Multi-objective Evolutionary Learning
Advances in Theories and Algorithms

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

LAMDA Group
Nanjing University, China
Machine learning aims at learning generalizable models from data

Thus, a machine learning problem is often formulated as an optimization problem

A typical machine learning process [Domingos, CACM’12]
The resulting optimization problems are usually complicated, where the objective can be non-differentiable, non-continuous, non-unique and have many local optima.

**Selective ensemble**
- Max: generalization performance
- Min: number of selected learners

**Neural architecture search**
- Max: accuracy
- Min: computation cost
Multi-objective Optimization

Multi-objective optimization tries to optimize multiple objectives simultaneously:

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

**x dominates z:**
\[ f_1(x) < f_1(z) \land f_2(x) < f_2(z) \]

**x is incomparable with y:**
\[ f_1(x) > f_1(y) \land f_2(x) < f_2(y) \]

Much more complicated than single-objective optimization.
The resulting optimization problems are usually complicated, where the objective can be non-differentiable, non-continuous, non-unique and have many local optima.

Selective ensemble
- Max: generalization performance
- Min: number of selected learners

Neural architecture search
- Max: accuracy
- Min: computation cost

Thus, the conventional optimization algorithms such as gradient descent may fail, while other powerful optimization algorithms are needed.
Evolutionary Algorithms

Evolutionary algorithms (EAs) are a kind of randomized heuristic optimization algorithms, inspired by nature evolution (reproduction with variation + nature selection)

In 1950, Turing described how evolution might be used for his optimization:

building intelligent machine

“Structure of the child machine = Hereditary material
Changes of the child machine = Mutations
Judgment of the experimenter = Natural selection”

Evolutionary algorithms (EAs) are a kind of randomized heuristic optimization algorithms, inspired by nature evolution (reproduction with variation + nature selection)

Many variants: genetic algorithm, evolutionary strategy, genetic programming, ...

EAs also include some heuristics inspired from nature phenomena

- particle swarm optimization
- ant colony optimization
A typical evolutionary process

arg max$_s$ $f(s)$
A typical evolutionary process

arg max$_S$ $f(S)$
A typical evolutionary process

\[ \text{arg max}_s f(s) \]
A typical evolutionary process

\[ \text{arg max}_s f(s) \]
A typical evolutionary process

\[ \text{arg max}_s f(s) \]
A typical evolutionary process

\[ \arg \max_s f(s) \]
A typical evolutionary process

arg max$_S f(S)$
Evolutionary Algorithms

The general structure of EAs

- Initial population
- Mutation & recombination
- Offspring solutions
- Fitness evaluation & selection
- New population
- Stop criterion
- Solution representation

- No requirement on the objective
- Population-based search

Thus, EAs can be applied to solve complicated optimization problems

- non-differentiable, non-continuous
- without explicit objective formulation
- multiple objective functions
Evolutionary Algorithms

The general structure of EAs

- No requirement on the objective
- Population-based search

Thus, EAs can be applied to solve complicated optimization problems

- non-differentiable, non-continuous
- without explicit objective formulation
- multiple objective functions

Multi-objective EAs (MOEAs)

- e.g., NSGA-II [Deb et al., TEC’02]

Google scholar: 41646
Applications of Evolutionary Algorithms

High-speed train head design

Series 700 evolve Series N700

save 19% energy

Technological overview of the next generation Shinkansen high-speed train Series N700

M. Ueno¹, S. Usui¹, H. Tanaka¹, A. Watanabe²

¹Central Japan Railway Company, Tokyo, Japan, ²West Japan Railway Company, Osaka, Japan

waves and other issues related to environmental compatibility such as external noise. To combat this, an aero double-wing-type has been adopted for nose shape (Fig. 3). This nose shape, which boasts the most appropriate aerodynamic performance, has been newly developed for railway rolling stock using the latest analytical technique (i.e., genetic algorithms) used to develop the main wings of airplanes. The shape resembles a bird in flight, suggesting a feeling of boldness and speed.

On the Tokaido Shinkansen line, Series N700 cars save 19% energy than Series 700 cars, despite a 30% increase in the output of their traction equipment for higher-speed operation (Fig. 4).

This is a result of adopting the aerodynamically excellent nose shape, reduced running resistance, and other improvements.

Antenna design

38% efficiency evolve

93% efficiency

Computer-Automated Evolution of an X-Band Antenna for NASA’s Space Technology 5 Mission

Gregory S. Hornby
University Affiliated Research Center, NASA Ames Research Park, UC Santa Cruz at Moffett Field, California, 94035

Gregory.S.Hornby@nasa.gov

Jason D. Lohn
Carnegie Mellon University, NASA Ames Research Park and Moffett Field, California 94035

Jason.Lohn@cmu.edu

this, different combinations of the two evolved antennas and the QHA were tried on the ST5 mock-up and measured in an anechoic chamber. With two QHAs 38% efficiency was achieved, using a QHA with an evolved antenna resulted in 80% efficiency, and using two evolved antennas resulted in 93% efficiency. Here “efficiency” means how...
Applications of Evolutionary Algorithms

The Nobel Prize in Chemistry 2018

“Evolution—the adaption of species to different environments—has created an enormous diversity of life. Frances Arnold has used the same principles—genetic change and selection—to develop proteins that solve humankind’s chemical problems. In 1993, Arnold conducted the first directed evolution of enzymes, which are proteins that catalyze chemical reactions. The uses of her results include more environmentally friendly manufacturing of chemical substances, such as pharmaceuticals, and the production of renewable fuels.”

Protein design

[Arnold, 1998]
Applications of Evolutionary Algorithms

自然科学四大基础科学问题之一：生命起源与演化

利用化石记录

三叶虫、笔石、珊瑚、腕足...

重现生命演化历史

地层剖面数据

海量化石记录数据

全球第一条高精度海洋生物多样性变化曲线

南京大学地球科学与工程学院研究成果

中国的地层剖面数据
3122个剖面
11268个物种

演化算法

地层剖面

生物多样性变化曲线

地质时期

Science："新的数据集和方法，推动整个演化生物学的变革"
Nature："古生物学家以惊人的细节绘制地球3亿年历史"

2020年中国十大科技进展
Multi-objective Evolutionary Learning

Multi-objective evolutionary learning applies MOEAs to solve multi-objective optimization problems in machine learning.

