Balancing exploitation and exploration in parallel Bayesian optimization under computing resource constraint

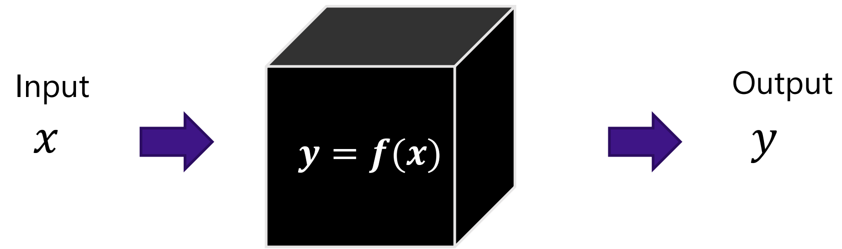
International Workshop on Automatic Performance Tuning
May 19, 2023

Moto Satake, Keichi Takahashi, Yoichi Shimomura,
and Hiroyuki Takizawa
(Tohoku University)

Background

Black Box Optimization

- The optimality is defined by an **unknown** objective function.
  - A **trial** is to calculate output y for a given input x.
  - Trials are repeated many times with different inputs to find the optimal solution, which maximizes the unknown objective function.



Unknown (the equation is not given) function

- It is required to optimize the function with a smaller number of trials especially if a trial is expensive.
  - Brute-force approaches (e.g., random search) could take a long time to find an optimal solution because of too many trials.

→ Only a promising input should be evaluated at each trial to find an optimal solution with as few trials as possible.

Background

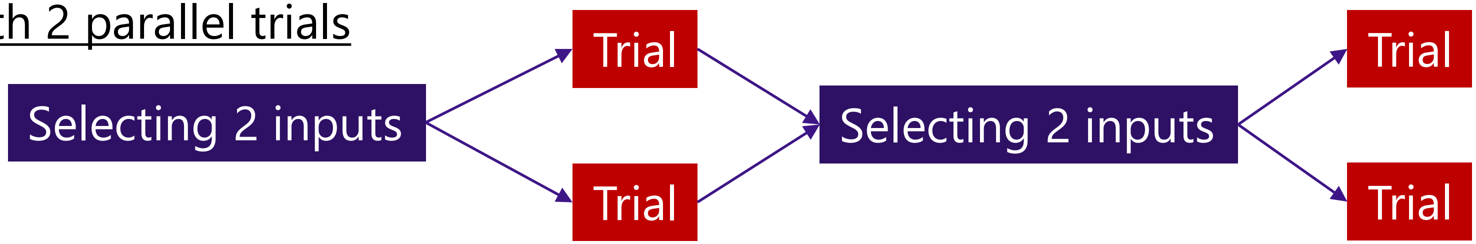
■ Bayesian Optimization (BO)

- A major approach to black box optimization with expensive trials.
  - Promising inputs are evaluated **selectively** based on the results of preceding trials.

■ Parallel Bayesian Optimization (PBO)

- The execution time would basically become shorter by making multiple trials in parallel.

PBO with 2 parallel trials



  - However, making too many trials in parallel is likely to increase the total number of trials to get a good solution [1] (as discussed later) .

[1] Chaoyi Zhang, Ryusuke Egawa, Hiroyuki Takizawa. Acceleration of Hyper-Parameter Auto-Tuning with Parallelization and Time Constraints. Poster presentation at HPC ASIA, 2020.

