Machine Learning to Optimize Parameters

Blair Urish
Dan Wagner
18 October 2017

Background Research

- Many fields require materials with properties that conflict
  - Low density + high strength
- Difficult/Impossible to achieve with traditional materials
- Hybrid materials
  - Organic and inorganic compounds
- Molecular dynamics simulations

(source: Narayanan et. al.)

Prior Solutions

- Reactive force field (ReaxFF)
  - Provides accurate descriptions of chemical states/qualities
  - Required for heterosystems like oxides
  - Highly complex with numerous parameters
  - Computationally expensive (doesn’t scale)

(Source: Narayanan et. al.)

Overall Problem

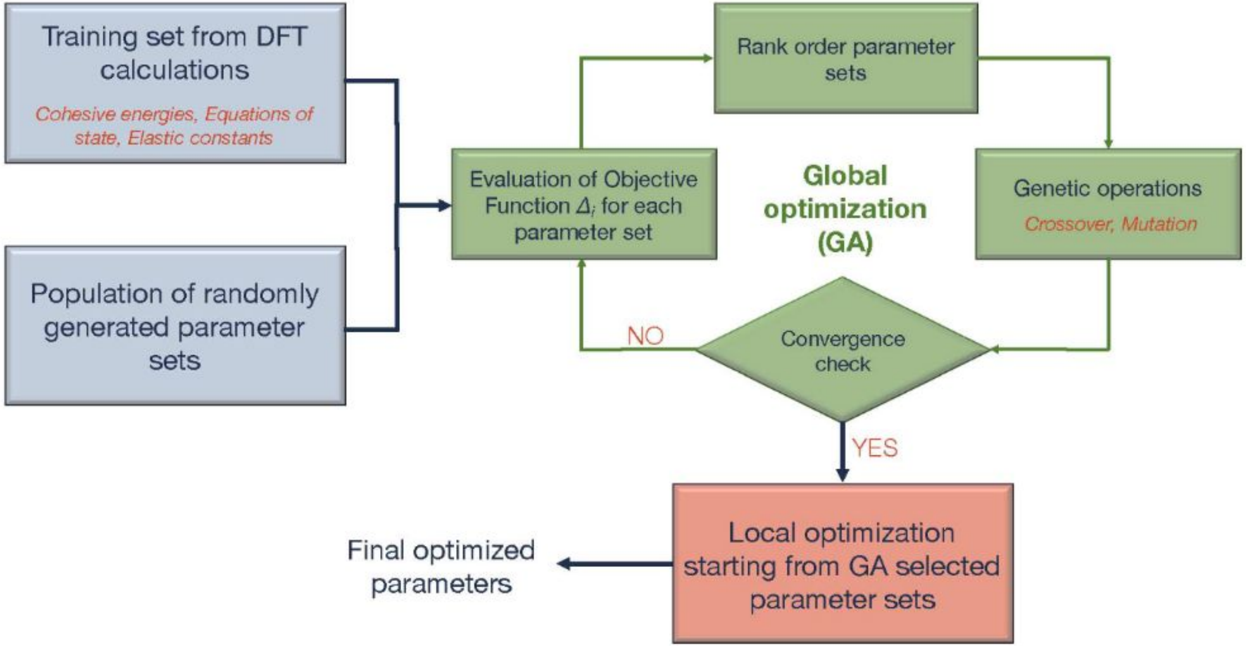
- Machine learning to optimize parameters in genetic algorithms
- Parallelizing the overall process to speed it up
- Interfacing with LAMMPS for output fitting

Genetic Algorithms

- Stochastic search algorithms based on evolution
  - Mutation and crossover of traits
  - Each population evaluated for best solutions
- Consist of generation sets that undergo changes
  - Bad traits are eliminated
  - Traits mutate or transmit to other generations
  - Probability higher for crossover than mutation

(Source: Cantú-Paz et. al.)

Our Approach



Training set from DFT calculations

Cohesive energies, Equations of state, Elastic constants

Population of randomly generated parameter sets

Evaluation of Objective Function Δ_i for each parameter set

Rank order parameter sets

Genetic operations

Crossover, Mutation

Global optimization (GA)

Convergence check

NO

YES

Local optimization starting from GA selected parameter sets

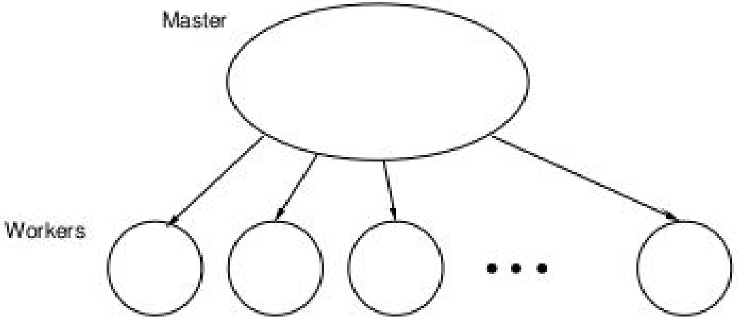
Final optimized parameters

LAMMPS

- Molecular dynamics simulator
- Works on a single processor or multiple in parallel
- Will use to compute objective function values
  - Describes attributes of an individual
- We have LAMMPS code, but it will need to be integrated with rest of program

Our Approach

- Use global single-population master-slave algorithm
    - Master performs algorithm operations
    - Distribute individuals for evaluation across slaves
    - OpenMPI will be used for distribution



(Source: Cantú-Paz et. al.)

Our Approach

- Local minimization via the simplex method
  - Top 20 individuals per generation analyzed
- Tune a parameter that improves the solution
- Procedurally find the lowest objective function
  - Find the optimized parameter set from this

Expected Results

- Currently takes one hour per generation (per Dr. Liu)
- Minimum goal: 10x speedup (~10min per generation)