Multi-objective evolutionary learning has yielded encouraging empirical outcomes, e.g.,

- **Evolutionary selective ensemble**
  - Achieves smaller error by using fewer learners [Zhou et al., AI’02]

- **Evolutionary neural architecture search**
  - Achieves competitive performance to the hand-designed models [Real et al., ICML’17]

Why not popular?
Multi-objective Evolutionary Learning

The theoretical foundation of MOEAs is underdeveloped

Arthur Intelligence: A Modern Approach

“... At present, it is not clear whether the appeal of genetic algorithms arises from their performance or from their aesthetically pleasing origins in the theory of evolution. Much work remains to be done to identify the conditions under which genetic algorithm perform well.”

L. Valiant

Evolvability

“there has existed no theory that would explain quantitatively which mechanisms can so evolve in realistic population sizes within realistic time ...”

Theoretical analysis is very difficult

- MOEAs: highly randomized and complex
- Problems: complicated
Outline

- Introduction
- Theoretical analysis tools for MOEAs
  - Theoretical perspectives of MOEAs
    - Recombination operator, constrained optimization, noisy optimization
  - Multi-objective evolutionary learning algorithms
    - Selective ensemble, subset selection
- Conclusion
Running Time Complexity

Objective $f$

$OPT$

$f(s)$

Running time $\tau$: 
#fitness evaluations until finding desired solutions for the first time

#fitness evaluations

the process with the highest cost of EA 

e.g., model evaluation

data

Training
Testing
An Example

Hard to be analyzed directly

EA with recombination

...
An Example

EA with recombination

Hard to be analyzed directly

EA without recombination

Easier to be analyzed
An Example

Hard to be analyzed directly

EA with recombination

Compare

EA without recombination

Easier to be analyzed

Simplify the analysis

One-step time difference

+ ... = Total time difference

+ ... = Expected running time

Expected running time
An Example

Hard to be analyzed directly

Complex EA

Compare

Simple EA

Easier to be analyzed

Simplify the analysis

One-step time difference

+ 

+ \ldots =

Expected running time

\|

Total time difference

+ 

\ldots

Expected running time

\ldots
Switch Analysis

Model an EA process as a Markov chain

Switch Analysis

Model an EA process as a Markov chain

The generation of the next population only depends on the current population

Markov property

\[ P(\xi_{t+1} \mid \xi_t, \ldots, \xi_0) = P(\xi_{t+1} \mid \xi_t) \]

Switch Analysis

Target chain $\xi$

Hard to be analyzed directly

$\xi_0 \rightarrow \xi_1 \rightarrow \xi_2 \rightarrow \xi_3 \rightarrow \xi_4 \rightarrow \xi_5 \rightarrow \cdots$

Switch Analysis

Hard to be analyzed directly

Target chain $\xi$

Reference chain $\xi'$

Easier to be analyzed

Switch Analysis

Hard to be analyzed directly

Target chain $\xi$

Reference chain $\xi'$

One-step time difference $\rho_0 + \rho_1 + \rho_2 + \rho_3 + \rho_4 + \cdots$

Easier to be analyzed

How to estimate one-step time difference $\rho_t$?

$\mathbb{E} [\tau']$ = Total time difference

$\mathbb{E} [\tau]$ = Expected running time

$\mathbb{E} [\tau']$ = Expected running time

Switch Analysis

How to estimate one-step time difference $\rho_t$?

Target chain $\xi$

$\xi_0 \rightarrow \xi_1 \rightarrow \xi_2 \rightarrow \xi_3 \rightarrow \xi_4 \rightarrow \ldots$

Intermediate chains

$\xi^3$

$\xi_0 \rightarrow \xi_1 \rightarrow \xi_2 \rightarrow \xi_3$

$\xi^2$

$\xi_0 \rightarrow \xi_1 \rightarrow \xi_2$

mapping $\phi$

$\xi^1$

$\xi_0 \rightarrow \xi_1 \rightarrow \phi$

$\xi'_{0} \rightarrow \xi'_{1} \rightarrow \xi'_{2} \rightarrow \xi'_{3} \rightarrow \ldots$

Reference chain $\xi'$

$\xi'_{0} \rightarrow \xi'_{1} \rightarrow \xi'_{2} \rightarrow \xi'_{3} \rightarrow \xi'_{4} \rightarrow \ldots$

Switch Analysis

How to estimate one-step time difference $\rho_{t}$?

Target chain $\xi$

Intermediate chains

Reference chain $\xi'$

Time difference between adjacent intermediate chains

One-step time difference

$\mathbb{E}[\tau] - \mathbb{E}[\tau^\infty] = \rho_\infty$

$\mathbb{E}[\tau^{t+1}] - \mathbb{E}[\tau^t] = \rho_t$

$\mathbb{E}[\tau^3] - \mathbb{E}[\tau^2] = \rho_2$

$\mathbb{E}[\tau^2] - \mathbb{E}[\tau^1] = \rho_1$

$\mathbb{E}[\tau^1] - \mathbb{E}[\tau'] = \rho_0$

Switch Analysis

How to estimate one-step time difference $\rho_t$?

Target chain $\xi$

Intermediate chains

Reference chain $\xi'$

Time difference between adjacent intermediate chains

$\mathbb{E}[\tau] - \mathbb{E}[\tau^\infty] = \rho_\infty$

$\mathbb{E}[\tau^{t+1}] - \mathbb{E}[\tau^t] = \rho_t$

$\mathbb{E}[\tau^3] - \mathbb{E}[\tau^2] = \rho_2$

$\mathbb{E}[\tau^2] - \mathbb{E}[\tau^1] = \rho_1$

$\mathbb{E}[\tau^1] - \mathbb{E}[\tau'] = \rho_0$

Total time difference

$\mathbb{E}[\tau] - \mathbb{E}[\tau']$

Switch Analysis

How to estimate one-step time difference $\rho_t$?