Background

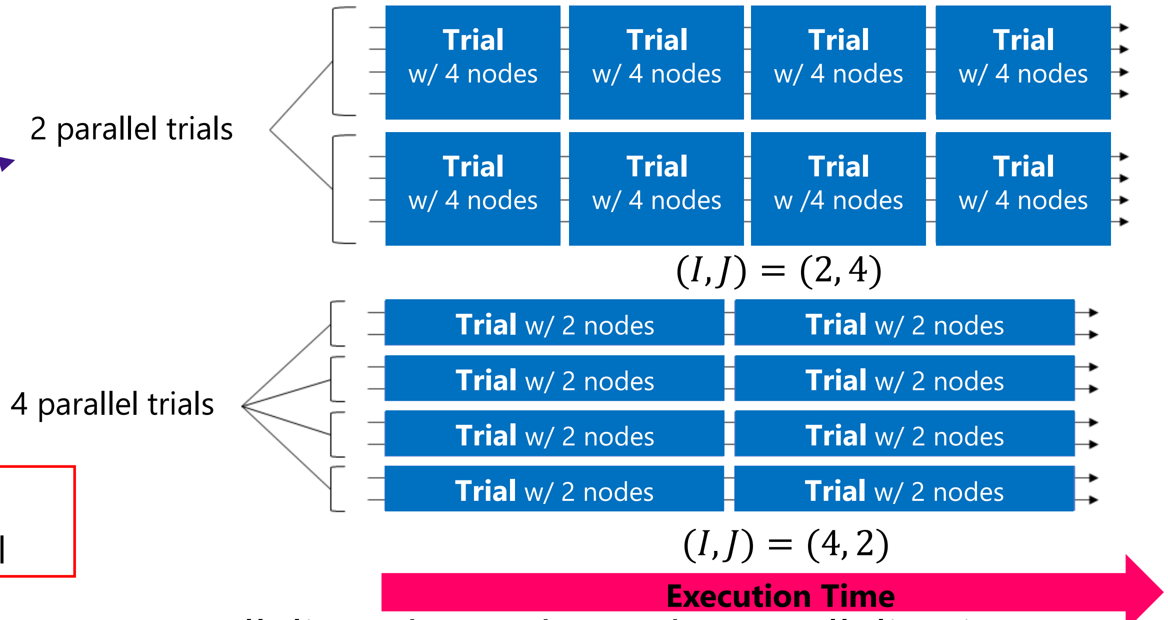
■ Suppose **a fixed number of compute nodes.**

- **More nodes** should be used for each trial (evaluation parallelism), or **more trials** should be executed in parallel (BO parallelism)?

8 nodes are available for PBO.

Make 2 parallel trials, each of which uses 4 nodes.

Make 4 parallel trials, each of which uses 2 nodes.



I : The number of parallel trials
J : The number of nodes for each trial

- The best way of using system parallelism depends on the parallelization efficiencies of PBO and each trial.
  - Both PBO and each trial do not scale linearly with the number of nodes.

→ This paper focuses on auto-tuning of *I* and *J*.

Goal and Approach

■ Goal

- Efficient execution of black box optimization by PBO with making a good use of system parallelism.
  - The total execution time is shorter

■ Approach = Auto-tuning

- The way of using system parallelism is dynamically changed by **auto-tuning** parameters I and J.
  - The best balance between *I* and *J* changes accordingly to the optimization progress.

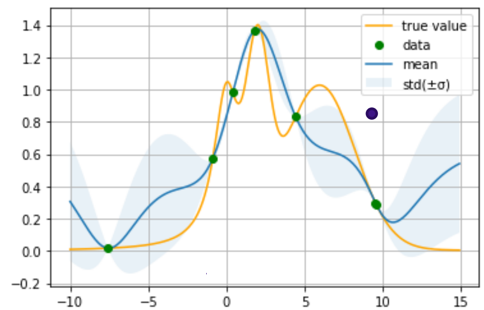
→ **This work proposes an auto-tuning method of *I* and *J*.**

What's BO?

■ Bayesian Optimization

- BO uses the results of preceding trials to find inputs that are promising to maximize the optimality defined by an unknown objective function.

