IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 6, Special Issue 3, March 2017

Algorithmic Self Assembly for ANT based Clustering for the Optimization using DNA Hybridization

K. Ganesh Babu

Assistant Professor, Department of Computer Application, Sri Kaliswari College, Sivakasi, India

Abstract: In this paper, algorithmic self assembly for ant based clustering for the optimization using DNA hybridization. DNA molecules can be self assembled to various shapes. The goal is to find the tile set that will self assemble in the target shape. Ant based clustering is data items are randomly scattered into a two dimensional grid. The probability of dropping an item is increased if ants are surrounded with similar data in the neighbourhood. Self assembly is fundamental to both biological processes and nanoscience and its probabilistic nature and local programmability in nature. The input to Ant based clustering is a collection of random generated tile sets. The output is to set of tiles that most closely assembled to the target shape.

Keywords: Data Mining, DNA Hybridization, Self-assembly, Optimization**.**

I. INTRODUCTION

A. Data Mining

Data mining is defined as the non-trivial process of searching and analyzing data in order to find implicit but potentially useful information. Let D = {d_1... d_n} be the dataset to be analyzed. The data mining process is described as the process of finding a subset D‟ of D and hypotheses H_U(D‟,C) about D‟ that a user U considers useful in an application context C. D‟ have fewer data elements than D, but it also have a lower dimensionality (m‟). In databases the data is partitioned into relations or object classes. D is considered as a union of relations R_1... R_k each has its own dimensionality (m_1... m_k)[1].

B. Bioinformatics

Bioinformatics is the Science of integrating, managing, mining and interpreting information from biological datasets at genomic, metabalomic, proteomics, phylogenetic and cellular or whole organism levels.

According to (National Institute of Health) NIH organization, the Bioinformatics and Computational Biology have been defined as “Bioinformatics is research and development or application of computational tools and approaches for expanding the use of biological, medical, health data including those to acquire store, organize, active, analyze or visualize such data” [2].

Genomics

DNA (Deoxyribonucleic Acid) is a molecule encoding the genetic instructions used in the development and functioning of all known living organisms many viruses. DNA is one of the three major macromolecules that are essential for all known forms of life. DNA–DNA hybridization generally refers to a molecular biology technique that measures the degree of genetic similarity between pools of DNA sequences. It is usually used to determine the genetic distance between two species.

When several species are compared that way, the similarity values allow the species to be arranged in a phylogenetic tree. It is therefore one possible approach to carrying out molecular systematic. DNA-DNA hybridization is considered as a gold-standard of distinguishing species. Genetic information is encoded as a sequence of nucleotides (Guanine, Adenine, Thymine, and Cytosine) recorded using the letters G, A, T, and C.

Most DNA molecules are double-stranded helices, consisting of two long polymers of simple units called nucleotides with the nucleobases (G, A, T, C) attached to the sugars. DNA is well-suited for biological information storage, since the DNA backbone is resistant to cleavage and the double-stranded structure provides the molecule with a built-in duplicate of the encoded information [3].

IJARCCE

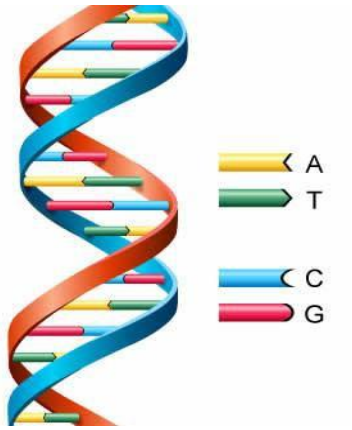
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 6, Special Issue 3, March 2017



(i) Nucleic Acids

Fig. 1 Structure of DNA

Nucleic acids are large biological molecules essential for all known forms of life. They include DNA. Together with proteins, nucleic acids are the most important biological macromolecules; each is found in abundance in all living things, where they function in encoding, transmitting and expressing genetic information.The nucleic acids Deoxyribonucleic acid DNA are polymers of nucleotides, arranged in a specific sequence. To form macromolecular polymers, nucleotides are joined between the 3‟ and 5‟ carbon atoms in their sugar moiety by a phosphodiester bond, giving rise to a nucleic acid with a sugar–phosphate „backbone‟ to which is attached a series of bases in a specific order. Hydrogen bonding between pairs of bases can occur, leading to the formation of double-stranded polymers if the sequences are complementary[4].

