Designing High Thermal Conductive Materials Using Artificial Evolution

MICHAEL DAVIES,
BASKAR GANAPATHYSUBRAMANIAN,
GANESH BALASUBRAMANIAN
The Problem

- **Graphene** is one of the most thermally conductive materials we know of

- Can we make it better?
Motivation

• More efficient heat sinks allow for
  • Faster processors / computers

• Better heat transfer (heating and cooling systems)
  • Lower cost air conditioning / heating
The Problem

- This is an “optimization problem”

- The only proven way to find the “correct answer” is exhaustive trial testing

- This approach would lead to performing $2^{8160}$ or $2.54 \times 10^{2456}$ trials, which would take $2.6 \times 10^{2453}$ years to complete
Our solution: Artificial Evolution

• Modeling the problem with the concept of evolution gives us a different approach to solving it

• At a high level, we generate a population of trials, and let them slowly evolve over time through the process of natural selection to find a good solution.

• Known as a “Genetic Algorithm”
Step 1: Modeling a genome in code

- In life, our DNA represents our ‘genetic code’
- The first part of developing a genetic algorithm is to design the structure of the ‘DNA’ in software

10110101 01001011 01110110 11001011 01101110 01011100
Step 1: Modeling a genome in code

- Our material trials are a string of 8160 graphene atoms with impurities placed randomly within it.
Step 1: Modeling a genome in code

- We used 1 byte to represent each atom
- With 8160 atoms, that is 8.2kb of data for one chromosome
Step 1: Modeling a genome in code

- A graphical representation
Step 2: Creating a population

- After describing a chromosome in code, a program first generates a random population – generation zero

```
0x01 0x01 0x01 0x01 0x01 0x01 0x02 0x01 0x01 0x02 0x01 0x02 0x01 0x02 0x02
0x02 0x02 0x01 0x01 0x02 0x01 0x01 0x02 0x01 0x02 0x01 0x02 0x02 0x02 0x02
0x01 0x01 0x01 0x01 0x02 0x01 0x01 0x01 0x01 0x01 0x01 0x02 0x01 0x01 0x02
0x02 0x02 0x01 0x02 0x01 0x01 0x02 0x01 0x01 0x02 0x01 0x02 0x01 0x01 0x02
```

...
Step 3: Evaluating Fitness

- LAMMPs (Large Atomic Molecular Massively Parallel Simulator) is a molecular dynamics simulator

- It allows us to give an input configuration of atoms, then simulates it

- We then parse the output to calculate the thermal conductivity
Step 3: Evaluating Fitness

- Once the initial population is created, each chromosome is evaluated
- We used a LAMMPS simulation to evaluate our chromosomes
Step 3: Evaluating Fitness

- These simulations were run in parallel on Stampede – a 10 petaflop supercomputer
Step 3: Evaluating Fitness

- LAMMPS simulations were run in parallel on Stampede using custom software called FTAdagio (Fault – Tolerant ADaptive sparse-GrId allOcator)
Step 3: Evaluating Fitness

- We do this by first finding the slope of the average temperature across the material
Step 3: Evaluating Fitness

- We do this by first finding the slope of the average temperature across the material.
Step 3: Evaluating Fitness

- We do this by first finding the slope of the average temperature across the material
Step 3: Evaluating Fitness

- We do this by first finding the slope of the average temperature across the material

![Diagram showing temperature distribution and calculation of average slope]

**Avg Slope = 0.0447716 K / Bin**
Step 3: Evaluating Fitness

- Then we find the average thermal energy over time

\[
\text{Average} = 4593.6 \text{ ev}
\]
Step 3: Evaluating Fitness

- A score is derived by dividing the **Average Therm / Average Slope**

\[
Fitness = \frac{\text{Avg Therm}}{\text{Avg Slope}} = \frac{4593.6 \text{ ev}}{0.0447728 \text{ K/Bin}} = 102600.8 \text{ ev/(K/Bin)}
\]

- Note – this value is some scalar times the thermal conductivity. We ignore this because a scalar will not affect relative scores.
Step 4: Selection - Roulette Wheel

• After each chromosome is evaluated, parents must be selected to produce offspring for the next generation

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 0x02 0x01 0x02 0x01 0x02 0x01 0x01 0x02 0x01</td>
<td>Fitness = 100</td>
</tr>
<tr>
<td>0x02 0x02 0x01 0x02 0x02 0x01 0x02 0x01 0x01 0x01 0x01 0x01</td>
<td>Fitness = 45</td>
</tr>
<tr>
<td>0x01 0x02 0x02 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01</td>
<td>Fitness = 73</td>
</tr>
<tr>
<td>0x02 0x02 0x01 0x02 0x02 0x01 0x01 0x01 0x01 0x01 0x01 0x01</td>
<td>Fitness = 20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Step 4: Selection - Roulette Wheel

- With roulette wheel selection, each chromosome is given a probability of being selected by their fitness.

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Fitness</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 0x02 0x01 0x01 0x02 0x01 0x01 0x02 0x01</td>
<td>100</td>
<td>.42</td>
</tr>
<tr>
<td>0x02 0x02 0x01 0x02 0x01 0x02 0x01 0x02 0x01</td>
<td>45</td>
<td>.19</td>
</tr>
<tr>
<td>0x01 0x02 0x01 0x01 0x01 0x01 0x01 0x01 0x01</td>
<td>73</td>
<td>.31</td>
</tr>
<tr>
<td>0x02 0x02 0x01 0x02 0x01 0x02 0x01 0x01 0x01</td>
<td>20</td>
<td>.08</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Step 4: Selection - Roulette Wheel

- Parents are then selected at random with these weighted probabilities:

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Fitness</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 0x02 0x01 0x02 0x01 0x01 0x02 0x01</td>
<td>100</td>
<td>.42</td>
</tr>
<tr>
<td>0x02 0x02 0x01 0x02 0x02 0x01 0x02 0x01</td>
<td>45</td>
<td>.19</td>
</tr>
<tr>
<td>0x01 0x02 0x02 0x01 0x01 0x01 0x01 0x01</td>
<td>73</td>
<td>.31</td>
</tr>
<tr>
<td>0x02 0x02 0x01 0x02 0x02 0x01 0x01 0x01</td>
<td>20</td>
<td>.08</td>
</tr>
</tbody>
</table>
Step 5: Crossover

• When two parents are selected, they undergo crossover, where part of each parent's chromosome is given to the offspring.

Parent 1

0x01 0x02 0x01 0x01 0x02 0x01 0x01 0x02

Parent 2

0x01 0x02 0x02 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x02

Offspring

0x01 0x02 0x01 0x01 0x02 0x01 0x01 0x02
Step 6: Mutation

- After the child is created, it goes through mutation, to introduce genetic diversity that could prove useful
Onward

- The selection, crossover, and mutation process is repeated $n$ times to produce the next generation of chromosomes.
- After the next generation is created, it is then evaluated, and the whole process starts over.
Uncertainty is a Feature

• Randomness promotes genetic diversity

• We use SIMD-Oriented Fast Marsenne Twister to generate random numbers

• This generator has a very large period, minimizing the likelihood of repeat information
Why all this work?

• Genetic algorithms give us a different approach to solve an optimization problem – within our lifetime

• They allow for the evolution of an answer through trial and combining good chromosomes together
Results

- Thermal conductivity shown over % doping results in a known trend
Results

• Known trend
Results

- Some points of interest
Results

• What we are looking at is the distribution at a specific % dope level
Distribution of scores

Data Histogram
Next steps

• Run the genetic algorithm with a constrained % doping

• Perform in depth data analysis of results
Questions?