Japan Architectural Review Open Access

Architectural Institute of Japan

Original Paper

Machine learning for combinatorial optimization of brace placement of steel frames

Takuya Tamura,[1,†] Makoto Ohsaki[1] and Jiro Takagi[2]

[1]Department of Architecture and Architectural Engineering, Kyoto University, Kyoto, Japan; [2]Department of Architecture and Building Engineering, Tokyo Metropolitan University, Hachioji, Japan

Correspondence
Makoto Ohsaki, Department of Architecture and Architectural Engineering, Kyoto University, Kyoto, Japan.
Email: ohsaki@archi.kyoto-u.ac.jp

[†]Present address: Shimizu Corporation, Tokyo, Japan

Funding information
Japan Society for the Promotion of Science, Grant Number: 16H03014, 16H04449

Part of this paper has been presented at OPTIS2016, Japan Soc. Mech. Eng., 2016; Annual Meeting of Architectural Inst. of Japan (AIJ), 2017; AIJ Kinki Chapter Research Meeting, 2017; 40th Symp. On Comp. Tech. of Information, Systems and Appl., AIJ, 2017.

Received May 2, 2018; Accepted June 28, 2018

doi: 10.1002/2475-8876.12059

Abstract

A method is presented for optimal placement of braces of plane frames using machine learning. The frame is subjected to static horizontal loads representing seismic loads. We consider the process of seismic retrofit by attaching braces. Therefore, the maximum value of additional stresses in the existing beams and columns and the maximum interstory drift angle are incorporated in the optimization problem. Characteristics of approximate optimal solutions and nonoptimal solutions are extracted using machine learning based on support vector machine and binary decision tree. Convolution and pooling are used for defining the features characterizing the solutions while reducing the number of variables. Optimization is carried out using a heuristic algorithm called simulated annealing based on local search. It is shown in the numerical examples that the computational cost is successfully reduced by avoiding costly structural analysis for a solution judged by machine learning as nonoptimal, and the important features in approximate optimal and nonoptimal solutions are identified.

Keywords
binary decision tree, braced frame, machine learning, optimization, simulated annealing, support vector machine

1. Introduction

Optimization of frame structures is a well-established field of research, and cross-sectional properties of small frames can be easily optimized using an appropriate method of mathematical programming or heuristic approach.[1,2] However, there still exist serious difficulties, if a real-world problem with many design variables and large number of degrees of freedom is to be solved, because substantial computational cost is needed for evaluation of functions defined by structural responses. Therefore, various approximation methods such as response surface approximation and kriging have been utilized.[3] A problem involving topology optimization is especially difficult to solve, because the problem becomes a combinatorial problem for which sensitivity information cannot be used.

Optimization methods are classified into mathematical programming and heuristic approach. The latter is further classified into population-based approach such as genetic algorithm (GA) and the methods based on local search including simulated annealing (SA)[4] and tabu search (TS).[5] In the method based on local search, a single solution is improved by successively selecting its best neighborhood solution. For a structural optimization problem, the properties of neighborhood solutions are evaluated by static and/or dynamic structural analysis, which demands large computational cost for a large-scale complex structure. Therefore, the computational cost may be substantially reduced if a solution that cannot be an approximate optimal solution or a feasible solution is excluded before carrying out analysis. For this purpose, machine learning can be effectively used.

Machine learning has been extensively applied in various fields of engineering including image processing,[6] automatic control, etc. Machine learning is regarded as a basic component of artificial intelligence and data mining.[7] Binary decision tree (BDT), support vector machine (SVM),[8] and association rule[9] are the basic tools of machine learning. However, only a few studies can be found for application of machine learning to structural optimization. In 1990s, artificial neural network (ANN) was extensively used for estimating the structural responses in the process of structural optimization.[10] Biedermann and Grierson[11] presented a method for training ANN for design knowledge such as member grouping in the process of optimization of plane frames. Lagaros and Papadrakakis[12] estimated nonlinear seismic responses of a 3D-frame using ANN. Ootao et al[13] used ANN for estimation of maximum stress ratio of a plate composed of functionally graded material. Adeli and Park[14] proposed a basic framework of structural optimization based on estimation of constraint functions by ANN. Kontovourkis et al[15] optimized the process of kinematic transformation using

ANN. However, most of the applications are for estimation of function values rather than classification of solutions that can be a candidate of approximate optimal solution. Furthermore, most of applications of ANN are in the second generation, when application to only small toy problems was possible.

Liu et al[16] proposed application of BDT to particle swarm optimization (PSO). Hanna[17] used SVM for estimating optimal geometry of modular structures for a given load set. SVM can also be used for constructing surrogate models.[18] Yang and Hsieh[19] used SVM for classification of feasible and infeasible solutions in the process of reliability optimization. Hagishita and Ohsaki[20] used association rule for topology optimization of framed structures. Szczepanik et al[21] applied constructive induction to wind design of tall steel building. Schwabacher et al[22] used inductive learning to find feasible initial design of a structural optimization problem. Reinforcement learning has been successfully applied for optimal control of robots.[23]

Topology optimization of braces under stress constraints is difficult to solve, because the stress constraint need not be satisfied if the brace does not exist[24]; ie, the stress constraint is design-dependent. It is possible to optimize brace location of highrise buildings using GA.[25] Hagishita and Ohsaki[20] presented an optimization method of braces with semirigid joints using scatter search. Continuum approaches have also been proposed for layout optimization of braces.[26,27] Zhu et al[28] optimized the brace locations for stochastic dynamic loads. It is important to consider the 3-dimensional properties of frame, if practical aspects are to be investigated. Park et al[29] optimized the locations of buckling restrained braces using multiobjective GA. However, to the authors' knowledge, there exists no research on application of machine learning to topology optimization of braces of a plane frame.

In this paper, we apply machine learning to reduce the computational cost for optimization of brace locations of building frames. The properties of approximate optimal solutions and nonoptimal solutions are identified by machine learning such as BDT and SVM. It is shown that the computational time is successfully reduced utilizing machine learning in the process of optimization using SA, and the properties of approximate optimal and nonoptimal solutions can be explicitly identified using machine learning. The primary goal of this research is to show the effectiveness of machine learning for complex topology optimization problems. Therefore, the properties of optimal solutions are not discussed in detail, and practical application of the optimization results is not the main subject of this study.

2. Outline of optimization problem and optimization method

We consider a steel building frame that is designed as an assembly of plane frames. Its seismic performances are upgraded using braces of various types including K-, V-, and diagonal braces as shown in Figure 1. Although a single type is usually used for each frame, it is possible to use different types simultaneously. In this paper, locations and combinations of braces are optimized for seismic retrofit; ie, the cross-sectional properties of beams and columns are fixed.

One of the important points in seismic retrofit is that increase of stresses in the existing beams and columns should be kept as small as possible,[30] so that fracture in existing members is to be prevented. Figure 2 illustrates distribution of axial forces in beams and columns of three types of 2-story 2-span frame subjected to horizontal load at the center of the roof, where the width of each member in the right-hand side figures is proportional to its absolute value of axial force. As seen from the figure, the maximum axial force depends on the types of braces. Therefore, it is important to optimize types and locations of braces considering additional stress to the existing beams and columns.

