Fast Multidimensional Performance Parameter Estimation with Multiple One-dimensional d-Spline Parameter Search

Masayoshi Mochizuki, Akihiro Fujii, Teruo Tanaka
Kogakuin University

Outline

1. Introduction
2. Proposed Method
3. Numerical Tests
4. Conclusion

Background

- Software automatic performance tuning (AT)
  - Technology to automatically optimize software according to computer environment and numerical problems
  - Search for optimum values from parameters (performance parameters) that affect performance
  - There are many applications with multiple performance parameters space

- We have studied the estimation in multidimensional space
  - We have proposed incremental performance parameter estimation (IPPE) method as performance estimation method
  - Implementation of IPPE for 2-dimensional parameter space was described in the following paper †

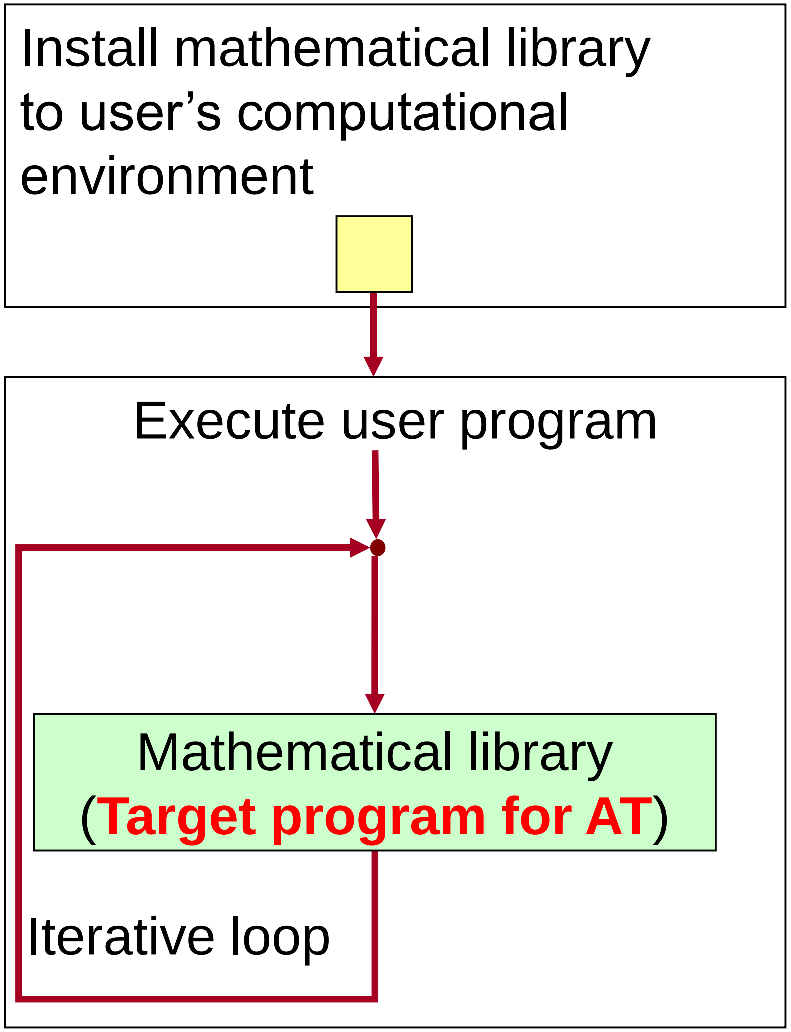
† Riku Murata, Jun Irie, Akihiro Fujii, Teruo Tanaka, Takahiro Katagiri,
Enhancement of Incremental Performance Parameter Estimation on ppOpen-AT,
In IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-15),pp.203-210,2015

Objective

- When the number of parameters increases, the total number of parameter value combinations increases exponentially
  - Computational cost for tuning parameters also becomes huge in our method

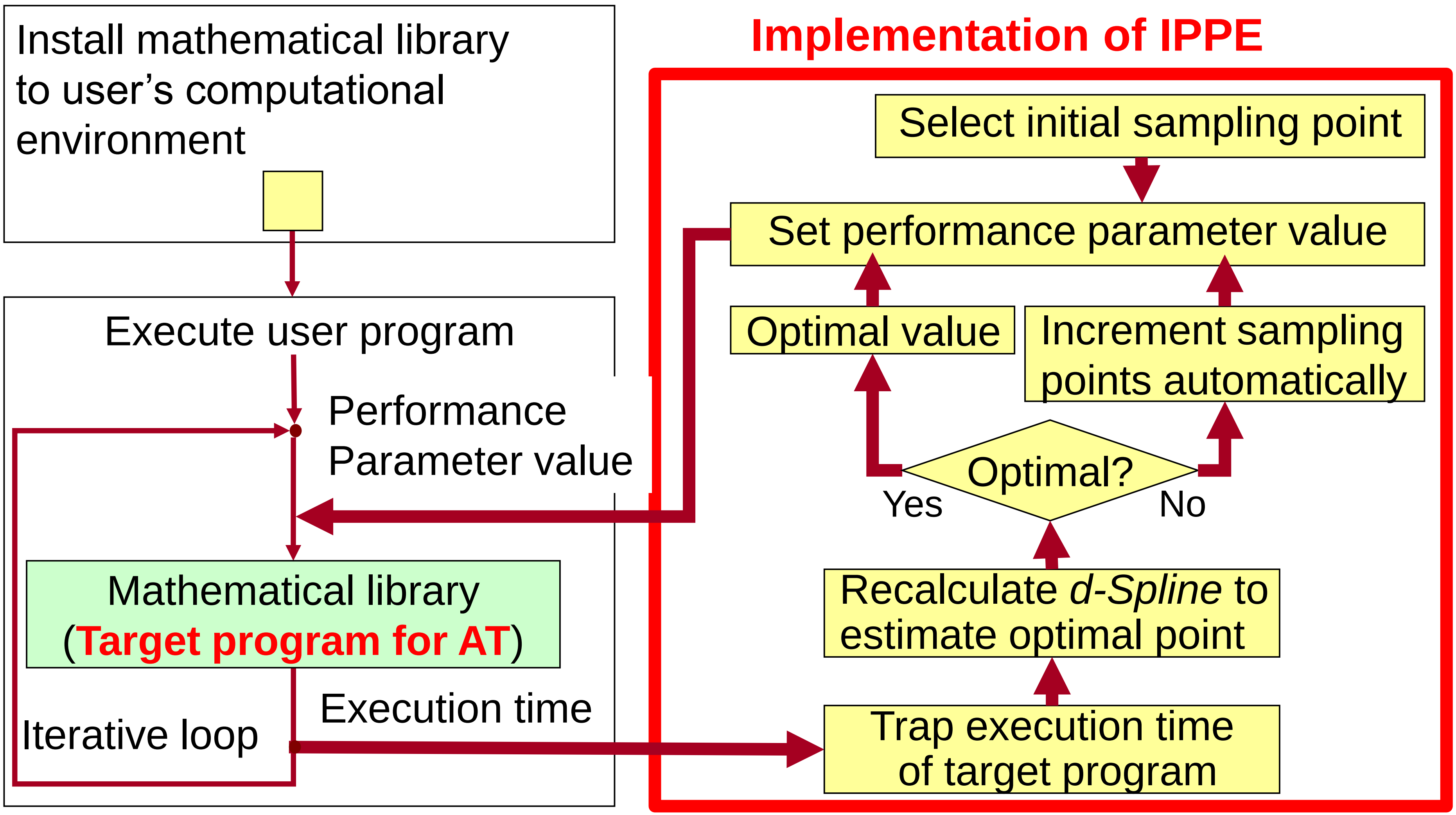
⇒We study a AT method which requires less computational cost and which can be applied to multiple performance parameter cases

IPPE method
(Incremental Performance Parameter Estimation)



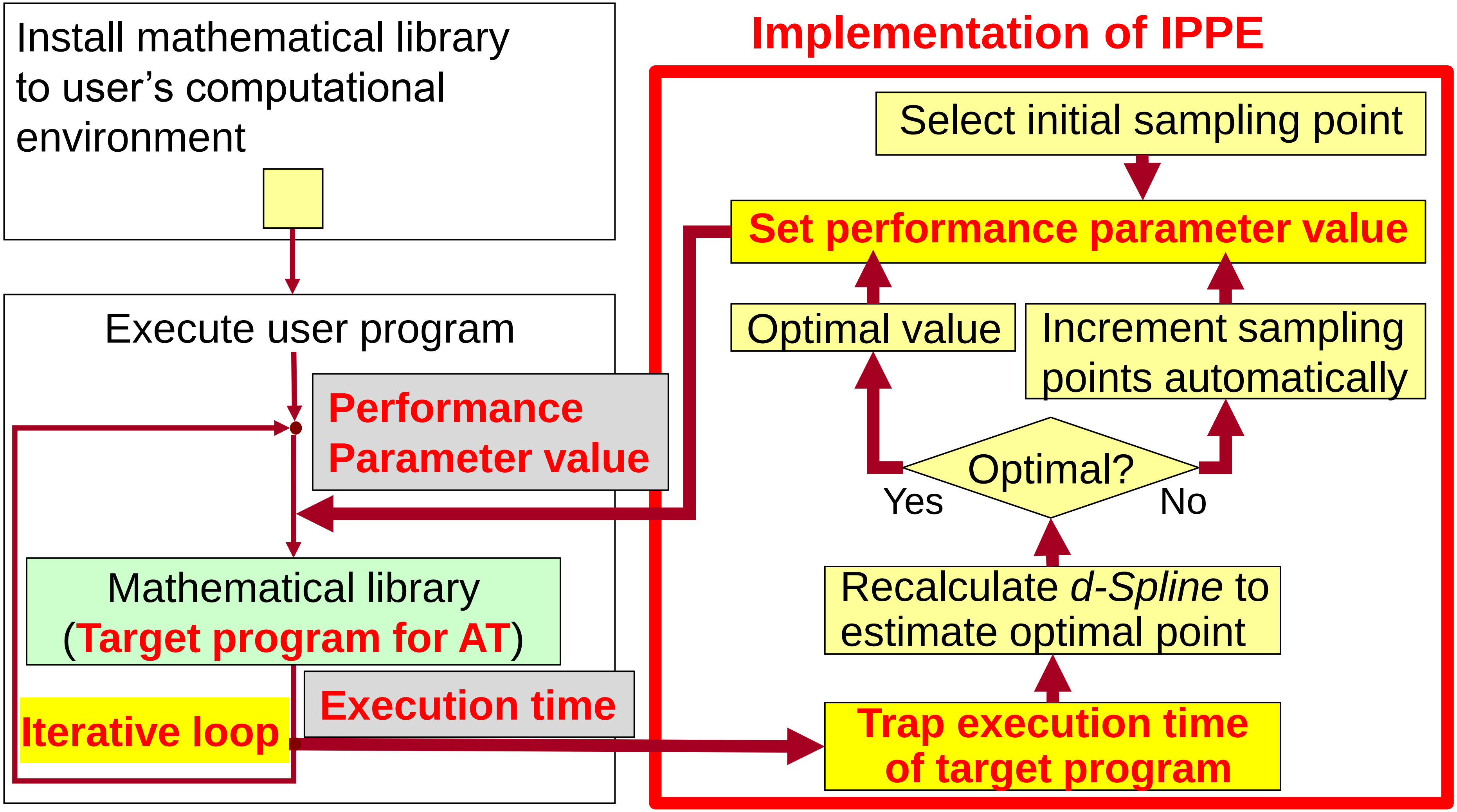
IPPE method

(Incremental Performance Parameter Estimation)



IPPE method

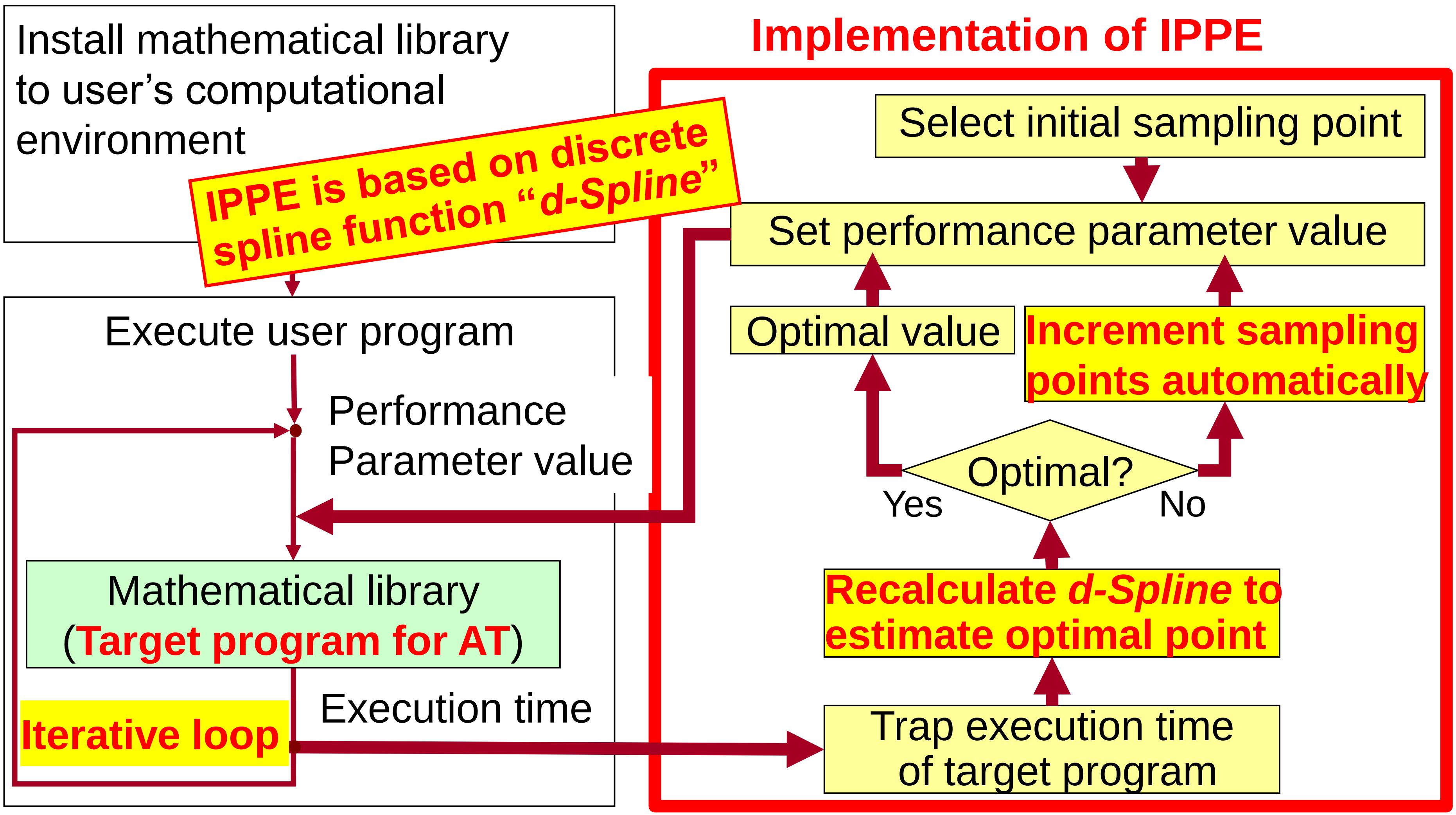
(Incremental Performance Parameter Estimation)



IPPE method

(Incremental Performance Parameter Estimation)

