Optimizing The Stiffness Distribution of a MDOF Structure under Seismic Load

Aaron Appelle [*] and Davyd Tamrazov[†]
Stanford University, Stanford, California, 94305

A chief goal in seismic engineering is to minimize the potential for extreme damage in a structure during an earthquake. One predictor of damage in a building is the interstory drift ratio (IDR), which is a measure of the relative displacement between two stories. This paper discusses methods to find the optimal distribution of story stiffnesses in a multiple-degree-of-freedom (MDOF) structure in order to minimize the peak IDR under seismic loading. This optimal stiffness distribution could represent the optimal height to place lateral force braces in a multistory structure. Three proposed optimization methods suggest that the optimal story stiffness is largest at the base and decreases as you move up the structure.

I. Nomenclature

α	=	positive weight, associated with the increase in stiffness of the structure
DE	=	differential evolution
GA	=	genetic algorithm
GP	=	gaussian process
h_s	=	story height
IDR	=	inter-story drift ratio
j	=	story number
$\mathbf{k}$	=	vector of stiffnesses, used as design variable
$\mathbf{k_0}$	=	vector of uniform stiffnesses, used as benchmark
k_{min}	=	lower bound on story stiffness
k_{max}	=	upper bound on story stiffness
MADS	=	mesh adaptive direct search
n	=	mode number
N	=	number of floors
NMS	=	Nelder-Mead simplex
t	=	time of the record
$\ddot{u}_g(t)$	=	ground acceleration
u_j	=	displacement at story j

II. Introduction

Earthquake or seismic engineering is a discipline that aims to analyze and design structures under seismic loads. The fundamental goal of earthquake engineers is to minimize the loss of life during a seismic event. However, some amount of damage to the building is typically allowed, as preventing all damage would require conservative and expensive designs in anticipation of a relatively rare event. Much research effort has gone into efficiently analyzing multistory buildings under seismic loading in order to predict their response. As of today, the most accurate form of analysis is a response history analysis (RHA), which computes the displacement, velocity, and acceleration of the structure at discrete points in time using finite difference time integration methods. RHA methods can return a good prediction of the deformation of a structure under earthquake loading, which allows engineers to estimate its degree of damage.

[*] Candidate for MS in Structural Engineering, Department of Civil and Environmental Engineering

[†] Candidate for MS in Structural Engineering, Department of Civil and Environmental Engineering

A. Problem

Consider a frame structure with N stories defined by a mass matrix $\mathbf{M}$, damping matrix $\mathbf{C}$, and stiffness matrix $\mathbf{K}$, subjected to a time-varying applied load $\mathbf{P}(t)$. This is a multiple-degree-of-freedom (MDOF) dynamic system which can be represented by the following dynamic equation of motion:

$$\mathbf{M}\ddot{\boldsymbol{u}} + \mathbf{C}\dot{\boldsymbol{u}} + \mathbf{K}\boldsymbol{u} = \mathbf{P}(t) \tag{1}$$

