A Combinatorial Genetic Algorithm for Computational Doping based Material Design

Emrah Atilgan
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC, 29208, USA
+1 (803) 777-1912
atilgan@email.sc.edu

Jianjun Hu
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC, 29208, USA
+1 (803) 777-7304
jianjunh@cse.sc.edu

ABSTRACT
Computational material discovery can search large design space to identify promising candidates for experimental material design. Density Functional Theory (DFT) based first principle calculation has been able to calculate many electrical and physical properties of materials, making them suitable for computational doping based material discovery. In material doping, given a base material, one can change its properties by substituting some elements with new ones or adding additional elements. In computational doping, we have a grid of atoms in a supercell, some of which can be substituted with dopant atoms. For each such substitution, we use the Vienna Ab-Initio Simulation Package (VASP) to calculate its physicochemical properties, which takes about 30 hours for a grid of 75 atoms. This is a typical optimization problem with expensive evaluation functions. Here we developed a genetic algorithm for finding the most stable structure of the doped material with the lowest free electronic energy. It can reduce the running time for computational doping by up to 70%. We used $SrTiO_3$ perovskite as the base material and Nb as the substitution element.

Categories and Subject Descriptors
I.2.8 **[Artificial Intelligence]:** Problem Solving, Control Methods and Search – *heuristic methods*

Keywords
Combinatorial Genetic Algorithms; Density Functional Theory; Expensive Fitness Optimizations.

1. INTRODUCTION
Genetic algorithms (GAs) [2] are powerful search algorithms for applications in various science and engineering fields. GAs generally produce satisfactory results in a reasonable amount of time. But in some cases, calculation of the objective function as the fitness value can take much longer. Such expensive evaluations make it impractical to explore all the search space. This is especially true for problems with exponentially increasing search space with regard to the problem sizes. One of such problems is the computational material design based on DFT calculations [1].

GECCO'15 Companion, July 11-15, 2015, Madrid, Spain
ACM 978-1-4503-3488-4/15/07.
http://dx.doi.org/10.1145/2739482.2764700

In this paper, we present a combinatorial genetic algorithm to reduce the number of expensive fitness calculations for DFT based material design. We tested our algorithm on finding the best doping positions for Nb-doped $SrTiO_3$, a solid oxide fuel cell (SOFC) material.

2. RELATED WORK
In this paper, we focused on computational discovery of materials for Solid oxide fuel cells (SOFCs), which are the energy conversion devices that convert chemical energy directly into electrical energy. Despite years of experimental efforts, the ideal set of SOFC materials remain to be discovered.

Computational Doping based on DFT Calculation
To speed up the search for better SOFC materials, it is helpful to explore the design space using first principle (DFT) calculations, in which quantum simulation is used to calculate atom-level material properties and energies. The common approach is to first build a supercell from unit cells and then replace the elements at some positions with the possible dopant atoms.

3. METHOD
Given a base material such as $SrTiO_3$, there are dozens of possible dopant elements for Sr/Ti/O, and for each dopant elements, there are many possible positions for the substitution, which leads to a complex doping space. Consider the example of a supercell consisting of 75 atoms of $SrTiO_3$ material with Nb as the dopant element. If 4 Nb atoms are substituted with 4 Ti atoms, it makes approximately 27% Nb-doped $SrTiO_3$. Since there are 15 Ti atoms in the supercell, we need to choose 4 of them to substitute with the Nbs. It turns our problem into a combinatorial optimization problem. Mathematically, 15-choose-4 is equal to 1365, which means there are exactly 1365 different options for the placement of Nb atoms in the supercell for the Nb-Ti substitution It is practically infeasible to exhaustively search all possible doping configurations even using DFT calculation since each DFT calculation of a configuration needs 30-35 hours. We used the VASP [3], the state-of-the-art DFT calculation software, to evaluate the energy of each configuration. This evaluations were used as our fitness values.

In our research we used a generation-based genetic algorithm. A binary representation is used for individuals but with some constraints. Each individual's length is equal to the possible positions. The only bits that occupy the candidate positions are equal to 1, and the remaining are 0 (see Figure 1).

0	0	1	0	0	0	0	1	0	1	0	0	0	1	0

Figure 1: A representation of an individual that places the 4 Nb atoms at 3rd, 8th, 10th, 14th position of 15 possible positions.

We start the genetic search by creating a random population. Simultaneously, we create another array, named *Pool* to keep track of all un-evaluated configurations. This precludes the re-evaluation of individuals, preventing the time consuming tasks of unnecessary fitness calculations. We implemented three different GAs depending on how we avoid evaluating the same individual multiple times and what we should do when a created individual has already been evaluated before (explained in section 3.1).

We run 10 generations with population size 40, with 0.1 elite probability. Elite individuals were also used for crossover and mutation, as with any others. *Tournament selection* is used with tournament size 3. Crossover probability is 0.9 with 0.1 mutation probability. The mutation is simply flipping bits. One of the 1's is chosen randomly and swapped with a randomly chosen 0.

3.1 Implemented Genetic Algorithms

We implemented and evaluated three genetic algorithms for finding stable doped materials based on VASP DFT calculations.

