Improvement of the Evolutionary Algorithm on the Atomic Simulation Environment Though Intuitive Starting Population Creation and Clustering

Nicholas Kellas
Department of Chemistry and Biochemistry
California State University Fullerton
Fullerton, CA
nickellas094@csu.fullerton.edu

Michael N. Groves
Department of Chemistry and Biochemistry
California State University Fullerton
Fullerton, CA
mgroves@fullerton.edu

ABSTRACT
The Evolutionary algorithm (EA), on the Atomic Simulation Environment (ASE), provides a means to find the lowest energy conformation molecule of a given stoichiometry. In this study we examine the ways in which the initial population of molecules affect the success of the EA. We have added a set of rules to the way in which the molecules are created that leads to more chemically relevant structures using chemical intuition. We have also implemented a clustering program that selects molecules that differ from each other from a large pool of molecules to form the initial population. Through testing of EA runs with and without clustering and intuitive population creation, the following success rates were obtained; no intuition and no clustering, 28±3%, no intuition with clustering, 31±4%, with fixed intuition but without clustering, 49±5%, with fixed intuition and clustering, 49±4%, with variable intuition and without clustering, 47±4%, and with variable intuition and clustering, 50±3%. A significant increase in success rate was found when implementing intuitive population creation while clustering the initial population seems to marginally help as the population becomes more diverse.

Keywords
Evolutionary Algorithm, Agglomerative Clustering

1. INTRODUCTION
Determining the structure of a molecule is an important heuristic in understanding its function. As chemical structures become larger, predicting the global minimum becomes increasingly challenging. To find global minimum structures, several strategies have been developed including molecular dynamics [1], Monte Carlo [2], particle swarm optimization [3], random search [4], as well as evolutionary algorithms [5]. Evolutionary algorithms are well suited for chemical structure problems because they can quickly cover large regions of configuration space and can be easily parallelized dramatically improving the search efficiency.

DOI: https://doi.org/10.22369/issn.2153-4136/11/2/5

An Evolutionary Algorithm (EA) is a metaheuristic that uses principles inspired by natural selection to find optimized solutions to complex problems. A population is created from possible solutions and through a process of combining information, optimized solutions can be found [6]. EA's utilize terminology derived from evolution theory such as individual, parent, offspring, and fitness. The EA we use involves 6 steps [7]: first, a starting population of individuals is generated from possible solutions to the complex problem being computed; second, the individuals are evaluated based on the desired metric and ranked from best solution to the problem to worst; third, two of these solutions are then selected to undergo the fourth step, a recombination of information to generate a new solution made from parts of the information from each; fifth, the new solution is compared to the population of solutions, and the worst solution is removed from the population; lastly, a check is done to evaluate if a completion condition has been met, and the selection, crossover, and re-evaluation steps are repeated until one of these conditions is met.

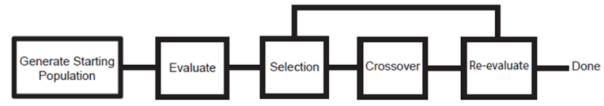

Figure 1. Steps of an evolutionary algorithm: 1. Generate Starting Population, 2. evaluate the starting population energy, 3. select two candidates to undergo crossover, 4. crossover of two molecules via cut and splice, 5. re-evaluate population to determine if improvement has occurred, and 6. check for completion condition (as the end of the re-evaluation step). This process loops from selection through check for completion until a completion condition is met.

Originally written on by Alan Turing in 1950, computational scientists have theorized about machines using the principles of evolution to solve problems since the beginning of computational science [8]. By the 1960s "artificial evolution" had become a widely-used optimization method and notably was used by Ingo Rechenberg to generate new aerodynamic wing designs [9]. At the same time, Lawrence J. Fogel developed evolutionary programming in his attempts to create artificial intelligence [10]. The modern evolutionary algorithm was developed by John Henry Holland and published in his 1975 book, "Adaptation in Natural and Artificial Systems" [11]. In the 1980s EA's began to see commercial use by General Electric [12] and Axcelis, Inc. [13] and in 2006 NASA used an EA to develop the evolved entenae [11].

Today, evolutionary algorithms are used in the field of computational chemistry to find the morphology of bulk surface, and nanoparticle systems [14, 15, 16, 17].