The design variables are the integer numbers representing types of braces including "no-brace" as illustrated in Figure 1. Let n_s and n_f denote the numbers of spans and stories (floors), respectively. Then, the number of design variables, which is equal to the number of rectangular spaces surrounded by beams and columns, is $m = n_s n_f$. The number of types of braces including no-brace is denoted by t. The values of integer variables $\boldsymbol{y} = (y_1,\ldots, y_m)$ are selected from the list $\{1,\ldots, t\}$. For the example in Figures 1 and 2, $m = 4$ and $t = 5$.

The objective function such as the total structural volume and a representative response value defined by stress or displacement is denoted by $F(\boldsymbol{y})$. Inequality constraints are given with upper bound $\bar{g}_i$ as $g_i(\boldsymbol{y}) \leq \bar{g}_i$ $(i = 1,\ldots, n_c)$, where n_c is the number of constraints. Then, the optimization problem is formulated as follows:

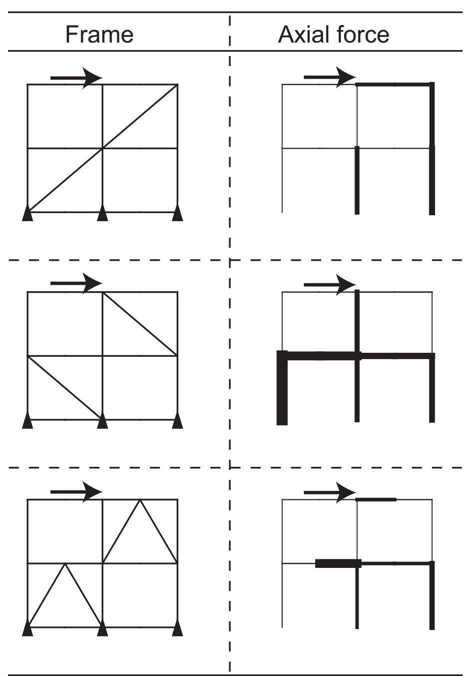

Figure 2. Additional axial forces of beams and columns under a horizontal load for a 2-story 2-span frame with three types of braces

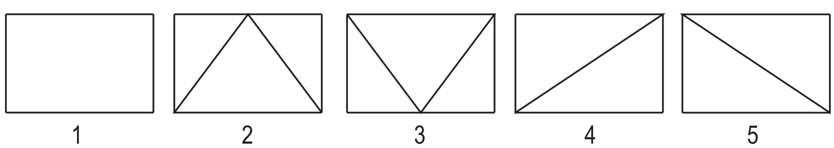

Figure 1. Types of braces including "no-brace"; 1: no-brace, 2: K-brace, 3: V-brace, 4: right diagonal brace, 5: left diagonal brace

$$\begin{aligned}&\text{Minimize } F(\boldsymbol{y})\\&\text{subject to } g_i(\boldsymbol{y}) \leq \bar{g}_i, \quad (i = 1, \ldots, n_c)\end{aligned} \tag{1}$$

Since problem (1) is a combinatorial problem, a gradient-based method is not applicable. Furthermore, the number of function evaluations during optimization should be restricted to a small value, because generally substantial computational cost is required for evaluation of the objective and/or constraint functions defined by structural responses such as stress and displacement. Therefore, population-based methods such as GA and PSO are not suitable, and we use a heuristic method called SA that is an extension of local search.[4] It simulates annealing process of metal. In the early stage of process, movement to a worse neighborhood solution is allowed to search a wide space in the feasible region to find the global optimal solution. A parameter representing the temperature in the annealing process is gradually reduced to restrict acceptance of a worse solution to converge to a local optimal solution. Since the values 1,…, t of each variable in our problem are not related to the order of mechanical property, it is not appropriate to change the values of all variables randomly in the process of generating the neighborhood solutions. Therefore, we select n_v variables to be modified.

The algorithm of SA for this study is summarized as follows:

Step 1 Randomly assign initial value $\boldsymbol{y}^0$ of $\boldsymbol{y}$, which satisfies all constraints, and let $F(\boldsymbol{y}^0)$. Assign initial value T_0 for temperature parameter T. Set the iteration counter $k = 0$, and assign the scaling parameter s, the temperature reduction parameter α that is slightly less than 1, the total number of steps n_t, the number of neighborhood solutions n_b, and the number of variables n_v to be modified at each step to generate neighborhood solutions.

Step 2 Randomly modify n_v variables from the current value $\boldsymbol{y}^k$ to generate neighborhood solutions $\boldsymbol{y}_i^*$ $(i = 1,\ldots, n_b)$. Obtain structural responses of each neighborhood solution and compute the objective function value $F(\boldsymbol{y}_i^*)$. Assign a large value to $F(\boldsymbol{y}_i^*)$ if the constraints are not satisfied. Find the best feasible solution $\hat{\boldsymbol{y}}$ that minimizes $F(\boldsymbol{y}_i^*)$.

Step 3 Accept $\hat{\boldsymbol{y}}$ as $\boldsymbol{y}^{k+1} = \hat{\boldsymbol{y}}$ if $F(\hat{\boldsymbol{y}}) < F(\boldsymbol{y}^k)$; otherwise, accept $\hat{\boldsymbol{y}}$ with the following probability p:

$$p = \exp\left(\frac{F(\boldsymbol{y}^k) - F(\hat{\boldsymbol{y}})}{Ts}\right) \tag{2}$$

Step 4 Update the temperature parameter as $T \leftarrow \alpha T$.

Step 5 Terminate the process and output the best solution if k reaches the specified value n_t; otherwise, increase k as $k \leftarrow k + 1$ and go to Step 2.

3. Estimation of property of approximate optimal and nonoptimal solutions

In the search process by SA, some of the randomly generated neighborhood solutions obviously cannot be optimal. Therefore, computational cost will be drastically reduced by avoiding analysis for such solutions. For this purpose, we use tools of machine learning, namely, BDT and SVM. BDT expresses the hierarchical process of decision using binary tree. The data are classified through the decision associated with each node along the path from the root to a leaf node. SVM separates the data using a linear dividing hyperplane. The minimum value of margin from the dividing hyperplane to the data is maximized by solving an optimization problem. Complex data can be classified applying nonlinear transformation using kernels such as Gaussian kernel before solving an optimization problem.

Table 1. Combinations of estimated and true values of labels

		True label	
		1	−1
Estimated label	1	TP (true-positive)	FP (false-positive)
	−1	FN (false-negative)	TN (true-negative)

Learning data are generated by random sampling of feasible solutions, and approximate optimal solutions and nonoptimal solutions are labeled as 1 and −1, respectively. The total set of data is divided into N groups with equal size to carry out cross-validation. The $N - 1$ groups are used for learning, and the remaining one group for validation. Although the labels have integer values, a real number called score, which has a value between −1 and 1, is used to incorporated probabilistic estimation of characteristics of solutions. After learning is completed, the scores of validation data are evaluated and the labels are predicted. Possible combinations of the predicted and true labels are classified as shown in Table 1. In the table, FP indicates that the estimated positive value 1 is false, ie, the true value is negative. It is important to reduce the possibility of FN, so that we do not miss the approximate optimal solutions. Cross-validation error is also used for evaluating the efficiency of learning.

