Stochastic Population Update Can Provably Be Helpful in Multi-Objective Evolutionary Algorithms*

Chao Bian, Yawen Zhou, Miqing Li, Chao Qian

http://www.lamda.nju.edu.cn/qianc/
Email: qianc@nju.edu.cn

LAMDA Group
Nanjing University, China

*Published at IJCAI’23
Multi-objective Optimization

**Multi-objective optimization** tries to optimize multiple objectives simultaneously

\[
\min_{s \in S} (f_1(s), f_2(s), \ldots, f_m(s))
\]

- \(x\) dominates \(z\): \(f_1(x) < f_1(z) \land f_2(x) < f_2(z)\)
- \(x\) is incomparable with \(y\): \(f_1(x) > f_1(y) \land f_2(x) < f_2(y)\)

**Pareto optimal solution**: a solution that cannot be dominated by any other solution in \(S\)

**Pareto front**: the set of objective vectors of all the Pareto optimal solutions, which represents different optimal trade-offs between objectives

**Goal**: finding the Pareto front or its good approximation
Multi-objective Optimization

Multi-objective optimization has many applications:

**Buy cars**
- Max: performance
- Min: price

**Search neural architectures**
- Max: accuracy
- Min: network complexity

![Cars](image1.jpg)

![Neural Network Diagram](image2.png)
Evolutionary Algorithms (EAs)

The general structure of EAs

The population-based nature makes EAs suitable for solving multi-objective optimization problems.
Applications of Multi-objective EAs (MOEAs)

MOEAs have been widely applied for solving real-world multi-objective tasks

- **High entropy alloy design**
  - [Menou et al, Materials and Design’18]
  - • Max: strength
  - • Min: density

- **Gasoline engine design**
  - [Fujita et al, DETC’98]
  - • Max: acceleration ability
  - • Min: fuel consumption

- **Supply chain design**
  - [Benyoucef and Xie, Springer’11]
  - • Max: service quality
  - • Min: cost
Popular MOEAs

Pareto dominance based: NSGA-II, SPEA-II, ...


Performance indicator based: SMS-EMOA, HyPE, ....


Decomposition based: MOEA/D, ....

NSGA-II

Framework of NSGA-II:

1. **Solution representation**
   - Solution1
   - Solution2
   - Solution3

2. **Initial population**
   - Population size is $\mu$

3. **Parent selection**
   - Selects each solution once

4. **Parent solutions**

5. **Mutation & recombination**

6. **Offspring solutions**
   - Generates $\mu$ offspring solutions

7. **Fitness evaluation**

8. **Population update**
   - $\mu + \mu$ selection?
NSGA-II

Population Update of NSGA-II:

Use non-dominated sorting and crowding distance sorting to rank the solutions, and delete the worst ones.
Non-dominated sorting

Partition the solutions in $P \cup P'$ into $R_1, R_2, \ldots, R_n$

- solutions in $R_1$ (has rank 1): cannot be dominated by any solution in $P \cup P'$
- solutions in $R_2$ (has rank 2): cannot be dominated by any solution in $(P \cup P') \setminus R_1$
- ...

Rank reflects the convergence of a solution

Solutions with smaller rank are better
Population Update of NSGA-II:

Use **non-dominated sorting** and **crowding distance sorting** to rank the solutions, and delete the worst ones.
NSGA-II

Crowding distance assignment

For each objective $f_i$:

- sort the solutions w.r.t. $f_i$ (ascending)
- for each solution $x$, compute the normalized distance w.r.t. $f_i$, and add the distance to the final crowding distance value of $x$

normalized distance:

- if $x$ is sorted in the first or last position, set the distance to $\infty$
- otherwise, set the distance to $\frac{f_i(\text{succ}(x)) - f_i(\text{prec}(x))}{\max f_i - \min f_i}$

Crowding distance reflects the diversity of a solution

Solutions with larger crowding distance are better
NSGA-II

Population Update of NSGA-II:

Use **non-dominated sorting** and **crowding distance sorting** to rank the solutions, and delete the worst ones.

The current population $P$ is non-dominated sorted to get $R_1$ and $R_2$. Then, the offspring population $P'$ is non-dominated sorted to get $R_3$ and $R_4$. The **crowding distance sorting** is applied to $R_1$ and $R_2$, and the worst ones $R_3$ and $R_4$ are deleted. The remaining solutions form the next population.
Framework of SMS-EMOA:

1. **Initial population**
   - Population size is $\mu$

2. **Parent selection**
   - Randomly selects one solution

3. **Parent solutions**

4. **Mutation & recombination**

5. **Offspring solutions**
   - Generates one offspring solution

6. **Solution representation**

7. **Stop criterion**
   - No

8. **New population**
   - $\mu + 1$ selection

9. **Population update**

10. **Fitness evaluation**

11. **Stop criterion**
    - Yes

12. **New population update**
Population Update of SMS-EMOA:

Use non-dominated sorting and quality indicators (e.g., hypervolume) to rank the solutions, and delete the worst solution.

