Electronic Journal of Structural Engineering

Original Article

Cite this: DOI:10.56748/ejse.233642

Received Date: 22 September 2022
Accepted Date: 30 January 2023

1443-9255
https://ejsei.com/ejse

Machine learning-based optimum reinforced concrete design for progressive collapse

Mohammad Javad Esfandiari [a]*, Homa Haghighi [a] and Girum Urgessa [a]
[a] Department of Civil Engineering, George Mason University, VA, USA
* Corresponding Author: Mesfandi@gmu.edu

Abstract

This paper investigated progressive collapse analysis of three-dimensional (3D) reinforced concrete (RC) frames that are optimized for carrying structural loads by introducing a unique simultaneous multi-column removal using Machine Learning. The various load paths resulting from multiple-column removal are incorporated in the optimization automatically. The investigation includes formulating an integrated computational framework that incorporates a self-training machine learning algorithm. The efficiency of the algorithm is tested by using several hundreds of optimized structures. The efficiency of the computational framework was shown by conducting a comprehensive study on the optimization and behavior of structures considering seismic loading, alternative load path due to progressive collapse, and second order (P–delta) effects. The results show that the proposed framework ensures that system solutions meet both structural integrity and constructability requirements of the ACI and the Unified Facilities Criteria.

Keywords

Machine Learning, Structural Optimization, Progressive Collapse, Reinforced Concrete, Artificial intelligence

1. Introduction

Local failure of structural members will result in increased internal forces and overloading and may cause a progressive collapse of the entire or a part of a given structures. A few examples of progressive collapse (PC) include the Murrah Building in Oklahoma City in 1995, the Twin Towers in New York City in 2001, and the Plasco Building in Tehran in 2017, and the Hard Rock Hotel in New Orleans in 2019. PC results in economic losses and death of occupants. Hence, structures must be designed to better withstand progressive collapse.

The number of PC studies consisting of experimental programs and numerical studies have increased significantly since 2001. Byfield et al (Byfield et al., 2014), Wang et al (Wang et al., 2014) and Qian and Li (Qian and Li, 2015) provided detailed reviews on this topic.

As far as numerical studies are concerned, various representative models are found in literature (Bao et al., 2008; Talaat and Mosalam, 2007; Buscemi and Marjanishvili, 2005) including computational FEM models for determining the response of structures before failure with a reasonable accuracy and DEM models that are more effective for moving and collision between rigid bodies after failure. Marchand et al (Marchand and Stevens, 2015) argued that common structural design software (e.g., SAP2000 and ETABS) can provide reasonable results than those determined with high-fidelity physics-based software. Esfandiari et al (Esfandiari and Urgessa, 2018) presented a non-linear time history pull-down of a two-span steel frame in ETABS and showed that the numerical results were in excellent agreement with experimental results.

Machine Learning (ML) and Artificial Intelligence (AI) applications in recent years have opened new opportunities for use in traditional engineering problems. Machine learning mostly deals with problems where paired examples, X → Y, exist. For this paper, X and Y can be interpreted as features of the structure and the optimum solution, respectively. As such, the machine learning technique attempts to map G: X → Y` with the translated domain Y` distributed identically to Y. However, in the field structural engineering, the main challenge is the lack of adequate optimum data that is needed to train the algorithm. Therefore, seeking an algorithm that could learn to map between domains without paired input-output scenarios is crucial.

Zue et al. (Zhu et al., 2017) used Cycle-Consistent Adversarial Networks in image-to-image transformation for learning to map between an input and output image when paired examples are absent. They simultaneously trained G: X → Y and another translator F: Y→ X with a cycle consistency loss such that F(G(x)) ≈ x and G(F(y)) ≈ y. However, in structural analysis, it is important to create an alternative algorithm for capturing the complex system behavior where few training paired data are available.

Sra et al. (Sra et al., 2012) showed that learning from available dataset combined with optimization is applicable to a myriad of complex, dynamic, and stochastic problems. Mosavi et al. (Mosavi and Varkonyi-Koczy, 2017) combined machine learning with optimization to increase the learning ability of robots. They concluded that integrating ML and optimization significantly increases the quality of decision making and learning capability in decision systems.

As far as machine learning techniques, Jong-Su Jeon et al. (Jeon et al., 2014) proposed probabilistic joint shear strength models by implementing ML. They used the prediction model as joint response models for evaluation of seismic performance and inelastic responses of frames. (Nick et al., 2015) have used different machine learning techniques for identifying the existence and location of damage, and the type and severity of damage. Ni Hong-Guang and Wang Ji-Zong (Ni and Wang, 2000) used a multi-layer feed-forward neural network and presented a method for predicting 28-day compressive strength of concrete. Dac-Khuong Bui et al. (Bui et al., 2018) developed a model for determining the tensile strength of High-Strength Concrete. They selected neural network for their research due to the nonlinear relation between concrete strength and its components.

In this paper, classification techniques are used because the algorithm needs to correctly determine the class labels for unseen instances on the basis of previously observed optimum structural system data and suggests a specific class to the optimization for further constraints handlings. The algorithm starts to run with a few training data and as it is applied on different structures, it can consider the result for its future training data. As the training data becomes more available, the performance improves dramatically over time. Different classification techniques such as decision tree, SoftMax, and nearest neighbors are presented.

2. Structural optimization problem for reinforced concrete frames

Esfandiari et al (Esfandiari et al., 2018b) presented a general structural optimization problem as shown in Equation 1.

$$\begin{aligned} & x = [x_1, \dots, x_n]^T \\ & g(x) \le 0, \qquad k = 1, \dots, m \\ & x^L \le x \le x^U \end{aligned} \tag{1}$$

where x is a vector of n structural system variables, $f(x)$: $R^n \rightarrow R$ is the objective function which returns a scalar value to be minimized, the vector function $g(x)$: $R^n \rightarrow R^m$ returns a vector of length m containing the values of the inequality constraints evaluated at x, and x^L, $x^U$are two vectors of length n containing the lower and upper bounds of the structural system variables, respectively. The above equation contains only inequality constraints because equality constraints are usually not found in structural optimizations.

Equation 2 shows a common constraint k in a structural optimization problem.

$$g_k(x) = |q_k(x)| - q_{allow,k} \tag{2}$$

where $q_k(x)$ is a response measure for analysis and design x; and $q_{allow,k}$ is its maximum allowable absolute value.

The objective function in structural optimization problems is generally defined by the weight or total cost of the structure (Tahmouresi et al., 2021). When considering total cost, the concrete, steel and labor costs are included. However, when the objective function is the total weight, only the weight of concrete and steel are included. Equation 3 shows the resulting objective function.

$$f_{obj} = Obj_c + Obj_s + Obj_f \tag{3}$$

where $Obj_c, Obj_s, and\ Obj_f$ are the costs of concrete, reinforcing steel bars, and labor, respectively.