(ii) Base pairs

Base pairs are the building blocks of the DNA double helix, and contribute to the folded structure of both DNA. Dictated by specific hydrogen bonding patterns, Watson-Crick base pairs (Guanine-Cytosine and Adenine-Thymine) allow the DNA helix to maintain a regular helical structure that is independent of its nucleotide sequence. The complementary nature of this based-paired structure provides a backup copy of all genetic information encoded within double-stranded DNA. Fig. 2 shows the pairs of ATGC. The regular structure and data redundancy provided by the DNA helix make DNA an optimal molecule for the storage of genetic information, while base-pairing between DNA and incoming nucleotides provide the mechanism through which DNA polymerase replicates DNA transcribes. Many DNA-binding proteins can recognize specific base pairing patterns that identify particular regulatory regions of genes.

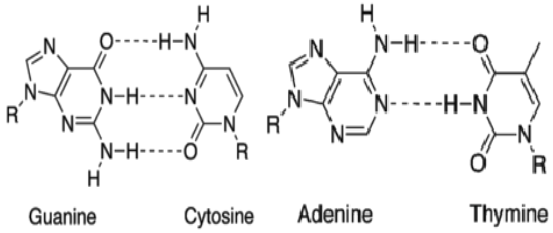


Fig. 2 Base pairs of ATGC

(iii) Swarm Intelligence (SI)

SI is a recent and emerging paradigm in bio inspired computing for implementing adaptive systems. SI encompasses the implementation of collective intelligence of groups of simple agents that are based on the behavior of real world insect swarms, as a problem solving tool. The word swarm comes from the irregular movements of the particles in the problem space. SI has been developed alongside with EAs. The SI algorithms being inspired by the collective behavior of animals, exhibit decentralized, self-organized patterns in the foraging process.

(iv) DNA Motifs

Complex designs are often created using a relatively small set of common building blocks called motifs. DNA self-assembly can exploit this same design principle to hierarchically create more sophisticated a periodic structures. There are many possible DNA motifs and the focus here is on only a few in the context of the target nanostructure. Motifs

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

include junctions that enable three or more double stranded helices of DNA to interact and thus form specific structures (e.g., a triangle, a corner, and so on). Another important motif is a single strand of DNA protruding from a double stranded helix called a sticky-end.Two motifs with complementary sequences on their sticky-ends will bind to form a composite motif. Composite motifs may also have embedded sticky-end motifs and thus can also bind with other composite motifs to form another, larger, composite motif. This results in a hierarchical structure for motifs.

(iv) Self assembly

Self-assembly is a process by which components may form complex structures autonomously. Due to the natural properties of DNA structures, self assembly will occur when DNA molecules are allowed to inter act. DNA molecules that can assemble into complex structures can be modeled by a tile self-assembly system. In the Wang model of tile self-assembly a tile can represent a DNA structure with four one-stranded "sticky ends". These ends are composed of a finite combination of short nucleotide sequences that will naturally bind with other nucleotide sequences. Each of the tile"s four sides is given a value or "color" that represents the binding properties of the tile.

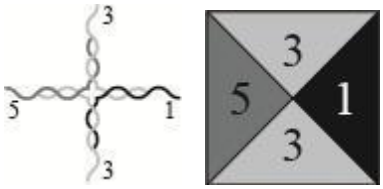


Fig.3 Left: DNA tile as made up of DNA helices. Notice the single stranded "sticky ends". Right: Representational tile model of this structure.

(v) DNA Self-assembly

Self-assembly is the process in which disordered objects come together without any external activity to form a complex structure. First demonstrated by biological and inorganic physical system. [5].