Kogakuin University



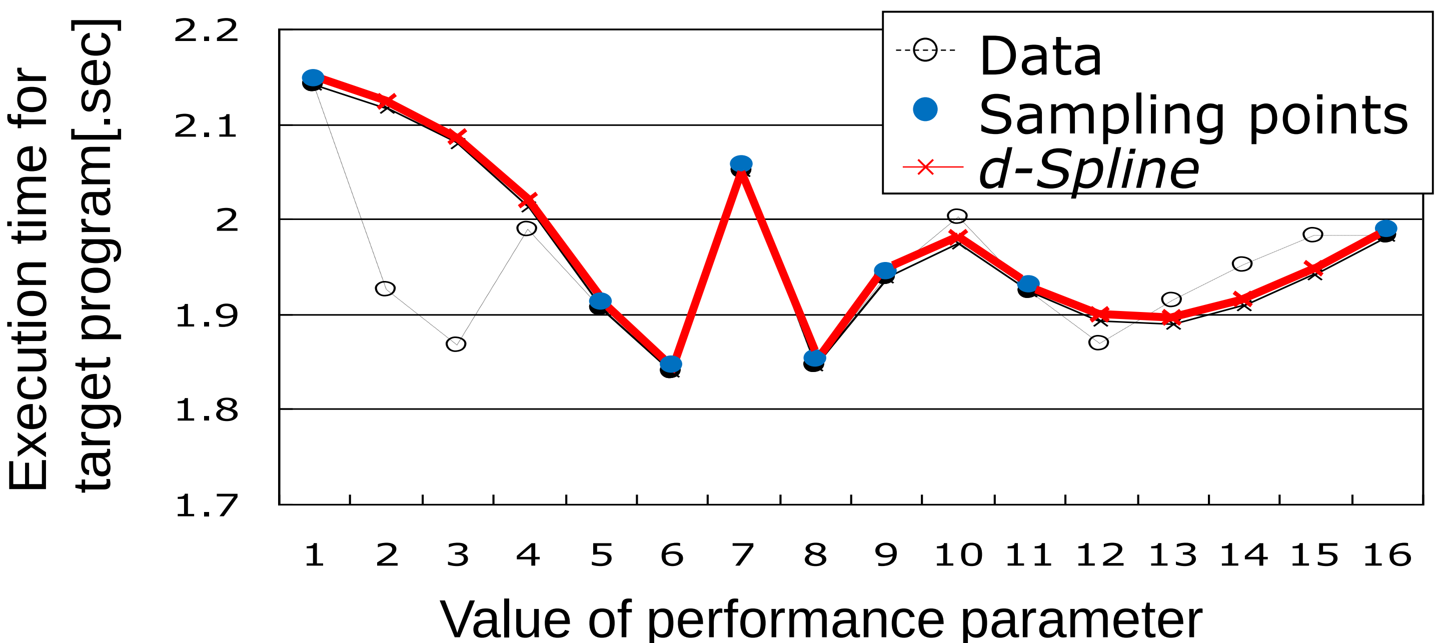
Fitting function d-Spline

- d-Spline f is denoted by $f = (f_1, f_2, \cdots, f_i, \cdots, f_n)^t$
- To determine d-Spline, smoothness of f is defined by

$$\left|f_{j-1} - 2f_j + f_{j+1}\right|, 2 \leq j \leq n-1$$

- Select d-Spline to minimize the following equation

$$\min\left(\left|\left|y - Ef\right|\right|^2 + \alpha^2 \left|\left|Df\right|\right|^2\right)$$



Multidimensional d-Spline

- Multidimensional d-Spline was used in IPPE method for estimation of multiple performance parameters
  - Smoothness of f can be calculated as second order difference

1-dimensional : $\left|f_{j-1} - 2f_j + f_{j+1}\right|, 2 \le j \le n_1 - 1$

2-dimensional : $\left|f_{j-1,k} + f_{j,k-1} - 4f_{j,k} + f_{j,k+1} + f_{j+1,k}\right|,$

$2 \le j < n_1 - 1, 2 \le k \le n_2 - 1$

n_i:performance parameter value

$$min(\|y - Ef\|^2 + \alpha^2 \|\boldsymbol{D} f\|^2)$$

1D performance parameter

$$D = \begin{pmatrix} 1 & -2 & 1 & & & & 0 \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & 0 & & & \ddots & & \\ & & & & 1 & -2 & 1 \end{pmatrix}$$

$(n_1 - 2) \times n_1$

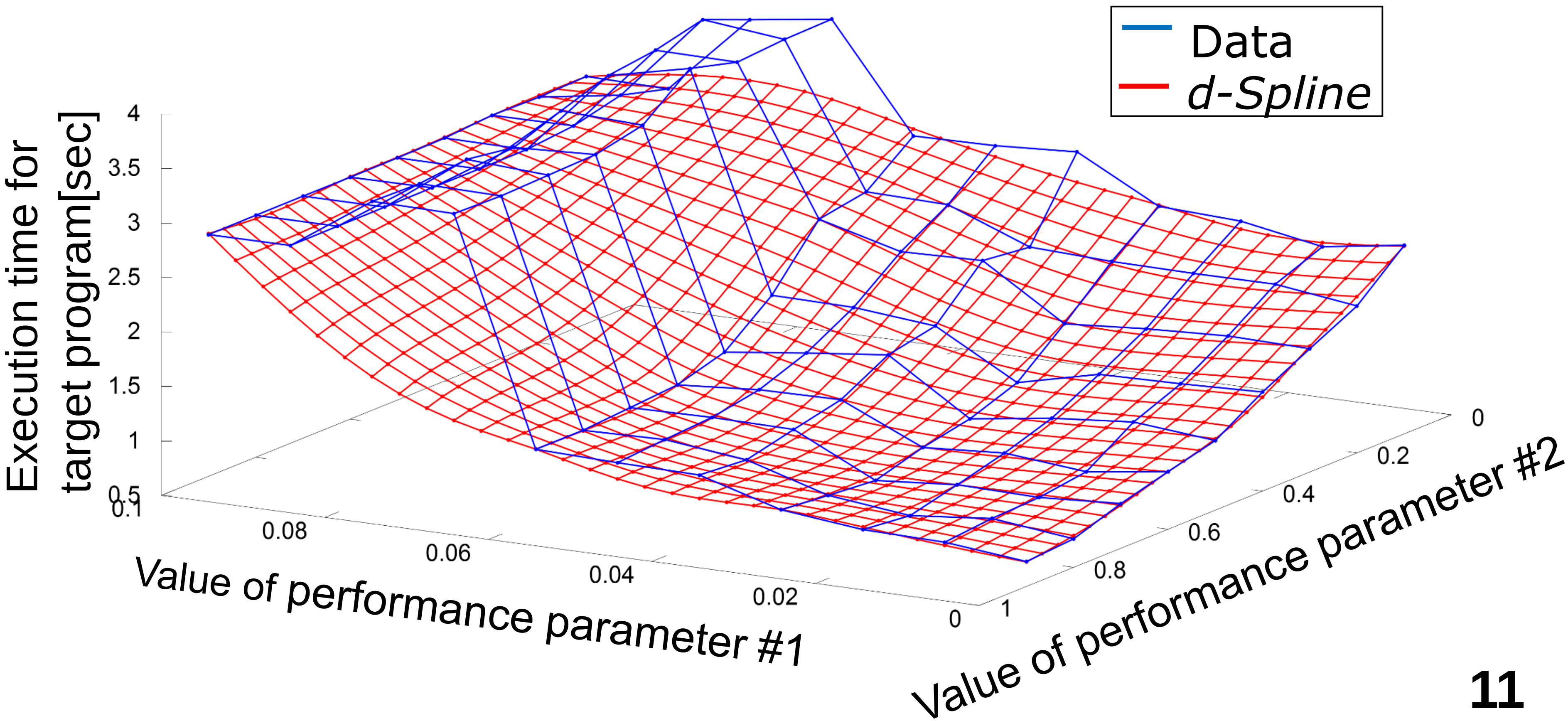
2D performance parameter

$$D_{n_1,n_2} = \begin{pmatrix} 1 & -2 & 1 & & & & & & & & & \\ & & \ddots & & & & & & & & & \\ & & & 1 & -2 & 1 & & & & & & 0 \\ 1 & & & & -2 & & & 1 & & & & \\ & 1 & & & 1 & -4 & 1 & & 1 & & & \\ & & & & & \ddots & & & & & & \\ & & 1 & & & 1 & -4 & 1 & & 1 & & \\ & & & 1 & & & -2 & & & & 1 & \\ & & & & & & & \ddots & & & & \\ & & & & & & & & \ddots & & & \\ & & 0 & & & & & & & 1 & -2 & 1 \\ & & & & & & & & & & \ddots & \\ & & & & & & & & & 1 & -2 & 1 \end{pmatrix}$$

$(n_1 \times n_2 - 4) \times (n_1 \times n_2)$

Multidimensional d-Spline

- Estimation of 2-dimensional performance parameter
  - It is the result of obtained by adding 16 points as a sample point and approximating by d-Spline
  - Target program is AMG method



Computational cost of d-Spline

- Computational cost of d-Spline increases exponentially according to number of performance parameter

Multidimensional performance parameter(*PP*)	Computational cost
One *PP* case	$O(n_1)$
Two *PP* case	$O(n_2^3 \times n_1)$
Three *PP* case	$O(n_2^3 \times n_3^3 \times n_1)$

n_i:Number of *PP* value

Computational cost of d-Spline

- Computational cost of d-Spline increases exponentially according to number of performance parameter

Multidimensional performance parameter(*PP*)	Co
One *PP* case	$\boldsymbol{O(n_1)}$
Two *PP* case	$O(n_2^3 \times n_1)$
Three *PP* case	$O(n_2^3 \times n_3^3 \times n_1)$

Computational cost is very small by our invention

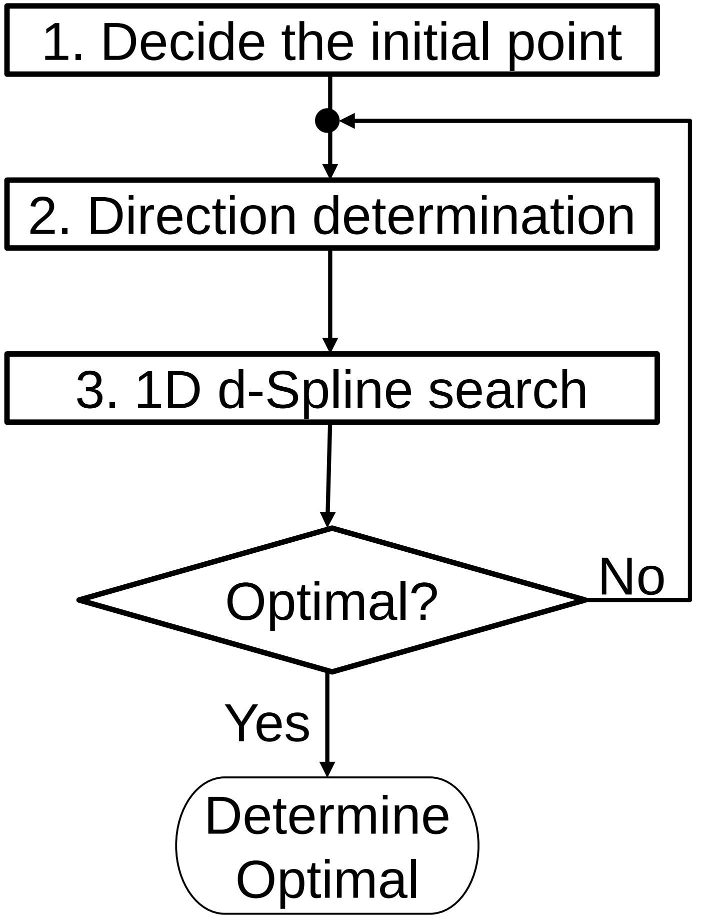
n_i:Number of *PP* value

⇒Fast d-Spline parameter estimation
for multidimensional parameter space is realized
by using one-dimensional search repeatedly

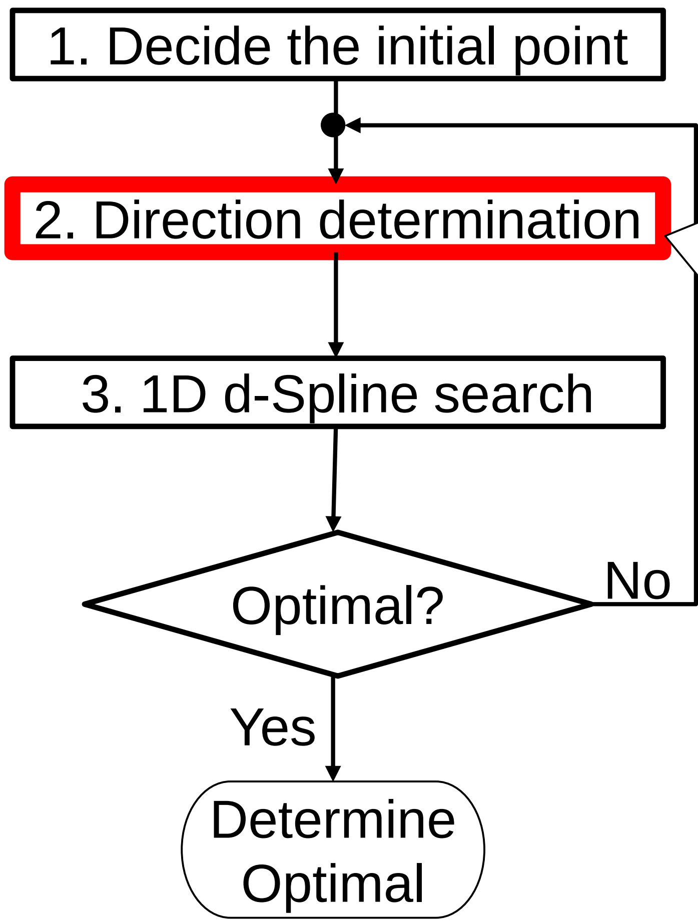
Our proposed method

- Using iteration of 1D d-Spline search for the estimation of multiple performance parameters
  - We evaluate the method with a 3-dimensional parameter space

Algorithm for repeating 1D d-Spline search

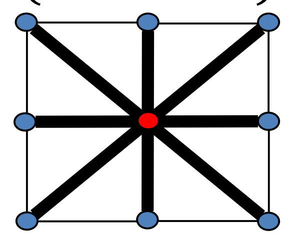


Algorithm for repeating 1D d-Spline search

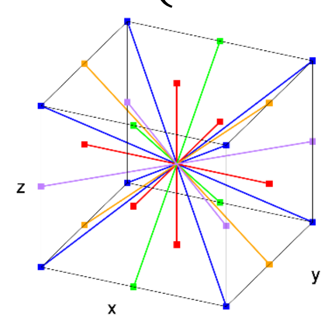


Determine the search direction that has a minimum value among the surrounding points

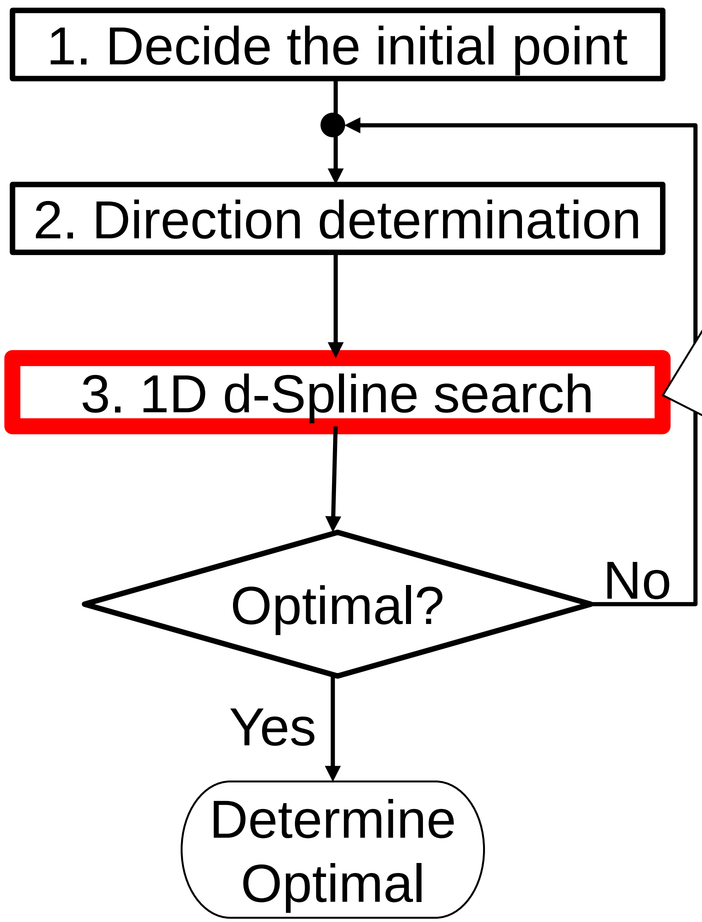
- 2 performance parameter : $8(3 \times 3 - 1)$