Equations 4-6 show the costs of each component when the objective function is the total cost of a structural frame.

$$Obj_c = C_C\left(\sum_{i=1}^{N_{col}} b_i.d_i.L_{n\ column,i} + \sum_{j=1}^{N_{beam}} b_{w\ j}.h_j.L_{beam\ j}\right) \tag{4}$$

$$Obj_s = C_s.\gamma_s.\left(\sum_{i=1}^{N_{col}}\sum_{j=1}^{N_{bar,i}} A_{St\ j}.L_{bar\ j} + \sum_{i=1}^{N_{col}}\sum_{k=1}^{N_{tie,i}} A_{Sh\ k}.L_{tie\ k} + \sum_{m=1}^{N_{beam}}\sum_{l=1}^{N_{bar,m}} A_{St\ l}.L_{bar\ l} + \sum_{m=1}^{N_{col}}\sum_{n=1}^{N_{tie,m}} A_{Sh\ n}.L_{bar\ n}\right) \tag{5}$$

$$Obj_f = C_f\left(\sum_{i=1}^{N_{col}}\left[2(b_i + h_i).L_{n\ column,i}\right] + \sum_{j=1}^{N_{beam}}\left[\left(b_{w\ j} + 2h_j\right).L_{beam\ j}\right] - \sum_{k=1}^{N_{col}} b_k + d_k\right) \tag{6}$$

where N_{col} , N_{beam} , b, d, b_w , h, L, and L_n are the number of columns, the number of beams, the width of column, the depth of column, the width of beam, the height of beam, the length of the members, and the length of clear span between supports, respectively; C_C, C_f and C_s are unit cost of the concrete, the labor and the steel, respectively; A_{st} , L_{bar} , and N_{bar} are the area, the length and the number of main rebars placed in the member while A_{sh} , L_{tie}, and N_{tie} are the area, the length and the number of shear reinforcement used in the member respectively; and γ_s is the density of rebars (kg/m3).

3. Proposed DMPSO-ML algorithm

3.1 Overview

Figure 1 shows the basic concept of the particle movement in the traditional Particle Swarm Optimization method (PSO) versus the proposed Decision-Making Particle Swarm Optimization method (DMPSO). As shown in the figure, each particle in PSO only searches for the best solution according to its own best experience, and the best solution is determined by all particles (Kennedy and Eberhart, 1995). However, in structural optimization, it is important to ensure that the structure is stable and safe. An experienced structural engineer can decide the sort of alterations in the system variables that could lead to a preferable solution. As an example, if the demand to capacity ratio of a member is greater than one, the acceptable solution for addressing this issue may be changing the cross-sectional sections of the member or the rebar reinforcement ratio of the section. However, a human decision maker cannot be available and actively participate in the solution process and direct it according to the preferences in the entire process of an optimization. The decision maker (DM) algorithm acts similar to an experienced structural engineer.

More details regarding how the DM is formulated and operates, and how it is fused with PSO, are discussed in the following subsections.

3.2 DMPSO Algorithm Enhanced with ML

DMPSO uses an informed strategy and the knowledge beyond the definition of the problem itself, to empower PSO optimization algorithm and accelerate convergence toward the optimum solution. The DM formulation in this paper, which is inspired by Bayes' theorem, seeks the probability of a member not failing given the geometry and loading application. Bayes' theorem is stated mathematically by Equation 7.

$$P(B) = \frac{P(A)P(A)}{P(B)} \tag{7}$$

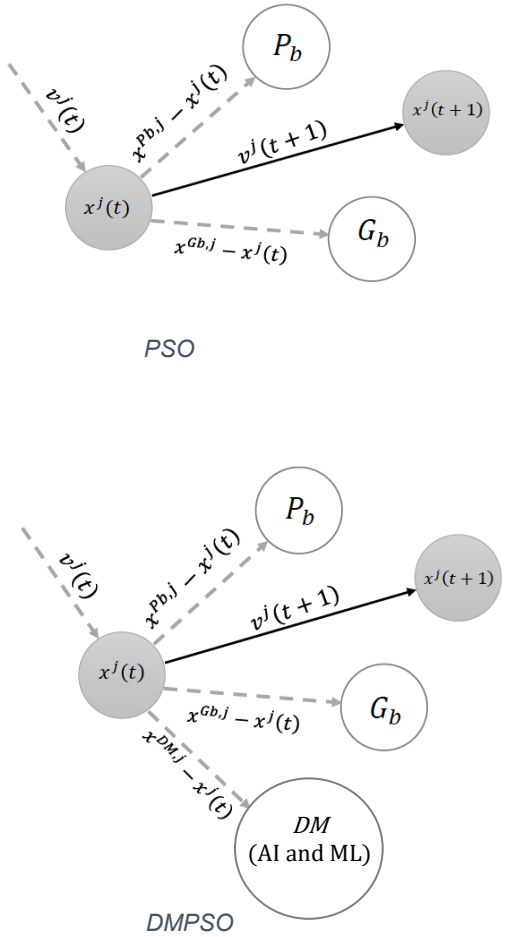

Fig 1. PSO versus DMPSO

The probability of geometry and loading for a given member specification can be determined from previously analyzed optimum structures. Accordingly, DM implements machine learning to find the most probable structural member according to the geometry, loading condition and location of the member. However, for incorporating dependencies, Bayes' theorem needs fundamental assumptions about dependence and independence between system variables, and determining the marginal in the Bayes' theorem is computationally expensive. For that reason, alternative machine learning methods can be cost effective and are investigated in this paper.

The decision maker algorithm can also disregard a solution at any given time of the computation process when it determines that a better fitness cannot be obtained. The principle of pruning from AI was adopted in this paper, which allows the DM algorithm to ignore portions of the search space or analysis that make no difference to the final choice. The heuristic evaluation function allows us to estimate the objective function without doing a complete analysis. When pruning is applied to a standard search tree, it returns the same move as a search would, but it prunes away branches that cannot possibly influence the final decision. This requires examining first the successors that are likely to be the optimum solution.

To equip the DM with machine learning power, different ML methods were investigated in this paper.

The first step was the collection and preparation of the training data set. Since there was no training data available at the beginning, a small sub-set of structures was selected, and the optimization algorithm was used for producing the training set. The small sub-set of structures was later used to produce more complex training sets. Then, the behavior of 640 more complex structures were considered as the training data. For each structure, the structure was separately optimized for 10 random column removal scenarios for progressive collapse analysis. The goal was to classify the best cross section for the elements under different loading conditions. For this purpose, three separate machine learning (ML) models were trained. Table 1 provides the specifics of these ML models.

Table 1. Detail of the ML models

	Model 1	Model 2
Features	Number of bays in each direction Maximum bay span in each direction Number of stories Dead load Live load Seismic parameters	Element type Maximum adjacent bays length at the element Moment and shear of the element Top and bottom connected beams, The number of stories above the element The number of stories below the element.
Usage	Initial randomized section at the beginning	Optimum separate elements based on conventional loading

Model 1 related the final result of the average size of optimum elements in each story to the overall geometric feature of the structure, including number of bays in each direction, maximum bay span in each direction, number of stories, dead load, live load and seismic parameters. This model was only used in the first iteration to generate initial randomized sections.

The second and third models connected the output of the optimum separate elements to its learning feature for the whole structure and progressive collapse removal scenario cases, respectively. The learning features considered include element type, actual bay length, moment and shear of the element before and after removal scenarios, bottom and top connected beams, and the number of stories above and below the removed element.

To streamline the problem for ML and to avoid overfitting, the class of the sections for columns and beams were restricted to 8 and 6 sections, respectively as shown in Fig. 2 and Fig. 3. Rectangular cross-sections were deliberately considered with 100mm difference in width and height to have the best arrangement of the classes covering most of the practical results.

