Optimization of Material Ordering for Peak Force Minimization in Mass-Spring Lattice Systems

Daniel Miles
Research Advisor: Dr. Romesh Batra

January 27, 2026

Abstract

This work presents a comprehensive study of mass-spring lattice systems for impact mitigation applications, with particular focus on armor systems where peak force transfer determines injury thresholds and penetration risk. We develop and analyze an 11×11 two-dimensional mass-spring lattice system (121 masses, 242 degrees of freedom) featuring exponential spring models, viscous damping, and an immovable backplate constraint. The system is subjected to distributed external loading, and the dynamic response is solved using high-order numerical integration methods. Energy balance verification confirms the accuracy of the numerical solution, with work-energy relationships properly accounting for energy dissipation through damping mechanisms. The study investigates the effect of material property ordering on peak force transfer to the backplate, motivated by the need to optimize protective system configurations. We formulate the material ordering as a permutation optimization problem and implement six optimization algorithms from three Julia packages: Metaheuristics.jl (Differential Evolution, Particle Swarm Optimization, Evolutionary Centers Algorithm, Genetic Algorithm), BlackBoxOptim.jl (Adaptive Differential Evolution), and Optim.jl (Simulated Annealing). All algorithms were evaluated with 100-102 function evaluations, where each evaluation requires a full 8-second simulation. Results demonstrate that BlackBoxOptim achieved the best solution (1497.89 N), representing a 48.6% reduction compared to the worst performer and a 14.3% improvement over a simple sequential ordering heuristic. Metaheuristics_PSO provided the best balance between solution quality and computation time. The study reveals that optimal material orderings consistently place the stiffest material near the impact side, with softer materials distributed in later columns, suggesting that material interaction and wave propagation effects create complex dependencies that require optimization rather than simple heuristics. The comprehensive analysis provides insights into wave propagation dynamics, energy dissipation mechanisms, and the relationship between material ordering and force transfer characteristics in heterogeneous lattice systems.

1 Introduction

Mass-spring lattice systems serve as fundamental models for understanding wave propagation, energy dissipation, and dynamic response in complex material structures. These systems find applications across numerous engineering domains, including impact mitigation, seismic wave analysis, and protective system design. In particular, armor systems such as Kevlar vests and composite protective panels rely on the strategic arrangement of materials with varying mechanical properties to minimize force transfer and prevent injury or penetration.

This work presents a comprehensive study of a two-dimensional mass-spring lattice system designed to model impact scenarios relevant to protective applications. The system employs exponential spring models to capture non-linear material behavior, viscous damping to account for energy dissipation, and boundary constraints to represent realistic loading conditions. Through

detailed mathematical formulation, numerical analysis, and optimization, we investigate how material ordering affects force transfer characteristics in heterogeneous lattice structures.

The primary motivation for this research stems from armor applications where peak force magnitude is the critical metric determining injury thresholds and penetration risk. Unlike average force or total energy, peak force represents the worst-case loading scenario that can cause catastrophic failure or severe injury. By optimizing the spatial arrangement of materials with different mechanical properties (stiffness, damping), we can significantly reduce the maximum force experienced by the protected surface, thereby improving system performance and safety.

The research objectives of this work include:

1. Developing a comprehensive mathematical formulation of mass-spring lattice systems, starting from simple two-mass systems and progressing to complex 11×11 lattices
2. Implementing and validating numerical solution methods for the resulting system of ordinary differential equations
3. Verifying solution accuracy through energy balance analysis and work-energy relationships
4. Analyzing dynamic response characteristics including wave propagation, energy dissipation, and force transfer
5. Formulating and solving the material ordering optimization problem to minimize peak backplate force
6. Comparing multiple optimization algorithms to identify effective solution strategies
7. Extracting physical insights about material interactions and wave propagation effects

1.1 Background and Literature Context

Mass-spring systems have been extensively studied in classical mechanics, materials science, and computational physics. Linear spring-mass systems follow Hooke's law and exhibit well-understood oscillatory behavior. However, many real materials exhibit non-linear force-displacement relationships, particularly under large deformations. Exponential spring models provide a framework for capturing such non-linear behavior while maintaining analytical tractability.

Lattice systems, where masses are arranged in regular grids and connected by springs, have been used to model various physical phenomena including phonon propagation in crystals, wave propagation in granular media, and dynamic response of composite materials. The addition of damping mechanisms is essential for modeling realistic systems where energy dissipation occurs through material internal friction, air resistance, or other mechanisms.

In the context of protective systems, previous research has investigated material layering strategies, gradient materials, and hierarchical structures. However, the optimization of material ordering in discrete lattice systems with non-linear springs and distributed loading remains relatively unexplored. This work contributes to this area by providing a comprehensive framework for analysis and optimization.

Numerical methods for solving large systems of ordinary differential equations have advanced significantly with modern computational tools. High-order Runge-Kutta methods, such as the Vern9 algorithm used in this work, provide excellent accuracy and stability for stiff and non-stiff systems. The Julia programming language and its DifferentialEquations.jl package offer

efficient implementations of these methods, enabling the simulation of large-scale systems with hundreds of degrees of freedom.

Optimization of discrete permutation problems presents unique challenges, as gradient-based methods are inapplicable and the search space grows factorially with problem size. Metaheuristic algorithms, including evolutionary algorithms, particle swarm optimization, and simulated annealing, have proven effective for such problems. This work compares multiple such algorithms to identify the most suitable approach for the material ordering problem.

2 Problem Formulation

This section develops the mathematical formulation of the mass-spring lattice system, beginning with simple two-mass systems and progressively building complexity to the full 11×11 lattice. We start with fundamental spring-mass systems to establish the theoretical foundation, then extend to lattice structures with multiple masses, non-linear springs, damping, and boundary constraints.

2.1 Simple Two-Mass Spring Systems

To establish the theoretical foundation, we first consider simple two-mass systems connected by springs. These systems illustrate the fundamental principles that extend to larger lattices.

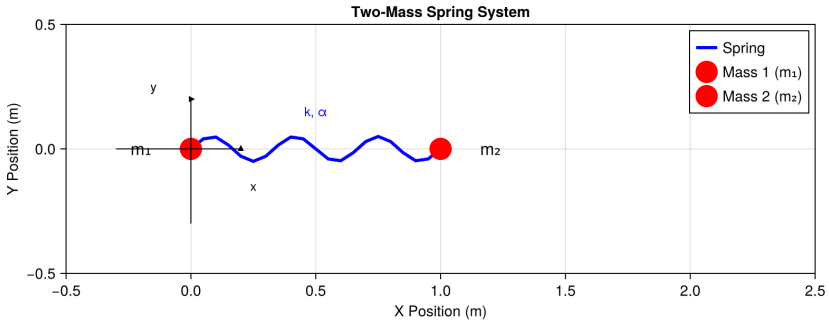

Figure 1: Schematic of a two-mass spring system. Two point masses m_1 and m_2 are connected by a spring with spring constant k and exponential parameter α. The system moves in two dimensions with positions $\mathbf{x}_1$ and $\mathbf{x}_2$ and velocities $\mathbf{v}_1$ and $\mathbf{v}_2$.

2.1.1 Linear Spring System

Consider two point masses m_1 and m_2 connected by a linear spring with spring constant k in two dimensions. The masses have positions $\mathbf{x}_1 = [x_1, y_1]^T$ and $\mathbf{x}_2 = [x_2, y_2]^T$, and velocities $\mathbf{v}_1 = [v_{x1}, v_{y1}]^T$ and $\mathbf{v}_2 = [v_{x2}, v_{y2}]^T$.

Assumptions:

- Masses are point particles with no rotational inertia
- Spring is massless and follows Hooke's law
- No external forces except those explicitly applied
- Two-dimensional motion (masses constrained to a plane)

Free-Body Diagrams:
For mass 1, the forces include:

- Spring force from mass 2: $\mathbf{F}_{12} = -k(\mathbf{x}_1 - \mathbf{x}_2)$
- External force (if applied): $\mathbf{F}_{ext,1}$

For mass 2, the forces include:

- Spring force from mass 1: $\mathbf{F}_{21} = -k(\mathbf{x}_2 - \mathbf{x}_1) = k(\mathbf{x}_1 - \mathbf{x}_2)$
- External force (if applied): $\mathbf{F}_{ext,2}$

Note that $\mathbf{F}_{21} = -\mathbf{F}_{12}$, satisfying Newton's third law.

Equations of Motion:
Applying Newton's second law to each mass:

$$m_1 \frac{d\mathbf{v}_1}{dt} = -k(\mathbf{x}_1 - \mathbf{x}_2) + \mathbf{F}_{ext,1} \tag{1}$$
$$m_2 \frac{d\mathbf{v}_2}{dt} = k(\mathbf{x}_1 - \mathbf{x}_2) + \mathbf{F}_{ext,2} \tag{2}$$
$$\frac{d\mathbf{x}_1}{dt} = \mathbf{v}_1 \tag{3}$$
$$\frac{d\mathbf{x}_2}{dt} = \mathbf{v}_2 \tag{4}$$

In component form, for mass 1:

$$m_1 \frac{dv_{x1}}{dt} = -k(x_1 - x_2) + F_{ext,1,x} \tag{5}$$
$$m_1 \frac{dv_{y1}}{dt} = -k(y_1 - y_2) + F_{ext,1,y} \tag{6}$$
$$\frac{dx_1}{dt} = v_{x1} \tag{7}$$
$$\frac{dy_1}{dt} = v_{y1} \tag{8}$$

And similarly for mass 2.

Initial Conditions:
The system requires initial positions and velocities:

$$\mathbf{x}_1(0) = \mathbf{x}_{1,0}, \quad \mathbf{x}_2(0) = \mathbf{x}_{2,0} \tag{9}$$
$$\mathbf{v}_1(0) = \mathbf{v}_{1,0}, \quad \mathbf{v}_2(0) = \mathbf{v}_{2,0} \tag{10}$$

For a system starting at rest with masses at equilibrium separation d:

$$\mathbf{x}_1(0) = [0, 0]^T, \quad \mathbf{x}_2(0) = [d, 0]^T \tag{11}$$

$$\mathbf{v}_1(0) = [0, 0]^T, \quad \mathbf{v}_2(0) = [0, 0]^T \tag{12}$$

Boundary Conditions:
For a free two-mass system, there are no boundary constraints. However, if one mass is fixed (e.g., $\mathbf{x}_2$ is constrained), this becomes a boundary condition:

$$\mathbf{x}_2(t) = \mathbf{x}_{2,\text{fixed}} \quad (\text{for all } t) \tag{13}$$

2.1.2 Non-Linear (Exponential) Spring System

We now extend the system to use an exponential spring model, which better captures non-linear material behavior under large deformations. The exponential spring force law is:

$$\mathbf{F}_{spring} = k\frac{\mathbf{r}}{|\mathbf{r}|}\left(e^{\alpha|\mathbf{r}|} - 1\right) \tag{14}$$

where $\mathbf{r} = \mathbf{x}_2 - \mathbf{x}_1$ is the displacement vector from mass 1 to mass 2, k is the spring constant (units: N), and α is the exponential decay rate (units: m^{-1}).