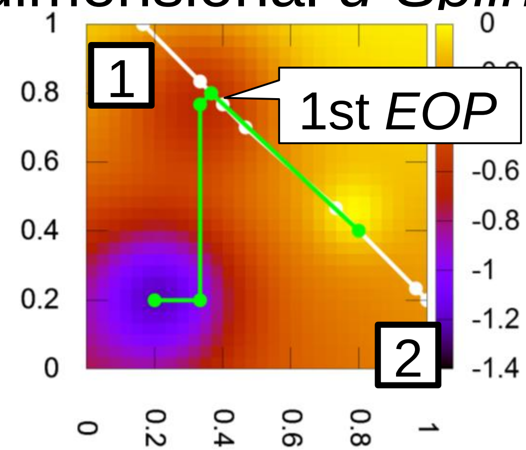
- 3 performance parameter : $26(3 \times 3 \times 3 - 1)$



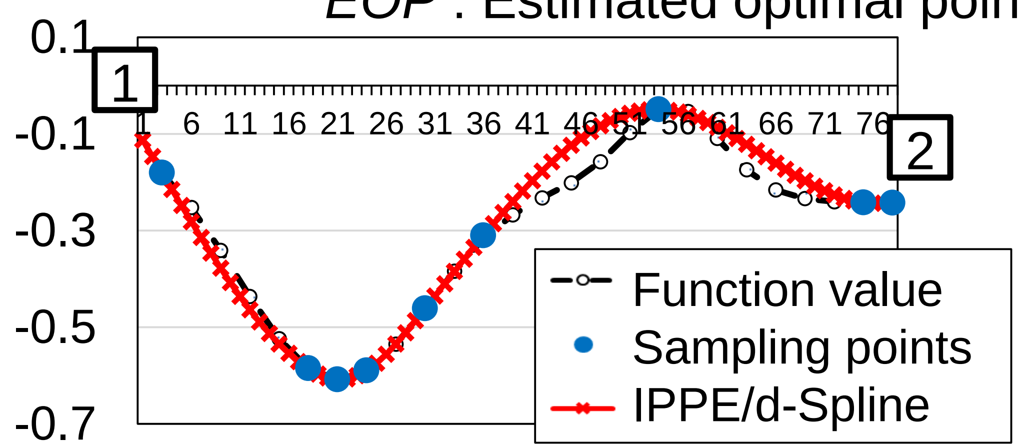
Algorithm for repeating 1D d-Spline search



Estimate the optimal point in the one dimensional space using one-dimensional *d-Spline* search

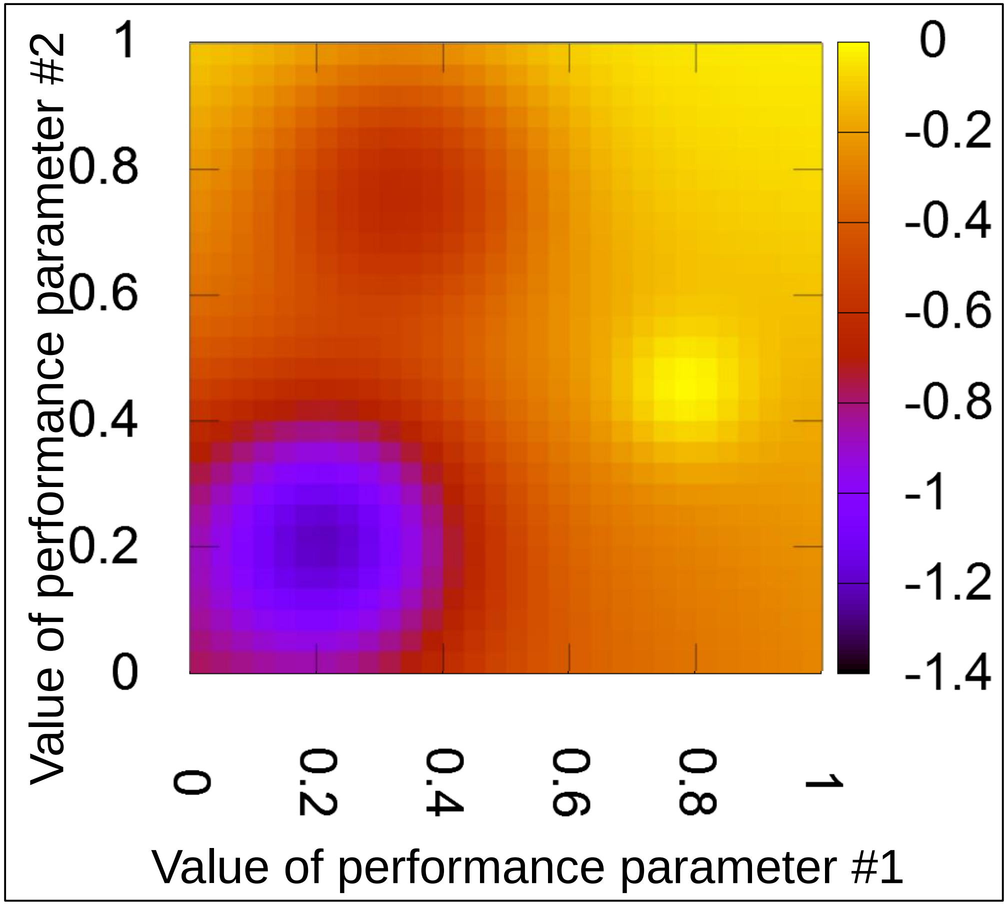


EOP : Estimated optimal point



Algorithm for repeating 1D d-Spline search

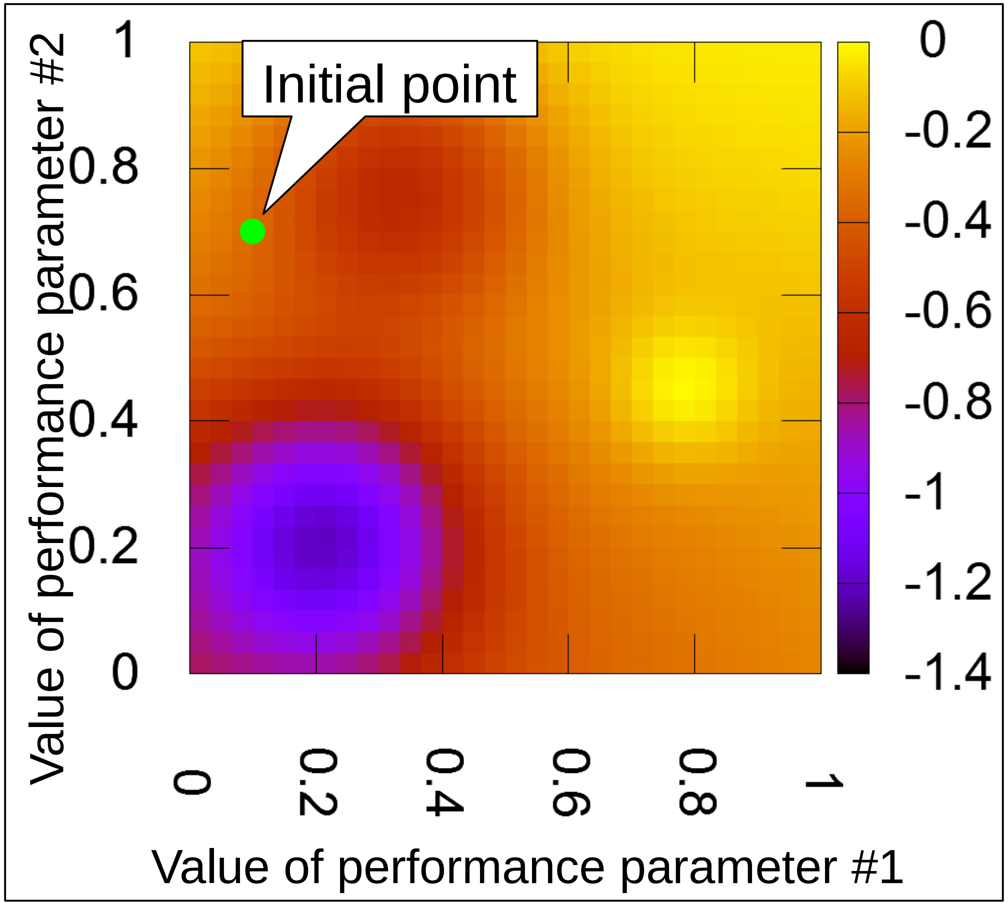
Contour map of 2-dimension（Test function：Franke）



- Contour map of Franke function
- Yellow part is high and blue part is low
- Two dips and one top
- Each performance parameter value is 31
- Our object is estimated optimal point