① Gaussian Process Regression of unknown objective function



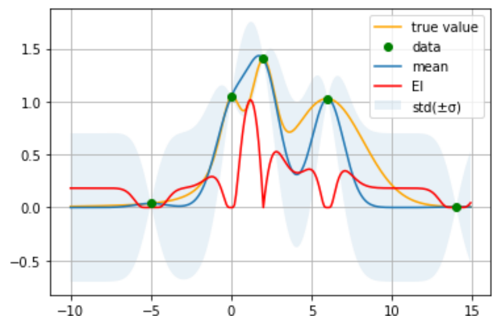
Objective function
The actual input-output relationship is unknown

Observed value
The output value observed at a trial in the past

Expected value (estimated function by GPR)

Standard deviation
becomes large in uncertain region

②Acquisition function $\alpha(x)$ to select promising inputs

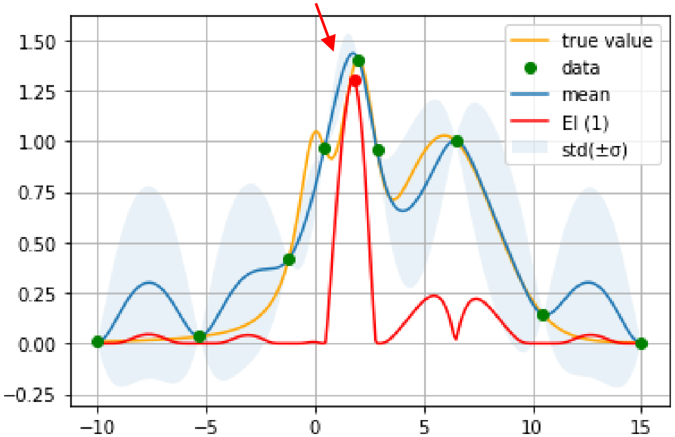


- The acquisition function is defined by using the expectation and standard deviation.
  - **Expected Improvement (EI)** is used in this work.
- The input with the maximum value is selected.
  - The input has a larger expectation (exploitive) or a larger standard deviation (explorative).

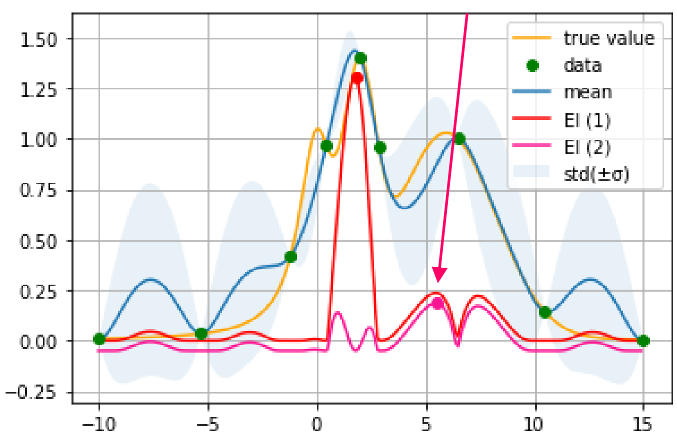
What's PBO?

■ Parallel Bayesian Optimization

①



②



- Multiple inputs are selected at once.
  - Once an input is selected, the next one is selected near the previous one by

$$x^{(q)} = \mathrm{argmax}\left(a_{\mathrm{EI}} \cdot \prod_{\mathrm{i}=1}^{\mathrm{q}-1} [1 - \exp(-\frac{1}{2\theta} \parallel x - x^{(\mathrm{i})} \parallel^2)] \right)$$

$x^{(q)}$: the q-th input to be selected

① The input with maximum EI is selected.

② The EI is updated nearby the input selected at ①

③ The updated EI is used to select the next input.

・The next input must not be the same as the previous one. → The input with the second-largest EI is selected.

Exploration and Exploitation

Balancing exploration and exploitation

- **Exploration** : search uncertain regions with larger standard deviations to find an unexplored better region.
- **Exploitation** : search promising regions with larger expectations to find a better solution within an explored promising region.

→ Both are needed and important for efficient optimization.

How to balance exploration and exploitation

- Exploitation is not efficient **at first** because GPR is inaccurate [2].
  - There is no information about the objective function at first, and it is impossible for BO to select appropriate inputs.

