Towards a Principled Learning Rate Adaptation for Natural Evolution Strategies

Masahiro Nomura, Isao Ono
(Tokyo Institute of Technology)

EvoApplications 2022

Outline

- Introduction
- xNES
- Learning Rate Adaptation
- Experiments
- Conclusion

Introduction

- Natural Evolution Strategies (NES)
  - Promising framework for black-box continuous optimization problems
  - NES optimizes the parameter of a probability distribution
  - This update is performed based on the estimated natural gradient
- Learning Rate in NES
  - One of the critical parameters in NES is a learning rate
  - If the learning rate is too high, the parameter update will be unstable
  - If the learning rate is too low, the speed of approaching the optimal solution will be slow
- Proposal: A new learning rate adaptation mechanism in view of the natural gradient method
  - The learning rate is adapted based on estimation accuracy of the natural gradient

xNES [Glasmachers et al., 2011]

1. Create $\mathcal{N}(\boldsymbol{m}^{(0)}, \mathbf{C}^{(0)} = {\sigma^{(0)}}^2 \mathbf{B}^{(0)} {\mathbf{B}^{(0)}}^{\mathrm{T}})$ and set $g = 0$

   $\sigma^{(0)}$: step size, $\mathbf{B}^{(0)}$: normalization transformation matrix

2. Generate λ solutions following $\mathcal{N}(\boldsymbol{m}^{(g)}, \mathbf{C}^{(g)})$

   $\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \boldsymbol{z}_i \,, \ \boldsymbol{z}_i \sim \mathcal{N}(0, \mathbf{I})$

3. Evaluate the solutions and give weights to them

4. Update the parameters $\boldsymbol{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{B}^{(g)}$

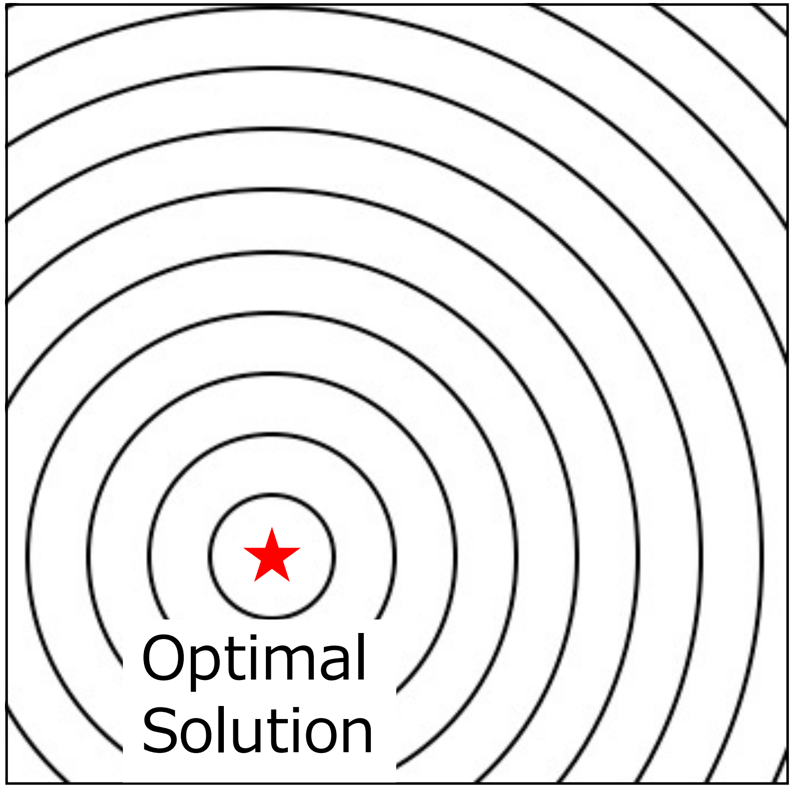
   $\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{G}_\delta \qquad \mathbf{G}_\delta = \sum_{i=1}^{\lambda} w_i \boldsymbol{z}_i$

   $\sigma^{(g+1)} = \sigma^{(g)} \exp(\eta_\sigma G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \sum_{i=1}^{\lambda} w_i (\boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}} - \mathbf{I})$

   $\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\eta_\mathrm{B} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = \mathrm{tr}(\mathbf{G}_\mathrm{M})/d \,, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



xNES [Glasmachers et al., 2011]

1. **Create $\mathcal{N}(\boldsymbol{m}^{(0)}, \mathbf{C}^{(0)} = \boldsymbol{\sigma}^{(0)^2}\mathbf{B}^{(0)}\mathbf{B}^{(0)^{\mathrm{T}}})$ and set $g = 0$**

   $\boldsymbol{\sigma}^{(0)}$: step size, $\mathbf{B}^{(0)}$: normalization transformation matrix

2. Generate λ solutions following $\mathcal{N}(\boldsymbol{m}^{(g)}, \mathbf{C}^{(g)})$

   $$\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \sigma^{(g)}\mathbf{B}^{(g)}\boldsymbol{z}_i\,, \quad \boldsymbol{z}_i \sim \mathcal{N}(0, \mathbf{I})$$

3. Evaluate the solutions and give weights to them

4. Update the parameters $\boldsymbol{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{B}^{(g)}$

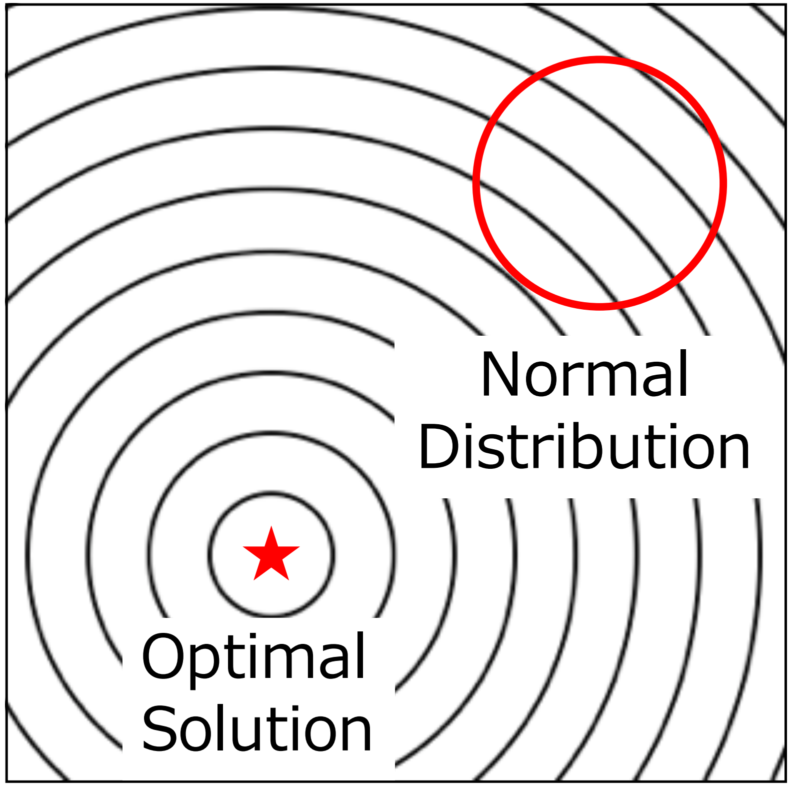
   $$\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \sigma^{(g)}\mathbf{B}^{(g)}\mathbf{G}_\delta \qquad \mathbf{G}_\delta = \textstyle\sum_{i=1}^{\lambda} w_i \boldsymbol{z}_i$$

   $$\sigma^{(g+1)} = \sigma^{(g)} \exp(\eta_\sigma G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \textstyle\sum_{i=1}^{\lambda} w_i (\boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}} - \mathbf{I})$$

   $$\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\eta_\mathrm{B} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = \mathrm{tr}(\mathbf{G}_\mathrm{M})/d\,, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



xNES [Glasmachers et al., 2011]

1. Create $\mathcal{N}(m^{(0)}, \boldsymbol{C}^{(0)} = \sigma^{(0)^2} \boldsymbol{B}^{(0)} \boldsymbol{B}^{(0)^T})$ and set $g = 0$

   $\sigma^{(0)}$: step size, $\boldsymbol{B}^{(0)}$: normalization transformation matrix

2. **Generate λ solutions following $\mathcal{N}(m^{(g)}, \mathbf{C}^{(g)})$**

   $$\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \boldsymbol{\sigma}^{(g)} \mathbf{B}^{(g)} \mathbf{z}_i\,,\ \ \mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