Target chain $\xi$

Reference chain $\xi'$

Theorem (Switch Analysis). Given two absorbing Markov chains $\xi \in \mathcal{X}$ and $\xi' \in \mathcal{Y}$, a series of values $\{\rho_t \in \mathbb{R}\}_{t=0}^{+\infty}$ with $\rho = \sum_{t=0}^{+\infty} \rho_t$ and a right (or left)-aligned mapping $\phi: \mathcal{X} \rightarrow \mathcal{Y}$, if $\mathbb{E}[\tau | \xi_0 \sim \pi_0]$ is finite and $\forall t: \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \pi_t(x) P(\xi_{t+1} \in \phi^{-1}(y) | \xi_t = x) \mathbb{E}[\tau' | \xi_0' = y] \leq (or \geq) \sum_{u, y \in \mathcal{Y}} \pi_t(u) P(\xi_1' = y | \tau')$

How to estimate one-step time difference $\rho_t$?

Application of Switch Analysis

Example: Analyze GSEMO solving the $m$COCZ problem

GSEMO:
1. $s := \text{randomly selected from } \{0,1\}^n$; $P := \{s\}$
2. Repeat until some termination criterion is met
3. Choose $s$ from $P$ uniformly at random
4. apply bit-wise mutation on $s$ to generate $s'$
5. if $\nexists z \in P$ such that $z > s'$
6. $P := (P - \{z \in P | s' \geq z\}) \cup \{s'\}$

$m$COCZ: $\max_{s \in \{0,1\}^n} (f_1(s), f_2(s), \ldots, f_m(s))$

Previous results: $O(n^{m+1})$ [Laumanns, Thiele and Zitzler, TEC’04]

Switch analysis: $O(n^m)$ [Bian et al., IJCAI’18]

[Bian, Qian and Tang, IJCAI 2018]
Outline

- Introduction
- Theoretical analysis tools for MOEAs
- Theoretical perspectives of MOEAs
  - Recombination operator, constrained optimization, noisy optimization
- Multi-objective evolutionary learning algorithms
  - Selective ensemble, subset selection
- Conclusion
Recombination

Mutation and recombination are two characterizing features of EAs

Example of mutation

Parent: 1 0 1 1 1 0 0 0
Offspring: 1 0 1 1 0 0 0 0

simulates the gene altering of a chromosome in biological mutation

Example of recombination

Parent1: 1 0 1 1 1 0 0 0
Parent2: 0 0 1 0 1 0 1 0
Offspring1: 1 0 1 1 1 0 1 0
Offspring2: 0 0 1 0 1 0 0 0

More complicated

simulates the chromosome exchange phenomena in zoogamy reproductions
Recombination

Most theoretical studies focused on EAs with mutation, while only a few included recombination, which is difficult to be analyzed due to the irregular behavior.

Mainly focused on single-objective optimization

How about the influence of recombination for multi-objective optimization?

- Often involved in machine learning
- More complex than single-objective optimization
Recombination

Our result:

Recombination can accelerate the filling of the Pareto front by recombining diverse Pareto optimal solutions

Unique to multi-objective optimization

[Qian et al., Artificial Intelligence 2013, ACM GECCO’11 Best Theory Paper Award]
Recombination

Our result:

Recombination can accelerate the filling of the Pareto front by recombining diverse Pareto optimal solutions

Unique to multi-objective optimization

Example: MOEA solving the LOTZ Problem

Expected running time \( \Theta(n^3) \) recombination \( \rightarrow \) \( \Theta(n^2) \)

[Qian et al., Artificial Intelligence 2013, ACM GECCO’11 Best Theory Paper Award]
Constrained Optimization

The optimization problems in machine learning often come with constraints. e.g., to avoid overfitting, one often needs to minimize the error of a model, while constraining the model complexity.

General formulation of constrained optimization:

\[
\min_{s \in S} f(s) \quad \text{objective function}
\]

s. t. \[
g_i(s) = 0, \quad 1 \leq i \leq q; \quad \text{equality constraints}
\]

\[
h_i(s) \leq 0, \quad q + 1 \leq i \leq m \quad \text{inequality constraints}
\]

The goal is to find a feasible solution minimizing the objective \( f \).

Remark: A solution is (in)feasible if it does (not) satisfy the constraints.
Constrained Optimization

How to deal with constraints for EAs?

The penalty function method transforms the original constrained optimization problem into an unconstrained optimization problem [Hadj-Alouane and Bean, OR’97]

\[
\begin{align*}
\text{constrained} & \quad \min f(s) \\
\text{s.t.} & \quad g_i(s) = 0, \quad 1 \leq i \leq q; \\
& \quad h_i(s) \leq 0, \quad q + 1 \leq i \leq m
\end{align*}
\]

\[
\begin{align*}
\text{unconstrained} & \quad \min f(s) + \lambda \sum_{i=1}^{m} f_i(s) \\
\text{constraint violation degree} & \quad f_i(s) = \begin{cases} \
g_i(s) & 1 \leq i \leq q \\ 
\max\{0, h_i(s)\} & q + 1 \leq i \leq m
\end{cases}
\end{align*}
\]
Constrained Optimization

How to deal with constraints for EAs?

Pareto optimization transforms the original constrained optimization problem into a bi-objective optimization problem [Coello Coello, 2002]

Constrained optimization: optimize $f$ under some constraints

Transform

Bi-objective optimization: optimize $f$ and a constraint-related objective simultaneously

An example

$$\min f(s)$$
$$\text{s.t. } g_i(s) = 0, \quad 1 \leq i \leq q;$$
$$\quad h_i(s) \leq 0, \quad q + 1 \leq i \leq m$$

$$\min (f(s), \sum_{i=1}^{m} f_i(s))$$

constraint violation degree

$$f_i(s) = \begin{cases} |g_i(s)| & 1 \leq i \leq q \\ \max\{0, h_i(s)\} & q + 1 \leq i \leq m \end{cases}$$
Constrained Optimization

How to deal with constraints for EAs?

