TARLETON STATE UNIVERSITY
Member of The Texas A&M University System

N-body Adaptive Optimization of Lattice Towers

J. Domine, H. Grant, W. Young, Dr. R. Vuddandam, Dr. B. Wyatt
Tarleton State University Department of Mathematics
Particle Modeling Group

Abstract

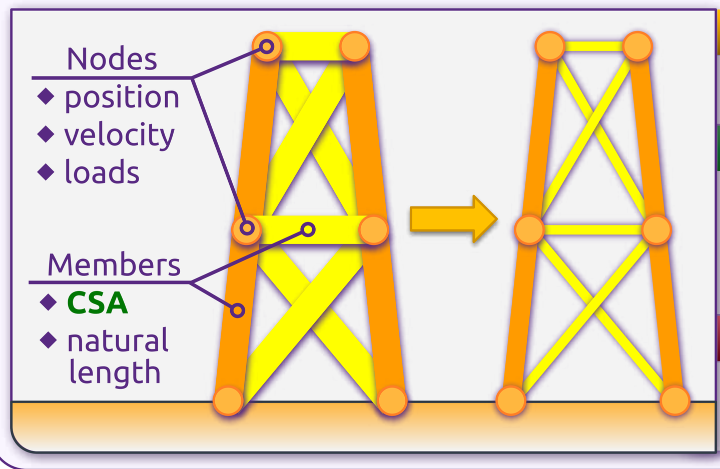
In the US alone, transmission towers number in the tens of millions; most are lattice-based, comprising a framework of beams connected at nodes. At tens- to hundreds of thousands of dollars each, primarily in material costs, they are a sizable expense to taxpayers and transmission companies.

The dilemma central to their optimization is that a beam's cross-sectional area is proportional to both its strength *and* cost, i.e. a thicker beam, while able to support a given load with less strain, will cost more and weigh more heavily on the beams supporting it.

By varying the cross-section of each beam, we want to make towers as light and inexpensive as possible without sacrificing their structural integrity. Known as truss-sizing optimization, this problem is differential in nature and heavily dependent on tower geometry, lending it to a computational approach.

Drawing inspiration from the atrophy and hypertrophy of muscles, we develop and evaluate an optimization algorithm that adaptively resizes beams based on their stress – a process that produces rapid results and allows the application of both static and dynamic loads, setting it apart from popular algorithms in this intensely studied field.

Objective & Constraints

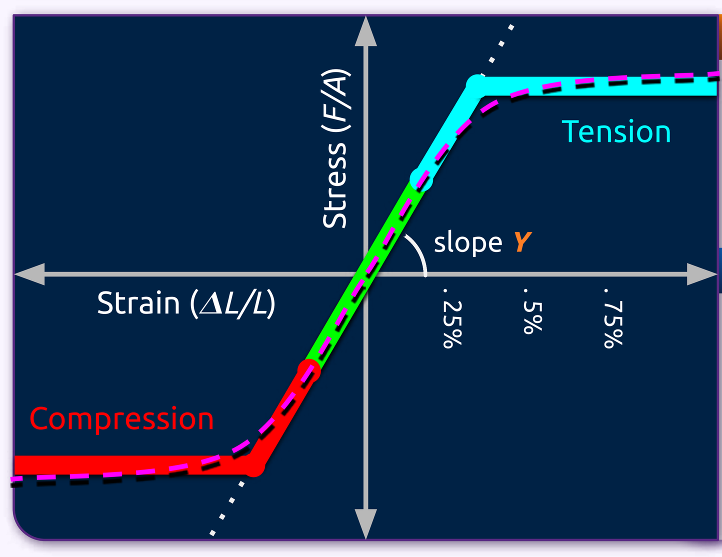


Objective
Make a given lattice tower as light as possible.

Variable
Each member has a cross-sectional area (CSA) that determines its strength and weight. The CSA of each member or member group is varied.