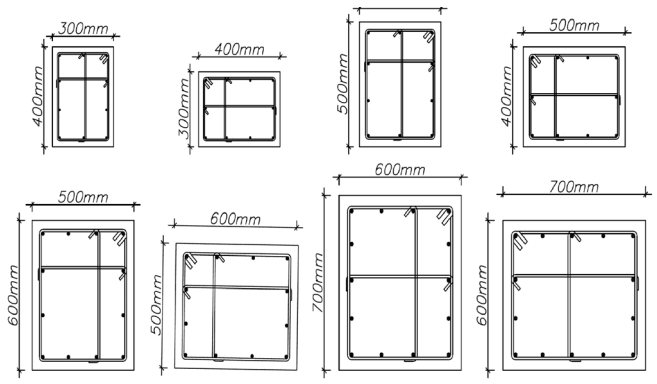

Fig 2. Different section classes for columns in the ML classification problem

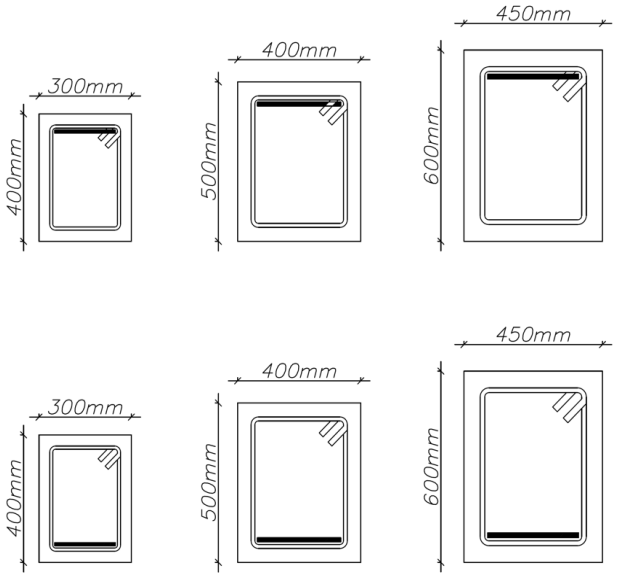

Fig 3. Different section classes for beams in the ML classification problem

For the beams, cross sectional dimensions and the location of the rebar were considered as variables. The corresponding reinforcement amount was calculated and placed later based on strength requirements.

The section restrictions did not affect the final results because the initial guess from ML is fed to the optimization algorithm with no restriction on its variable. In other words, these initial guesses show the preference of the DM to the optimization algorithm and guide it to find a better solution.

For identifying the best ML technique that will be integrated with the optimization algorithm, the following classifiers were investigated.

3.3 K-Nearest Neighbor (KNN)

The K-nearest-neighbor (KNN) density estimation method (Paya et al., 2008) was applied to each class followed by employing Bayes' theorem. Consider a data set comprising Nk members' cross section in class Ck with a total of N points. To classify a new structural element x, a sphere centered on x can be drawn in the feature space, precisely holding K elements' cross section regardless of their class. Suppose this sphere contains Kk member from class Ck. The posterior probability of selecting a structural element's cross section can be obtained by applying the Bayes' theorem as shown in Equation 8.

$$p(C_k) = \frac{P(C_k)p(C_k)}{p(x)} = \frac{K_k}{K} \tag{8}$$

In the KNN formulation, the cross section with the largest posterior probability should be assigned to element x to minimize the probability of misclassification. When the algorithm needs to find an appropriate cross section for a new element, it can identify the K nearest similar members from the training data set of optimum structures and then assign a cross section with the average of variables from the KNN. It is important to find the best K for the problem. Small Ks result in many small regions of each class and make the model more biased. On the other hand, large Ks led to fewer larger regions which may affect the final result.

3.4 SoftMax Classifier

SoftMax classifier (Duan et al., 2003) is the generalization of binary Logistic Regression classifier to multiple classes. Softmax classifier uses a linear classifier for mapping and generating scores as the unnormalized log probabilities with cross-entropy loss having the form shown in Equation 9.

$$L_i = \log\left(\frac{e^{f_{yi}}}{\sum_j e^{f_j}}\right) \tag{9}$$

Where fj is the j-th element of the vector of class scores f. The use of the exponential scores gives the unnormalized probabilities, and the division for normalization purpose. This will ensure that the sum of the probabilities is one. The stochastic gradient descent was used for training. Here the best section that has the highest probability for the corresponding element was sought. The data was trained in 16 mini batches.

3.5 Decision Tree Classifier

Decision tree classification algorithms (Safavian and Landgrebe, 1991) have a significant potential for a variety of problems and have been used in civil engineering applications. There are different measures that can be utilized to determine the best way to split between classes. Gini index and entropy were used for selecting the best split based on the degree of impurity of the child nodes. Binary decision tree with 5 and 6 depths were tested.

4. Metrics and performance evaluation

The data set used here was divided into 3 different groups: training, validation, and testing. The training dataset was used to train models with various hyper parameter values. Then the validation dataset was used to identify the best working parameters. To validate the model, a 5-fold cross validation was used. After that, the training and validation datasets were used to train the final model.

Evaluation presented a major challenge. Consider a single column under pure compression loading. This column can be designed with different cross sections that can satisfy the stability requirements of the structure. Since the optimization does not warrant the global optimum in a complex system, it accepted near-optimum solutions in the evaluation. Moreover, the element classes of the final result did not exactly match the output of the machine learning classes. For the machine learning outputs, only 8 classes were considered for the columns and 6 classes for the beams, while the final result of the optimum structure does not have any restriction and the dimensions of the elements might change through the optimization process. Therefore, if the structural requirements were only checked, all

over designed solutions would pass the evaluation criteria. On the other hand, if the optimum solution was only checked, the accuracy would be very low. This issue was addressed by finding the nearest neighbor of the actual sections of the structure with class samples of one increment threshold for accepting the result.

Finding the final optimum solution was not the goal of this initial step but rather keeping the variables within an acceptable range of initial guesses. Later the algorithm would find the best optimum solution. Therefore, different hyperparameters and methods were investigated, and the best parameters were selected for integrating it with the optimization algorithm. The comparison of the results is shown in Fig. 4.

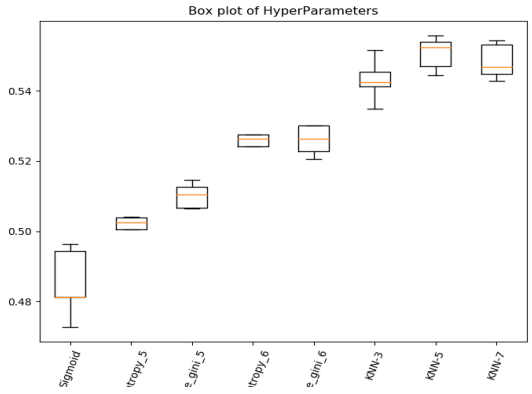

Fig 4. Box plot of different hyper parameters

The number and size of rebars in each direction of the concrete section presented a challenge because specific required rebar area could be placed in several arrangements. As an example, 450 mm2 of required rebar area can be satisfied by using 6-D10, 4-D12, or 2-D25 bars. The results can be improved by only considering the cross-sectional dimension parameters and the overall required area of the rebar for the section with 10 percent threshold for the columns and the beams. Later, when this section is ready to be fed to the optimization algorithm, the overall required area would be converted to the best arrangement for the rebars size and numbers in each direction, for that specific section. This approach dramatically improved the findings obtained as shown in Fig. 5.