The ASE EA begins by generating an initial population of molecules, made from the desired molecular formula by plotting the positions of the atoms as a virtual object. This step can be done stochastically, by assigning the positions of each atom randomly, or semi randomly by specifying properties that the molecules must have such as having set distances between each atom or set angles formed by 3 atoms. By placing restrictions on the kinds of molecules that can be made we are limiting the regions on the potential energy surface that can be represented in the starting population. The algorithm makes a number of these molecules specified by the scientist to form the starting population. Each of these molecules then has its potential energy calculated and they are ranked from highest energy to lowest energy.

Two of the molecules from the initial population are then selected to undergo crossover. This selection is skewed to favor lower energy molecules over higher energy molecules so that more favorable structures are more likely to be replicated [15]. Each of the two molecules are then cut and spliced [7] along a plane and the portion of the molecule on one side of this plane is combined with a counterpart from the other molecule as demonstrated in Figure 2. This process is random and may result in the creation of molecules with incorrect molecular formula. The newly created molecule will either be rejected, and new matched planes will be made, or the molecule will be altered, with extra atoms being removed or added to random locations to correct the molecule.

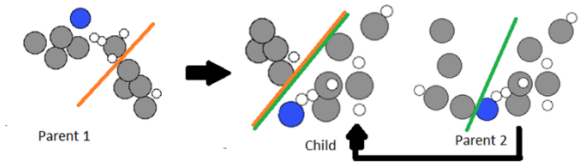

Figure 2. Crossover of molecules with a portion of the structure of two parent molecules making one child molecule.

The new molecule is analyzed and compared with the population and the molecule with the highest energy is removed. If the newly created molecule is higher in energy than all the molecules in the population, it is discarded, and the next generation is identical to the first. If a molecule from the population is higher in energy then the newly created individual, the highest energy molecule from the population is removed and the new molecule is added. This new generation is again sorted from low to high energy.

Through this process, the energy of the population will either stay the same or be lowered with each new generation. The selection, crossover, and analysis steps of the EA will be repeated until a set goal is met. This goal can be a number of generations, a desired energy being reached, or convergence, where the same lowest energy molecule is created frequently enough to suggest the population has become trapped into some local minima and cannot escape. This minimum may be the global minima, or it may be a local minimum.

The most computationally intense portion of the EA is the analysis step, in which the candidate molecule is optimized into its nearest local minimum energy structure and its energy is calculated. The energy calculation can be done in many ways depending on the accuracy needed and the complexity of the system being calculated.

Although energy calculators are effective in locating local energy minima, for large molecules, it becomes increasingly more costly for them to find the global energy minimum. When a molecule is diatomic, containing only two atoms, there is only one optimized shape for it to be in, shown in Figure 3. This shape is the point where the intermolecular attractions most overwhelm the repulsions.

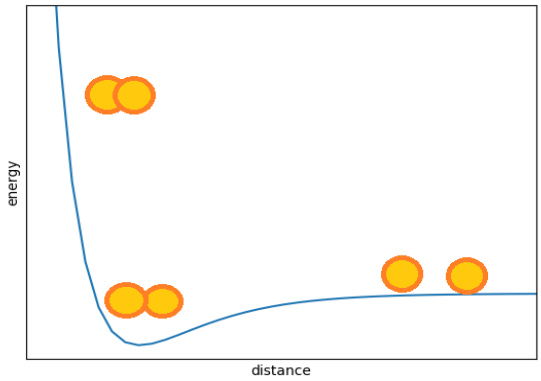

Figure 3. Morse potential diagram showing the energy of a diatomic molecule in response to the distance between the atoms.

When dealing with multi-atom molecules, the interactions between the atoms becomes increasingly complex and multiple configurations can be created that are lower in energy then all other nearby structures; these are known as local minima. Even though these structures are stable the molecules are still capable of restructuring into more stable configuration. The only point where this is not true is the global minimum, the structure that is the most stable for the given stoichiometry of atoms. The many ways that the molecule can be arranged can be represented in a 3N dimensional matrix where N is the number of atoms in the molecule. The dimensions are the x, y, and z coordinates of each atom. This matrix is the potential energy surface of the molecule, and, when energy is added as an extra dimension, it forms a topographical map of all molecular structures involving the target molecular formula.