Constraints
- Maximum strain
- Allowable CSA range
- Maximum node displacement

Stress, Strain & Loads



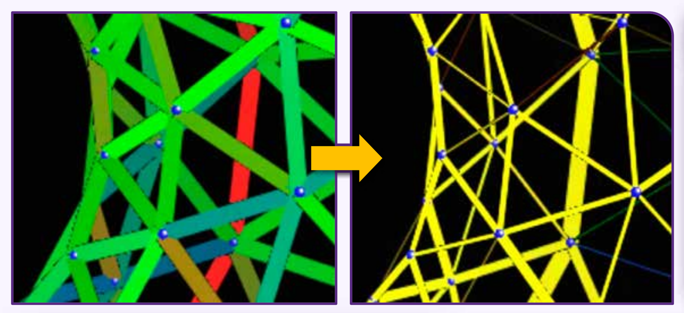
Material stress-strain curve, true (magenta) & simulated (tricolor)

Force Curve
A beam behaves like a spring, having a natural length and a strength dependent on CSA, Young's modulus Y, and **strain** (relative change in length):
$$F = -\text{strain} \cdot Y \cdot \text{CSA}$$

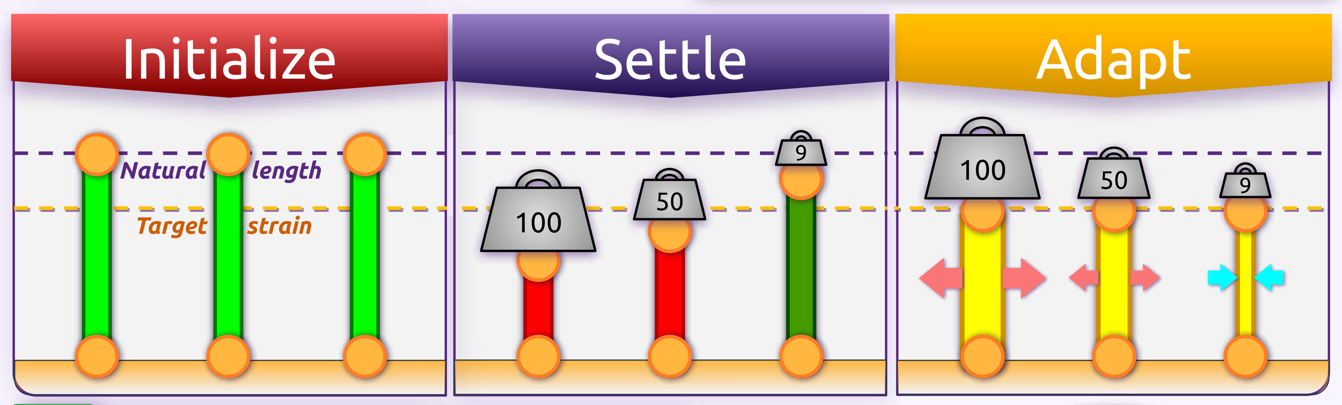
Applied Loads
In our simulation, lattice towers are subjected to various static and dynamic loads:
- Dead load (self-weight, power lines, etc.)
- Live load (occupants, temporary loads)
- Wind, seismic, ice, etc. loads

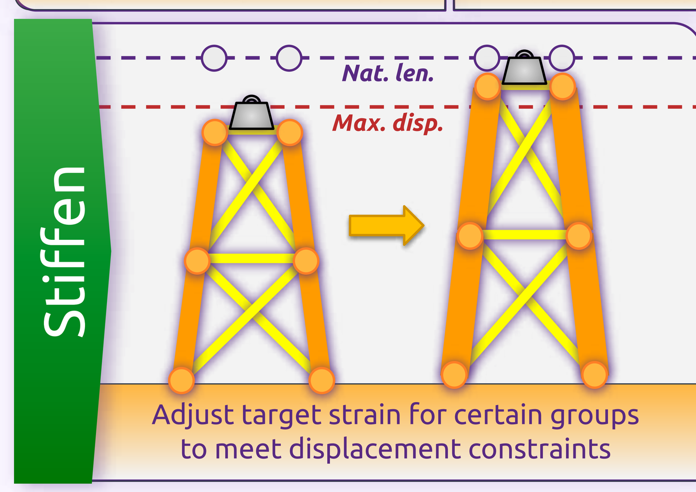
Loads are applied at nodes, e.g. the weight of a member is divided evenly between its nodes.