The current population $P$ is non-dominated sorted, and one offspring solution $x'$ is introduced. Then, $x'$ is also non-dominated sorted to a set $R_1$. The hypervolume loss calculation is performed between $R_1$ and $R_2$, where $R_2$ is the current population. The worst solution is deleted from the current population.
SMS-EMOA

Hypervolume: volume of the space dominated by a set of solutions, reflecting the convergence and diversity of the solutions.

Hypervolume loss calculation

Hypervolume loss of \( x \):
- decreased hypervolume value of the solution set when \( x \) is removed

\[
\Delta(x) = HV(R_1) - HV(R_1 \setminus \{x\})
\]

Solutions with larger hypervolume loss are better.
SMS-EMOA

Population Update of SMS-EMOA:

Use non-dominated sorting and quality indicators (e.g., hypervolume) to rank the solutions, and delete the worst solution.

The current population $P$ goes through non-dominated sorting to create a set $R_1$. Then, one offspring solution $x'$ is added to $R_1$ to create set $R_2$. The hypervolume loss calculation is performed on $R_2$ to determine which solutions to keep for the next population. The worst solution is deleted, and the remaining solutions form the next population.
MOEA/D

Framework of MOEA/D:

- multi-objective problem
  - decomposition
    - single-objective sub-problem 1
    - single-objective sub-problem 2
    - ... single-objective sub-problem \( N \)
  - optimization by collaboration
    - solution 1
    - solution 2
    - ... solution \( N \)
MOEA/D

Population Update of MOEA/D:
For each single-objective sub-problem, the newly generated solution will replace the worse solutions.
The prominent feature in population update of MOEAs: greedy and deterministic

- the next-generation population is formed by selecting the best-ranked solutions

"One common aspect of these first-generation multi-objective algorithms is that they did not use any elite-preservation operator, thereby compromising the performance and was also contrary to Rudolph’s asymptotic convergence proof which required the preservation of elites from one generation to the next."

An Interview with Kalyanmoy Deb 2022 ACM Fellow

Is deterministic population update always better?

NO!
Our Main Results

Expected number of generations of SMS-EMOA and NSGA-II for solving the OneJumpZeroJump [Doerr and Zheng, AAAI’21] and bi-objective RealRoyalroad [Dang et al., AAAI’23] problems

<table>
<thead>
<tr>
<th>Population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMS-EMOA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$O(\mu n^k)$ [\mu \geq n - 2k + 3]$</td>
<td>$O(\mu n^{n/5-2})$ [\mu \geq 2n/5 + 1]$</td>
</tr>
<tr>
<td></td>
<td>$\Omega(n^k)$ [n - 2k = \Omega(n) \land \mu = \text{poly}(n)]</td>
<td>$\Omega(n^{n/5-1})$ [\mu = \text{poly}(n)]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(\mu^2 n^k / 2^{k/4})$ [\mu \geq 2(n - 2k + 4)]</td>
<td>$O(\mu^2 n^{n/5} / 2^{n/20})$ [\mu \geq 2(2n/5 + 2)]</td>
</tr>
<tr>
<td><strong>NSGA-II</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$\Omega(n^k / \mu)$ [n - 2k = \Omega(n) \land \mu = \text{poly}(n)]</td>
<td>$\Omega(n^{n/5-1} / \mu)$ [\mu = \text{poly}(n)]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(k(n/2)^k)$ [\mu \geq 8(n - 2k + 3)]</td>
<td>$O(n(20e^2)^{n/5})$ [\mu \geq 8(2n/5 + 1)]</td>
</tr>
</tbody>
</table>

Green color: results of our IJCAI’23 work; Yellow color: extended results
Our Main Results

<table>
<thead>
<tr>
<th>population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deterministic</strong></td>
<td>$O(\mu n^k)$ $[\mu \geq n - 2k + 3]$</td>
<td>$O(\mu n^{n/5 - 2})$ $[\mu \geq 2n/5 + 1]$</td>
</tr>
<tr>
<td><strong>Stochastic</strong></td>
<td>$\Omega(n^k)$ $[n - 2k = \Omega(n) \land \mu = \text{poly}(n)]$</td>
<td>$\Omega(n^{n/5 - 1})$ $[\mu = \text{poly}(n)]$</td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>$\Omega(n^k/\mu)$ $[n - 2k = \Omega(n) \land \mu = \text{poly}(n)]$</td>
<td>$\Omega(n^{n/5 - 1}/\mu)$ $[\mu = \text{poly}(n)]$</td>
</tr>
<tr>
<td><strong>Stochastic</strong></td>
<td>$O(k(n/2)^k)$ $[\mu \geq 8(n - 2k + 3)]$</td>
<td>$O\left(n(20e^2)^{n/5}\right)$ $[\mu \geq 8(2n/5 + 1)]$</td>
</tr>
</tbody>
</table>

For example, for SMS-EMOA solving the OneJumpZeroJump problem

Expected running time $\Omega(n^k)$ accelerated by $2^{k/4}/\mu^2$ to $O(\mu^2 n^k/2^{k/4})$ exponential acceleration if $k = \Omega(n) \land k = n/2 - \Omega(n) \land 2(n - 2k + 4) \leq \mu = \text{poly}(n)$
OneJumpZeroJump Problem

