Structures 93 (2026) 112983

Contents lists available at ScienceDirect

Structures

journal homepage: www.elsevier.com/locate/structures

Design optimization of post–beam–panel mass timber frame systems using Monte Carlo Tree Search and policy neural network

Samia Zakir Sarothi, Hoang D. Nguyen, Qipei Mei *, Ying Hei Chui

Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Alberta, T6G 1H9, Canada

ARTICLE INFO

Keywords:
Mass timber frame system
Monte Carlo tree search
Policy neural network
Cost optimization
AI-assisted structural design

ABSTRACT

Mass timber construction offers significant sustainability benefits, but its adoption is limited by the high cost of engineered timber components. This study develops a Monte Carlo Tree Search with policy neural network (MCTS+NN) optimization framework to minimize the material cost of post-beam-panel mass timber systems under gravity loads. The proposed framework formulates the structural design process as a Markov Decision Process and integrates a policy neural network to guide the MCTS search. Its performance is evaluated using three building design scenarios together with a parametric study of the MCTS + NN algorithmic hyper-parameters, namely neural-network training frequency and exploration rate. The results indicate that more frequent neural-network training combined with lower exploration rates provides the best optimization performance. Under the reported experimental conditions and equivalent computational effort, the proposed MCTS + NN framework achieved higher objective values than conventional MCTS, demonstrating the potential of the proposed methodology for structural design optimization.

1. Introduction

With the rapid pace of urbanization and the increasing global demand for housing and infrastructure, it has become crucial to prioritize construction materials that exhibit a lower environmental impact than conventional, energy-intensive alternatives. Among the available options, timber has emerged as a promising sustainable material due to its environmental advantages and relatively low energy requirements throughout its life cycle [1]. When compared with traditional building materials like steel and concrete, the lightweight nature of wood simplifies handling, manufacturing, and transportation processes [2,3]. Recent advances in research and manufacturing technologies have led to the introduction of different engineered mass timber products in the construction industry. Mass timber (MT) refers to a family of engineered wood products designed to overcome the inherent limitations of solid sawn timber, including size restrictions, dimensional instability, and variability in material properties [4–6]. Owing to limitations in structural capacity and fire performance, conventional wood-frame buildings were historically restricted to low- and mid-rise construction, which is typically below eight stories [7]. However, the introduction and increasing acceptance of mass timber systems have significantly expanded the applicability of wood construction to high-rise commercial structures [8]. MT structural members consist of multiple wood components connected using adhesives and/or mechanical fasteners. Common types of mass timber products include cross-laminated timber (CLT), glue-laminated timber (glulam), dowel-laminated timber, and nail-laminated timber.

Recent advancements in timber technology and the growing adoption of wood-based construction have led to the emergence of several mass timber building systems [9]. For gravity load resisting systems, the post-beam-panel system is popular among designers (Fig. 1). This framing system consists of CLT floor panels supported by a structural framework of glulam beams and columns. This system can be applied to both hybrid material and full timber-based structures. Owing to its adaptable grid layout and modularity, this system offers enhanced design flexibility, making it particularly suitable for applications requiring open floor plans and adaptable spatial arrangements, such as corporate offices, high-end residential buildings, multifamily housing, commercial facilities, and public buildings.

Despite its numerous environmental and structural advantages, the global adoption of mass timber construction has progressed relatively slowly, primarily due to higher project costs [10]. Ahmed and Arocho [11] reported that timber buildings can be 6.43% more expensive than their concrete counterparts, with the elevated cost of engineered wood

* Corresponding author.
E-mail address: qipei.mei@ualberta.ca (Q. Mei).

https://doi.org/10.1016/j.istruc.2026.112983
Received 22 May 2026; Received in revised form 5 August 2026; Accepted 2 September 2026
Available online 9 September 2026

Fig. 1. Typical layout of post-beam-panel system (adapted from [9]).

identified as the principal factor. Similarly, Burback and Pei [12] found that, for single-family homes in the U.S., constructing using CLT can increase costs by up to 23% compared to conventional light-frame wood systems. One major reason for this cost disparity is that mass timber panels typically require about three times more wood material than traditional wood-frame construction, thereby leading to significantly higher material expenses [13]. A notable example is the Murray Grove Building in the United Kingdom, where the use of cross-laminated timber (CLT) resulted in an estimated 30% increase in material costs relative to reinforced concrete alternatives [14]. To enhance the economic feasibility and promote the wider adoption of this sustainable building system, structural design optimization presents a promising strategy to minimize costs while maintaining performance and competitiveness within the construction industry.

Structural optimization primarily aims to minimize the overall weight or material cost of a structure while ensuring compliance with relevant performance standards and design codes [15]. In structural design, the large number of interdependent design variables and the extensive search space associated with each parameter can result in millions of potential design combinations for a single building case. Identifying the optimal solution within this vast design space- while satisfying all structural performance requirements within a practical computational timeframe- poses a significant challenge. Traditional mathematical and metaheuristic optimization techniques have been widely applied to timber structure optimization problems [16–23]. These algorithms are highly dependent on the choice of initial solutions and tuning parameters, which are typically determined through trial and error, potentially compromising optimization reliability and efficiency [24,25]. However, in recent years, the integration of artificial intelligence (AI) into structural design optimization has gained increasing attention [26–35]. Studies have demonstrated the considerable potential of AI to enhance the productivity of experienced engineers, assist less experienced designers, and promote effective knowledge transfer within engineering teams [36–41].

One prominent AI technique is Monte Carlo Tree Search (MCTS), a probabilistic, heuristic-driven search algorithm that merges traditional tree search methods with reinforcement learning principles. MCTS incrementally constructs an asymmetric search tree based on simulation outcomes obtained through random sampling [42]. Its ability to perform well with minimal domain-specific input makes it suitable for navigating large and complex search spaces [43]. Unlike supervised and unsupervised learning, data in MCTS are generated during the interaction between the agent and the environment instead of prepared datasets, offering greater flexibility. Originally popularized in board games and video games, MCTS gained international attention when it was used in AlphaGo [44] and its successors [45,46], becoming the first AI to defeat a professional Go player in 2016, a major milestone in AI, especially given Go's immense solution space of 3^{361} possible board states. A limitation of MCTS is its built-in randomness, arising from the exploration–exploitation trade-off. While this randomness supports broader search, it can occasionally cause the algorithm to deviate significantly from the true optimum, especially in problems with large or highly sensitive search spaces [36,47].

The first version of AlphaGo, known as AlphaGo Fan [44], integrated MCTS with two deep neural networks: a policy neural network, which generated move probabilities, and a value network, which estimated the expected outcome of a given position. Before its integration with MCTS, the policy neural network was trained using data derived from expert human data. Subsequently, an improved version called AlphaGo Zero [45] was introduced, which eliminated the reliance on human expert data. Instead, the neural network was trained entirely through self-play, starting from random moves. In this approach, the network initially produced random actions, and successful outcomes from self-play were then used to train both the policy and value networks in an iterative loop.

Inspired by the success of AlphaGo, this study presents a material cost optimization framework for post-beam-panel type mass timber building design based on MCTS integrated with a neural network (MCTS+NN). The neural network is incorporated into the original MCTS to guide the search toward more promising design decisions, thereby reducing the excessive randomness of pure MCTS and improving the efficiency of exploring the design space by learning from the design data generated throughout the search process. A similar principle of self-play data generation is employed, eliminating the need for pre-training with expert-engineered design data. However, unlike AlphaGo Zero–style methods that integrate both policy and value networks, this work employs only the policy neural network within MCTS. The value network, which in other domains predicts the expected reward of an intermediate state, was deliberately excluded. In engineering design, the validity of a solution cannot be reliably inferred from partial states; all design components must be specified and evaluated against structural codes. Relying on a learned value approximation risks misclassifying infeasible designs as valid. By contrast, the policy neural network can safely guide action selection, while full rollouts are always executed and terminal states are rigorously evaluated using deterministic, code-based checks. The primary objective of this study is to demonstrate the applicability of the MCTS + NN framework to mass timber structural design optimization while simplifying certain aspects of structural analysis and performance evaluation. To maintain computational efficiency and focus on the optimization process, the current study considers only a single-story post–beam–panel framing system subjected to gravity loads. This simplification enables faster design iterations and establishes a foundation for future work that can incorporate more complex load cases. Another key contribution of this study is the formulation of the mass timber design process as a Markov Decision Process (MDP), enabling reinforcement learning–based optimization and effectively addressing the limitations of conventional methods in managing large-scale design problems with extensive search spaces.

A key contribution of this study is the development of a methodology for integrating MCTS + NN into structural design optimization. While the proposed framework is demonstrated using a simplified mass timber benchmark, the underlying methodology is general and can be adapted to more complex timber structures as well as other structural design problems with appropriate modifications to the structural analysis model. Recent studies have successfully combined MCTS and reinforcement learning for truss optimization, including AlphaTruss and AutoTruss. While these methods focus on topology, geometry, and member-size optimization of truss systems, the present study addresses the sequential design optimization of post-beam-panel mass timber buildings. Furthermore, unlike AutoTruss, which performs MCTS and reinforcement learning in two separate stages, the proposed framework

integrates policy learning directly into the MCTS search process. The policy network is trained online during optimization using successful search trajectories generated by MCTS, and the updated policy immediately guides subsequent tree expansion and action selection throughout the search. The proposed framework adopts an online learning strategy in which the policy network is trained during the optimization of each design case using the successful search trajectories generated by MCTS. Consequently, the learned policy is specialized for the current optimization problem rather than intended to be transferable across different structural configurations. Developing a transferable policy would require offline training on a large and diverse dataset of structural designs, which is beyond the scope of the present study. The proposed model addresses the key design components of mass timber structures, including grid layout, floor system configuration, beam sizing, and column selection. Before initiating the MCTS procedure, the mass timber building design problem is formulated as a Markov Decision Process (MDP) to enable sequential decision-making within a structured optimization framework.

The remainder of this paper is organized as follows: Section 2 presents the details of the mass timber frame design considered in this study. Section 3 explains how the mass timber design problem is formulated as an MDP. Section 4 provides an overview of both the MCTS + NN and basic MCTS methods, along with their applications to design optimization. Section 5 discusses and compares the results obtained from three design cases between MCTS and MCTS + NN. In addition, a parametric study of the MCTS + NN approach is conducted to recommend suitable parameter settings for the search process. Finally, Section 6 concludes the study by summarizing the key findings and suggesting potential directions for future research.

2. Structural design framework

In this study, the mass timber building is designed based on eight initial input design parameters: total span in the x direction (L_x), total span in the y direction (L_y), length of the CLT panel (L_{clt}), length of the glulam (L_{glt}), superimposed dead load per unit area (dl), live load per unit area (ll), floor height (ht), and species of the mass timber products. Two glulam species are considered in this study, namely Douglas Fir-Larch (DFL) and Spruce-Pine (SP). In a particular design solution, the same species is used for all the beams and columns to ensure aesthetic consistency.

The overall structural system, including floor panels, beams, and columns, can be idealized as a three-dimensional (3D) frame. However, to simplify computation and reduce complexity, the current study models the system using a two-dimensional (2D) frame analysis approach. This simplification is justified as the building layout is symmetrical and free from torsional irregularities. The beam-to-column connections are assumed to behave as pinned joints, allowing rotation but not transmitting bending moments. The structural analysis is performed using the Python package IndeterminateBeam [48].

The post-beam-panel system considered in this study is formed with CLT floor panels supported by glulam beams and columns. Two types of floor layouts (i.e., with and without the secondary beams) are investigated based on the floor vibration checks, as presented in Fig. 2. The selection between these two floor systems is determined based on the check of the vibration requirement of the floor panel with the span in the shorter direction. In case the vibration limit of the CLT floor, l_v is greater than the span length in the short direction, s_y, the floor arrangement follows the configuration shown in Fig. 2a, which consists of a floor panel, primary beam, and column. On the other hand, if l_v is smaller than the span length (s_y), the floor vibration is controlled by adding the secondary beams and supporting girders along with the elements mentioned in Fig. 2a (Fig. 2b). The secondary beams are equally spaced within the floor span. For uniformity, the same section size is used for each type of member (e.g., all the primary beams have the same section size).