Adaptive Algorithm

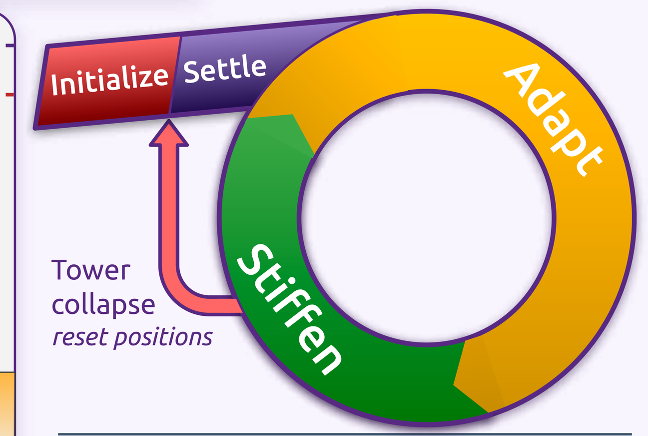


Rules
- Each member has a *target strain σ*
- If strain < σ, member *atrophies*
- If strain > σ, member *grows*



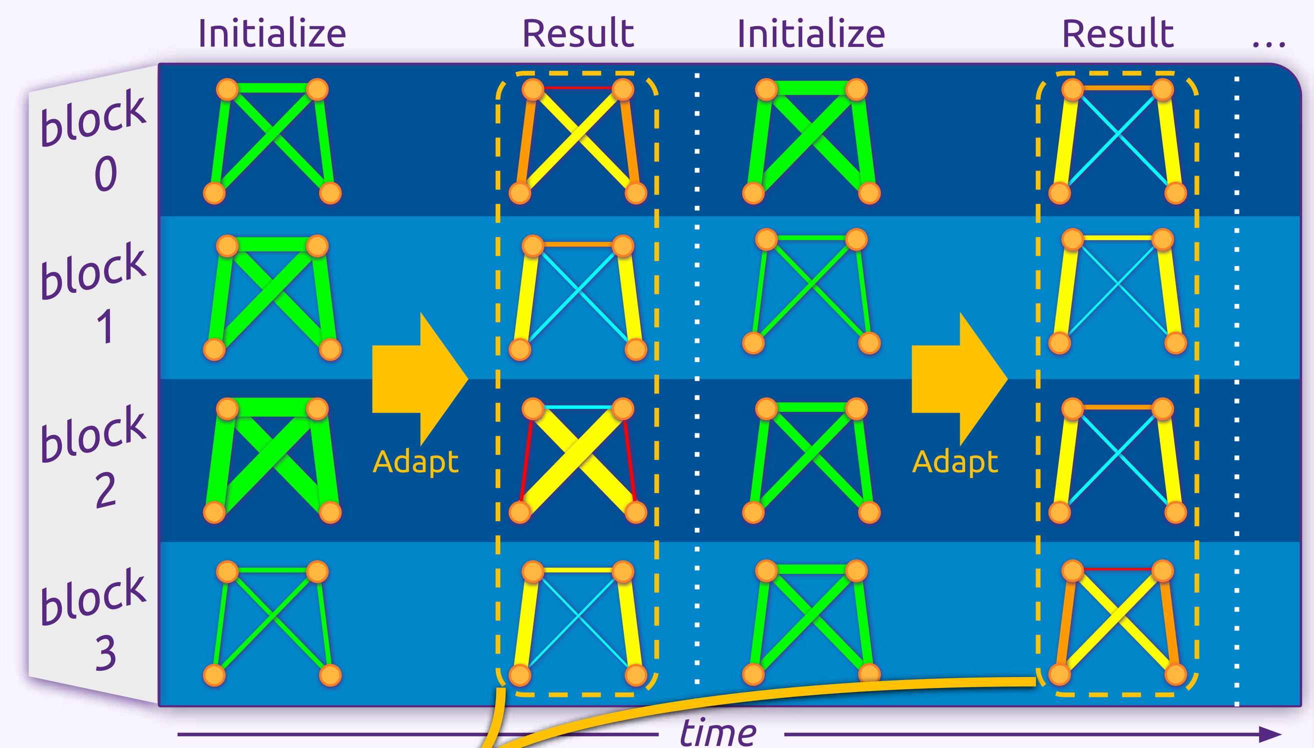


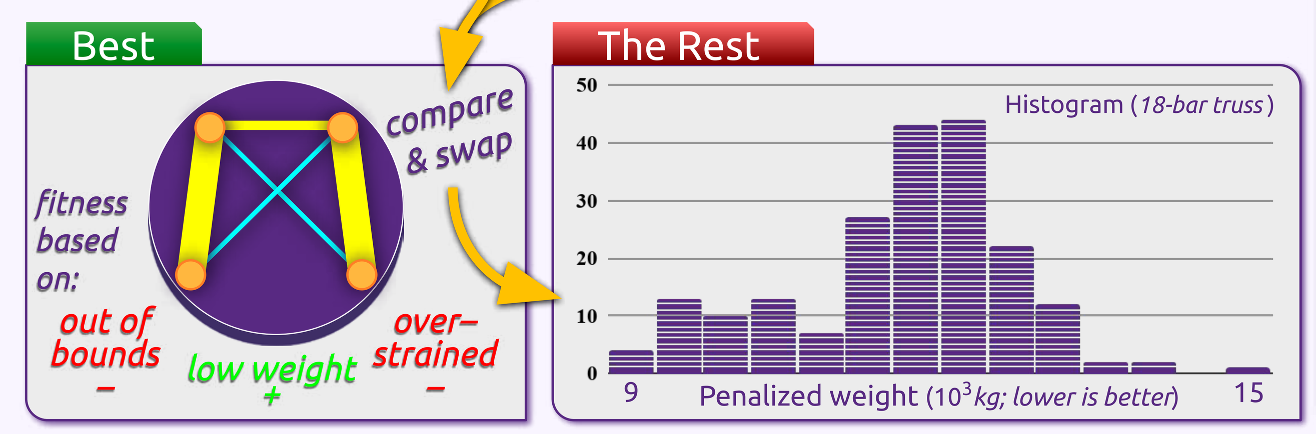
Adjust target strain for certain groups to meet displacement constraints

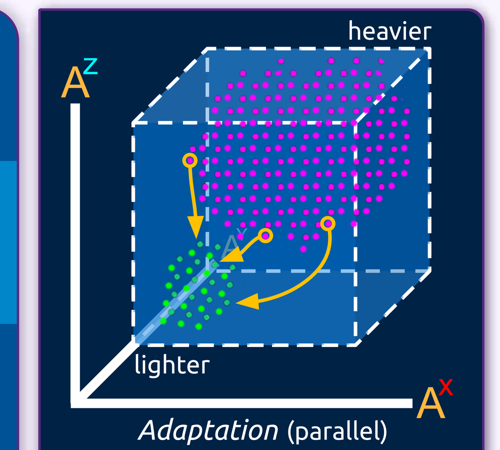


