ARTIFICIAL INTELLIGENCE FOR MOLECULAR MODELING

Meparishvili Badri, Meparishvili Tamar, Gvinianidze Tamar
Georgian Technical University

Abstract

Molecular modeling is an extremely useful tool for design of nanodevices. The theory and techniques of artificial intelligence have great impact on modern chemical applications. Information is defined as the correlated entropy between two ensembles, and the concept of entropy represents a fundamental link between thermodynamics and information theory. This link between matter and information is most evidently manifested by the molecular constitution of the genetic information, which in the form of nucleic acids, is propagated by a thermodynamic mechanism. This concept combines the replication and variability of polymers that underlies Darwinian evolution. In building molecular phrases, composed by different nucleotides in relevant sequences, molecules are connecting in given order. The initial formation directs the synthesis of sequences, which logically are not random, and there is an optimization of structure within the system. Such optimization can be expressed in terms of Shannon's and fuzzy entropy and relates directly to the definition of information. Formation of the model system is based on two types of bindings of DNA occurring between the objects. The model and algorithms presented in this paper demonstrate the relationship between thermodynamics, information theory and the fundamental dynamics of living systems by analyzing accumulation of complexity in a computer based evolution system. With appropriate molecular CAD software, molecular modeling software and related tools are more promising way to explore and analysis of designs on a computer before actually molecular manufacturing systems. Advantages and limitations of new approaches for computer-aided molecular design are discussed.

***Keywords*:** Molecular nanotechnogy. Information theory. Entropy.

1. Target setting and system approach

Molecular nanotechnology is a proposed approach which involves manipulating single molecules in finely controlled, deterministic ways. More generally, molecular self-assembly seeks to use concepts of supermolecular physics and chemistry, and molecular recognition in particular, to cause single-molecule components to automatically arrange themselves into some useful conformation. Molecular nanotechnology, sometimes called molecular manufacturing, is a term given to the concept of engineered nanosystems operating on the molecular scale. It is especially associated with the concept of a molecular assembler, desired structure or device atom-by-atom using the principles of mechanosynthesis. Molecular electronics seeks to develop molecules with useful electronic properties. These could then be used as single-molecule components in a nanoelectronic device. The other important direction of development is research in modeling at the nanoscale and analyze of new modeling methods of inorganic molecular systems for creation of prospective materials. Practical applications of these materials are connected with the fabrication of molecular machines and molecular devices. Molecular modeling is an extremely useful tool for synthesis. The solving of molecular modeling problems is based on the following methods:

- Simplest computational method for modeling – uses classic physics principles.
- "Ball and spring model" (atoms and bonds) to approximate geometry and strain energy.
- Uses quantum mechanics and the Schrödinger equation to calculate the location of all valence electrons in addition to atom and bond locations.
- The standard computational chemistry calculation (Geometry Optimization) to find the lowest energy or most relaxed conformation for that molecule. Performs a series of computational cycles on the molecule.
- Semi-Empirical Method using empirical data from typical organic molecules to estimate locations of inner shell electrons and do theoretical calculations to determine the probability of finding an electron at any point in space.

- Object-oriented molecular modeling methods.
- The methods of Artificial Intelligence (Evolution programming, Genetic Algorithms, Artificial Neural networks and etc.).

2. Brief overview of AI

Usually, intelligent solutions lead to complex systems, where it is often difficult to prove the correctness of the presented solution. When intelligence is seen as the capability to solve (new) problems, it is possible to identify "intelligent" solutions in engineering. Intelligence in engineering means systems that are able to react appropriately to changing situations without input from a human operator. In other words, an intelligent algorithm is one that is able to solve problems that stem from changing situations. This definition, of course, encompasses a wide range of engineering applications and many different methods and algorithms. Generally there are some fundamental approaches in the field of artificial intelligence.