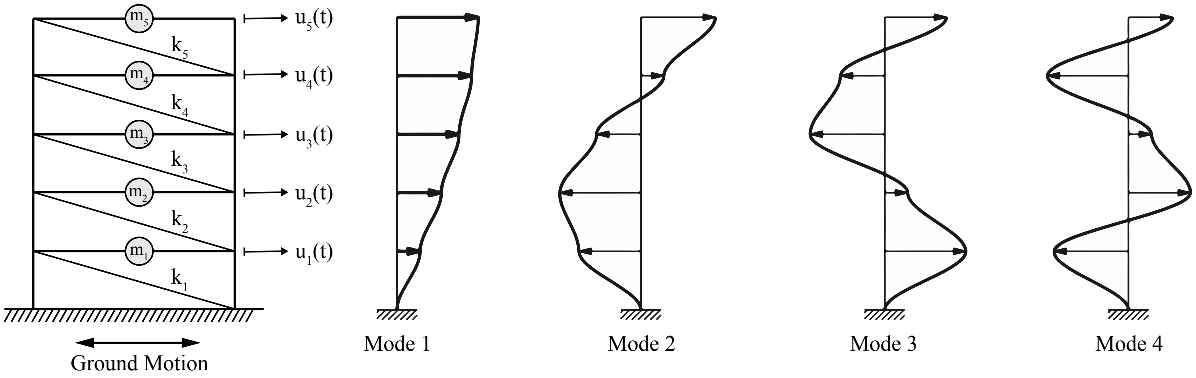


Fig. 1 Lumped mass model of multi-story structure and modal shapes

This model can be used for a structure under ground motion where the applied "load" at the base of the structure is equal and opposite to the ground acceleration. At any instant in time, the deformation of the structure at story j from its initial position is u_j. The response of the structure in an earthquake depends primarily on the building's natural period(s) of vibration. A MDOF system can respond in as many different modes of vibration as it has stories. Examples of the different modal shapes of vibration are shown in Figure 1.

The natural periods of a structure with a prescribed mass and damping will depend only on the distribution of stiffnesses over the height of the building. Once the story stiffnesses $\mathbf{k} = \{k_1, k_2, \ldots, k_N\}$ are known, the natural periods can be determined via eigenvalue analysis described in Section II.B.

An important predictor of damage in a building is the inter-story drift ratio (IDR) which is a measure of the relative displacement between two stories. IDR is defined as:

$$\mathrm{IDR}_j = \frac{u_j - u_{j-1}}{h_j} \tag{2}$$

where h_j is the height of the story equal to the distance between level j and level $j-1$. The typical range of IDR in a structure is between 0% and 2.5%, where anything larger than this tends to cause dangerous levels of damage. The ASCE 7-16 manual defines allowable IDRs for different types of structures in Table 12.12-1, Allowable Story Drift [1].

The goal considered in this paper is to minimize the peak IDR experienced by a multistory building when subjected to an earthquake. The lower the maximum IDR in a building, the less damage expected in the building.

B. Formulation of Objective

The structural response of a MDOF structure can be determined by solving a system of uncoupled equations of motion corresponding to each mode [2]. The unknown for each equation is the modal relative displacement coordinates $D_n(t)$ of an equivalent SDOF system with the modal natural frequency, mass and damping. For the purpose of this paper, damping (ξ_n) is assumed as 2.0% for all modes.

$$\ddot{D}_n(t) + 2\xi_n\omega_n\dot{D}_n(t) + \omega_n^2 D_n(t) = -\ddot{u}_g(t) \tag{3}$$

The natural frequencies and modal shapes are obtained by solving the eigenvalue problem $(\mathbf{A} - \lambda\mathbf{I})x = 0$. The eigenvalues are $\lambda = \omega^2$, and the eigenvector is $x = \phi$:

$$(\mathbf{K} - \omega^2\mathbf{M})\{\phi\} = 0 \tag{4}$$

The time history of any response parameter can be computed by summing the scaled modal responses, according to the principle of modal superposition [2]. Using this approach, inter-story drifts are calculated per Equation 5 and illustrated in Figure 2. Since lower modes tend to dominate IDR response of the structure, only the first 3 modes are included in this report in order to reduce computational cost of the objective function.

$$\mathrm{IDR}_j(t) = \frac{1}{h_{s,j}} \sum_{n=1}^{N} \Gamma_n (\phi_{n,j} - \phi_{n,j-1}) D_n(t) \tag{5}$$