In this study, the analysis focuses solely on gravity load. All the structural components are evaluated for ultimate and serviceability limits in accordance with the timber design standard of Canada CSA O86:24 [49], National Building Code of Canada 2020 (NBCC 2020) [50], and the Wood Design Manual [51]. In addition to the limit state verifications, several supplementary checks are incorporated in accordance with NBCC 2020 and recommendations from industry practitioners. The full description of the structural design framework and material cost computation is documented in a previous article by the authors [47]. This section provides only a condensed overview.

3. Formulating the design problem as MDP

Before initiating the search process for mass timber building design, the problem is formulated as an MDP to facilitate sequential decision-making throughout the design stages. An MDP serves as a mathematical framework for representing decision-making scenarios in which the results depend both on the actions of the decision-maker and on probabilistic events. It offers a structured approach to model sequential decision problems that occur within uncertain or stochastic environments. In this framework, the building design process is conceptualized as a sequence of interdependent decisions, where each choice affects the following stages and collectively determines the overall performance of the building. Fig. 3 illustrates the formulation of the MDP for the mass timber building design problem. At each time step t, the agent observes the current state S_t and selects an action, a_t, which transitions the system

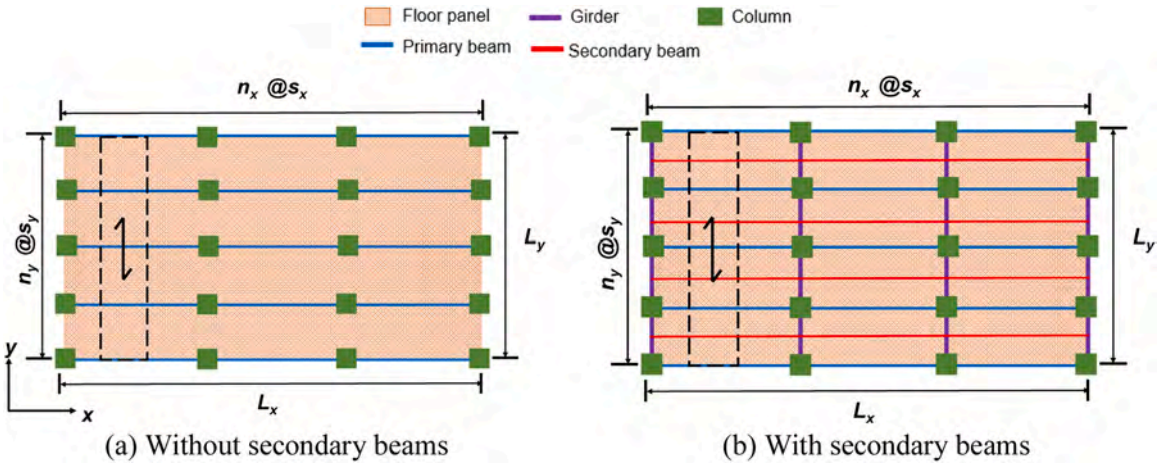


Fig. 2. Types of floor layout for the post-panel-beam system.

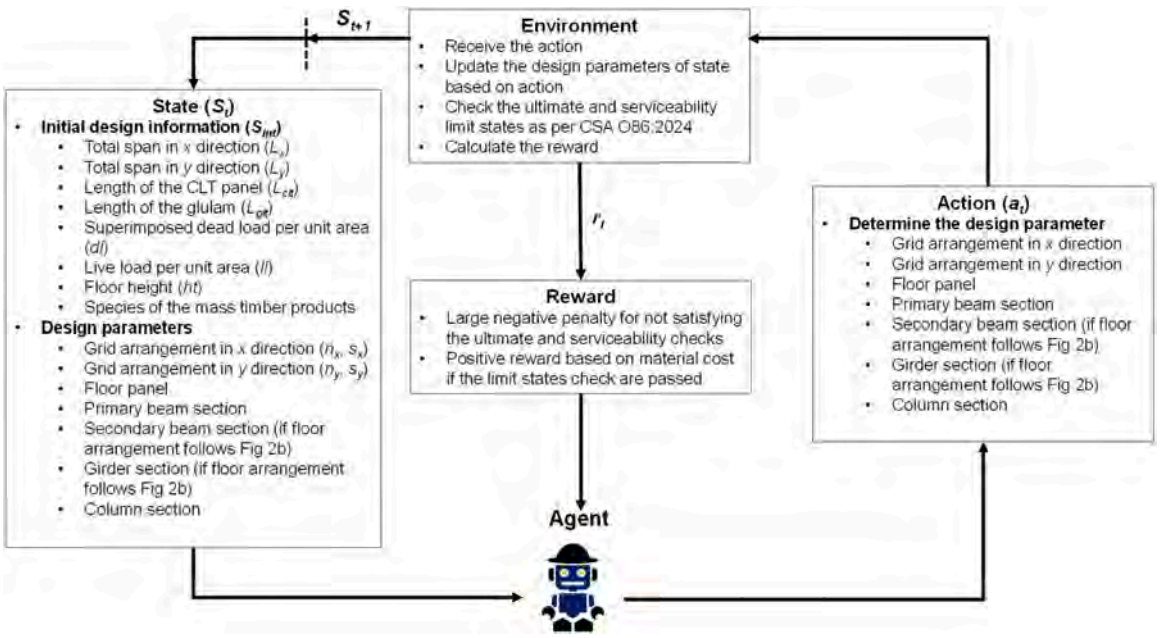


Fig. 3. Formulating the mass timber building design process as MDP.

from S_t to S_{t+1}. This process continues iteratively until all necessary design parameters are determined, reaching what is defined as the terminal state or complete design. Upon reaching the terminal state, the agent receives a reward r_t that reflects the performance of the final design in terms of compliance with structural limit states and overall material cost. The objective of the MCTS + NN algorithm is to identify the optimal sequence of actions that minimizes material cost for a given initial state based on how the design solution complies with the limit states and material cost for a given initial state (S_{int}) in the MDP framework. A complete description of the MDP formulation, which covers the state representation, action space, and reward design, has been provided in the literature [47]. To avoid repetition, this paper only presents a concise overview. In this study, two different rewards were adopted: positive and negative, as presented in 1a and 1b. When the required design criteria are fulfilled, a positive reward, R_{pos} (Eq. 1a) is provided to the agent. On the other hand, if one of the limit states fails to be satisfied, a large negative penalty, R_{neg} (Eq. 1b) is assigned. While deciding on the cost efficiency of the design solution, structural designers tend to check the cost per unit area mostly which motivated the reward to be based on cost per unit area (Eq. 1a). When formulating the positive reward (Eq. 1a), it was observed that scaling the inverse of the cost by a large constant encourages the agent to converge toward better solutions.

$$R_{pos} = \frac{1}{Cost/(L_x * L_y)} * 10^9 \tag{1a}$$

$$R_{neg} = -10^6 \tag{1b}$$

4. Methodology

4.1. Monte Carlo Tree Search

The MCTS + NN framework is an enhanced version of the basic MCTS algorithm. This section provides a brief introduction to the core concept of basic MCTS. MCTS is an iterative, guided, random best-first tree search method that systemically searches a space of possible solutions to obtain an optimal solution in decision-making problems [52]. The MCTS method begins with a search tree having only an initial root node built from the given initial state (S_{int}). Subsequently, an iterative analysis is performed, expanding the search tree until the allotted search time is terminated. Each iteration consists of four steps [53], including selection, expansion, simulation, and backpropagation (Fig. 4). As shown in Fig. 4, directed edges represent actions taken, while the nodes correspond to the current states in the search process, which may be either intermediate or terminal.

- Selection: Starting with the root node (or initial state, S_{int}), the MCTS algorithm traverses down the tree until it reaches a leaf node, a node that either has not been fully expanded or represents a terminal state. At each step during this traversal, the algorithm must decide which child node to descend into based on the Upper Confidence Bound for Trees (UCT) policy [53] as defined in Eq. 2. In Eq. 2, w_i is the total reward of node i and n_i is the number of times node i has been visited. N represents the total number of visits to the parent node. C is the exploration parameter that controls the balance between exploration and exploitation. A higher value of C indicates more exploration. In this study, $C = 1.4$ is used as recommended by Auer et al. [54]. The first term in Eq. 2, $\frac{w_i}{n_i}$ promotes exploitation, choosing the node with a high average reward. The next term, $C\sqrt{\frac{\ln N}{n_i}}$ encourages exploration of less-visited nodes. In case of unvisited nodes ($n_i = 0$), an infinite UCT value is assigned to ensure that it is selected at least once.

$$UCT_i = \begin{cases} \infty if\ n_i = 0 \\ \frac{w_i}{n_i} + C\sqrt{\frac{\ln N}{n_i}}\ otherwise \end{cases} \tag{2}$$

- **Expansion**: When the MCTS process reaches a node that has not been expanded before, the process ends the selection phase and moves to the expansion stage. During this phase, one or more child nodes are added by randomly selecting an available action. This selected action generates a new node, which is then appended to the search tree.
- **Simulation**: From the newly expanded node, the algorithm proceeds to the simulation phase, where a random sequence of actions is executed until the terminal state (S_{ter}) is reached, marking the completion of the design or decision process. Once this state is obtained, the simulation outcome is evaluated and represented as a

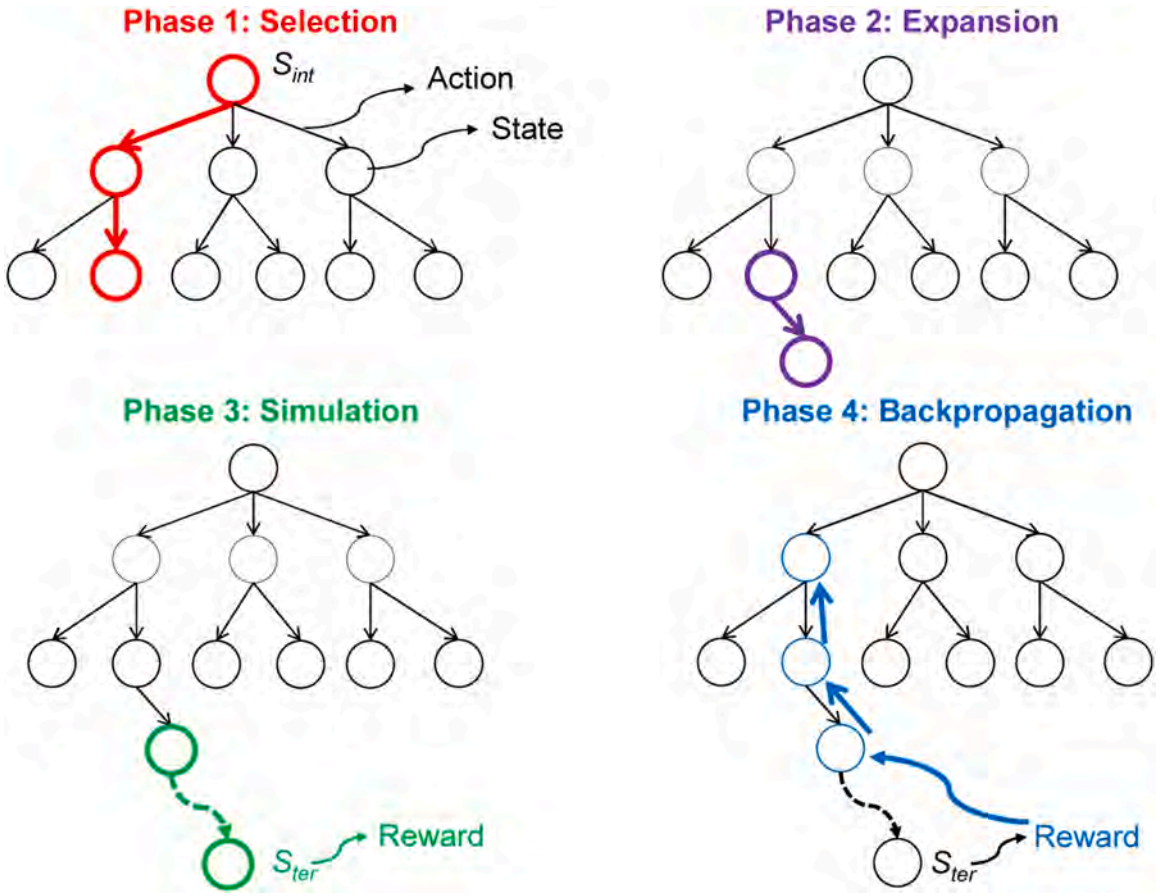


Fig. 4. Four phases of the basic MCTS method.

reward value, which is then used in subsequent phases to guide and improve future decision-making within the search tree.