The most frequent example for biologically inspired computing is that of neural networks (NNs). A NN consists of interconnected neurons, each with a set of input and output connections. In principle, a neuron contains a simple add-and-compare mechanism that sums up the input signals and generates an output signal if a particular threshold has been exceeded. While the concept of such a neuron cell is very simple, a whole NN shows emergent properties such as learning and reasoning. NNs are extreme versatile. NNs support supervised and unsupervised learning. In supervised learning, back-propagation NNs are used [1]. During a training phase, the parameters of the NN are adapted until the system performs the desired function. Thus, the data processing mechanism of a NN cannot be programmed, understood, or verified in terms of rules.

A genetic algorithm (GA) is a derivative-free and stochastic optimization method that builds on ideas from the natural selection and the evolutionary process [2]. It is some kind of search algorithm that is advantageous if the given search space is too large to be searched by exhaustive search algorithms and too unstructured to be able to use straight forward search algorithms. Moreover, a GA needs only a minimum on information about the problem to be solved and is thus easily applied. Basically, a GA needs an initial population of "genes", an algorithm that allows to crossmix these genes, and a fitness function that produces a comparable value on the quality of an actual solution. After recombination and mutation of genes the GA uses the fitness function to select the best genes for the new population. By making multiple iterations, the GA approaches an solution that is equal or better than the start value. In other words, GAs are usually very fast in finding a good solution, but in general they will not find the best solution.

Intelligent algorithms for computer-aided molecular design have become an important part of lead discovery and optimization, biological target identification, protein and nucleic acid design. Molecular nanotechnology is a term given to the concept of engineered nanosystems (nanoscale machines) operating on the molecular scale. It is especially associated with the concept of a molecular assembler, a machine that can produce a desired structure or device atom-by-atom using the principles of mechano-synthesis [3]. The development of optimization strategies in the molecular nanotechnology (for example, protein folding and atomistic structural determination of macromolecules and clusters) is a subject of great importance.

In modern gen synthesis machines there are the self-organized polymers, where DNA represented by the four nucleotides – genetic alphabite letters. In building molecular phrases, composed by different nucleotides in relevant sequences, molecules are connecting in given order. It is the exact sequence of amino acids in a protein, encoded by the genetic material (DNA), that determines the function of the protein in either its structural or enzymatic role in the cell. The chain of amino acids in the protein represents the structure of the protein [4-6].

The nucleic acids are among the largest macromolecules that occur in cells. There are two types of nucleic acids found in all cellular organisms: Deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). DNA acts as the genetic material of the chromosome, and RNA mainly functions during the synthesis and construction of proteins. Both DNA and RNA are composed of repeating subunits called nucleotides with the complex structure. The ladder-like DNA molecule is formed due to hydrogen bonds between them.

Generically, in optimization problems the overall performance of the system is represented by a multi-variate function called the objective function; for example, the energy of a polyatomic aggregate with the atomic coordinates as variables. Optimal conditions are achieved when the objective function reaches its global extremum, i.e., minimum energy in the above example. However, for systems characterized by a large number of parameters, finding the extrema and, in particular, the global extremum is a vexing problem. The main difficulty is that the global extremum of a real multivariate function is actually a local property, thus requiring an exhaustive search. Furthermore, proving that the global extremum has indeed been found seems to be a rather unattainable task for most systems of interest. Several methodologies aimed at global optimization have been developed. These include gradient and Hessian methods, stochastic and simulated annealing, deterministic techniques, genetic algorithms and other approaches (such as a basinhopping method applied recently in a study of Lennard-Jones clusters which is very similar to a Monte Carlo minimization method).

3. Principles of Genetic Algorithms

Artificial Intelligence have found a widespread use for classification tasks and function approximation in many fields of chemistry and bioinformatics**.** As has been discussed in several recent papers, genetic algorithms (GAs) offer a powerful tool for atomic and molecular cluster structural optimization [7]. It is primarily due to the recent innovations in mating procedures that the GA, long an important method in discrete optimization tasks became relevant to continuous variable optimization as well. An innovative GA approach has been proposed by Deaven and Ho (DH-GA), for structural (energetic) optimization of molecular cluster systems [8]. Rather than trying to create a complicated coding scheme that would allow one to use the traditional mating scheme of cutting and pasting binary strings, they suggested using the direct three-dimensional coordinate space representation of the clusters' structures in conjunction with a particular mating procedure. Graph representation of molecular structures is widely used in nanotechnology researches. Molecular structures are represented by graphs where vertices correspond to atoms, and edges to chemical bonds. This kind of a graph, called now a molecular graph, is the object of study in the theory of ordinary graphs.