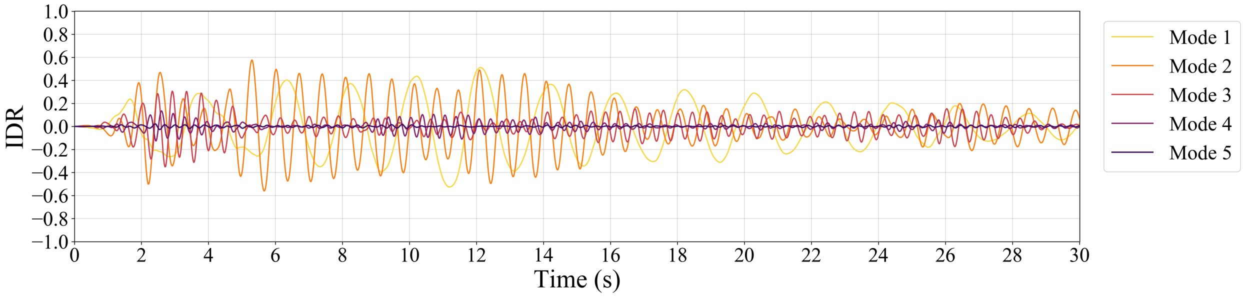


Fig. 2 Example computation of IDR for the ground story of of a 5-floor structure under the El Centro (1940) North-South direction earthquake. This is a common test ground motion used in the discipline.

C. Single Objective Optimization

The objective function finds the stiffness distribution which minimizes the maximum inter-story drift ratio over the duration of an earthquake. Since ordinary structures are normally built with the uniform stiffness distribution (i.e. k is constant at all stories), the maximum IDR obtained with the ordinary structure is used as the benchmark for success. This is quantified by defining the objective function as the maximum IDR normalized by the benchmark value having stiffness $\boldsymbol{k_0} = \{k, k, \ldots, k\}$. The formal definition of the optimization problem is outlined in Equation 6.

$$\begin{aligned} \underset{\mathbf{k}}{\text{minimize}} \quad & \frac{\mathrm{IDR}_{\max}(\mathbf{k})}{\mathrm{IDR}_{\max}(\mathbf{k_0})} \\ \text{subject to} \quad & k_i \geq k_{min} \\ & k_i \leq k_{max} \end{aligned} \tag{6}$$

In the above formulation, the design point is the vector of story stiffnesses $\mathbf{k} = \{k_1, k_2, \ldots, k_N\}$, and the dimensionality of the problem is equal to the number of stories of the building. Limits are placed on the stiffness values for each floor in order to keep the change in stiffness within a certain percentage of the uniform distribution, thereby achieving a more realistic approach that could be taken by a designer. For the purpose of this project, constraints are placed at $0.25k_0$ and $1.75k_0$, allowing for up to 75% change from the initial stiffness value (set as 1.0).

All optimization strategies in the paper consider the single-objective problem except as illustrated in Figure 7b.

D. Multi-Objective Optimization

The multi-objective approach given in Equation 7 balances the desire to minimize the maximum IDR with the desire to keep stiffness values low. The magnitude of stiffness directly influences the cost of the structure, as higher stiffness requires larger structural members or more complicated lateral bracing solutions. The linear sum of the stiffnesses is normalized by the sum of the uniform stiffness vector, using their respective ℓ_1-norms. This allows to quantitatively compare two objective functions with otherwise different units — IDR and stiffness. The weight α signifies the degree of importance of each objective.

$$\begin{aligned} \underset{\mathbf{k}}{\text{minimize}} \quad & (1-\alpha) \cdot \frac{\mathrm{IDR}_{\max}(\mathbf{k})}{\mathrm{IDR}_{\max}(\mathbf{k_0})} + \alpha \cdot \frac{\|\mathbf{k}\|_1}{\|\mathbf{k_0}\|_1} \\ \text{subject to} \quad & k_i \geq k_{min} \\ & k_i \leq k_{max} \end{aligned} \tag{7}$$

III. Optimization Methods

The following section summarizes methods that were used to explore and optimize the objective function. Optimization strategies are presented chronologically in the order of development.

A. Mesh Adaptive Direct Search

The initial exploration of the objective function was based on the assumption of its non-convexity, since it relies on an analysis of a highly irregular ground motion record. Hence, a stochastic Mesh Adaptive Direct Search (MADS) algorithm was used to explore the degree of convexity of the objective function. The implementation is provided by Kochenderfer [3].