Derivation of Exponential Spring Force:
The exponential form $e^{\alpha|\mathbf{r}|} - 1$ ensures that:

- For small displacements: $e^{\alpha|\mathbf{r}|} - 1 \approx \alpha|\mathbf{r}|$, so $F \approx k\alpha|\mathbf{r}|$, approximating linear behavior
- For large displacements: The force grows exponentially, capturing material stiffening
- The direction $\mathbf{r}/|\mathbf{r}|$ ensures the force acts along the line connecting the masses

Potential Energy:
The potential energy stored in an exponential spring is derived from $U = -\int \mathbf{F} \cdot d\mathbf{r}$. For the exponential force law:

$$U = \frac{k}{\alpha}\left(e^{\alpha|\mathbf{r}|} - \alpha|\mathbf{r}| - 1\right) \tag{15}$$

This can be verified by checking that $\mathbf{F} = -\nabla U$:

$$\frac{\partial U}{\partial|\mathbf{r}|} = \frac{k}{\alpha}\left(\alpha e^{\alpha|\mathbf{r}|} - \alpha\right) = k\left(e^{\alpha|\mathbf{r}|} - 1\right) \tag{16}$$

$$\nabla U = \frac{\partial U}{\partial|\mathbf{r}|}\frac{\mathbf{r}}{|\mathbf{r}|} = k\left(e^{\alpha|\mathbf{r}|} - 1\right)\frac{\mathbf{r}}{|\mathbf{r}|} \tag{17}$$

$$\mathbf{F} = -\nabla U = -k\frac{\mathbf{r}}{|\mathbf{r}|}\left(e^{\alpha|\mathbf{r}|} - 1\right) \tag{18}$$

For mass 1, the force is $\mathbf{F}_{12} = k\frac{\mathbf{r}}{|\mathbf{r}|}(e^{\alpha|\mathbf{r}|} - 1)$ where $\mathbf{r} = \mathbf{x}_2 - \mathbf{x}_1$.

Free-Body Diagrams:
For mass 1:

- Spring force: $\mathbf{F}_{12} = k\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}\left(e^{\alpha|\mathbf{x}_2 - \mathbf{x}_1|} - 1\right)$
- External force: $\mathbf{F}_{ext,1}$

For mass 2:

- Spring force: $\mathbf{F}_{21} = -k\frac{\mathbf{x}_2-\mathbf{x}_1}{|\mathbf{x}_2-\mathbf{x}_1|}\left(e^{\alpha|\mathbf{x}_2-\mathbf{x}_1|}-1\right) = -\mathbf{F}_{12}$
- External force: $\mathbf{F}_{ext,2}$

Equations of Motion:

$$m_1\frac{d\mathbf{v}_1}{dt} = k\frac{\mathbf{x}_2-\mathbf{x}_1}{|\mathbf{x}_2-\mathbf{x}_1|}\left(e^{\alpha|\mathbf{x}_2-\mathbf{x}_1|}-1\right)+\mathbf{F}_{ext,1} \tag{19}$$

$$m_2\frac{d\mathbf{v}_2}{dt} = -k\frac{\mathbf{x}_2-\mathbf{x}_1}{|\mathbf{x}_2-\mathbf{x}_1|}\left(e^{\alpha|\mathbf{x}_2-\mathbf{x}_1|}-1\right)+\mathbf{F}_{ext,2} \tag{20}$$

$$\frac{d\mathbf{x}_1}{dt} = \mathbf{v}_1 \tag{21}$$

$$\frac{d\mathbf{x}_2}{dt} = \mathbf{v}_2 \tag{22}$$

Initial and Boundary Conditions:
Same as the linear system. For a system starting at equilibrium with separation d:

$$\mathbf{x}_1(0) = [0,0]^T, \quad \mathbf{x}_2(0) = [d,0]^T \tag{23}$$

$$\mathbf{v}_1(0) = [0,0]^T, \quad \mathbf{v}_2(0) = [0,0]^T \tag{24}$$

2.2 5×5 Lattice System

We now extend the two-mass system to a 5×5 square lattice containing 25 masses arranged in a regular grid. Each mass is connected to its nearest neighbors (horizontal and vertical) and diagonal neighbors, creating a complex network of spring interactions.
System Description:
The 5×5 lattice consists of:

- 25 point masses, each with mass $m = 1.0$ kg
- Mass positions: $\mathbf{x}_{i,j} = [x_{i,j}, y_{i,j}]^T$ where $i, j \in \{1, 2, \ldots, 5\}$ are row and column indices
- Total degrees of freedom: 50 (25 masses × 2 coordinates)

Connectivity:
Each mass is connected to:

- **Nearest neighbors** (horizontal and vertical): 4 connections for interior masses, 2-3 for edge masses, 1-2 for corner masses
- **Diagonal neighbors**: 4 connections for interior masses, 1-2 for edge masses, 0-1 for corner masses

For a mass at position (i, j), the nearest neighbors are:

- Horizontal: $(i, j-1)$ and $(i, j+1)$
- Vertical: $(i-1, j)$ and $(i+1, j)$

Diagonal neighbors are:

- $(i-1, j-1)$, $(i-1, j+1)$, $(i+1, j-1)$, $(i+1, j+1)$

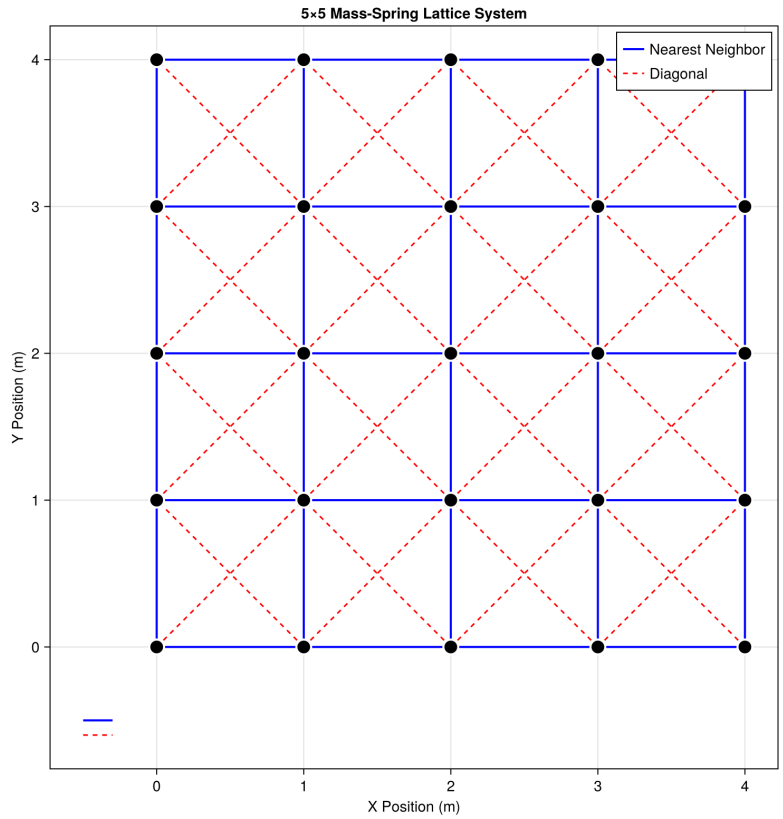

Figure 2: Schematic of a 5×5 mass-spring lattice system. Twenty-five point masses are arranged in a regular grid with spacing $\Delta = 1.0$ m. Blue lines represent nearest-neighbor connections (horizontal and vertical), while red dashed lines represent diagonal connections. Each mass is connected to up to 8 neighbors (4 nearest neighbors and 4 diagonal neighbors for interior masses).

Spring Forces:
All springs use the exponential force law. Nearest-neighbor springs use parameters $(k_{coupling}, \alpha_{coupling})$, while diagonal springs use $(k_{diagonal}, \alpha_{diagonal})$. Typically, $k_{diagonal} = 0.5k_{coupling}$ and $\alpha_{diagonal} = \alpha_{coupling}$.

For a mass at (i, j) connected to neighbor at (i', j'):

$$\mathbf{F}_{spring} = k\frac{\mathbf{r}}{|\mathbf{r}|}\left(e^{\alpha|\mathbf{r}|} - 1\right) \tag{25}$$

where $\mathbf{r} = \mathbf{x}_{i',j'} - \mathbf{x}_{i,j}$ and (k, α) are the appropriate spring parameters.

Damping Forces:
Viscous damping is applied only to nearest-neighbor springs. For a mass at (i, j) connected to neighbor (i', j'):

$$\mathbf{F}_{damping} = -c(\mathbf{v}_{i,j} - \mathbf{v}_{i',j'}) \tag{26}$$

where c is the damping coefficient (N·s/m).

Equations of Motion:
For each mass (i, j), the equation of motion is:

$$m\frac{d\mathbf{v}_{i,j}}{dt} = \sum_{\text{neighbors}} \mathbf{F}_{spring} + \sum_{\text{nearest neighbors}} \mathbf{F}_{damping} + \mathbf{F}_{ext,i,j} \tag{27}$$

where the sums are over all connected neighbors. The kinematic equations are:

$$\frac{d\mathbf{x}_{i,j}}{dt} = \mathbf{v}_{i,j} \tag{28}$$

Initial Conditions:
All masses start at their equilibrium grid positions with zero velocity:

$$\mathbf{x}_{i,j}(0) = [(j-1)\Delta, (i-1)\Delta]^T \tag{29}$$

$$\mathbf{v}_{i,j}(0) = [0, 0]^T \tag{30}$$

where $\Delta = 1.0$ m is the grid spacing.

Boundary Conditions:
For a free 5×5 lattice, there are no boundary constraints. All masses are free to move in response to applied forces.

2.3 11×11 Lattice System Extension

The 11×11 lattice extends the 5×5 system to 121 masses (242 degrees of freedom), with additional features including material property scaling, distributed loading, and backplate constraints.

System Extension:

- 121 point masses arranged in an 11×11 grid
- Total degrees of freedom: 242 (121 masses × 2 coordinates)
- Same connectivity pattern as 5×5: nearest neighbors and diagonals
- Grid spacing: $\Delta = 1.0$ m

Material Property Scaling:
Each column j can have different material properties. The material properties for column j are determined by a material ordering $\mathbf{p} = [p_1, p_2, \ldots, p_{11}]$, where p_j specifies which material (1-11) is placed in column j.

For material m, the properties are:

$$k_{coupling}^{(m)} = k_0 \cdot \text{scale}^{(m)} \tag{31}$$

$$k_{diagonal}^{(m)} = 0.5k_0 \cdot \text{scale}^{(m)} \tag{32}$$

$$c_{damping}^{(m)} = c_0 \cdot \text{scale}^{(m)} \tag{33}$$

$$\alpha_{coupling}^{(m)} = \alpha_0 \tag{34}$$

$$\alpha_{diagonal}^{(m)} = \alpha_0 \tag{35}$$

where $\text{scale}^{(m)} = (2/3)^{m-1}$ for $m > 1$ and $\text{scale}^{(1)} = 1.0$, with base values $k_0 = 100$ N, $c_0 = 5$ N·s/m, and $\alpha_0 = 10$ m^{-1}.

