Comparing State-of-the-Art Multi-Objective Evolutionary Algorithms (MOEAS) for Long-Term Groundwater Monitoring (LTM) Design

Joshua Kollat\textsuperscript{1} and Patrick Reed\textsuperscript{2}

Department of Civil and Environmental Engineering
The Pennsylvania State University

\textsuperscript{1}juk124@psu.edu, \textsuperscript{2}preed@engr.psu.edu
Presentation Information

Citation:


For additional information, refer to:


Note:

This presentation has been slightly modified from its original version in order to better suite the static nature of PDF format.
The Goals

- Seeking to solve large, complex problems quickly, efficiently and reliably
- Make the optimization process easier by reducing the need for EA expertise
- Understand precision requirements:
  - Increased accuracy requires increased computation
  - Example: Do we care about costs in dollars, cents
    Concentrations in ppb, ppm, etc.
Introduction: Why MOEAs for LTM Design

- **Challenging aspects:**
  - **Huge Decision Spaces**
    - Example: a 25 well sampling problem contains $2^{25}$ (or over 33-million) possible designs
  - **“Many-Objective” problems – four or more design objectives**
    - Examples: cost, error, uncertainty, risk, multiple contaminants
  - **Conflicting objectives – increased performance in one objective degrades performance in another**
    - Example: decreasing error requires increased costs
  - **Diverse groups of stakeholders**
    - Examples: scientists, engineers, regulators, and public
What are Evolutionary Algorithms (EAs)?

- Evolutionary algorithms
  - Population based search heuristic
  - Population is evolved toward optimal designs using natural selection
  - Operators: selection, mating, mutation

- An example of how EAs represent decisions

![Binary Transform Diagram]

System Being Designed

\[
\begin{align*}
  &d_1 \\
  &d_2 \\
  &\vdots \\
  &d_{k-1} \\
  &d_k
\end{align*}
\]

\[
\begin{bmatrix}
  1 \\
  0 \\
  \vdots \\
  0 \\
  1
\end{bmatrix} \rightarrow \{10\ldots01\} \quad \text{Potential Design}
\]

Decision Vars
Multiobjective Optimization

- **Pareto front:**
  - No solutions exist that exceed performance in all objectives
  - Improvements to one objective degrade performance in another objective

- **Nondominated front:**
  - Local approximation to the Pareto optimal set

\[
\min \left[ f_1(x_1, x_2), f_2(x_1, x_2) \right]
\]
The Study

- Compare the performance of four state-of-the-art MOEAs at solving a LTM design test case
  - NSGAII – Deb et al. 2002
    - Non-domination sorting and crowding distance
  - $\varepsilon$-NSGAII – Reed et al. 2005
    - $\varepsilon$-non-dominance archiving, dynamic population sizing, automatic termination
  - $\varepsilon$MOEA – Deb et al. 2003
    - Steady-state, $\varepsilon$-non-dominance archiving
  - SPEA2 – Zitzler et al. 2001
    - Strength Pareto fitness assignment, kth nearest neighbor method

- Demonstrate the improved performance and usability of the $\varepsilon$-NSGAII
LTM Test Case

- High resolution flow and transport simulation of a PCE contamination plume
- 25 well sampling locations available
Objectives

- **Four objectives:**
  1. Minimize sampling cost
  2. Minimize contaminant concentration estimation error
  3. Minimize contaminant estimation uncertainty
  4. Minimize contaminant mass estimation error

- Subject to the constraint that no points are left unestimated

$$f_{\text{cost}} (\mathbf{x}_\kappa) = \sum_{i=1}^{n_{\text{wells}}} C_s (i) x_{\kappa,i}^{\text{samp}}$$

$$f_{\text{CONC}} (\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} (c_{\text{all}}^* (\mathbf{u}_j) - c_{\text{est}}^\kappa (\mathbf{u}_j))^2$$

$$f_{\text{uncert}} (\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} A_j \sqrt{\sigma^2 (\mathbf{u}_j)}$$

$$f_{\text{mass}} (\mathbf{x}_\kappa) = \left| \frac{\text{Mass}_{\text{all}} - \text{Mass}_{\kappa}}{\text{Mass}_{\text{all}}} \right| \cdot 100$$
LTM Test Case Enumeration

- Enumerated four-objective Pareto front:

  - $2^{25}$ possible designs (over 33-million)
  - 1 week of continuous computing
  - 2,439 Pareto optimal designs
  - 45.6% of the decision space is infeasible
## ε-NSGAII Details