For this optimization scheme, a starting point is randomly generated from a uniform distribution, bounded by the limits on the story stiffnesses (see Eq. 6). The algorithm then generates a random positive spanning set covering the distance defined by the initial step size of 1.0 and evaluates points in each direction. The next design point is selected contingent upon several criteria: its objective function value should be lower than that of the current point, and all of the constraints should be satisfied.

This algorithm was run for a number of random starting design points for a 5-story building. The convergence curves of the objective function for these points are shown on the Figure 3 and, despite the large number of iterations, MADS algorithm converges very quickly. The opportunistic nature of the algorithm and the non-convexity of the objective function can be clearly observed as different points get attracted to various local minima across the design space. However, precisely due to the randomness of this algorithm, some very efficient designs were obtained, which, in fact, outperform some of the population methods.

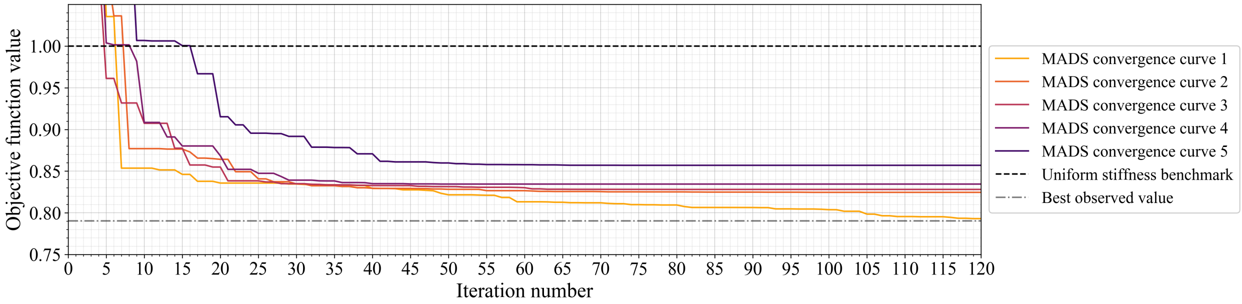


Fig. 3 Convergence examples for a 5-story building using the MADS algorithm

B. Surrogate Optimization

A call to the objective function (Eq. 6) is relatively expensive. As the dimensionality of the problem increases, optimization techniques which require repeated function calls become infeasible. A surrogate model which is less expensive to call was desired to speed up standard direct methods or descent methods.

The Gaussian process (GP) was used to compute a probabilistic surrogate model based on a set of training points. The model was implemented in Python using the `scikit-learn` package, which provides a number of built in kernel functions and fitting methods to choose from [4]. For this experiment, a zero-mean function was used in combination with a squared exponential kernel with the characteristic length-scale ℓ selected as 0.1 [3]:

$$k(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) \tag{8}$$

A training set of 50 random points was initially used to fit the three-dimensional surrogate model. After fitting a model with the initial training set, lower confidence bound exploration was used to select the next training point. The next sample is intended to balance the desire to minimize the function and minimize the uncertainty of the model [3]:

$$LB(\mathbf{x}) = \hat{\mu}(\mathbf{x}) - \alpha\hat{\sigma}(\mathbf{x}) \tag{9}$$

This method proved unsuccessful. Since the GP utilized a zero-mean function, it would predict a function evaluation of 0 with $\sigma = 1$ at all locations that strayed too far from the training points. For a multi-dimensional problem with a small training set, this rendered most of the locations in the design space equally profitable to explore. It was concluded that the initial training set needed to be much larger to be able to accurately identify features of the analyzed non-convex objective function. A Halton sequence was chosen to populate the initial low-discrepancy training set using the `Chaospy` package [5] in Python.

The feasibility of a GP surrogate model with the larger training set was tested on a 2-story building to optimize $\mathbf{k} = \{k_1, k_2\}$ representing the stiffness of the first and second stories, respectively, that yield minimum IDR. Figure 4 below illustrates the results. Halton sets of up to 10000 points were necessary to smoothly fit the GP in two dimensions, yet even this number does not sufficiently cover higher-dimensional problems. Since it is not feasible to compute a larger training set given the computation time of the objective function, other methods had to be considered for high-dimensional problems.