Pareto optimization transforms the original constrained optimization problem into a bi-objective optimization problem [Coello Coello, 2002]
Pareto optimization can be better by exploiting infeasible solutions

- **Penalty function**
  - prefers feasible solutions
  - if initialized far from the global optimum, easy to get trapped by local optimum

- **Pareto optimization**
  - allows infeasible solutions to participate in the evolutionary process naturally
  - follows a shortcut from infeasible space to feasible space to find good solutions

---

[Qian, Yu and Zhou, IJCAI 2015]
Constrained Optimization

Our result: Pareto optimization can be better by exploiting infeasible solutions

Example: Minimum set cover problem

One of Karp's 21 NP-complete problems

Expected running time

Penalty function exponential

Pareto optimization $O(mn (\log n + \log w_{max} + m))$
Noisy Optimization

The objective (i.e., fitness) evaluation in machine learning is often disturbed by noise.

Model evaluation

Exact fitness value: $f(s)$

Noisy fitness value: $f^n(s)$ e.g., $f(s) + N(0, \sigma^2)$

How to reduce the negative influence of noise?

Threshold selection [Markon et al., CEC’01]

accepts an offspring solution only if its fitness becomes better by at least a threshold $\tau$

$$f^n(s) > f^n(s') \iff f^n(s) > f^n(s') + \tau$$

Its effectiveness is not yet clear
Noisy Optimization

Our result: Threshold selection can bring robustness against noise

reduces the risk of deleting a good solution

Example: (1+1)-EA solving the OneMax problem under noise

Expected running time: exponential $\xrightarrow{\text{threshold selection}}$ polynomial

[Qian, Yu and Zhou, Evolutionary Computation 2018]
Outline

- Introduction
- Theoretical analysis tools for MOEAs
- Theoretical perspectives of MOEAs
  - Recombination operator, constrained optimization, noisy optimization
- Multi-objective evolutionary learning algorithms
  - Selective ensemble, subset selection
- Conclusion
Selective Ensemble

Ensemble learning

- Learner $h_1$
- Learner $h_2$
- Learner $h_3$
- ...  
- Learner $h_n$

$\rightarrow$ Combination $\rightarrow y$

• achieves better performance than a single learner

[Zhou, 2012]

Selective ensemble

- Learner $h_1$
- Learner $h_2$
- Learner $h_3$
- ...  
- Learner $h_n$

$\rightarrow$ Combination $\rightarrow y$

• achieves better performance than the complete ensemble
• reduces storage and improve efficiency

Chapter 6 of [Zhou, 2012]
Selective Ensemble

Selective ensemble naturally bears two goals

- maximize the generalization performance
- minimize the number of selected learners

Previous methods can be roughly categorized into two branches

- **Ordering-based selective ensemble methods (OSE):**
  - e.g., error minimization [Margineantu and Dietterich, ICML’97], diversity-like criterion maximization [Martínez-Munóz et al., TPAMI’09], combined criterion [Li et al., ECML’12]

- **Single-objective optimization-based methods (SOSE):**
  - e.g., genetic algorithms [Zhou et al., AIJ’02]

\[
\max_{s \in \{0,1\}^n} f(s)
\]

A subset of learners | Generalization performance | Genetic algorithm
----------------------|---------------------------|-----------------------
No theoretical guarantee
Pareto Optimization for Selective Ensemble

Introduce the Pareto optimization algorithm for selective ensemble (POSE)

Algorithm 13.3 POSE Algorithm

Input: trained individual learners $H = \{h_i\}_{i=1}^n$; objective $f : 2^H \rightarrow \mathbb{R}$; criterion $\text{eval}$
Output: subset of $H$

Process:
1: let $g(s) = (f(s), -|s|)$ be the bi-objective formulation;
2: let $s$ = a solution uniformly and randomly selected from $\{0, 1\}^n$;
3: let $P = \{s\}$;
4: while criterion is not met do
5:    select a solution $s$ from $P$ uniformly at random;
6:    apply bit-wise mutation on $s$ to generate $s'$;
7:    if $\exists z \in P$ such that $z \succ s'$ then
8:        $P = (P \setminus \{z \in P \mid s' \succeq z\}) \cup \{s'\}$;
9:        $Q = \text{VDS}(f, s')$;
10:       for $q \in Q$
11:          if $\exists z \in P$ such that $z \succ q$ then
12:              $P = (P \setminus \{z \in P \mid q \succeq z\}) \cup \{q\}$
13:       end if
14:    end if
15: end while
16: return $\text{arg min}_{s \in P} \text{eval}(s)$

Bi-objective formulation:

$$\max_{s \in \{0, 1\}^n} (f(s), -|s|_1)$$

Max generalization performance  Min #learners

Initialization: randomly generate a solution, put it into the population $P$

Reproduction: pick a solution randomly from $P$, and mutate it to generate a new one

Evaluation & selection: if the new solution is not dominated, put it and its good neighbors into $P$

Output: select a final solution

[Qian, Yu and Zhou, AAAI 2015]
Theoretical Results

POSE can do better than ordering-based methods

**Theorem 1.** For any objective and any size, POSE within $O(n^4 \log n)$ expected running time can find a solution weakly dominating that generated by OSE at the fixed size.

**Theorem 2.** For Example 13.1, OSE using Eq. (13.2) finds a solution with objective vector $(\leq 0, \leq -3)$ where the two equalities never hold simultaneously, whereas POSE finds a solution with objective vector $(0, -3)$ in $O(n^4 \log n)$ expected running time.

POSE can do better than single-objective optimization-based methods

**Theorem 3.** For Example 13.2, OSE using Eq. (13.2) finds the optimal solution in $O(n^2)$ running time, whereas the running time of SOSE is at least $2^{\Omega(n)}$ with probability $1 - 2^{-\Omega(n)}$.

The first evolutionary learning algorithm with theoretical guarantee!