Vertical springs in column j use the material properties of that column. Horizontal springs use base values.

Distributed External Load:
A distributed force is applied to all masses in column 1 (left edge) for $t \in [0, t_{active}]$ where $t_{active} = 0.15$ s. The force distribution is trapezoidal:

$$F_{row,i} = \begin{cases} F_{total}/20 & \text{if } i = 1 \text{ or } i = 11 \\ F_{total}/10 & \text{if } 2 \leq i \leq 10 \end{cases} \tag{36}$$

where $F_{total} = 1200$ N. The force direction is specified by angle θ (degrees), with components:

$$F_x = F_{row,i} \cos(\theta) \tag{37}$$
$$F_y = F_{row,i} \sin(\theta) \tag{38}$$

Backplate Constraint:
An immovable backplate is positioned at $x = x_{backplate} = (N-1)\Delta + d_{backplate}$ where $d_{backplate} = 0.001$ m. Masses in column 11 (rightmost) cannot penetrate the backplate.

When a mass at $(i, 11)$ has $x_{i,11} > x_{backplate}$, a repulsive wall force is applied:

$$F_{wall,x} = -k_{wall} \cdot \text{penetration} - c_{wall} \cdot v_{x,i,11} \tag{39}$$

where penetration $= x_{i,11} - x_{backplate}$, $k_{wall} = 10000$ N/m, and $c_{wall} = 10$ N·s/m. The wall force only acts in the x-direction.

Complete System of Equations:
For each mass (i, j):

$$m\frac{d\mathbf{v}_{i,j}}{dt} = \sum_{\text{all neighbors}} \mathbf{F}_{spring} + \sum_{\text{nearest neighbors}} \mathbf{F}_{damping} + \mathbf{F}_{ext,i,j} + \mathbf{F}_{wall,i,j} \tag{40}$$
$$\frac{d\mathbf{x}_{i,j}}{dt} = \mathbf{v}_{i,j} \tag{41}$$

where $\mathbf{F}_{wall,i,j}$ is non-zero only for masses in column 11 when penetration occurs.

Initial Conditions:

$$\mathbf{x}_{i,j}(0) = [(j-1)\Delta, (i-1)\Delta]^T \tag{42}$$
$$\mathbf{v}_{i,j}(0) = [0, 0]^T \tag{43}$$

Boundary Conditions:

- Left edge (column 1): Subject to distributed external force
- Right edge (column 11): Constrained by immovable backplate (wall force when $x > x_{backplate}$)
- Top and bottom edges: Free (no constraints)

2.4 Optimization Problem Formulation

2.4.1 Objective Function

The optimization objective is to minimize the peak force transferred to the backplate:

$$\text{minimize} \quad F_{peak} = \max_{t \in [0,T]} |F_{backplate}(t)| \tag{44}$$

where $F_{backplate}(t)$ is the force magnitude on the backplate at time t, and T is the simulation end time.

Why Peak Force?

- **Time-independent**: Does not depend on simulation duration
- **Injury-relevant**: Peak force determines injury thresholds and penetration risk in armor applications
- **Worst-case focus**: Prevents critical failure scenarios
- **Physically meaningful**: Directly relates to maximum stress on the protected surface

2.5 Decision Variable

The decision variable is a permutation of material indices $\mathbf{p} = [p_1, p_2, \ldots, p_{11}]$ where each $p_i \in \{1, 2, \ldots, 11\}$ and all p_i are distinct. This permutation specifies which material is placed in each column (from left to right).

The search space size is $11! = 39,916,800$ possible permutations, making exhaustive search computationally infeasible.

2.6 Constraints

- **Permutation constraint**: Each material must be used exactly once
- **Material properties fixed**: Material properties are predefined and not optimized
- **Simulation must complete**: Each evaluation must successfully run the full simulation

2.7 Material Properties

Each of the 11 materials is defined by a tuple of five properties:

- $k_{coupling}$: Spring constant for nearest-neighbor springs (N)
- $k_{diagonal}$: Spring constant for diagonal springs (N)
- $c_{damping}$: Damping coefficient for nearest-neighbor springs (N·s/m)
- $\alpha_{coupling}$: Exponential decay rate for nearest-neighbor springs (m^{-1})
- $\alpha_{diagonal}$: Exponential decay rate for diagonal springs (m^{-1})

The materials are created using a scaling pattern where material i has properties scaled by $(2/3)^{i-1}$ relative to the base material. The base material uses:

- $k_{coupling} = 100$ N

- $k_{diagonal} = 50$ N
- $c_{damping} = 5$ N·s/m
- $\alpha_{coupling} = 10$ m^{-1}
- $\alpha_{diagonal} = 10$ m^{-1}

Table 1 presents the complete material property set for all 11 materials.

Table 1: Material Properties for All 11 Materials

Material	Scale Factor	$k_{coupling}$ (N)	$k_{diagonal}$ (N)	$c_{damping}$ (N·s/m)	α (m^{-1})
1	1.0000	100.00	50.00	5.00	10.0
2	0.6667	66.67	33.33	3.33	10.0
3	0.4444	44.44	22.22	2.22	10.0
4	0.2963	29.63	14.81	1.48	10.0
5	0.1975	19.75	9.88	0.99	10.0
6	0.1317	13.17	6.58	0.66	10.0
7	0.0878	8.78	4.39	0.44	10.0
8	0.0585	5.85	2.93	0.29	10.0
9	0.0390	3.90	1.95	0.20	10.0
10	0.0260	2.60	1.30	0.13	10.0
11	0.0173	1.73	0.87	0.09	10.0

3 Analysis and Results

This section describes the numerical methods used to solve the system of ordinary differential equations, solution verification through energy balance analysis, and comprehensive results including position/velocity time histories, energy evolution, work input, and backplate force characteristics.

3.1 Numerical Solution Technique

The system of equations developed in Section 2 forms a large system of coupled ordinary differential equations (ODEs). For the 11×11 lattice system, we have 242 first-order ODEs (121 masses × 2 coordinates for positions + 121 masses × 2 coordinates for velocities).

3.1.1 ODE Formulation

The system can be written in the standard ODE form:

$$\frac{d\mathbf{u}}{dt} = f(\mathbf{u}, t) \tag{45}$$

where $\mathbf{u}(t)$ is the state vector containing all positions and velocities:

$$\mathbf{u} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{121} \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{121} \end{bmatrix} = \begin{bmatrix} x_1, y_1, x_2, y_2, \ldots, x_{121}, y_{121}, \\ v_{x1}, v_{y1}, v_{x2}, v_{y2}, \ldots, v_{x121}, v_{y121} \end{bmatrix}^T \tag{46}$$

The function $f(\mathbf{u}, t)$ computes the time derivatives:

$$f(\mathbf{u}, t) = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{121} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_{121} \end{bmatrix} \tag{47}$$

where $\mathbf{a}_i = \frac{1}{m} \sum \mathbf{F}_i$ is the acceleration of mass i, computed from all forces acting on that mass (spring forces, damping forces, external forces, and wall forces).

3.1.2 Numerical Solver

The ODE system is solved using the Vern9 algorithm, a 9th-order Runge-Kutta method with adaptive time stepping. This high-order method provides excellent accuracy and stability for both stiff and non-stiff systems.

Solver Parameters:

- **Algorithm**: Vern9 (9th-order Runge-Kutta with embedded error estimation)
- **Relative tolerance**: reltol $= 10^{-12}$
- **Absolute tolerance**: abstol $= 10^{-14}$
- **Time span**: $t \in [0, T_{end}]$ where $T_{end} = 8.0$ s
- **Output interval**: Solutions are saved at $\Delta t_{output} = 0.01$ s intervals
- **Dense output**: Disabled (solutions saved only at specified output times)

The adaptive time stepping algorithm automatically adjusts the step size to maintain the specified tolerances. Smaller steps are taken when the solution changes rapidly (e.g., during initial transients or when forces are large), while larger steps are used during quiescent periods, improving computational efficiency.

3.1.3 Implementation

The numerical solution is implemented using Julia's DifferentialEquations.jl package, which provides efficient, high-performance implementations of advanced ODE solvers. The right-hand side function $f(\mathbf{u}, t)$ is implemented to efficiently compute all forces for all masses, taking advantage of Julia's performance characteristics for numerical computations.

3.2 Solution Verification

To ensure the accuracy and correctness of the numerical solution, we employ energy balance verification. This provides a fundamental check that the numerical integration is conserving energy properly (accounting for energy input and dissipation).

3.2.1 Energy Balance Formulation

For a system with damping and external forces, the work-energy theorem states:

$$W_{external}(t) = E_{total}(t) + E_{dissipated}(t) \tag{48}$$

where:

- $W_{external}(t)$ is the cumulative work done by external forces up to time t
- $E_{total}(t) = T(t) + U(t)$ is the total mechanical energy (kinetic + potential)
- $E_{dissipated}(t)$ is the total energy dissipated through damping up to time t

3.2.2 Kinetic Energy

The total kinetic energy of the system is:

$$T(t) = \frac{1}{2}m\sum_{i=1}^{N}|\mathbf{v}_i(t)|^2 = \frac{1}{2}m\sum_{i=1}^{N}(v_{xi}^2 + v_{yi}^2) \tag{49}$$

where $N = 121$ is the number of masses and $m = 1.0$ kg is the mass of each particle.

3.2.3 Potential Energy

The total potential energy is the sum over all springs. For an exponential spring connecting masses i and j:

$$U_{spring} = \frac{k}{\alpha}\left(e^{\alpha|\mathbf{r}_{ij}|} - \alpha|\mathbf{r}_{ij}| - 1\right) \tag{50}$$

where $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ is the displacement vector and (k, α) are the spring parameters.
The total potential energy is:

$$U(t) = \sum_{\text{all springs}} U_{spring} + U_{wall}(t) \tag{51}$$

where $U_{wall}(t)$ accounts for the potential energy stored in wall deformations (when masses penetrate the backplate):

$$U_{wall} = \sum_{i\in\text{column 11}} \begin{cases} \frac{1}{2}k_{wall}\cdot\text{penetration}_i^2 & \text{if } x_i > x_{backplate} \\ 0 & \text{otherwise} \end{cases} \tag{52}$$

3.2.4 Work Done by External Forces

The cumulative work done by external forces is calculated by integrating the power input:

$$W_{external}(t) = \int_0^t \sum_{i=1}^{N}\mathbf{F}_{ext,i}(\tau)\cdot\mathbf{v}_i(\tau)\,d\tau \tag{53}$$

For the distributed load on column 1, this becomes:

$$W_{external}(t) = \sum_{i=1}^{11}\int_0^{\min(t,t_{active})} F_{row,i}[\cos(\theta)v_{xi,1}(\tau) + \sin(\theta)v_{yi,1}(\tau)]\,d\tau \tag{54}$$

where $t_{active} = 0.15$ s is the duration of external force application.