Definition of OneJumpZeroJump:

\[ f_1(x) = \begin{cases} 
  k + |x|_1, & \text{if } |x|_1 \leq n - k \text{ or } x = 1^n \\
  n - |x|_1, & \text{else} 
\end{cases} \]

\[ f_2(x) = \begin{cases} 
  k + |x|_0, & \text{if } |x|_0 \leq n - k \text{ or } x = 0^n \\
  n - |x|_0, & \text{else} 
\end{cases} \]

• Pareto set: \( \{x \mid |x|_1 \in [k..n-k] \cup \{0,n\} \} \)

• Pareto front: \( \{(a, n + 2k - a) \mid a \in [2k..n] \cup \{k, n + k\} \} \)

Characterize a class of problems where some adjacent Pareto optimal solutions in the objective space locate far away in the decision space.
Population Update of SMS-EMOA:

Use non-dominated sorting and quality indicators (e.g., hypervolume) to rank the solutions, and delete the worst solution.
Theorem. For SMS-EMOA solving OneJumpZeroJump with $n - 2k = \Theta(n)$, if using a population size $\mu$ such that $\mu = \text{poly}(n)$, then the expected number of generations for finding the Pareto front is $\Omega(n^k)$.

Proof sketch:

• all the solutions in the initial population belong to the inner part of the Pareto front with probability $1 - o(1)$
• the solution with number of 1-bits in $[1, k - 1] \cup [n - k + 1, n - 1]$ cannot be maintained
• the extreme solution $1^n$ (and $0^n$) can only be generated by flipping $k$ bits of a solution simultaneously (whose probability is at most $1/n^k$)
SMS-EMOA Using Stochastic Population Update

Procedure of stochastic population update

1. $Q' \leftarrow \left\lfloor |Q|/2 \right\rfloor$ solutions uniformly and randomly selected from $Q$ without replacement
2. partition $Q'$ into non-dominated sets $R_1, R_2, \ldots, R_v$
3. let $z = \arg \min_{x \in R_v} \Delta_r(x, R_v)$
4. return $Q \setminus \{z\}$

The difference:
the removed solution is selected from a subset $Q'$ of $Q$, instead of the entire set $Q$
Analysis of SMS-EMOA Using Stochastic Population Update

**Theorem.** For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size $\mu$ such that $\mu \geq 2(n - 2k + 4)$, then the expected number of generations for finding the Pareto front is $O(\mu^2 n^k / 2^{k/4})$.

**Lemma.** For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size $\mu$ such that $\mu \geq 2(n - 2k + 4)$, then
- an objective vector $f^*$ in the Pareto front will always be maintained once it has been found
- any solution in $P \cup \{x\}'$ can be maintained in the next population with probability at least $1/2$
Analysis of SMS-EMOA Using Stochastic Population Update

**Theorem.** For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size $\mu$ such that $\mu \geq 2(n - 2k + 4)$, then the expected number of generations for finding the Pareto front is $O(\mu^2 n^k / 2^{k/4})$.

**Proof sketch:**

- find a solution in the inner part of the Pareto set: $O(\mu k^k)$
- find the whole inner part of the Pareto front: $O(\mu n \log n)$
- the extreme solution $1^n$ (or $0^n$) can be generated by gradually flipping the 0-bits (or 1-bits)

Use additive drift [He and Yao, AIJ’01]
Theorem. For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size $\mu$ such that $\mu \geq 2(n - 2k + 4)$, then the expected number of generations for finding the Pareto front is $O(\mu^2 n^k / 2^{k/4})$.

Proof sketch:

Use additive drift [He and Yao, AIJ’01]: consider the change of $\max_{x \in P} |x|_1$

- the distance function is defined as

$$V(P) = \begin{cases} 
0 & \text{if } \max_{x \in P} |x|_1 = n, \\
e\mu n^{k/2} & \text{if } n - k/2 \leq \max_{x \in P} |x|_1 \leq n - 1, \\
e\mu n^{k/2} + 1 & \text{if } n - k \leq \max_{x \in P} |x|_1 < n - k/2.
\end{cases}$$

Target state

The probability of jumping to the target state is small, thus the distance to the target state is set to be a large value.

The probability of jumping to the better state is large, thus the distance to the better state is set to be a small value.
Analysis of SMS-EMOA Using Stochastic Population Update

**Theorem.** For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size \( \mu \) such that \( \mu \geq 2(n - 2k + 4) \), then the expected number of generations for finding the Pareto front is \( O(\mu^2 n^k / 2^{k/4}) \).

**Proof sketch:**

Use additive drift [He and Yao, AIJ’01]: consider the change of \( \max_{x \in P} |x|_1 \)

- if \( n - k/2 \leq q \equiv \max_{x \in P} |x|_1 \leq n - 1 \)
  
  - expected decrease of \( V \): \( \frac{1}{(e\mu n^{n-q})} \cdot e\mu n^{k/2} \geq 1 \)
    
    Prob. of generating \( 1^n \)

  - expected increase of \( V \): \( \frac{1}{2} \cdot 1 \)
    
    Prob. of losing the solution with \( q \) 1-bits

  - expected change of \( V \): \( 1/2 \)

\[ \text{Change of } V \]

\[ \text{Change of } V \]
Analysis of SMS-EMOA Using Stochastic Population Update

**Theorem.** For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size $\mu$ such that $\mu \geq 2(n - 2k + 4)$, then the expected number of generations for finding the Pareto front is $O(\mu^2 n^k/2^{k/4})$.