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

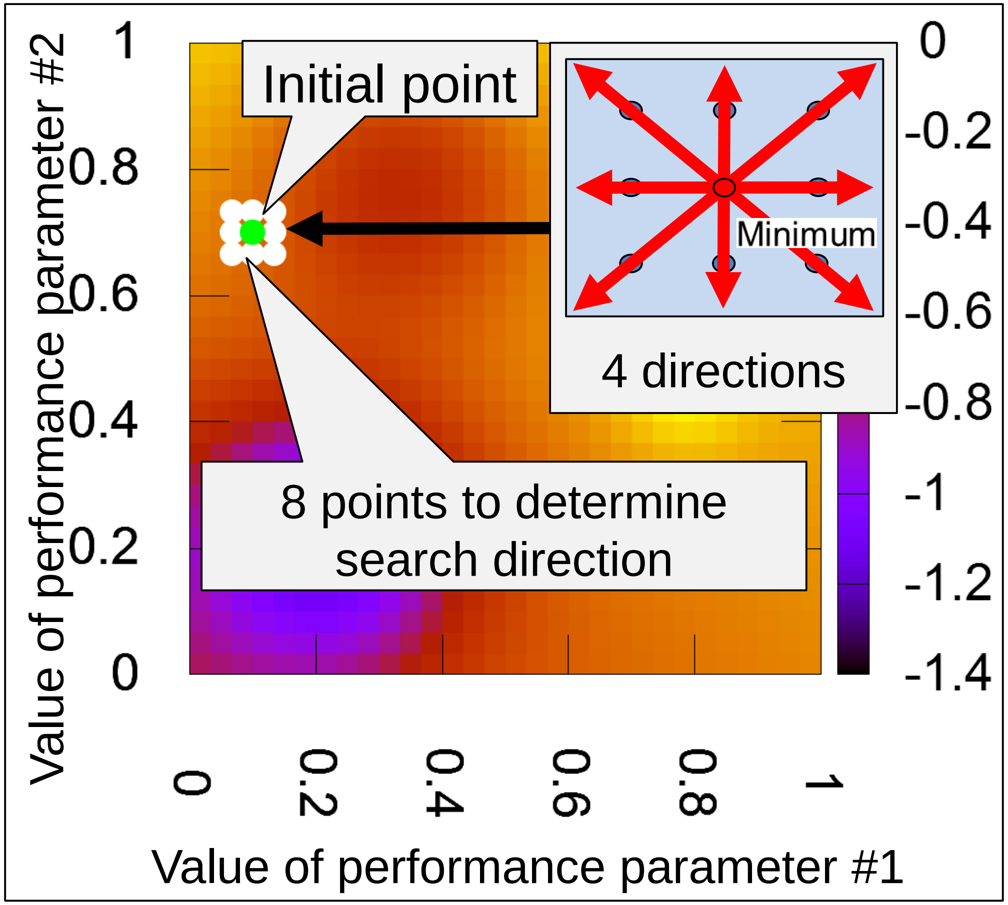
EOP : Estimated optimal point

Sampling points：

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

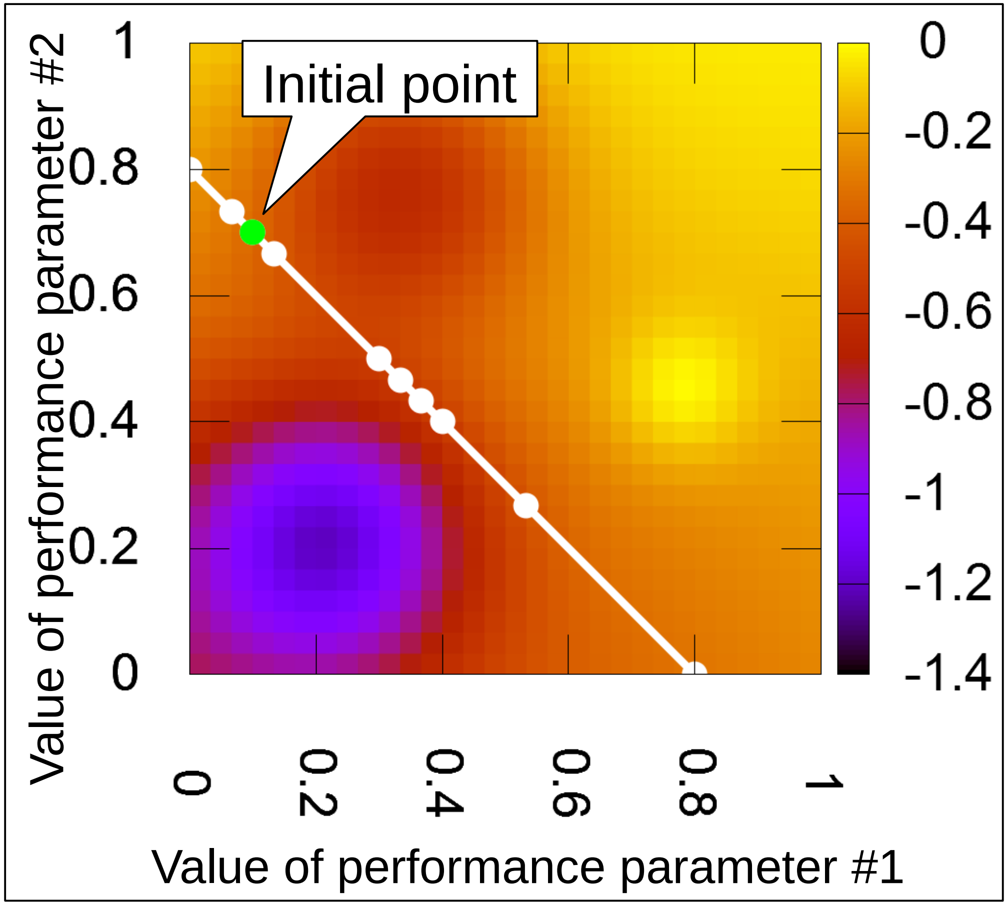
EOP : Estimated optimal point

Sampling points：+8 = 8

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

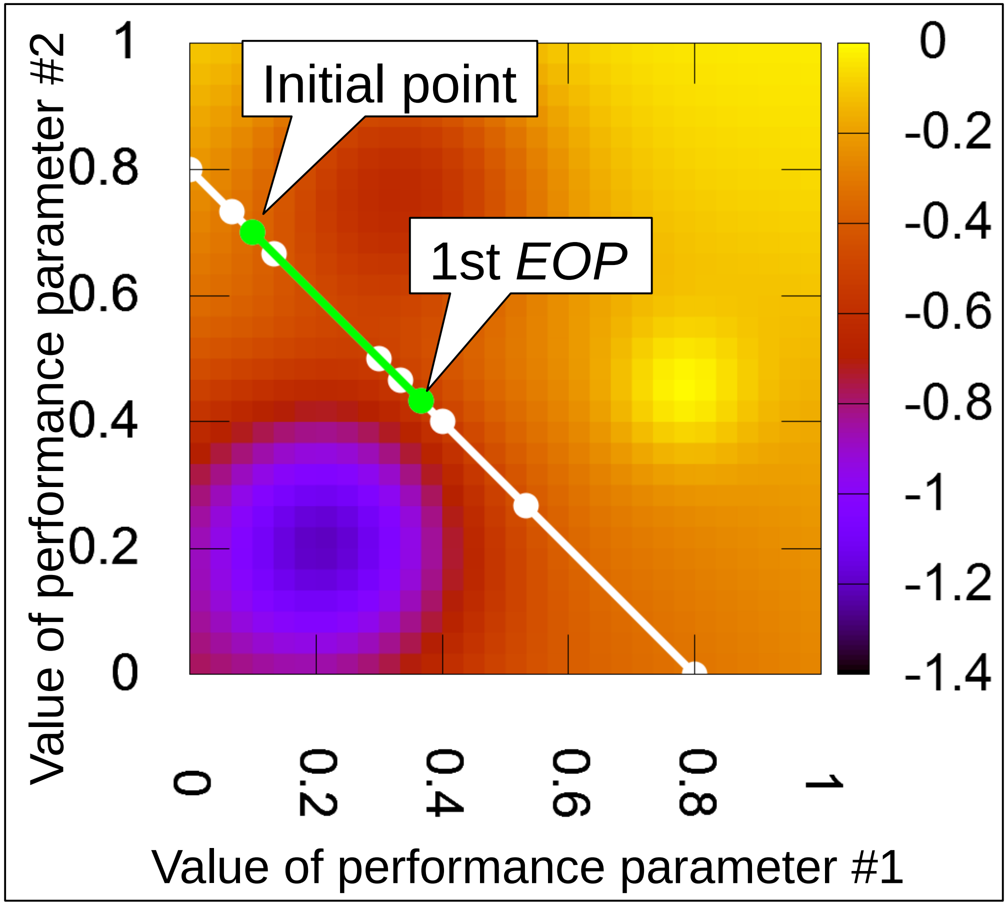
EOP : Estimated optimal point

Sampling points：8 + 7 = 15

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

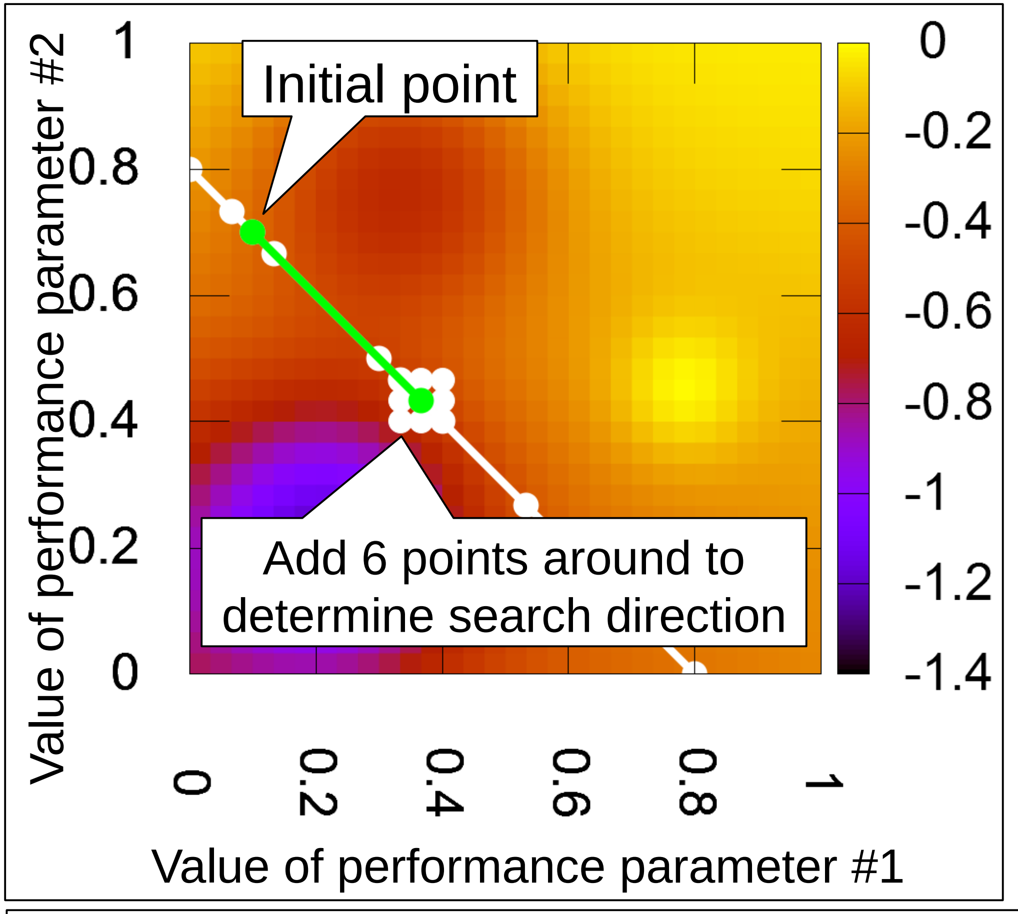
EOP : Estimated optimal point

Sampling points：8 + 7 = 15

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

EOP : Estimated optimal point

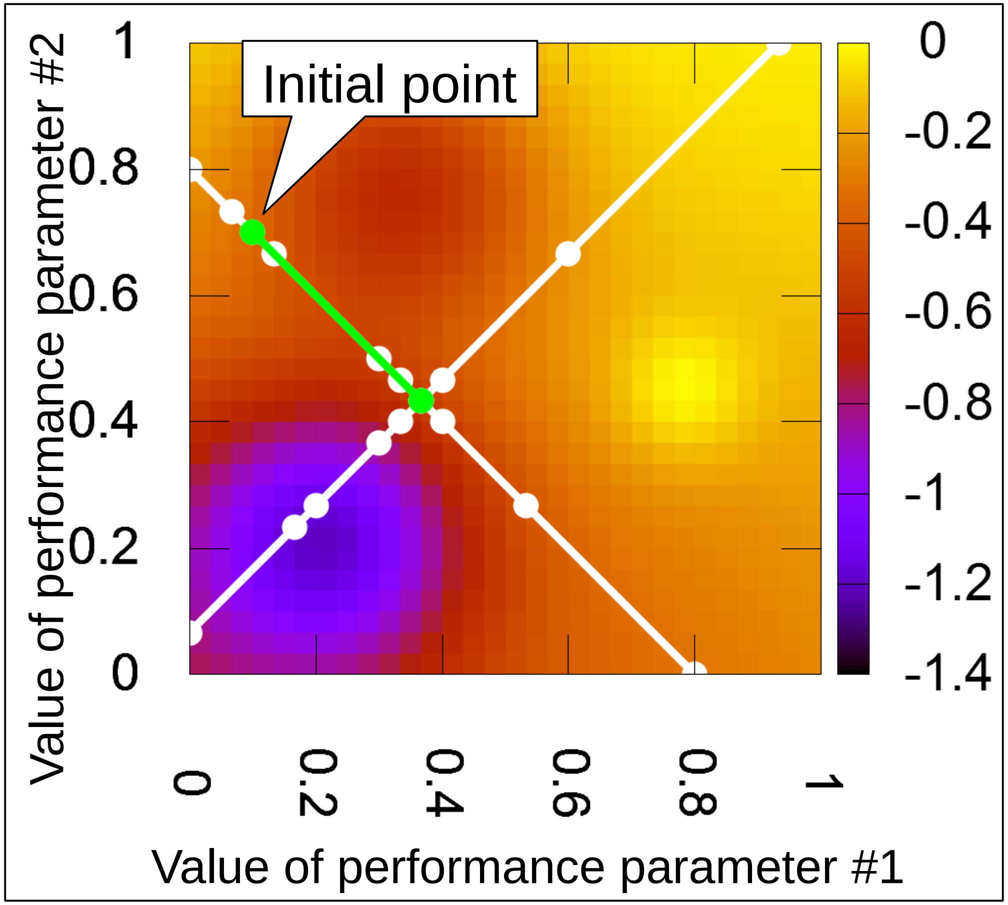
Sampling points：8 + 7 + 6 = 21

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension (Test function : Franke)

Kogakuin University



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

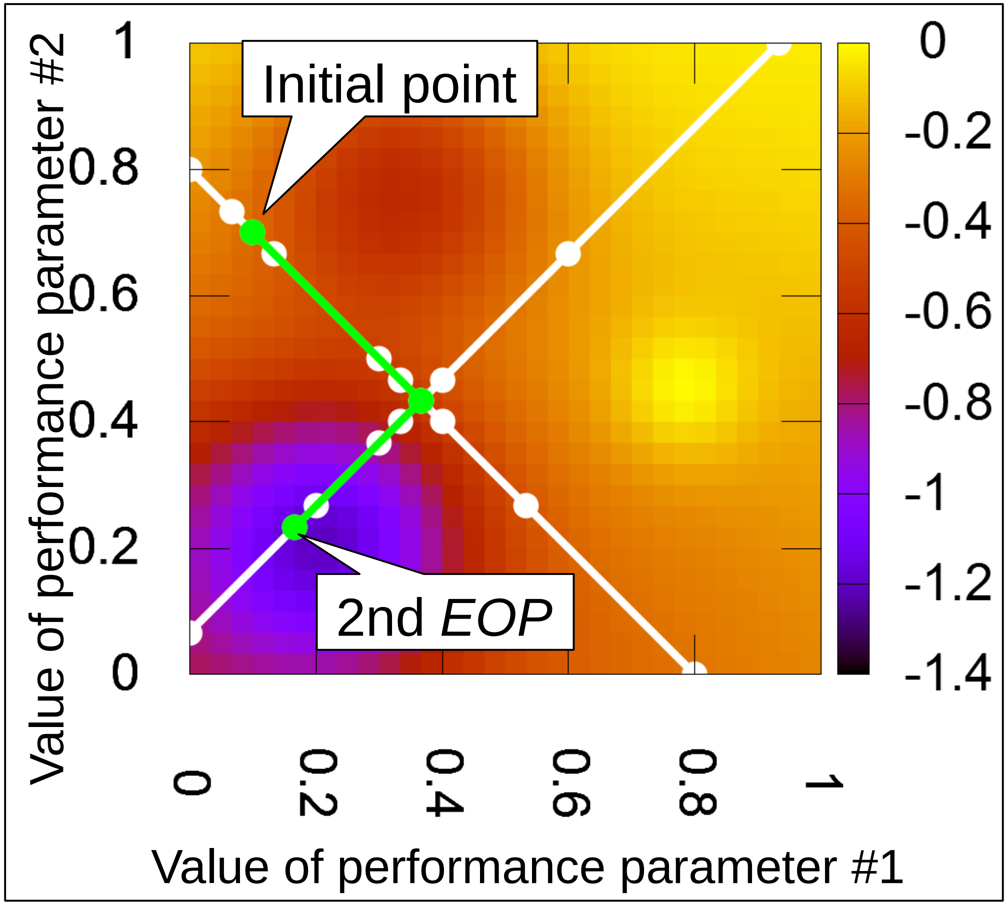
EOP : Estimated optimal point

Sampling points : 8 + 7 + 6 + 6 = 27

1st step	2nd step		3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

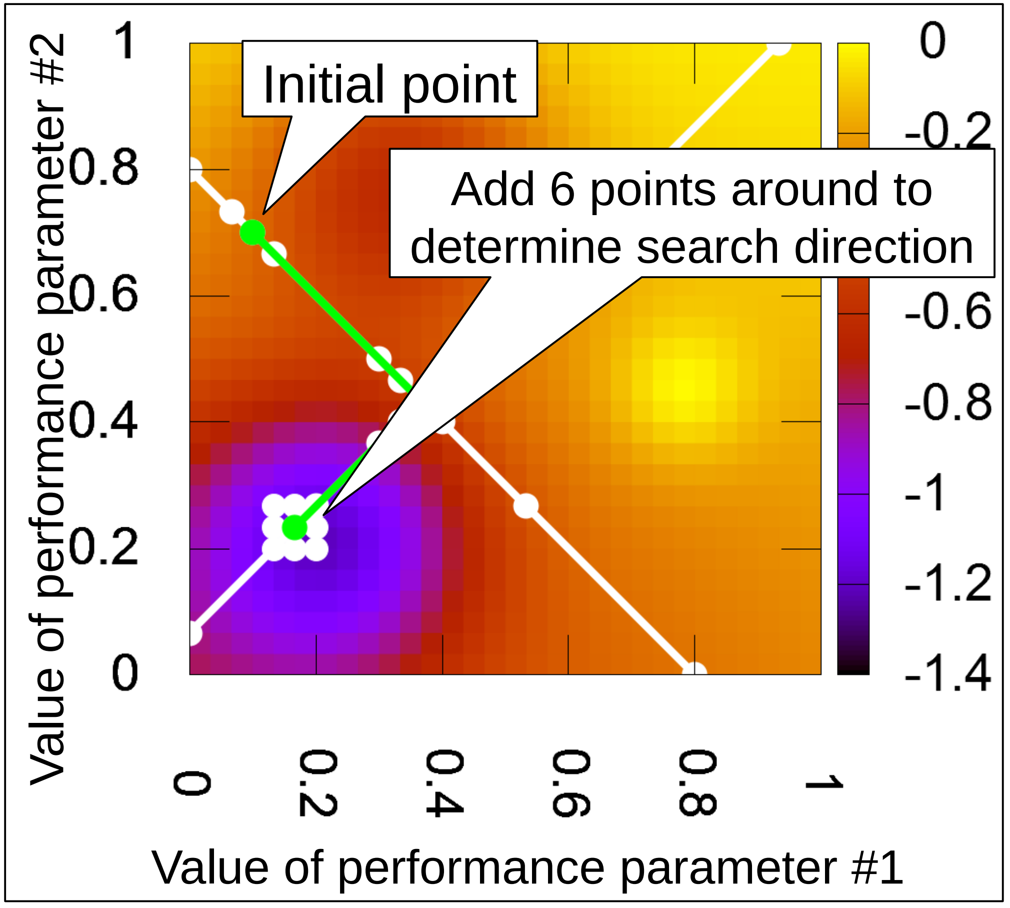
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 = 27

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

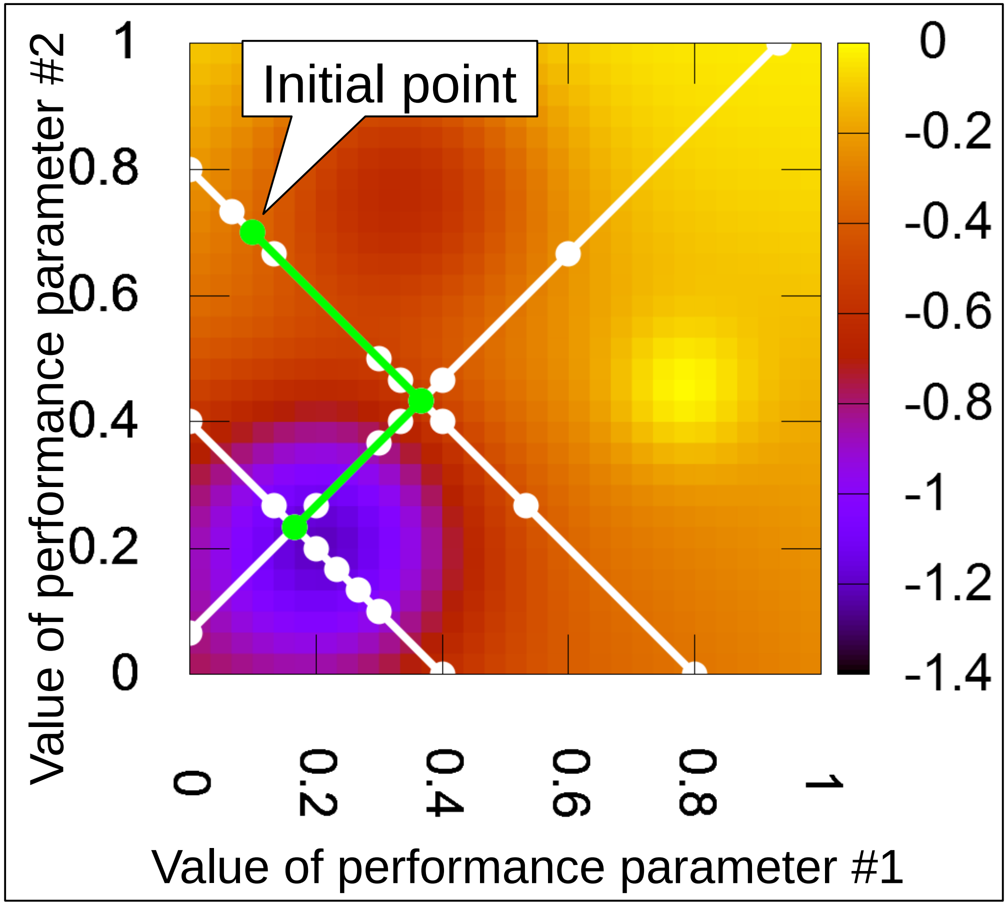
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 + 6 = 33