Genetic algorithm is a heuristic method that operates on pieces of information like nature does on genes in the course of evolution. Chromosomes are represented by a linear string of letters of an alphabet (in nature nucleotides, in genetic algorithms bits, characters, strings, numbers or other data structures) and they are allowed to *mutate*, *crossover* and *reproduce* [10]. All chromosomes of one generation are evaluated by a *fitness function*.

Depending on the generation replacement mode a subset of parents and offspring enters the next reproduction cycle. After a number of iterations the population consists of chromosomes that are well adapted in terms of the fitness function. Although this setting is reminiscent of a classical function optimisation problem genetic algorithms were originally designed to demonstrate the benefit of genetic crossover in an evolutionary scenario, not for function optimisation. It cannot be proven that the chromosomes of a final generation contain an optimal solution for the objective encoded in the fitness function but it can be shown mathematically that the genetic algorithm optimises the effort of testing and producing new chromosomes if their representation permits development of *building blocks* (also called *schemata*) [9]. In that case, the genetic algorithm is driven by an implicit parallelism and generates significantly more successful progeny than random search. In a number of applications where the search space was too large for other heuristic methods or too complex for analytic treatment genetic algorithms produced favourable results.

The basic outline of a genetic algorithm is as follows:

1. *Initialise a population of chromosomes.* This can be done either randomly or with domain specific background knowledge to start the search with promising seed chromosomes. Where available the latter is always recommended.
   - Chromosomes are represented as a string of bits. This is not a restriction for the type of problem because other data types (numbers, strings, structures) can also be encoded as bit strings.

   - A *fitness function* must be defined that takes as input an individual and returns a number (or a vector) that can be used as a measure for the quality (fitness) of that individual.
   - The application should be formulated in a way that the desired solution to the problem coincides with the most successful individual according to the fitness function.
2. *Evaluate all chromosomes* of the initial population.
3. *Generate new chromosomes.* The reproduction probability for an individual is proportional to its relative fitness within the current generation. Reproduction involves domain specific genetic operators. Operations to produce new chromosomes are:
   - *Mutation*. Substitute one or more bits of an individual randomly by a new value (0 or 1).
   - *Variation*. Change the bits in a way that the number encoded by them is slightly incremented or decremented.
   - *Crossover*. Exchange parts (single bits or strings of bits) of one individual with the corresponding parts of another individual. Originally, only one-point crossover was performed but theoretically one can process up to L - 1 different crossover sites (with L as the length of the individual). For one-point crossover, two chromosomes are aligned and one location on their strings is randomly chosen as the crossover site. Now the parts from the beginning of the chromosomes to the crossover site are exchanged between them. The resulting hybrid chromosomes are taken as the new offspring chromosomes.
4. Select chromosomes for the new parent generation.
   - In the original genetic algorithm simply the complete offspring was selected while all parents were discarded. This is motivated by the biological model and is called total generation replacement.
   - More recent variations of generation replacement compare the original parent chromosomes and the offspring which are then ranked by their fitness values. Only the *n* best chromosomes (*n* is the population size, i.e. the number of chromosomes in one generation) are taken into the next generation. This method is called *elitist generation replacement*. It guarantees that good chromosomes are not lost during a run. With total generation replacement it can happen that good chromosomes „die out“ because they produce only offspring inferior in terms of the fitness function. Another variant is *steady state replacement*. There, two chromosomes are randomly selected from the current population. The genetic operators are applied and the offspring is used to replace the parents in the population. Steady state replacement often converges sooner because on average it requires fewer fitness evaluations than elitist or total generation replacement.
5. Go back to step 2 until either a desired fitness value was reached or until a predefined number of iterations was performed.