**Proof sketch:**

Use additive drift [He and Yao, AIJ’01]: consider the change of $\max_{x \in P} |x|_1$ if $n - k \leq q := \max_{x \in P} |x|_1 < n - k/2$

- expected decrease of $V = \left(\frac{k/2}{k/4}\right)/(2\mu n^{k/2})$ 
  - Prob. of decreasing $V$
  - Change of $V$
- $V$ cannot increase because the solution with $(n - k)$ 1-bits will always be maintained
- expected change of $V = \left(\frac{k/2}{k/4}\right)/(2\mu n^{k/2}) \geq 2^{k/4}/(2\mu n^{k/2})$
Analysis of SMS-EMOA Using Stochastic Population Update

**Theorem.** For SMS-EMOA solving OneJumpZeroJump, if using stochastic population update, and a population size $\mu$ such that $\mu \geq 2(n - 2k + 4)$, then the expected number of generations for finding the Pareto front is $O(\mu^2n^k/2^{k/4})$.

Proof sketch:

Use additive drift [He and Yao, AIJ’01]: consider the change of $\max_{x \in P} |x|_1$

Combining the analysis of the two cases

- expected change of $V$: $2^{k/4}/(2e\mu n^{k/2})$
- $V(P) \leq e\mu n^{k/2} + 1$

- expected number of generations for finding $1^n$: $O(\mu^2n^k/2^{k/4})$
# Our Main Results

<table>
<thead>
<tr>
<th>Population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMS-EMOA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$O(\mu n^k)$ [$\mu \geq n - 2k + 3$]</td>
<td>$O(\mu n^{n/5 - 2})$ [$\mu \geq 2n/5 + 1$]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$\Omega(n^k)$ [$n - 2k = \Omega(n) \land \mu = \text{poly}(n)$]</td>
<td>$\Omega(n^{n/5 - 1})$ [$\mu = \text{poly}(n)$]</td>
</tr>
<tr>
<td><strong>NSGA-II</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$\Omega(n^k/\mu)$ [$n - 2k = \Omega(n) \land \mu = \text{poly}(n)$]</td>
<td>$\Omega(n^{n/5 - 1}/\mu)$ [$\mu = \text{poly}(n)$]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(k(n/2)^k)$ [$\mu \geq 8(n - 2k + 3)$]</td>
<td>$O(n(20e^2)^{n/5})$ [$\mu \geq 8(2n/5 + 1)$]</td>
</tr>
</tbody>
</table>

For SMS-EMOA solving the OneJumpZeroJump problem

Expected running time $\Omega(n^k)$ accelerated by $2^{k/4}/\mu^2$ $O(\mu^2 n^k/2^{k/4})$ exponential acceleration if $k = \Omega(n) \land k = n/2 - \Omega(n)$ $2(n - 2k + 4) \leq \mu = \text{poly}(n)$
By introducing randomness into population update, MOEAs can go across inferior regions between different Pareto optimal solutions more easily.

- **Deterministic**
  - prefers non-dominated solutions
  - if objective vectors in the Pareto front are far away in the solution space, easy to get trapped

- **Stochastic**
  - allows dominated solutions to participate in the evolutionary process
  - follows an easier path in the solution space to find adjacent points in the Pareto front
Our Main Results

<table>
<thead>
<tr>
<th>Population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMS-EMOA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$O(\mu n^k)$ [(\mu \geq n - 2k + 3)]</td>
<td>$O(\mu n^{n/5-2})$ [(\mu \geq 2n/5 + 1)]</td>
</tr>
<tr>
<td></td>
<td>$\Omega(n^k)$ [(n - 2k = \Omega(n) \land \mu = \text{poly}(n))]</td>
<td>$\Omega(n^{n/5-1})$ [(\mu = \text{poly}(n))]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(\mu^2n^{k/2^{k/4}})$ [(\mu \geq 2(n - 2k + 4))]</td>
<td>$O(\mu^2n^{n/5}/2^{n/20})$ [(\mu \geq 2(2n/5 + 2))]</td>
</tr>
<tr>
<td><strong>NSGA-II</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$\Omega(n^k/\mu)$ [(n - 2k = \Omega(n) \land \mu = \text{poly}(n))]</td>
<td>$\Omega(n^{n/5-1}/\mu)$ [(\mu = \text{poly}(n))]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(k(n/2)^k)$ [(\mu \geq 8(n - 2k + 3))]</td>
<td>$O(n(20e^2)^{n/5})$ [(\mu \geq 8(2n/5 + 1))]</td>
</tr>
</tbody>
</table>