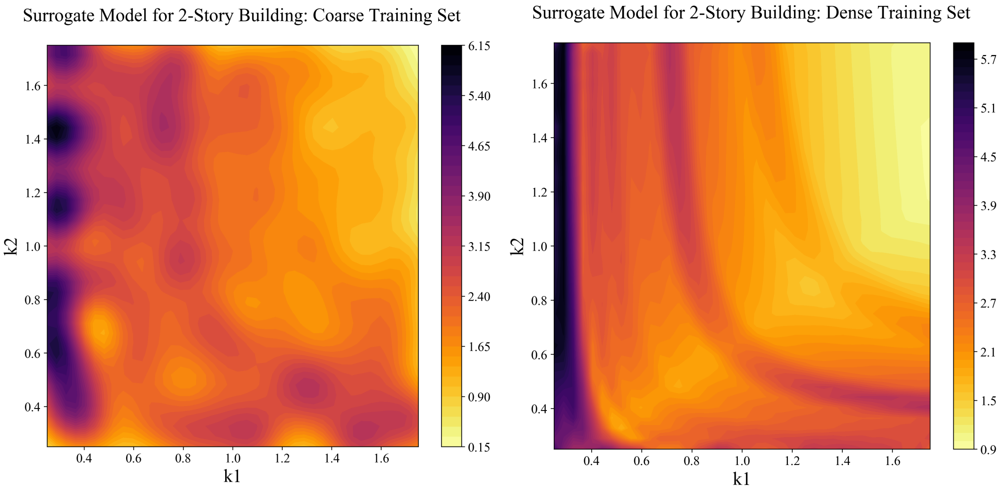


Fig. 4 The GP surrogate model trained using 100 Halton points (left) does not sufficiently represent the objective function. A large number of training points (right) is required to smoothly fit the objective function.

C. Differential Evolution

As the Gaussian Processes strategy struggled with the higher number of dimensions, the more efficient way to cover the majority of the design space in high-dimensional problems is by using a stochastic population method. One of the simplest evolutionary algorithms, Differential Evolution (DE), was selected to initially test the feasibility of this approach using `scipy` Python library [6]. This algorithm attempts to explore the design space and find a better individual by the crossover operation with a simple recombination rule, bounded by the specified constraints. Different variations of the recombination strategies exist, however, the most successful approach was found to be *best1bin*, formulated as follows:

$$x_{mut} = x_{best} + F \cdot (x_{r2} - x_{r3}) \tag{10}$$

The mutant individual (x_{mut}) is found by the mix of the best individual observed so far and the vector difference between two randomly selected points weighted with parameter F. This parameter is set to be randomly selected between 0.5 and 1.0 for each recombination. A binomial crossover operation is then performed between the target and mutant individuals, where the replacement of each component of the target vector has a probability of 0.7. Convergence plots for three different iterations are illustrated on Figure 5, where it can be observed that the algorithm provides a similar level of performance to that of the MADS approach, converging to several local minima.

The convergence of the DE algorithm is slow and inconsistent, even with the reasonably small $2 * N$ population size used for the examples shown on the plot. Since the accuracy of the optima reached with any evolutionary population approach is directly influenced by the population size, the dimensionality of the considered problem with DE is limited. As a result, in order to accelerate the convergence of larger populations, a more elaborate evolutionary approach is considered in the next section.