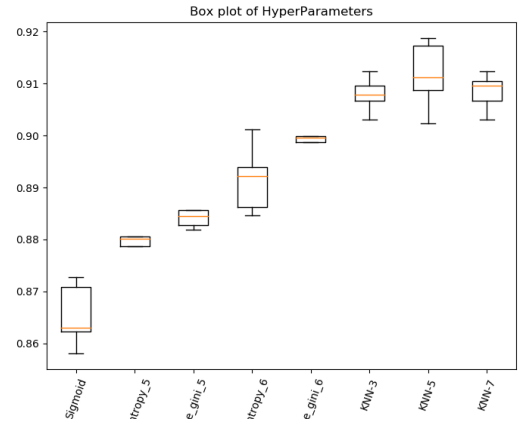

Fig 5. Box plot when only cross-sectional dimensions were considered as variables

The KNN-5 method resulted in the highest accuracy compared to all the techniques investigated. The DMPSO then modifies its velocity and position according to its experience, neighboring particles' experience, and the preference of the decision maker (DM) by employing Equation 10 and Equation 11 (Esfandiari et al., 2018a).

$$v^j(t+1) = wv^j(t) + c_1 r_1 \odot \left(x^{Pb,j} - x^j(t)\right) + c_2 r_2 \odot \left(x^{Gb,j} - x^j(t)\right) + c_3 r_3 \odot \left(x^{DM,j} - x^j(t)\right) \quad (10)$$

$$x^j(t+1) = x^j(t) + v^j(t+1) \quad (11)$$

Where $v^j(t)$ *and* $x^j(t)$ represent the velocity and the position vectors of particle j at time t, respectively. The term w is a modifier employed to control the exploration capabilities of the swarm. Vector $x^{Pb,j}$ denotes the personal best position which is registered by particle j, vector $x^{Gb,j}$ is the global best position attained by the entire swarm up to the current iteration, and vector $x^{DM,j}$ indicates the position of preference of the decision maker in the search space. The acceleration coefficients c_1 and c_2, and c_3 rule the impact of the particle's own experiences, the other particles' experiences, and the decision maker's preference on the trajectory of each particle, respectively. r_1, r_2, r_3 are three random vectors with uniformly distributed numbers in the interval [0, 1]. The symbol "⊙" is the element-wise product of two vectors. The two acceleration coefficients (c1 and c2) and the smallest and largest value of inertial factor (wmin and wmax) were taken as 2.025, 2.025 and 0.4, 0.9, respectively (Alam et al., 2015; Eberhart and Shi, 2000). The acceleration coefficient for DM, c_3, was initially taken as 2.025 (the same as c_1 and c_2) and decreased over iterations. Therefore, for the first few iterations, the algorithm mostly relies on the DM. To ensure the functionality, if the demand to capacity ratio of a member is not within 50% in the optimization process, the DM algorithm suggests its preference, such as increasing or decreasing a relevant parameter, to the DMPSO algorithm.

The DM algorithm gathers statistics from a database of previously analyzed structures to determine members most often lead to an optimum structure. In the early iterations, there were a few choices among the large number of possible variables. Thus, the DM commentary based on past structures has a higher impact on DMPSO. Usually after the first 100 iterations, the DMPSO algorithm mostly relies on optimization rather than the DM preference.

5. Incorporating progressive collapse in DMPSO-ML

In progressive collapse analysis, multiple scenarios of removing critical members should be considered, which drives the structural system and cross-section selection to be tedious and costly. Therefore, investigating the formulation of a computational framework is important for producing d cost-effective solutions. This was achieved by a series of steps. First, a finite element model capable of accurately modeling new load paths to progressive collapse analysis was developed. Then, the finite element model was integrated with DMPSO to automatically evaluate structural response for progressive collapse.

There are two groups of constraints needed for expanding the optimization problem and incorporating progressive collapse. These include general concrete structural system constraints and progressive collapse (UFC) constraints. The first group includes parameters typical of structural systems subjected to traditional lateral loads such as plastic rotations.

The second group includes those related to progressive collapse as defined in UFC (Gsa, 2003; Defense, 2005) and GSA. These constraints ensure that the structure is capable of bridging over critical vertical load-carrying elements that are eliminated during a progressive collapse scenario such as redundancy requirements.

The integrated framework of the DMPSO algorithm that is empowered by ML is shown schematically in Fig. 6 and Fig. 7.

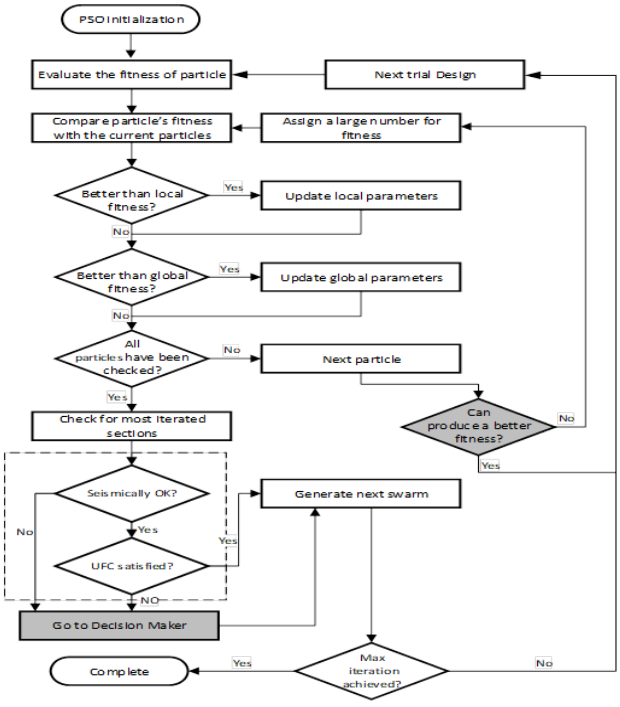

Fig 6. Modified DMPSO for Progressive Collapse

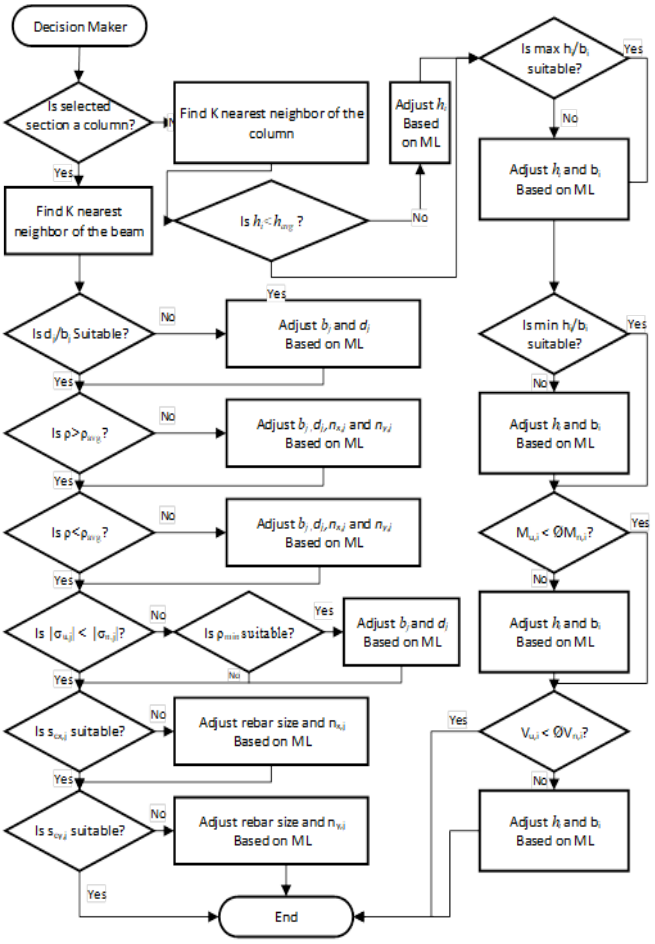

Fig 7. Flowchart of decision maker empowered by Machine Learning (ML)