For NSGA-II solving the bi-objective RealRoyalroad problem

Expected running time

- **Stochastic**

\[ \Omega(n^{n/5-1}/\mu) \] accelerated by \((n/(20e^2))^{n/5}/(\mu n^2)\) to

\[ O(n(20e^2)^{n/5}) \]
Bi-objective RealRoyalroad Problem

Definition of bi-objective RealRoyalroad:

\[
f(x) = \begin{cases} 
(n|x|_1 + TZ(x), n|x|_1 + LZ(x)), & \text{if } x \in H \cup G; \\
(0, 0), & \text{else,}
\end{cases}
\]

where \(H = \{x \mid |x|_1 = 4n/5 \land TZ(x) + LZ(x) = n/5\}\) and \(G = \{x \mid |x|_1 \leq 3n/5\}\) e.g., \(H = \{11110, 01111\}\) for \(n = 5\)

- Pareto set: \(H\)
- Pareto front: \(\{(4n^2/5 + a, 4n^2/5 + n/5 - a) \mid a \in [0..n/5]\}\)

Characterize a class of problems where a large gap exists between the Pareto front and the second front in the decision space

\(\{x \mid |x|_1 = 3n/5 \land TZ(x) + LZ(x) = 2n/5\}\)

Illustration of function values when \(n = 5\)
NSGA-II

Population Update of NSGA-II:
Use non-dominated sorting and crowding distance sorting to rank the solutions, and delete the worst ones.

Diagram:
- The current population $P$ is non-dominated sorted to produce $R_1$.
- The offspring population $P'$ is non-dominated sorted to produce $R_2$.
- $R_1$ and $R_2$ are combined and crowding distance sorted to rank the solutions.
- The worst solutions, $R_3$ and $R_4$, are deleted.
- The remaining solutions constitute the next population.
Analysis of NSGA-II

**Theorem.** For NSGA-II solving bi-objective RealRoyalroad, if using a population size $\mu$ such that $\mu = \text{poly}(n)$, then the expected number of generations for finding the Pareto front is $\Omega\left(\frac{n^{n/5-1}}{\mu}\right)$.

Proof sketch:

- all the solutions in the initial population have at most $\frac{3n}{5}$ 1-bits with probability $1 - o(1)$
- the solution with more than $\frac{3n}{5}$ 1-bits (not in $H$) has the objective vector $(0,0)$, and cannot be maintained
- the solutions in $H$ can only be generated by flipping at least $n/5$ bits of a solution simultaneously
NSGA-II Using Stochastic Population Update

The removed solutions are selected from a random subset of $P \cup P'$, instead of the entire set.

- The current population $P$ is randomly selected.
- The offspring population $P'$ is non-dominated sorted.
- $[\mu/2]$ solutions are directly maintained.
- $[\mu/2]$ solutions are selected from a random subset of $P \cup P'$, instead of the entire set.
- $\lceil \mu/2 \rceil$ solutions are directly maintained.
- $\lfloor \mu/2 \rfloor$ solutions are selected for competition, where $[\mu/2]$ of them can survive.
Theorem. For NSGA-II solving bi-objective RealRoyalroad, if using stochastic population update and a population size $\mu$ such that $\mu \geq 8(2n/5 + 1)$, then the expected number of generations for finding the Pareto front is $O(n(20e^2)^n/5)$.

Proof sketch:

- a solution with $3n/5$ 1-bits can be found in $O(n \log n)$
- a solution in $G' = \{0^i 1^{3n/5} 0^{2n/5-i} | 0 \leq i \leq 2n/5\}$ can be found in $O(n^3)$
- all the solutions in $G'$ can be found in $O(n^3)$
- a Pareto optimal solution can be found in $O(n(20e^2)^{n/5})$
- all the Pareto optimal solutions can be found in $O(n^3)$

Use “lucky way” argument [Doerr, TCS’21]
Analysis of NSGA-II Using Stochastic Population Update

**Theorem.** For NSGA-II solving bi-objective RealRoyalroad, if using stochastic population update and a population size $\mu$ such that $\mu \geq 8(2n/5 + 1)$, then the expected number of generations for finding the Pareto front is $O(n(20e^2)^{n/5})$.

**Proof sketch:**

Use “lucky way” argument [Doerr, TCS’21]

- consider a phase of consecutive $n/5$ generations

\[ \frac{1^{3n/5}}{n/5}(n/5)/(en) \cdot (1/4) \quad \text{find a solution } x \text{ with } |x|_1 = 3n/5 + 1 \text{ and } TZ(x) \geq n/5 \]

- any solution can survive with prob. at least 1/4

\[ \frac{1^{4n/5}}{(n/5 - 1)/(en) \cdot (1/4)} \quad \text{find a solution } x \text{ with } |x|_1 = 3n/5 + 2 \text{ and } TZ(x) \geq n/5 \]

... with prob. at least $n/5 - 1/4e \cdot (1/4)$

Any solution can survive with prob. at least $n/5 - 1/4e \cdot (1/4)$. 

Pareto optimal
Analysis of NSGA-II Using Stochastic Population Update

**Theorem.** For NSGA-II solving bi-objective RealRoyalroad, if using stochastic population update and a population size $\mu$ such that $\mu \geq 8(2n/5 + 1)$, then the expected number of generations for finding the Pareto front is $O(n(20e^2)^{n/5})$.