Preprocessing is very important to improve the accuracy of learning. The following procedures are used here:

- *Dummy binary variable*: Since SVM and BDT are effective for ordered variables, they cannot be directly used for a function defined by categorical variables. Therefore, the categorical variables defining the types of braces are converted to dummy 0-1 variables. For example, if a categorical variable

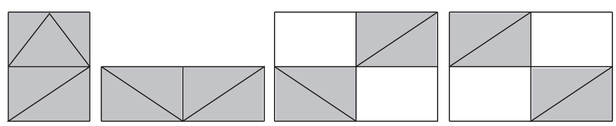

Figure 3. Examples of convolution filters defined by relative locations of a pair of braces; each filter has 16 patterns

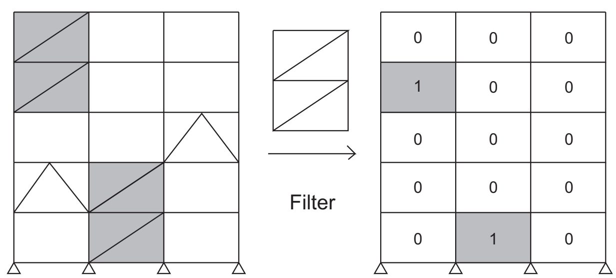

Figure 4. Illustration of application process of convolution filter

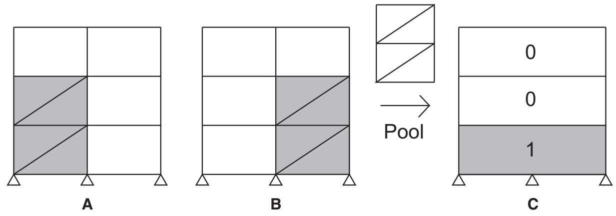

Figure 5. Illustration of application process of pooling

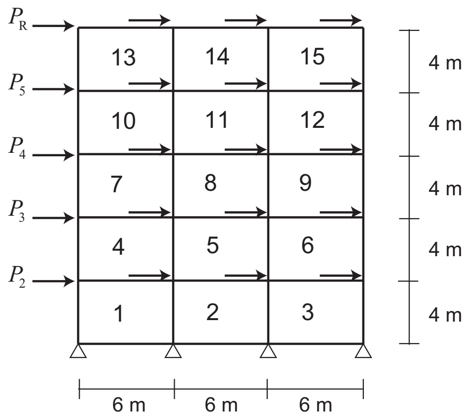

Figure 6. A 5-story 3-span frame

has the value "4" among the set {1, 2, 3, 4, 5} in Figure 1, then the corresponding binary variables are (0, 0, 0, 1, 0).

- *Convolution*: The property of braced frame is not defined only from the type of brace at each location. The relative locations of pairs of braces are also important for smooth transmission of the forces to the supports. To extract relative locations, we use the technique called convolution, which is often used in the field of image processing for feature extraction through filters. In our problem, the pairs of braces located in vertical, horizontal, and two diagonal directions, as shown in Figure 3, are used as filters as illustrated in Figure 4. The value 1 is given if the pair of braces matches the filter, otherwise, the value is 0. If there are four types of braces as shown in Figure 1, then each filter has 16 (=4 × 4) patterns; accordingly, the filters in four directions in Figure 3 have 64 (=16 × 4) patterns in total. Figure 4 illustrates application process of convolution filter for a 5-story 3-span frame. In this case, there are 15 (=3 × 5) locations for each filter; therefore, the total number of filter variables becomes 960 (=64 × 15).
- *Pooling*: Although convolution enhances capability of extracting properties of solutions, it increases the total number of variables in the learning process as described above. Therefore, we reduce the number of variables using pooling. For example, the two solutions in Figure 5A and B have a similar property, because both patterns are effective for reducing the interstory drift angles of the 1st and 2nd stories. Therefore, they are converted to Figure 5C using the procedure called pooling. This way, the redundancy of solutions is reduced.

The definitions of approximate optimal solutions and nonoptimal solutions are also important to enhance the performance of machine learning. The following two patterns are investigated in the numerical examples:

- Pattern 1: Best 10% are approximate optimal solutions, and the remaining solutions are nonoptimal.
- Pattern 2: Best 10% are approximate optimal solutions, and the worst 10% are nonoptimal.

Note that approximate optimal solutions and nonoptimal solutions are distinctly separated if Pattern 2 is used.

We carry out machine learning before optimization, and the Step 2 of algorithm of SA is modified as follows:

Step 2 Randomly modify n_v variables from the current value $\boldsymbol{y}^k$ to generate neighborhood solutions $\boldsymbol{y}_i^*$ $(i = 1,\ldots, n_b)$. *Predict the characteristics of each solution based on the result of machine learning. Carry out analysis if it is judged as an approximate optimal solution; otherwise, assign a large value to* $F(\boldsymbol{y}_i^*)$ *without carrying out analysis.* Also assign a large value to $F(\boldsymbol{y}_i^*)$ if the constraints are not satisfied. Find the best feasible solution $\hat{\boldsymbol{y}}$ that minimizes $F(\boldsymbol{y}_i^*)$.

4. Optimization problem of braced frame

We consider a 5-story 3-span frame as shown in Figure 6. The numbers in the figures indicate the variable numbers.

A frame analysis software package OpenSees[31] is used for evaluating static responses of frames. The properties of model and analysis are summarized as follows:

1 The frame has stiff base beams, and is pin-supported at the column base.

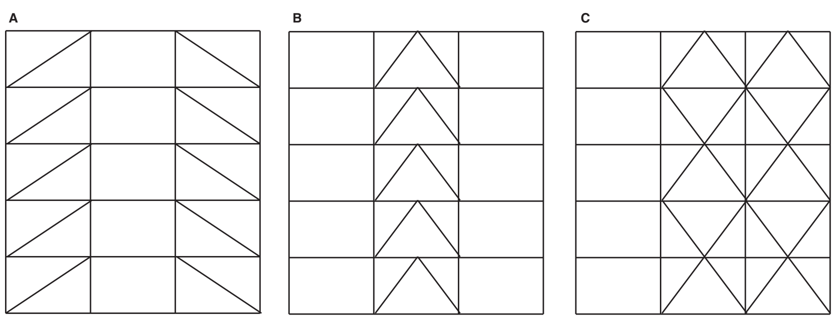

Figure 7. Typical patterns of brace locations; (A) pattern A, (B) pattern B, (C) pattern C

Table 2. Maximum values of stresses and interstory drift angles of patterns A, B, and C

	σ^{max}(N/mm^2)	Member	θ^{max}	V (m^3)
No brace	649.54	2B-ex	0.0220	0
Optimal	84.83	1B-ex	0.0019	0.35
Pattern A	169.32	1C-ex	0.0028	0.27
Pattern B	221.82	1C-in	0.0034	0.18
Pattern C	118.57	1C-in	0.0016	0.37

2 The columns, beams, and braces are modeled using elastic beam-column element, and the braces are rigidly connected to the beams and columns.
3 Buckling and yielding of beams, columns, and braces are not considered.
4 Increase of edge stresses considering axial force and bending moment of beams and columns due to horizontal loads is incorporated in the optimization problem to simulate the process of seismic retrofit; therefore, the stress and displacement under self-weight are not computed.
5 The axial stiffness of beam is multiplied by 10 to evaluate in-plane stiffness of slab without using the conventional assumption of neglecting the in-plane deformation of slab. Accordingly, horizontal loads are applied at beam-column joints as shown in Figure 6, and deformation and stress in beams can be computed.