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

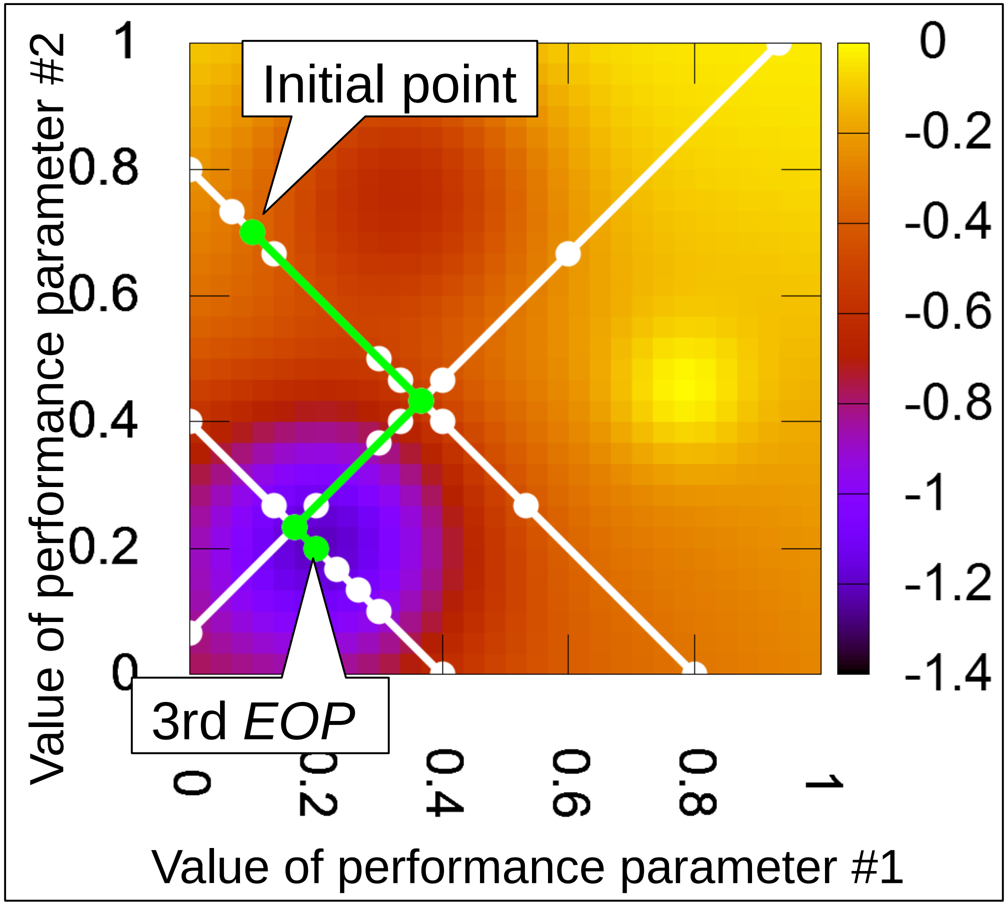
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 + 6 + 5 = 38

1st step	2nd step	3rd step		4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

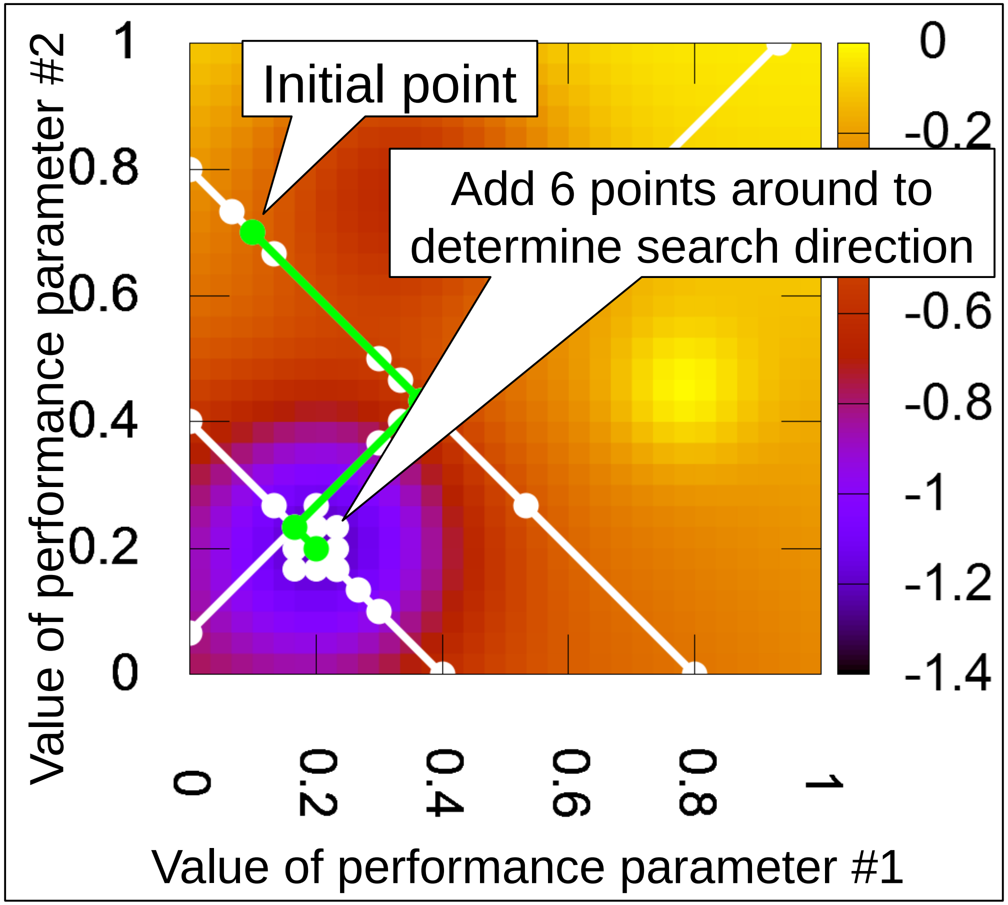
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 + 6 + 5 = 38

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

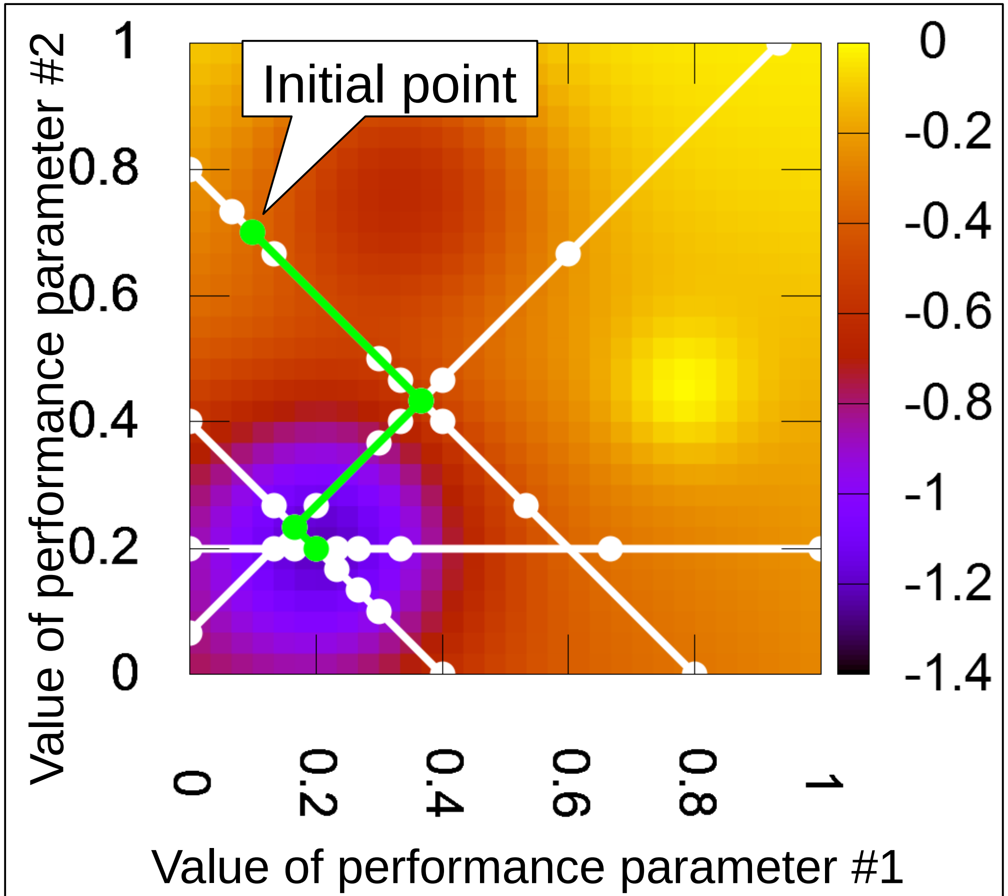
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 + 6 + 5 + 6 = 44

1st step	2nd step	3rd step	4th step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

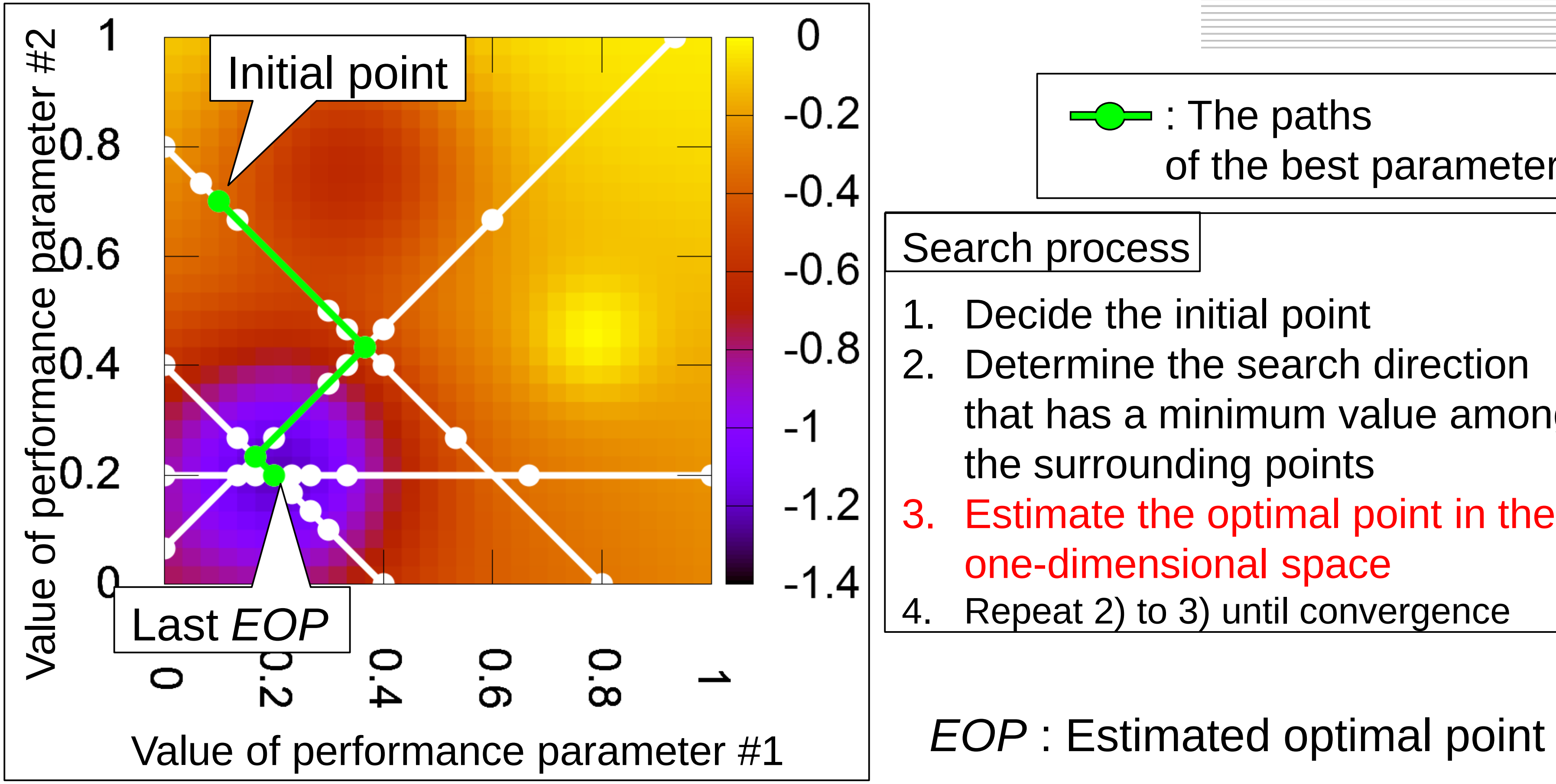
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 + 6 + 5 + 6 + 6 = 50

1st step	2nd step	3rd step	4th step	

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points
3. Estimate the optimal point in the one-dimensional space
4. Repeat 2) to 3) until convergence