**Proof sketch:**

Use “lucky way” argument [Doerr, TCS’21]

- consider a phase of consecutive $n/5$ generations

  $1^{3n/5}0^{2n/5} \rightarrow \cdots \rightarrow 1^{4n/5}0^{n/5}$  

  Pareto optimal

- the probability of the above event is at least

  $\prod_{i=1}^{n/5} (n/5 - i + 1)/(4en) \geq 2/(20e^2)^{n/5}$

- the expected number of generations of finding a Pareto optimal solution: $(20e^2)^{n/5}/2 \cdot (n/5)$
## Our Main Results

<table>
<thead>
<tr>
<th></th>
<th>Population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMS-EMOA</strong></td>
<td>Deterministic</td>
<td>$O(\mu n^k)$ [μ ≥ n − 2k + 3]</td>
<td>$O(\mu n^{n/5-2})$ [μ ≥ 2n/5 + 1]</td>
</tr>
<tr>
<td></td>
<td>Deterministic</td>
<td>$\Omega(n^k)$ [n − 2k = Ω(n) ∧ μ = poly(n)]</td>
<td>$\Omega(n^{n/5-1})$ [μ = poly(n)]</td>
</tr>
<tr>
<td></td>
<td>Stochastic</td>
<td>$O(\mu^2 n^k / 2^{k/4})$ [μ ≥ 2(n − 2k + 4)]</td>
<td>$O(\mu^2 n^{n/5} / 2^{n/20})$ [μ ≥ 2(2n/5 + 2)]</td>
</tr>
<tr>
<td><strong>NSGA-II</strong></td>
<td>Deterministic</td>
<td>$\Omega(n^k / \mu)$ [n − 2k = Ω(n) ∧ μ = poly(n)]</td>
<td>$\Omega(n^{n/5-1} / \mu)$ [μ = poly(n)]</td>
</tr>
<tr>
<td></td>
<td>Stochastic</td>
<td>$O(k(n/2)^k)$ [μ ≥ 8(n − 2k + 3)]</td>
<td>$O(n(20e^2)^{n/5})$ [μ ≥ 8(2n/5 + 1)]</td>
</tr>
</tbody>
</table>

For NSGA-II solving the bi-objective RealRoyalroad problem

**Expected running time**: $\Omega(n^{n/5-1} / \mu)$

**Stochastic**: accelerated by $(n/(20e^2))^{n/5} / (\mu n^2)$

$O(n(20e^2)^{n/5})$
By introducing randomness into population update, MOEAs can go across inferior regions between Pareto optimal solutions and sub-optimal solutions.

- **Deterministic**
  - prefers non-dominated solutions
  - if Pareto optimal solutions are far away from sub-optimal solutions in the solution space, easy to get trapped

- **Stochastic**
  - allows dominated solutions to participate in the evolutionary process
  - follows an easier path in the solution space to find Pareto optimal solutions from sub-optimal solutions
Our Main Results

Expected number of generations of SMS-EMOA and NSGA-II for solving the OneJumpZeroJump [Doerr and Zheng, AAAI’21] and bi-objective RealRoyalroad [Dang et al., AAAI’23] problems

<table>
<thead>
<tr>
<th>Population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMS-EMOA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$O(\mu n^k)$ \ [\mu \geq n - 2k + 3]</td>
<td>$O(\mu n^{n/5-2})$ \ [\mu \geq 2n/5 + 1]</td>
</tr>
<tr>
<td></td>
<td>$\Omega(n^k)$ \ [n - 2k = \Omega(n) \land \mu = \text{poly}(n)]</td>
<td>$\Omega(n^{n/5-1})$ \ [\mu = \text{poly}(n)]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(\mu^2 n^k/2^k/4)$ \ [\mu \geq 2(n - 2k + 4)]</td>
<td>$O(\mu^2 n^{n/5}/2^{n/20})$ \ [\mu \geq 2(2n/5 + 2)]</td>
</tr>
<tr>
<td><strong>NSGA-II</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>$\Omega(n^k/\mu)$ \ [n - 2k = \Omega(n) \land \mu = \text{poly}(n)]</td>
<td>$\Omega(n^{n/5-1}/\mu)$ \ [\mu = \text{poly}(n)]</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$O(k(n/2)^k)$ \ [\mu \geq 8(n - 2k + 3)]</td>
<td>$O(n(20e^2)^{n/5})$ \ [\mu \geq 8(2n/5 + 1)]</td>
</tr>
</tbody>
</table>

Green color: results of our IJCAI’23 work; Yellow color: extended results
By introducing randomness into population update, MOEAs can go across inferior regions around the Pareto optimal solutions more easily, thus facilitating to find the whole Pareto front.
Experiments

Estimated number of generations (average of 1000 independent runs) of SMS-EMOA and NSGA-II for solving the OneJumpZeroJump and bi-objective RealRoyalroad problems