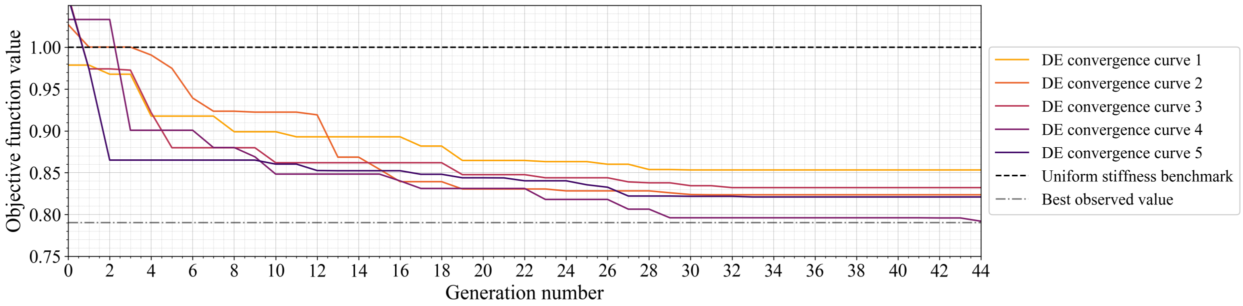


Fig. 5 Convergence examples for a 5-story building using the Differential Evolution algorithm

D. Genetic Algorithms

A genetic algorithm (GA) is a type of population method using biology-inspired schemes to select, cross, and mutate the design points which are a part of the population [3]. GAs are efficient methods for high-dimensional objectives with many local minima, such as the one considered in this paper. Each *chromosome* is a design point $\mathbf{k} = \{k_1, k_2, \ldots, k_N\}$ consisting of the story stiffnesses. The GA was implemented using the package Distributed Evolutionary Algorithms in Python (DEAP) [7].

1. Initial Population

The individuals were generated uniformly at random such that the chromosomes fall within the feasible range, i.e. $k_{min} \leq k_i \leq k_{max}$. The performance of the GA improved with a larger number of individuals. A population size of 500 was used for the results discussed below as a compromise between performance and computation time.

2. Crossover

Crossover of chromosomes refers to the combination of parents to form children in the next generation [3]. Two-point crossover, uniform crossover, and simulated binary crossover were tested. Uniform crossover generated the best results by maintaining the diversity of the population. In this scheme, each bit of the chromosome has a prescribed probability of coming from either one of two parents [3]. The global probability of two parents crossing was set to $p_{\text{cross}} = 0.5$, and the independent probability of each bit being exchanged was $p_{\text{ind}} = 0.5$.

3. Mutation

Mutation is the ability for a chromosome to spontaneously develop new traits by changing the bits within, thereby allowing the GA to more robustly explore the design space [3]. A polynomial mutation scheme [7] changes the individual according to a polynomial function probability distribution with the lower and upper bounds determined by the constraints. The global probability of mutation was set to $p_{\text{mut}} = 0.5$, the independent probability of each bit being mutated was $p_{\text{ind}} = 0.3$, and the similarity to the original chromosome is $eta = 0.3$ (higher is more similar to the parent).

4. Selection

Selection is the approach used to bias the population towards the fittest individuals [3]. Tournament selection was used, where a tournament size of n chromosomes are compared and the fittest is selected. For a population size of 500 individuals, a tournament size of 10 was chosen to sufficiently bias fit individuals while maintaining a certain amount of genetic diversity.

5. Algorithm Performance

Similarly to the previous algorithms, performance of the GA approach was initially tested on a 5-story building using single-objective optimization (per Eq. 6). The results for the optimal stiffness distribution tended to agree with those obtained using MADS and DE optimization strategies. The convergence trends for the GA are shown in Figure 6, along with the uniform benchmark and the best observed value over all strategies for comparison. As noted previously, this algorithm allows to analyze larger population sizes and still converge in a relatively efficient time. Hence, this approach was further expanded to a multi-objective optimization (per Eq. 7) of the stiffness distribution of a 10-story building.