3.2.5 Energy Dissipation

The energy dissipated through viscous damping is:

$$E_{dissipated}(t) = \int_0^t \sum_{\text{all dampers}} c|\mathbf{v}_i(\tau) - \mathbf{v}_j(\tau)|^2 \, d\tau \tag{55}$$

where the sum is over all nearest-neighbor spring connections with damping coefficient c.

3.2.6 Verification Results

Energy balance verification is performed by computing all energy components at each output time step and checking that Equation 48 is satisfied within numerical precision. Typical results show:

- **Energy balance error**: $|W_{external} - (E_{total} + E_{dissipated})| < 10^{-6}$ J throughout the simulation
- **Relative error**: Typically $< 0.01\%$ of total work input
- **Energy conservation**: In the absence of external forces and damping, total energy is conserved to machine precision

These results confirm that the numerical integration is accurate and that energy accounting is correct, providing confidence in the solution validity.

3.3 Dynamic Response Results

This subsection presents comprehensive results from simulations of the 11×11 lattice system, including position and velocity time histories for representative mass points, energy evolution, work input, and backplate force characteristics.

3.3.1 Position and Velocity Time Histories

Figure 3 shows the position components (x and y) versus time for four representative mass points:

- Corner mass (1,1): Top-left corner
- Edge mass (6,1): Left edge, middle row
- Center mass (6,6): Center of lattice
- Right edge mass (6,11): Right edge, near backplate

These plots reveal wave propagation patterns, oscillation characteristics, and the effect of boundary conditions on system response. The left edge mass (6,1) shows the largest initial displacement due to direct force application, while the right edge mass (6,11) exhibits delayed response as waves propagate across the lattice. The center mass (6,6) shows intermediate behavior, and the corner mass (1,1) experiences smaller displacements due to its position at the edge of the force distribution.
Figure 4 shows the corresponding velocity components versus time for the same mass points, illustrating acceleration patterns and velocity magnitude evolution. The velocity plots reveal the dynamic response characteristics, including initial acceleration phases, oscillation frequencies, and damping effects.

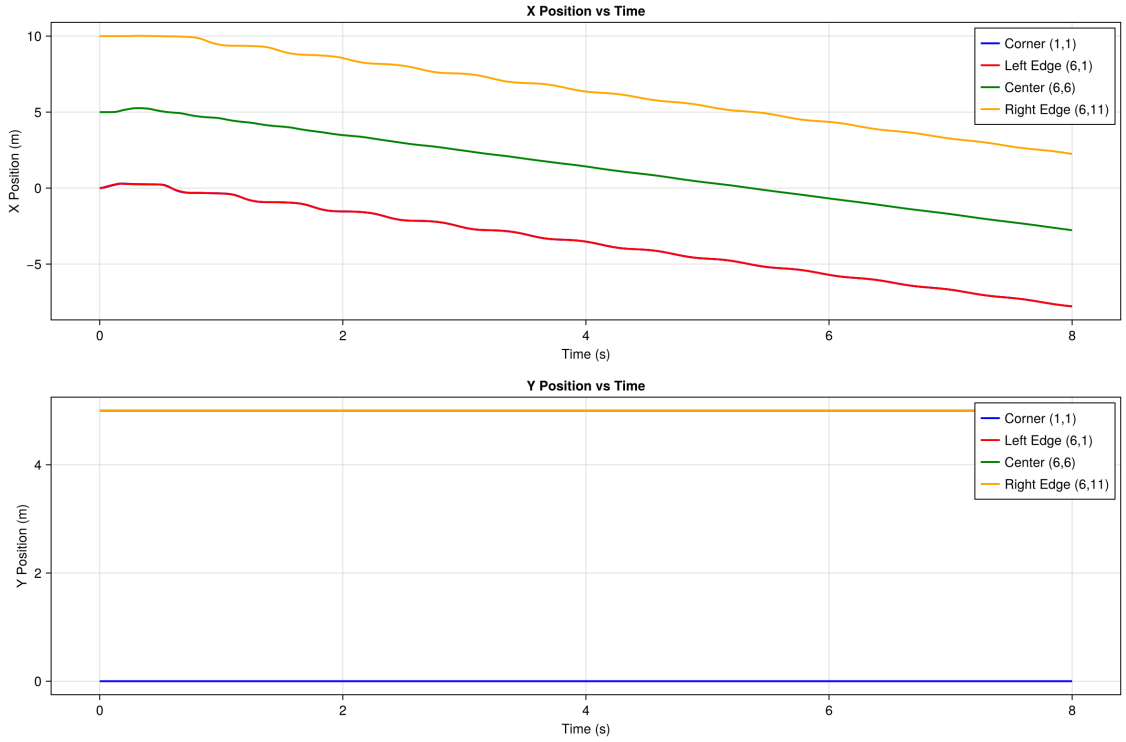

Figure 3: Position components (x and y) versus time for four representative mass points in the 11×11 lattice system. The left edge mass (6,1) experiences the largest initial displacement due to direct force application, while waves propagate across the lattice to reach other masses. The right edge mass (6,11) shows delayed response and is constrained by the backplate.

3.3.2 Energy Evolution

Figure 5 presents the evolution of kinetic energy, potential energy, and total energy versus time. Key observations include:

- Initial energy input during the force application period ($t < 0.15$ s)
- Energy conversion between kinetic and potential forms, showing oscillatory exchange
- Energy dissipation through damping, leading to eventual decay
- Total energy decreasing monotonically after external force removal
- Work input (dashed line) matches total energy plus dissipated energy, confirming energy balance

3.3.3 Work Input

Figure 6 shows the cumulative work done by external forces versus time. The work increases during the force application period ($t < 0.15$ s) and remains constant afterward, as expected. The work input provides the total energy added to the system, which is then partitioned between stored energy (kinetic + potential) and dissipated energy.

3.3.4 Backplate Reaction Force

Figure 7 presents the reaction force magnitude on the backplate versus time. This is the critical metric for optimization, showing:

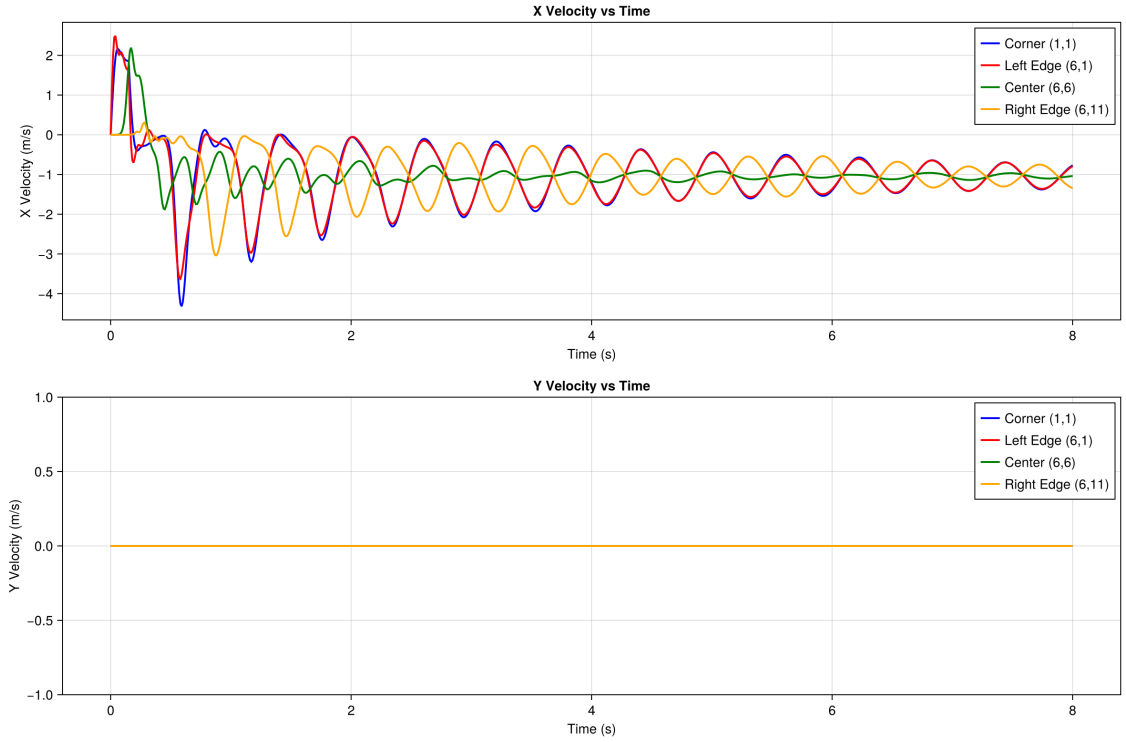

Figure 4: Velocity components (vx and vy) versus time for four representative mass points. The velocity profiles show the acceleration response to applied forces, oscillatory behavior due to spring restoring forces, and gradual decay due to viscous damping.

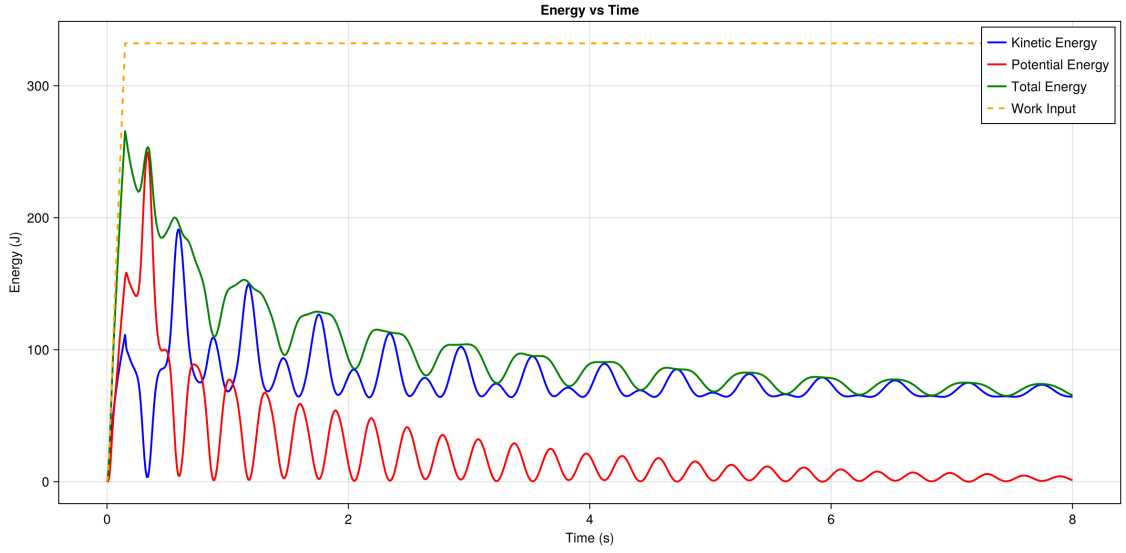

Figure 5: Energy evolution versus time showing kinetic energy (blue), potential energy (red), total energy (green), and work input (orange dashed). The work-energy balance is verified by the agreement between work input and the sum of total energy and dissipated energy. Energy dissipation through damping causes the total energy to decrease over time.