Algorithmic self-assembly

DNA tiles can be designed to contain multiple sticky ends with sequences chosen so that they act as Wang tiles. A DX array has been demonstrated whose assembly encodes an XOR operation. This shows that computation can be incorporated into the assembly of DNA arrays, increasing its scope beyond simple periodic arrays[6].While other materials can be used, most models use a fluorescence-based substrate because it is very easy to detect, even at the single molecule limit. The amount of fluorescence can then be measured to tell whether or not a reaction took place. The DNAzyme that changes is then "used," and cannot initiate any more reactions. Because of this, these reactions take place in a device such as a continuous stirred-tank reactor, where old product is removed and new molecules added. A design called a stem loop, consisting of a single strand of DNA which has a loop at an end are a dynamic structure that opens and closes when a piece of DNA bonds to the loop part[7].

a. Enzymes

Enzyme based DNA computers are usually of the form of a simple Turing machine; there is analogous hardware, in the form of an enzyme, and software, in the form of DNA.

b. Toehold exchange

DNA computers that, (DNA computing is a form of computing which uses DNA, biochemistry and molecular biology, instead of the traditional silicon-based computer technologies) have also been constructed using the concept of toehold exchange. In this system, an input DNA strand binds to a sticky end, or toehold, on another DNA molecule, which allows it to displace another strand segment from the molecules.

(viii) DNA Nanotechnology

DNA molecules today have been employed as chemical building blocks to build nanometer-sized structures. DNA has several structural considerations that make it well-suited for making nanostructures its complementary strands, an antiparallel double-helical backbone that is largely regular regardless of sequence, The recognition of DNA strands by their complements can be used for more than the formation of a simple double helix.

Biologists recognized in the early 1970s that single-stranded overhangs (sticky ends) could be used to direct intermolecular association of different DNA molecules. The intermolecular structures formed are predictable, due to the base pairing rules. Such sticky ends (usually four to nine bases long in DNA nanotechnology) are, thus, convenient in constructing intermolecular structures, especially because complementary program.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 6, Special Issue 3, March 2017

Such molecules are found in biological systems as the ephemeral four-arm Holliday junction, an intermediate in genetic recombination. The basic notion of structural DNA nanotechnology is to combine stable branched DNA molecules with sticky-ended cohesion or other forms of cohesion that are structurally well-defined[8].

II. PREVIOUS WORKS

U. Boryczka, 2008 presented a new ant clustering algorithm called ACA for data clustering in a knowledge discovery context. Monmarche et. al., 1999 combined the stochastic and exploratory principles of clustering ants with the deterministic and heuristic of the popular k-means algorithm in order to improve the convergence of the ant-based clustering algorithm.
Labroche et. al., 2002 proposed a clustering algorithm, called ANTCLUST, based on a modeling of the chemical recognition system of ants. N.
X. Xu and Y. Chen, 2004 presented an artificial ants sleeping model (ASM) and an adaptive artificial ants clustering algorithm (A4C) to resolve the clustering problem in data mining by simulating the behaviors of gregarious ant colonies. P.S. Shelokar et. al., described an ant colony optimization methodology for optimally clustering „N" objects into „K" clusters.
B. Gillner, 2004 investigated the performance of ACLUSTER and ATTA under the measures and datasets proposed by Handl et. al., 2007 Based on performance results of both algorithms gathered from numerous runs. X. Huang described an improved version, called chaotic ant clustering algorithm (CACAS), adopting an important strategy of using chaotic perturbation to improve individual quality.
Y. Wang et. al., 2007 proposed an advanced clustering method called ant colony ISODATA algorithm (ACIA) in real time computer simulation. Z. Sadeghi, et.al., presented a new strategy for clustering using artificial ants in which groups of ants try to do clustering by inserting and removing operations.
D. A. Ingaramo, et.al., 2008 proposed a new version of the Ant-Tree algorithm called Adaptive Ant Tree (AAT), an approach inspired on the self-assembling behavior observed in some species of real ants. Vizine et. al., 2009 presented a new algorithm called ACA which comes with a cooling scheme for picking probabilities. The local error function remains unchanged. This algorithm improves convergence.