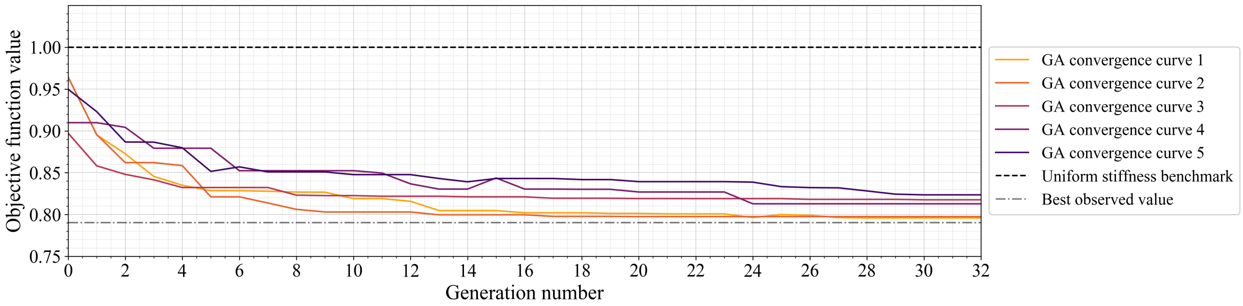


Fig. 6 Convergence examples for a 5-story building using the Genetic Algorithm

E. Hybrid Method using Local search

Once the potential location of the global minima had been identified using the above optimization strategies, a local search was performed to find the best performing design point within the identified local region. A direct Nelder-Mead Simplex (NMS) method was implemented using the algorithm from the `scipy` package [6] in Python. The Nelder-Mead method performs an unbounded optimization and, as a result, may produce design points that violate the constraints if initiated at a point close to the constraint boundary. In order to avoid the potential constraint violation, a simple penalty was applied to the objective function, formulated as the weighted count of violated constraints with the weight ρ set to 1:

$$p_{count}(x) = \rho \cdot \left[\sum_i (k_i > k_{max}) + \sum_i (k_i < k_{min}) \right] \tag{11}$$

IV. Results

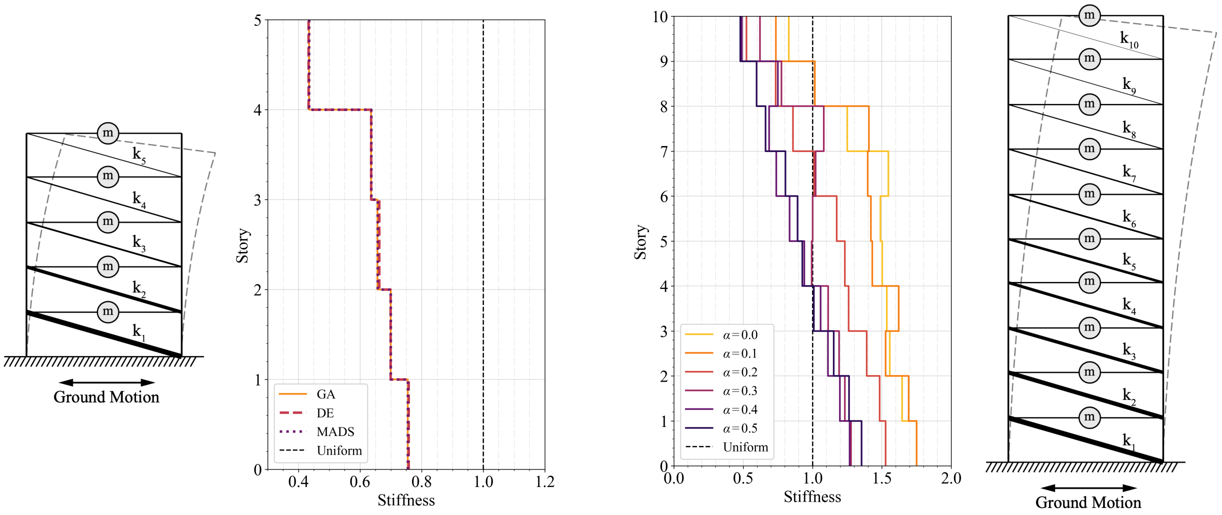
The optimal stiffness distribution decreases from the base of the structure to the roof. All optimization methods and all building heights display this trend. In the case of the five-story building (Figure 7a), all optimization strategies yielded nearly identical results (Table 1), reinforcing the belief that the obtained distribution is a global optimum. The largest story stiffness is $k_1 \approx 0.76$ at the base, and this decreases to the lowest story stiffness of $k_5 \approx 0.43$ at the top level. The decrease in stiffness becomes more extreme in the upper stories of the building. This distribution produces a maximum IDR which is 21% lower than a structure which has the same stiffness at all stories. It is worth noting that, for this structure, optimal stiffness at all stories is decreased from the base case.

