2009)*, pp. 340–349.

Fleming, P. J., Purshouse, R. C., and Lygoe, R. J. (2005). Many-objective optimization: An engineering design perspective. In *Evolutionary multi-criterion optimization*. *Lecture notes in computer science*, Vol. 3410 (pp. 14–32).

Fonseca, C. M., and Fleming, P. J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part 1: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* 28(1):26–37.

Fonseca, C. M., and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423.

Goldberg, D. E. (1989a). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Goldberg, D. E. (1989b). Sizing populations for serial and parallel genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 70–79.

Goldberg, D. E. (1998). The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. Technical Report IlliGAL Report No. 98007, University of Illinois.

Hadka, D., and Reed, P. (2012). Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evolutionary Computation*, doi: 10.1162/EVCO_a_00053.

Hanne, T. (1999). On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research,* 117:553–564.

Hanne, T. (2001). Global multiobjective optimization with evolutionary algorithms: Selection mechanisms and mutation control. In *Evolutionary multi-criterion optimization*. *Lecture notes in computer science*, Vol. 1993 (pp. 197–212). Berlin: Springer.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Horn, J. (1995). *The nature of niching: Genetic algorithms and the evolution of optimal, cooperative populations*. PhD thesis, University of Illinois.

Horn, J., and Nafpliotis, N. (1993). Multiobjective optimization using the niched Pareto genetic algorithm. Technical Report 93005, University of Illinois.

Huband, S., Hingston, P., Barone, L., and While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation,* 10(5):477–506.

Ikeda, K., Kita, H., and Kobayashi, S. (2001). Failure of Pareto-based MOEAs: Does non-dominated really mean near optimal? In *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 957–962.

Iorio, A., and Li, X. (2008). Improving the performance and scalability of differential evolution. In *Simulated evolution and learning*. *Lecture notes in computer science*, Vol. 5361 (pp. 131–140). Berlin: Springer.

Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., and Nojima, Y. (2008). Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, pp. 649–656.

Ishibuchi, H., Tsukamoto, N., Sakane, Y., and Nojima, Y. (2010). Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10*, pp. 527–534.

Kasprzyk, J. R., Reed, P. M., Kirsch, B. R., and Characklis, G. W. (2009). Managing population and drought risks using many-objective water portfolio planning under uncertainty. *Water Resources Research*, 45, doi: 10.1029/2009WR008121.

Kasprzyk, J. R., Reed, P. M., Kirsch, B. R., and Characklis, G. W. (2011). Many-objective de novo water supply portfolio planning under deep uncertainty. *Environmental Modelling & Software*, doi: 10.1016/j.enrsoft.2011.04.003.

Kita, H., Ono, I., and Kobayashi, S. (1999). Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. In *Proceedings of Congress on Evolutionary Computation*, pp. 1581–1588.

Knowles, J., and Corne, D. (2002). On metrics for comparing non-dominated sets. In *Proceedings of the Congress on Evolutionary Computation (CEC 2002)*, pp. 711–716.

Knowles, J. D., and Corne, D. W. (1999). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8:149–172.

Kollat, J., Reed, P., and Maxwell, R. (2011). Many-objective groundwater monitoring network design using bias-aware ensemble Kalman filtering, evolutionary optimization, and visual analytics. *Water Resources Research*, 47, doi: 10.1029/2010WR009194.

Kollat, J. B., and Reed, P. M. (2006). Comparison of multi-objective evolutionary algorithms for long-term monitoring design. *Advances in Water Resources*, 29(6):792–807.

Kollat, J. B., and Reed, P. M. (2007). A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. *Advances in Water Resources*, 30(3):408–419.

Kukkonen, S., and Lampinen, J. (2005). GDE3: The third evolution step of generalized differential evolution. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 443–450.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282.

Mahfoud, S. W. (1995). *Niching methods for genetic algorithms*. PhD thesis, University of Illinois.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Purshouse, R. C., and Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6):770–784.

Reyes Sierra, M., and Coello Coello, C. A. (2005). Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance. In *Evolutionary Multi-Criterion Optimization. Lecture notes in computer science*, Vol. 3410 (pp. 505–519). Berlin: Springer.

Rudolph, G. (1998). Evolutionary search for minimal elements in partially ordered sets. In *Proceedings of the 7th Annual Conference on Evolutionary Programming. Evolutionary Programming VII*, pp. 345–353.

Rudolph, G., and Agapie, A. (2000). Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation (CEC 2000)*, Vol. 2, pp. 1010–1016.