The maximum value among the absolute values of edge stresses at two ends of all beams and columns is denoted by $\sigma^{max}(\boldsymbol{y})$, which is a function of $\boldsymbol{y}$ defining the locations and types of braces, and is computed by adding the stresses due to axial force and bending moment. In the following, the maximum absolute value is called maximum value for brevity. The maximum value among interstory drift angles of all stories is denoted by $\theta^{max}(\boldsymbol{y})$. Let $\bar{\theta}$ denote the upper bound for $\theta^{max}(\boldsymbol{y})$. The following optimization problem is to be solved:

$$\begin{aligned} &\text{Minimize}\ \sigma^{max}(\boldsymbol{y}) \\ &\text{subject to}\ \theta^{max}(\boldsymbol{y}) \le \bar{\theta} \\ &\qquad J_i(\boldsymbol{y}) \le \bar{J},\ (j = 1, \ldots, n_f) \end{aligned} \tag{3}$$

where $J_i(\boldsymbol{y})$ is the number of braces in the ith story, and $\bar{J}$ is its upper bound.

We consider a case where the constraint on the interstory drift angle is violated if no brace is assigned. Then, the constraints are to be satisfied by assigning braces; it leads to increase in stresses in beams and columns. Therefore, optimal locations of braces exist to satisfy the constraints while keeping the stresses small. For example, for the three typical patterns A, B, and C in Figure 7A-C, respectively, the maximum values of stress and interstory drift angle are computed as listed in Table 2. "Member" indicates the location of member that has the maximum stress, where the first integer is the story (floor) number, {C, B} indicates {column, beam}, and {ex, in} indicates {exterior, interior}. "Optimal" is the result to be obtained in next section. The total volume $V(\boldsymbol{y})$ of braces is also listed in Table 2 for comparison purpose. It is seen from Table 2 that the typical patterns will lead to large stresses in beams and columns, and optimization of brace locations efficiently reduces those stresses.

5. Optimization results without machine learning

Optimization is carried out for the 5-story frame in Figure 6. The best solution after 20 trials with different random seeds is regarded as the optimal solution. The upper bounds for interstory drift angle and the number of braces in each story are given as $\bar{\theta} = 0.005$ and $\bar{J} = 2$, respectively. Sufficiently large value is given for the bending stiffness of base beam, so that the frame is almost rigidly supported at the base. Note that the horizontal loads are applied only in one direction, because we consider linear elastic behavior of beams, columns, and braces. Consequently, the optimal locations of braces are asymmetric; however, the maximum stress and drift angle are the same for the loads in the opposite direction. The scaling parameter s for SA is given, so that the acceptance probability is 0.5 for a neighborhood solution that increases the objective value 10% at the initial solution with T = 1.0.

Table 4. Optimization results of 5-story frame

	σ^{max} (N/mm^2)	Member	Pattern	θ^{max}	V (m^3)
No brace	649.54	2B-ex	111111111111111	0.0220	0
Optimal	84.83	1B-ex	122133221431125	0.0019	0.35

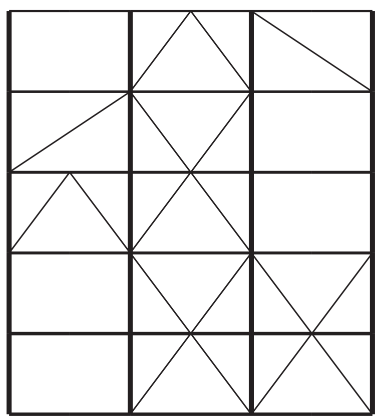

Figure 8. Locations of braces of the optimal solution

Table 3. Cross-sectional area and second moment of area of beams, columns, and braces of 5-story frame

Floor	Beam	A (cm^2)	I (cm^4)	Story	Column	A (cm^2)	I (cm^4)
R	H-346 × 174 × 6×9	52.45	11 000	1-5	HSS-350 × 350 × 9	122.76	23 800
5	H-350 × 175 × 7×11	62.91	13 500				
4	H-396 × 199 × 7×11	71.41	19 800				
3	H-396 × 199 × 7×11	71.41	19 800				
2	H-400 × 200 × 8×13	83.37	23 500	Brace	H-250 × 125 × 6×9	36.97	3960

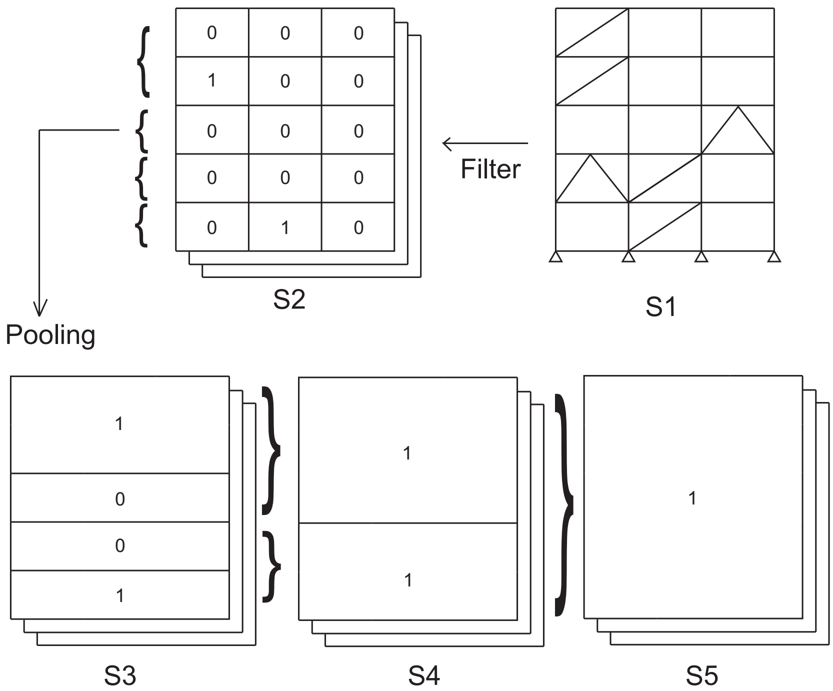

Figure 9. Five strategies of convolution and pooling

Table 5. Learning results of SVM for 5-story frame

Classification pattern	Strategy	Filter 4			Filter 5		
		Error	FN	FP	Error	FN	FP
Pattern 1	S1	0.1000	1000/1000	0/9000	0.1000	1000/1000	0/9000
	S2	0.0836	675/1000	159/9000	0.0770	586/1000	157/9000
	S3	0.0945	914/1000	32/9000	0.0823	611/1000	157/9000
	S4	0.1000	1000/1000	0/9000	0.1000	1000/1000	0/9000
	S5	0.1000	1000/1000	0/9000	0.1000	1000/1000	0/9000
Pattern 2	S1	0.0250	17/1000	35/1000	0.0250	17/1000	35/1000
	S2	0.0410	41/1000	28/1000	0.0110	10/1000	10/1000
	S3	0.0560	64/1000	28/1000	0.0120	15/1000	18/1000
	S4	0.1495	133/1000	144/1000	0.0385	26/1000	57/1000
	S5	0.2050	194/1000	221/1000	0.0965	107/1000	106/1000

The parameters of SA are specified as $n_v = 5$, $n_b = 15$, $n_t = 1000$, and $\alpha = 0.99$ after several parametric studies. The loads P_2, P_3, P_4, P_5, P_R are calculated as 97, 139, 185, 237, 465 (kN), which are the sum of four loads in each floor. Section properties of beams, columns, and braces are listed in Table 3.

The responses of the frame without brace and the optimization results are shown in Table 4. The optimal solution is shown in Figure 8, which does not have much regularity; however, most of the braces are connected with each other to allow continuous load path to the support.