[Qian, Yu and Zhou, AAAI 2015]
### Empirical Results

#### Pruning bagging base learners with size 100

<table>
<thead>
<tr>
<th>Data set</th>
<th>POSE</th>
<th>Bagging</th>
<th>BL</th>
<th>RE</th>
<th>Kappa</th>
<th>CP</th>
<th>MD</th>
<th>DREP</th>
<th>EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>australian</td>
<td>0.144±0.020</td>
<td>0.143±0.017</td>
<td>0.142±0.023</td>
<td>0.143±0.020</td>
<td>0.143±0.020</td>
<td>0.145±0.022</td>
<td>0.143±0.019</td>
<td>0.143±0.020</td>
<td></td>
</tr>
<tr>
<td>breast-cancer</td>
<td>0.275±0.041</td>
<td>0.279±0.037</td>
<td>0.298±0.046</td>
<td>0.277±0.031</td>
<td>0.287±0.033</td>
<td>0.282±0.043</td>
<td>0.295±0.044</td>
<td>0.275±0.036</td>
<td></td>
</tr>
<tr>
<td>disorders</td>
<td>0.304±0.039</td>
<td>0.327±0.047</td>
<td>0.365±0.047</td>
<td>0.320±0.044</td>
<td>0.326±0.042</td>
<td>0.306±0.039</td>
<td>0.337±0.035</td>
<td>0.316±0.045</td>
<td></td>
</tr>
<tr>
<td>heart-statlog</td>
<td>0.197±0.037</td>
<td>0.045±0.019</td>
<td>0.088±0.021</td>
<td>0.010±0.003</td>
<td>0.013±0.005</td>
<td>0.046±0.008</td>
<td>0.043±0.009</td>
<td>0.035±0.006</td>
<td></td>
</tr>
<tr>
<td>house-votes</td>
<td>0.013±0.005</td>
<td>0.073±0.012</td>
<td>0.048±0.014</td>
<td>0.048±0.012</td>
<td>0.057±0.014</td>
<td>0.048±0.009</td>
<td>0.053±0.011</td>
<td>0.053±0.011</td>
<td></td>
</tr>
</tbody>
</table>

**Baseline methods**

- **POSE** achieves the smallest error on 60% (12/20) of the data sets, while other methods are no more than 35% (7/20).

**Comparison on test error**

- **POSE** is better than any other method on more than 60% (12.5/20) data sets.

<table>
<thead>
<tr>
<th>Data set</th>
<th>POSE</th>
<th>Bagging</th>
<th>BL</th>
<th>RE</th>
<th>Kappa</th>
<th>CP</th>
<th>MD</th>
<th>DREP</th>
<th>EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>australian</td>
<td>0.144±0.020</td>
<td>0.143±0.017</td>
<td>0.142±0.023</td>
<td>0.143±0.020</td>
<td>0.143±0.020</td>
<td>0.145±0.022</td>
<td>0.143±0.019</td>
<td>0.143±0.020</td>
<td></td>
</tr>
<tr>
<td>breast-cancer</td>
<td>0.275±0.041</td>
<td>0.279±0.037</td>
<td>0.298±0.046</td>
<td>0.277±0.031</td>
<td>0.287±0.033</td>
<td>0.282±0.043</td>
<td>0.295±0.044</td>
<td>0.275±0.036</td>
<td></td>
</tr>
<tr>
<td>disorders</td>
<td>0.304±0.039</td>
<td>0.327±0.047</td>
<td>0.365±0.047</td>
<td>0.320±0.044</td>
<td>0.326±0.042</td>
<td>0.306±0.039</td>
<td>0.337±0.035</td>
<td>0.316±0.045</td>
<td></td>
</tr>
<tr>
<td>heart-statlog</td>
<td>0.197±0.037</td>
<td>0.045±0.019</td>
<td>0.088±0.021</td>
<td>0.010±0.003</td>
<td>0.013±0.005</td>
<td>0.046±0.008</td>
<td>0.043±0.009</td>
<td>0.035±0.006</td>
<td></td>
</tr>
<tr>
<td>house-votes</td>
<td>0.013±0.005</td>
<td>0.073±0.012</td>
<td>0.048±0.014</td>
<td>0.048±0.012</td>
<td>0.057±0.014</td>
<td>0.048±0.009</td>
<td>0.053±0.011</td>
<td>0.053±0.011</td>
<td></td>
</tr>
</tbody>
</table>

**Ordering-based methods**

- ● and ○ denote that POSE is significantly better and worse, respectively, by the t-test with confidence level 0.05.