3. Evaluate the solutions and give weights to them

4. Update the parameters $\boldsymbol{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{B}^{(g)}$

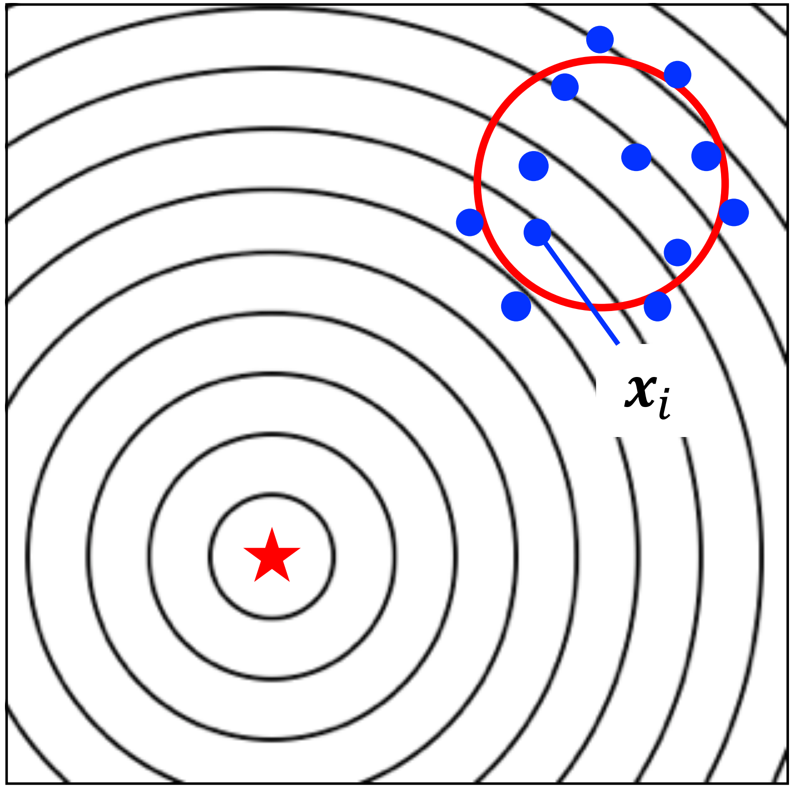
   $$\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{G}_\delta \qquad \mathbf{G}_\delta = \sum_{i=1}^{\lambda} w_i \mathbf{z}_i$$

   $$\sigma^{(g+1)} = \sigma^{(g)} \exp(\eta_\sigma G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \sum_{i=1}^{\lambda} w_i (\mathbf{z}_i \mathbf{z}_i^\mathrm{T} - \mathbf{I})$$

   $$\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\eta_\mathrm{B} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = \mathrm{tr}(\mathbf{G}_\mathrm{M})/d\,, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



xNES [Glasmachers et al., 2011]

1. Create $\mathcal{N}(\boldsymbol{m}^{(0)}, \mathbf{C}^{(0)} = {\sigma^{(0)}}^2 \mathbf{B}^{(0)} {\mathbf{B}^{(0)}}^{\mathrm{T}})$ and set $g = 0$

   $\sigma^{(0)}$: step size, $\mathbf{B}^{(0)}$: normalization transformation matrix

2. Generate λ solutions following $\mathcal{N}(\boldsymbol{m}^{(g)}, \mathbf{C}^{(g)})$

   $$\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \boldsymbol{z}_i\,, \quad \boldsymbol{z}_i \sim \mathcal{N}(0, \mathbf{I})$$

3. **Evaluate the solutions and give weights to them**

4. Update the parameters $\boldsymbol{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{B}^{(g)}$

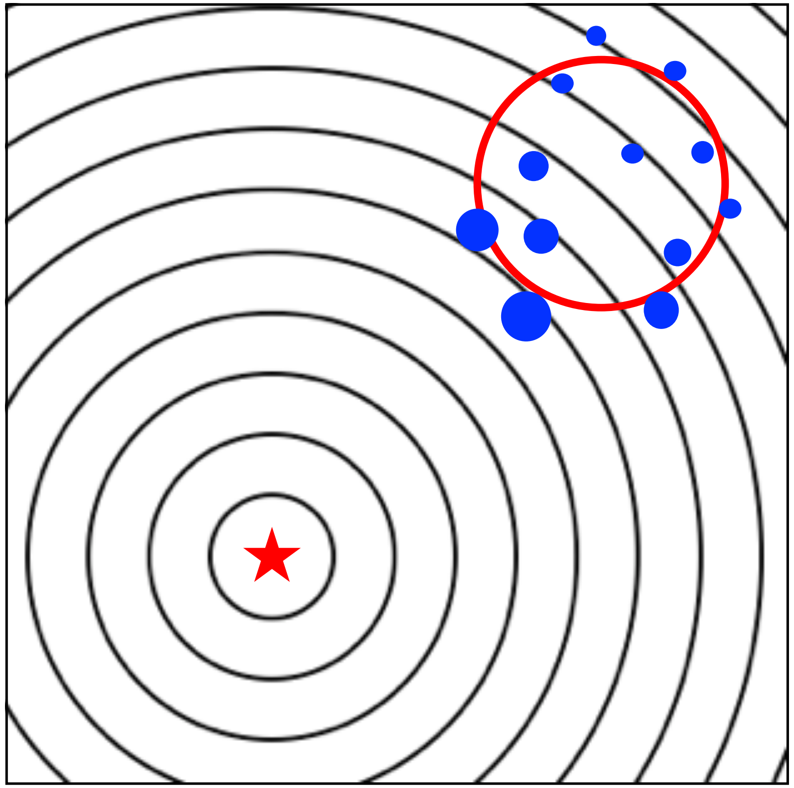
   $$\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{G}_\delta \qquad \mathbf{G}_\delta = \sum_{i=1}^{\lambda} w_i \boldsymbol{z}_i$$

   $$\sigma^{(g+1)} = \sigma^{(g)} \exp(\eta_\sigma G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \sum_{i=1}^{\lambda} w_i (\boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}} - \mathbf{I})$$

   $$\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\eta_\mathrm{B} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = \mathrm{tr}(\mathbf{G}_\mathrm{M}) / d\,, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



xNES [Glasmachers et al., 2011]

1. Create $\mathcal{N}(\boldsymbol{m}^{(0)}, \mathbf{C}^{(0)} = {\sigma^{(0)}}^2 \mathbf{B}^{(0)} {\mathbf{B}^{(0)}}^{\mathrm{T}})$ and set $g = 0$

   $\sigma^{(0)}$: step size, $\mathbf{B}^{(0)}$: normalization transformation matrix

2. Generate λ solutions following $\mathcal{N}(\boldsymbol{m}^{(g)}, \mathbf{C}^{(g)})$

   $$\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \boldsymbol{z}_i\,, \quad \boldsymbol{z}_i \sim \mathcal{N}(0, \mathbf{I})$$

3. Evaluate the solutions and give weights to them

4. **Update the parameters $\boldsymbol{m}^{(g)}$, $\boldsymbol{\sigma}^{(g)}$, $\mathbf{B}^{(g)}$**

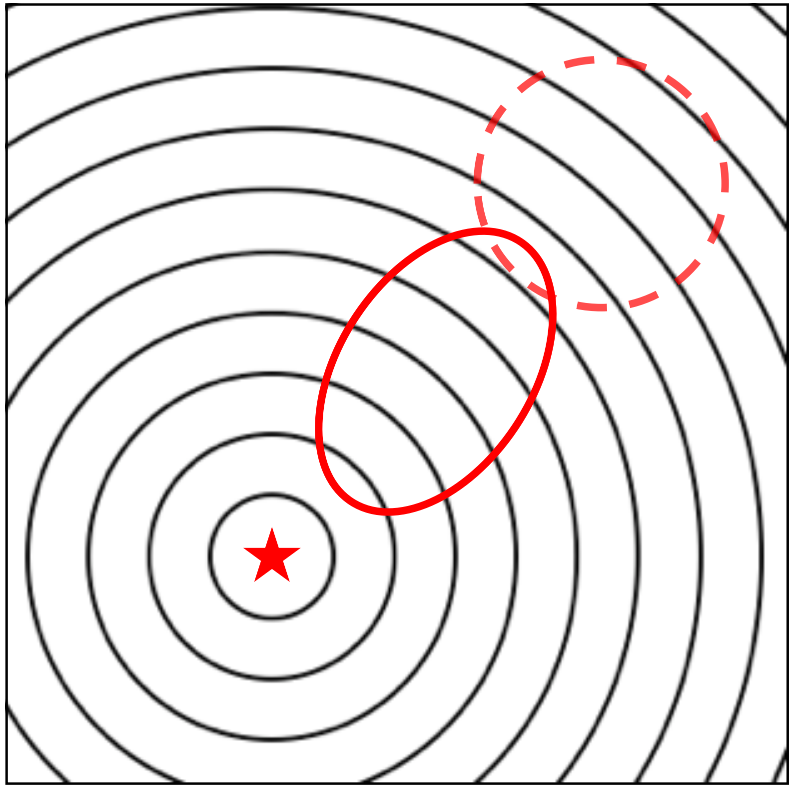
   $$\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \boldsymbol{\sigma}^{(g)} \mathbf{B}^{(g)} \mathbf{G}_\delta \qquad \mathbf{G}_\delta = \sum_{i=1}^{\lambda} w_i \mathbf{z}_i$$