Saxena, D., and Deb, K. (2008). Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 3204–3211.

Schaffer, D. J. (1984). *Multiple objective optimization with vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University.

Schaffer, D. J., Caruana, R. A., Eshelman, L. J., and Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 51–60.

Sheskin, D. J. (2004). *Handbook of parametric and nonparametric statistical procedures*. New York: Chapman & Hall.

Srinivas, N., and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation,* 2(3):221–248.

Srivastava, R. P. (2002). Time continuation in genetic algorithms. Technical report, Illinois Genetic Algorithm Laboratory.

Storn, R., and Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization,* 11(4):341–359.

Sülflow, A., Drechsler, N., and Drechsler, R. (2007). Robust multi-objective optimization in high dimensional spaces. In *Evolutionary multi-criterion optimization* (pp. 715–726). Berlin: Springer.

Tang, Y., Reed, P., and Wagener, T. (2006). How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? *Hydrology and Earth System Science, 10*:289–307.

Teytaud, O. (2006). How entropy-theorems can show that on-line approximating high-dim Pareto-fronts is too hard. In *Parallel problem solving from Nature. BTP Workshop*.

Teytaud, O. (2007). On the hardness of offline multi-objective optimization. *Evolutionary Computation,* 15(4):475–491.

Tsutsui, S., Yamamura, M., and Higuchi, T. (1999). Multi-parent recombination with simplex crossover in real coded genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, pp. 657–664.

Vrugt, J. A., and Robinson, B. A. (2007). Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences,* 104(3):708–711.

Vrugt, J. A., Robinson, B. A., and Hyman, J. M. (2009). Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Transactions on Evolutionary Computation,* 13(2):243–259.

Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary multi-criterion optimization* (pp. 742–756). Berlin: Springer.

Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Proceedings of the Congress on Evolutionary Computation (CEC 2009)*, pp. 203–208.

Zhang, Q., and Suganthan, P. N. (2009). Final report on CEC'09 MOEA competition. In *Proceedings of the Congress on Evolutionary Computation (CEC 2009)*. Piscataway, NJ: IEEE Press.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (2009). Multiobjective optimization test instances for the CEC 2009 special session and competition. Technical Report CES-487, University of Essex.

Zhang, X., Srinivasan, R., and Liew, M. V. (2010). On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model. *Hydrological Processes,* 24(8):955–969.

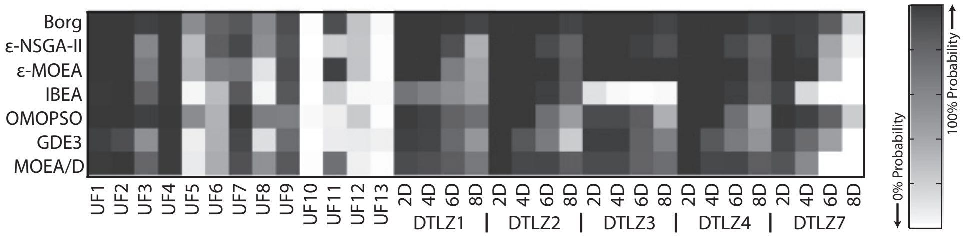
Zitzler, E., and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Parallel problem solving from nature (PPSN VIII). Lecture notes in computer science* (pp. 832–842). Berlin: Springer.

Zitzler, E., and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation,* 3(4):257–271.

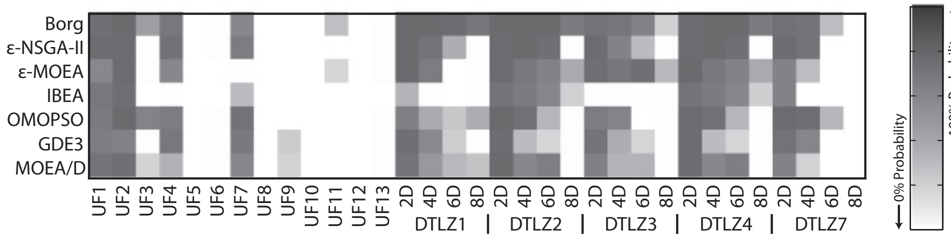
Appendix

Probability of Attainment for Various Probabilities

Figure A1 shows the probability of attainment for reaching the 50% and 90% attainment thresholds. These results conincide with the 75% attainment results shown in Figure 6(b).



(a) 50% Attainment



(b) 90% Attainment

Figure A1: 50% and 90% attainment results from the comparative study. Black indicates 100% probability; white indicates 0% probability.