It is impossible for a human to view a 3N dimensional matrix for any polyatomic molecule, so a simplified view of the potential energy surface must be designed to be understood. For the C_9H_7N formula, a representation of the surface is shown in Figure 4 from data collected in this study. This 2-dimensional contour graph represents the farthest carbon-nitrogen distance and farthest carbon-carbon distance of each molecule as its x and y coordinates with the energy of the molecule shown by its color with red being high energy and blue being low energy. The largest region of low energy occurs when the farthest carbon-carbon and carbon-nitrogen distances are both small, meaning the molecule is coiled into itself in the form of rings. The molecule quinoline is known to be the global minimum of this formula and is composed of two six-member rings meaning it would exist in this low energy area. The region of the graph with greatest carbon-carbon and carbon-nitrogen distance would conversely represent fully stretched linear molecules with nitrogen on one end.

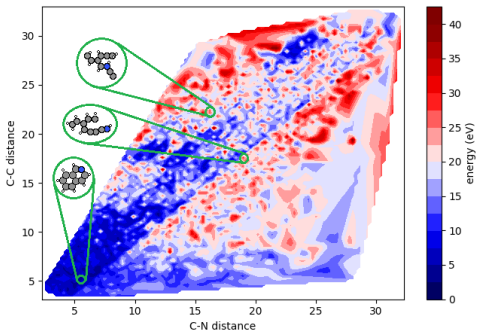

Figure 4. Simplified Potential Energy Surface for C_9H_7N using the furthest carbon-nitrogen distance and carbon-carbon distance in a molecule.

The starting population can be made more diverse by applying clustering [17]. Clustering is the process of grouping individuals based on similarity. In agglomerative hierarchical clustering, each individual is evaluated by analyzing target features. In this case, the features used are the intramolecular carbon-carbon and carbon-nitrogen distances that are compared to each other to determine the similarity between the individuals. When clustering begins each molecule is its own cluster. The program then begins clustering similar molecules together. These clusters are then evaluated based on the averaged characteristics of all the molecules in the cluster. This process continues until a similarity goal is reached, meaning that no clusters are similar enough to be clustered together based on the desired similarity level. This mapping can be seen in Figure 5. When using clustering, a large number of molecules are generated and then a similarity level [7] is found that results in a number of clusters equal to the numbers of individuals required for the starting population. By randomly selecting one molecule from each cluster to form the starting population the diversity of the population will be greater than when the population is made without clustering due to the larger pool of molecules to choose from [18].

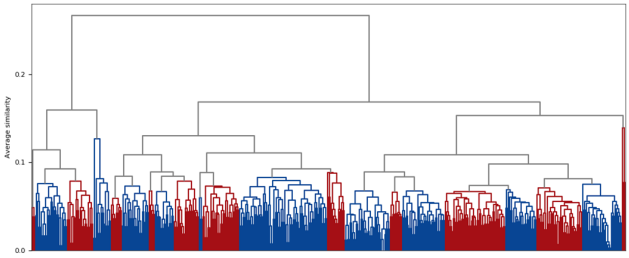

Figure 5. Example agglomerative hierarchical clustering dendrogram at an inconsistency threshold of 2.8.

When a starting population is made without clustering, there is a likelihood for areas of the potential energy surface to not be represented in the starting population. By generating a large number of molecules and then clustering, the diversity of the starting population can be increased by choosing molecules from all clustered groups. We predict that this increase in diversity will increase the likelihood that molecules similar to the global minimum molecule will be included in the starting population leading to an increase in EA efficiency.

There is, however, a competing element proposed by Oganov [14]. Oganov predicted that when dissimilar molecules undergo cut-and-splice, the offspring will be high in energy due to the merging of incompatible structures. This would mean that the increased diversity of the clustered populations would decrease the efficiency of the EA because of an increased number of unproductive cycles where lower energy molecules are not created and the population does not evolve.

In this study we will show that by building the initial population with chemically intuitive molecules, we can significantly increase the success rate of the GA to find the global minimum. Furthermore, we will show that the role of clustering is more nuanced. In cases where the population is diverse, agglomerative clustering seems to have a minor benefit to the success rate of the GA. When the population is more uniform, agglomerative clustering seems to hinder the success of the GA. This appears to be a demonstration of the trade-off between creating and maintaining a diverse population while being able to form competitive candidate structures which can enter the population.

2. EXPERIMENTAL AND COMPUTATIONAL DETAILS

By augmenting the way in which the EA creates its starting population, the efficiency with which it locates the most stable molecular configuration can be improved. For this purpose, the starting population generation was altered to generate molecules using standard hybridized orbital geometries to determine bond angles and bond lengths that would result in more optimized molecules. An agglomerative clustering program was also implemented in which molecules were analyzed for similarity and divided into groups, or clusters. C_9H_7N was used as the molecular stoichiometry for testing purposes because the global minimum (GM) is known (quinoline), the potential energy surface is well explored, and because the presence of a double ring structure in the GM makes it computationally challenging to find.