- **Backpropagation**: In the backpropagation phase, the reward obtained at the terminal state is propagated upward from the leaf node to the root, updating the number of visits and the estimated value of each node along the traversal path.

4.2. Monte Carlo Tree Search with policy neural network integration

In this study, the proposed methodology aims to integrate MCTS with a policy neural network (MCTS+NN) to optimize mass timber building design configurations. MCTS, which is presented in Section 4.1, serves as the search backbone, exploring discrete decision sequences that represent valid building designs, while the policy neural network biases action selection towards promising choices based on prior experience. In the standard MCTS procedure, all untried actions are initially explored with equal priority, leading to a significant portion of iterations being spent on suboptimal or infeasible regions of the search space. In the proposed MCTS + NN approach, a policy neural network is trained in parallel with the search to model the relationship between design states and promising actions, using data collected from prior successful simulations.

Fig. 5 explains the framework of the MCTS + NN method. The workflow begins with an initial phase in which the search runs as plain MCTS for a fixed number of iterations: Selection traverses the tree using UCT, *Expansion* and *Simulation* choose actions uniformly at random, and completed designs are evaluated by structural checks with a cost-based reward (1a and 1b). The policy neural network $\pi\theta$ is uninitialized at the

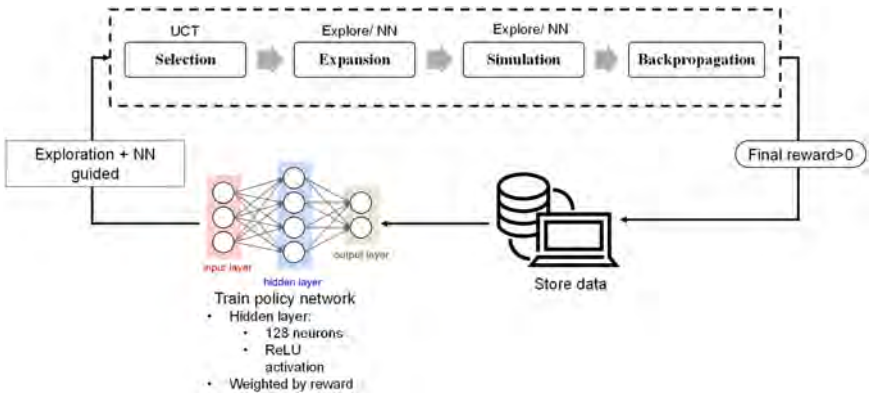


Fig. 5. Workflow of the proposed MCTS + NN framework.

beginning phase and is updated periodically according to a training interval *K*. After this phase, all state-action pairs along trajectories that achieved a positive reward- including both intermediate and terminal states- are stored and periodically used to train the policy network (a feed-forward MLP with one hidden layer of 128 ReLU units) via reward-weighted cross-entropy. In the subsequent MCTS + NN phase, *Selection* remains UCT-based, while *Expansion* and *Simulation* become policy-guided: at each decision point, the action is sampled from the network's probabilities over the feasible set with epsilon-greedy (ε-greedy) randomness to avoid premature convergence. In this study, two types of ε-greedy randomness, namely fixed-exploration setting and decaying-exploration setting, are investigated. The ε-greedy is parameterized by ε_o, $\varepsilon_{\min}$, and $\Delta\eta$, which indicate initial exploration rate, minimum exploration rate, and exploration decay, respectively. During the fixed exploration setting ε_o is used throughout the search. On the other hand, for the decay settings exploration is initially ε_o and with every NN training, it reduces by $\Delta\eta$, but never falls beyond $\varepsilon_{\min}$. Each iteration is again scored by the structural checks and cost objective; positive-reward trajectories are stored and used in the next training update. This loop of search, store successful experience, train, and reuse the policy progressively focuses exploration on promising regions of the design space while retaining exact physics-based evaluation.

The policy network architecture was intentionally kept fixed throughout the study to isolate the influence of the proposed MCTS + NN algorithmic hyperparameters on optimization performance. A simple feed-forward multilayer perceptron with ReLU activation was adopted following established deep learning practices [55], while the Adam optimizer was employed because of its effectiveness and computational efficiency in training deep neural networks [56].

Regarding the policy neural network ($\pi\theta$) training, an input dimension of 15 is used, which includes both the initial data and design parameters. The rewards (R_i) corresponding to each state action pair was turned into a weight vector (w_i) as per 3 and 4. To prevent a few very large returns from dominating the gradient while ensuring every positive example contributes, rewards are normalized and rescaled to [0.1, 1.0], applying 3 and 4 with a small $\epsilon > 0$ for numerical stability. The glulam beam action space has the highest count (714) of options compared to the other two lists (CLT and glulam column), and 714 is used as the output size of the NN. The NN is trained for 100 epochs with the Adam optimizer with a learning rate of 10^{-3}. The objective of the training is to minimize the reward-weighted average loss ($\mathscr{L}$) as represented by Eq. 5, where B is the batch size (64), x_i is the encoded state before the action and y_i is the corresponding action label. $CE(\pi_\theta(x_i), y_i)$ denotes the per-sample cross-entropy and $\pi_\theta(x_i)$ are the logits over all actions.

After the first policy training, the decision is made randomly or based on the trained policy during the expansion and simulation phases. For a given state (S), the policy neural network outputs logits corresponding to all 714 actions within the global action space. These logits are temperature-scaled (T) and passed through a Softmax function to generate an initial probability distribution (Eq. 6). The resulting probabilities are then restricted to the subset of feasible actions (A), ensuring that only valid design decisions are considered for the current state. Eq. 6 explains the probability of selecting action (a) given state (S). If any option, a, within the 714 action list does not fall under the feasible action list (A), probability of 0 is assigned to it. Otherwise probability will be calculated again by updating the distribution among the feasible options only (Eq. 6). In Eq. 6, z_a and z_b denote the logits corresponding to actions a and b, respectively. Temperature (T) controls the trade-off between exploration and exploitation within the policy. Higher T (T>1) means more variety in chosen action (flatter probability). Lower T (T<1) means a stronger preference for top-scored actions (more deterministic; mass concentrates on the largest logit). Temperature, T= 3.0, is used in this study. Also, the Softmax function gives us proper probabilities to sample actions stochastically (instead of always taking the single best score). While using the trained NN to make a decision, the action with the highest probability is picked.

$$\widetilde{R}_i = \frac{R_i - R_{\min}}{R_{\max} - R_{\min} + \epsilon} \tag{3}$$

$$w_i = 0.1 + 0.9\widetilde{R}_i \tag{4}$$

$$\mathscr{L} = \frac{1}{B}\sum_{i=1}^{B} w_i CE(\pi_\theta(x_i), y_i) \tag{5}$$

$$p_A\left(\frac{a}{S}\right) = \begin{cases} \dfrac{\exp(z_a/T)}{\sum_{b\epsilon A}\exp(\frac{z_b}{T})} & \text{if } a\epsilon A \\ 0 \quad \text{if } a \notin A \end{cases} \tag{6}$$

4.3. Incorporation of MCTS into the timber design problem

In this study, the results of MCTS + NN are compared with those from MCTS. Therefore, the procedure of incorporating basic MCTS into the timber design problem is also presented. Accordingly, Fig. 6 explains how the MCTS is adopted to solve the sequential decision problem of mass timber building design based on the information introduced in Section 2. The MCTS search focuses on finding an optimum design solution for a given design case of a mass timber building, which satisfies all the limit states and at the same time minimizes material cost. The root node represents the initial design data, S_{int} (Fig. 6). Any other node on the search tree indicates a partial or completed design solution. Fig. 6 shows how both the design approach (with and without a secondary beam) is adopted in the MCTS process.

The cells with dashed boxes (Fig. 6) indicate the selection step where the algorithm picks an action based on UCT. It is assumed that the node (S_{int}, grid x, and grid y) has no child node. Consequently, the algorithm goes to the expansion phase to get the next dotted box cell (S_{int}, grid x, grid y, floor), as presented in Fig. 6. After expansion, the process transfers to the simulation phase (solid box) and the algorithm keeps on taking the rest of the action (building components) until it reaches the terminal state, S_{ter}, which is a completed design solution. The completed design is evaluated based on the condition described in Section 3 and assigned R_{pos} (Eq. 1a) or R_{neg} (Eq. 1b) reward (marked as a star in Fig. 6). The action space for the beam and column is filtered, as described in previous literature [47]. In the final phase, the received reward is backpropagated to all the intermediate states along the design path, which is described as putting the star in all along the path in Fig. 6. In the initial iterations, when the search tree contains only a few expanded nodes, the algorithm begins directly with the expansion phase rather than the selection phase. As more nodes become available, the process transitions to performing selection steps. During the early stages, the algorithm emphasizes exploration through frequent expansion and simulation to adequately cover the search space. In later iterations, once the tree has been extensively explored, the algorithm shifts its focus toward selection, thereby exploiting the knowledge accumulated from previous searches.

4.4. Incorporation of MCTS + NN into the timber design problem

Fig. 7 presents the implementation of MCTS + NN in mass timber building design. Similar to basic MCTS, the process starts from the root node, which is defined as per the initial state (S_{int}). Fig. 7 shows how a typical iteration starts in the MCTS + NN method when the policy neural network is ready to use. The first two design decisions (grid x and grid y) are decided in the selection process (dashed box), and the selection is UCT-based. The node (S_{int}, grid x, and grid y) has no child node herein, and consequently, the process moves to the expansion phase. During this step, the trained NN is called, and it provides a probability distribution over feasible floor sections (π_{floor}). The floor section is then

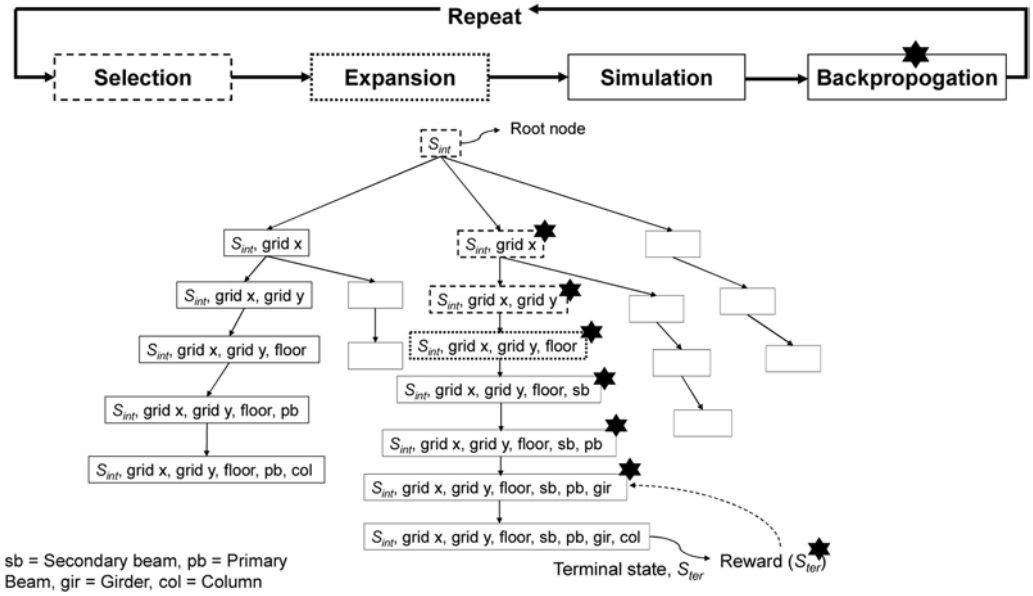


Fig. 6. MCTS-based mass timber building design process.