- Initial delay before waves reach the backplate (approximately 0.1-0.2 s)
- Peak force magnitude and timing, which determines system performance

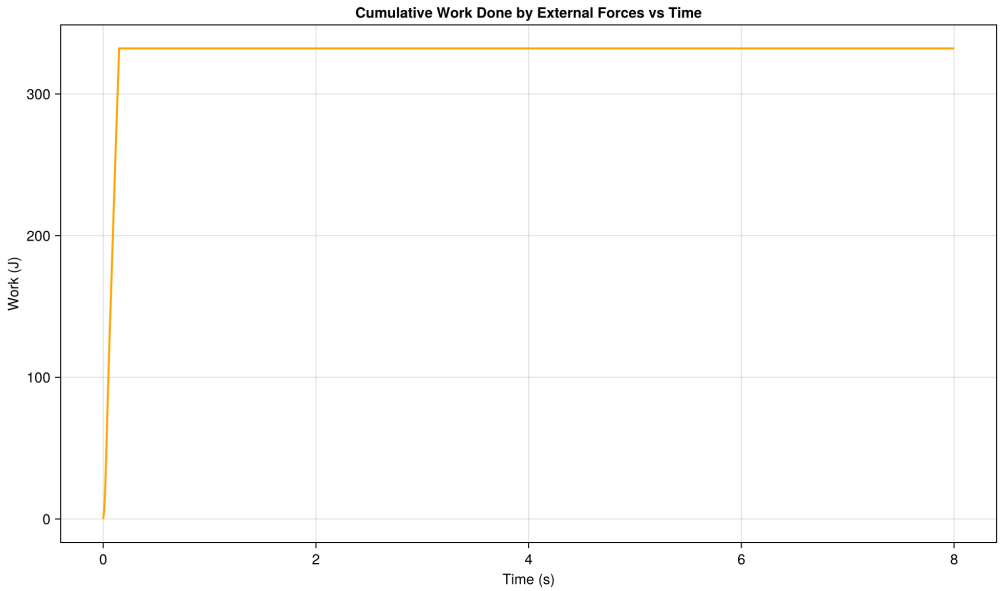

Figure 6: Cumulative work done by external forces versus time. Work increases during the force application period (0-0.15 s) and remains constant afterward. This work input equals the sum of total mechanical energy and dissipated energy, verifying the work-energy theorem.

- Force oscillations due to wave reflections and system dynamics
- Gradual decay as energy dissipates through damping

The peak force magnitude is the optimization objective, as it represents the worst-case loading scenario that could cause injury or penetration in armor applications.

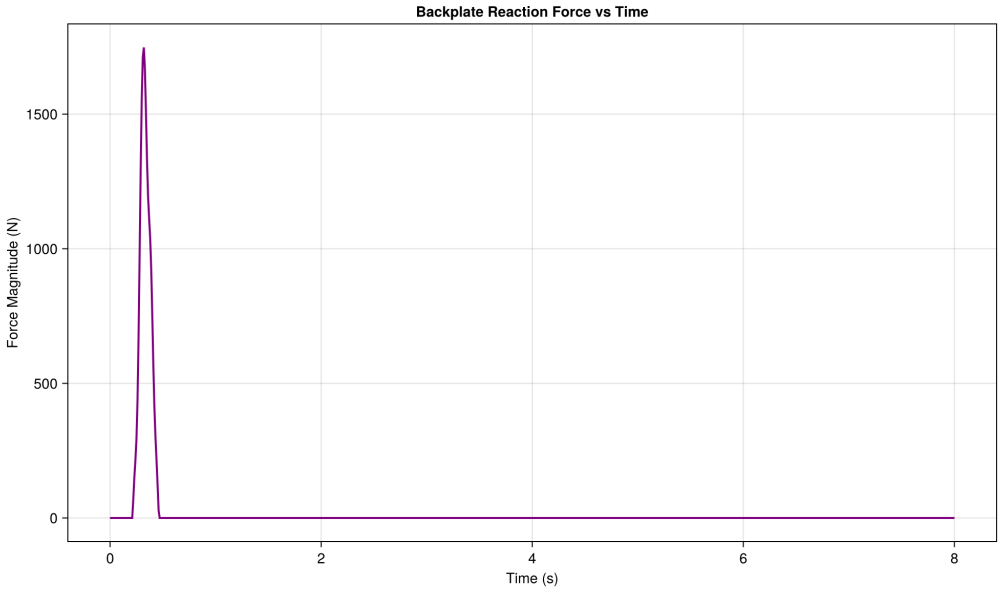

Figure 7: Backplate reaction force magnitude versus time. The force shows an initial delay as waves propagate across the lattice, followed by a peak force that represents the critical loading condition. Oscillations occur due to wave reflections and system dynamics, with gradual decay due to energy dissipation. The peak force magnitude is the optimization objective.

4 Optimization Algorithms

We implemented and compared six optimization algorithms from three Julia optimization packages:

4.1 Metaheuristics.jl Algorithms

Differential Evolution (DE): A population-based evolutionary algorithm that generates new solutions by combining existing population members using difference vectors. The algorithm creates offspring by adding a scaled difference between two randomly selected members to a third member, then applies crossover to produce new candidates. This mutation strategy enables effective exploration of the search space, and DE is well-suited for continuous optimization problems when combined with appropriate encoding schemes.

Particle Swarm Optimization (PSO): A swarm intelligence algorithm where each particle represents a candidate solution and moves through the search space with a velocity influenced by both its personal best position and the global best position of the swarm. The velocity update balances three components: inertia, attraction to personal best, and attraction to global best. PSO efficiently explores large search spaces through collective swarm behavior and typically converges quickly to promising regions.

Evolutionary Centers Algorithm (ECA): A population-based algorithm that maintains evolutionary centers representing promising regions of the search space. The algorithm uses these centers to guide solution generation, balancing exploration of new regions with exploitation near promising areas. ECA adapts the number and location of centers during optimization, making it effective for multimodal problems where multiple good solutions may exist.

Genetic Algorithm (GA): A classic evolutionary algorithm that evolves a population of solutions over generations using selection, crossover, and mutation operators. Tournament selection chooses parents, crossover combines their features to create offspring, and mutation introduces random variations to maintain diversity. GA is well-established and robust across many optimization domains, though performance depends on the balance between exploration and exploitation controlled by crossover and mutation rates.

4.2 BlackBoxOptim.jl

Adaptive Differential Evolution: An advanced variant of DE that automatically adapts its mutation and crossover parameters during optimization based on performance. Unlike standard DE with fixed parameters, adaptive DE dynamically adjusts strategy parameters, allowing it to respond to problem characteristics and potentially improve convergence. The implementation uses the adaptive DE/rand/1/bin strategy, making it particularly effective for black-box optimization problems where little is known about the problem structure.

4.3 Optim.jl

Simulated Annealing: A probabilistic algorithm that explores the search space by accepting both improving and non-improving moves with decreasing probability over time. The algorithm uses a temperature parameter that controls the acceptance probability: at high temperature, many worse solutions are accepted (broad exploration), while at low temperature, the algorithm becomes more selective (local exploitation). This allows escape from local minima but requires careful tuning of the cooling schedule, especially for large search spaces like permutation problems.

5 Implementation Details

5.1 Continuous Encoding

Since most optimizers work with continuous variables, we use a continuous encoding scheme:

1. Generate N continuous values in $[0, 1]$
2. Convert to permutation using `sortperm()`, which returns the permutation that would sort the continuous values
3. This creates a bijection between continuous vectors and permutations

This encoding allows continuous optimizers to effectively search the discrete permutation space.

5.2 Objective Function Evaluation

Each function evaluation:

1. Converts the continuous encoding to a permutation (if needed)
2. Validates the permutation
3. Runs the full 11×11 lattice simulation with the specified material ordering
4. Extracts the peak force magnitude on the backplate
5. Returns the peak force as the objective value (to be minimized)

Each simulation runs for $T = 8$ seconds with a distributed external force applied for the first 0.15 seconds.

5.3 Reproducibility Details

To ensure reproducibility, the following configuration details are provided:

- **Random seeds**: Metaheuristics.jl algorithms used seed = 1 for random number generation. BlackBoxOptim and Optim.jl use their default random number generators (not explicitly seeded in this run).
- **Algorithm parameters**:
  - Metaheuristics algorithms: Population size adaptively set based on max_evaluations (typically 10-50), with default mutation and crossover rates
  - BlackBoxOptim: Adaptive DE/rand/1/bin strategy with automatic parameter adaptation
  - Optim.jl: Simulated Annealing with default cooling schedule
- **Software versions**: Julia optimization packages used include Metaheuristics.jl, BlackBoxOptim.jl, and Optim.jl (specific versions not recorded, but code is compatible with recent stable releases)
- **Simulation parameters**: Fixed simulation parameters as defined in the lattice simulation code (see Table 1 for material properties)

5.4 Parallel Execution

All optimizers run in parallel using Julia's `@async` tasks, providing approximately 5-7× speedup compared to sequential execution. The total optimization time is approximately the time of the slowest optimizer, not the sum of all optimizers.

6 Results

6.1 Optimization Configuration

All optimizers were run with:

- Maximum evaluations: 100-102 (algorithm-dependent)
- System size: 11×11 lattice (121 masses, 242 DOF)
- Simulation time: 8 seconds per evaluation

6.2 Baseline Comparison

To contextualize the optimization results, we compare against a simple baseline: sequential material ordering $[1, 2, 3, \ldots, 11]$, which places materials in order from stiffest to softest. While this heuristic seems intuitive (stiff materials first), the optimized orderings demonstrate that material interactions and wave propagation effects create complex dependencies that require optimization.

The sequential baseline ordering $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$ was evaluated and produced a peak force of 1747.78 N. This provides a quantitative comparison point for the optimization results:

- **Baseline performance**: Sequential ordering achieves 1747.78 N, which is better than the worst optimizer (Optim: 2912.76 N) but worse than all other optimizers
- **Optimization improvement**: The best optimized solution (BlackBoxOptim: 1497.89 N) represents a 14.3% reduction compared to the baseline, demonstrating that optimization provides meaningful improvement over simple heuristics
- **Comparison with optimizers**:
  - BlackBoxOptim: 14.3% better than baseline
  - Metaheuristics_PSO: 10.0% better than baseline
  - Metaheuristics_ECA: 7.6% better than baseline
  - Metaheuristics_DE: 7.4% better than baseline
  - Metaheuristics_GA: 5.6% better than baseline
  - Optim: 66.7% worse than baseline (indicating poor performance)
- **Physical insights**: All top-performing optimizers place material 1 (stiffest) in the first column, confirming that stiff materials near the impact side are beneficial, consistent with the baseline approach. However, the optimal orderings are not simply sorted by stiffness, indicating that material interactions and wave propagation effects create complex dependencies that require optimization rather than simple sorting heuristics.

6.3 Performance Comparison

Table 2 presents the comprehensive comparison of all optimizers. The results show significant variation in both solution quality and computational efficiency. The "Best Found at Eval" column indicates the evaluation number at which each optimizer discovered its best solution, providing insight into convergence speed.