**Single-objective optimization-based methods**

- previous EA without theoretical guarantee

- POSE is never significantly worse

---

*Qian, Yu and Zhou, AAAI 2015 [Qian, Yu and Zhou, AAAI 2015]*
Empirical Results

Pruning bagging base learners with size 100

<table>
<thead>
<tr>
<th>Data set</th>
<th>POSE</th>
<th>RE</th>
<th>Kappa</th>
<th>CP</th>
<th>MD</th>
<th>DREP</th>
<th>EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>australian</td>
<td>10.6±4.2</td>
<td>12.5±6.0</td>
<td>14.7±12.6</td>
<td>11.0±9.7</td>
<td>8.5±14.8</td>
<td>11.7±4.7</td>
<td>41.9±6.7○</td>
</tr>
<tr>
<td>breast-cancer</td>
<td>8.4±3.5</td>
<td>8.7±3.6</td>
<td>26.1±21.7●</td>
<td>8.8±12.3</td>
<td>7.8±15.2</td>
<td>9.2±3.7</td>
<td>44.6±6.6●</td>
</tr>
<tr>
<td>disorders</td>
<td>14.7±4.2</td>
<td>13.8±14.3</td>
<td>26.7±16.2</td>
<td>15.3±10.6</td>
<td>17.7±20.8</td>
<td>12.9±5.8</td>
<td>42.0±6.9○</td>
</tr>
<tr>
<td>heart-statlog</td>
<td>9.3±2.3</td>
<td>9.3±2.3</td>
<td>15.3±10.6</td>
<td>17.7±20.8</td>
<td>12.9±5.8</td>
<td>42.0±6.9○</td>
<td></td>
</tr>
<tr>
<td>ionosphere</td>
<td>5.2±2.2</td>
<td>5.2±2.2</td>
<td>5.2±2.2</td>
<td>5.2±2.2</td>
<td>5.2±2.2</td>
<td>5.2±2.2</td>
<td>5.2±2.2</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>4.2±1.8</td>
<td>4.2±1.8</td>
<td>4.2±1.8</td>
<td>4.2±1.8</td>
<td>4.2±1.8</td>
<td>4.2±1.8</td>
<td>4.2±1.8</td>
</tr>
<tr>
<td>letter-ah</td>
<td>5.0±1.9</td>
<td>5.0±1.9</td>
<td>5.0±1.9</td>
<td>5.0±1.9</td>
<td>5.0±1.9</td>
<td>5.0±1.9</td>
<td>5.0±1.9</td>
</tr>
<tr>
<td>letter-br</td>
<td>10.9±2.6</td>
<td>10.9±2.6</td>
<td>10.9±2.6</td>
<td>10.9±2.6</td>
<td>10.9±2.6</td>
<td>10.9±2.6</td>
<td>10.9±2.6</td>
</tr>
<tr>
<td>letter-oq</td>
<td>12.0±3.7</td>
<td>13.6±5.8</td>
<td>13.9±6.0</td>
<td>12.3±4.9</td>
<td>23.0±15.6●</td>
<td>13.7±4.9</td>
<td>39.3±8.2●</td>
</tr>
<tr>
<td>optdigits</td>
<td>22.7±3.1</td>
<td>22.7±3.1</td>
<td>22.7±3.1</td>
<td>22.7±3.1</td>
<td>22.7±3.1</td>
<td>22.7±3.1</td>
<td>22.7±3.1</td>
</tr>
<tr>
<td>satimage-12v57</td>
<td>17.1±5.0</td>
<td>17.1±5.0</td>
<td>17.1±5.0</td>
<td>17.1±5.0</td>
<td>17.1±5.0</td>
<td>17.1±5.0</td>
<td>17.1±5.0</td>
</tr>
<tr>
<td>satimage-2v5</td>
<td>5.7±1.7</td>
<td>5.7±1.7</td>
<td>5.7±1.7</td>
<td>5.7±1.7</td>
<td>5.7±1.7</td>
<td>5.7±1.7</td>
<td>5.7±1.7</td>
</tr>
<tr>
<td>sick</td>
<td>6.9±2.8</td>
<td>6.9±2.8</td>
<td>6.9±2.8</td>
<td>6.9±2.8</td>
<td>6.9±2.8</td>
<td>6.9±2.8</td>
<td>6.9±2.8</td>
</tr>
<tr>
<td>sonar</td>
<td>11.4±4.2</td>
<td>11.4±4.2</td>
<td>11.4±4.2</td>
<td>11.4±4.2</td>
<td>11.4±4.2</td>
<td>11.4±4.2</td>
<td>11.4±4.2</td>
</tr>
<tr>
<td>spambase</td>
<td>17.5±4.5</td>
<td>18.5±5.0</td>
<td>20.0±8.1</td>
<td>19.0±9.9</td>
<td>28.8±17.0●</td>
<td>16.7±4.6</td>
<td>39.7±6.4●</td>
</tr>
<tr>
<td>tic-tac-toe</td>
<td>14.5±3.8</td>
<td>16.1±5.4</td>
<td>17.4±6.5</td>
<td>15.4±6.3</td>
<td>28.0±22.6●</td>
<td>13.6±3.4</td>
<td>39.8±8.2●</td>
</tr>
<tr>
<td>vehicle-bo-vs</td>
<td>16.5±4.5</td>
<td>15.7±5.7</td>
<td>16.5±8.2</td>
<td>11.2±5.7●</td>
<td>21.6±20.4</td>
<td>3.2±5.0●</td>
<td>41.9±5.6●</td>
</tr>
<tr>
<td>vehicle-b-v</td>
<td>2.8±1.1</td>
<td>3.4±2.1</td>
<td>4.5±1.6●</td>
<td>5.3±7.4</td>
<td>2.8±3.8</td>
<td>4.0±3.9</td>
<td>48.0±5.6●</td>
</tr>
<tr>
<td>vote</td>
<td>2.7±1.1</td>
<td>3.2±2.7</td>
<td>5.1±2.6●</td>
<td>5.4±5.2●</td>
<td>6.0±9.8</td>
<td>3.9±2.5●</td>
<td>47.8±6.1●</td>
</tr>
<tr>
<td>count of the best</td>
<td>12</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>POSE count of direct win</td>
<td>17</td>
<td>19.5</td>
<td>18</td>
<td>17.5</td>
<td>16</td>
<td>16</td>
<td>20</td>
</tr>
</tbody>
</table>

Comparison on ensemble size

POSE achieves the smallest size on 60% (12/20) of the data sets, while other methods are no more than 15% (3/20)

POSE is better than any other method on at least 80% (16/20) data sets

previous EA without theoretical guarantee

- ● and ○ denote that POSE is significantly better and worse, respectively, by the t-test with confidence level 0.05

[Qian, Yu and Zhou, AAAI 2015]
There are many other applications of selecting a good subset from a ground set.
Subset Selection

There are many other applications of selecting a good subset from a ground set

Influence maximization

Influential users

- Joe
- Kate
- Peter
- Tim

http://www.lamda.nju.edu.cn/qianc/
Subset Selection

There are many other applications of selecting a good subset from a ground set:
- Sparse regression
- Influence maximization
- Document summarization
- Sensor placement

Subset Selection: Given all items $V = \{v_1, \ldots, v_n\}$, an objective function $f: 2^V \rightarrow \mathbb{R}$ and a budget $b$, to select a subset $S \subseteq V$ such that

$$\max_{S \subseteq V} f(S) \quad \text{s.t.} \quad |S| \leq b$$

NP-hard
Pareto Optimization for Subset Selection

Introduce the Pareto optimization algorithm for subset selection (POSS)

\[
\begin{align*}
\text{Constrained} & \quad \text{Transformation} & \quad \text{Bi-objective} \\
\max_{S \subseteq V} f(S) & \quad \text{s.t.} & |S| \leq b & \quad \min_{S \subseteq V} (-f(S), |S|) \\
\end{align*}
\]