The method was written to generate molecules based on rules of molecular geometry using the standard documented orbital hybridization geometries of the atoms. The program begins when it is fed the desired molecular formula. An atom is chosen randomly from those available to fit the formula and placed in position (0,0,0) of the configuration space and then is given a geometry consistent with the element. For example, Carbon can form sp^3 hybridization, where four single bonds are formed 109.5 degrees from each other; sp^2 hybridization, where one double bond and two single bonds are formed 120° apart; or sp hybridization, where either one triple bond and one single bond, or two double bonds, are formed 180° from each other. Unit vectors signifying these bonds are added to the information stored for this atom. When another atom is chosen, an already-placed atom is selected for it to bond to. The new atom is placed along an available unit vector a distance away from the bonding atom equal to the covalent radii of both atoms added together. The program then checks if the atom is too close to any other atoms, defined as less than 70% of a bond radius as determined above, to prevent energy calculation errors that may occur when atoms are placed inside each other's atomic radii. The new atoms are then similarly assigned a geometry and aligned so that one of the unit vectors is pointed towards the bonded atom. Finally, the unit vectors of bonds that were used in this process are removed from the list of available bonds. Molecules created in this manner have many advantages over molecules created randomly; these molecules require less time to optimize and are far more likely

to form complex, chemically-relevant shapes such as rings and long chains.

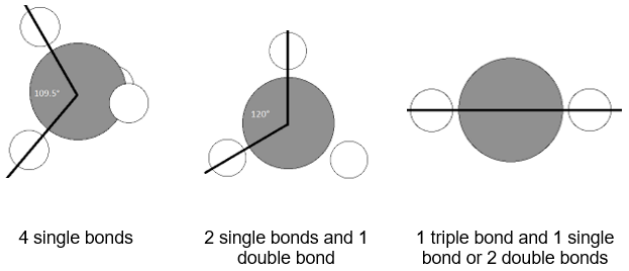

Figure 6. Molecular geometries of carbon.

Two different forms of intuitive population generation were tested in this study; in fixed hybridization, each hybridization type was given equal likelihood to be chosen for each atom. 20 of these molecules are generated in the non-clustered cases and 500 are created in the clustered cases and then made into 20 clusters with one molecule chosen randomly from each to form the starting population. In the second, called variational hybridization, variable probabilities to select different hybridizations for carbon and nitrogen were utilized in the following manner: the possible hybridizations of both carbon and nitrogen are sp, sp^2, and sp^3. Each of these hybridizations are assigned a 0, 25, 50, 75, or 100% chance to be selected to be assigned to each atom with the restriction that the total probability to select any of the three is 100%. All potential combinations for the probabilities are sampled. For example, a molecule could be generated with each carbon having a 50% chance to be sp^3 hybridized, a 50% chance to be sp^2 hybridized, and a 0% chance to be sp hybridized. The nitrogen is independently assigned to have a 100% chance to be sp hybridized with no chance to be sp^2 or sp^3 hybridized. Not all these probabilities can be used to make a viable molecule with a given stoichiometry. As a result, the program must also be able to identify and skip those cases. When this generation method is used in the non-clustered case, each variation is used to produce 4 molecules, and then 20 are chosen at random from this pool. In the clustered case, 4 molecules are created for each variation and then clustered to generate 20 clusters, and one molecule is chosen from each randomly to form the starting population. These methods are compared to the null case by running random molecule generation trials. For the non-clustered random cases 20 molecules are generated and for the clustered cases 500 molecules are created randomly and then clustered down to 20.

The most obvious alternative to the intuitive population generation method developed here would be one involving the SMILES technique [14] for generating molecular structures. This technique can be used to generate varied molecules easily and with built-in chemical intuition. The reason we chose to write our own program for molecule generation is to have more control over the amount of intuition used in molecule creation and so that the EA can be used to study metallic compounds, as SMILES does not support inorganic complexing [20].

The clustering algorithm takes in a large number of unrelaxed molecules and sorts them based on intermolecular distances. For the test molecular formula, C_9H_7N, the fact that there was only one Nitrogen was taken advantage of, and the C-N and C-C distances were compared. Hydrogens are ignored because it was found that including C-H and N-H distances did not meaningfully change the functionality of the clustering process [7]. The factors that will be effective in clustering are strongly dependent on the stoichiometry given, and so additional work must be done to fit clustering to an EA run for different formulas. The molecules are then fitted to centroids as described in the introduction to form a number of clusters equal to the desired starting population size using the methods detailed in Jørgensen et al [7]. One molecule from each cluster is chosen at random to form the starting population.