   $$\boldsymbol{\sigma}^{(g+1)} = \boldsymbol{\sigma}^{(g)} \exp(\eta_\sigma G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \sum_{i=1}^{\lambda} w_i (\mathbf{z}_i \mathbf{z}_i^{\mathrm{T}} - \mathbf{I})$$

   $$\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\eta_\mathrm{B} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = tr(\mathbf{G}_\mathrm{M})/d\,, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



xNES [Glasmachers et al., 2011]

1. Create $\mathcal{N}(\boldsymbol{m}^{(0)}, \mathbf{C}^{(0)} = {\sigma^{(0)}}^2 \mathbf{B}^{(0)} {\mathbf{B}^{(0)}}^{\mathrm{T}})$ and set $g = 0$

   $\sigma^{(0)}$: step size, $\mathbf{B}^{(0)}$: normalization transformation matrix

2. Generate λ soluti

   $\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \sigma$

Learning rates for the covariance matrix
- Fixed during optimization

3. Evaluate the solutions and give weights to them

4. Update the parameters $\boldsymbol{m}^{(g)}$, $\boldsymbol{\sigma}^{(g)}$, $\mathbf{B}^{(g)}$

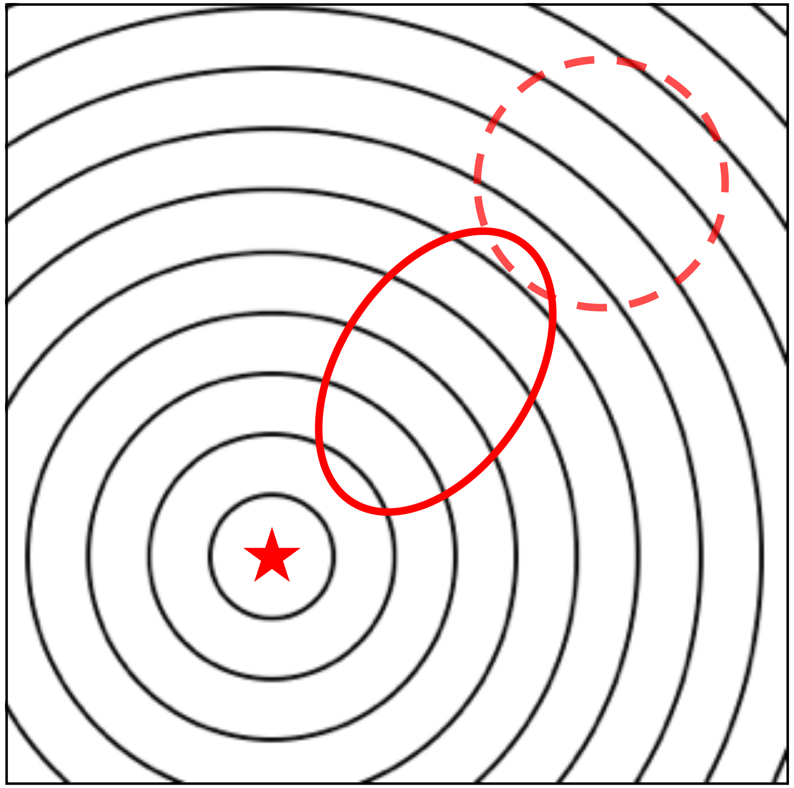
$$\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \boldsymbol{\sigma}^{(g)} \mathbf{B}^{(g)} \mathbf{G}_\delta \qquad \mathbf{G}_\delta = \sum_{i=1}^{\lambda} w_i \mathbf{z}_i$$

$$\boldsymbol{\sigma}^{(g+1)} = \boldsymbol{\sigma}^{(g)} \exp(\boxed{\eta_\sigma} G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \sum_{i=1}^{\lambda} w_i (\mathbf{z}_i \mathbf{z}_i^{\mathrm{T}} - \mathbf{I})$$

$$\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\boxed{\eta_\mathrm{B}} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = tr(\mathbf{G}_\mathrm{M})/d, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



xNES [Glasmachers et al., 2011]

1. Create $\mathcal{N}(\boldsymbol{m}^{(0)}, \mathbf{C}^{(0)} = {\sigma^{(0)}}^2 \mathbf{B}^{(0)} {\mathbf{B}^{(0)}}^{\mathrm{T}})$ and set $g = 0$

   $\sigma^{(0)}$: step size, $\mathbf{B}^{(0)}$: normalization transformation matrix

2. Generate λ solutions following $\mathcal{N}(\boldsymbol{m}^{(g)}, \mathbf{C}^{(g)})$

   $\boldsymbol{x}_i = \boldsymbol{m}^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{z}_i \,, \;\; \mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I})$

3. Evaluate the solutions and give weights to them

4. Update the parameters $\boldsymbol{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{B}^{(g)}$

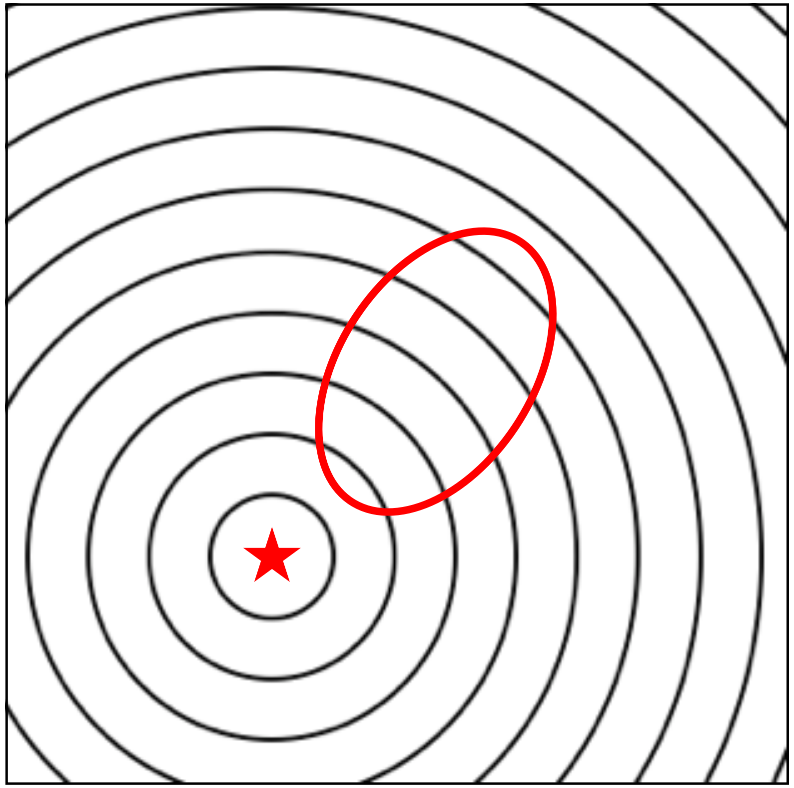
   $\boldsymbol{m}^{(g+1)} = \boldsymbol{m}^{(g)} + \eta_m \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{G}_\delta \qquad \mathbf{G}_\delta = \sum_{i=1}^{\lambda} w_i \mathbf{z}_i$

   $\sigma^{(g+1)} = \sigma^{(g)} \exp(\eta_\sigma G_\sigma / 2) \qquad \mathbf{G}_\mathrm{M} = \sum_{i=1}^{\lambda} w_i (\mathbf{z}_i \mathbf{z}_i^{\mathrm{T}} - \mathbf{I})$

   $\mathbf{B}^{(g+1)} = \mathbf{B}^{(g)} \exp(\eta_\mathrm{B} \mathbf{G}_\mathrm{B} / 2) \qquad G_\sigma = \mathrm{tr}(\mathbf{G}_\mathrm{M})/d \,, \mathbf{G}_\mathrm{B} = \mathbf{G}_\mathrm{M} - G_\sigma \mathbf{I}$

5. Terminate if the criterion is met,

   otherwise $g \leftarrow g + 1$ and go to Step 2.