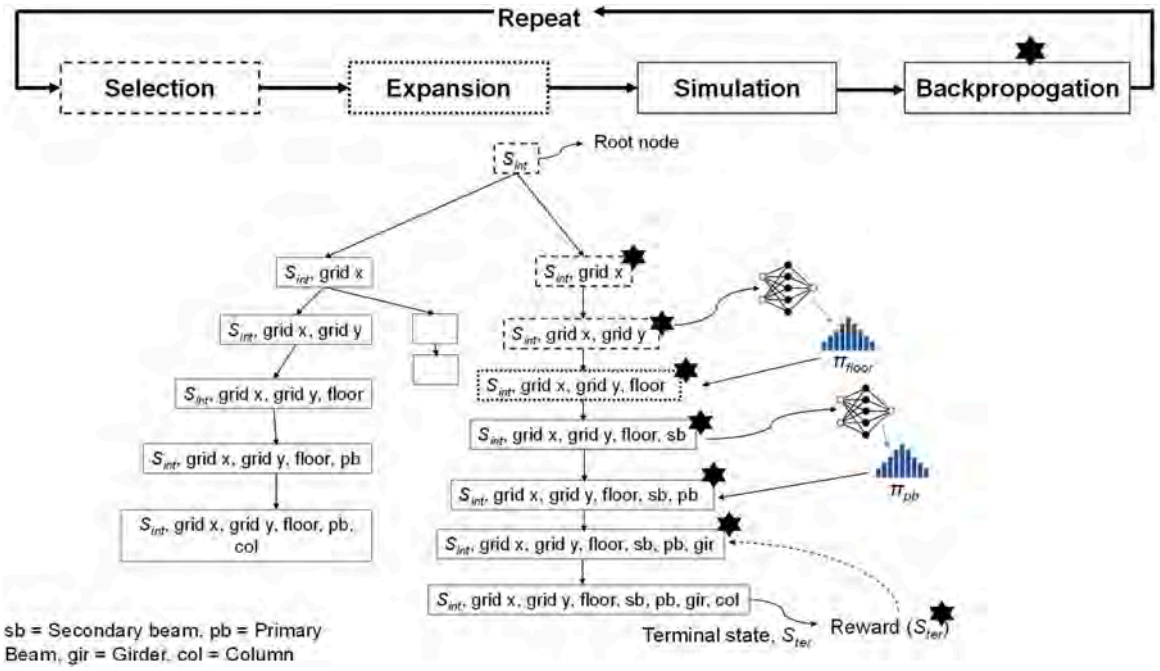


Fig. 7. MCTS + NN-based mass timber building design process.

chosen deterministically as the feasible option with the highest network probability. After the expansion, the process moves to the simulation phase, and the first action, secondary beam (*sb*), is selected randomly herein. The next action, primary beam, is selected based on the trained policy. Again, the policy returns a probability distribution over all feasible primary beam sections (π_{pb}) and the highest-probability feasible section is picked. The subsequent actions (girder and column) are decided randomly, similar to the basic MCTS. In the end, the terminal state or completed design outcome is evaluated based on heuristics, and a reward is assigned. The reward also gets backpropagated to all the nodes along the path.

Table 1
Details of the design cases.

Case	L_x (m)	L_y (m)	*dl* (kPa)	*ll* (kPa)	L_{clt} (m)	L_{glt} (m)	h_t (m)	Species
Case 1	30	20	4	5	12	10	3	SP
Case 2	25	45	4	5	12	10	3	SP
Case 3	45	55	4	5	12	10	3	DFL

5. Performance of MCTS + NN in design case studies

Three different design cases, as presented in Table 1, are analyzed in this study to illustrate the performance of the proposed MCTS + NN approach. The design cases cover different building spans and species of wood. The span of the buildings (Table 1) is decided based on actual

mass timber building projects [57–61]. The typical length of CLT panels is around 12 m [62]. The length of glulam members, on the other hand, largely depends on the manufacturer's production capability [62]. In this study, the glulam length of 10 m is considered. The floor height (h_t) is taken as 3 m. In addition, the analyses are conducted considering the building to be single story.

5.1. Parametric study of the MCTS + NN method

In this section, a parametric study is conducted on the training frequency of NN (K) and exploration rate (ε) in the MCTS + NN method. It should be noted that this parametric study focuses on the algorithmic hyperparameters of the proposed MCTS + NN framework rather than the architectural hyperparameters of the neural network, which remain fixed throughout the study. For the design cases mentioned in Table 1, the MCTS + NN algorithm is executed for 3000 iterations, considering both the varying training frequency and exploration rate. Three different training intervals (K) of 200, 400, and 800 are considered in this study. For the exploration rate, two types, namely fixed rate of exploration and exploration with decay (diff%), are examined as mentioned in Section 4.2. For the fixed exploration rate, 50% and 70% explorations are investigated. Specifically, 50% exploration indicates that after the policy neural network is trained, 50% of the time the decision is based on the NN, and for the remaining 50%, the action is picked randomly. Meanwhile, for a 70% rate of exploration, decisions are 70% NN-based and 30% of the time random. In addition, an exploration with decay was also investigated. Specifically, the exploration rate is initially 70% ε_o (after the first NN training) and with subsequent NN training, it goes down by 5% ($\Delta\eta$). However, there is always a minimum of 10% ($\varepsilon_{\min}$) of exploration involved.

Fig. 8 presents the results of MCTS + NN for Case 1 across three different training frequencies and exploration rates. The successful design solutions are plotted in Fig. 8 with 3000 iterations, and the rewards are normalized to make the comparison reasonable. In this case, the rewards are normalized by dividing them by the average reward of the first phase, where the search is based on basic MCTS (e.g., with a training frequency of 200, the valid designs obtained from the first 200 iterations are used to train the neural network). Each of the plots in Fig. 8 has vertical lines indicating when the NN is trained. In addition, a moving average line is shown in Fig. 8 (solid black line) for each of the plots to observe the trend of changing rewards. In this case, the plots are compared based on the value of reward in the plateau (concentration of points in a line). Figs. 8a, 8b, and 8c represent the result for atraining frequency of 200 with 50%, 70% and diff% exploration rates, respectively. In terms of normalized reward value at the plateau, diff% returns the best result of 1.96, as shown in Fig. 8c. It is observed from Figs. 8a to 8c that the reward continues on improving (as observed from the scatter and moving average line) with the NN training before reaching the plateau. The improvement is most significant for diff% (Fig. 8c). At the beginning of the search, the normalized reward is around 1, and by iteration 1150, it shows a 96% increase. Figs. 8d to 8f show the outcomes with 400 training frequency. It can be seen that 70% exploration returns the best result of 1.81 in terms of the reward value at the plateau (Fig. 8e). On the other hand, 50% and diff% exploration rates give 1.2 (Figs. 8d) and 1.22 (Fig. 8f) reward during convergence. Figs. 8g to 8i show the cases with a training frequency of 800 for varying exploration rate. It is found that 50% exploration rate achieves the best result (1.48), as shown in Fig. 8g. In the case of a 70% exploration rate (Fig. 8h), the curves converge during 550 iterations even before the first NN training. This indicates that the MCTS process converges before the NN training occurs, thereby diminishing the benefit of incorporating the NN. Similarly, for 50% (Fig. 8g) and diff% (Fig. 8i) exploration rate, the curves converge at 900 and 850 iterations, which is right after the first NN training. This outcome implies that, with the current MCTS parameter setup and number of iterations used, late NN training (i.e., 400 and 800 training frequency) is not recommended. Overall, the observation from Fig. 8 indicates that more frequent NN training (every 200 iterations) yields the most effective performance for Case 1. Regarding the exploration rate, the diff% exploration rate, along with frequent NN training, is well-suited in Case 1.

Similar parametric studies for Case 2 and Case 3 are reported in Figs. A.1 and A.2 in Appendix A, respectively. For Case 2, a training frequency of 200 with 50% exploration yields the best reward value of 1.83 in the plateau (Fig. A.1a). At the beginning of the search, the normalized reward is around 0.66, which showed a 177% increase by iteration 1100 (Fig. A.1a). Among Figs A.1a to A.1c, Fig A.1a shows better improvement with every NN training as indicated by the scatter points and moving average curve. Regarding the 400 training interval (Figs A.1d to A.1 f), a 70% exploration rate achieves the best reward value of 1.58 during convergence. For 800 training frequency and 50% exploration rate (Fig. A.1 g), the plateau is observed at 800. Meanwhile, it is 800 and 1100 iterations for 70% (Fig A.1 h) and diff% (Fig A.1i) exploration, respectively. This finding again implies early convergence of the MCTS itself, and it does not benefit from the NN training. In summary, for Case 2, more frequent NN training (200) performs better than other training intervals (400 and 800). As shown in Fig. A.2c, Case 3 also observes the best performance for 200 training frequency, which results in a normalized reward value of 2.17 for diff% exploration. Figs. A.2 g to A.2i present the outcomes for an 800 training interval, and it is found that convergence occurs at 1300 (Figs. A.2 g and A.2 h) and 1900 iterations (Fig A.2i). It indicates the plateau is reached after the first (Figs. A.2 g and A.2 h) and second (Fig. A.2i) NN training, which implies not enough leverage of the NN.

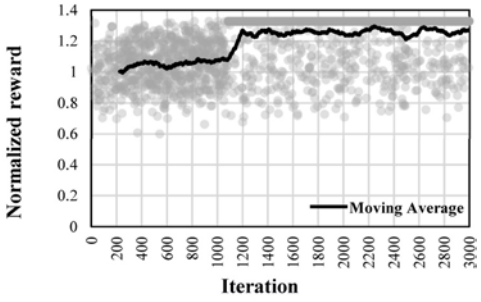
Overall, the results indicate that, with the current MCTS parameter settings and a total of 3000 iterations, adopting more frequent neural network training (every 200 iterations) along with reduced exploration (50% and diff%) yields better rewards and improved design outcomes. For delayed training frequencies (400 and 800 iterations), achieving comparable performance would require adjusting the base MCTS configuration by increasing the exploration rate and extending the total number of iterations.

5.2. Comparative study between MCTS + NN and basic MCTS

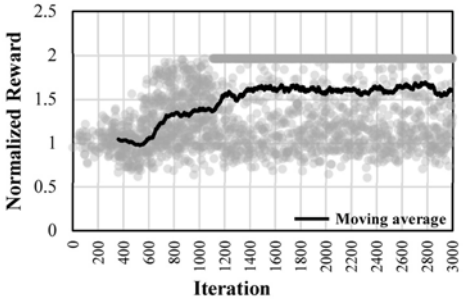
To demonstrate the performance of MCTS + NN, this section compares the best results obtained from the MCTS + NN and MCTS approaches in terms of the achieved reward. Table 2 summarizes the highest rewards for each approach, along with the corresponding improvement of MCTS + NN over basic MCTS across the three design cases. The results for different training frequencies and exploration rates are denoted as "*MCTS + NN_training interval_exploration rate*" (Table 2). Both methods were executed for 3000 iterations, and the best reward values identified during the search are reported here. It should be noted that the rewards presented in this section are the maximum rewards observed across all iterations, not the rewards at the plateau. Due to the stochastic nature of the process and the introduction of randomness, occasional outliers may appear, producing rewards higher than the plateau values. The primary objective of this section is to compare the overall performance of MCTS + NN and basic MCTS with 3000 iterations, rather than analyzing how neural network training influences convergence over iterations. All MCTS and MCTS + NN computations were carried out on a PC featuring an AMD Ryzen 9 5900 12-Core Processor @ 3.00 GHz and 32 GB RAM. On average, approximately 30,000 s is required to complete 3000 MCTS iterations across the different design cases. On the other hand, MCTS + NN required approximately 42,900 s to complete 3000 iterations. It is noted that this study mostly focuses on MCTS + NN and its improvement compared to basic MCTS. The authors have a separate publication [47] showing the efficiency of MCTS over brute force and another optimization technique, namely the genetic algorithm.