Molecules were optimized using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [21] and their energy calculated using Density Functional Tight Binding (DFTB) method [22] with the calculator DFTB+ [23]. A more popular energy calculation method is Density Functional Theory (DFT) [13] which calculates the energy of the system using the density of the electron clouds. While this method is a good mix of fast and accurate and is widely used, it is best to have an extremely fast method for testing molecular EAs. For this reason, the empirical method Density Functional Tight Binding (DFTB) is used. DFTB uses a library of ab initio constants to quickly calculate energy at the expense of accuracy. Using this method reduces the time spent optimizing a 17-atom molecule, like the one used for testing this EA, from minutes per molecule to seconds per molecule. Though the method is not highly accurate, DFTB+ was chosen due to the speed at which it calculates energy. This was acceptable to test the hypothesis because high accuracy of calculation was not needed to test the program's ability to move from relatively high energy molecules to low energy molecules [7].

There are two stop conditions for each of these trials. The first is if the EA locates quinolone, and the second is if 5000 cycles have occurred. We set a step count end condition, because we believe that after 5000 cycles the chance of the EA finding quinoline goes down greatly. This is because the EA has likely become stuck in some local minima with all the members of the current population being too similar to find any other structure and escape. This is demonstrated in Figure 7 where we can see that the likelihood of the EA finding quinoline goes down as it reaches higher cycle counts.

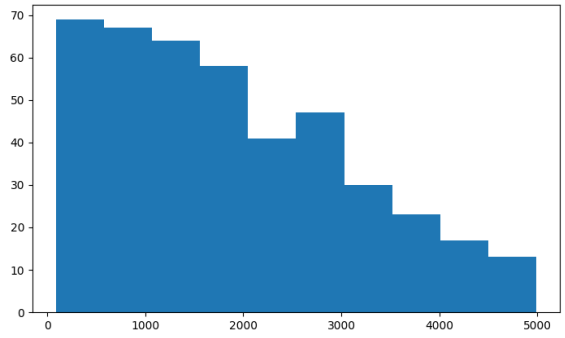

Figure 7. Histogram of the completion steps of the fixed intuitive starting population generation with clustering runs.

3. RESULTS

Each of the six permutations of the experiment were run until they converged to a consistent success rate and are shown in Table 1. The null case, run with random population generation and without clustering, was found to have a 28±3% success rate.

The runs conducted with random population generation and with clustering were found to have a 31±4% success rate. The runs conducted with fixed intuitive starting population generation and without clustering were found to have a 49±5% success rate, and the runs conducted with both fixed intuitive starting population generation and clustering were found to have a 49±4% success rate. For variational intuitive starting population generation without clustering, a 47±4% success rate was found. Finally, for variational intuitive starting population generation with clustering, a 50±3% success rate was found.

Data from the above trials is re-expressed in Figure 8 showing the percentage of runs that locate quinoline at or before each cycle count until the 5000th cycle. The value of this plot at the 5000th iteration corresponds to the Success Rate column in Table 1.

Table 1. Evolutionary Algorithm Results

Run type	Completed	Found GM (%)	Success rate (%)	Average	Standard deviation
Random, not clustered	667	185	28±3	4295	1330
Random, clustered	462	143	31±4	4179	1421
Fixed, not clustered	480	234	49±5	3378	1881
Fixed, clustered	618	303	49±4	3509	1771
Variational, not clustered	701	330	47±4	3615	1761
Variational, clustered	959	484	50±3	3431	1811

A representation of the energy of the molecules in the starting populations generated with each method is shown in figure 9. The box plots illustrate the median energy as well as all four quartiles of the energy of the molecules relative to the energy of quinoline. The top three quartiles of the randomly generated starting molecules are in the same range as the top of the highest quartile of the intuitively generated populations. Furthermore, the energy ranges of the intuitively generated starting populations overlap significantly.

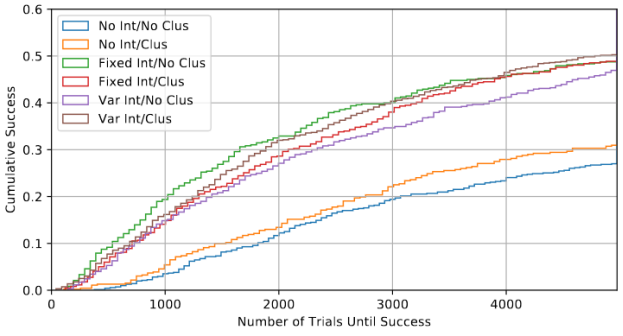

Figure 8. Success rate for each trial type expressed as percent completed by cycle.