6. Practical example on progressive collapse analysis of 3D optimized RC frames.

The proposed DMPSO-ML framework was used to analyze a given 7-story RC frame with three spans whose geometry, grouping details, and removal scenarios are shown in Fig. 8. A total of sixteen unique column removal scenarios were implemented in the alternate path investigation of this example frame. Although a symmetric plan was used, sections A and C differ in size. This implies that one set of eight removal scenarios, as shown in Fig. 8, is considered in each of these sections. These columns eliminated are located at the corner and middle of each direction of the members in the first story above the grade, the story directly below the roof, and the story above the location of a column splice at every other floor. The objective function here is the construction weight similar to a baseline optimized structure for conventional loadings without considering progressive collapse (Esfandiari et al., 2018b). The frame includes 180 members, 84 beams and 96 columns, which were arranged into 42 groups: 28 groups for beams and 14 groups for columns. It contains 574 system variables, 504 were for beams and 70 were for columns. Beams and columns were grouped to satisfy the uniformity of members and having similar behaviors according to their place in the frame and loading conditions. To ensure best results for the stochastic decline, a population size of 150 was selected.

Table 2 shows the optimal frame systems from the present progressive collapse optimization, considering all of the given removal scenarios Both linear static (LS) and nonlinear dynamic (ND) methods were considered. Furthermore, the result of the optimal frame analysis without including progressive collapse is also presented as the baseline to depict the changes made in the structural member and reinforcing steel sizes of optimal results when compared to current integrated progressive collapse analysis.

Fig. 9 shows DMPSO-ML algorithm evolutions for obtaining the solutions. DMPSO did not confine local values and carried on converging. As validated, the baseline structure without considering the progressive collapse requirements converged to the optimum results in smaller number of structural analysis and flatten out within less generations. This is expected because for progressive collapse analysis DMPSO has to confirm the constraints in two steps. In the first step, constraints related to traditional lateral systems (e.g., seismic) requirements were checked (Randall W. Poston and Basile G. Rabbat, 2011). If the criteria are not met here, it proceeds to the Decision Maker (DM) portion of the algorithm for adjusting parameters. Otherwise, it proceeds to the second step for checking progressive collapse requirements. The results are not registered until every constraint consisting of both lateral and progressive collapse requirements are met.

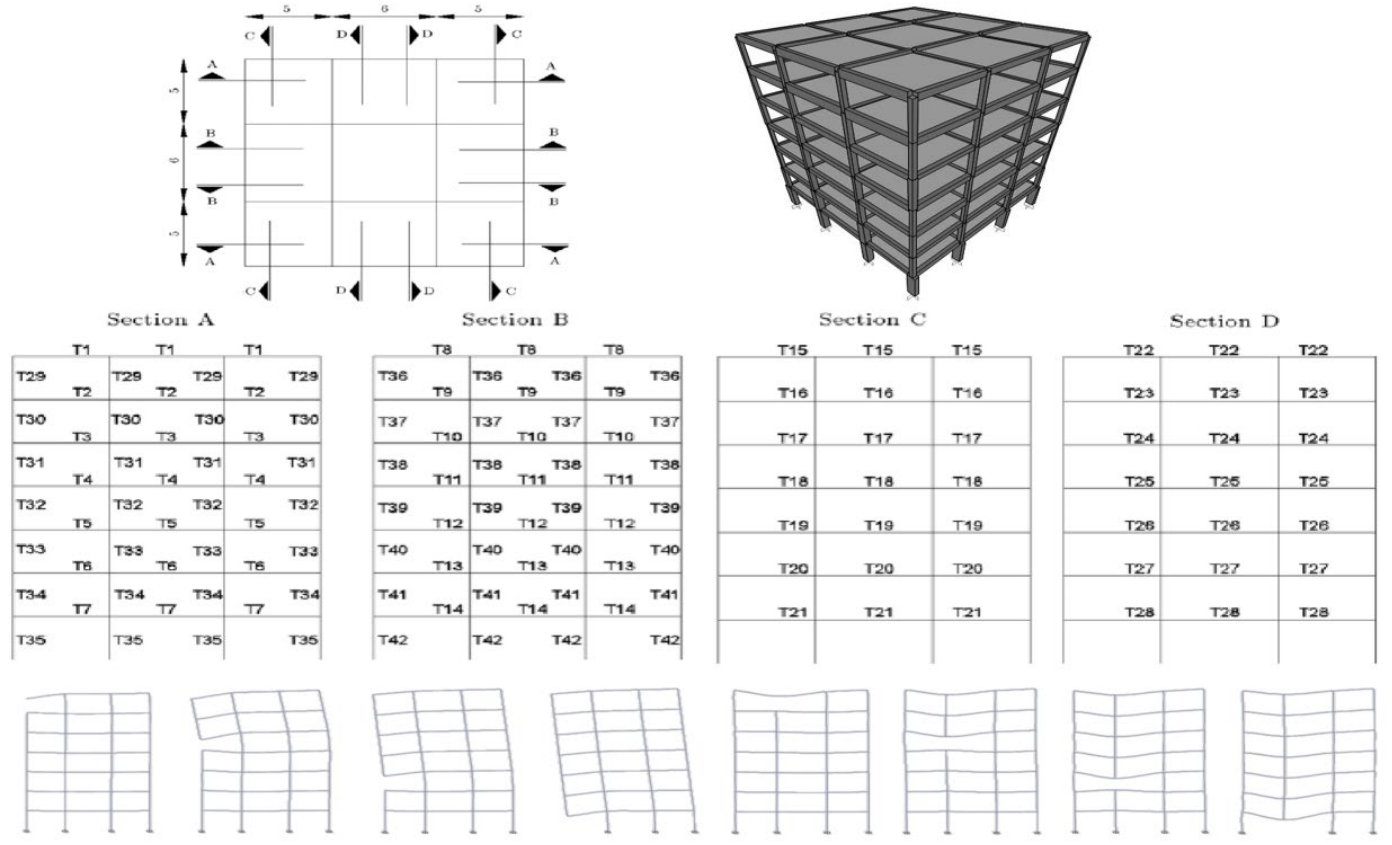

Fig 8. Geometry, member classification, and removal scenarios

Thus, the need for evolutionary generation to handle these two types of constraints is evident in the progressive analysis case.