→ BO should be more explorative until GPR becomes accurate after enough trials, and then become more exploitative.

[2] Stefan Falker, Aaron Klein, Frank Hutter. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. Proceedings of the 35 th International Conference on Machine Learning(ICML 2018). July. 2018.

Parallel Execution and Balance

■ Execution configuration of PBO

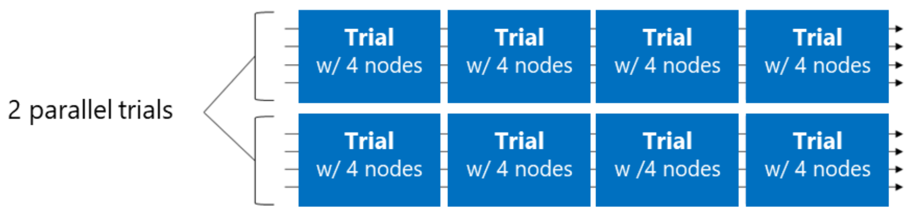
- If J (the number of nodes for one trial) is smaller, I (the number of parallel trials) can be larger to increase the throughput of trials.
- If I is larger, non-promising inputs are being evaluated.

⇨ PBO becomes more explorative.

(I, J)=(4,2)

More explorative

(I, J)=(2,4)



More exploitative

→ The balance between exploration and exploitation can be changed by adjusting I and J accordingly to optimization progress.

Autotuning of *I* and *J*

Adapting to the optimization progress

- When I parallel trials are done,
  - **The best-ever solution is found?**
  - **How much the GPR accuracy is improved?**

Check if the best-ever solution is found

- Not found

$$y_{max,c} \le y_{max,c-1}$$

$y_{max,c}$: the best-ever solution at cycle c

Increase *I*

  - The input space should be extensively searched ⇨ **explorative**

- Found

$$y_{max,c} > y_{max,c-1}$$

Increase *J*

  - The region near the solution should be searched ⇨ **exploitative**

Autotuning of *I* and *J* (Cont'd)

GPR accuracy: coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_{obs,i} - y_{pred,i})^2}{\sum_{i=1}^{N}(y_{obs,i} - \bar{y}_{obs})^2}$$

y_{obs} : observed value
y_{pred} : predicted value

Observed and predicted values are compared.

Higher is better. $R^2 = 1$ if perfect.

The accuracy generally improves with the number of trials.

Change in coefficient of determination

- Improvement is larger than that at the previous cycle.

$$R_c^2 - R_{c-1}^2 > R_{c-1}^2 - R_{c-2}^2$$

Increase *I*

  - More data will further improve the accuracy ⇨ **explorative**

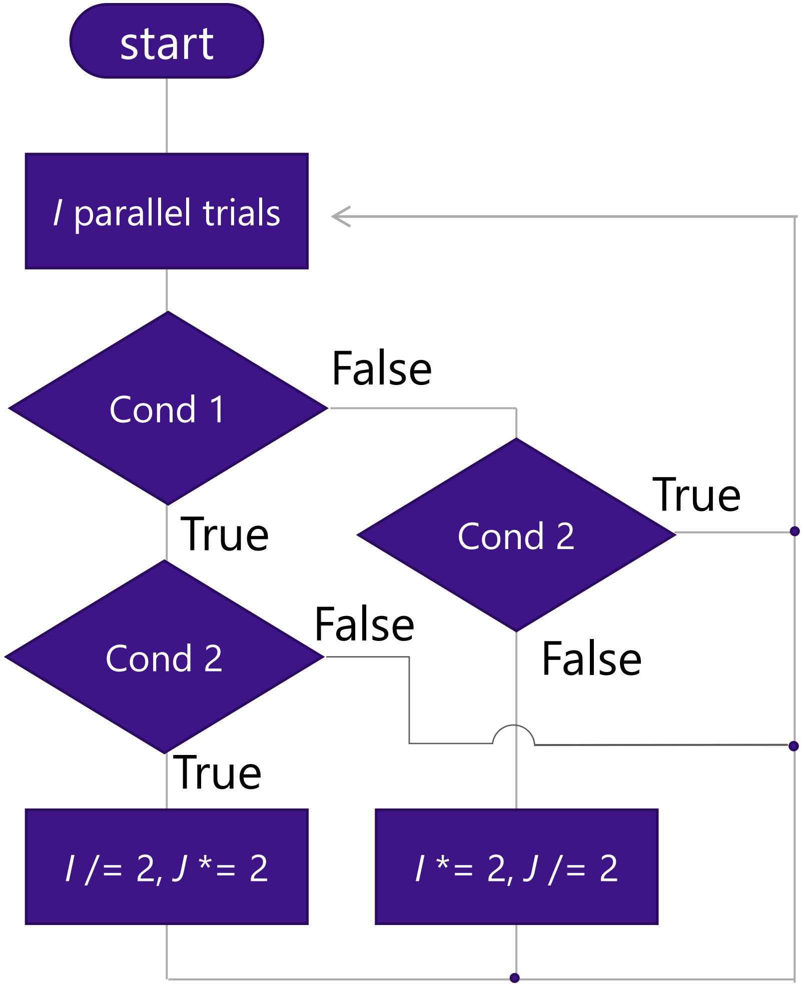
- Improvement is smaller than that at the previous cycle.