4. DISCUSSION

The implementation of intuitive starting population generation led to a significant increase in success rate, going from a roughly 30% success rate for the two randomly generated cases to roughly 50% success rate for the intuitively generated cases. Intuition led to a sharp increase in the efficiency of the EA by selectively generating molecules in lower energy areas of the potential energy surface. There is also the possibility that the intuitive population generator created molecules with a structure similar to the global minimum. From these results, we can say that seeding the population with chemically intuitive molecules has a positive effect on the EA finding the GM in the case of quinoline.

As reported in Table 1, the average number of candidate structures to find the global minimum was higher for the random cases, but the standard deviation was lower. This lower deviation is theorized to be due to the average completion cycle being closer to 5000 cycles where data collection was set to end. This causes the portion above the average to be cut off at that point, leading to an artificially lower standard deviation.

When clustering is applied to the EA runs without intuitive population generation, the average success rate increases by 3%. When applied to the EA runs with fixed intuitive population generation, little or no change in success rate is found, and when applied to the EA runs with variable intuitive population generation, the success rate increases by 3%. These differences are within the margin of error, so we cannot say that there is a definite difference. We can only remark on the pattern of the differences.

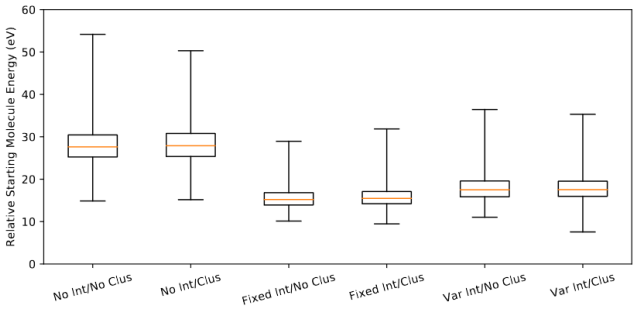

Figure 9. Box plots for the starting populations of each trial type.

Figure 8 shows the cumulative success rate as a function of cycle count. When looking at the trials completed using randomly generated starting populations, it can be seen that there was a consistently higher success rate for the clustered case at all cycle counts. Because these molecules were created using less rules then the intuitive populations, we believe they are capable of forming the most diverse populations and as such benefit most from clustering. Clustering was effective in seeding the starting population with molecules whose structures had similarities with the global minimum. It should be noted that the difference between the clustered and non-clustered success rate never varied greater than the margin of error, but the consistent difference in success is indicative of a positive, albeit minor, effect.

For the populations generated using fixed intuitive starting populations, it was seen that the non-clustered trials were much more successful for the first roughly 3000 steps. After this point the clustered runs slowly caught up to them until they had the same success rate. The fixed intuitive populations are the most focused

due to only utilizing one hybridization ratio. This means that the starting population will contain similar molecules. Clustering attempts to maximize diversity, but because this population creation method produces a fairly homogeneous population, it instead creates a population that has a greater challenge forming viable candidates during crossover. This agrees with Oganov's theory [9] that diverse molecules have difficulty producing lower energy offspring when cut-and-spliced. Because of this effect, the non-clustered populations can move toward the global minimum structure more quickly while the clustered populations undergo ineffective crossover until the population filters out its diversity through repeated selection of the more closely related molecules.

For the populations generated using variable intuitive population generation, the clustered populations outperformed the non-clustered cases. These populations were made from many different hybridization ratios, which presumably produced a much more diverse initial population then the fixed intuitive method. Clustering is beneficial in this case to ensure that this diversity is maintained. It appears that Oganov's crossover hindrance overcomes the benefits of clustering only in cases were the population generation method is apparently more uniform.

Clustering initial populations and using crossover to generate new candidate structures appear to be competing processes. Clustering is used to increase the diversity of a population while crossover benefits from homogeneity in the population to form viable candidate structures. The benefit or penalty due to clustering appears to be minor and temporary. In contrast to this, building molecules using chemical intuition appears to significantly increase the success rate of the GA compared to random population generation. This can be attributed to the fact that it takes many cycles for the random population to start creating candidates that exhibit the same geometries that the intuitive generator creates in the starting population. This is a deficit which cannot be overcome. Furthermore, the two different intuitive generation schemes produced different types of starting populations with similar results. This indicates that this process might be robust and tolerant of imperfect schemes which can result in significant improvements in other applications.