Learning Rate Adaptation for NES

- The learning rate of the natural gradient method should depend on its estimation accuracy.
- To quantify the estimation accuracy, we use an evolution path in the parameter space.
  - The evolution path accumulates successive parameter movements.
  - This was first introduced in the population size adaptation of the CMA-ES [Nishida and Akimoto, 2016;2018]
- If the length of the evolution path is larger than its expectation under a random function,

  => the accuracy is high, as the tendency of parameter update can be captured.
- If the length of the evolution path is close to its expectation under a random function.

  => the estimation is dominated by noise, and the accuracy is low.
- In this study, we consider an evolution path for only the covariance matrix, not the mean vector.
  - We fix the learning rate for the mean vector.

Evolution Path for Covariance Matrix (1)

- To capture the movement from iteration t to $t+1$, we define the covariance movement matrix

$$\delta\Sigma^{(t+1)} = \left(\sigma^{(t+1)}\right)^2 B^{(t+1)}\left(B^{(t+1)}\right)^{\mathrm{T}} - \left(\sigma^{(t)}\right)^2 B^{(t)}\left(B^{(t)}\right)^{\mathrm{T}}$$

- We define the evolution path in the parameter space of the covariance matrix.

$$p_{\Sigma}^{(t+1)} = (1-\beta)p_{\Sigma}^{(t)} + \sqrt{\beta(2-\beta)}I_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\Sigma^{(t+1)}/\mathbb{E}\left[||I_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\Sigma^{(t+1)}||^2\right]^{\frac{1}{2}}$$

  - β : cumulation factor of the evolution path
  - $I_{\Sigma^{(t)}}$: Fisher information matrix of the covariance matrix

- The expectation $\mathbb{E}[\cdot]$ is taken under a random function.

- We use the approximation of $\mathbb{E}\left[||I_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\Sigma^{(t+1)}||^2\right]^{\frac{1}{2}}$, which will be introduced later.

Evolution Path for Covariance Matrix (2)

- Using the result from [Nishida and Akimoto, 2016], we define the length of the evolution path p_Σ.

$$l_\theta^{(t+1)} \coloneqq \frac{\mathrm{Tr}\left(\left(p_\Sigma^{(t+1)}\right)^2\right)}{2}.$$

- It represents the movement of the KL divergence in the parameter space.
- Under a random function, the length of the evolution path approaches 1 as the itr. t increases.

=> Comparing the length of the evolution path with the normalization factor $\gamma_\theta^{(t+1)}$ which is updated as

$$\gamma_\theta^{(t+1)} = (1-\beta)^2 \gamma_\theta^{(t)} + \beta(2-\beta),$$

we can obtain the estimation of the accuracy of the parameter update.

Updating Learning Rate

- When the accuracy is high (resp. low), the learning rate should be increased (resp. decreased).

$$\eta_\sigma^{(t+1)} = \eta_\sigma^{(t)} \exp\left(\beta_\sigma \left(\frac{l_\theta^{(t+1)}}{\alpha} - \gamma_\theta^{(t+1)}\right)\right),$$

$$\eta_B^{(t+1)} = \eta_B^{(t)} \exp\left(\beta_B \left(\frac{l_\theta^{(t+1)}}{\alpha} - \gamma_\theta^{(t+1)}\right)\right),$$

where α, β_σ, and β_B are pre-defined hyperparameters.

- We clip the learning rates as

$$\eta_\sigma^{(t+1)} \leftarrow \mathrm{clip}\left(\eta_\sigma^{(t+1)}, \eta_\sigma^{\min}, \eta_\sigma^{max}\right),$$

$$\eta_B^{(t+1)} \leftarrow \mathrm{clip}\left(\eta_B^{(t+1)}, \eta_B^{\min}, \eta_B^{max}\right).$$

- We set $\eta_\sigma^{max} = \eta_B^{max} = 1$ to prevent extrapolation in the update of the parameter.
- We set the default values in the original xNES paper for $\eta_\sigma^{\min}$ and $\eta_B^{\min}$.

  <= The setting of the learning rates in xNES is often too conservative [Fukushima et al., 2011].

Approximation of Expectation

- We approximate $\mathbb{E}\left[||I_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\Sigma^{(t+1)}||^2\right]^{\frac{1}{2}}$ used in the update of the evolution path p_Σ.

$$\mathbb{E}\left[||I_{\Sigma^{(t)}}^{\frac{1}{2}}\delta\Sigma^{(t+1)}|\big|^2\right]^{\frac{1}{2}} \approx \frac{1}{\mu_w}\left(\frac{\eta_B^2}{2}\left(1+\frac{4\eta_\sigma^2}{d\mu_w}\right)(d^2+d-2)+\eta_\sigma^2\right)$$

where $\mu_w = \sum_{i=1}^{\lambda} 1/w_i^2$, d is the number of dimension.

- (This is derived by using Slepian-Bangs formula and Taylor approximation.)

- We recalculate this approximation every iteration because it depends on η_σ and η_B.

Experiments

- Research Questions (RQs)
  - RQ1. When the learning rate is fixed, how does the evolution path behave?
  - RQ2. How is the learning rate adapted in xNES with the proposed adaptation mechanism?
  - RQ3. Does xNES with the proposed learning rate adaptation mechanism achieve better performance than xNES with fixed learning rate?

- The code is available at https://github.com/nomuramasahir0/xnes-adaptive-lr.