Method	k_1	k_2	k_3	k_4	k_5	$f(k)$
MADS + NMS	0.7553	0.6995	0.6568	0.6362	0.4345	0.7902
DE + NMS	0.7577	0.6987	0.6624	0.6360	0.4325	0.7911
GA + NMS	0.7553	0.6997	0.6567	0.6360	0.4346	0.7903
Uniform Stiffness	1.0	1.0	1.0	1.0	1.0	1.0

Table 1 Summary of optimal stiffness distributions for a 5-story building

For a 10-story building case study, only the GA with local search strategy was utilized, since the other methods did not perform well in high dimensions. In this example, the multiobjective problem was considered (Eq. 7) in order to prioritize solutions with smaller stiffnesses when α is larger. The solutions adhered to the expected behavior: the optimal stiffness distributions still decrease as you go up the building, and the stiffness values are globally smaller for solutions with higher α. The resulting solutions are visible in Figure 7b, where it can be observed that the stiffness distributions tend to be more linear with larger α. The reason for this is that a larger α decreases the relative importance of the peak IDR, thereby favoring a more even distribution of stiffnesses along the structure.

On the other hand, smaller α tend to concentrate more stiffness at the bottom half of the structure. In contrast with the 5-story structure, here, stiffness values increase from the base case for at least half of the building stories, depending on the value of α. This is due to the fact that taller structures have higher shear forces carried by the bottom floors, which directly influence inter-story drift ratios.



(a) 5-story building, different methods, $\alpha = 0$

(b) 10-story building, GA method, different α

Fig. 7 Optimal stiffness distributions using various optimization methods and α parameter values

V. Conclusion

The work presented suggests that engineers can design a more efficient building by using a maximum stiffness in the first story and reducing the stiffness moving up the building. Practically, this would mean using smaller sections for structural bracing members or using fewer braces in the higher stories. In fact, this is commonly practiced in highrise design. The most commonly observed stiffness distribution in skyscrapers is parabolically decreasing. The solution therefore conforms to expectations from standard practice.

The optimal stiffness distribution is subject to change for a building subjected to a different earthquake. In this paper, the optimal distributions were derived using only the El Centro (1940) ground motion. In order to create a robust solution, the objective function should be calculated based on the building's average response to many different ground motions. In addition to that, many other building performance parameters have to be considered for a more holistic optimization approach, such as maximum roof displacements, shear forces and floor accelerations.

References

[1] American Society of Civil Engineers, *Minimum Design Loads and Associated Criteria for Buildings and Other Structures*, 2017. https://doi.org/10.1061/9780784414248.

[2] Chopra, A., *Dynamics of Structures: Theory and Applications to Earthquake Engineering*, Prentice-Hall international series in civil engineering and engineering mechanics, Prentice Hall, 2000. URL https://books.google.com/books?id=MzexQgAACAAJ.

[3] Kochenderfer, M., and Wheeler, T., *Algorithms for Optimization*, 2019.

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[5] Feinberg, J., and Langtangen, H. P., "Chaospy: An open source tool for designing methods of uncertainty quantification," *Journal of Computational Science*, Vol. 11, 2015, pp. 46 – 57. https://doi.org/https://doi.org/10.1016/j.jocs.2015.08.008, URL http://www.sciencedirect.com/science/article/pii/S1877750315300119.

[6] Virtanen, P., Gommers, R., et al, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, Vol. 17, 2020, pp. 261–272. https://doi.org/https://doi.org/10.1038/s41592-019-0686-2.

[7] Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., and Gagné, C., "DEAP: Evolutionary Algorithms Made Easy," *Journal of Machine Learning Research*, Vol. 13, 2012, pp. 2171–2175.