Mutation exchanges one single bit. Variation modifies the encoded value by a small increment (or decrement). Crossover (single-point) exchanges a contiguous fragment of an individual. Analogously, more than one crossover point can be selected and only the fragments between those positions exchanged (two-point crossover for two crossover points; uniform crossover for as many crossover sites as positions in the individual).

4. Criteria Optimisation of molecular cluster systems

In this section we will introduce new fitness criteria for the protein folding application with genetic algorithms. The rationale behind is that a more of information about genuine protein conformations should improve the fitness function to guide the genetic algorithm towards native-like conformations. Some properties of protein conformations can be used as additional fitness components whereas others can be incorporated into genetic operators. For such an extended fitness function several incommensurable quantities will have to be combined: energy, preferred torsion angles, secondary structure propensities or distributions of polar and hydrophobic residues [12,13].

This creates the problem of how to combine the different fitness contributions to arrive at the total fitness of a single individual. Simple summation of different components has the disadvantage that components with larger numbers would dominate the fitness function whether or not they are important or of any significance at all for a particular conformation. To cope with this difficulty new fitness criteria could be introduced.

Therefore, for structural optimization of molecular cluster systems, we propose the inverse value of entropy of the entire molecule as new generalized abstract fitness criteria.

5. Heuristic Algorithm

For convenience, let us define each chromosome of population to correspond to the model of molecular cluster systems and a gen in every chromosome to its components. Assuming that a fitness function have been chosen, the genetic algorithm proceeds as follows:

Step 1. Randomly generate initial population of n chromosomes (neural graph).

Step 2. For each chromosome of generations evaluate entropy i.e. the fitness function $f(i)$ with its relative probability

$$P(i) = \frac{f(i)}{\sum_{i=1}^{n} f(i)}$$

Step 3. Sorting of chromosomes population following fitness function.

Step 4. Generate new chromosomes. Probabilistically select a specified number of pairs of chromosomes according to fitness levels. Higher fitness levels increase a chromosome's chance of being selected.

Step 5. Apply the specified genetic operators (*Crossover, Mutation, Inversion)* to these chosen pairs to produce new chromosomes. Create Offspring. Replace them with the newly produced chromosomes.

Step 6. Return to step 4 until either a desired fitness value was reached or until a predefined number of iterations was performed.

6. Conclusion

Summarising mentioned above we are led to the following conclusions:

1. Genetic algorithms proved to be an efficient search tool for representations of proteins. Using a rather small population the genetic algorithm produced *several* chromosomes (i.e. protein conformations) of dissimilar topology but each with highly optimised fitness values.
2. In general, the molecular cluster representation in form the neural graph shows great promise in achieving better similarity search performances than using single structural patterns as targets. The neural model and algorithm increases the efficiency of the original method at the expense of precision and recall of the search results.
3. The major problem lies in the fitness function. Neither mathematical models, empirical, semi-empirical or statistical force fields are yet accurate enough to reliably discriminate native from non-native conformations without additional constraints. Thus, the genetic algorithm with measurement of fuzzy entropy produces (sub-)optimal conformations in a different sense than that of „nativeness“.
4. This model didactically indicates the fundamental connection between information theory and molecular nanotechnogy, and may involve an important contribution in the ongoing quest to develop living and intelligent technology.

References:

1 Smith L. S. Biologically-Inspired Systems. Department of Computing Science and Mathematics, University of Stirling, Scotland, UK, 1999.

2. Holland J. H. Adaptation in Natural and Artificial Systems, 2nd Ed., MIT Press, 1992

3. Frank Michael P. “Nanocomputer Systems Engineering,” in Nanotech 2003: Technical Proceedings of the 2003 Nanotechnology Conference and Trade Show, held Feb. 23-27, 2003, San Francisco, CA, vol. 2, pp. 182-185.