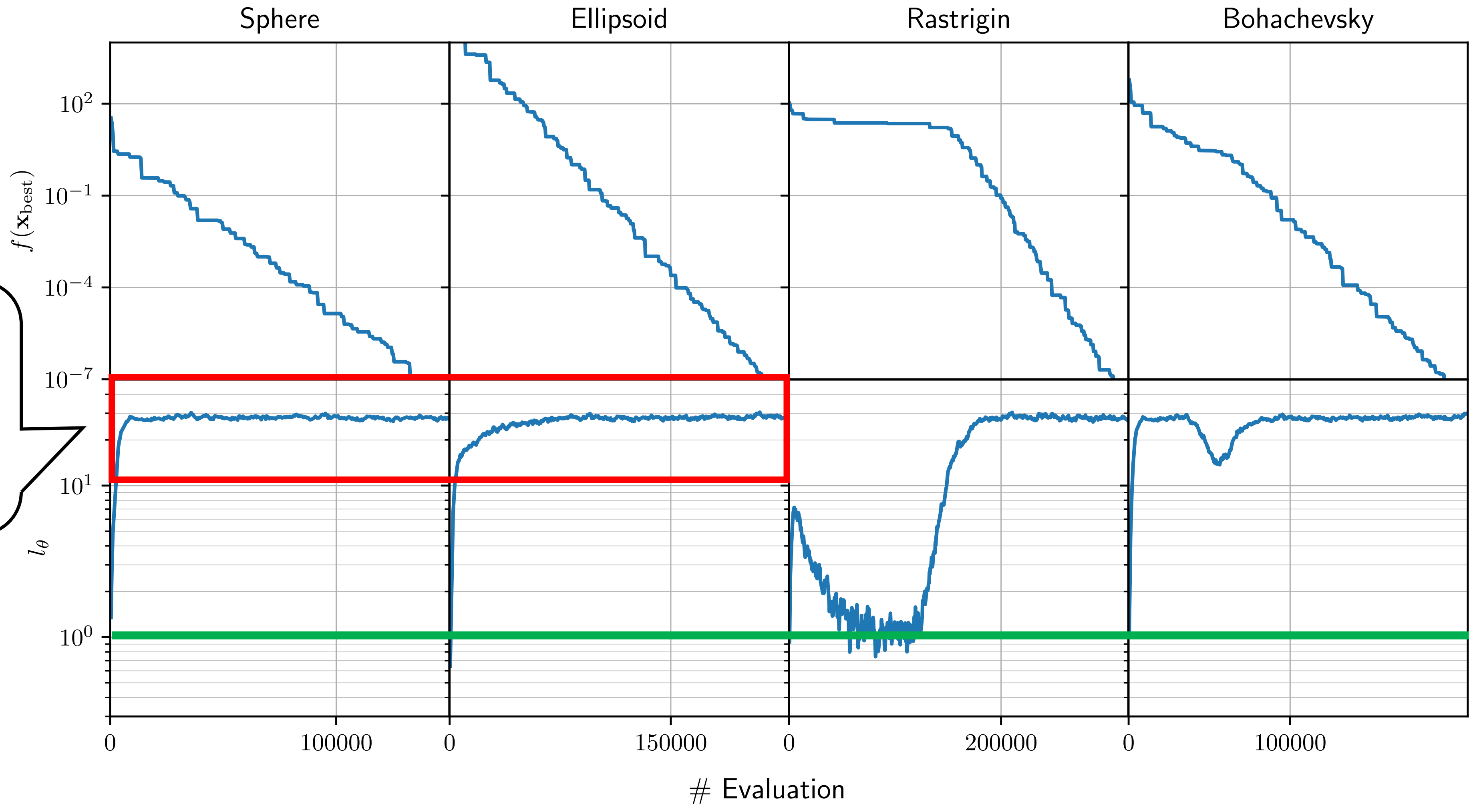
Experiments: Benchmark Problems

- We use the following four functions:
  - Two unimodal functions (Sphere and Ellipsoid)
  - Two multimodal functions (Rastrigin and Bohachevsky)
    - The Rastrigin function has strong multimodality.
    - The Bohachevsky function has relatively weak multimodality.

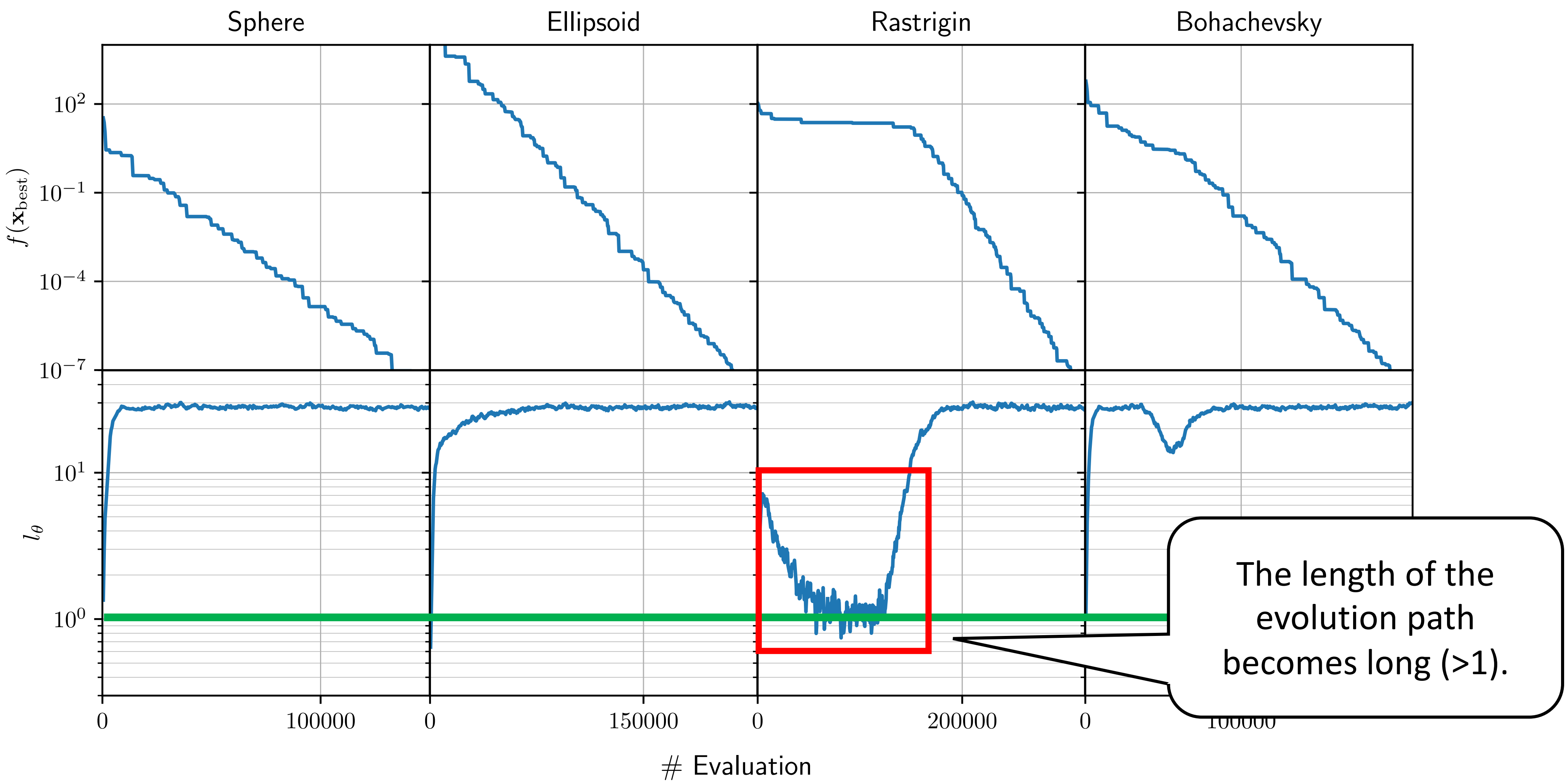
- The dimension is $d = 10$.
- The initial parameters are set to locate in the outside of the optimum.

RQ1: Evolution Path with Fixed Learning Rate

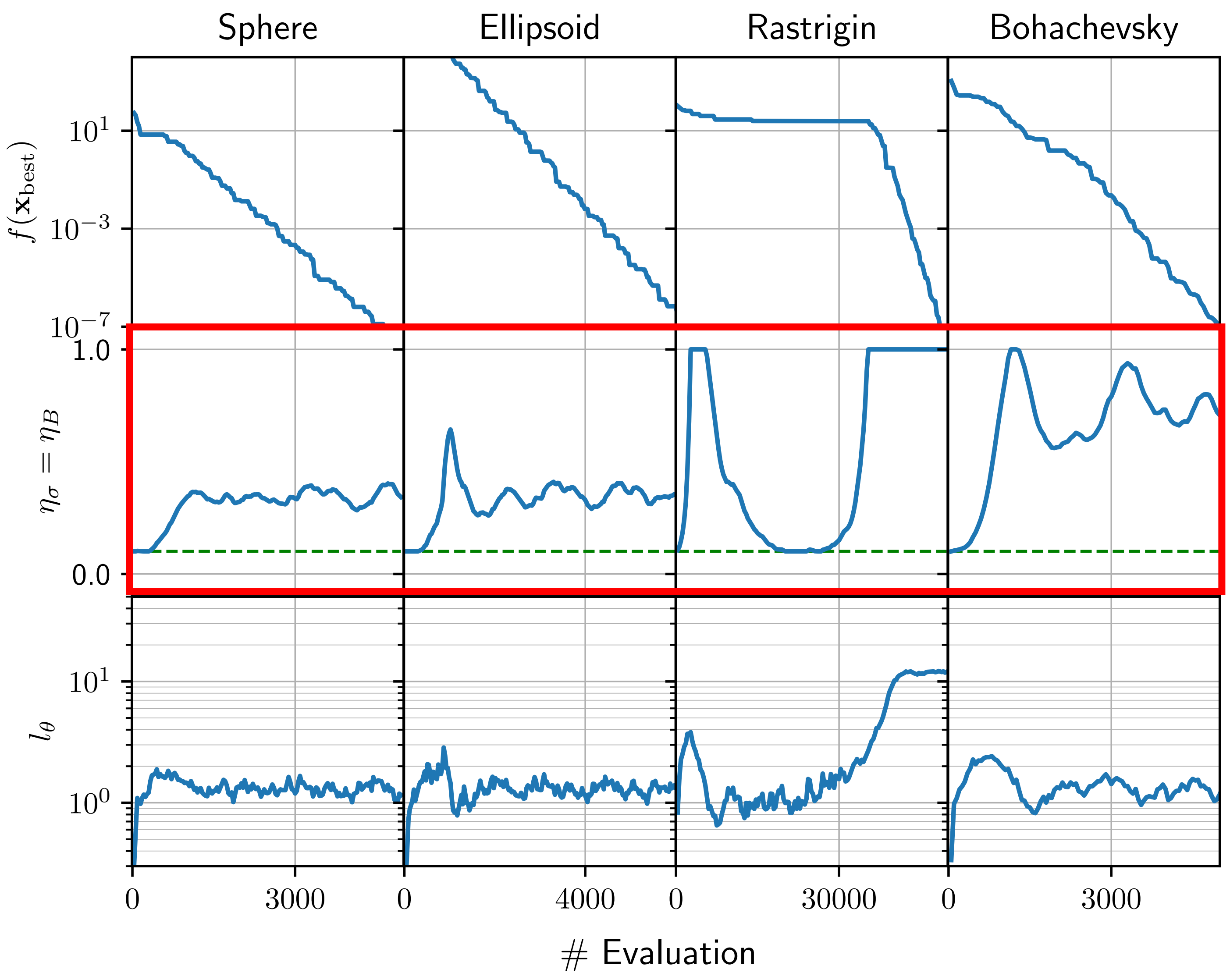
The length of the evolution path becomes long (>1).