6. Machine learning for properties of approximate optimal and nonoptimal solutions

The BDT and SVM, which are available in Statistics and Machine Learning Toolbox in Matlab Ver. 2016,[32] are used for learning the properties of approximate optimal solutions and nonoptimal solutions. The solutions are classified by Patterns 1 and 2 described in Section 3. The number N of groups for cross validation is 10 for both BDT and SVM. The maximum number of splits is 100 in BDT. For SVM, linear kernel with standardization and autoscale are used, and the parameter for box constraint is 3.

Depending on the use of convolution and pooling, one of the following strategies S1, … S5, which are illustrated in Figure 9, is used for extracting the patterns of braces of the 5-story frame:

S1: Do not use convolution.

S2: Use convolution, but do not use pooling.

S3: Use convolution. Classify the five stories to four groups consisting of 1st, 2nd, 3rd, and upper {4th, 5th} stories, respectively, and carry out pooling for each group.

Table 6. Learning results of BDT for 5-story frame

Classification pattern	Strategy	Filter 4			Filter 5		
		Error	FN	FP	Error	FN	FP
Pattern 1	S1	0.0972	720/1000	268/9000	0.0972	720/1000	268/9000
	S2	0.1034	840/1000	177/9000	0.0935	586/1000	375/9000
	S3	0.1086	854/1000	170/9000	0.0997	719/1000	231/9000
	S4	0.1127	891/1000	202/9000	0.1123	840/1000	291/9000
	S5	0.1114	947/1000	198/9000	0.1155	945/1000	150/9000
Pattern 2	S1	0.0390	38/1000	40/1000	0.0390	38/1000	40/1000
	S2	0.0725	65/1000	63/1000	0.0290	33/1000	21/1000
	S3	0.0920	76/1000	73/1000	0.0340	24/1000	24/1000
	S4	0.2309	317/1000	107/1000	0.0930	83/1000	95/1000
	S5	0.3413	407/1000	263/1000	0.2194	170/1000	250/1000

S4: Use convolution. Classify the five stories to two groups consisting of lower {1st, 2nd} and upper {3rd, 4th, 5th} stories, respectively, and carry out pooling for each group.

S5: Use convolution, and carry out pooling for the whole stories as single group.

The filters as shown in Figure 3, for example, are used for convolution for extracting the properties in vertical, horizontal, and two diagonal directions. We consider the following two combinations of brace types in each filter:

Filter 4: Do not include the case without brace. There are $4 \times 4 = 16$ types for each of four relative locations in Figure 3; the total number of filters is $4 \times 4 \times 4 = 64$.

Filter 5: Include the case without brace. There are $5 \times 5 = 25$ types for each of four relative locations in Figure 3; the total number of filters is $5 \times 5 \times 4 = 100$.

Note that Filter 5 is expected to have better performance, because it includes Filter 4. However, Filter 5 demands more computational time than Filter 4.

The learning results of the 5-story frame are listed in Tables 5 and 6, respectively, for SVM and BDT. In the tables, “Error” is the cross-validation error, and FN and TN are the ratios of false-negative and true-negative to the total number of verification data. Note that the total number of “negative” (nonoptimal) solutions is 9000 and 1000 for Patterns 1 and 2, respectively, although the total number of “positive” (approximate optimal) solutions is 1000 for both patterns. Therefore, FN is the ratio of

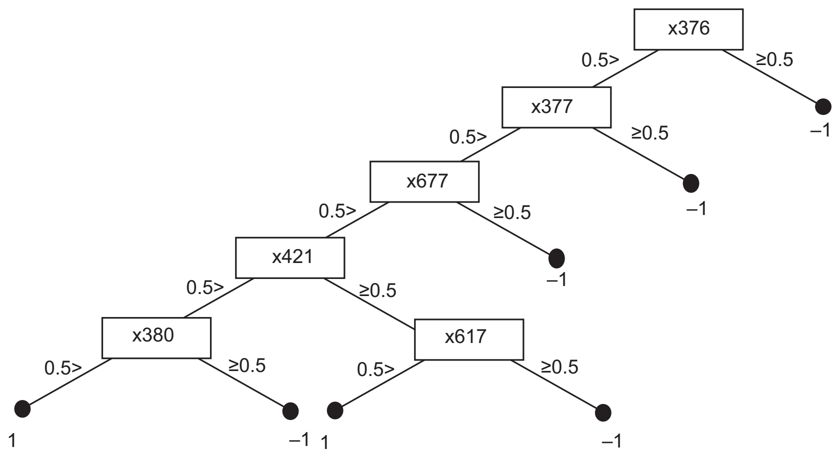

Figure 10. Feature tree for 5-story frame

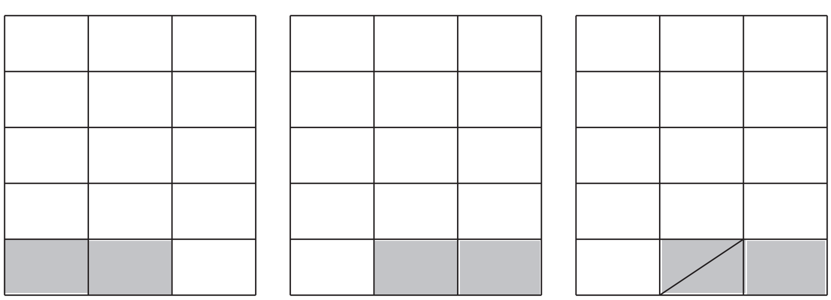

Figure 11. Filter variables indicating branch patterns near the root of BDT

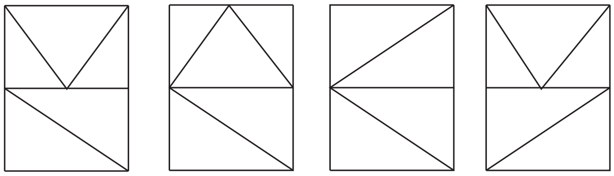

Figure 12. Frequently observed patterns in nonoptimal solutions

number of solutions that are judged as "negative" among 1000 positive solutions, and FN is preferred to be small, so that we do not miss the approximate optimal solutions.

The following properties are observed from the tables:

1 The value of FN is large for both of SVM and BDT if Pattern 1 is used, because the approximate optimal and nonoptimal solutions are more distinctly classified by Pattern 2 than Pattern 1.
2 Performance generally improves if Filter 5 is used; however, the performance is still not good for Pattern 1. The use of Filter 5 and Pattern 2 leads to the best performance, which means that "no brace" is needed in the definition of filters.
3 Use of pooling has good accuracy, if lower three stories are distinguished. Obviously, strategy S5 has the largest error and FN value among the five strategies for all cases, because the structural responses depend on the locations of braces.

7. Property of approximate optimal and nonoptimal solutions

Properties of approximate optimal solutions and nonoptimal solutions are investigated using the results of machine learning, where Filter 5 and strategy S2 without pooling are used to clearly extract the properties. In this case, we have $5 \times 5 \times 4 \times 15 = 1500$ filter variables which are denoted by $x_1, \ldots, x_{1500}$.

First, we investigate the results of BDT. The feature tree is shown in Figure 10. As seen from the figure, the features of nonoptimal solutions, rather than approximate optimal solutions, are extracted in the first four branches in Figure 10. This fact indicates that the nonoptimal solutions have more distinct features than the approximate optimal solutions.