For Case 1, the configuration MCTS + NN_200_diff% achieved the highest objective value, corresponding to an observed improvement of

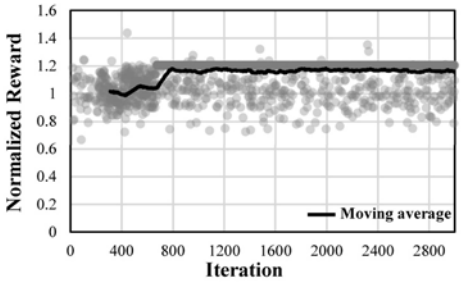
(a) 200 training frequency, 50% exploration



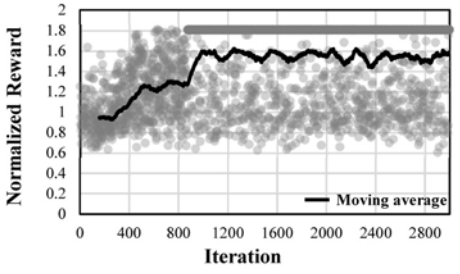
(b) 200 training frequency, 70% exploration



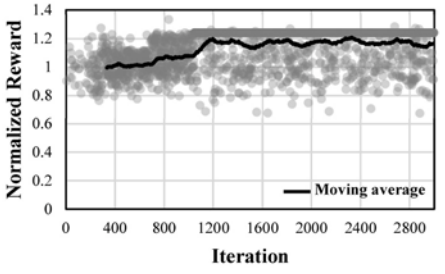
(c) 200 training frequency, diff% exploration



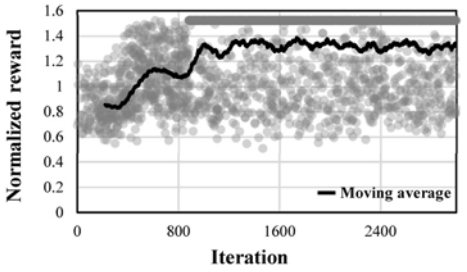
(d) 400 training frequency, 50% exploration



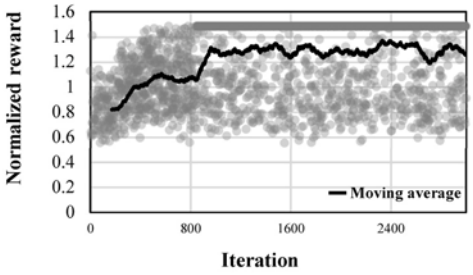
(e) 400 training frequency, 70% exploration



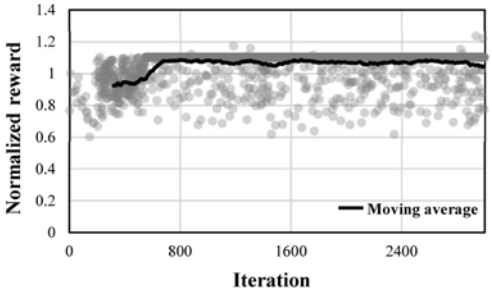
(f) 400 training frequency, diff% exploration



(g) 800 training frequency, 50% exploration



(h) 800 training frequency, 70% exploration



(i) 800 training frequency, diff% exploration

Fig. 8. MCTS + NN result for Case 1.

Table 2
Comparison of the best result between MCTS + NN and MCTS.

Case	Method	Reward	Improvement w.r.t MCTS
1	MCTS	1867815	-
	MCTS + NN_200_50%	1930750	3.37%
	MCTS + NN_200_70%	1919476	2.77%
	MCTS + NN_200_diff%	2696613	44.37%
	MCTS + NN_400_50%	1921200	2.86%
	MCTS + NN_400_70%	2580023	38.13%
	MCTS + NN_400_diff%	1951981	4.51%
	MCTS + NN_800_50%	2484537	33.02%
	MCTS + NN_800_70%	2425345	29.85%
	MCTS + NN_800_diff%	1885007	0.92%
2	MCTS	1805016	-
	MCTS + NN_200_50%	2381255	31.92%
	MCTS + NN_200_70%	1853155	2.67%
	MCTS + NN_200_diff%	1817486	0.69%
	MCTS + NN_400_50%	1935049	7.20%
	MCTS + NN_400_70%	2486243	37.74%
	MCTS + NN_400_diff%	2253985	24.87%
	MCTS + NN_800_50%	2174949	20.49%
	MCTS + NN_800_70%	2111429	16.98%
	MCTS + NN_800_diff%	1907623	5.68%
3	MCTS	2565538	-
	MCTS + NN_200_50%	3438105	34.01%
	MCTS + NN_200_70%	3480435	35.66%
	MCTS + NN_200_diff%	3199746	24.72%
	MCTS + NN_400_50%	3480435	35.66%
	MCTS + NN_400_70%	3480435	35.66%
	MCTS + NN_400_diff%	2537949	-1.08%
	MCTS + NN_800_50%	2745958	7.03%
	MCTS + NN_800_70%	2927226	14.10%
	MCTS + NN_800_diff%	2506264	-2.31%

44.37% over the baseline MCTS. In contrast, MCTS + NN_800_diff% yielded the poorest result, with only a 0.92% improvement over MCTS. In Case 2, the configuration MCTS + NN_400_70% yielded an observed improvement of 37.74%, followed by MCTS + NN_200_50%, which showed a 31.92% increase relative to MCTS. For Case 3, all configurations with training frequencies of 200 and 400 and exploration rates of 50% and 70% (i.e., MCTS+NN_200_50%, MCTS+NN_200_70%, MCTS+NN_400_50%, and MCTS+NN_400_70%) showed an observed improvement of approximately 35% over MCTS. This indicates that, under the considered experimental conditions, different combinations of NN training intervals and exploration settings can identify solutions with the same objective value. This observation suggests that multiple hyperparameter combinations are capable of reaching similarly high-quality solutions, although their search trajectories may differ. However, both MCTS + NN_400_diff% and MCTS + NN_800_diff% failed to show any improvement compared to the baseline. These results align with the findings discussed in Section 5.1, reaffirming that delayed neural network training is not recommended for the current search setup, as early and more frequent training provides better optimization outcomes.

Since both MCTS and MCTS + NN are stochastic optimization algorithms and only one independent optimization run was conducted for each configuration, the reported improvements represent the observed outcomes of the reported experiments. These results demonstrate the relative performance of the proposed MCTS + NN framework under the reported experimental conditions but should not be interpreted as statistically established superiority. Additional independent runs and statistical analyses would be required to quantify the variability and consistency of the observed improvements.

It should be noted that the optimization problem considered in this study differs from that investigated in literature [47]. Although Case 1 is based on the same building configuration as that is the literature [47], the optimization problem has been substantially expanded by including all 15 floor-system alternatives as design variables. Consequently, the brute-force optimum reported in literature [47] is not directly applicable to the present optimization problem. Therefore, the reported improvements should be interpreted as a relative comparison between MCTS + NN and the baseline MCTS rather than as an indication of their absolute proximity to the global optimum.

For further comparison of MCTS + NN with MCTS, the success rate was calculated for all three design cases. Success rate indicates the count of solutions that satisfy all the design requirements mentioned in Section 3. Fig. 9 reports the average success rate for both MCTS + NN and MCTS at three different iteration intervals. Herein, only the MCTS + NN results for atraining frequency of 200 with varying (50%, 70% and diff%) exploration rate is compared with the MCTS. For Case 1 (Fig. 9a), it is observed that for all three iteration intervals, MCTS + NN has a higher success rate compared to MCTS. Similar findings are found for Case 2 (Fig. 9b) and Case 3 (Fig. 9c). This finding indicates that training the NN using the successful design solutions improves its ability to provide additional solutions that satisfy the design requirements.

Overall, the comparisons between MCTS + NN and the basic MCTS indicate that incorporating the policy neural network noticeably improves the search efficiency. By learning from successful design solutions, the policy neural network guides the tree toward more promising actions, reducing ineffective exploration. As a result, MCTS + NN generates a higher proportion of feasible designs that satisfy the prescribed structural requirements compared to the basic MCTS.

5.3. *Design outcomes from MCTS + NN*

This section presents the design outcomes obtained from the MCTS + NN process for all three cases. Fig. 10 and Table 3 show the most cost-efficient floor layouts identified through MCTS + NN with a training interval of 200 iterations for varying exploration rate (ε). Generally, it is observed that the thinner floor panels are more economical due to lower material usage; however, this advantage is often offset by the additional cost of secondary beams required to control vibration [63]. In contrast, thicker floor panels typically do not require secondary beams but are inherently more expensive. The study, therefore, explores both approaches, which are thin floors with more beams and thick floors with fewer beams, to determine the most cost-effective configuration.

For Case 1 (Table 3), the configurations corresponding to 50% and diff% exploration rates resulted in designs with secondary beams, while the 70% rate produced a design without secondary beams. Despite the added members, the solutions with secondary beams (50% and diff%) proved to be more cost-efficient than the one without secondary beams (70%). Both 50% and diff% configurations employed 3-ply CLT, where additional beams were introduced to satisfy vibration limits, whereas the 70% case used 5-ply CLT. This finding suggests that using thinner panels supplemented with secondary beams is more effective for Case 1. As shown in Figs. 10a and 10c, a single secondary beam is required at mid-span to control vibration.

For Case 2 (Table 3), all three exploration rates (50%, 70%, and diff %) produced floor layouts that included one secondary beam at mid-span (Figs. 10d and 10e) for vibration control, each utilizing 3-ply CLT panels. This again confirms the cost efficiency of thinner floor panels. For Case 2 (Table 3), a substantial variation in the final design cost was observed among the three exploration strategies, highlighting the sensitivity of the MCTS + NN framework to the exploration–exploitation balance. Under the current parameter settings, a search budget of 3000 iterations, and a neural network training frequency of 200 iterations, the 50% exploration strategy enabled the algorithm to identify the lowest-cost design, whereas the 70% and diff% exploration strategies converged to comparatively higher-cost solutions. These results indicate that the exploration setting has a significant influence on the quality of the final design under the considered experimental conditions.

In Case 3 (Table 3), the optimal designs for all exploration rates also employed 3-ply CLT, further demonstrating the efficiency of thinner floors. However, because Case 3 features a closely spaced grid along the

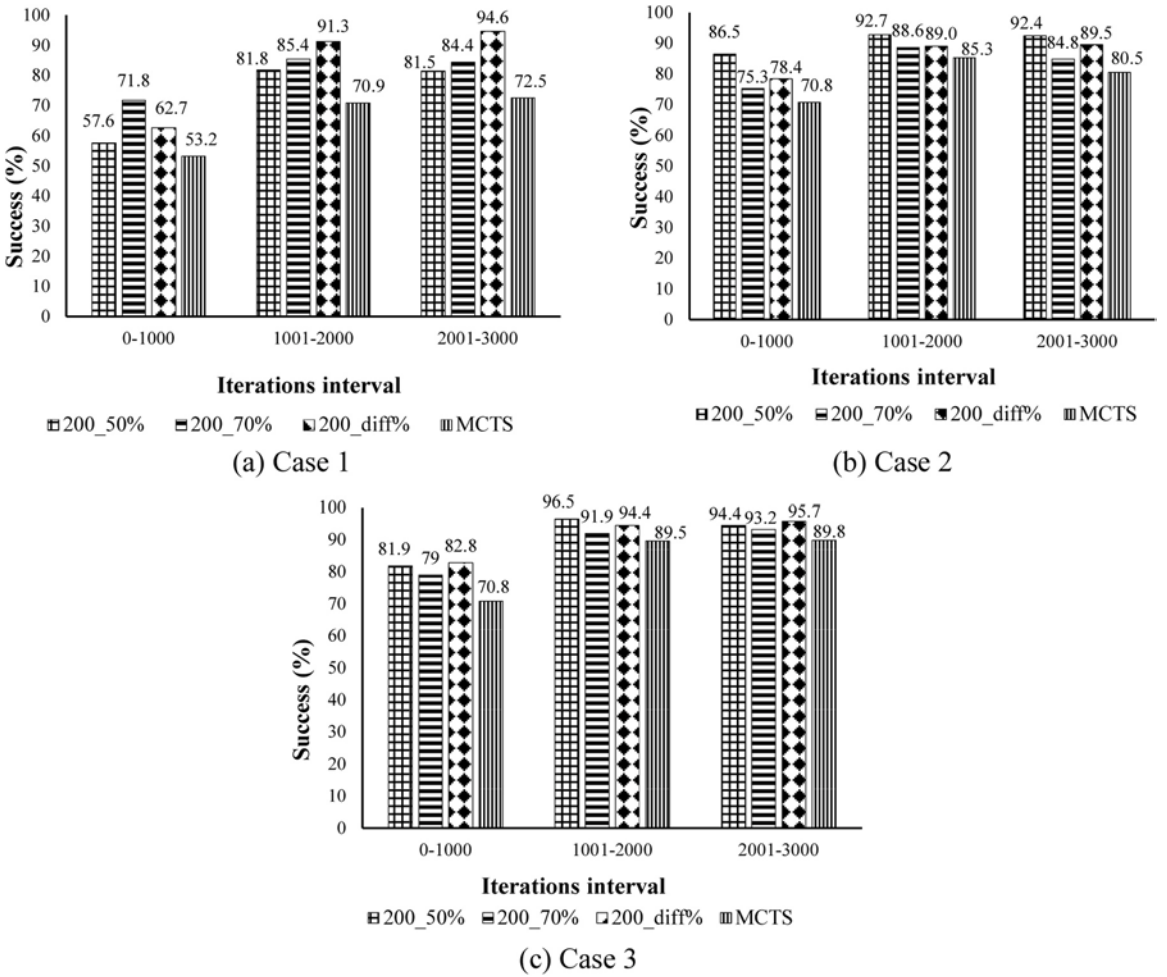


Fig. 9. Comparison of the success rate between MCTS + NN and MCTS.