RQ1: Evolution Path with Fixed Learning Rate



RQ2: Behavior of Learning Rate Adaptation

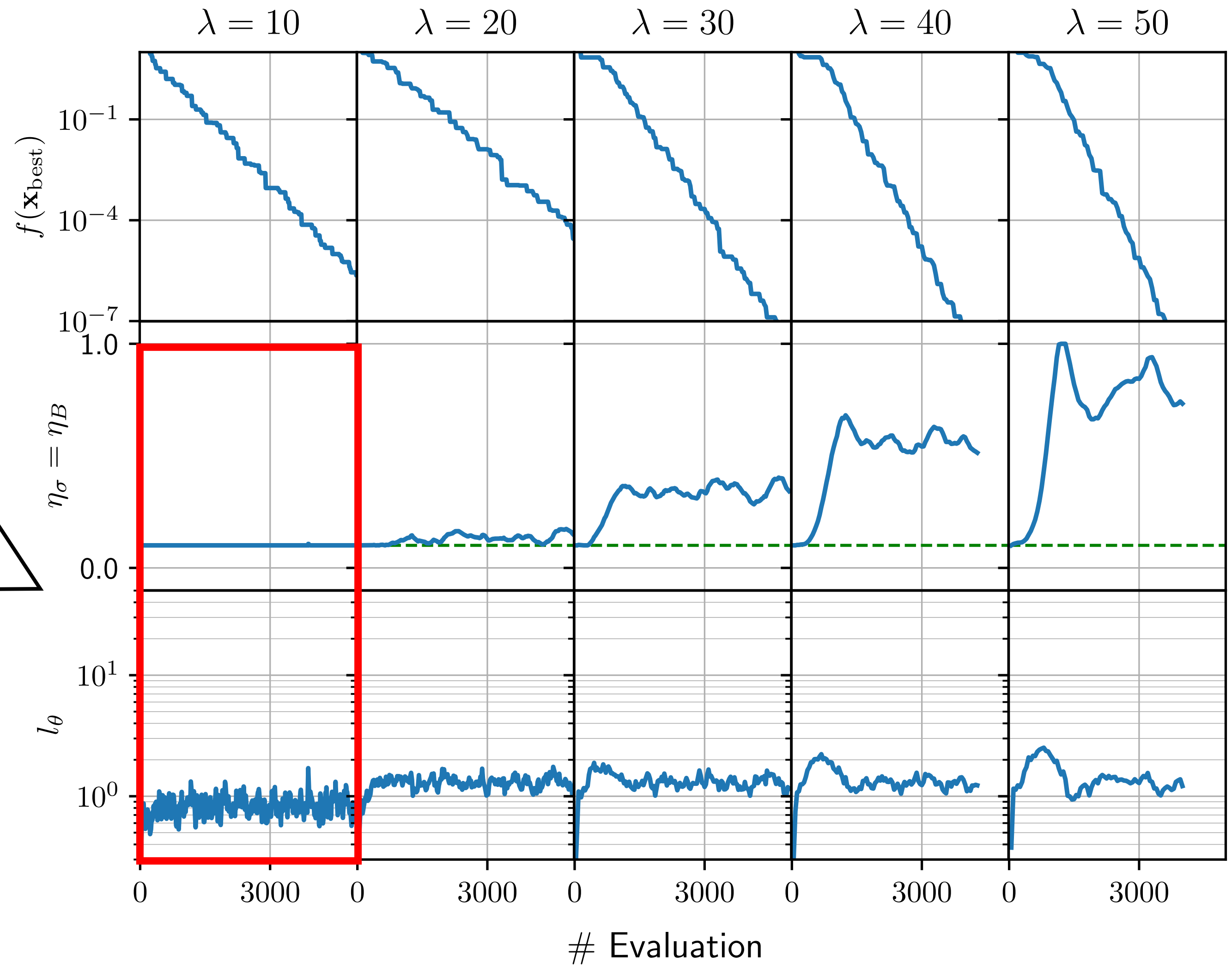


λ=30 for Sphere and Ellipsoid
λ=300 for Rastrigin
λ=50 for Bohachevsky

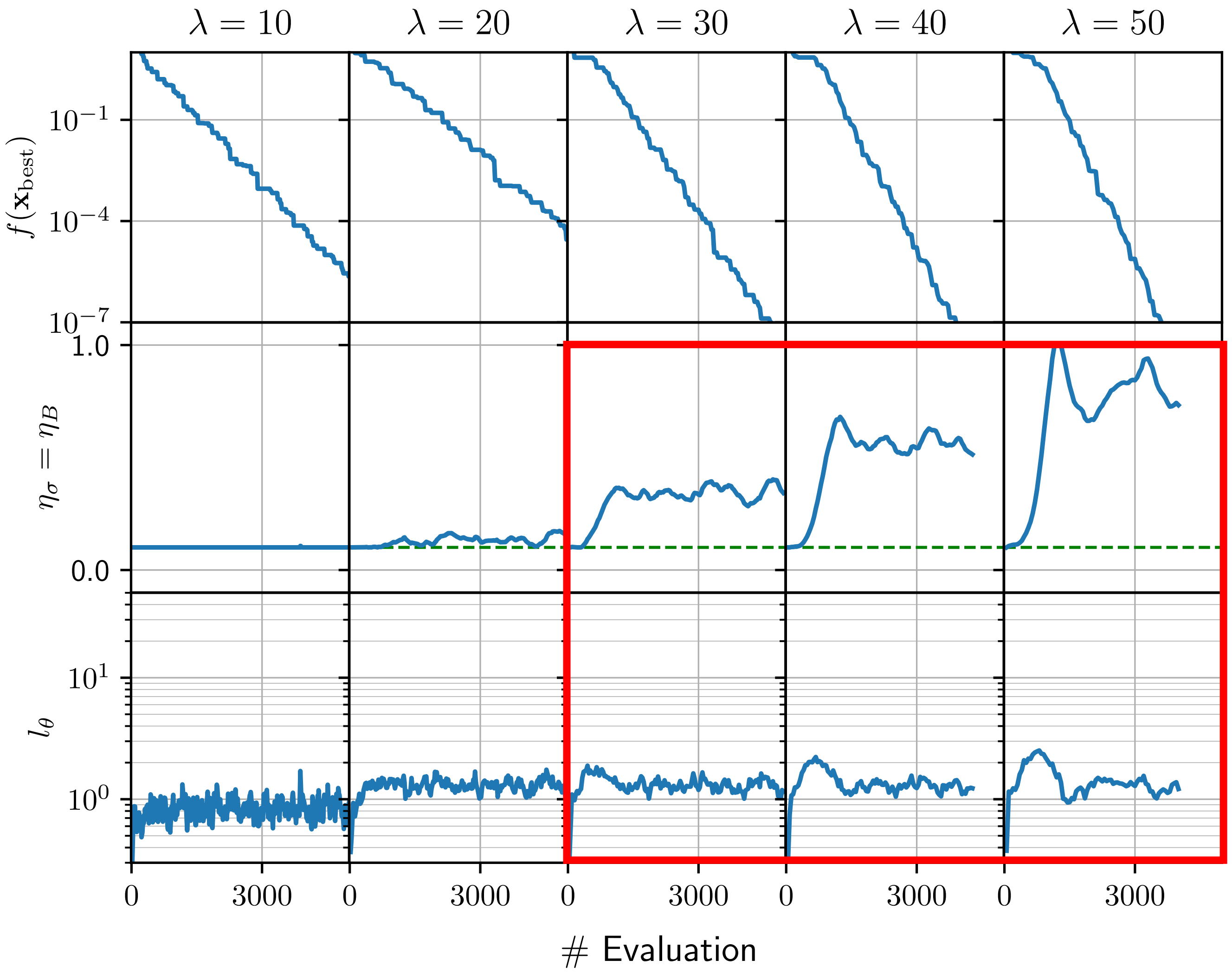
When the length of the evolution path increases, the learning rates also increase.