Adaptation cycle

Parallel Approach

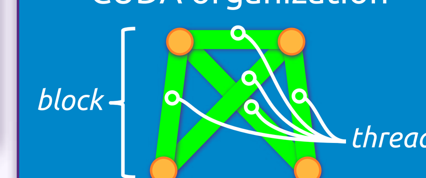






Adaptation (parallel)

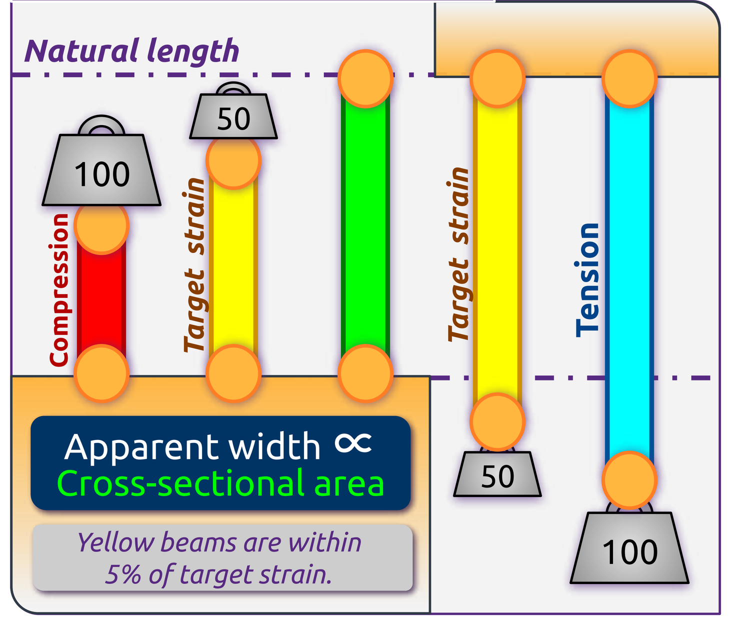
CUDA organization



- Many (1,000,000+) blocks
- Several consec. runs per block
- **M** (members) threads per block
- First **N** (nodes) control nodes
- First **G** (groups) control groups