Figure 11 illustrates the definitions of filters corresponding to variables x_{376}, x_{377}, and x_{677}. For example, the variable x_{367} is equal to 1 if there is no brace at locations 1 and 2 defined in Figure 6, while $x_{376} = 1$ is satisfied if there is no brace at locations 1 and 2. The first and second branches in Figure 10 indicate that the frame is labeled as -1 if $x_{376} \geq 0.5$ or $x_{377} \geq 0.5$, which means that the frame without any brace in the 1st story cannot be an approximate optimal solution. Frequently observed patterns in nonoptimal and approximate optimal solutions, respectively, in the randomly generated 10 000 solutions are shown in Figures 12 and 13. An example of approximate optimal solution and its distribution of axial forces in beams, columns, and braces are shown in Figure 14. Although the stress due to bending moment is included in evaluation of maximum stress for the optimization problem, the load path of frame can be clearly seen by the distribution of axial forces. It can be seen from Figure 14 that the external loads are transmitted mainly through the V-braces and K-braces forming an X-brace in the two vertically connected stories.

Next, we investigate the features of approximate optimal and nonoptimal solutions using the learning results of SVM. After learning, the score $S(\boldsymbol{x})$ of solution $\boldsymbol{x}$ is evaluated using the following equation:

$$S(\boldsymbol{x}) = \frac{1}{a}\boldsymbol{\beta} \cdot \boldsymbol{x} + \boldsymbol{b} \tag{4}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ is the coefficient vector of m variables, a is the scaling factor, and $\boldsymbol{b}$ is the bias vector. It is seen from Equation (4) that a large value of β_i leads to a large score when $x_i = 1$, while a small (negative) value of β_i leads a small (negative) score when $x_i = 1$. Therefore, the features of approximate optimal and nonoptimal solutions can be estimated by the values of β_i. The filter variables with ten largest and smallest values of β_i are listed in Tables 7 and 8, respectively, where the corresponding patterns are shown in Figures 15 and 16. Note that all filter variables in Figure 10 except x_{677} are included in Table 8.

8. Optimization using machine learning

Machine learning is incorporated to SA algorithm to reduce the total computation time. Classification of the neighborhood solution is based on the score that has real numbers. The scores for 100% confident classification for approximate

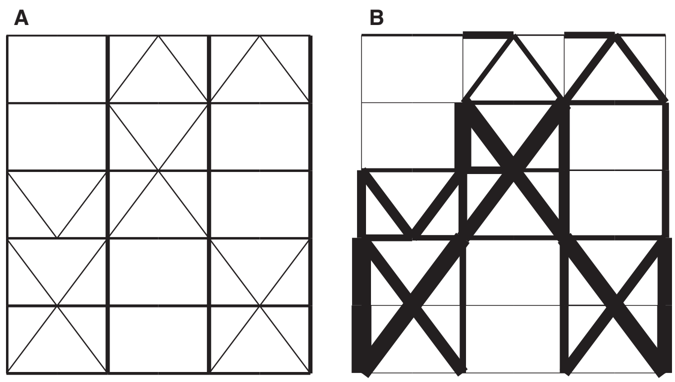

Figure 14. An example of approximate optimal solution; (A) brace locations, (B) axial forces of beams, columns, and braces

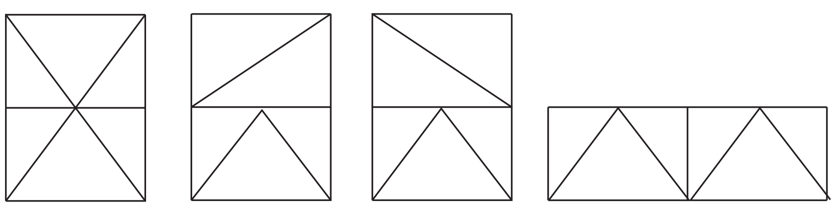

Figure 13. Frequently observed patterns in approximate optimal solutions

Table 7. Filter variables with ten largest values of β_i

Order	1	2	3	4	5	6	7	8	9	10
Variable number	692	106	496	856	93	108	617	91	451	1352
β_i	3.81	3.56	3.29	3.28	3.14	3.10	3.09	3.08	3.00	2.80

Table 8. Filter variables with ten smallest values of β_i

Order	1500	1499	1498	1497	1496	1495	1494	1493	1492	1491
Variable number	376	377	380	379	382	383	4	751	796	1337
β_i	−7.78	−7.21	−4.63	−4.02	−3.97	−3.63	−3.24	−3.13	−2.82	−2.77

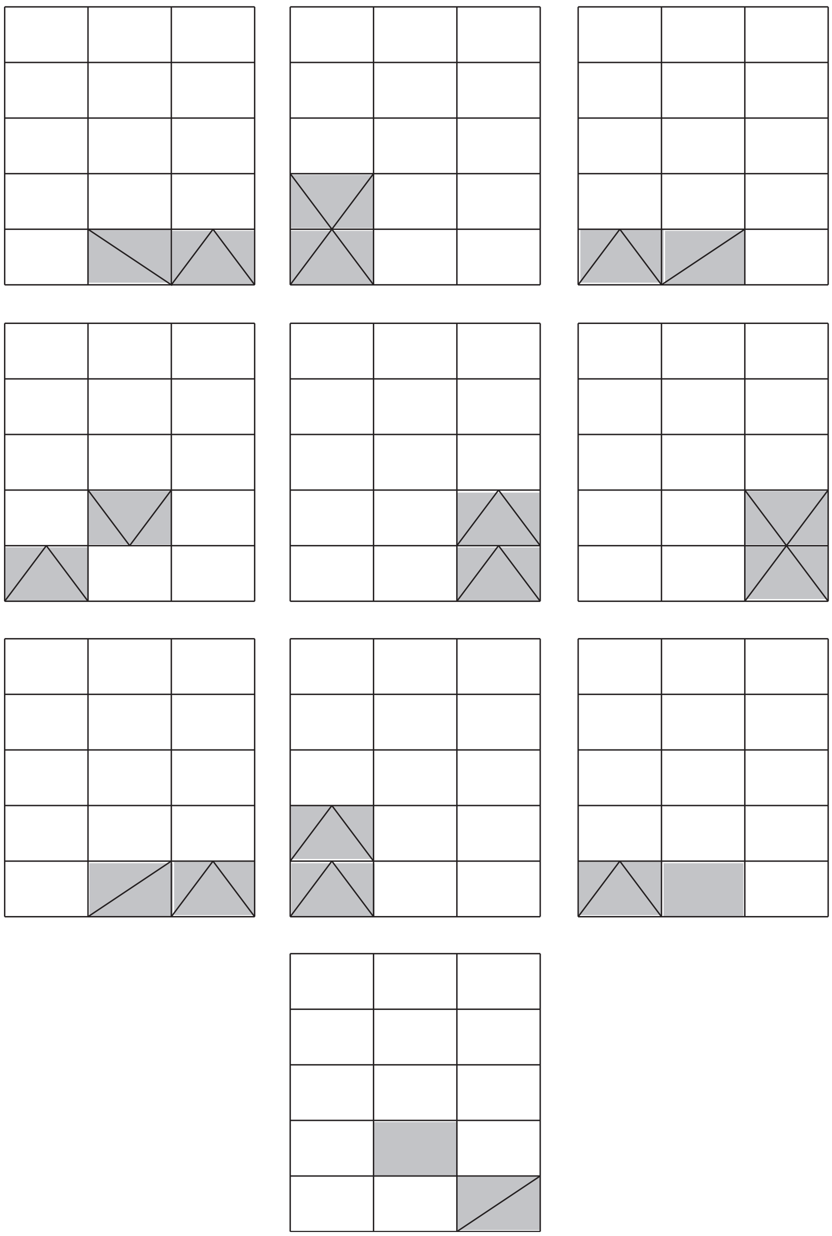

Figure 15. Filter variables with large value of β_i indicating features of approximate optimal solutions