$$R_c^2 - R_{c-1}^2 \leq R_{c-1}^2 - R_{c-2}^2$$

Increase *J*

  - A sufficient amount of data have been obtained ⇨ **exploitative**

PBO with AT



Cond 1: $y_{max,c} > y_{max,c-1}$

Cond 2: $R_c^2 - R_{c-1}^2 \leq R_{c-1}^2 - R_{c-2}^2$

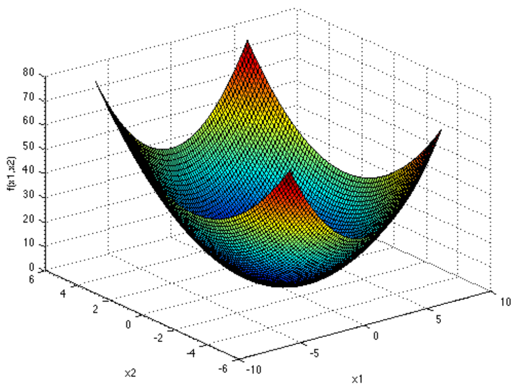
- **Start with explorative search**
- **Conditions 1 and 2 are checked**
  - The best-ever solution found?
  - The accuracy improved?
- **If both are true, be more exploitative**
  - *I* /=2, *J* *= 2
- **If both are false, be more explorative**
  - *I* *=2, *J* /= 2
- **Otherwise, *I* and *J* remain unchanged**

Evaluation Setup

関数	入力値範囲	最大値 y^*
Sphere	[-50, 50]	0
Ackley	[-50, 50]	0
Michalewicz($d = 5$)	[0,5]	4.687658
Michalewicz($d = 10$)	[0,5]	9.66015

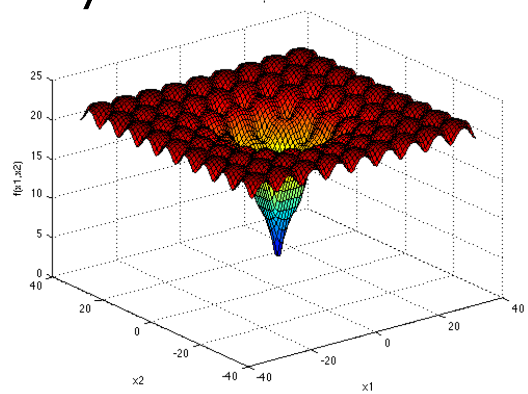
■ Optimization Problems

- 3 benchmark functions with different characteristics
- The dimensionality d is 5 or 10