Parameter variation
- Initial CSA
- CSA distribution (unif/exp/rand)
- Adaptation rate
- Time-variant strain/disp. limits

Strain Legend



Apparent width ∝ Cross-sectional area

Yellow beams are within 5% of target strain.

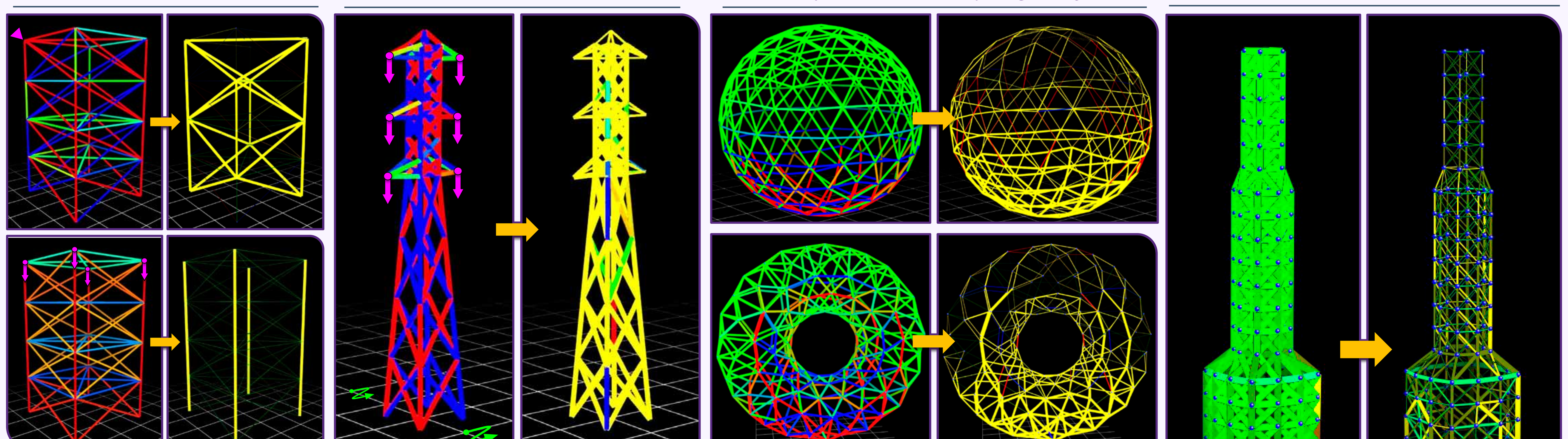
Hardware

Nvidia GeForce GTX Titan
Intel i7 950

GPU vs. CPU
On single-run tests, GPU outperforms CPU by ~20×.
The GPU can run 1,048,576+ adaptations at once.
Speedup over CPU roughly 20 million ×.
(5-10 s/run; simultaneous runs do not affect performance)

Results

72-bar truss, 2 load configurations · *Small tower, cable and seismic loads.* · *Icosphere and torus, self-weight only* · *26-story tower, asymmetric load*