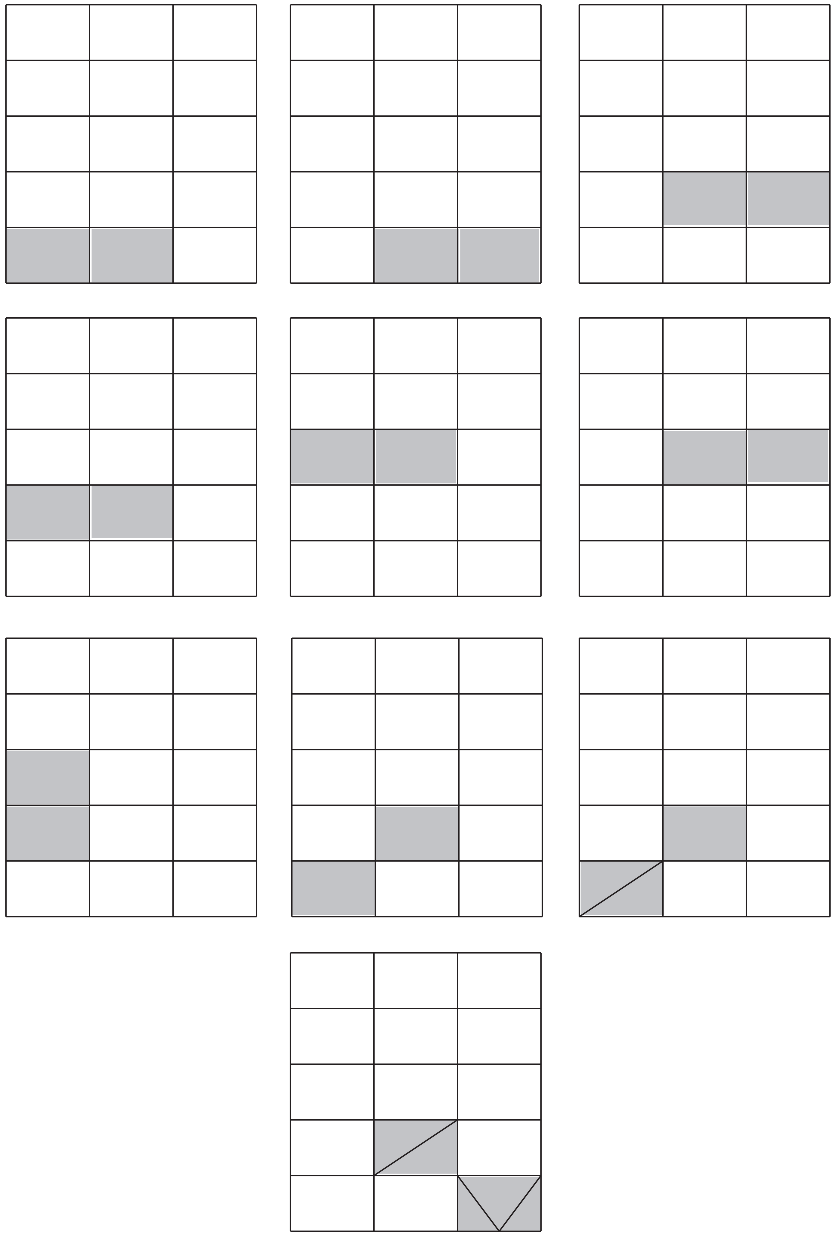

Figure 16. Filter variables with small value of β_i indicating features of nonoptimal solution

optimal and nonoptimal solutions are 1 and −1, respectively, while the threshold values are given dynamically, as follows, during the optimization process:

1 Assign the initial value 0.0 for the threshold value τ.
2 Add $\Delta\tau_1$ to τ at the beginning of each step of SA.
3 If the score obtained by prediction is greater than τ, then assume the solution as approximate optimal and carry out structural analysis.
4 Reduce τ by $\Delta\tau_2$, if the objective value $\sigma^{\max}(\boldsymbol{y})$ obtained by analysis is less than the specified value $\bar{\sigma}$.

In the preliminary investigation, we found that the computational time for prediction by SVM is very large, if we have large number of filter variables. Therefore, the strategy S3 with pooling is used for SVM. By contrast, the strategy S2 without pooling is used for BDT to maintain accuracy of prediction. The values of $\Delta\tau_1$ and $\Delta\tau_2$ in the following examples are 0.03 and 0.03, respectively, for BDT, and 1.00 and 0.50 for SVM. The upper-bound stress is $\bar{\sigma}$ = 120 N/mm^2, which is specified in view of distribution of $\sigma^{\max}(\boldsymbol{y})$ values in the 10 000 data. Computation is carried out using a PC with Intel Core i5-5200U 2.20 GHz and 8 GB memory.

Table 9 shows the sum of computation time and number of analyses, respectively, for 20 trials as well as the optimal objective value for each case. Note that "computation time" means the elapsed time (wall-clock time) as an output of Matlab. As seen from the table, the reduced numbers of

Table 9. Computation time, number of analyses, and objective function value

	Without machine learning	SVM	BDT
Learning			
Time for analysis	–	2093 seconds	2093 seconds
Time for learning	–	12.4 seconds	6.1 seconds
Optimization			
Time for prediction	–	483.7 seconds	360.5 seconds
Time for analysis	14 314.3 seconds	7961.6 seconds	7162.6 seconds
Time for optimization including learning	14 314.3 seconds	10 550.7 seconds	9622.2 seconds
Number of analyses	67 368	35 710	27 819
Optimal objective value	84.83 N/mm^2	87.08 N/mm^2	87.51 N/mm^2

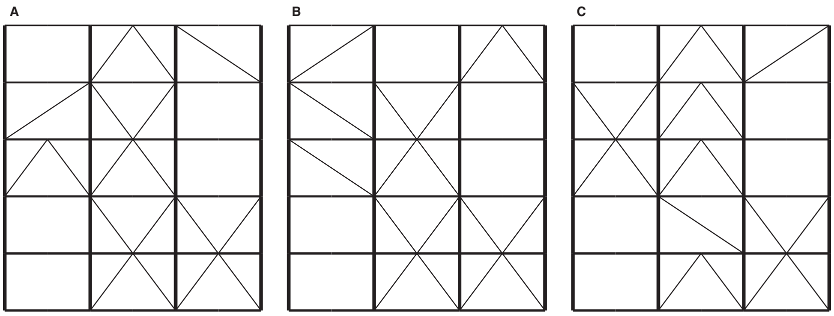

Figure 17. Locations of braces of the optimal solutions of 5-story 3-span frames; (A) without machine learning, (B) SVM, (C) BDT

analyses are 31 658 and 39 549, respectively, for SVM and BDT. The reduced CPU time is about 3800 for SVM, and about 4700 seconds. for BDT. This way, computation time has been successfully reduced using machine learning. The optimization results are shown in Figure 17, which confirms that the solutions by SVM and without learning are very similar.