<table>
<thead>
<tr>
<th>Changes:</th>
<th>Results:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε-non-dominance archiving</td>
<td>Intuitive precision specification</td>
</tr>
<tr>
<td></td>
<td>Reduces computational cost</td>
</tr>
<tr>
<td>Dynamic population sizing</td>
<td>Increases population size according to problem difficulty</td>
</tr>
<tr>
<td></td>
<td>Reduces parameter requirements</td>
</tr>
<tr>
<td>Automatic termination</td>
<td>Eliminates unnecessary computation time</td>
</tr>
<tr>
<td></td>
<td>More intuitive parameterization</td>
</tr>
</tbody>
</table>
\( \varepsilon \)-Dominance

- Step 1: User specified \( \varepsilon \) grid is applied to the objective space and duplicate solutions are removed
ε-Dominance

- Step 1: User specified ε grid is applied to the objective space and duplicate solutions are removed
- Step 2: Non-domination based on grid blocks is performed
**ε-Dominance**

- **Step 1:** User specified $\varepsilon$ grid is applied to the objective space and duplicate solutions are removed.
- **Step 2:** Non-dominance based on grid blocks is performed.
\(\varepsilon\)-Dominance

- Step 1: User specified \(\varepsilon\) grid is applied to the objective space and duplicate solutions are removed
- Step 2: Non-domination based on grid blocks is performed
- Step 3: Solutions are “thinned” based on precision goals
$\varepsilon$-Dominance Applied to LTM Test Case

$\varepsilon = [0.0, 0.0, 0.0, 0.0]$
ε-Dominance Applied to LTM Test Case

ε = [ 1.0, 0.1, 1.1, 0.1 ]

1502 Solutions
ε-Dominance Applied to LTM Test Case

$\epsilon = [1.0, 2.0, 10.0, 1.0]$
Dynamic Population Sizing

- Allows population size to increase commensurate with problem difficulty
- 25% injection used to size population
- Population size is bounded by the user specified $\varepsilon$-non-dominance archive
Experimental Details

- **Effectiveness:**
  - How close do they get to the true solution?
  - Assessed using metrics which measure different aspect of performance

- **Efficiency:**
  - How fast do they find the true solution?
  - Dynamic performance assessed using runtime metrics

- **Reliability:**
  - How do the initial conditions effect the final result?
  - Ran 50 trials for each algorithm

- **Ease-of-Use:**
  - Do the algorithms require extensive parameterization?
  - Are the required parameters meaningless to all but the experts or are they intuitive?
Results: Runtime Convergence

- **NSGAII**
- **εMOEA**
- **ε-NSGAII**
- **SPEA2**

Range of Performance

Standard Deviation

Mean performance

Convergence vs. Total Function Evaluations (x10^3)
Results: Runtime Convergence

\( \varepsilon \)-NSGAII

SPEA2

Convergence \((x 10^3)\)

Total Function Evaluations \((x 10^3)\)
Results: Runtime Convergence

ɛ-NSGAII

SPEA2

Convergence (x10^3)

Total Function Evaluations (x10^3)
Results: Runtime Diversity

- NSGAIi
- εMOEA
- ε-NSGAIi
- SPEA2

Diversity vs. Total Function Evaluations (x10^3)

ASCE EWRI 2005

May 18, 2005
Results: Runtime $\varepsilon$-Performance

- NSGAII
- $\varepsilon$MOEA
- $\varepsilon$-NSGAII
- SPEA2

Graphs showing $\varepsilon$-Performance over Total Function Evaluations ($\times 10^3$).
Results: Runtime $\varepsilon$-Performance

$\varepsilon$-NSGAII

- Performance

SPEA2

- Performance

Total Function Evaluations ($\times 10^3$)

~8000

~16000
Results: Runtime $\varepsilon$-Performance

$\varepsilon$-NSGAII

SPEA2

$\varepsilon$-Performance

Total Function Evaluations ($x10^3$)

~8000

~16000
Conclusions

- The ε-NSGAII is more effective, efficient, and reliable than the NSGAII and the εMOEA
- Although the SPEA2 performs well
  - ε-NSGAII is easier to use
- Advantages of the ε-NSGAII over the other algorithms:
  - Eliminates population size parameter
  - Eliminates runtime parameter
  - Allows for intuitive specification of precision requirements
  - Eliminates the need for trial and error analysis
Questions??
References