**Algorithm 14.2 POSS Algorithm**

**Input:** \( V = \{v_1, v_2, \ldots, v_n\} \); objective function \( f : \{0,1\}^n \rightarrow \mathbb{R} \); budget \( b \in [n] \)

**Parameter:** number \( T \) of iterations; isolation function \( I : \{0,1\}^n \rightarrow \mathbb{R} \)

**Output:** solution \( s \in \{0,1\}^n \) with \(|s|_1 \leq b\)

**Process:**

1: let \( s = 0^n \) and \( P = \{s\} \);
2: let \( t = 0 \);
3: while \( t < T \) do
4: select a solution \( s \) from \( P \) uniformly at random;
5: apply bit-wise mutation on \( s \) to generate \( s' \);
6: if \( \exists z \in P \) such that \( I(z) = I(s') \) and \( z > s' \) then
7: \( Q = \{z \in P | I(z) = I(s') \wedge s' \geq z\} \);
8: \( P = (P \setminus Q) \cup \{s'\} \);
9: end if
10: \( t = t + 1 \)
11: end while
12: return \( \arg \max_{s \in |P|_1 \leq b} f_1(s) \)

**Initialization:** put the special solution \( 0^n \) into the population \( P \)

**Reproduction:** pick a solution randomly from \( P \), and mutate it to generate a new one

**Evaluation & selection:** if the new solution is not dominated, put it into \( P \) and delete bad solutions

**Output:** select the best feasible solution

---

[Qian, Yu and Zhou, NIPS 2015]
Theoretical Results

POSS can achieve the optimal polynomial-time approximation guarantee

**Theorem 14.1.** For subset selection with **monotone** objective functions, POSS with $\mathbb{E}[T] \leq 2e b^2 n$ and $I(\cdot) = 0$, i.e., a constant function, can find a solution $s$ with $|s|_1 \leq b$ and $f(s) \geq (1 - e^{-\gamma_{\min}}) \cdot \text{OPT}$, where $\gamma_{\min} = \min_{s: |s|_1 = b-1} \gamma_s b$.

$$\forall S \subseteq T \subseteq V: f(S) \leq f(T)$$

Proved to be the optimal polynomial-time approximation [Harshaw et al., ICML’19]

**Remark:** Approximation guarantee implies worst-case performance

In practice, POSS can do better than the greedy algorithm by escaping from local optima

**Theorem 14.2.** For the Exponential Decay subclass of sparse regression, POSS using $\mathbb{E}[T] = O(b^2 (n - b) n \log n)$ and $I(s \in \{0,1\}^n) = \min\{i \mid s_i = 1\}$ can find an optimal solution, while the greedy algorithm cannot.

[Qian, Yu and Zhou, NIPS 2015]

[Previously obtained by the greedy algorithm]

[Das and Kempe, ICML’11]

[Das and Kempe, ICML’11]

[Prof., USC General Chair of STOC’18]

[D. Kempe]
Empirical Results

Comparison on sparse regression

<table>
<thead>
<tr>
<th>Data set</th>
<th>OPT</th>
<th>POSS</th>
<th>FR</th>
<th>FoBa</th>
<th>OMP</th>
<th>RPE</th>
<th>MCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>housing</td>
<td>.7437±.0297</td>
<td>.7437±.0297</td>
<td>.7429±.0300</td>
<td>.7423±.0301</td>
<td>.7415±.0300</td>
<td>.7388±.0304</td>
<td>.7354±.0297</td>
</tr>
<tr>
<td>eunite2001</td>
<td>.8484±.0132</td>
<td>.8482±.0132</td>
<td>.8348±.0143</td>
<td>.8442±.0144</td>
<td>.8349±.0150</td>
<td>.8424±.0153</td>
<td>.8320±.0150</td>
</tr>
<tr>
<td>symguide3</td>
<td>.2705±.0255</td>
<td>.2701±.0257</td>
<td>.2615±.0260</td>
<td>.2601±.0279</td>
<td>.2557±.0270</td>
<td>.2136±.0325</td>
<td>.2397±.0237</td>
</tr>
<tr>
<td>ionosphere</td>
<td>.5995±.0326</td>
<td>.5990±.0329</td>
<td>.5920±.0352</td>
<td>.5929±.0346</td>
<td>.5921±.0353</td>
<td>.5832±.0415</td>
<td>.5740±.0348</td>
</tr>
<tr>
<td>sonar</td>
<td>–</td>
<td>.5365±.0410</td>
<td>.5171±.0440</td>
<td>.5138±.0432</td>
<td>.5112±.0425</td>
<td>.4321±.0636</td>
<td>.4496±.0482</td>
</tr>
<tr>
<td>triazines</td>
<td>–</td>
<td>.4301±.0603</td>
<td>.4150±.0592</td>
<td>.4107±.0600</td>
<td>.4073±.0591</td>
<td>.3615±.0712</td>
<td>.3793±.0584</td>
</tr>
<tr>
<td>coil2000</td>
<td>–</td>
<td>.0627±.0076</td>
<td>.0624±.0076</td>
<td>.0619±.0075</td>
<td>.0619±.0075</td>
<td>.0363±.0141</td>
<td>.0570±.0075</td>
</tr>
<tr>
<td>mushrooms</td>
<td>–</td>
<td>.9912±.0020</td>
<td>.9909±.0021</td>
<td>.9909±.0022</td>
<td>.9909±.0022</td>
<td>.6813±.1294</td>
<td>.8652±.0474</td>
</tr>
<tr>
<td>clean1</td>
<td>–</td>
<td>.4368±.0300</td>
<td>.4169±.0299</td>
<td>.4145±.0309</td>
<td>.4123±.0315</td>
<td>.3196±.0562</td>
<td>.3563±.0364</td>
</tr>
<tr>
<td>wsu</td>
<td>–</td>
<td>.3376±.0267</td>
<td>.3319±.0247</td>
<td>.3341±.0258</td>
<td>.3313±.0246</td>
<td>.3342±.0276</td>
<td>.2694±.0385</td>
</tr>
<tr>
<td>gisette</td>
<td>–</td>
<td>.7265±.0098</td>
<td>.7001±.0116</td>
<td>.6747±.0145</td>
<td>.6731±.0134</td>
<td>.5360±.0318</td>
<td>.5709±.0123</td>
</tr>
<tr>
<td>farm-ads</td>
<td>–</td>
<td>.4217±.0100</td>
<td>.4196±.0101</td>
<td>.4170±.0113</td>
<td>.4170±.0113</td>
<td>–</td>
<td>.3771±.0110</td>
</tr>
</tbody>
</table>