Fig. 10 illustrates the demand to capacity ratio (DCR). The demand capacities for LS and ND were calculated for regular loading after the structure was totally designed for progressive collapse. This implies that after the process of removing elements are completed and the optimum result was obtained, the DCR is calculated using those sections. Bigger sections were used in LS and ND when compared to the structure considered without progressive collapse. As a result, the DCR of elements in LS is the least among all the methods. Nevertheless, all the DCRs obtained were above 0.61, which shows that the algorithm can obtain acceptable results.

Note that in the ML algorithm, the results are obtained instantaneously given data is used to train the algorithm and the pretrained model is used for prediction. The optimization piece requires close to 12 hours to run because each iteration takes around 10 seconds.

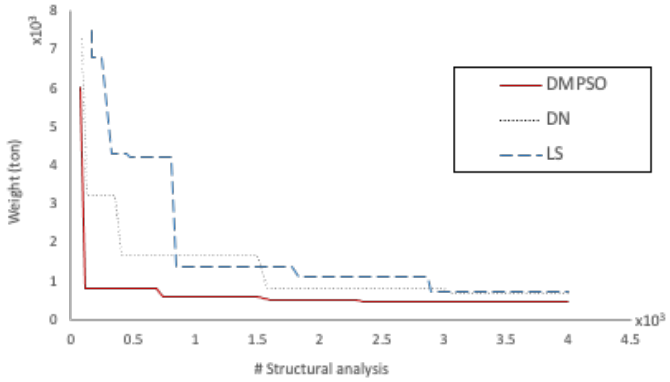

Fig 9. Convergence rate from DMPSO, based on LS, ND compared to the system without considering progressive collapse.

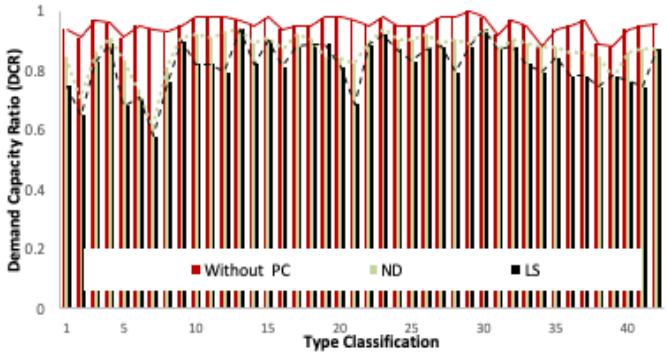

Fig 10. Maximum DCR of members for each analysis method

7. Conclusion

This paper presented the integration of optimization and progressive collapse analysis computational framework empowered by machine learning. The main objective was to evaluate the behavior of reinforced concrete structures while satisfying the limitations and specifications of the American Concrete Institute 318 code and Unified Facilities Criteria progressive collapse requirements. Three machine learning algorithms, K nearest neighbor, SoftMax, and decision tree classifiers were evaluated. The KNN machine learning algorithm provided better performance to alleviate the computational challenge for the structural optimization problem involving progressive collapse.

Using a case study, the analysis was shown to enhance load redistribution capability of the structure by considering the alternate path criteria through finding appropriate structural member sizes. Meanwhile, the cost of the example frame analyzed reduced substantially through the optimization process. The case study showed the capability of the DMPSO-ML algorithm to accelerate convergence toward the optimum system solutions while reducing computational effort.

References

Alam MN, Das B and Pant V. (2015) A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination. *Electric Power Systems Research* 128: 39-52.

Bao Y, Kunnath SK, El-Tawil S, et al. (2008) Macromodel-Based Simulation of Progressive Collapse: RC Frame Structures. *Journal of Structural Engineering* 134: 1079-1091.

Bui D-K, Nguyen T, Chou J-S, et al. (2018) A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete. *Construction and Building Materials* 180: 320-333.

Buscemi N and Marjanishvili S. (2005) SDOF Model for Progressive Collapse Analysis. *Structures Congress 2005.* 1-12.

Byfield M, Mudalige W, Morison C, et al. (2014) A review of progressive collapse research and regulations. *Proceedings of the Institution of Civil Engineers - Structures and Buildings* 167: 447-456.

Defense Do. (2005) Unified Facilities Criteria: Design of Buildings to Resist Progressive Collapse (UFC 4-023-03). Department of Defense Washington, DC, USA.

Duan K, Keerthi SS, Chu W, et al. (2003) *Multi-category Classification by Soft-Max Combination of Binary Classifiers,* Berlin, Heidelberg: Springer Berlin Heidelberg.

Eberhart RC and Shi Y. (2000) Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512).* 84-88

Esfandiari M and Urgessa G. (2018) A Pull-down Dynamic Analysis of Two-Span Steel Frames Subjected to Progressive Collapse.

Esfandiari MJ, Urgessa GS, Sheikholarefin S, et al. (2018a) Optimization of reinforced concrete frames subjected to historical time-history loadings using DMPSO algorithm. *Structural and Multidisciplinary Optimization* 58: 2119-2134.

Esfandiari MJ, Urgessa GS, Sheikholarefin S, et al. (2018b) Optimum design of 3D reinforced concrete frames using DMPSO algorithm. *Advances in Engineering Software* 115: 149-160.

Gsa U. (2003) Progressive collapse analysis and design guidelines for new federal office buildings and major modernization projects. *Washington, DC*.

Jeon J-S, Shafieezadeh A and DesRoches R. (2014) Statistical models for shear strength of RC beam-column joints using machine-learning techniques. *Earthquake Engineering & Structural Dynamics* 43: 2075-2095.

Kennedy J and Eberhart R. (1995) Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks.* Perth, WA, Australia: IEEE, 1942-1948

Marchand KA and Stevens DJ. (2015) Progressive Collapse Criteria and Design Approaches Improvement. *Journal of Performance of Constructed Facilities* 29: B4015004.

Mosavi A and Varkonyi-Koczy AR. (2017) Integration of Machine Learning and Optimization for Robot Learning. *Advances in Intelligent Systems and Computing Recent Global Research and Education: Technological Challenges,* 519: 349-355.

Ni H-G and Wang J-Z. (2000) Prediction of compressive strength of concrete by neural networks. *Cement and Concrete Research* 30: 1245-1250.

Nick W, Asamene K, Bullock G, et al. (2015) A study of machine learning techniques for detecting and classifying structural damage. *International Journal of Machine Learning and Computing* 5: 313.

Paya I, Yepes V, González-Vidosa F, et al. (2008) Multiobjective Optimization of Concrete Frames by Simulated Annealing. *Computer-Aided Civil and Infrastructure Engineering* 23: 596-610.

Qian K and Li B. (2015) Research Advances in Design of Structures to Resist Progressive Collapse. *Journal of Performance of Constructed Facilities* 29: B4014007.

Randall W. Poston and Basile G. Rabbat. (2011) *Building Code Requirements for Structural Concrete (ACI 318-11)*: American Concrete Institute.

Safavian SR and Landgrebe D. (1991) A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics* 21: 660-674.

Sra S, Nowozin S and Wright SJ. (2012) *Optimization for machine learning,* Cambridge, Massachusetts, London, England: Mit Press.