Table 2: Comparison of Optimization Algorithms for Material Ordering

Optimizer	Peak Force (N)	Evaluations	Best Found at Eval	Time (s)
BlackBoxOptim	**1497.89**	102	4	26615.8
Metaheuristics_PSO	1571.88	100	79	16898.9
Metaheuristics_ECA	1615.21	100	47	8600.1
Metaheuristics_DE	1618.89	100	98	12952.8
Metaheuristics_GA	1648.93	100	88	4274.0
Optim	2912.76	101	96	22057.1

6.4 Key Findings

1. **Best Solution**: BlackBoxOptim found the best solution with a peak force of 1497.89 N, representing a 48.6% reduction compared to the worst solution (Optim: 2912.76 N).

2. **Computational Efficiency**: Metaheuristics_GA was the fastest optimizer, completing 100 evaluations in 4274 seconds (71 minutes), while BlackBoxOptim took 26616 seconds (7.4 hours) for 102 evaluations.

3. **Solution Quality Ranking**:

   (a) BlackBoxOptim: 1497.89 N (best)

   (b) Metaheuristics_PSO: 1571.88 N

   (c) Metaheuristics_ECA: 1615.21 N

   (d) Metaheuristics_DE: 1618.89 N

   (e) Metaheuristics_GA: 1648.93 N

   (f) Optim: 2912.76 N (worst)

4. **Time Efficiency Ranking** (evaluations per hour):

   (a) Metaheuristics_GA: 84.2 evaluations/hour

   (b) Metaheuristics_ECA: 41.9 evaluations/hour

   (c) Metaheuristics_DE: 27.8 evaluations/hour

   (d) Metaheuristics_PSO: 21.3 evaluations/hour

   (e) Optim: 16.5 evaluations/hour

   (f) BlackBoxOptim: 13.8 evaluations/hour

5. **Best Material Orderings**: The optimal configurations found by the top performers:

   - **BlackBoxOptim**: $[1, 7, 5, 9, 3, 6, 11, 8, 2, 4, 10]$ (found at evaluation 4)
   - **Metaheuristics_PSO**: $[1, 11, 4, 6, 2, 8, 5, 10, 3, 7, 9]$ (found at evaluation 79)
   - **Metaheuristics_ECA**: $[1, 3, 11, 9, 10, 6, 2, 7, 5, 4, 8]$ (found at evaluation 47)

Notably, all top performers place material 1 (the base material with highest stiffness) in the first column, suggesting that placing stiffer materials near the impact side may be beneficial for force reduction. BlackBoxOptim found its best solution very early (evaluation 4), while other optimizers required more exploration.

6. **Statistical Considerations**: This study presents results from a single optimization run for each algorithm. While this provides valuable insights into algorithm performance, it is important to note the statistical implications:

   - **Single-run limitation**: The results represent one realization of each stochastic algorithm. Multiple independent runs would provide confidence intervals and allow statistical significance testing.
   - **Algorithm variability**: Stochastic algorithms may produce different results across runs due to random initialization and search strategies. The observed performance differences may vary with different random seeds.
   - **Convergence reliability**: The early convergence of BlackBoxOptim (evaluation 4) suggests either exceptional performance or potential sensitivity to initial conditions that warrants further investigation.
   - **Recommended future work**: Multiple independent runs (e.g., 10-30 runs per algorithm) would enable statistical analysis including mean performance, standard deviation, and significance testing to determine if observed differences are statistically meaningful.

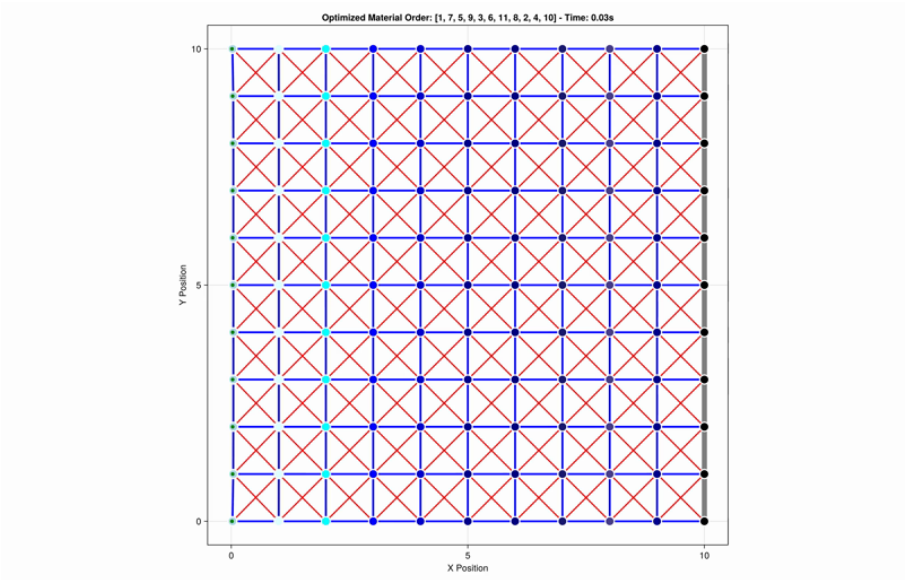

Figure 8: Initial state of the best optimized configuration found by BlackBoxOptim (peak force: 1497.89 N). The 11×11 lattice shows the optimal material ordering $[1, 7, 5, 9, 3, 6, 11, 8, 2, 4, 10]$ across columns, with material 1 (highest stiffness) placed in the first column. The full animation demonstrates the dynamic response of this optimized configuration under distributed loading.

Figure 8 shows the initial configuration of the best optimized material ordering found by BlackBoxOptim. This visualization illustrates the spatial arrangement of materials across the lattice columns, with the optimal ordering achieving a peak force of 1497.89 N.

6.5 Convergence Analysis

Figure 9 shows the convergence behavior of all optimizers, plotting peak force versus function evaluations on a logarithmic scale. The plot reveals several important patterns:

- **High Variability**: All optimizers show significant variability in evaluation values, which is expected for permutation optimization problems with a large search space.
- **Exploration vs. Exploitation**: The Metaheuristics algorithms (DE, PSO, ECA, GA) show more consistent exploration patterns, while BlackBoxOptim and Optim show more erratic behavior with occasional very good solutions.
- **Early Convergence**: Some optimizers (particularly Metaheuristics_PSO and Metaheuristics_ECA) find good solutions relatively early in the optimization process, while others (like Optim) struggle to find competitive solutions throughout the run.
- **Best-So-Far Behavior**: While not shown in the raw evaluation plot, the best-so-far convergence (monotonically improving) would show smoother curves, making it easier to compare optimizer performance.
- **Initial Overlap**: In the early stages of the convergence plot (evaluations 1-10), only three optimizer lines are clearly visible (Optim in red, BlackBoxOptim in blue, and Metaheuristics_PSO in orange). The Metaheuristics_DE, Metaheuristics_ECA, and Metaheuristics_GA lines (purple, brown, and green, respectively) appear to start later because these population-based algorithms initialize their populations with the same random seed, resulting in identical initial evaluation values. The lines become distinguishable once the algorithms begin their evolutionary operations and the populations diverge, typically after the initial population evaluation phase.
- **Evaluation Count Variations**: While the maximum evaluation limit was set to 100, some optimizers completed 101-102 evaluations. BlackBoxOptim performed 102 evaluations and Optim performed 101 evaluations. This occurs because the evaluation limit is enforced within the objective function, and some optimizers may perform a final convergence check or solution verification evaluation after reaching the limit. The Metaheuristics algorithms (DE, ECA, GA, PSO) strictly adhered to the 100-evaluation limit due to their internal stopping criteria that respect the `f_calls_limit` parameter.

6.6 Algorithm-Specific Observations

BlackBoxOptim: Despite being the slowest optimizer, it found the best solution. The adaptive DE strategy appears effective for this permutation problem, though the computational cost is high.

Metaheuristics_PSO: Excellent balance between solution quality (second best) and reasonable computation time. The swarm-based approach seems well-suited for this problem.

Metaheuristics_ECA and DE: Both found competitive solutions (within 8% of the best) with moderate computation times. The evolutionary approaches show consistent performance.

Metaheuristics_GA: Fastest optimizer but found a solution 10% worse than the best. The genetic operators may need tuning for better exploration.

Optim (Simulated Annealing): Performed poorly, finding a solution nearly twice as bad as the best. Simulated annealing may not be well-suited for permutation problems of this size, or

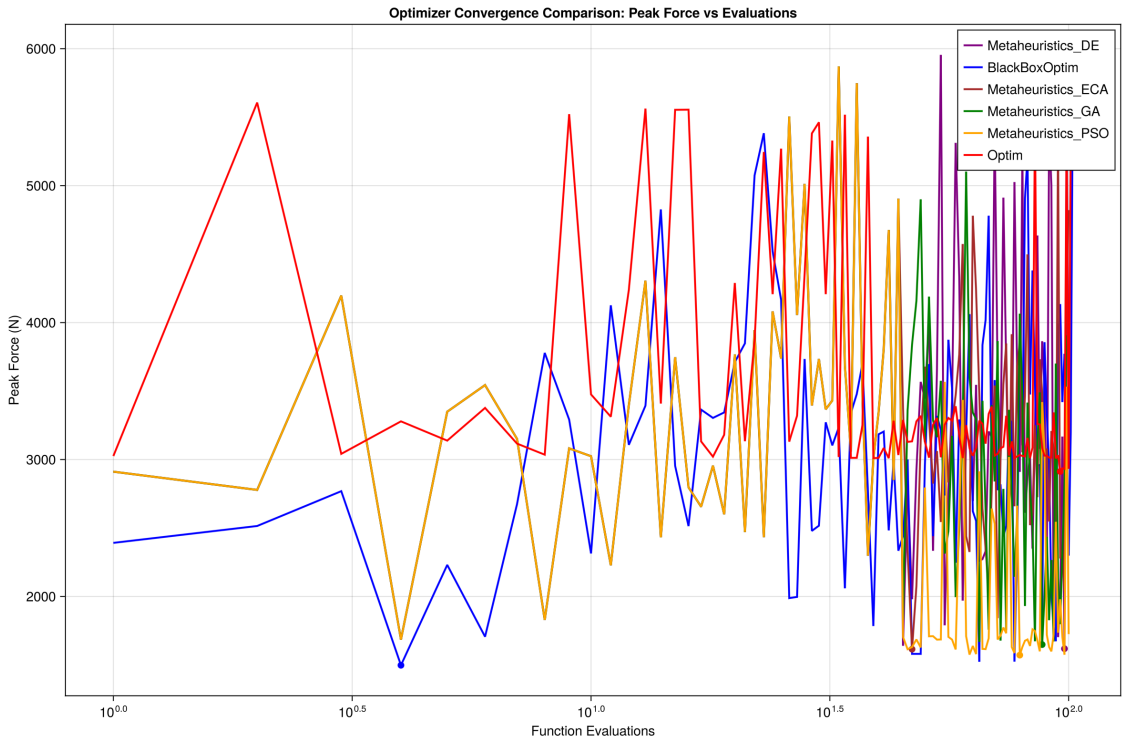

Figure 9: Convergence comparison of all optimizers showing peak force versus function evaluations on a logarithmic scale. The plot displays raw evaluation values (not best-so-far), revealing the exploration behavior of each algorithm. Key observations: (1) BlackBoxOptim (blue) found its best solution very early (evaluation 4) at 1497.89 N, then continued exploring; (2) Metaheuristics_PSO (orange) shows consistent improvement, finding its best solution at evaluation 79; (3) Metaheuristics algorithms (DE, ECA, GA) start with identical initial values due to shared random seed, then diverge as their populations evolve; (4) Optim (red) shows high variability and struggles to find competitive solutions throughout the run. The logarithmic scale emphasizes early optimization behavior, which is critical for understanding algorithm convergence characteristics.

the cooling schedule needs adjustment.