OneJumpZeroJump with $k = 2$

<table>
<thead>
<tr>
<th>Bi-objective RealRoyalroad</th>
<th>SMS-EMOA</th>
<th>NSGA-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>$n = 5$</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>$n = 10$</td>
<td>704</td>
</tr>
<tr>
<td></td>
<td>$n = 15$</td>
<td>6572</td>
</tr>
<tr>
<td></td>
<td>$n = 20$</td>
<td>202558</td>
</tr>
<tr>
<td></td>
<td>$n = 25$</td>
<td>10792477</td>
</tr>
<tr>
<td>Stochastic</td>
<td>$n = 5$</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>$n = 10$</td>
<td>702</td>
</tr>
<tr>
<td></td>
<td>$n = 15$</td>
<td>5747</td>
</tr>
<tr>
<td></td>
<td>$n = 20$</td>
<td>144222</td>
</tr>
<tr>
<td></td>
<td>$n = 25$</td>
<td>5797043</td>
</tr>
</tbody>
</table>

Stochastic population update can bring a clear acceleration
Summary

- Prove that stochastic population update can significantly decrease the expected running time

<table>
<thead>
<tr>
<th>Population update</th>
<th>OneJumpZeroJump</th>
<th>Bi-objective RealRoyalroad</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Doerr and Zheng, AAAI’21]</td>
<td>[Dang et al., AAAI’23]</td>
<td></td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>$O(\mu n^k)$ [$\mu \geq n - 2k + 3$]</td>
<td>$O(\mu n^{n/5-2})$ [$\mu \geq 2n/5 + 1$]</td>
</tr>
<tr>
<td></td>
<td>$\Omega(n^k)$ [$n - 2k = \Omega(n) \land \mu = \text{poly}(n)$]</td>
<td>$\Omega(n^{n/5-1})$ [$\mu = \text{poly}(n)$]</td>
</tr>
<tr>
<td><strong>Stochastic</strong></td>
<td>$O(\mu^2 n^k/2^{k/4})$ [$\mu \geq 2(n - 2k + 4)$]</td>
<td>$O(\mu^2 n^{n/5}/2^{n/20})$ [$\mu \geq 2(2n/5 + 2)$]</td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>$\Omega(n^k/\mu)$ [$n - 2k = \Omega(n) \land \mu = \text{poly}(n)$]</td>
<td>$\Omega(n^{n/5-1}/\mu)$ [$\mu = \text{poly}(n)$]</td>
</tr>
<tr>
<td><strong>Stochastic</strong></td>
<td>$O(k(n/2)^k)$ [$\mu \geq 8(n - 2k + 3)$]</td>
<td>$O(n(20e^2)^{n/5})$ [$\mu \geq 8(2n/5 + 1)$]</td>
</tr>
</tbody>
</table>

- Challenge the common practice of MOEAs, i.e., greedy and deterministic population update
- Encourage the exploration of developing new MOEAs in the area
Similar observation

The recent empirical study [Liang, Li and Lehre, GECCO’23] shows that non-elitist population update (the generated offspring solutions form the next population directly) can be helpful.

NSGA-II vs. Non-elitist MOEA (NE-MOEA)

On knapsack with 100 items

On NK-Landscape with $n = 200$ and $k = 10$
Similar observation

A subsequent theoretical study [Zheng and Doerr, AAAI’24] considers SMS-EMOA solving the many-objective $m$OneJumpZeroJump problem, and confirms the same advantage of using stochastic population update.

A direct extension of OneJumpZeroJump, which has $m$ objectives.

Expected number of generations for $\mu = \Theta(M)$:

- Deterministic population update: $O(M^2 n^k)$
- Stochastic population update: $O((Mk^{1/2}/2^{k/2}) \cdot M^2 n^k)$

exponential if $m$ is not too large (e.g., constant), and $k$ is large (e.g., $\Theta(n)$).

Size of the Pareto front, i.e., $(2n/m - 2k + 3)^{m/2}$.
Finally, I will give a brief review of running time analysis of MOEAs
GSEMO Solving Multi-objective Synthetic Problems

**GSEMO**: a simple MOEA which employs bit-wise mutation to generate an offspring solution in each iteration and keeps the non-dominated solutions in the population

**SEMO**: a counterpart of GSEMO which employs one-bit mutation

Summary of GSEMO and SEMO solving multi-objective synthetic problems:

<table>
<thead>
<tr>
<th>Problem</th>
<th>GSEMO</th>
<th>SEMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeadingOnesTrailingZeroes</td>
<td>$O(n^3)$ [Giel, CEC’03] (\Omega(n^2/p)) [Doerr et al., CEC’13]</td>
<td>$\Theta(n^3)$ [Laumanns et al, TEC’04]</td>
</tr>
<tr>
<td>CountOnesCountZeroes</td>
<td>$O(n^2 \log n)$ [Qian et al., AIJ’13]</td>
<td>$O(n^2 \log n)$ [Laumanns et al, TEC’04]</td>
</tr>
<tr>
<td>OneMinMax</td>
<td>$O(n^2 \log n)$ [Giel and Lehre, ECJ’10]</td>
<td>$O(n^2 \log n)$ [Giel and Lehre, ECJ’10]</td>
</tr>
<tr>
<td>OneJumpZeroJump</td>
<td>$O((n - 2k)n^k)$ [Zheng and Doerr, ECJ’23]</td>
<td>cannot find the Pareto front [Zheng and Doerr, ECJ’23]</td>
</tr>
</tbody>
</table>
GSEMO Solving Multi-objective Combinatorial Problems

**GSEMO**: a simple MOEA which employs bit-wise mutation to generate an offspring solution in each iteration and keeps the non-dominated solutions in the population