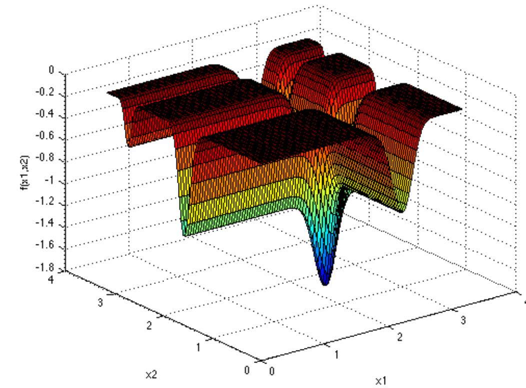
Sphere function [3]
・Simple convex

$$f_{Sphere}(\mathbf{x}) = \sum_{i=1}^{d} x_i^2.$$



Ackley function [3]
・many local minima

$$f_{Ackley}(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20.$$



Michalewicz function [3]
・flat regions
・smaller input ranges

$$f_{Michalewicz}(x) = -\sum_{i=1}^{d} \sin(x_i)(\sin)^{20}\left(\frac{i x_i^2}{\pi}\right)$$

[3] Sonja Surjanovic and Derek Bingham. 2013. Virtual Library of Simulation Experiments：Test Functions and Datasets. Last accessed Jan. 26, 2023 https://www.sfu.ca/~ssurjano/index.html

Evaluation Setup (Cont'd)

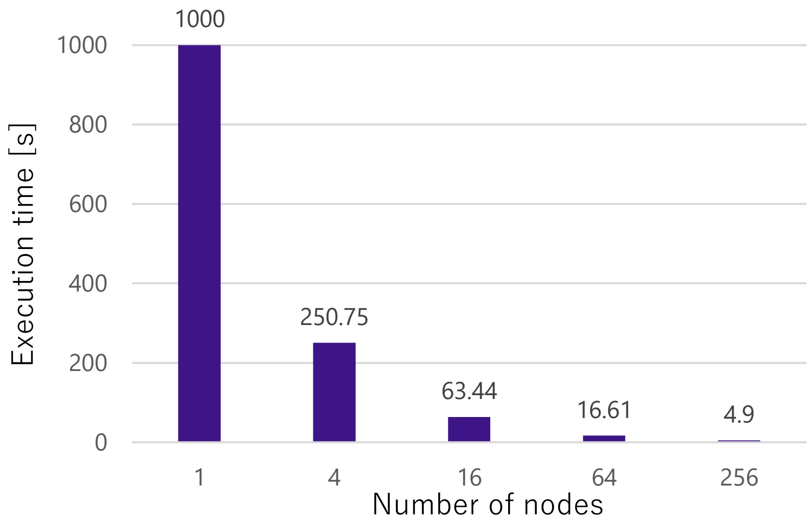
Execution time

- Each trial is expensive and parallelized with J nodes
  - Amdahl's law

$$t_J = t_1 \cdot (1 - \alpha + \frac{\alpha}{J})$$

t_J : parallel execution time
J : the number of nodes
α : parallelization ratio

$t_1 = 1000s$
$\alpha = 0.999$



Other conditions

- The total number of trials: 1024 (initial random search : 256 trials)
- System parallelism $I \times J = 256$
  - Initial condition: $(I, J) = (256,1)$ → most explorative configuration
- 30 runs to calculate the average results