RQ2: Behavior of Learning Rate Adaptation

The length of the evolution path does not increase.
=> The learning rates are not changed at all.

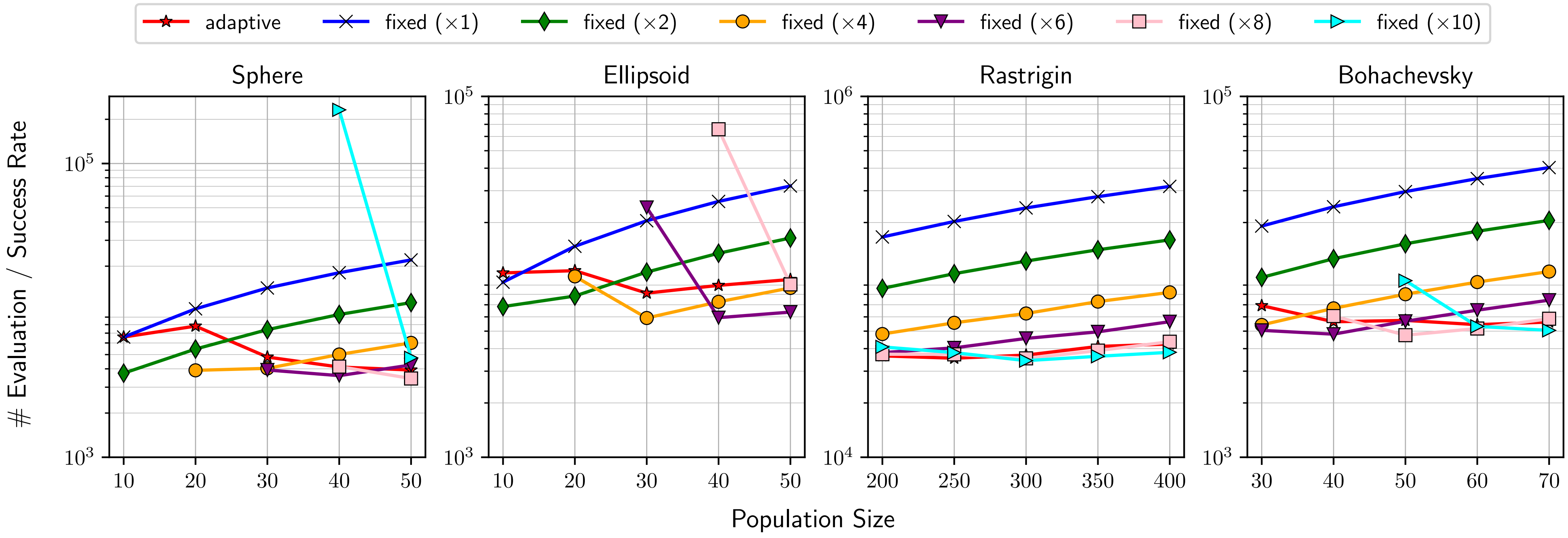


RQ2: Behavior of Learning Rate Adaptation

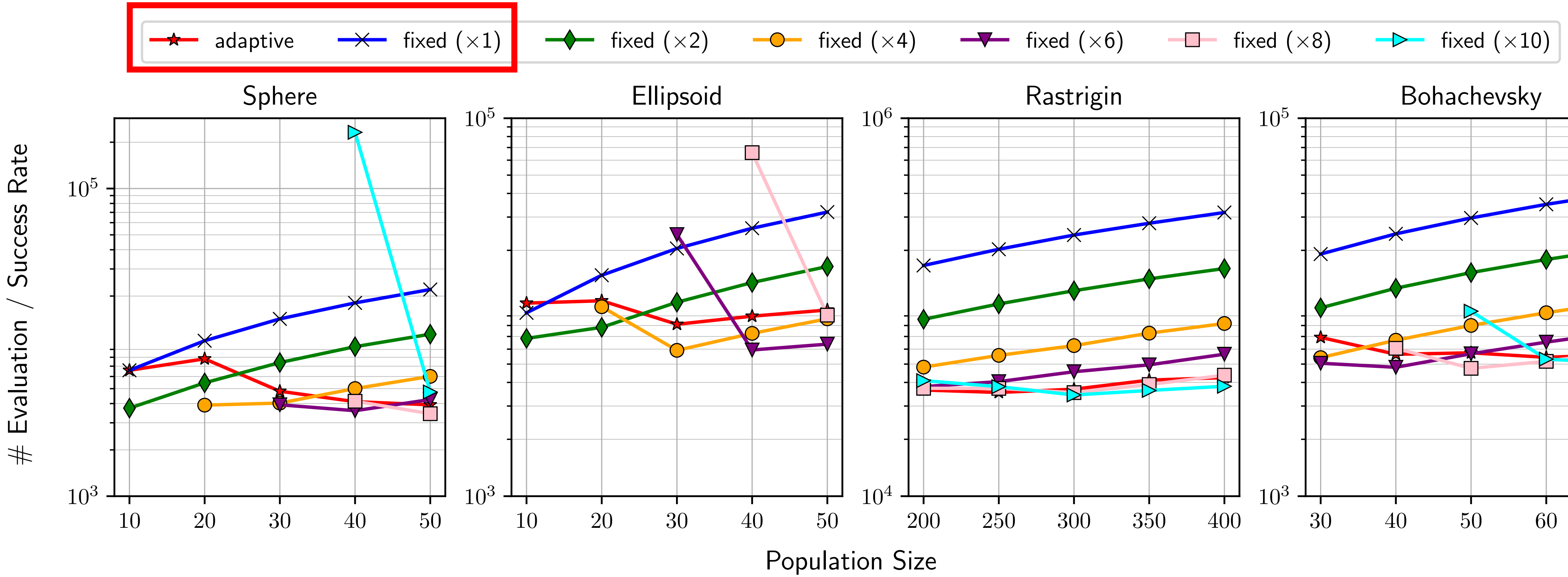


The length of the evolution path increases.
=> The learning rates also increases.

RQ3: Fixed LR vs. Adaptive LR



RQ3: Fixed LR vs. Adaptive LR

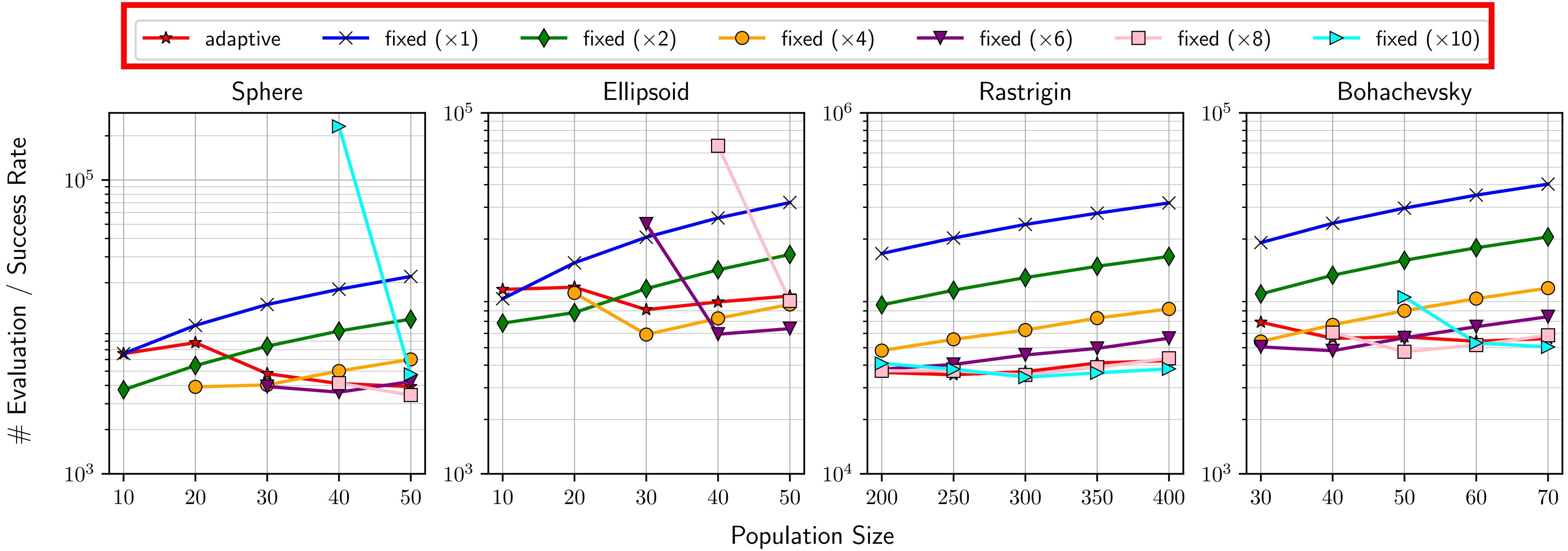


- In Sphere and Ellipsoid, when λ=10, the performance is almost the same.
  - As λ increases, the proposed mechanism shows better performance the xNES with the default learning rate.