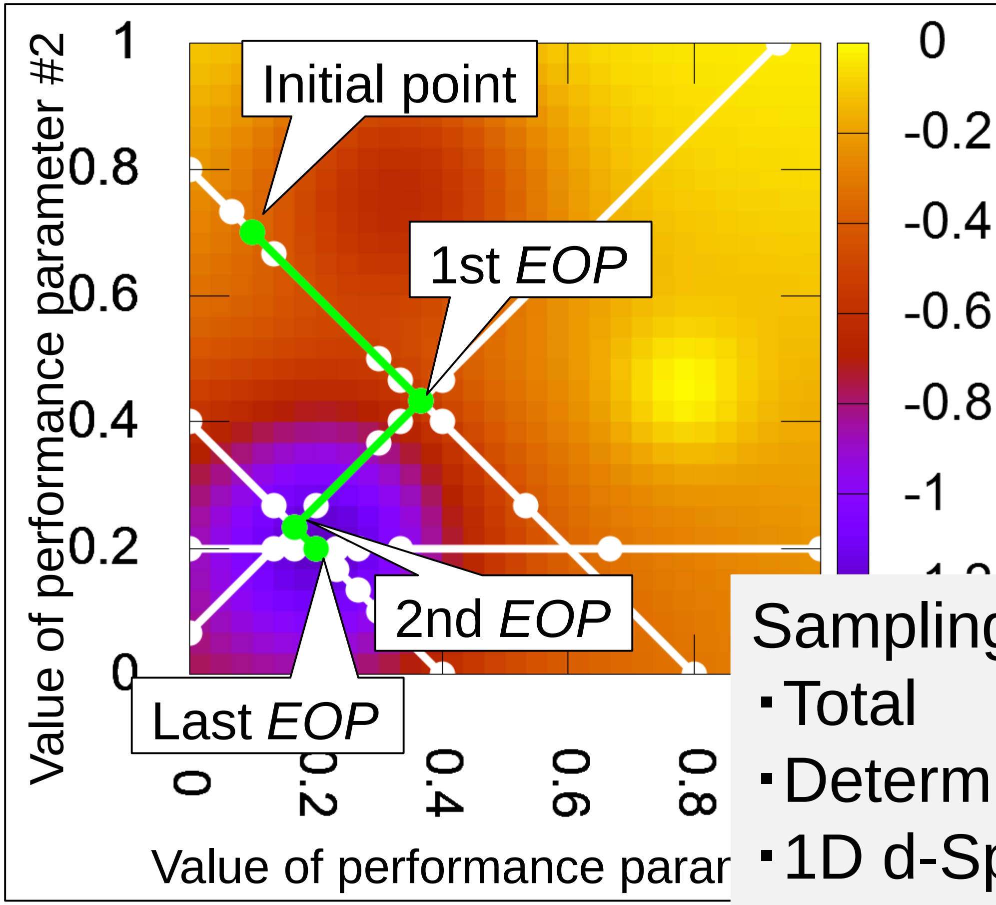
EOP : Estimated optimal point

Sampling points：8 + 7 + 6 + 6 + 6 + 5 + 6 + 6 = 50

1^{st} step	2^{nd} step	3^{rd} step	4^{th} step

Algorithm for repeating 1D d-Spline search

Contour map of 2-dimension（Test function：Franke）



: The paths of the best parameter

Search process

1. Decide the initial point
2. Determine the search direction that has a minimum value among the surrounding points

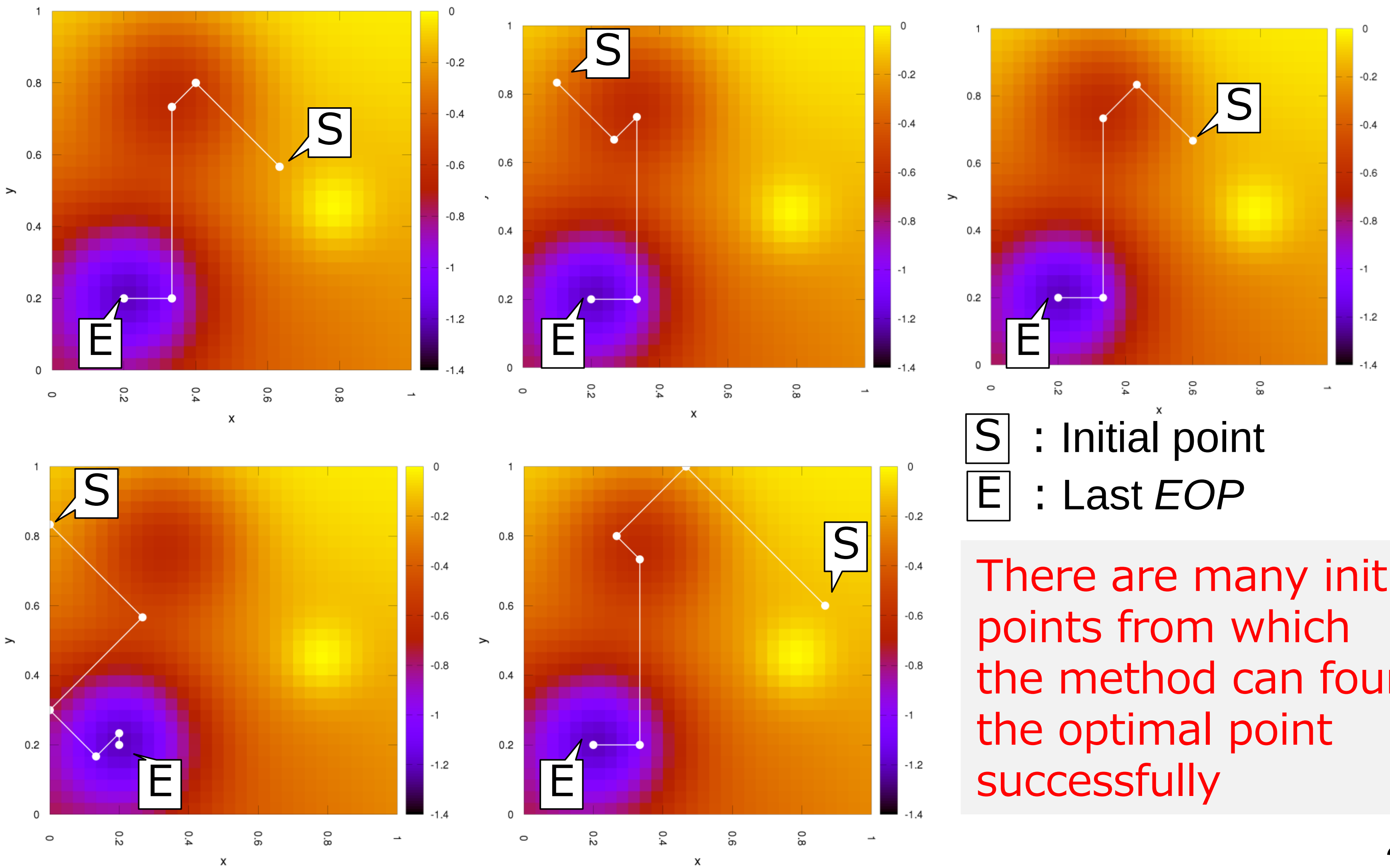
Sampling points

- Total : 50 points
- Determine direction : 26 points
- 1D d-Spline search : 24 points

Sampling points：8 + 7 + 6 + 6 + 6 + 5 + 6 + 6 = 50

Two performance parameter estimation case

Contour map (Test function : Franke)



S : Initial point

E : Last *EOP*

There are many initial points from which the method can found the optimal point successfully

Modified algorithm for repeating 1D d-Spline search

- The number of sampling points
  for determining the search direction
  depends on the dimension of performance parameter
  - 2 parameters : 8 （3×3 - 1）points.
  - 3 parameters : 26（3×3×3 - 1） points.
  - 4 parameters : 80（3×3×3×3 - 1）points.

⇒Modified algorithm aims to reduce computational cost
of determining the search direction

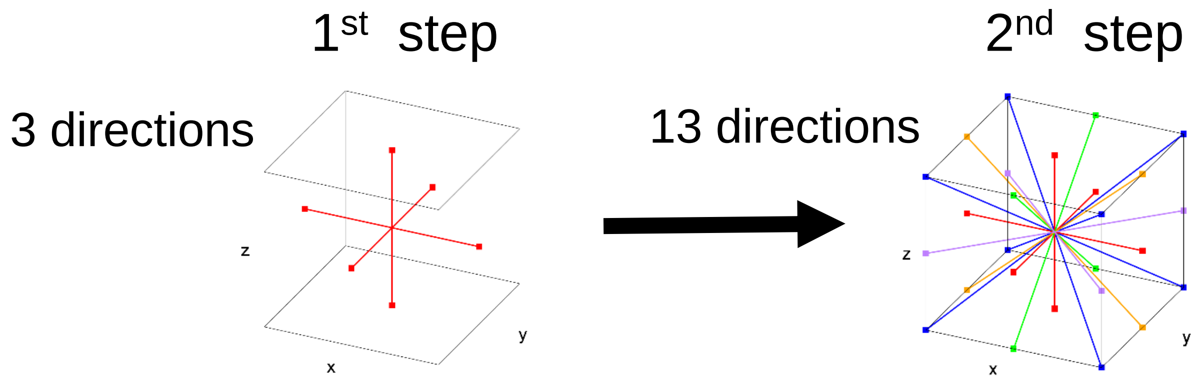
Modified algorithm for repeating 1D d-Spline search

Kogakuin University

- Modified algorithm construct the search process in two steps

- 1st step
  - Search only each axis direction (i.e. x, y or z…)
  - Determine the optimal point in restricted directions
- 2nd step
  - Expand search directions (i.e. 13 directions)
  - Search until convergence in global space

Case of
3 performance
parameters



Experimental environment
(3 performance parameters)

Kogakuin University

- Target program : AMG preconditioned BiCGSTAB method
  - A Poisson equation on a three-dimensional cubic domain
    - Mesh size : 120×120×120
- Tuning strategies are evaluated with various initial points
  - All initial points : $16 \times 10 \times 9 = 1440$ times

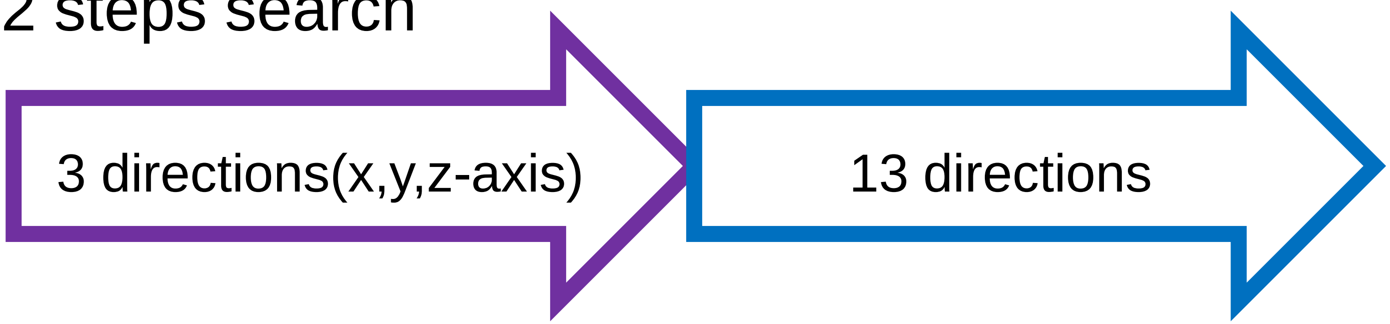
Performance parameter (*PP*)	θ	θ $reduction\ rate$	$dump\ jacobi\ coefficient$
Range	0.00 ~ 0.15	0.1 ~ 1.0	0.1 ~ 0.9
Interval	0.01	0.1	0.1
Number of *PP* values	16	10	9
Number of *PP* combination	1440（=16×10×9）		

Tuning strategies for experiments

- Repeating 1D d-Spline parameter search

  (1) 13 directions

  (2) 2 steps search



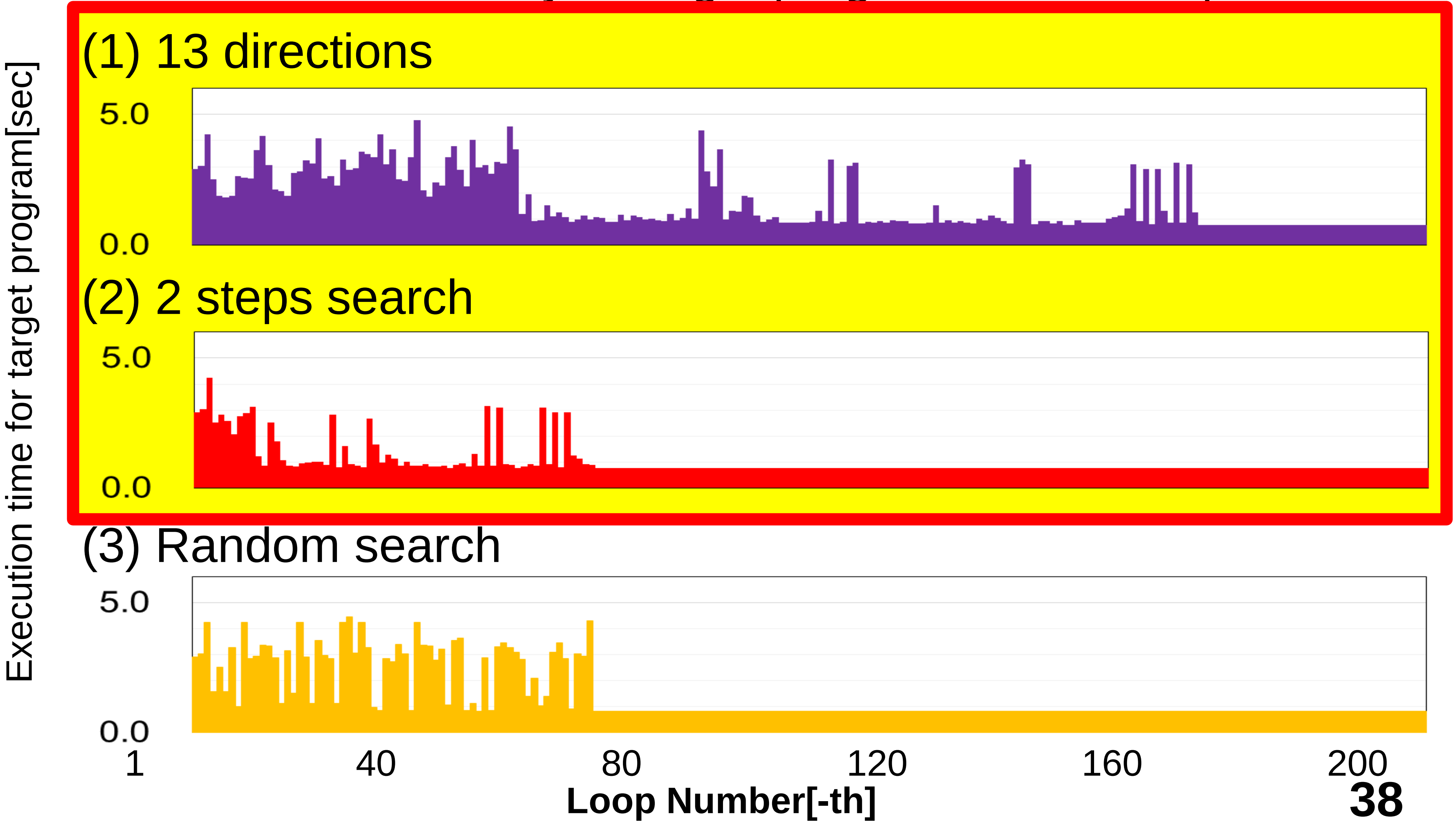
- As a comparison

  (3) Random search

$_3C_1 * 1 = 3$ directions
$_3C_2 * 2 = 6$ directions
$_3C_3 * 4 = 4$ directions
Total : 13 directions

Tuning result with 3 strategies

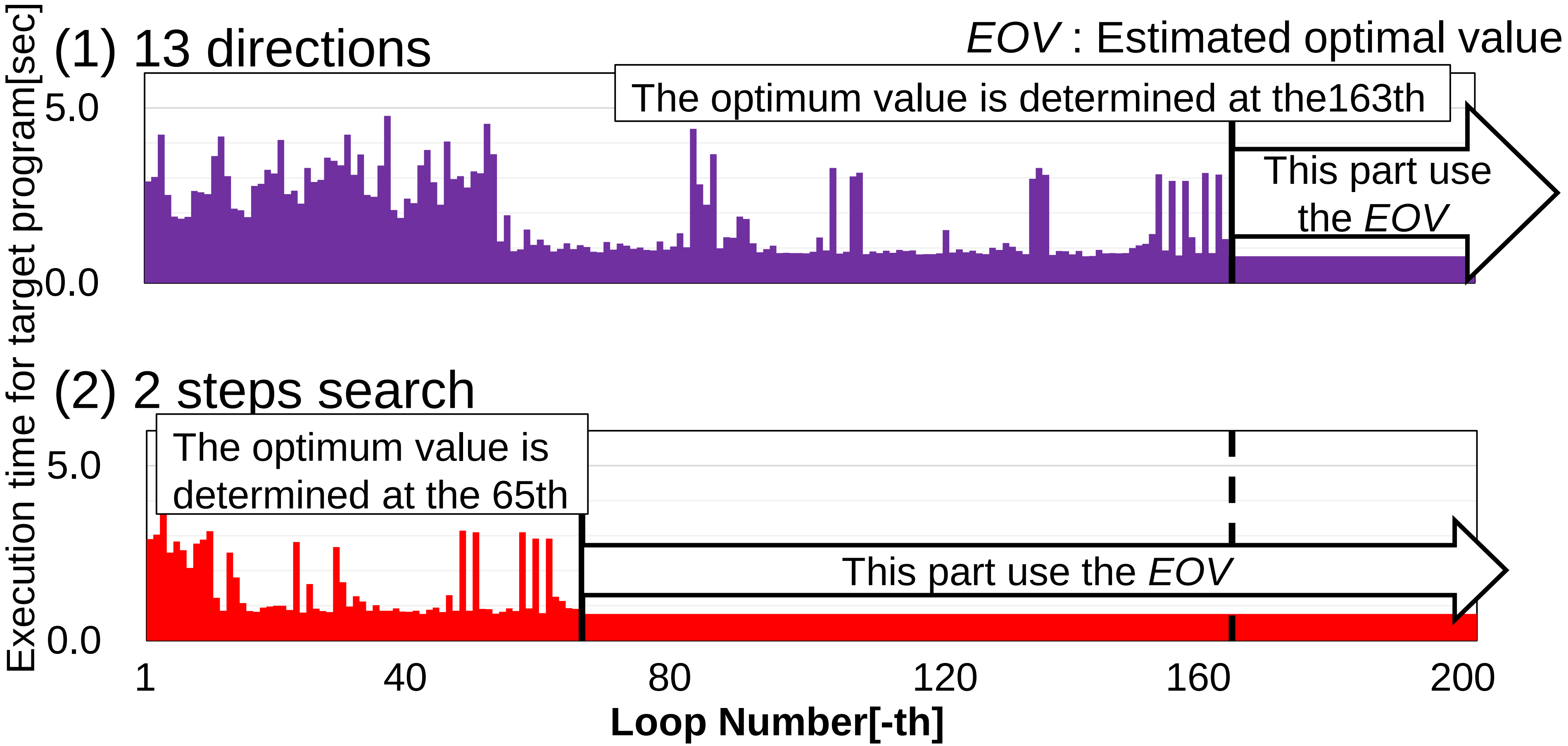
- Execution time history of target program for 200 loops