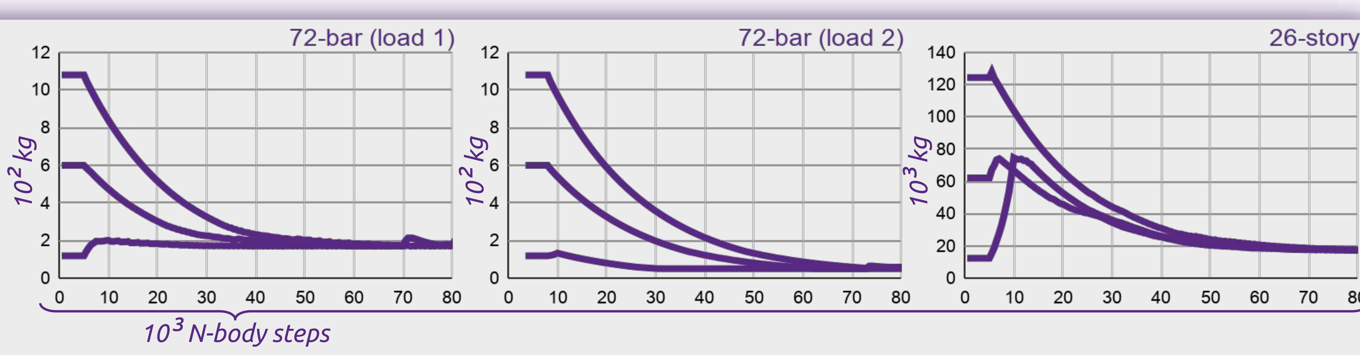


Convergence

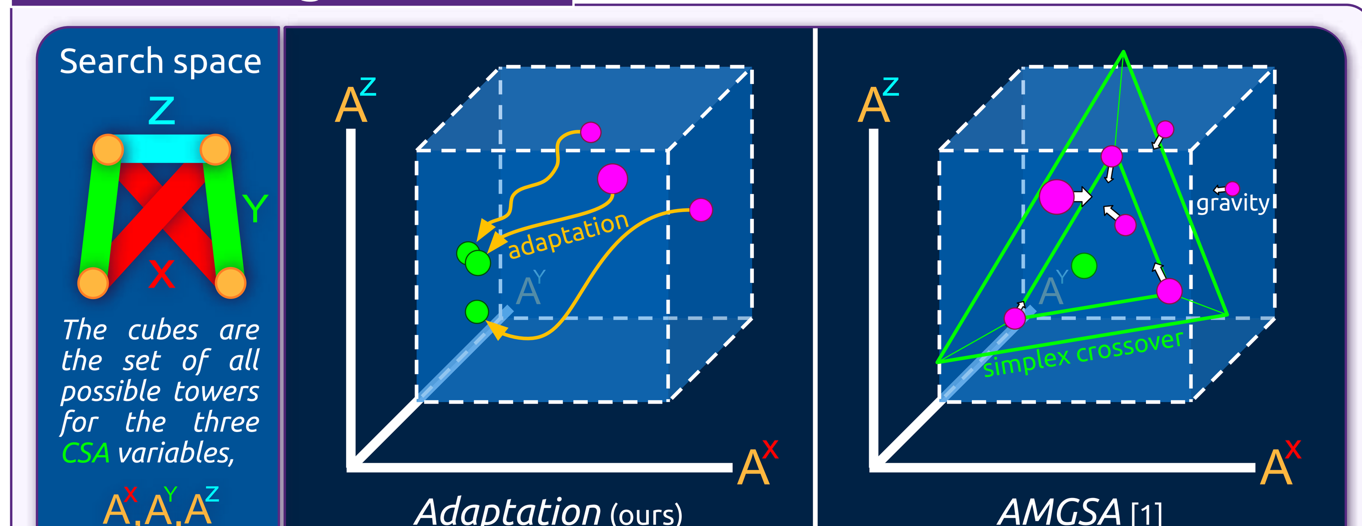
Initial CSA generally has little impact on convergence.

In the last figure, however, the lowest initial CSA caused a collapse, prompting sudden growth.

Simulation videos



Search Algorithms



	Adaptation (ours)	AMGSA [1]
Inspiration	Muscle growth and atrophy	Gravitation, Genetics
Physics model	N-body (numeric integration)	Matrix algebra
Search behavior	Gradual refinement	Attraction, breeding, mutation
Advantage	Dynamic loads, rapid convergence, GPU acceleration	Matrix efficiency, deep mathematical foundation idealistic physics

References:
[1] Khatibinia, Mohsen & Yazdani, Hessam. (2017, July). Accelerated multi-gravitational search algorithm for size optimization of truss structures. In *Swarm and Evolutionary Computation.*

Acknowledgements:
Thank you to the Tarleton Math Department, Dr. Rajesh Vuddandam and Dr. Bryant Wyatt for the ideas and resources provided for this project.

Poster PDF