TABLE 1 PREVIOUS WORKS RELATED TO ANT BASED CLUSTERING

Authors/ Year	Algorithms	Description
U. Boryczka, 2008	ACA	For data clustering in a knowledge discovery context.
Monmarche et. al., 1999	Heuristic K-Means algorithm	Combined the stochastic and exploratory principles of clustering ants to improve the convergence of the ant-based clustering algorithm.
Labroche et. al., 2002	ANTCLUST	Based on a modeling of the chemical recognition system of ants.
X. Xu and Y. Chen, 2004	ASM	An adaptive artificial ants clustering algorithm.
P.S. Shelokar et. al., 2004	A4C	To resolve the clustering problem in data mining by simulating the behaviors of gregarious ant colonies.
B. Gillner, 2007	ACLUSTER and ATTA	Based on performance results of both algorithms gathered from numerous runs.
X. Huang, 2007	CACAS	Adopting an important strategy of using chaotic perturbation to improve individual quality.
Y. Wang et. al., 2007	ACIA	An advanced clustering method called ant colony ISODATA algorithm (ACIA) in real time computer simulation.
D. A. Ingaramo, et. al., 2007	AAT	An approach inspired on the self- assembling behavior observed in some species of real ants.

III. ALGORITHMS USED

A. Ant Clustering algorithm

The ant clustering algorithms are mainly based on versions proposed by Deneu-bourg, Lumer and Faieta. A number of slight modifications have been introduced that improve the quality of the clustering and, in particular, the spatial separation between clusters on the grid. Recently Handl and Meyer (2002) extended Lumer and Faieta"s algorithm and proposed an application to the classification of Web documents. The model proposed by Handl and Meyer has inspired

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

us to use this idea to classical cluster analysis. The basic idea is to pick up or drop a data item on the grid.

Each ant has a permission to exploit its memory according these rules: if an ant situated at grid cell p, and carrying a data item i, it uses its memory to proceed to all remembered positions, one after the other. Each of them is evaluated using the neighbourhood function f*(i) for finding a dropping site for the currently carried data item i.
Algorithm 1:

1. Initialization Phase
2. Randomly scatter o_i object on the grid file
3. for each agent a_j do
4. random select object (o_i)
5. pick up object o_i
6. place agent a_j at randomly selected empty grid location
7. end for
8. for t = 1 to t_{max} do
9. random select agent (a_j)
10. Compute $f^*(o_i)$ and $p^*drop(o_i)$
11. if drop = True then
12. while pick = False do
13. i = random select object o
14. Compute $f^*(o_i)$ and $p^*pick(o_i)$
15. Pick up object o_i
16. end while
17. end if
18. end for
19. end

Fig. 4 The original Ant Clustering algorithm

B. Particle Swarm Optimization
Particle Swarm Optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search -space according to simple mathematical formulae over the particle's position and velocity.

Algorithm 2:
1. For each particle{
2. REPEAT initialize particle until it satisfies all the constraints
3. }Do
4. {For each particle {
5. Calculate fitness value
6. If the fitness value is better than the best fitness value (pBest)
7. In history AND the particle is in the feasible space
8. set current value as the new pBest value among all the neighbors as the nBest Calculate the particle velocity
9. Update particle position}
10. } While maximum iterations or minimum criteria are attained.
11. End While
12. End For
13. End.

Fig. 5 The original PSO algorithm

IV. PROPOSED ALGORITHM

In this work, the Modified Ant Clustering (MAC) is proposed to optimize the tile set generation for DNA Nanotechnology. MAC is the optimization algorithm to optimize the tile set for self assembled the DNA nanostrucutres. The Ant Clustering is a optimization algorithm that deals with the classical clustering analysis. The

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

basic idea is to pick up or drop a nucleotide sequence on grid.The PSO is a simple computational method to optimize the tile sets for simulating the tile assemblies based on the x axis, y axis which we given the directional factors.

A. Modified Ant Clustering algorithm

The MAC begins by generating a set of tiles based on the number of unique nucleotide sequences to be used. This set contains exactly one copy of every possible tile type. From this random sets of tiles, or individuals, are generated. An individual consists of a given number of unique tile types with any number of copies for each types. Wang tile system simulation is run to determine the shape to which the tile set will assemble. In this simulation allows each tile to move randomly in four cardinal directions on a 50x50 unit grid. If a tile comes into contact with another tile or a group of tiles, the tile may bind and stop moving. Tiles are allowed to bind with each other based on an arbitrarily generated, symmetric binding matrix of existing single-stranded nucleotide sequences that bond at the given strengths. The binding matrix is a numeric representation of the physical binding properties of given nucleotide sequences.