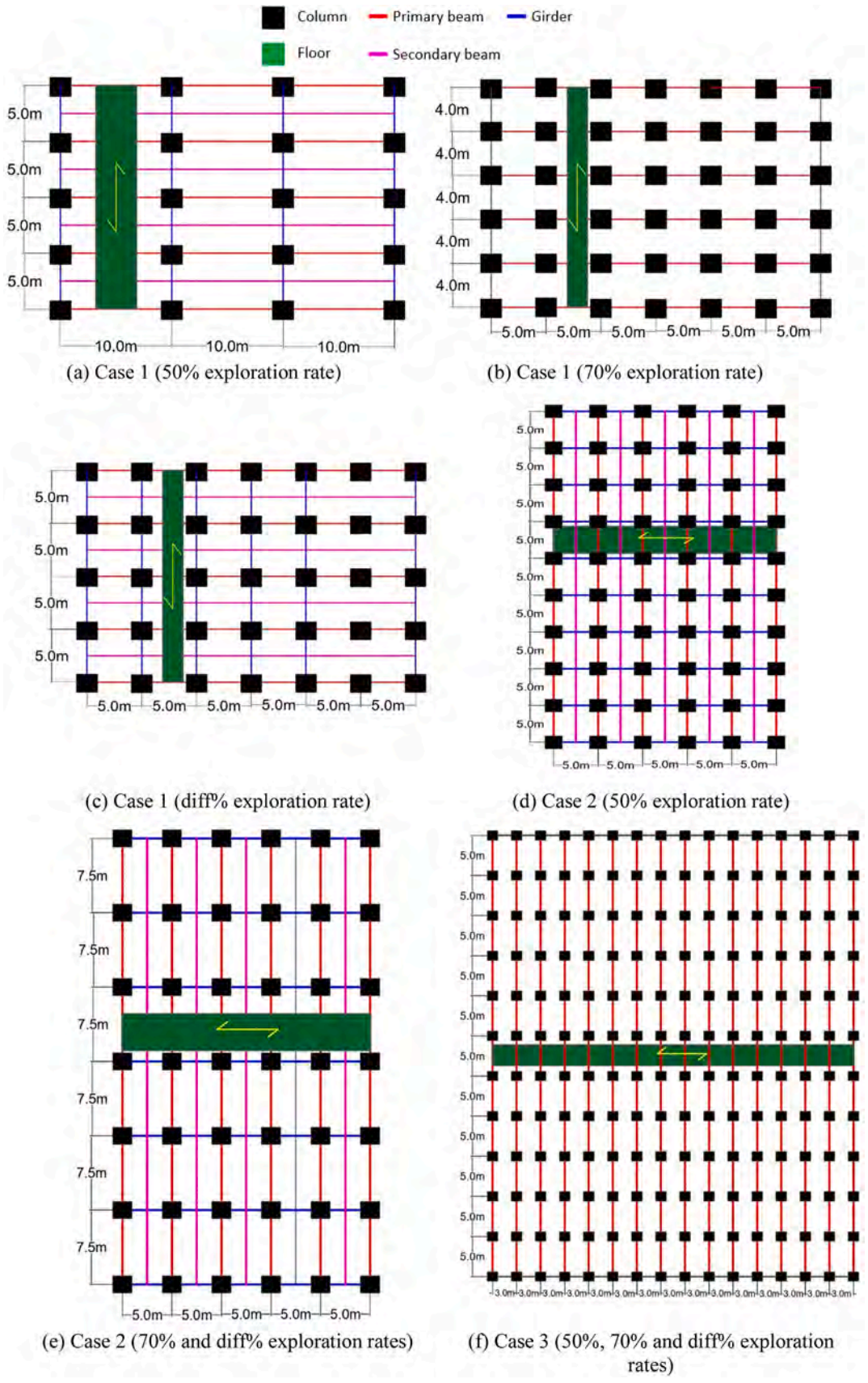


Fig. 10. Floor layouts obtained from MCTS + NN for 200 training frequency.

Table 3
Design outcomes for all three cases (200 training frequency).

Case	MCTS + NN % ε	Grid x	Grid y	Floor	Primary beam	Column	Secondary beam	Girder	Cost ($CAD)
1	50%	(3, 10.0)	(4, 5.0)	E1–3	SP_20f-E_215 × 798	SP_12c-E_175 × 228	SP_20f-E_175 × 798	SP_20f-E_130 × 798	310,760
	70%	(6, 5.0)	(5, 4.0)	E1–5	SP_20f-E_130 × 684	SP_12c-E_175 × 190	-	-	312,585
	diff%	(6, 5.0)	(4, 5.0)	E2–3	SP_20f-E_80 × 532	SP_12c-E_130 × 190	SP_20f-E_80 × 532	SP_20f-E_130 × 532	222,501
2	50%	(5, 5.0)	(9, 5.0)	V1–3	SP_20f-E_80 × 532	SP_12c-E_130 × 190	SP_20f-E_80 × 532	SP_20f-E_130 × 532	472,440
	70%	(5, 5.0)	(6, 7.5)	E1–3	SP_20f-E_130 × 684	SP_12c-E_130 × 684	SP_20f-E_130 × 684	SP_20f-E_215 × 684	607,073
	diff%	(5, 5.0)	(6, 7.5)	E2–3	SP_20f-E_215 × 646	SP_12c-E_130 × 570	SP_20f-E_130 × 646	SP_20f-E_130 × 646	618,987
3	50%	(15, 3.0)	(11, 5.0)	V2–3	DFL_24f-E_80 × 570	DFL_16c-E_130 × 152	-	-	719,873
	70%	(15, 3.0)	(11, 5.0)	E1–3	DFL_24f-E_80 × 532	DFL_16c-E_130 × 152	-	-	711,118
	diff%	(15, 3.0)	(11, 5.0)	V2–3	DFL_24f-E_130 × 494	DFL_16c-E_130 × 152	-	-	773,499

y-direction, no secondary beams were required, as shown in Fig. 10f. Additionally, Tables A.1, A.2, and A.3 summarize the best designs obtained for training frequencies of 400 and 800, which show consistent findings favoring thin 3-ply CLT panels across all configurations.

6. Conclusion

This study presents an MCTS integrated with a policy neural network (MCTS+NN)–based cost optimization framework for a post-beam-panel-type mass timber framing system. The building is designed for gravity loads and considers a single-story configuration, with structural checks performed in accordance with the Canadian design standard (CSA O86:24). Before initiating the search process, the design problem is formulated as an MDP. A reward function based on cost per unit area is developed to guide the search toward optimal and cost-effective design solutions. Furthermore, action filtering is applied during beam and column selection to reduce sparse rewards and enhance computational efficiency. The research investigates two floor layout configurations: with and without secondary beams, depending on the results of the floor vibration check. A parametric study is conducted to examine the influence of the NN training interval and exploration rate on the performance of the MCTS + NN method. Three design cases, adopted from practical building projects, are used to demonstrate the application of MCTS + NN in mass timber design optimization. The results from MCTS + NN are then compared with those from the original MCTS. The key observations from this study are summarized as follows:

- A systematic methodology is established for integrating the MCTS + NN framework into structural design optimization, providing a foundation for extending the proposed approach to more complex structural systems and other engineering design problems.
- The MCTS + NN algorithm was executed for 3000 iterations, considering three NN training intervals (200, 400, and 800) and three exploration settings (fixed: 50% and 70%, and decaying: diff% (ε_o=70% and $\Delta\eta$=5%)). The results show that, under the current MCTS parameter configuration and considered iterations, more frequent NN training (every 200 iterations) combined with lower exploration (50% or diff%) yields the best performance. To achieve better results for later training intervals (400 and 800), the MCTS parameters should be adjusted to include greater exploration and more total iterations to fully utilize the NN's learned knowledge.
- The basic MCTS was also executed for 3000 iterations across all three design cases for comparison with the proposed MCTS + NN framework. Under the reported experimental conditions, MCTS + NN achieved higher objective values and success rates than the baseline MCTS. This observed improvement is attributed to the policy network, which, once trained on successful design trajectories, provides informed guidance during action selection. Instead of relying solely on random exploration, MCTS + NN prioritizes actions with a higher likelihood of leading to feasible and cost-efficient designs. The observed reward improvements were approximately 44%, 37%, and 35% for Case 1, Case 2, and Case 3, respectively, relative to the baseline MCTS. These results correspond to the reported optimization runs and should not be interpreted as statistically established superiority.

Overall, the findings highlight the potential of the enhanced MCTS + NN framework in optimizing mass timber building designs. The study demonstrates that integrating a neural network into MCTS improves search efficiency and solution quality without significantly increasing computational effort. Future research could extend this work by allocating higher computational resources to investigate the effects of delayed NN training and exploring a broader range of design cases to further validate the model's applicability. In addition, the costs of CLT and glulam, which impact the reward function, were fixed in this study. These parameters could be set up as the input parameters in the

optimization framework. These limitations will be addressed in future studies by the authors.

CRediT authorship contribution statement

Samia Zakir Sarothi: Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Hoang D. Nguyen:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization. **Qipei Mei:** Writing – review & editing, Supervision, Investigation, Funding acquisition, Conceptualization. **Ying Hei Chui:** Writing – review & editing, Supervision, Investigation, Funding acquisition, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Alliance Grant program (ALLRP 5810742022), Alberta Innovates (212201668), and Timber Engineering Inc.

Appendix

Table A.1
Design outcome for Case 1 (training frequency 400 and 800)

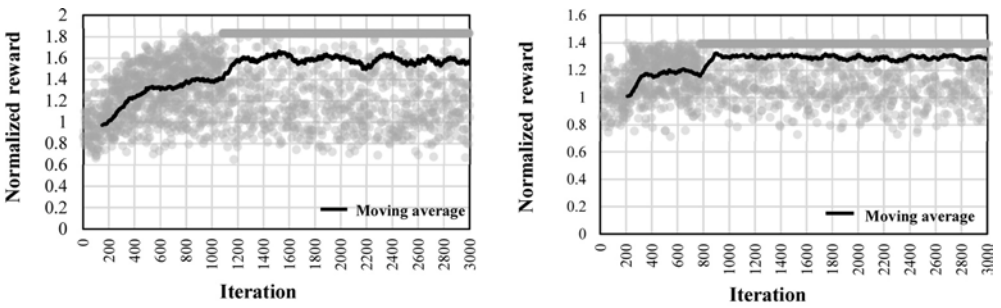
MCTS + NN		Grid x	Grid y	Floor	Primary beam	Column	Secondary beam	Girder	Cost ($CAD)
Training frequency	% ε								
400	50%	(4, 7.5)	(4, 5.0)	V1–3	SP_ 20f-E_130 × 722	SP_12c-E_130 × 608	SP_ 20f-E_215 × 722	SP_ 20f-E_130 × 798	312,305
	70%	(6, 5.0)	(4, 5.0)	E2–3	SP_ 20f-E_80 × 532	SP_12c-E_130 × 190	SP_ 20f-E_80 × 532	SP_ 20f-E_175 × 532	232,556
	diff%	(3, 10.0)	(4, 5.0)	E2–3	SP_ 20f-E_175 × 836	SP_12c-E_175 × 798	SP_ 20f-E_130 × 836	SP_ 20f-E_130 × 836	307,380
800	50%	(6, 5.0)	(4, 5.0)	E3–3	SP_ 20f-E_80 × 532	SP_12c-E_130 × 190	SP_ 20f-E_80 × 532	SP_ 20f-E_215 × 532	241,494
	70%	(6, 5.0)	(4, 5.0)	E2–3	SP_ 20f-E_80 × 570	SP_12c-E_130 × 190	SP_ 20f-E_80 × 570	SP_ 20f-E_215 × 570	247,388
	diff%	(6, 5.0)	(2, 10.0)	E3–3	SP_ 20f-E_215 × 532	SP_12c-E_175 × 1064	SP_ 20f-E_130 × 532	SP_ 20f-E_265 × 684	318,301