Tahmouresi, A., Robati, A. et al. (2021) A Combined Genetic Algorithm-Artificial Neural Network Optimization Method for Mix Design of Self Consolidating Concrete, *International Journal of Structural and Civil Engineering Research*, Vol. 10, No. 3, doi: 10.18178/ijscer.10.3.106-112.

Talaat M and Mosalam KM. (2007) Towards Modeling Progressive Collapse in Reinforced Concrete Buildings. *Research Frontiers at Structures Congress* Long Beach, California, United States: American Society of Civil Engineers, 1-16.

Wang H, Zhang A, Li Y, et al. (2014) A review on progressive collapse of building structures. *The Open Civil Engineering Journal* 8: 183-192.

Zhu J-Y, Park T, Isola P, et al. (2017) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. (Accessed March 01, 2017).

Appendix

Table 2. Results of optimum systems based on LS; ND compared to the system without considering progressive collapse.

	LS								ND								DMPSO							
	Sectional dimensions		Reinforcements						Sectional dimensions		Reinforcements						Sectional dimensions		Reinforcements					
Type	Height	Width	Top-rebar				Bot-rebar		Height	Width	Top -rebar		Bot -rebar				Height	Width	Top -rebar		Bot -rebar			
T1	350	350	5-D16 3-D12	3-D14	6-D16 4-D12	6-D16 4-D12	3-D12 1-D16	5-D16 3-D12	350	350	5-D12 4-D12	3-D14	5-D12 3-D14	3-D12 3-D14	4-D14 4-D16	3-D12 3-D14	300	300	3-D12 3-D14	3-D12	3-D12 3-D14	3-D12 3-D14	3-D12 1-D16	3-D12 3-D14
T2	400	350	3-D16 3-D16	3-D14	5-D16 5-D16	3-D16 3-D16	6-D14	4-D16 4-D16	350	350	5-D12 5-D12	3-D14	5-D12 5-D12	5-D14	4-D12 4-D14	5-D14	350	350	3-D14 3-D14	3-D14	3-D14 3-D14	3-D14	3-D14	3-D14
T3	350	350	4-D14 4-D12	3-D12	4-D14 4-D12	5-D14 4-D12	3-D12 1-D16	6-D14 4-D12	350	300	4-D16 2-D14	3-D12	3-D16 3-D14	3-D12	2-D12 2-D12	3-D12	300	300	3-D12 4-D16	3-D12	3-D12 4-D16	3-D12	3-D12 1-D16	3-D12
T4	350	350	5-D16 4-D14	3-D14	4-D16 3-D14	2-D16 2-D14	5-D14	2-D16 3-D14	350	350	4-D14 4-D14	3-D14	3-D14 3-D14	5-D14	5-D14 5-D12	4-D14	350	350	3-D14 2-D22	3-D14	3-D14 2-D22	4-D14	4-D14	4-D14
T5	400	300	5-D16 4-D14	3-D12	5-D16 4-D14	3-D16 1-D14	3-D12 1-D16	2-D16 2-D14	350	350	5-D14 4-D12	3-D12	5-D14 4-D12	3-D12 2-D16	4-D12 3-D12	3-D12 3-D14	300	300	3-D12 3-D20	3-D12	3-D12 3-D20	3-D12 1-D14	3-D12 1-D16	3-D12 1-D14
T6	400	350	4-D16 5-D12	3-D14	5-D16 3-D12	3-D16 2-D12	4-D14	3-D16 4-D12	400	350	5-D14 5-D14	3-D14	4-D14 4-D14	5-D16 3-D12	4-D14 2-D12	6-D16	350	350	3-D14 2-D22	3-D14	3-D14 2-D22	4-D14	4-D14	4-D14
T7	450	350	9-D14 9-D14	3-D14	5-D14 5-D14	5-D14 3-D12	3-D12 1-D16	4-D14 4-D12	400	300	4-D16 3-D14	3-D12	3-D16 3-D14	6-D12	4-D16 4-D16	5-D12	300	300	3-D12 2-D22	3-D12	3-D12 2-D22	3-D12	3-D12 1-D16	3-D12
T8	400	300	4-D12 5-D14	3-D14	5-D12 6-D14	5-D12 2-D14	3-D12 4-D16	5-D12 2-D14	350	300	5-D12 5-D12	3-D12	4-D12 4-D12	3-D12 1-D14	2-D14 3-D14	4-D14	300	300	3-D12 1-D16	3-D12	3-D12 1-D16	3-D12 1-D10	3-D12 4-D16	3-D12 1-D10
T9	300	300	3-D16 2-D12	3-D14	3-D16 2-D12	3-D16 2-D12	3-D14 1-D12	3-D16 2-D12	300	300	3-D14 2-D12	3-D14	3-D14 2-D12	3-D14	3-D16 3-D14	3-D14	300	300	3-D14 3-D22	3-D14	3-D14 3-D22	3-D14	3-D14 1-D12	3-D14
T10	350	300	3-D16 2-D14	3-D12	4-D16 4-D14	2-D16 2-D14	3-D12 3-D18	4-D16 3-D14	300	300	2-D12 3-D16	3-D12	2-D12 3-D16	3-D12 2-D14	2-D16 2-D12	3-D12 2-D14	300	300	3-D12 3-D24	3-D12	3-D12 3-D24	3-D12 3-D12	3-D12 3-D18	3-D12 3-D12
T11	300	300	3-D16 3-D14	3-D12	3-D16 3-D14	3-D16 2-D14	3-D12 3-D18	2-D16 2-D14	300	300	4-D14 3-D12	3-D12	4-D14 3-D12	3-D12 2-D20	2-D12 2-D14	3-D12 2-D10	300	300	3-D12 3-D25	3-D12	3-D12 3-D25	3-D12 2-D18	3-D12 3-D18	3-D12 2-D18
T12	350	350	4-D16 4-D14	3-D12	5-D16 4-D14	4-D16 3-D14	3-D12 3-D18	5-D16 4-D14	300	300	2-D16 2-D16	3-D12	2-D16 2-D16	3-D12 3-D16	3-D12 3-D12	3-D12 3-D16	300	300	3-D12 4-D22	3-D12	3-D12 4-D22	3-D12 3-D16	3-D12 3-D18	3-D12 3-D16
T13	350	350	3-D14 4-D16	3-D12	4-D14 4-D16	3-D14 3-D16	3-D14 3-D16	2-D14 3-D16	350	350	4-D12 4-D14	3-D14	4-D12 5-D14	3-D14 2-D12	4-D14 4-D16	3-D14 2-D12	350	350	3-D14 3-D25	3-D14	3-D14 3-D20	3-D14 2-D12	3-D14 3-D16	3-D14 2-D12
T14	400	350	5-D16 3-D12	3-D12	8-D16 5-D12	6-D16 4-D12	3-D12 3-D18	5-D16 3-D12	350	300	5-D12 5-D12	3-D12	4-D12 4-D12	3-D12 2-D12	3-D14 3-D14	3-D12 2-D12	300	300	3-D12 3-D24	3-D12	3-D12 3-D24	3-D12 2-D12	3-D12 3-D18	3-D12 2-D12
T15	350	350	3-D12 5-D16	3-D12	4-D12 6-D16	2-D12 3-D16	3-D12 1-D16	3-D12 4-D16	350	350	5-D12 5-D12	3-D12	5-D12 5-D12	3-D12	3-D12 3-D14	3-D12	300	300	3-D12 2-D16	3-D12	3-D12 2-D16	3-D12	3-D12 1-D16	3-D12
T16	400	350	5-D16 5-D16	3-D14	3-D16 3-D16	2-D16 1-D16	3-D12 1-D16	2-D16 1-D16	350	350	4-D12 5-D14	3-D12	3-D12 4-D14	5-D12	5-D12 4-D12	6-D12	300	300	3-D12 3-D16	3-D12	3-D12 3-D16	3-D12	3-D12 1-D16	3-D12
T17	350	350	7-D16 4-D12	3-D12	4-D16 3-D12	5-D16 3-D12	3-D12 1-D12	4-D16 3-D12	350	300	3-D14 3-D14	3-D12	3-D14 3-D14	3-D14	2-D14 3-D14	3-D12 1-D14	350	300	3-D12 4-D16	3-D12	3-D12 4-D16	3-D12	3-D12 1-D12	3-D12
T18	350	350	3-D16 3-D16	3-D12	3-D16 3-D16	3-D16 3-D16	3-D12 2-D16	3-D16 3-D16	350	350	5-D14 4-D12	3-D12	5-D14 4-D12	3-D12 1-D16	4-D14 4-D14	3-D12 1-D16	350	300	3-D12 3-D20	3-D12	3-D12 3-D20	3-D12 1-D16	3-D12 2-D16	3-D12 1-D16
T19	450	300	4-D14 4-D14	3-D14	6-D14 6-D14	3-D14 3-D14	3-D14 1-D12	4-D14 4-D14	450	350	5-D12 6-D14	3-D14	4-D12 6-D14	3-D14 3-D16	6-D14 2-D14	3-D14 2-D16	400	300	3-D14 3-D20	3-D14	3-D14 3-D20	3-D14 2-D16	3-D14 1-D12	3-D14 2-D16
T20	400	350	5-D14 6-D16	3-D14	4-D14 4-D16	3-D14 4-D16	3-D12 1-D16	2-D14 5-D16	400	350	4-D14 4-D14	3-D12	4-D14 4-D14	3-D14 3-D12	3-D12 3-D12	6-D12	300	300	3-D12 2-D25	3-D12	3-D12 2-D25	3-D12 1-D16	3-D12 1-D16	3-D12 1-D16
T21	450	350	5-D14 3-D16	2-D16	5-D14 3-D14	3-D14 7-D14	2-D16 3-D14	5-D14 5-D14	400	300	5-D14 4-D12	2-D16	5-D14 4-D12	2-D16 2-D14	5-D12 5-D12	2-D16 2-D14	350	300	2-D16 2-D22	2-D16	2-D16 2-D22	2-D16 3-D14	2-D16 3-D14	2-D16 3-D14
T22	500	400	5-D14 1-D16	3-D20	5-D14 1-D16	5-D14 2-D14	3-D20	5-D14 5-D14	500	350	4-D16 4-D16	2-D20	4-D16 4-D16	3-D14 2-D12	5-D12 1-D12	2-D20	450	300	2-D20	2-D20	2-D20	2-D20	2-D20	2-D20
T23	300	300	3-D16 3-D14	3-D12	3-D16 3-D14	2-D16 2-D14	3-D12 2-D24	2-D16 2-D14	300	300	2-D14 3-D16	3-D12	2-D14 3-D16	3-D12	2-D12 3-D14	3-D12	300	300	3-D12 4-D20	3-D12	3-D12 4-D20	3-D12	3-D12 2-D24	3-D12
T24	350	300	4-D12 5-D14	3-D12	4-D12 5-D14	3-D12 3-D14	3-D12 2-D24	3-D12 4-D14	300	300	3-D12 3-D14	3-D12	3-D12 3-D14	3-D12 1-D16	3-D12 2-D14	3-D12 2-D16	300	300	3-D12 3-D25	3-D12	3-D12 3-D25	3-D12 3-D14	3-D12 2-D24	3-D12 3-D14
T25	450	350	6-D16 4-D12	3-D14	8-D16 5-D12	8-D16 5-D12	3-D12 2-D24	7-D16 5-D12	400	300	5-D16 3-D12	3-D12	4-D16 3-D12	3-D12 2-D20	4-D12 3-D14	3-D12 2-D20	300	300	3-D12 5-D20	3-D12	3-D12 5-D20	3-D12 2-D20	3-D12 2-D24	3-D12 2-D20
T26	350	350	6-D12 3-D14	3-D12	5-D12 3-D14	3-D12 4-D14	3-D12 2-D24	5-D12 7-D14	300	300	2-D12 3-D16	3-D12	2-D12 3-D16	3-D12 2-D20	2-D14 2-D14	3-D12 2-D20	300	300	3-D12 5-D20	3-D12	3-D12 4-D20	3-D12 2-D20	3-D12 2-D24	3-D12 2-D20
T27	350	350	6-D16 5-D14	3-D12	4-D16 4-D14	3-D16 2-D14	3-D12 2-D24	4-D16 3-D14	350	350	4-D16 3-D12	3-D12	4-D16 3-D12	3-D12 2-D20	3-D12 4-D16	3-D12 2-D20	300	300	3-D12 5-D20	3-D12	3-D12 5-D20	3-D12 2-D20	3-D12 2-D24	3-D12 2-D20
T28	400	350	6-D12 6-D12	3-D14	7-D12 7-D12	7-D12 7-D12	3-D12 2-D20	7-D12 7-D12	350	300	4-D16 3-D12	3-D12	3-D16 2-D12	3-D12 2-D16	3-D12 3-D14	3-D12 2-D16	300	300	3-D12 3-D20	3-D12	3-D12 3-D20	3-D12 2-D16	3-D12 2-D20	3-D12 2-D16
T29	350	350		4-D16			4-D16		350	350		3D-14			4D-14		300	350		2D-14			4D-14	