Best-Ever Solution Improvement

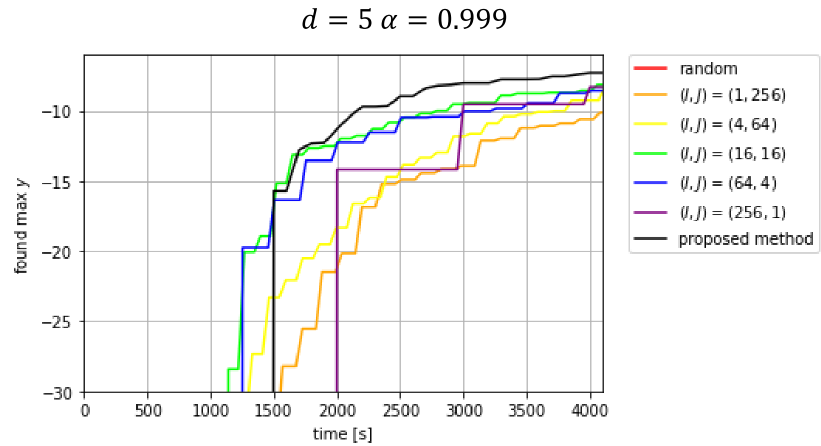
■ Evaluation Results

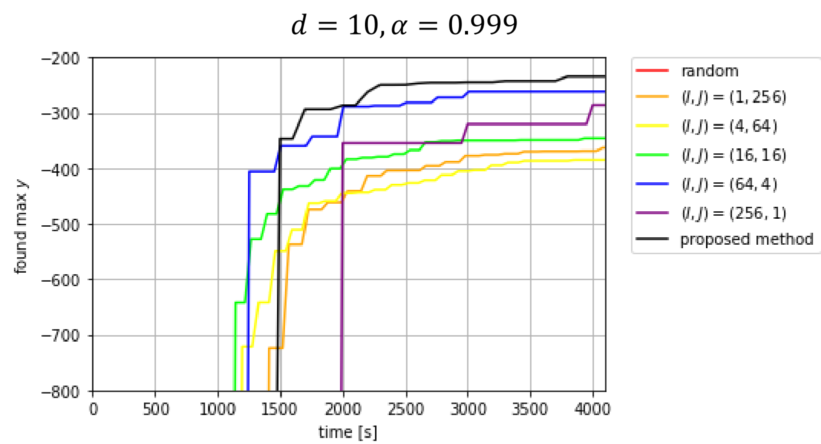
- Horizontal Axis : Elapsed Time [s]
- Vertical Axis : The output of the best solution found at the time. (**best-ever solution**)

■ Sphere Function

- The proposed method quickly improves the solution and finally finds a better solution.

⇨ AT can improve the efficiency

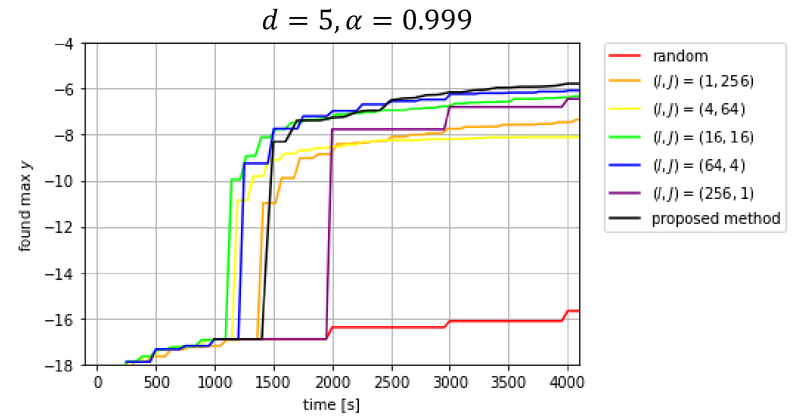




Best-Ever Solution Improvement

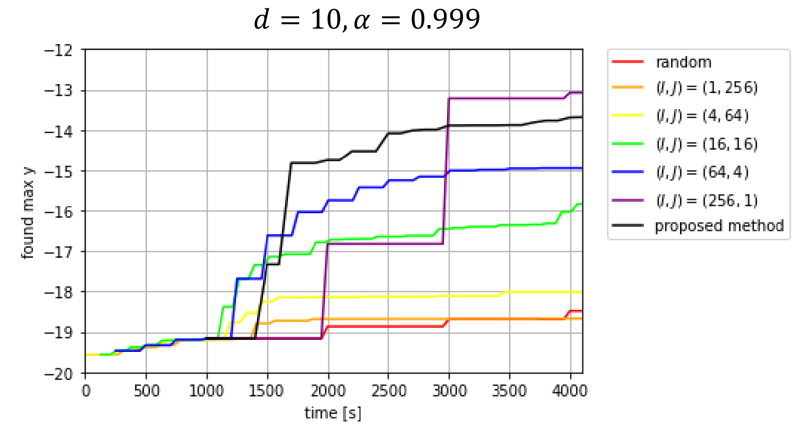
■ Ackley ($d = 5$)