Table A.2
Design outcome for Case 2 (training frequency 400 and 800)

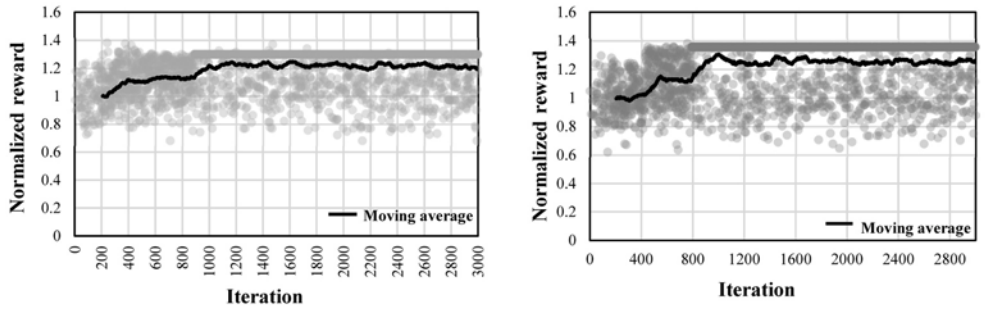
MCTS + NN		Grid x	Grid y	Floor	Primary beam	Column	Secondary beam	Girder	Cost ($CAD)
Training frequency	% ε								
400	50%	(5, 5.0)	(6, 7.5)	V1–3	SP_ 20f-E_130 × 722	SP_12c-E_130 × 342	SP_ 20f-E_130 × 722	SP_ 20f-E_130 × 874	581,381
	70%	(5, 5.0)	(9, 5.0)	V2–3	SP_ 20f-E_80 × 532	SP_12c-E_130 × 190	SP_ 20f-E_80 × 532	SP_ 20f-E_80 × 532	452,490
	diff%	(5, 5.0)	(9, 5.0)	V2–3	SP_ 20f-E_80 × 532	SP_12c-E_130 × 570	SP_ 20f-E_80 × 532	SP_ 20f-E_130 × 532	499,116
800	50%	(5, 5.0)	(9, 5.0)	E2–3	SP_ 20f-E_130 × 570	SP_12c-E_130 × 342	SP_ 20f-E_80 × 570	SP_ 20f-E_130 × 570	517,253
	70%	(5, 5.0)	,(9, 5.0)	E1–3	SP_ 20f-E_215 × 456	SP_12c-E_215 × 304	SP_ 20f-E_80 × 456	SP_ 20f-E_130 × 456	532,814
	diff%	(5, 5.0)	(6, 7.5)	E2–3	SP_ 20f-E_130 × 760	SP_12c-E_130 × 494	SP_ 20f-E_130 × 760	SP_ 20f-E_130 × 760	589,739

Table A.3
Design outcome for Case 3 (training frequency 400 and 800)

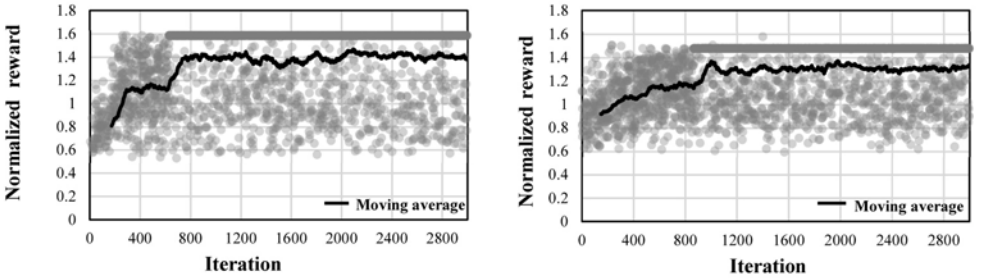
MCTS + NN		Grid x	Grid y	Floor	Primary beam	Column	Secondary beam	Girder	Cost ($CAD)
Training frequency	% ε								
400	50%	(15, 3.0)	(11, 5.0)	E3–3	DFL_ 24f-E_80 × 532	DFL_16c-E_130 × 152	-	-	711,118
	70%	(15, 3.0)	(11, 5.0)	E2–3	DFL_ 24f-E_80 × 532	DFL_16c-E_130 × 152	-	-	711,118
	diff %	(15, 3.0)	(11, 5.0)	V1–3	DFL_ 24f-E_215 × 494	DFL_16c-E_175 × 380	-	-	975,197
800	50%	(15, 3.0)	(11, 5.0)	V2–3	DFL_ 24f-E_175 × 532	DFL_16c-E_80 × 570	-	-	901,325
	70%	(15, 3.0)	(11, 5.0)	V2–3	DFL_ 20f-E_175 × 418	DFL_16c-E_175 × 266	-	-	845,510
	diff %	(5, 9.0)	(11, 5.0)	V1–3	DFL_ 24f-E_80 × 532	DFL_16c-E_130 × 266	DFL_ 20f-E_130 × 532	DFL_ 24f-E_130 × 532	987,526



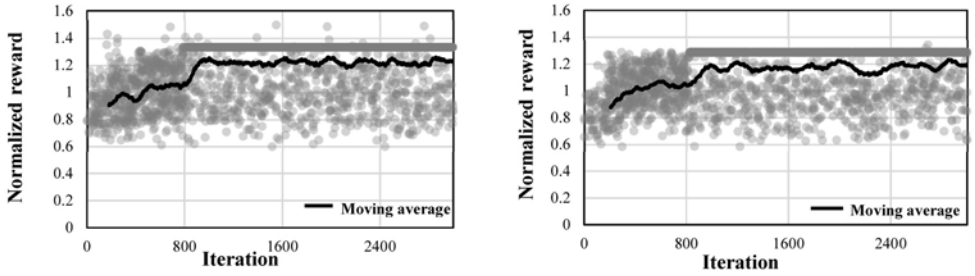
(a) 200 training frequency, 50% exploration (b) 200 training frequency, 70% exploration



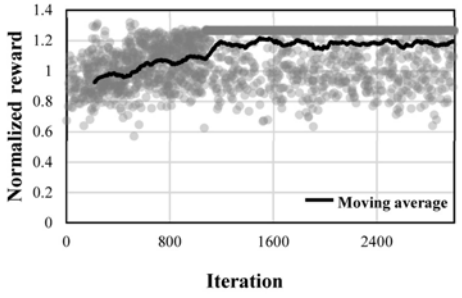
(c) 200 training frequency, diff% exploration (d) 400 training frequency, 50% exploration



(e) 400 training frequency, 70% exploration (f) 400 training frequency, diff% exploration

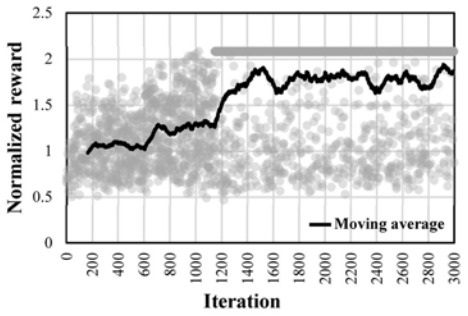


(g) 800 training frequency, 50% exploration (h) 800 training frequency, 70% exploration

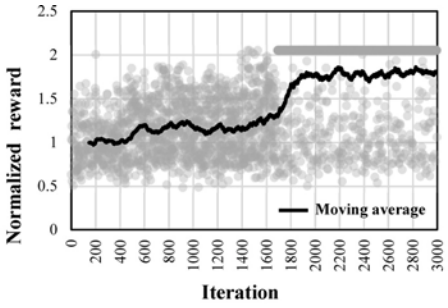


(i) 800 training frequency, diff% exploration

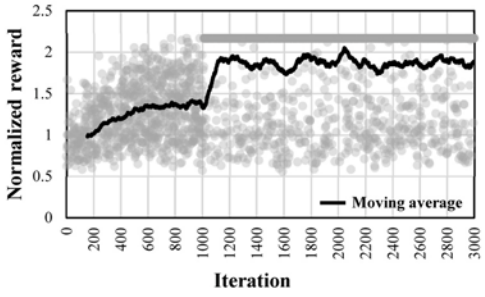
Fig. A.1. MCTS + NN result for Case-2.



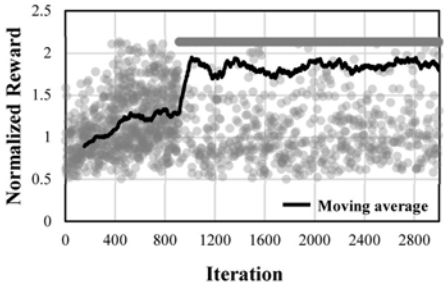
(a) 200 training frequency, 50% exploration



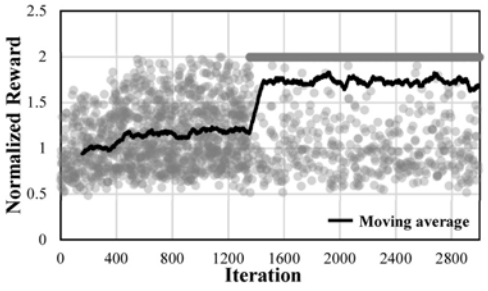
(b) 200 training frequency, 70% exploration



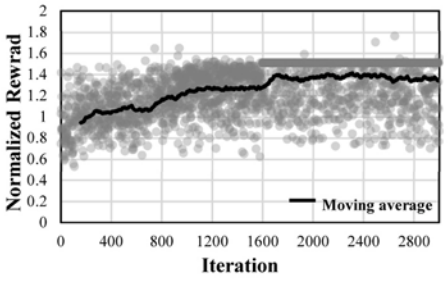
(c) 200 training frequency, diff% exploration



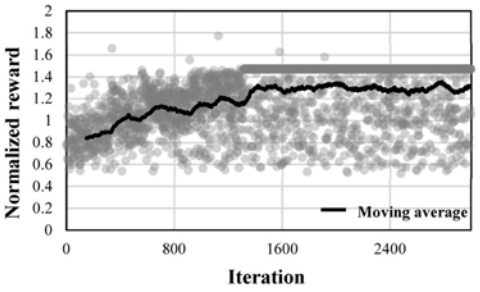
(d) 400 training frequency, 50% exploration



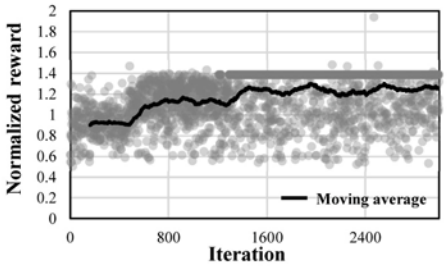
(e) 400 training frequency, 70% exploration



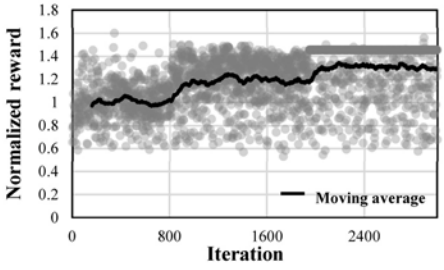
(f) 400 training frequency, diff% exploration



(g) 800 training frequency, 50% exploration



(h) 800 training frequency, 70% exploration



(i) 800 training frequency, diff% exploration

Fig. A.2. MCTS + NN result for Case-3.

Data Availability

No data was used for the research described in the article.

References

[1] Mohammadi J, Ling L. Can wood become an alternative material for tall building construction? Pract Period Struct Des Constr 2017;22. https://doi.org/10.1061/(asce)sc.1943-5576.0000334.

[2] Concu G. Introductory chapter: timber and sustainability in construction. Timber Build Sustain 2019. https://doi.org/10.5772/intechopen.90079.

[3] Campbell A. What engineers (still) do not know about wood: an engineer's perspective on key knowledge gaps in the use of mass timber. Int Wood Prod J 2020;11. https://doi.org/10.1080/20426445.2020.1730047.