Tuning result with 3 strategies

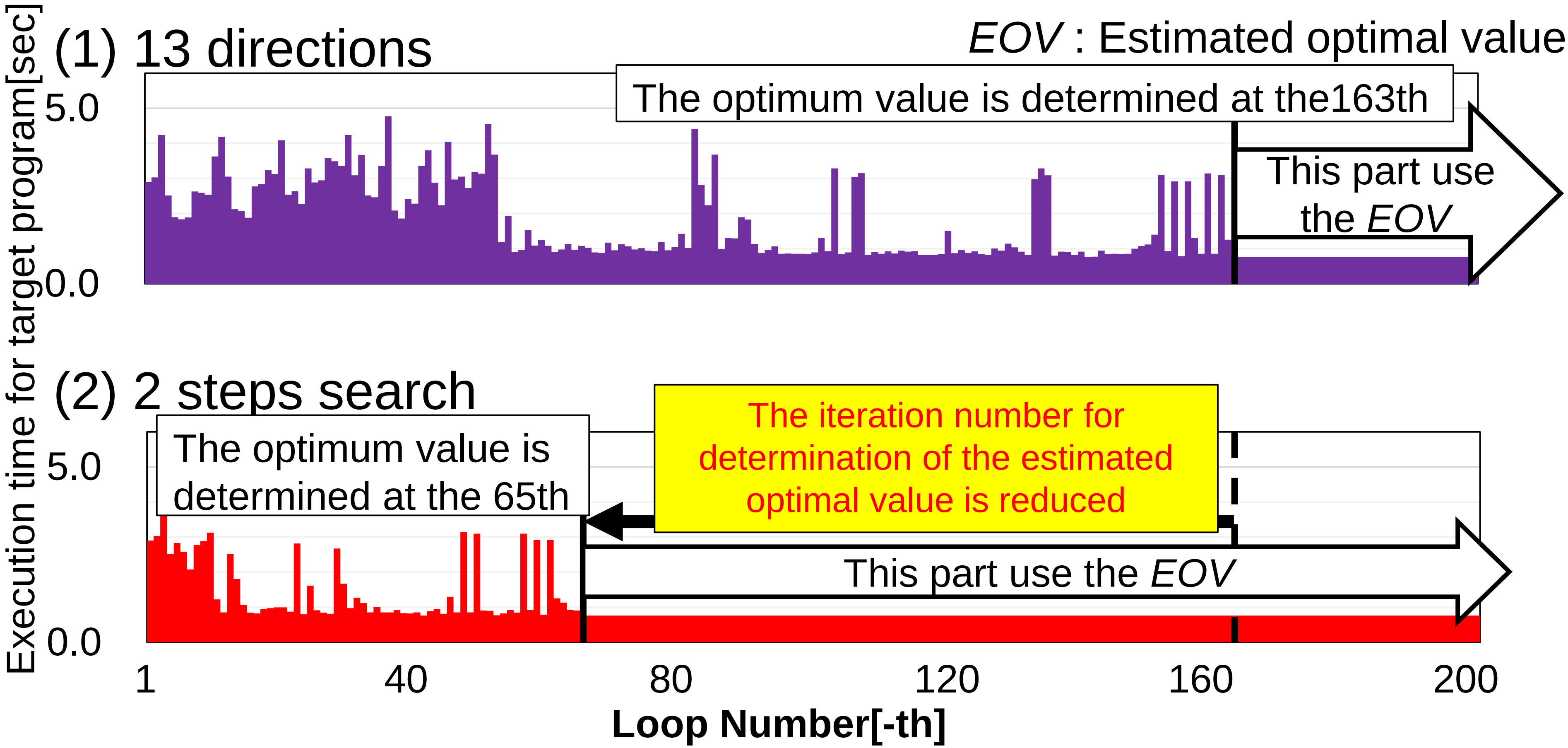
Kogakuin University

- Execution time history of target program for 200 loops



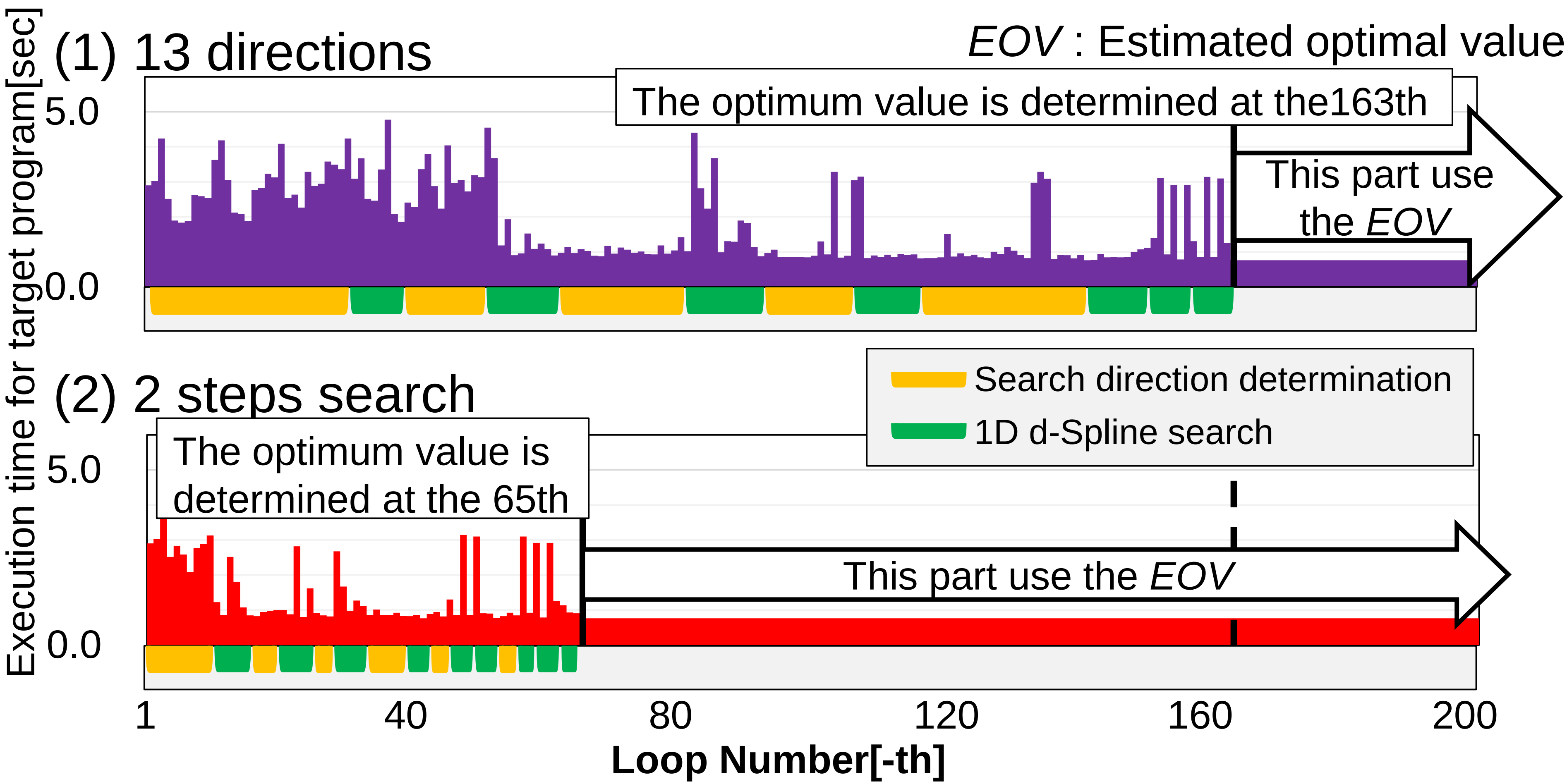
Tuning result with 3 strategies

- Execution time history of target program for 200 loops



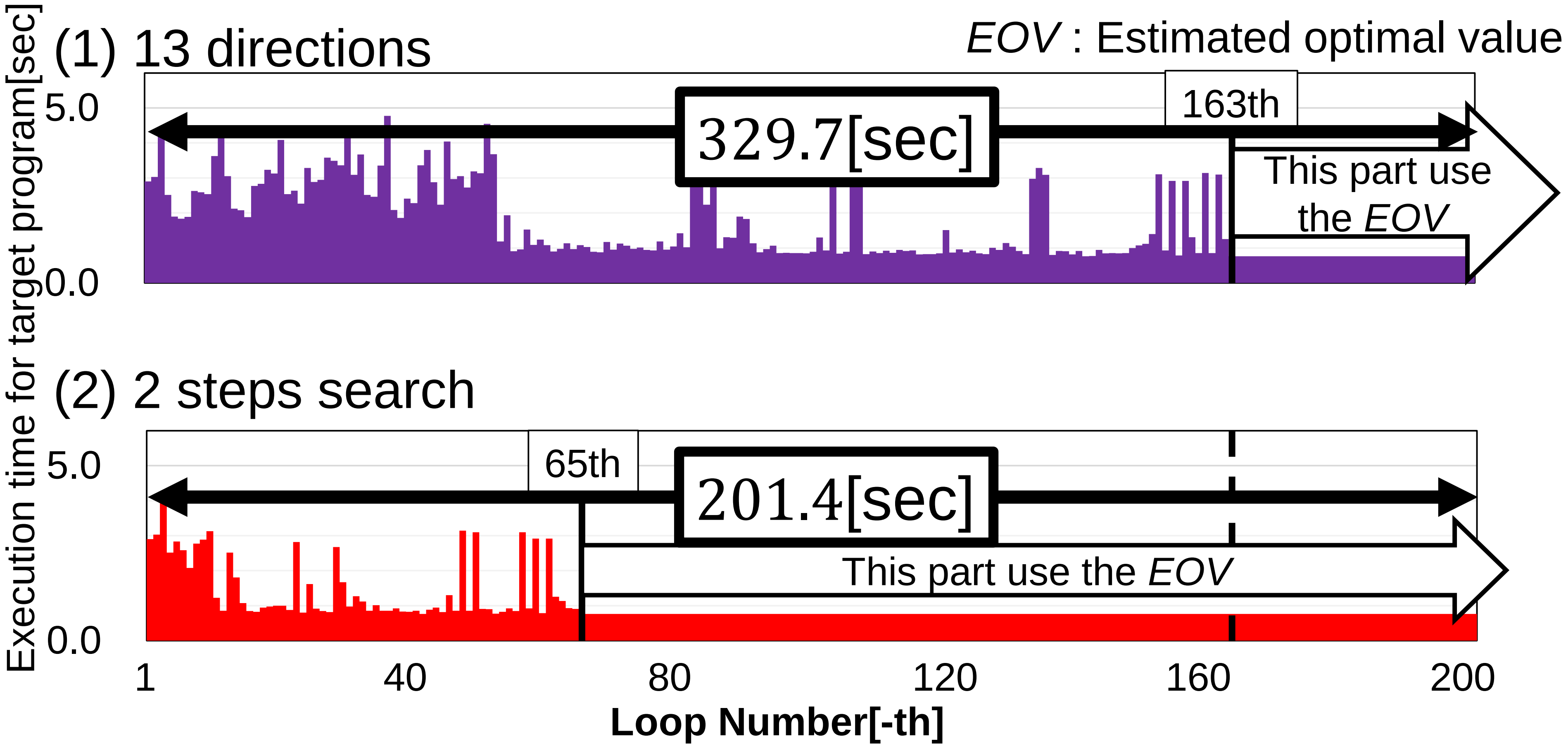
Tuning result with 3 strategies

- Execution time history of target program for 200 loops



Tuning result with 3 strategies

- Execution time history of target program for 200 loops

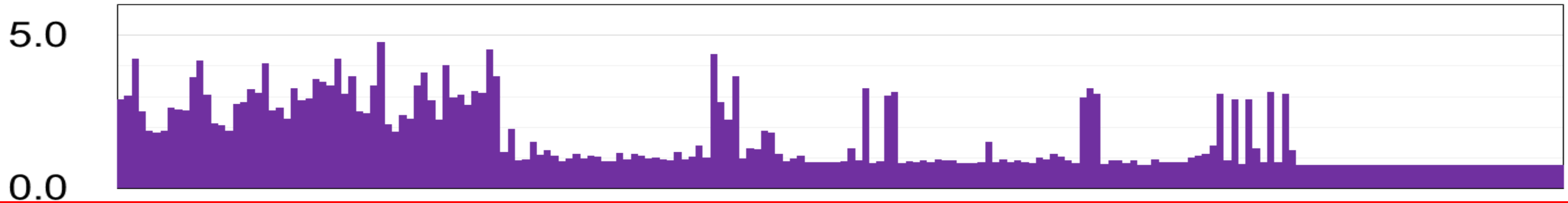


Tuning result with 3 strategies

Kogakuin University

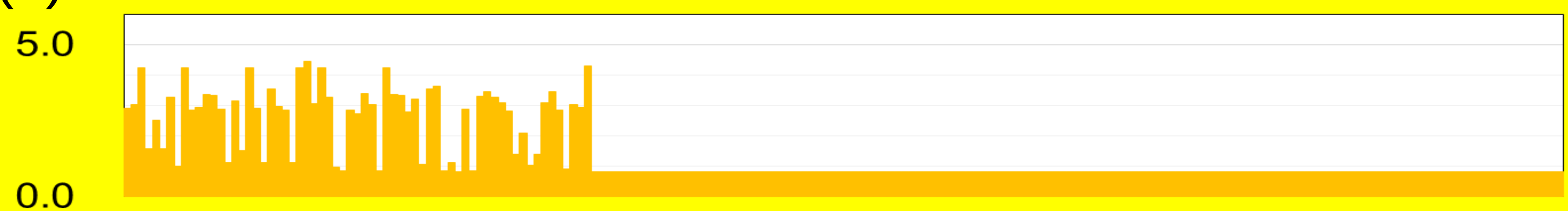
- Execution time history of target program for 200 loops