Although this experiment was conducted on a single molecular formula, the techniques used are readily applicable to other molecules to test for optimized structures. C_9H_7N was chosen to test improvements on the genetic algorithm because of the complexity of its structure and the relatively low atom count leading to comparatively low computational time. The technique has been built so that any number of atoms and configurations can be applied including organics and metals. For example, a surface reconstruction of a step edge of TiO_2 is a second test case that has been used previously [7] and would validate the increased cumulative success rate of this methodology. A similar starting population generator can be created based on typical Ti and O hybridizations so that a similar study could be conducted.

5. CONCLUSIONS

This study was conducted to examine the effect that intuitive starting population generation and clustering have on an evolutionary algorithm. Testing was conducted using the EA on the Atomic Simulation Environment [24] written for the Python programming language. The effect of intuitive population generation and clustering was tested by running EA's on the molecular formula C_9H_7N until the known global minima structure, quinoline, was found, or until 5000 cycles were completed. Intuitive population generation was tested by comparison with random population generation and was split into two forms of intuitive population generation: a form where each hybridization type had equal chance of being selected, and a variable type where molecules were made with various likelihoods of hybridization selection. The effect of clustering on the EA was evaluated by testing each of these population generation methods with and without clustering, creating baseline runs and comparing them to modified runs. It was found that both methods of intuitive population generation had a similar positive effect on the success of the EA, increasing the success rate by approximately 30% in 5000 iterations. Clustering was found to have little effect on the efficiency of the EA, with all results having overlapping confidence intervals between the clustered and non-clustered versions of each generation type. We can remark on trends showing a relationship between the population diversity possible from each molecule generation method and the effect of clustering. The methods that generated the most diverse populations benefited from clustering due to an increased likelihood of the population being seeded with molecules with similar structure to the global minimum structure. However, methods that generate focused populations that include global minimum-like structures were slowed in their ability to find the global minimum due to poor crossover between dissimilar molecules.

6. REFLECTION

This project was conducted as part of the Blue Waters Student Internship Program. At the start of the internship, I attended a two-week long computational science institute hosted by the Shodor foundation at the University of Illinois at Urbana-Champaign. There, my fellow interns and I were taught many facets of high performance computing (HPC) including basics like programming in the C programming language and submitting jobs to a supercomputer to compute large problems as well as more complex areas of HPC including many different methods for program parallelization and how to use them to the greatest effect. The information I received from the two-week institute and the internship as a whole have been invaluable for both myself and my fellow researchers in the Groves lab at California State University Fullerton by giving me the knowledge and skill to conduct my project and help others to understand their projects in the field of computational chemistry.

Perhaps just as important to me as the knowledge I received, the friends and connections I have made as a result of this internship have been a massive source of opportunity and community to me. The internship put me into contact with 25 of the most brilliant young computational scientists I have ever met, the staff of Shodor, the National Center for Supercomputing Applications, and the Blue Waters supercomputer and I became part of a community of like-minded individuals that have pushed me to be the best computational scientist I can be, and I can only hope I have done the same for them. The confidence I gained from becoming a part of this community has been the most noticeable benefit of my involvement.

7. ACKNOWLEDGMENTS

This study was conducted as part of the Blue Waters Student Internship Program, which is run by Shodor with funding provided by the National Science Foundation. Data was collected using Kepler, a cluster located at California State University, Fullerton. Clustering code was provided by Mathias S. Jørgensen.

8. REFERENCES

[1] Goedecker, S. J. 2004. Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems. J. Chem. Phys. 120, 21, 9911-9917. DOI=10.1063/1.1724816.

[2] Wales, D. J., and Doye, J. P. K. 1997. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. J. Phys. Chem. A 101, 28, 5111−5116. DOI= 10.1021/jp970984n.