POSS is always significantly better

max_{S \subseteq V} R_{z,S}^2 = \frac{\text{Var}(z) - \text{MSE}_{z,x}}{\text{Var}(z)} \text{ s.t. } |S| \leq b

points denoted that POSS is significantly better by the t-test with confidence level 0.05

[Qian, Yu and Zhou, NIPS 2015]

http://www.lamda.nju.edu.cn/qianc/
Noisy Subset Selection

Previous analyses assume that the objective function can be evaluated exactly. However, only a noisy value can be obtained in many applications of subset selection.

Consider a general noise model: \( (1 - \epsilon) \cdot f(S) \leq f^n(S) \leq (1 + \epsilon) \cdot f(S) \)

- Computing the \( R^2 \) objective is very expensive.
- Estimation by using a set of limited data brings noise.
- Estimation by simulating random diffusion brings noise.

Influential Users

Influence maximization
Pareto Optimization for Noisy Subset Selection

Inspired by the robustness of threshold selection against noise, accepts an offspring solution only if its fitness becomes better by at least $\tau$

$$f^n(S) \geq f^n(S') \quad \rightarrow \quad f^n(S) \geq f^n(S') + \tau$$

reduce the risk of deleting a good solution

Introduce the Pareto optimization algorithm for noisy subset selection (PONSS) modifies the domination-based comparison of POSS

POSS

$$S \succeq S' \iff \begin{cases} f^n(S) \geq f^n(S') \\ |S| \leq |S'| \end{cases}$$

PONSS

$$\theta \in [0,1]$$

$$S \succeq S' \iff \begin{cases} f^n(S) \geq \frac{1 + \theta}{1 - \theta} f^n(S') \\ |S| \leq |S'| \end{cases}$$

[Qian, Shi, Yu, Tang and Zhou, NIPS 2017]
Theoretical Results

Approximation ratio under noise

**Theorem 16.1.** For subset selection under multiplicative noise with the assumption Eq. (17.29), with probability at least \((1/2)(1 - (12nb^2 \log 2b) / l^2\delta)\), PONSS with \(\theta \geq \epsilon\) and \(T = 2\epsilon l n b^2 \log 2b\) finds a solution \(s\) with \(|s|_1 \leq b\) and \(f(s) \geq \frac{1-\epsilon}{1+\epsilon} (1 - e^{-\gamma}) \cdot \text{OPT}\).

**PONSS**

\[
\frac{f(S)}{\text{OPT}} \geq \frac{1 - \epsilon}{1 + \epsilon} (1 - e^{-\gamma})
\]

**Greedy** [Horel and Singer, NIPS’16]

\[
\frac{f(S)}{\text{OPT}} \geq \frac{1}{1 + \frac{2\epsilon b}{(1 - \epsilon)^\gamma}} \left(1 - \left(\frac{1 - \epsilon}{1 + \epsilon}\right)^b e^{-\gamma}\right)
\]

EAs achieve better approximation guarantees than conventional algorithms

[Qian, Shi, Yu, Tang and Zhou, NIPS 2017]
Large-scale Subset Selection

The applications of subset selection are often **large-scale**

- **Millions of variables**
- **Millions of social network users**

The diagram illustrates the process of subset selection using PONSS (Parallel Online Non-dominated Sorting Selection) in two stages:

**Stage 1:**
- Single machine
- Data: solution \((0)^n\)
- Population: pick a solution
  - a new solution

**Stage 2:**
- Single machine
- Data: solution \((0)^n\)
- Population: pick a solution
  - a new solution

Output the subset satisfying
\[
\arg \max_{S_i} \{ f(S_i) \mid 1 \leq i \leq m + 1 \}
\]

Empirical Results:
- Approximation ratio:
- Very close to the centralized algorithm
- \#machines

References:
[Qian, IEEE Trans. Evolutionary Computation 2020]

[http://www.lamda.nju.edu.cn/qianc/](http://www.lamda.nju.edu.cn/qianc/)
Dynamic Subset Selection

How about the performance of POSS under dynamic environments?

Ground set
\[ V = \{v_1, ..., v_n\} \]

Subset \( S \subseteq V \)
\[ |S| \leq b \]

Result Diversification
\[ \arg \max_{X \subseteq V} \left( f(S) + \lambda \cdot \text{div}(S) \right) \quad \text{s.t.} \quad |S| \leq b \]

Open problem: When the objective changes dynamically, is it possible to maintain the \((1/2)\)-approximation ratio in polynomial running time? [Borodin et al., PODS’12]

A. Borodin
Prof., Univ. of Toronto
Member of the Royal Society of Canada

[Qian, Liu and Zhou, Artificial Intelligence 2022]
Outline

- Introduction
- Theoretical analysis tools for MOEAs
- Theoretical perspectives of MOEAs
  - Recombination operator, constrained optimization, noisy optimization
- Multi-objective evolutionary learning algorithms
  - Selective ensemble, subset selection
- Conclusion
Conclusion

Can we build theoretical foundation of multi-objective evolutionary learning?

Part I: Analysis Methodology
Propose a general theoretical analysis tool, i.e., switch analysis, for MOEAs

Part II: Theoretical Perspectives
Derive theoretical results of MOEAs about operators, constraints and noise

Part III: Learning Algorithms
Develop theoretically grounded multi-objective evolutionary learning algorithms

Yes
Evolutionary Learning: Advances in Theories and Algorithms

- Presents theoretical results for evolutionary learning
- Provides general theoretical tools for analysing evolutionary algorithms
- Proposes evolutionary learning algorithms with provable theoretical guarantees

Thanks