GA-basic

If the crossover or mutation operators create an individual that has already been evaluated before, we need to replace it with a new one. In this case, GA-basic randomly chooses one single position of the individual, let's say 8^{th} position of [3, 5, 8, 11], and then look for individuals consisting of the rest positions, 3^{th}, 5^{th}, and 11^{th}. The algorithm chooses one of the available individuals from the *Pool* and creates the new individual. If there is no individual in the *Pool* consisting of these 3^{th}, 5^{th}, and 11^{th} positions, then the algorithm chooses another positions to change. If there is no individuals left in the *Pool* for 3-tuples, then the algorithm randomly chooses a pair, and so on.

GA-SS

GA-SS is genetic algorithm with statistical similarity. Instead of choosing only similar positions like in Ga-basic, GA-SS checks the previously evaluated individuals containing each position separately. For example, let the individual be [3, 8, 10, 14] which placed the Nb atoms in 3^{th}, 8^{th}, 10^{th}, 14^{th} positions, respectively. Considering we randomly choose 8^{th} position and we will keep 3^{th}, 10^{th} and14^{th} positions occupied. Then, we look for the candidates from *Pool* which they haven't been evaluated before. We determine the positions consisting of [3, 10, 14] and keep the fourth position as candidate for substation for 8^{th}. Let's say 1^{st}, 2^{nd}, 5^{th}, 7^{th}, and 12^{th} positions are available and can be used. Then, we calculate the average fitness values of the individuals which have already been calculated and choose the best position which has the maximum average fitness (lowest energy).

GA-SC

GA-SC is a genetic algorithm with statistical crossover. The method is similar to GA-SS. However, we implement this statistical sampling process at different levels of the algorithm. While GA-SS uses statistical sampling after crossover or mutation, GA-SC uses statistical sampling during the crossover process. If the parents have no common positions, uniform crossover is applied and new offspring are created. If the parents have common positions, these common positions are kept for each offspring and the statistical sub-combinatorial approach (explained in GA-SS) is applied for "not-common positions".

4. RESULTS

4.1 27% Nb-doped $SrTiO_3$

In this system, the supercell is created from 15 unit cells (5x3x1). There are 15 Sr, 15 Ti and 45 O atoms in the supercell. Our goal is to find the best positions for 4 Nb atoms to substitute with 4 Ti atoms. In this real-world optimization problem, 1365 DFT calculations are needed to find the best dopant positions if done exhaustively and each takes approximately 30 hours on a 12-core 2.4GHz CPU Linux computing node.

We tested the ability of our different GA implementations for finding the optimal doping configurations given a fixed numbers of evaluations. Here we only allow 400 fitness evaluations out of 1365, and compared the best found results of three GAs with the exhaustive search results. 50 independent runs for each algorithm were executed to check the robustness of the algorithms (see Figure 2). The average value of GA-SS outperformed GA-basic. However, GA-SS can be prone to becoming stuck in local optima, resulting in a higher standard deviation. The GA-SC not only found the smallest average and the standard deviation, but also found top 3 best solutions in 48 out of 50 runs. That means we can save 965*30=28950 hours of computing resources on a 12-core high-end Computer.

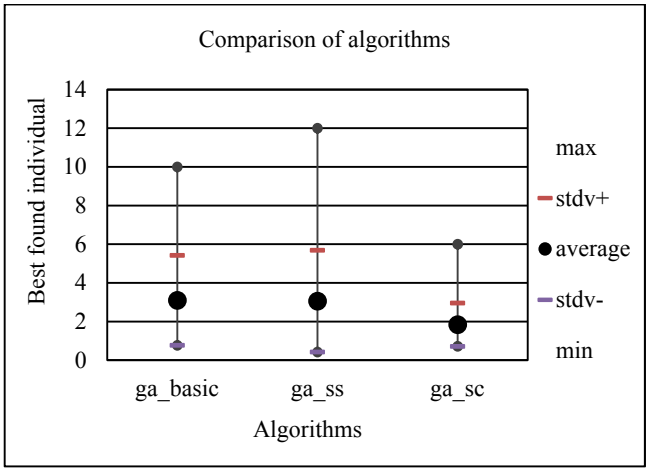

Figure 2: The results of 50 independent runs of three algorithms to check when they find the best solution. The numbers in y-axis show that the ranks of the best results found by each algorithm in each run.

5. CONCLUSIONS and FUTURE WORK

We have proposed a specialized combinational genetic algorithm for finding stable doping positions in computational doping experiments based on VASP DFT calculations. Experimental results showed that our algorithms are robust and can achieve up to 70% saving of the computational resources and speed up compared to exhaustive search approach currently used by most of computational material scientists.

6. REFERENCES

[1] Hautier, G. et al. 2012. From the computer to the laboratory: materials discovery and design using first-principles calculations. *Journal of Materials Science*. 47, 21 (May 2012), 7317–7340.

[2] Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems: An introductory analysis with applications to biology, control, and artificial intelligence.*

[3] Kresse, G. 1996. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical Review B*. 54, 16 (1996), 11169–11186.