[3] Call, S. T., Zubarev, D. Y., and Boldyrev, A. I. 2007. Global minimum structure searches via particle swarm optimization. J. Comput. Chem. 28, 7, 1177−1186. DOI= 10.1002/jcc.20621

[4] Pickard, C. J. and Needs, R. J. 2011. Ab initio random structure searching J. Phys.: Condens. Matter. 23, 5, 053201. DOI=10.1088/0953-8984/23/5/053201

[5] Hartke, B.J. 1993. Global geometry optimization of clusters using genetic algorithms. J. Phys. Chem. 97, 39, 9973−9976. DOI=10.1021/j100141a013

[6] Lohn, J. D., Hornby, G. S., and Linden, D. D. 2005, An evolved antenna for deployment on NASA's Space Technology 5 Mission. In Genetic Programming Theory and Practice II. Genetic Programming, Vol 8. U. M. O'Reilly, T. Yu, R. Riolo, and B. Worzel Eds. Springer, Boston, MA.

[7] Jørgensen, M. S., Groves, M. N., and Hammer, B. 2017, Combining Evolutionary Algorithms with Clustering toward Rational Global Structure Optimization at the Atomic Scale, J. Chem. Theory Comput. 13, 3, 1486-1493. DOI=10.1021/acs.jctc.6b01119

[8] Turing, A. M. 1950, Computing machinery and intelligence". Mind. 59, 236, 433–460. DOI=10.1093/mind/LIX.236.433

[9] Rechenberg, I. 1973. Evolutionsstrategie. Holzmann-Froboog, Stuttgart.

[10] Fogel, D. B. (editor) 1998. Evolutionary Computation: The Fossil Record. IEEE Press., New York, NY

[11] Holland, J. 1992. Adaptation in Natural and Artificial Systems. MIT Press. Cambridge, MA.

[12] Aldawoodi, N. 2008. An Approach to Designing an Unmanned Helicopter Autopilot Using Genetic Algorithms and Simulated Annealing. Doctoral Thesis. University of South Florida

[13] Evolver: Sophisticated Optimization for Spreadsheets. 2019. Palisade. https://www.palisade.com/evolver/

[14] Lyakhov A.O., Oganov A.R., and Valle M. 2010. How to predict very large and complex crystal structures. Comp. Phys. Comm. 181, 9, 1623-1632. DOI= 10.1016/j.cpc.2010.06.007

[15] Vilhelmsen L. B., and Hammer, B. 2014. A genetic algorithm for first principles global structure optimization of supported nano structures. J. Chem. Phys. 141, 4, 044711. DOI= 10.1063/1.4886337.

[16] Zhai, H., and Anastassia N. Alexandrova. 2017. Fluxionality of Catalytic Clusters: When It Matters and How to Address It. ACS Catal. 7, 3, 1905-1911. DOI= 10.1021/acscatal.6b03243

[17] Lones, M. A., and Tyrrell, A. M. 2007. Regulatory Motif Discovery Using a Population Clustering Evolutionary Algorithm. IEEE/ACM Transactions On Computational Biology And Bioinformatics. 4, 3, 403-414. DOI= 10.1109/tcbb.2007.1044

[18] Lipkowitz, K.B. and Boyd, D.B. 2003. Reviews in Computational Chemistry, Vol. 18. John Wiley & Sons, New York, NY.

[19] Anderson E, Veith GD, Weininger D. 1987. SMILES: A line notation and computerized interpreter for chemical structures. U.S. EPA, Environmental Research Laboratory-Duluth. Duluth, MN

[20] Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comp. Sci. 28, 1, 31-36, DOI: 10.1021/ci00057a005

[21] Fletcher, R. 1987. Practical methods of optimization (2nd ed.), John Wiley & Sons, New York, NY

[22] Elstner, M., Seifert, G. 2014, Density functional tight binding. Phil. Trans. R. Soc. London, Ser. A. 372. 20120483. DOI=10.1098/rsta.2012.0483

[23] Aradi, B., Hourahine, B., and Frauenheim, 2007. Th. DFTB+, a sparse matrix-based implementation of the DFTB method, J. Phys. Chem. A. 111, 26, 5678-5684. DOI=10.1021/jp070186p

[24] Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., Hermes, E. D., Jennings, P. C., Jensen, P. B., Kermode, J., Kitchin, J. R., Kolsbjerg, E. L., Kubal, J., Kaasbjerg, K., Lysgaard, S., Maronsson, J. B., Maxson, T., Olsen, T., Pastewka, L., Peterson, A., Rostgaard, C., Schiøtz, J., Schütt, O., Strange, M., Thygesen, K. S., Vegge, T., Vilhelmsen, L., Walter, M., Zeng, Z., Jacobsen, K. W. 2017. The Atomic Simulation Environment—A Python library for working with atoms, J. Phys.: Condens. Matter. 29, 27, 273002. DOI= 10.1088/1361-648x/aa680e