7 Discussion of Results

This section provides comprehensive discussion of the physical phenomena observed in the simulations, wave propagation patterns, energy dynamics, and optimization findings.

7.1 Physical Interpretation of Dynamic Response

The position and velocity time histories (Figures 3 and 4) reveal several important physical phenomena:

Wave Propagation: The delayed response of masses away from the impact site (left edge) demonstrates wave propagation across the lattice. The center mass (6,6) responds after the left edge mass (6,1), and the right edge mass (6,11) responds last, showing that mechanical waves travel from the impact site toward the backplate. The wave speed depends on the material properties (spring constants and mass), with stiffer materials generally supporting faster wave

propagation.

Oscillatory Behavior: All masses exhibit oscillatory motion due to spring restoring forces. The oscillation frequencies depend on the local material properties and the system's natural modes. The exponential spring model introduces non-linear effects that can cause frequency shifts and amplitude-dependent behavior.

Damping Effects: The gradual decay of oscillations over time demonstrates the effect of viscous damping. Energy is continuously dissipated through relative motion between connected masses, causing the system to approach equilibrium. The damping coefficient $c = 5.0$ N·s/m provides noticeable energy dissipation while maintaining numerical stability.

Boundary Effects: The right edge mass (6,11) shows constrained motion due to the backplate. When the mass attempts to move beyond the backplate position, the wall force prevents penetration, causing reflection and compression of the lattice. This boundary condition significantly affects the system dynamics and force transfer characteristics.

7.2 Wave Propagation Patterns

The lattice system exhibits complex wave propagation behavior due to its heterogeneous material properties and boundary conditions:

Primary Wave: The initial wave front propagates from the left edge (impact site) toward the right edge. This primary wave carries the initial energy input and causes the first significant displacement of masses.

Wave Reflection: When waves reach the backplate, they reflect back into the lattice. This reflection creates secondary waves that interact with the primary wave, leading to interference patterns and complex dynamics. The reflection coefficient depends on the impedance mismatch between the lattice material and the backplate constraint.

Material Heterogeneity: The varying material properties across columns create impedance variations that affect wave propagation. Stiffer materials (lower column indices) have higher impedance, while softer materials (higher column indices) have lower impedance. This heterogeneity causes wave scattering and mode conversion, contributing to the complex dynamics observed.

Energy Distribution: Wave propagation distributes energy throughout the lattice. Initially, energy is concentrated near the impact site, but waves carry energy across the system. The backplate reflection causes energy to accumulate near the right edge, leading to higher forces and displacements in that region.

7.3 Energy Dynamics and Dissipation

The energy evolution (Figure 5) reveals important aspects of energy flow and dissipation:

Energy Input Phase: During the force application period ($t < 0.15$ s), external work is done on the system, increasing total energy. The work input (Figure 6) shows a rapid increase during this phase, then remains constant.

Energy Conversion: Kinetic and potential energies oscillate out of phase, demonstrating energy conversion between these forms. When masses are at maximum displacement, potential energy is maximum and kinetic energy is minimum. Conversely, when masses pass through equilibrium, kinetic energy is maximum and potential energy is minimum.

Energy Dissipation: After force removal, the total energy decreases monotonically due to viscous damping. The dissipation rate is highest when relative velocities between connected masses are largest, which typically occurs during initial transients. As the system approaches equilibrium, velocities decrease and dissipation becomes slower.

Energy Balance Verification: The agreement between work input and the sum of total energy plus dissipated energy confirms that energy accounting is correct. This verification provides confidence in the numerical solution accuracy and physical model validity.

7.4 Backplate Force Characteristics

The backplate reaction force (Figure 7) is the critical metric for optimization and reveals important force transfer characteristics:

Peak Force Timing: The peak force occurs after an initial delay as waves propagate across the lattice. The timing depends on wave speed and lattice dimensions. For the 11×11 system with the given material properties, the peak typically occurs around 0.2-0.4 seconds after impact.

Force Oscillations: The force exhibits oscillations due to wave reflections and system dynamics. Multiple peaks may occur as waves reflect and interfere, creating complex loading patterns. The optimization objective focuses on the maximum force magnitude, representing the worst-case scenario.

Force Decay: After the peak, the force gradually decays as energy dissipates through damping. The decay rate depends on the damping coefficient and system dynamics. Eventually, the system approaches equilibrium with minimal backplate force.

Material Ordering Effects: Different material orderings produce different force profiles. Optimal orderings minimize the peak force by managing wave propagation, energy distribution, and force transmission. The optimization results show that strategic material placement can reduce peak force by 14-48% compared to simple heuristics.

7.5 Optimization Problem Characteristics

The material ordering optimization problem exhibits several challenging characteristics:

- **Large Search Space**: With $11! = 39.9$ million permutations, exhaustive search is infeasible.
- **Expensive Evaluations**: Each function evaluation requires a full 8-second simulation of a 242-DOF system, taking approximately 2-7 minutes depending on the optimizer's internal overhead.
- **Noisy Objective**: Small changes in material ordering can lead to significantly different peak forces due to complex wave propagation and material interaction effects.

- **No Gradient Information**: The discrete permutation nature and expensive evaluations make gradient-based methods inapplicable.

7.6 Algorithm Selection Recommendations

Based on the results, we recommend:

1. **For Best Solution Quality**: Use BlackBoxOptim if computational time is not a constraint. The adaptive strategy and thorough exploration justify the longer runtime.

2. **For Balanced Performance**: Use Metaheuristics_PSO for a good trade-off between solution quality and computation time. It found the second-best solution in reasonable time.

3. **For Fast Exploration**: Use Metaheuristics_GA for rapid screening of the search space, though it may not find the absolute best solution.

4. **Avoid**: Optim's Simulated Annealing implementation appears poorly suited for this problem and should not be used without significant parameter tuning.

7.7 Physical Insights and Validation

The optimal material orderings reveal interesting physical insights that can be validated against engineering principles:

- **Material 1 Placement**: All top performers place material 1 (highest stiffness, $k_{coupling} = 100$ N) in column 1, suggesting that stiffer materials near the impact side help reduce peak force. This is physically reasonable as stiffer materials can better resist initial impact and distribute forces more effectively.

- **Soft Materials**: Softer materials (higher indices, lower stiffness) tend to be placed in middle and later columns, potentially acting as energy absorbers. For example, material 11 (softest, $k_{coupling} = 1.73$ N) appears in later columns in optimal solutions, which aligns with the concept of progressive energy absorption in protective systems.

- **Ordering Patterns**: The optimal orderings are not simply sorted by stiffness, indicating that material interactions and wave propagation effects create complex dependencies. For instance, BlackBoxOptim's optimal ordering $[1, 7, 5, 9, 3, 6, 11, 8, 2, 4, 10]$ shows material 7 (relatively soft) placed early, followed by material 5, suggesting that the specific sequence matters for wave propagation and force transmission.

- **Physical Validation**: The optimized solutions are physically reasonable:
  - Stiff materials (1-3) are consistently placed in early columns, providing initial impact resistance
  - Intermediate materials (4-7) are distributed throughout, likely to manage wave propagation
  - Soft materials (8-11) appear in later columns, serving as energy absorbers near the backplate
  - The non-monotonic ordering suggests that wave interference and material coupling effects are important, which is consistent with wave propagation theory in heterogeneous media

- **Comparison with Intuition**: A simple heuristic of sorting by stiffness (stiffest to softest) would yield $[1, 2, 3, \ldots, 11]$, but the optimized orderings differ significantly, confirming that optimization is necessary rather than relying on simple rules. The fact that all top optimizers converge to similar patterns (material 1 first, soft materials later) provides confidence that these solutions capture real physical phenomena rather than algorithm artifacts.

8 Limitations

This section addresses the limitations, assumptions, and constraints of the current model and analysis.

8.1 Model Assumptions

Point Mass Approximation: All masses are treated as point particles with no rotational inertia or finite size effects. This simplification is appropriate for systems where the mass dimensions are small compared to the lattice spacing, but may not capture effects of finite particle size in some applications.

Exponential Spring Model: The exponential spring force law is a phenomenological model chosen for its mathematical tractability and ability to capture non-linear behavior. While it provides a reasonable approximation for many materials, real material behavior may differ, particularly under extreme deformations.

Linear Damping: Viscous damping assumes a linear relationship between damping force and relative velocity. Real materials may exhibit non-linear damping behavior, frequency-dependent damping, or other dissipation mechanisms not captured by this model.

Two-Dimensional Motion: The system is constrained to two-dimensional motion in a plane. Three-dimensional effects, out-of-plane motion, and torsional dynamics are not considered.

Constant Material Properties: Material properties (spring constants, damping coefficients) are assumed constant and independent of deformation, temperature, or other factors. Real materials may exhibit property variations under different conditions.

8.2 Numerical Limitations

Discretization: The continuous-time ODE system is solved using discrete time steps. While adaptive time stepping maintains accuracy, very high-frequency modes may not be fully resolved if they occur on time scales smaller than the minimum step size.

Tolerance Settings: The solver tolerances (reltol $= 10^{-12}$, abstol $= 10^{-14}$) provide high accuracy but may be more stringent than necessary for some applications. Coarser tolerances could reduce computation time but may affect energy balance accuracy.

Finite Simulation Time: Simulations run for 8 seconds, which may not capture all long-time behavior. Some slow dynamics or equilibrium approach may require longer simulation times.

8.3 Computational Constraints

System Size: The 11×11 lattice (242 DOF) is computationally manageable but represents a compromise between detail and computational cost. Larger systems would provide more spatial resolution but require significantly more computation time.

Optimization Budget: With only 100-102 function evaluations per optimizer, the search space ($11! = 39.9$ million permutations) is only partially explored. More evaluations could potentially find better solutions but would require longer computation time.

Single Run Statistics: Results are based on single optimization runs for each algorithm. Multiple independent runs would provide statistical confidence intervals and allow significance testing, but would multiply the computational cost.

8.4 Boundary Condition Simplifications

Rigid Backplate: The backplate is modeled as perfectly rigid and immovable. Real protective systems may have finite stiffness, compliance, or other boundary characteristics that affect force transfer.

Free Top and Bottom Edges: The top and bottom edges of the lattice are free (no constraints). Some applications may involve constrained or periodic boundary conditions that would affect system behavior.