T30	350	350	4-D16	3-D16	350	350	4D-16	3D-16	300	350	2D-20	2D-20
T31	350	350	4-D20	4-D20	350	350	4D-20	4D-20	300	350	2D-25	2D-25
T32	350	350	4-D20	4-D20	350	350	3D-25	2D-25	300	350	3D-25	2D-25
T33	350	400	3-D20	4-D20	350	350	2D-25	2D-25	300	350	2D-25	2D-25
T34	400	400	5-D16	5-D16	350	400	3D-16	3D-16	350	400	3D-16	3D-16
T35	400	400	3-D20	5-D20	350	400	4D-16	4D-16	400	400	2D-16	6D-16
T36	350	400	5-D20	3-D20	350	350	4D-20	3D-20	300	300	4D-20	2D-20
T37	350	400	3-D20	5-D20	350	350	2D-20	2D-20	300	300	2D-20	2D-20
T38	400	400	4-D20	4-D20	400	400	3D-16	5D-16	400	300	2D-16	6D-16
T39	450	400	5-D16	4-D20	450	400	2D-20	4D-20	450	300	2D-16	6D-16
T40	450	400	4-D25	4-D25	450	400	2D-25	2D-25	450	350	2D-25	2D-25
T41	450	450	4-D25	5-D25	450	450	3D-25	3D-25	450	400	2D-25	2D-25
T42	500	500	6-D25	4-D25	500	500	5D-25	5D-25	450	400	2D-25	6D-25