

# 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

*(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



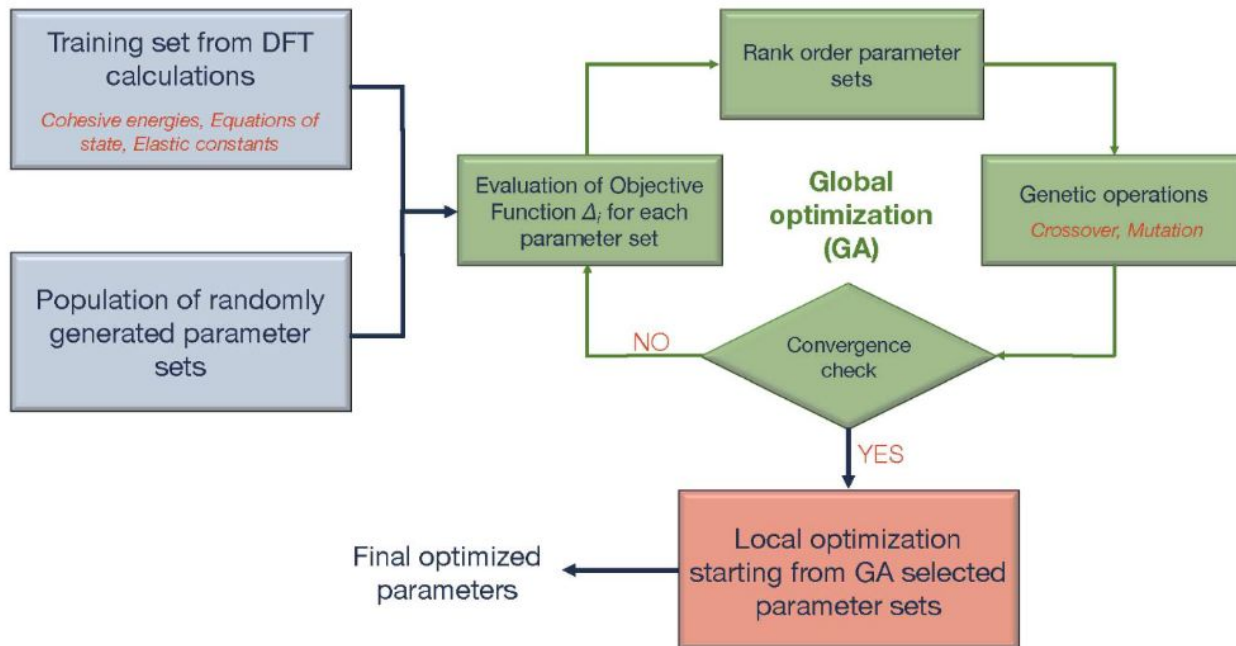
# Genetic Algorithms

- 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



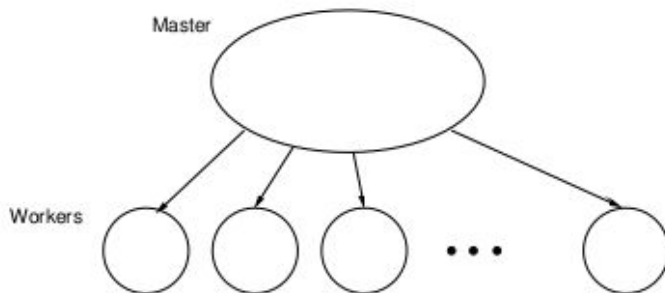
# LAMMPS

- Molecular dynamics simulator
- Works in serial or in parallel (supports OpenMPI)
- 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
    - Dataset size ideally fits into cache
  - OpenMPI will be used for distribution
  - Computation heavy, so communication expense small



(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)