- The proposed method can achieve a better final solution.
- The improvement is as fast as $(I,J) = (16,16), (64,4)$.



■ Ackley (d =10)

- The best solution is found with $(I,J) = (256,1)$.
  - The improvement is slow.
- The proposed method can achieve the second-best solution.



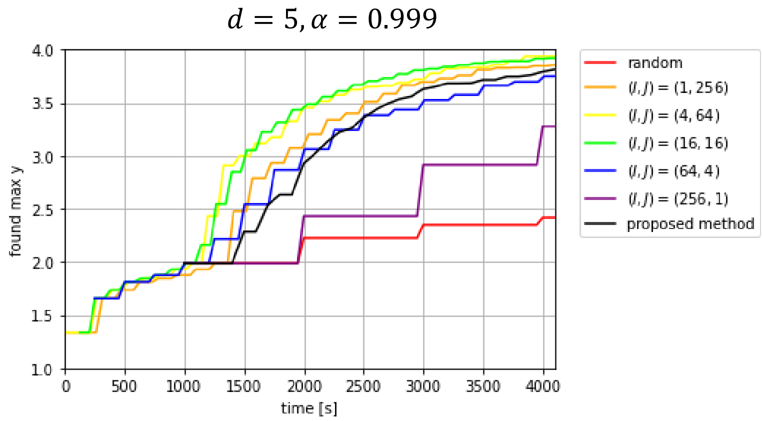
⇨ The proposed can appropriately adjust the parameters and achieve comparable performance to the best parameter configuration.

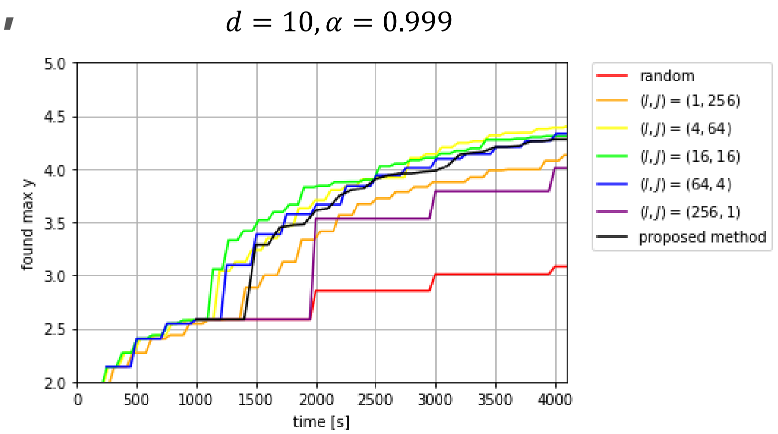
Best-Ever Solution Improvement

■ Michalewicz

- The proposed method shows slow improvement at first.
- The difference becomes smaller at the later stage of optimization.

The proposed method starts with the most explorative parameters (I,J)=(256,1), and becomes exploitative only gradually.





Summary

■ Purpose

- Efficient execution of black box optimization by PBO with making a good use of system parallelism under computing resource constraint.

■ Approach = Auto-tuning

- System parallelism is used for whether executing **more parallel trials** or using **more computing nodes** for each trial.
  - Balancing exploration and exploitation accordingly to the optimization progress.

■ Evaluation

- The proposed method can stably achieve a fast improvement and a good final result.
  - Many parallel trials are executed in parallel at the early stage to reduce the total execution time.
  - Each trial is accelerated by using many nodes at the later stage of the optimization to find a better solution.