<= The estimation accuracy should become high when λ is large.

- In Rastrigin and Bohachevsky, the proposed mechanism outperforms xNES with the default learning rate.

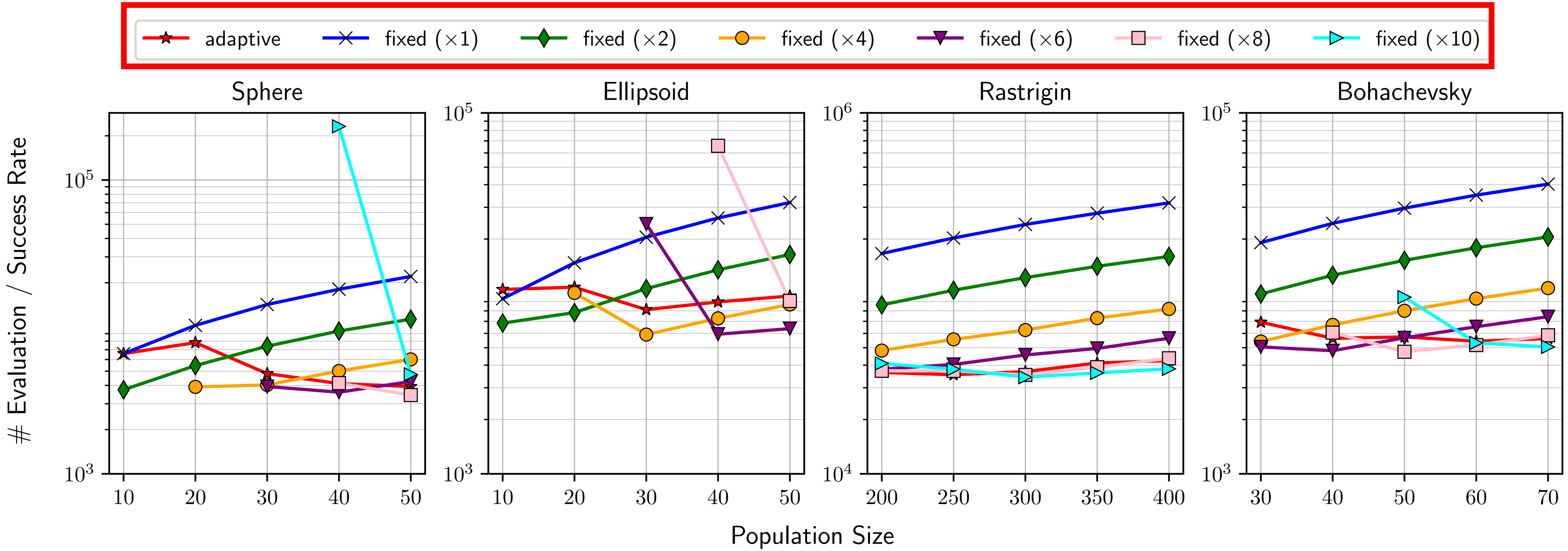
RQ3: Fixed LR vs. Adaptive LR



- When the population size is large, the performance of the proposed mechanism is close to pink, i.e., xNES (x8).
- However, pink fails to find the optimum in Sphere and Ellipsoid with small population sizes.
- When the population size is small, the proposed mechanism does not increase the learning rate so much.

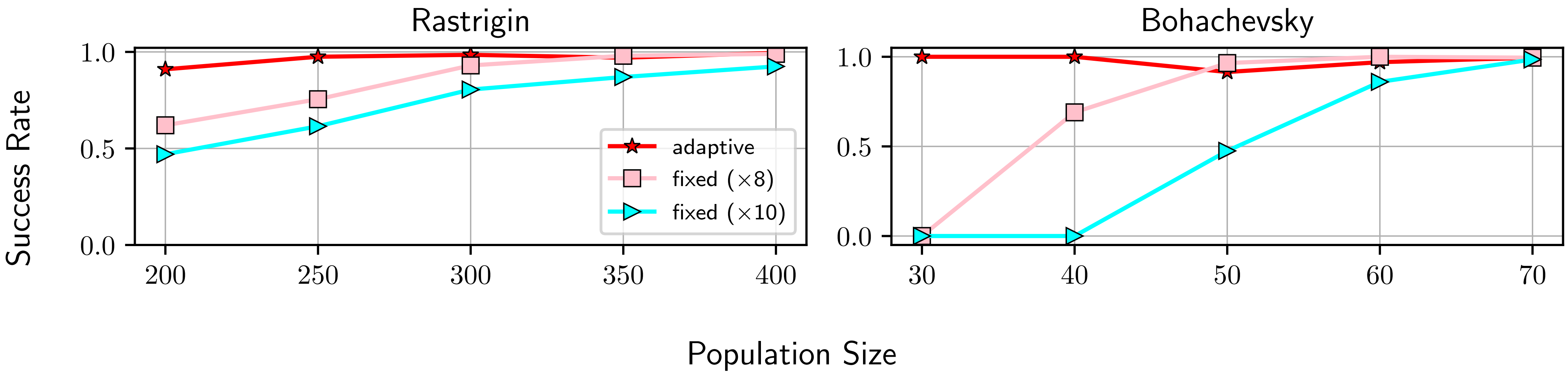
  => enables stable search

RQ3: Fixed LR vs. Adaptive LR



- In the multimodal functions, the proposed mechanism is competitive with xNES w/ high learn. rates when λ is large.
- In Rastrigin, the proposed mechanism, pink, and cyan achieve almost same performance.
- The metrics is divided by the success rate.
  - Is there a difference in these success rates?

RQ3: Fixed LR vs. Adaptive LR



- These methods are competitive when the population size is large.
- xNES with the fixed learning rates (pink and cyan) are more likely to fail when the population size is small.
- This result suggests that the proposed mechanism is more robust than xNES with fixed learning rates.

Conclusion

- Summary
  - Problem:
    - Learning rate adaptation for NES
  - Approach:
    - Adapting learning rate based on estimation accuracy of natural gradient
    - Considering KL divergence as estimation accuracy
  - Evaluation:
    - Evolution path with fixed learning rate
    - Behavior of learning rate adaptation
    - Fixed learning rate vs. adaptive learning rate
- Future work
  - Incorporating the proposed mechanism to state-of-the-art NES variants [Nomura and Ono, 2021]

References

- [Fukushima et al., 2011] N. Fukushima, Y. Nagata, S. Kobayashi, and I. Ono, “Proposal of distance-weighted exponential natural evolution strategies,” in 2011 IEEE Congress of Evolutionary Computation (CEC). IEEE, 2011, pp.164–171.
- [Glasmachers et al., 2010] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber, “Exponential Natural Evolution Strategies,” in Proceedings of the 12th annual conference on Genetic and evolutionary computation. ACM, 2010, pp.393–400.
- [Nishida and Akimoto, 2016] K. Nishida, Y. Akimoto, “Population Size Adaptation for the CMA-ES based on the Estimation Accuracy of the Natural Gradient”, In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 237–244 (2016)
- [Nishida and Akimoto, 2018] K. Nishida, Y. Akimoto, “PSA-CMA-ES: CMA-ES with population size adaptation”. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 865–872 (2018)
- [Nomura and Ono, 2021] M. Nomura, I. Ono, Natural Evolution Strategy for Unconstrained and Implicitly Constrained Problems with Ridge Structure. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–7. IEEE (2021)