9. Conclusions

A method based on SA has been proposed for optimization of brace locations of building frames utilizing machine learning for detecting nonoptimal solutions during optimization. The objective function is the maximum additional stress considering axial force and bending moment of beams and columns under horizontal static loads representing seismic loads. It has been shown that the computational cost can be successfully reduced using BDT or SVM.

Various strategies of modeling features of brace locations and preprocessing of variables have been investigated in view of accuracy and computational cost of learning. It has been shown that distinct classification of approximate optimal and nonoptimal solutions is effective to improve the accuracy of learning and prediction. Convolution using filters with "no brace" generally improves the accuracy of prediction; however, it increases the computational cost. Therefore, pooling in each of lower stories is effective to reduce the number of variables, while maintaining the accuracy.

It has been shown in the numerical examples of 5-story 3-span frames that the computational cost for optimization by SA can be drastically reduced using SVM or BDT. In the process of SA using the learning results, it is effective to vary dynamically the threshold value of the score for detecting a non-optimal solution. The properties of approximate optimal and nonoptimal solutions can be extracted from the feature trees as an output of BDT and the coefficients of the function for estimating the score of SVM. For example, the braces should be continuously located to reduce the additional stresses in beams and columns due to horizontal seismic loads.

Note that the primary goal of this research is to show the effectiveness of machine learning for complex topology optimization problems. Therefore, the properties of optimal solutions are not discussed in detail, and practical application of the optimization results may be a subject of future study.

Acknowledgment

This study is partially supported by JSPS KAKENHI No. 16H04449 and 16H03014.

Disclosure

The authors have no conflict of interest to declare.

References

1 Saka MP, Geem ZW. Mathematical and metaheuristic applications in design optimization of steel frame structures: an extensive review. *Math Problems Eng*. 2013;**2013**, Article ID 201031:33.
2 Ohsaki M. *Optimization of Finite Dimensional Structures*. Hoboken, NJ: CRC Press; 2010.
3 Forrester AIJ, Keane AJ. Recent advances in surrogate-based optimization. *Prog Aerospace Sci*. 2009;**45**:50-79.
4 van Laarhoven PJ, Aarts EH. *Simulated Annealing: Theory and Applications*. Dordrecht: Springer; 1987.
5 Glover FW, Laguna M. *Tabu Search*. Boston, MA: Springer; 1997.
6 Bishop CM. *Pattern Recognition and Machine Learning*. New York, NY: Springer; 2006.
7 Witten IH, Frank E, Hall MA. *Data Mining: Practical Machine Learning Tools and Techniques*. Amsterdam: Elsevier; 2011.
8 Cristianini N, Shawe-Taylor J. *An Introduction to Support Vector Machines*. Cambridge: Cambridge University Press; 2000.

9 Hagishita T, Ohsaki M. Optimal placement of braces for steel frames with semi-rigid joints by scatter search. *Comp Struct*. 2008;**86**:1983-1993.
10 Reich Y, Barai SV. Evaluating machine learning models for engineering problems. *Artif Intell Eng*. 1999;**13**:257-272.
11 Biedermann JD, Grierson DE. Training and using neural networks to represent heuristic design knowledge. *Adv Eng Softw*. 1996;**27**:117-128.
12 Lagaros ND, Papadrakakis M. Neural network based prediction schemes of the non-linear seismic response of 3D buildings. *Adv Eng Softw*. 2012;**44**:92-115.
13 Ootao Y, Kawamura R, Tanigawa Y, Nakamura T. Neural network optimization of material composition of a functionally graded material plate at arbitrary temperature range and temperature rise. *Arch Appl Mech*. 1998;**68**: 662-676.
14 Adeli H, Park HS. A neural dynamics model for structural optimization: theory. *Comp Struct*. 1995;**57**(3):383-390.
15 Kontovourkis O, Phocas MC, Lamprou I. Adaptive kinetic structural behavior through machine learning: optimizing the process of kinematic transformation using artificial neural networks. *Artif Intell Eng Design Anal Manuf*. 2015;**29**:371-391.
16 Liu Z, Zhu P, Chen W, Yang RJ. Improved particle swarm optimization algorithm using design of experiment and data mining techniques. *Struct Multidisc Optim*. 2015;**52**:813-826.
17 Hanna S. Inductive learning of optimal modular structures: estimating solutions using support vector machine. *Artif Intell Eng Design Anal Manuf*. 2007;**21**:351-366.
18 Liu X, Wu Y, Wang B, Ding J, Jie H. An adaptive local range sampling method for reliability-based design optimization using support vector machine and Kriging model. *Struct Multidisc Optim*. 2017;**55**:2285-2304.
19 Yang IT, Hsieh YH. Reliability-based design optimization with corporation between support vector machine and particle swarm optimization. *Struct Multidisc Optim*. 2013;**29**:151-163.
20 Hagishita T, Ohsaki M. Topology mining for optimization of framed structures. *J Adv Mech Design Sys Manuf*. 2008;**2**(3):417-428.
21 Szczepanik W, Arciszewski T, Wnek J. Empirical performance comparison of selective and constructive induction. *Eng Appl Artif Intell*. 1996;**9**(6):627-637.
22 Schwabacher M, Ellman T, Hirsh H. Learning to set up numerical optimizations of engineering design. *Artif Intell Eng Design Anal Manuf*. 1998;**12**:173-192.
23 Song KT, Sun WY. Robot control optimization using reinforcement learning. *J Intell Robotic Sys*. 1998;**21**:221-238.
24 Hagishita T, Ohsaki M. Topology optimization of discrete structures by integration of heuristics and nonlinear programming. *J Struct Constr Eng AIJ*. 2008;**73**(633):1959-1965. (in Japanese).
25 Aldwaik M, Adeli H. Advances in optimization of high rise building structures. *Struct Multidisc Optim*. 2014;**50**:899-919.
26 Mijar AR, Swan CC, Arora JS, Kosaka I. Continuum topology optimization for concept design of frame bracing systems. *J Struct Eng ASCE*. 1998;**124** (5):541-550.
27 Stromberg LL, Beghini A, Baker WF, Paulino GH. Application of layout and topology optimization using pattern gradation for the conceptual design buildings. *Struct Multidisc Optim*. 2011;**43**:165-180.
28 Zhu M, Yang Y, Guest JK, Shields MD. Topology optimization for linear stationary stochastic dynamics: applications to frame structures. *Struct Safety*. 2017;**67**:116-131.
29 Park K, Oh BK, Park HS. GA-based multi-objective optimization for retrofit design on a multi-core PC cluster. *Comp Aided Civil Infrastructural Eng*. 2015;**30**:965-980.
30 Fukushima K, Ohsaki M, Mikami T, Miyazu Y. Combinatorial optimization of latticed blocks composed of various unit shapes for seismic retrofit. *J Struct Constr Eng AIJ*. 2016;**81**(728):1657-1664. (in Japanese).
31 Open System for Earthquake Engineering Simulation (OpenSees), PEERC, UC Berkeley. http://opensees.berkeley.edu/. Accessed March 22, 2018.
32 The MathWorks Inc. *Matlab User's Manual Ver. R2016a*. 2016.

How to cite this article: Tamura T, Ohsaki M, Takagi J. Machine learning for combinatorial optimization of brace placement of steel frames. *Jpn Archit Rev*. 2018;1:419–430. https://doi.org/10.1002/2475-8876.12059