[4] Silva A, Martins AC, Feio AO, Machado JS. Feasibility of creosote treatment for glued-laminated pine-timber railway sleepers. J Mater Civil Eng 2015;27. https://doi.org/10.1061/(asce)mt.1943-5533.0001073.

[5] Harte AM. Mass timber – the emergence of a modern construction material. J Struct Integr Maint 2017;2. https://doi.org/10.1080/24705314.2017.1354156.

[6] Gong M. Lumber-Based Mass Timber Products in Construction. Timber Building and Sustainability. IntechOpen; 2019.

[7] Kordziel S, Pei S, Glass SV, Zelinka S, Tabares-Velasco PC. Structure moisture monitoring of an 8-story mass timber building in the Pacific Northwest. J Archit Eng 2019;25. https://doi.org/10.1061/(asce)ae.1943-5568.0000367.

[8] APAStandard for Performance-Rated Cross-Laminated Timber, 2020.

[9] Version US. Mass Timber Technical Guide for CrossLam ® CLT and GlulamPLUS ®. n.d.

[10] Ritchie L., Stephan A.Engineered timber for apartment buildings in Melbourne, Australia: A construction cost comparison with traditional concrete systems, 2018..

[11] Ahmed S, Arocho I. Analysis of cost comparison and effects of change orders during construction: study of a mass timber and a concrete building project. J Build Eng 2021;33. https://doi.org/10.1016/j.jobe.2020.101856.

[12] Burback B, Pei S. Cross-laminated timber for single-family residential construction: comparative cost study. J Archit Eng 2017;23. https://doi.org/10.1061/(asce)ae.1943-5568.0000267.

[13] Laguarda Mallo MF, Espinoza O. Awareness, perceptions and willingness to adopt Cross-Laminated Timber by the architecture community in the United States. J Clean Prod 2015;94. https://doi.org/10.1016/j.jclepro.2015.01.090.

[14] Ahmed S, Arocho I. Mass timber building material in the U.S. construction industry: Determining the existing awareness level, construction-related challenges, and recommendations to increase its current acceptance level. Clean Eng Technol 2020;1. https://doi.org/10.1016/j.clet.2020.100007.

[15] Venkayya VB. Structural optimization: A review and some recommendations. Int J Numer Methods Eng 1978;13. https://doi.org/10.1002/nme.1620130202.

[16] Jelušič P, Kravanja S. Optimal design and competitive spans of timber floor joists based on multi-parametric MINLP Optimization. Materials 2022;15. https://doi.org/10.3390/ma15093217.

[17] Jelusic P, Kravanja S. Optimal design of timber-concrete composite floors based on the multi-parametric MINLP optimization. Compos Struct 2017;179. https://doi.org/10.1016/j.compstruct.2017.07.062.

[18] Kravanja S, Žula T. Optimization of a single-storey timber building structure. Int J Comput Methods Exp Meas 2021;9. https://doi.org/10.2495/CMEM-V9-N2-126-140.

[19] Kravanja S, Žula T. Optimization of a timber hall structure. WIT Trans Built Environ 2020;196. https://doi.org/10.2495/HPSM200191.

[20] Šilih S, Kravanja S, Premrov M. Shape and discrete sizing optimization of timber trusses by considering of joint flexibility. Adv Eng Softw 2010;41. https://doi.org/10.1016/j.advengsoft.2009.07.002.

[21] Villar-García JR, Vidal-López P, Rodríguez-Robles D, Guaita M. Cost optimisation of glued laminated timber roof structures using genetic algorithms. Biosyst Eng 2019;187. https://doi.org/10.1016/j.biosystemseng.2019.09.008.

[22] Villar JR, Vidal P, Fernández MS, Guaita M. Genetic algorithm optimisation of heavy timber trusses with dowel joints according to Eurocode 5. Biosyst Eng 2016;144. https://doi.org/10.1016/j.biosystemseng.2016.02.011.

[23] Simón-Portela M, Villar-García JR, Rodríguez-Robles D, Vidal-López P. Optimization of Glulam Regular Double-Tapered Beams for Agroforestry Constructions. Appl Sci (Switz) 2023;13. https://doi.org/10.3390/app13095731.

[24] Hayashi K, Ohsaki M. Graph-based reinforcement learning for discrete cross-section optimization of planar steel frames. Adv Eng Inform 2022;51. https://doi.org/10.1016/j.aei.2021.101512.

[25] Hayashi K, Ohsaki M. Reinforcement learning for optimum design of a plane frame under static loads. Eng Comput 2021;37. https://doi.org/10.1007/s00366-019-00926-7.

[26] Sarhan MM, Al-Zwainy FMS, Sarhan MM, Al-Zwainy FMS. Structural performance of composite beams using advanced modeling for nonlinear analysis. Dyna (Medellin) 2025;92:103–10. https://doi.org/10.15446/DYNA.V92N238.119561.

[27] Al-Zwainy FMS, Salih SA, Aldikheeli MR. Prediction of residual strength of sustainable self-consolidating concrete exposed to elevated temperature using artificial intelligent technique. Int J Appl Sci Eng 2021;18:1–15. https://doi.org/10.6703/IJASE.202106_18(2).012.

[28] Al-Somaydaii JA, Albadri AT, Al-Zwainy FMS. Hybrid approach for cost estimation of sustainable building projects using artificial neural networks. Open Eng 2024;14. https://doi.org/10.1515/ENG-2022-0485/XML.

[29] Jasim NA, Maruf SM, Aljumaily HSM, Al-Zwainy FMS. Predicting index to complete schedule performance indicator in highway projects using artificial neural network model. Arch Civil Eng 2020;66:541–54. https://doi.org/10.24425/ACE.2020.134412.

[30] MGS Al-khazrajy, FMS Al-Zwainy, Obaid AH, Sobaih AEE, Elshaer IA. Harnessing Artificial Neural Networks for Predictive KPI Insights in Infrastructure Projects. Lecture Notes in Networks and Systems, 1393. LNNS; 2026. p. 769–89. https://doi.org/10.1007/978-3-031-90893-4_52/TABLES/10.

[31] FMS Al-Zwainy, MGS Al-khazrajy, Hussein NM, Mohamed S, Sarhan MM, TJ Al-Musawi, et al. Utilizing artificial neural networks for predictive KPI analysis in bridge projects. J Comput Anal & Appl 2024;33:1290.

[32] Ali LN, Dhamad SHRAl, Al-Zwainy FMS, Zaki RIK, Varouqa IF, Dulaimi SDSAl, et al. Development of a support vector machine-based predictive model for bored pile productivity in residential construction projects in Iraq. ISSN 0012-7353, Vol 93, N° 240, 2026, Págs 81-88 DYNA Rev De La Fac De Minas Univ Nac De Colomb Sede MedellíN 2026;93:81–8. https://doi.org/10.15446/dyna.v93n240.121949.

[33] Nguyen HD, Dao ND, Shin M. Capability of machine learning to predict seismic damage states of reinforced concrete wall structures. J Build Eng 2025;106:112620. https://doi.org/10.1016/J.JOBE.2025.112620.

[34] Nguyen HD, Kim C, Lee K, Shin M. Development of data-driven models to predict seismic drift response of RC wall structures: An application of deep neural networks. Soil Dyn Earthq Eng 2024;186:108952. https://doi.org/10.1016/J.SOILDYN.2024.108952.

[35] Nguyen VQ, Nguyen HD, Petrone F, Park D. Rapid damage state classification for underground box tunnels using machine learning. Struct Infrastruct Eng 2025;21:1395–408. https://doi.org/10.1080/15732479.2023.2266709.

[36] Rossi L, Winands MHM, Butenweg C. Monte Carlo Tree Search as an intelligent search tool in structural design problems. Eng Comput 2022;38. https://doi.org/10.1007/s00366-021-01338-2.

[37] Nguyen HD, Dao ND, Shin M. Capability of machine learning to predict seismic damage states of reinforced concrete wall structures. J Build Eng 2025;106. https://doi.org/10.1016/j.jobe.2025.112620.

[38] Nguyen HD, Mei Q, Chui YH. A genetic algorithm-based calibration procedure for hysteresis loops in timber structures. Eng Struct 2026;346. https://doi.org/10.1016/j.engstruct.2025.121622.

[39] Aldhamad SHR, Maya R, Alazawy SFM, Alzwainy FM. Forecasting models for time and cost performance predicting of infrastructural projects. Civil Environ Eng 2024;20:1024–39. https://doi.org/10.2478/CEE-2024-0074.

[40] Risan HK, Serhan FM, AA Al-Azzawi. Management of a typical experiment in engineering and science. AIP Conf Proc 2024:2864. https://doi.org/10.1063/5.0186079/3179014.

[41] Sarhan M, Al-Zwainy FMS. Structural investigations of confined reinforced concrete beam-bending members. 2025. SpringerMM Sarhan, FMS Al-ZwainyInnovative Infrastructure Solutions. Springer; 2025. p. 201. https://doi.org/10.1007/S41062-025-02001-1. 2025.

[42] Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, et al. A survey of Monte Carlo Tree Search methods. IEEE Trans Comput Intell AI Games 2012;4. https://doi.org/10.1109/TCIAIG.2012.2186810.

[43] Ko FY, Suzuki K, Yonekura K. Improved Monte Carlo Tree Search formulation with multiple root nodes for discrete sizing optimization of truss structures. Eng Optim 2025;57. https://doi.org/10.1080/0305215X.2024.2413962.

[44] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016;529. https://doi.org/10.1038/nature16961.

[45] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. Nature 2017;550. https://doi.org/10.1038/nature24270.

[46] Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. Nature 2020;588. https://doi.org/10.1038/s41586-020-03051-4.

[47] Sarothi SZ, Nguyen HD, Mei Q, Chui YH. Monte Carlo Tree Search for mass timber building design optimization. Comput-Aided Civil Infrastruct Eng 2025;40. https://doi.org/10.1111/mice.70151.

[48] IndeterminateBeam. Welcome to IndeterminateBeam's documentation! — IndeterminateBeam 0.1 documentation n.d. https://indeterminatebeam.readthedocs.io/en/main/ (accessed February 2, 2026).

[49] CSACSA O86:24 Engineering design in wood, 2024.

[50] NRCC. National Building Code of Canada 2020. vol., 1. Canadian Commission on Building and Fire Codes; 2022.

[51] CWC. Wood design manual, 2020: the complete reference for wood design in Canada. 2021.

[52] Luo R, Wang Y, Xiao W, Zhao X. AlphaTruss: Monte Carlo Tree Search for Optimal Truss Layout Design. Buildings 2022;12. https://doi.org/10.3390/buildings12050641.

[53] Kocsis L, Szepesvári C. Bandit based Monte-Carlo planning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 4212. LNAI; 2006. https://doi.org/10.1007/11871842_29.

[54] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Mach Learn 2002;47. https://doi.org/10.1023/A:1013689704352.
[55] Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville - Google Books n. d. https://books.google.ca/books?hl=en&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&ots=MOU3gmnCRV&sig=CdGxgzCdn8B-Os-PIbfPUU4gVBw&redir_esc=y#v=onepage&q&f=false (accessed July 31, 2026).
[56] Kingma DP, Ba JL. Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations. ICLR 2015 - Conference Track Proceedings; 2014.
[57] Naturally:wood. BROCK COMMONS TALLWOOD HOUSE DESIGN AND PRECONSTRUCTION OVERVIEW. 2016.
[58] ICHSANNUAL REPORT, 2017.
[59] StructureCraft. T3 Minneapolis n.d. https://structurecraft.com/projects/t3-minneapolis (accessed February 2, 2026).
[60] Bullitt Center | Technoform n.d. https://www.technoform.com/en/project/bullitt-center (accessed July 31, 2026).
[61] UMass Amherst Design Building Zipper Trusses n.d. https://www.architectmagazine.com/technology/architectural-detail/umass-amherst-design-building-zipper-trusses_o/ (accessed July 31, 2026).
[62] Abed J, Rayburg S, Rodwell J, Neave M. A review of the performance and benefits of mass timber as an alternative to concrete and steel for improving the sustainability of structures. Sustain (Switz) 2022;14. https://doi.org/10.3390/su14095570.
[63] Woodworks. Creating Efficient Structural Grids in Mass Timber Buildings - WoodWorks | Wood Products Council n.d. https://www.woodworks.org/resources/creating-efficient-structural-grids-in-mass-timber-buildings/ (accessed February 2, 2026).