(1) 13 directions



(2) 2 steps search

(3) Random search

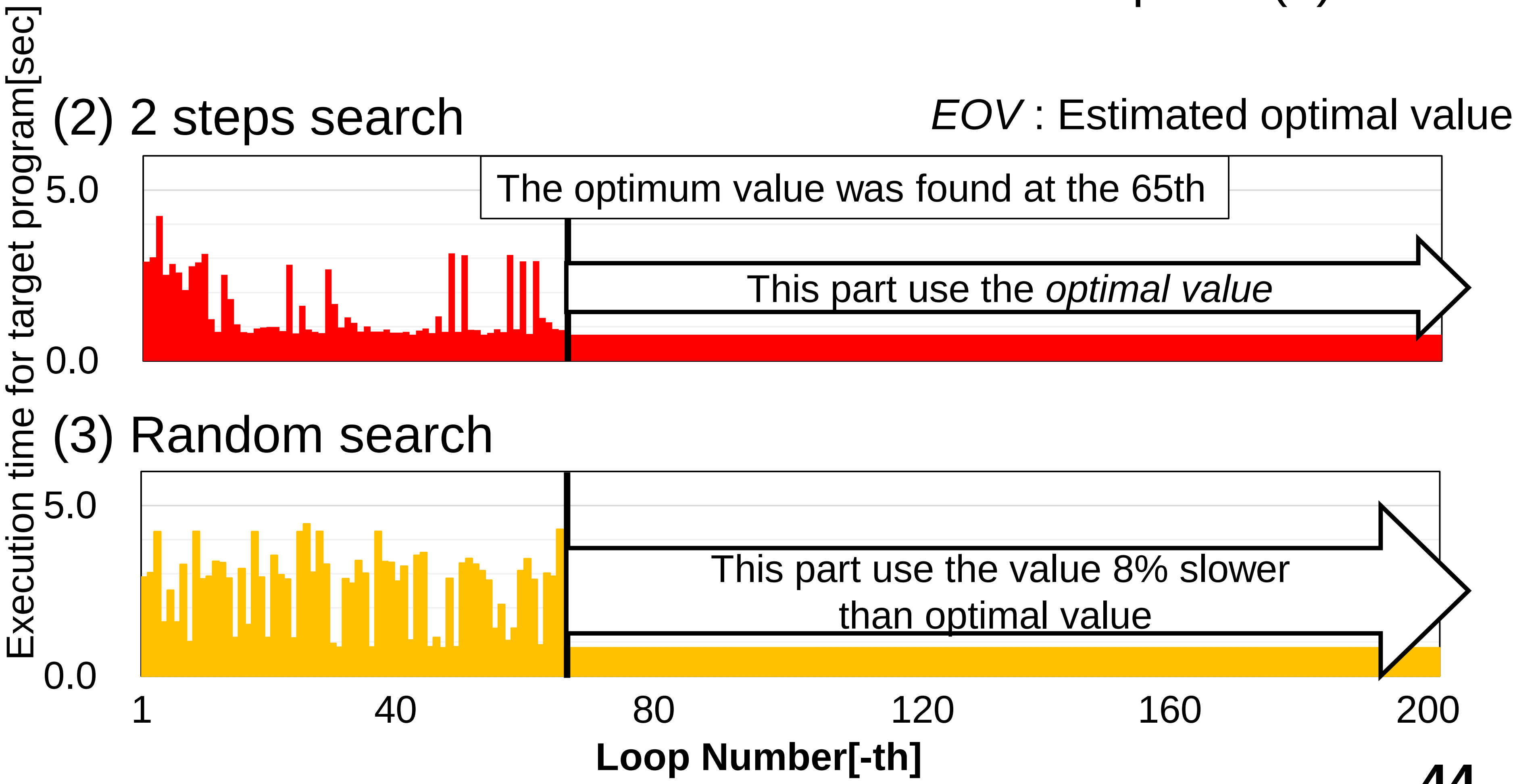


Execution time for target program[sec]

Loop Number[-th]

Tuning result with 3 strategies

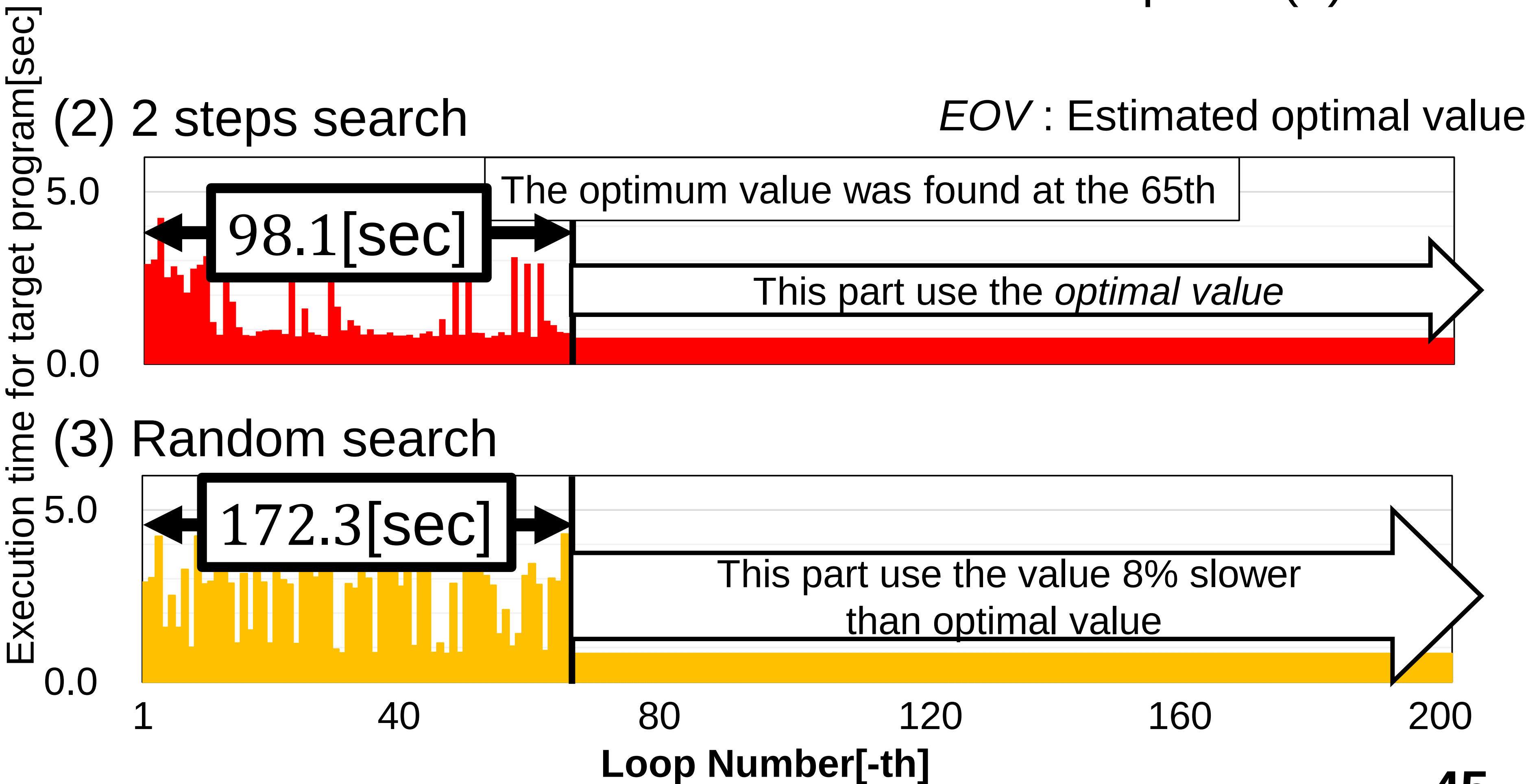
- For comparison of parameter search efficiency, random search was executed the same number of loops as (2)



Tuning result with 3 strategies

Kogakuin University

- For comparison of parameter search efficiency, random search was executed the same number of loops as (2)



Comparison of tuning strategies

- Initial performance parameter patterns : $16 \times 10 \times 9 = 1440$

Method		**Repeating 1D d-Spline search**	
		(1) 13 directions	(2) 2 steps search
Sampling points		126.1	104.8
	Direction determination	89.1	46.0
	IPPE/d-Spline	37.0	58.8
Relative error[%]		0.0003	0.0
Estimation time[msec] ※		0.59	0.71

※ The time does not include the execution time of the AMG method

Comparison of tuning strategies

- Initial performance parameter patterns : $16 \times 10 \times 9 = 1440$

Method		Repeating 1D d-Spline search (1) 13 directions	Repeating 1D d-Spline search (2) 2 steps search
Sampling points		126.1	104.8
	Direction determination	89.1	46.0
	IPPE/d-Spline	37.0	58.8
Relative error[%]		0.0003	0.0
Estimation time[msec] ※		0.59	0.71

※ The time does not include the execution time of the AMG method

Comparison of tuning strategies

- Initial performance parameter patterns : $16 \times 10 \times 9 = 1440$

Method		Repeating 1D d-Spline search	
		(1) 13 directions	(2) 2 steps search
Sampling points		126.1	104.8
	Direction determination	89.1	46.0
	IPPE/d-Spline	37.0	58.8
Relative error[%]		0.0003	0.0
Estimation time[msec] ※		0.59	0.71

※ The time does not include the execution time of the AMG method

Experimental environment
(4 performance parameters

- The target problem : AMG method
  - AMG method : One of solving simultaneous linear equations
- Tuning strategies are evaluated with various initial points
  - One strategy is executed with all initial points
    : $8 \times 8 \times 8 \times 8 = 4096$ times

Performance parameter (*PP*)	$dump\ jacobi\ coefficient\ Lv1$	$dump\ jacobi\ coefficient\ Lv2$	$Smoother\ accel\ coefficient\ Lv1$	$Smoother\ accel\ coefficient\ Lv2$
Range	0.1 ~ 0.8	0.1 ~ 0.8	0.8 ~ 1.5	0.8 ~ 1.5
Interval	0.1	0.1	0.1	0.1
Number of *PP* values	8	8	8	8
Number of *PP* combination	4096（＝8×8×8×8）			

4 performance parameters estimation

- Number of sampling points to determine the search direction

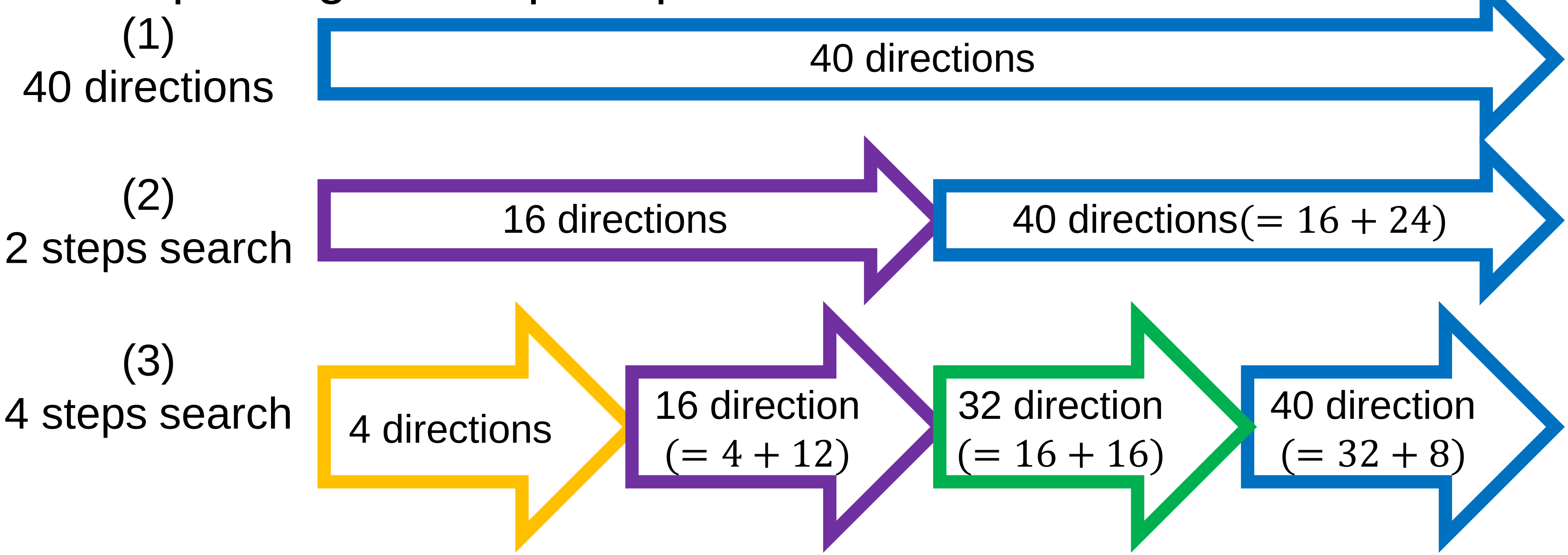
Process	Performance parameter(pp) space	
	Three dimension	Four dimension
Direction determination	$26\ (= 3^3 - 1)$	$\boxed{80\ (= 3^4 - 1)}$

- ${}_4C_1 * 1 = 4$ directions
- ${}_4C_2 * 2 = 12$ directions
- ${}_4C_3 * 4 = 16$ directions
- ${}_4C_4 * 8 = 8$ directions

Total : 40 directions

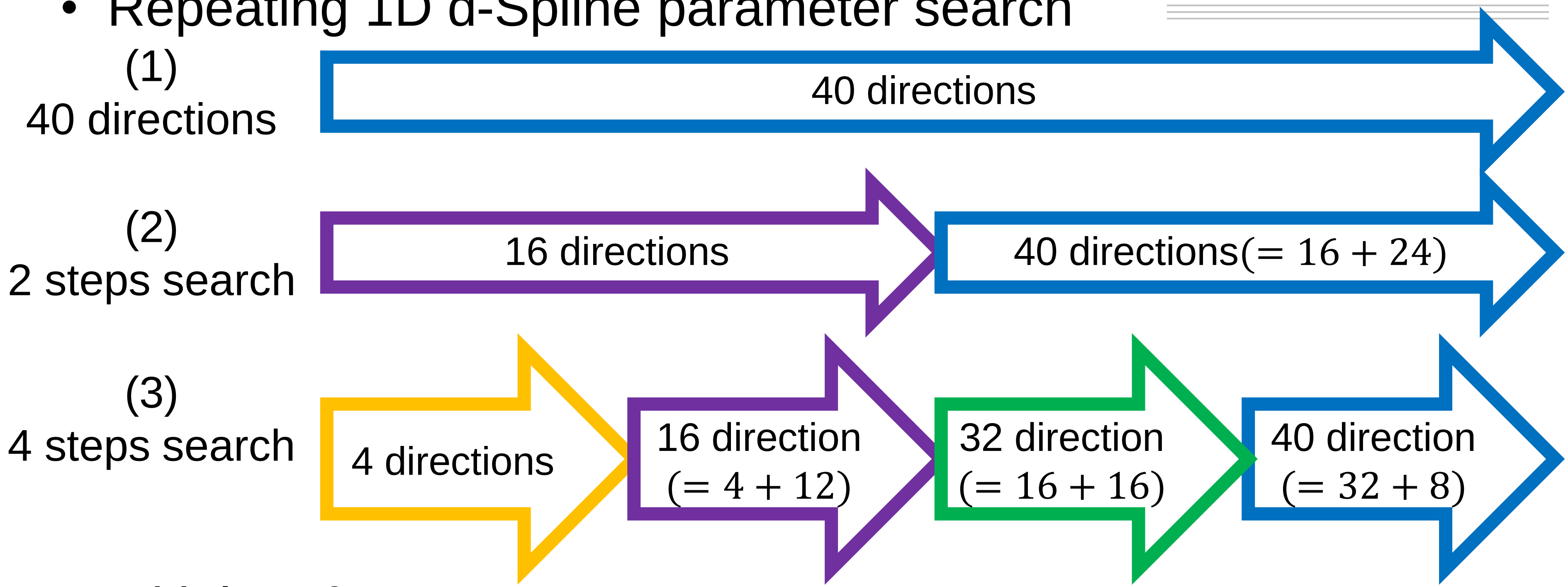
Comparison of tuning strategies

- Repeating 1D d-Spline parameter search



Comparison of tuning strategies

- Repeating 1D d-Spline parameter search



- Initial performance parameter patterns :$8 \times 8 \times 8 \times 8 = 4096$

Method	Repeating 1D d-Spline search		
	(1) 40 directions	(2) 2 steps search	(3) 4 steps search
Sampling points	209.1	143.0	120.4
Relative error[%]	0.97	0.87	0.12

Conclusion

- We proposed the parameter tuning strategy repeating 1D d-Spline parameter search
    - Computational cost of 1D d-Spline is very small
- Multiple steps parameter search strategy succeeded in reducing sampling points and maintaining the quality of estimated parameter value in multidimensional parameters
    - It searches parameter space in multiple steps
    - Adapted to 4 performance parameters case as well

Future works

- Implementation of proposed parameter search method in “ppOpen-AT”
  - Implement as a parameter search function

- Selection of initial point
  - Setting an appropriate initial point will reduces the iteration number of 1D d-Spline search and sampling points