**GSEMO**
- multi-objective minimum spanning tree [Neumann, EJOR’07; Qian et al., AIJ’13]

**DEMO** (diversity evolutionary multiobjective optimizer)
- multi-objective shortest path [Horoba, FOGA’09; Neumann and Theile, PPSN’10]

**GSEMO**
- multi-objective knapsack [Laumanns et al, NC’04]
Effectiveness of Some Strategies for MOEAs

Brief review:

- greedy selection [Laumanns et al., TEC’04]
- crossover [Qian et al., AIJ’13; Dang et al., AAAI’23]
- diversity-based parent selection [Osuna et al., 2020]
- Selection hyper-heuristics [Qian et al., PPSN’16]
- diversity [Friedrich et al., TCS’10]
- fairness [Laumanns et al., TEC’04; Friedrich et al., 2011]

For example, recombination can accelerate the filling of the Pareto front by recombining diverse Pareto optimal solutions [Qian et al., AIJ’13]
Running Time Analysis of Practical MOEAs

[Zheng, Liu and Doerr, AAAI’22] analyzed the expected running time of NSGA-II (without crossover) for the first time

<table>
<thead>
<tr>
<th></th>
<th>Upper bound</th>
<th>Lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeadingOnesTrailingZeroes</td>
<td>$O(\mu n^2)$, if $\mu \geq 4(n + 1)$</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>OneMinMax</td>
<td>$O(\mu n \log n)$, if $\mu \geq 4(n + 1)$</td>
<td>exponential, if $\mu = n + 1$</td>
</tr>
</tbody>
</table>

[Bian and Qian, PPSN’22] analyzed the standard NSGA-II which uses binary tournament selection, one-point crossover and bit-wise mutation

• solves LeadingOnesTrailingZeroes in $O(\mu n^2)$
• expected running time can be improved to $O(\mu n)$ if using stochastic tournament selection strategy
Running Time Analysis of Practical MOEAs

Other results of NSGA-II:

- **OneMinMax**: $\Omega(\mu n \log n)$, if $\mu = c(n + 1)$ for some $c \geq 4$ [Doerr and Qu, AAAI’23]
- **OneJumpZeroJump**:
  
  $\Theta(N n^k)$, where $\mu \geq 4(n - 2k + 3)$ [Doerr and Qu, TEC’23; Doerr and Qu, AAAI’23]

  ➢ bit-wise mutation and uniform crossover: $O\left(\frac{N^2 (Cn)^k}{(k-1)!}\right)$, where $\mu = c(n - 2k + 3)$ and $c > 4$, $C = \left(\frac{4c}{c-4}\right)^2$ [Doerr and Qu, AAAI’23]

NSGA-II solving bi-objective MST [Cerf et al., IJCAI’23]: $O(m^2 n w_{\max} \log n w_{\max})$

NSGA-III solving 3OneMinMax [Wietheger and Doerr, IJCAI’23]: $O(n \log n)$
Running Time Analysis of Practical MOEAs

Analysis of SIBEA (a simple indicator based MOEA):
• LeadingOnesTrailingZeroes: $O(\mu n^2)$ [Brockhoff et al., PPSN’08]
• OneMinMax: $O(\mu n \log n)$ [Nguyen et al., TCS’15]

SMS-EMOA solving OneJumpZeroJump: $O(\mu n^k) \land \Omega(n^k)$ [Bian et al., IJCAI’23]

Our work contributes to running time analysis of a major type of MOEAs, i.e., combining non-dominated sorting and quality indicators, for the first time

Analysis of MOEA/D:
• MOEA/D with Tchbycheff decomposition [Li et al., TEC’16]
  ➢ CountOnesCountZeroes: $O(n^2 \log n)$
  ➢ LPTNO (an extension of LeadingOnesTrailingZeroes): $O(n^3)$
• comparison of different decomposition methods [Huang et al., IJCAI’21]
Running Time Analysis of MOEAs under noise

Posterior noise [Dang et al., GECCO’23]:
• noise model: $(\delta, p)$-Bernoulli noise model $(\delta > f_{\text{max}} - f_{\text{min}})$

\[ f(x) = \begin{cases} 
   f(x) + \delta \cdot 1, & \text{with probability } p, \\
   f(x), & \text{otherwise} 
\end{cases} \]

• result: the expected running time of GSEMO and NSGA-II solving LeadingOnesTrailingZeroes is exponential and $O(\mu n^2)$, respectively

Prior noise [Dinot et al., IJCAI’23]:
• one-bit noise: flips a uniformly chosen bit of a solution with prob. $p$ before evaluation
• result: the expected running time of SEMO without reevaluation solving OneMinMax is $O(n^2 \log n)$, better than that using reevaluation
Conclusion

Previous running time analysis of MOEAs mainly focuses on simple MOEAs, while recently, researchers have started to analyze practical MOEAs.

- (G)SEMO on synthetic problems
- (G)SEMO on combinatorial problems
- Effectiveness of some strategies based on (G)SEMO
- Analysis of practical MOEAs
- Analysis of MOEAs under noise

Many theoretical works can be done in MOEAs

Thank you