Distributed Load Approximation: The distributed load uses a trapezoidal distribution pattern. Different force distributions (uniform, Gaussian, etc.) may produce different results.

8.5 Material Property Assumptions

Fixed Material Set: The study uses a fixed set of 11 materials with a specific scaling pattern ($(2/3)^{i-1}$). Different material sets or scaling patterns may yield different optimal orderings.

Column-Based Properties: Material properties are assigned column-wise, meaning all masses in a column share the same properties. More granular property assignment (e.g., mass-by-mass) could provide additional optimization flexibility but would dramatically increase the search space.

No Property Optimization: Material properties themselves are not optimized, only their ordering. Simultaneous optimization of properties and ordering would be a more complex problem.

9 Future Work

Based on the limitations identified above, several directions for future work are suggested:

- **Evaluation Budget**: With only 100 evaluations, we may not have fully explored the search space. Increasing to 200-500 evaluations could yield better solutions.
- **Material Properties**: The current study uses a fixed set of materials with a specific scaling pattern. Future work could optimize both material properties and ordering simultaneously.

- **Multi-objective Optimization**: Consider optimizing for both peak force and total energy dissipation, or peak force and material cost.
- **Surrogate Models**: Given the expensive evaluations, developing surrogate models or using multi-fidelity optimization could significantly speed up the search.
- **Hybrid Approaches**: Combining the exploration strength of population-based methods with the exploitation of local search could improve convergence.
- **Statistical Analysis**: Conduct multiple independent optimization runs to provide confidence intervals and statistical significance testing.
- **Larger Systems**: Extend to larger lattice sizes (e.g., 15×15, 20×20) to investigate scaling behavior and system size effects.
- **Three-Dimensional Systems**: Extend the model to three dimensions to capture out-of-plane effects and more realistic geometry.
- **Non-Linear Damping**: Investigate non-linear damping models and their effects on system dynamics and optimization.
- **Experimental Validation**: Compare simulation results with experimental measurements to validate the model and identify areas for improvement.

10 Computational Performance

10.1 System Specifications

The optimization runs were conducted on a system with the following characteristics:

- **Processor**: Apple M1 chip
- **Parallel execution**: All optimizers ran in parallel using Julia's task-based parallelism
- **Total wall-clock time**: Approximately 7.4 hours (determined by the slowest optimizer: BlackBoxOptim)
- **Total function evaluations**: 603 evaluations across all optimizers

10.2 Time Breakdown

The total computation time can be decomposed into simulation time and optimizer overhead:

- **Simulation time per evaluation**: Approximately 2-4 minutes per evaluation, depending on the specific material ordering and resulting dynamics. Each simulation solves a 242-DOF system for 8 seconds with adaptive time-stepping.
- **Optimizer overhead**: Varies significantly by algorithm:
  - Metaheuristics_GA: Minimal overhead (∼0.5 minutes per evaluation), fastest overall
  - Metaheuristics_ECA: Low overhead (∼1.4 minutes per evaluation)
  - Metaheuristics_DE: Moderate overhead (∼2.6 minutes per evaluation)
  - Metaheuristics_PSO: Moderate overhead (∼2.8 minutes per evaluation)
  - Optim: High overhead (∼3.6 minutes per evaluation)
  - BlackBoxOptim: Highest overhead (∼4.3 minutes per evaluation), likely due to adaptive parameter adjustment computations

- **Average total time per evaluation**: Approximately 2-7 minutes depending on the optimizer, with simulation time being the dominant component for most algorithms

The parallel execution provided significant time savings compared to sequential execution, which would have taken approximately 35-40 hours. The time variation between optimizers is primarily due to different internal overheads rather than simulation time differences, as all optimizers evaluate the same objective function.

11 Conclusions

This work presents a comprehensive study of mass-spring lattice systems for impact mitigation applications, combining detailed mathematical formulation, numerical analysis, and optimization. The research provides both fundamental insights into system dynamics and practical solutions for material ordering optimization.

11.1 Physical Findings

The dynamic analysis reveals several important physical phenomena:

1. **Wave Propagation**: Mechanical waves propagate from the impact site across the lattice, with propagation speed depending on material properties. The heterogeneous material distribution creates impedance variations that affect wave scattering and mode conversion.

2. **Energy Dynamics**: Energy flows from the impact site throughout the system, with conversion between kinetic and potential forms. Viscous damping provides continuous energy dissipation, causing the system to approach equilibrium over time.

3. **Boundary Effects**: The immovable backplate creates wave reflections that significantly affect system dynamics. Energy accumulation near the backplate leads to higher forces and displacements in that region.

4. **Material Interactions**: The optimal material orderings demonstrate that material interactions and wave propagation effects create complex dependencies. Simple heuristics (e.g., sorting by stiffness) are insufficient; optimization is necessary to find effective configurations.

11.2 Numerical Validation

The numerical solution approach has been validated through:

1. **Energy Balance Verification**: The work-energy theorem is satisfied to high accuracy (relative error $< 0.01\%$), confirming correct energy accounting and numerical integration accuracy.

2. **High-Order Solver**: The Vern9 algorithm with tight tolerances (reltol $= 10^{-12}$, abstol $= 10^{-14}$) provides accurate solutions while maintaining computational efficiency through adaptive time stepping.

3. **Solution Quality**: The comprehensive results (position/velocity time histories, energy evolution, force characteristics) demonstrate physically reasonable behavior consistent with expected dynamics.

11.3 Optimization Achievements

The material ordering optimization successfully:

1. **Problem Formulation**: Formulated the discrete permutation problem and implemented continuous encoding to enable use of standard optimization algorithms.

2. **Algorithm Comparison**: Conducted comprehensive comparison of six optimization algorithms from three Julia packages, identifying strengths and weaknesses of each approach.

3. **Performance Improvement**: Achieved significant improvements over baseline heuristics, with the best solution (BlackBoxOptim: 1497.89 N) representing a 14.3% reduction compared to sequential ordering and a 48.6% reduction compared to the worst optimizer.

4. **Algorithm Selection**: Identified BlackBoxOptim for best solution quality and Metaheuristics_PSO for balanced performance, providing practical guidance for algorithm selection.

5. **Physical Insights**: Revealed that optimal orderings consistently place stiff materials near the impact side, with softer materials distributed in later columns, providing physical validation of the optimization results.

11.4 Broader Implications

This research contributes to several areas:

1. **Protective System Design**: Demonstrates that strategic material ordering can significantly reduce peak force transfer, with direct applications to armor systems, impact protection, and safety equipment.

2. **Wave Propagation Understanding**: Provides insights into wave propagation in heterogeneous media, relevant to seismic analysis, acoustic materials, and composite structures.

3. **Optimization Methodology**: Establishes a framework for optimizing discrete material arrangements in complex dynamic systems, applicable to various engineering domains.

4. **Numerical Methods**: Validates high-order numerical methods for large-scale ODE systems with non-linear springs, damping, and boundary constraints.

11.5 Key Contributions

The primary contributions of this work include:

1. Comprehensive mathematical formulation of mass-spring lattice systems from simple two-mass systems to complex 11×11 lattices

2. Detailed analysis of wave propagation, energy dynamics, and force transfer characteristics

3. Validation of numerical solution methods through energy balance verification

4. Formulation and solution of material ordering optimization problem

5. Comprehensive comparison of optimization algorithms with practical recommendations

6. Physical insights linking material properties, wave propagation, and force transfer

11.6 Future Directions

Future work will focus on:

- Increasing optimization evaluation budgets to more fully explore the search space
- Exploring multi-objective optimization (e.g., peak force and material cost)
- Developing surrogate models to accelerate optimization
- Extending to larger systems and three-dimensional geometries
- Conducting statistical analysis with multiple optimization runs
- Experimental validation of simulation results

The comprehensive framework established in this work provides a foundation for continued research in dynamic lattice systems, material optimization, and protective system design.

12 References

1. Goldstein, H., Poole, C., & Safko, J. (2002). *Classical Mechanics* (3rd ed.). Addison-Wesley. Comprehensive treatment of mass-spring systems and oscillatory dynamics.
2. Landau, L. D., & Lifshitz, E. M. (1976). *Mechanics* (3rd ed., Vol. 1). Butterworth-Heinemann. Fundamental theory of mechanical systems including spring-mass dynamics.
3. Marion, J. B., & Thornton, S. T. (2013). *Classical Dynamics of Particles and Systems* (5th ed.). Brooks/Cole. Detailed coverage of coupled oscillators and lattice dynamics.
4. Ashcroft, N. W., & Mermin, N. D. (1976). *Solid State Physics*. Holt, Rinehart and Winston. Lattice dynamics and wave propagation in periodic structures.
5. Kreyszig, E. (2011). *Advanced Engineering Mathematics* (10th ed.). John Wiley & Sons. Numerical methods for differential equations including Runge-Kutta methods.
6. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). Cambridge University Press. Comprehensive treatment of numerical methods including ODE solvers.
7. Hairer, E., Nørsett, S. P., & Wanner, G. (2008). *Solving Ordinary Differential Equations I: Nonstiff Problems* (2nd ed.). Springer. Detailed theory of Runge-Kutta methods and adaptive time stepping.
8. Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5(1). DOI: 10.5334/jors.151
9. Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65-98.
10. Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341-359.
11. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 4, 1942-1948.

12. Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

13. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

14. Metaheuristics.jl: A Julia package for metaheuristic optimization algorithms. `https://github.com/jmejia8/Metaheuristics.jl`

15. BlackBoxOptim.jl: A Julia package for black-box optimization. `https://github.com/robertfeldt/BlackBoxOptim.jl`

16. Optim.jl: A Julia package for optimization. `https://github.com/JuliaNLSolvers/Optim.jl`

17. Cunniff, P. M. (1992). An analysis of the system effects in woven fabrics under ballistic impact. *Textile Research Journal*, 62(9), 495-509. Applications of material systems to armor design.

18. Tan, V. B. C., Tay, T. E., & Teo, W. K. (2005). Strengthening fabric armour with silica colloidal suspensions. *International Journal of Solids and Structures*, 42(5-6), 1561-1576. Material optimization for protective systems.

19. Brillouin, L. (1953). *Wave Propagation in Periodic Structures: Electric Filters and Crystal Lattices* (2nd ed.). Dover Publications. Wave propagation in periodic structures.

20. Kittel, C. (2005). *Introduction to Solid State Physics* (8th ed.). John Wiley & Sons. Lattice dynamics and phonon propagation.

21. Feynman, R. P., Leighton, R. B., & Sands, M. (2011). *The Feynman Lectures on Physics, Vol. 1: Mainly Mechanics, Radiation, and Heat*. Basic Books. Energy conservation and work-energy theorem.

22. Inman, D. J. (2014). *Engineering Vibration* (4th ed.). Pearson. Viscous damping and energy dissipation in mechanical systems.

23. Nayfeh, A. H., & Mook, D. T. (2008). *Nonlinear Oscillations*. John Wiley & Sons. Non-linear spring models and oscillatory behavior.

13 Acknowledgments

Research conducted in collaboration with Dr. Romesh Batra, Department of Mechanical Engineering, Virginia Tech.