4. Schulz G. E., Schirmer R. H. Principles of Protein Structure, Springer Verlag, 1979.

5. Lesk A. M. Protein Architecture - A Practical Approach, IRL Press, 1991.

6. Branden C., Tooze J. Introduction to Protein Structure, Garland Publishing New York, 1991.

7. Dandekar T., Argos P. Potential of genetic algorithms in protein folding and protein engineering simulations, Protein Engineering, vol 5, no 7, pp. 637-645, 1992.

8. Deaven, D. M.; Ho, K. M. Phys. ReV. Lett. 1995, 75, 288.

9. Goldberg D. E. Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.

10. Meparishvili B., Kervalishvili P., Kekelia V. Some Approaches Modeling of Molecular Machines „Information Technologies in Management“., Proceedings. Tbilisi, 2007. 107-111 pp.

11. Emptoz, H. Nonprobabilistic Entropies and Indetermination Measures in the Setting of Fuzzy Sets Theory. Fuzzy Sets and Systems 5, 307-317, 1981.

12. Le Grand S. M., Merz K. M. The application of the genetic algorithm to the minimization of potential energy functions. The Journal of Global Optimization, vol 3, pp.49-66, 1993.

13. Dandekar T., Argos P. Folding the Main Chain of Small Proteins with the Genetic Algorithm, Journal of Molecular Biology, vol 236, pp. 844-861, 1994.

ხელოვნური ინტელექტი მოლეკულური მოდელირებისათვის

ბადრი მეფარიშვილი, თამარ მეფარიშვილი, თამარ ღვინიანიძე
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

მოლეკულური მოდელირება წარმოადგენს ნანომოწყობილობათა დაპროექტებისათვის მეტად საჭირო ინსტრუმენტს. ხელოვნური ინტელექტის თეორია და მეთოდები ფართოდ გამოიყენება თანამედროვე ქიმიაში, კერძოდ პოლიმერებისა და აგრეთვე სხვადსხვა ნუკლეოტიდების მეშვეობით დნმ-ის ავტომატიზებული კომპოზიციისა და სინთეზის პროცესში. ზოგადად, სასურველი სტრუქტურისა და თვისებების მოლეკულების დასაპროექტებლად ერთერთ მთავარ კრიტერიუმს წარმოადგენს შიგამოლეკულური ენერგიის მინიმიზაცია. სტატიაში განხილულია ახალი მიდგომა, სადაც ნებისმიერი მოლეკულის მოდელი წარმოდგენილია ნეირონული გრაფის სახით, რომლის მყისიერი მდგომარეობა ხასიათდება შიგამოლეკულური და კლასტერთშორისი ბმების არამკაფიო ენტროპიებით. მისი მინიმიზაცია შეადგენს მიზნობრივი ფუნქციის კრიტერიუმს. ოპტიმიზაციის მეთოდად გამოყენებულია გენეტიკური ალგორითმები, რაც მნიშვნელოვნად ამცირებს გამოთვლების დროსა და დანახარჯებს.

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ДЛЯ МОЛЕКУЛЯРНОГО МОДЕЛИРОВАНИЯ

Мепаришвили В., Мепаришвили Т., Гвинианидзе Т.
Грузинский техничесский университет

Резюме

Молекулярное моделирование является очень нужным инструментом в проектировании наноустройств. Теория и методы искусственного интеллекта широко применяются в современной химии, в частности, в процессе синтеза полимеров и автоматизированной композиции ДНК посредством разных нуклеотидов. В общем, при проектировании молекул желаемой струтуры и свойств одним из главных критериев является минимизация внутримолекулярной энергии. В статье рассматривается совершенно новый подход, где любая молекула представлена в виде нейронного графа, состояние которого характеризуется энтропией внутримолекулярных и межкластерных связей, минимизация которой является критерием целевой функции. Методом оптимизации применяются генетические алгоритмы, что в значительной мере сокращает вычислительное время и расходы.