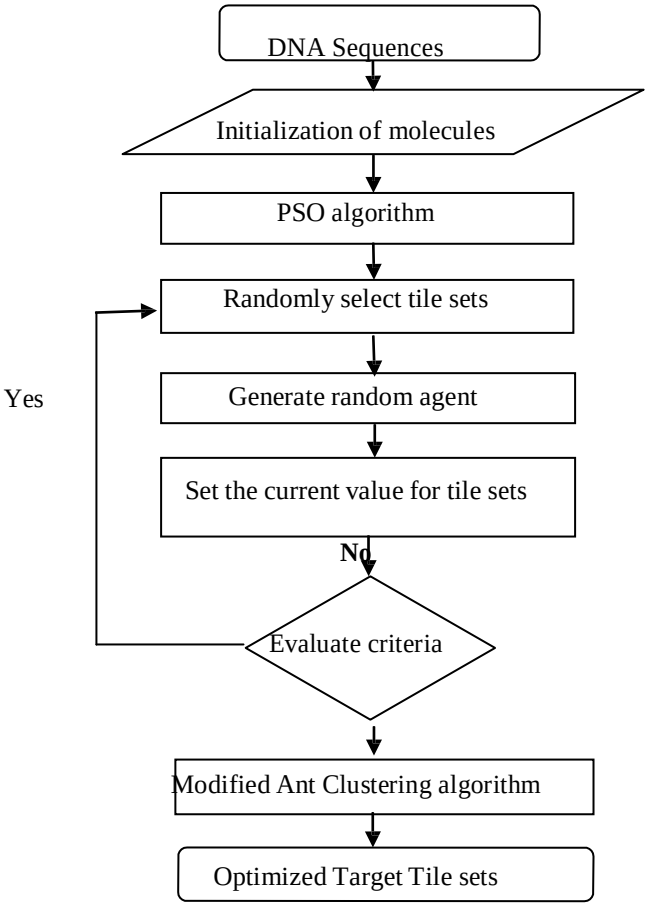


Fig.5 Flowchart for Modified Ant clustering algorithm

Algorithm:

1. Initialization Phase
2. Randomly scatter o_i object on the grid file
3. for each agent a_j do
4. random select tile sets (o_i)
5. pick up object o_i
6. place agent a_j at randomly selected empty grid location
7. End for
8. for t = 1 to t_{max} do
9. random select agent (a_j)
10. If { Tile sets generated
11. It reaches upto 15 dimensions}

12. Else
13. Tile sets not generated
14. move agent a_j to new location
15. set current value as the new pBest value among all the neighbors as the nBest
16. Calculate the particle velocity
17. Update particle position}
18. } While maximum iterations or minimum criteria are attained.
19. End for
20. End if
21. End

Fig. 6 Modified Ant Clustering algorithm

V. CONCLUSION

The simulation allows a user to create, load, and save tile assembly systems, either as a unit or as separate components. Simulation can be done one step at a time or in a fast-forward mode. Simulation steps are catches, so they can also be run in reverse. The simulation is optimized to maximize the speed of assembly while handling very large tile sets. To provide for maximum simulation speed, the simulator can be configured to redraw the display of the assembly only at user-specified intervals. In DNA nanotechnology, this has been done by using ”template” molecules (programmable tiles) that interact with DNA single strands, The pairing of the single stranded DNA present in the solution to a single strand subsequence of the tile induces this latter to change its conformation. Because of these conformational changes, tiles get a different status during the assembly, with the effect that one is able to control the dynamics of the algorithm and the direction of the assembly.

REFERENCES

[1] M. H. Dunham, Data Mining: Introductory and Advanced Topics, Pearson Education, Third Edition, New Delhi, India, ISBN 81-7758-880-X, 2012.
[2] Yi-Ping P. Chen (Ed.), Bioinformatics technologies, Springer International edition, Hiedelberg, ISBN 3-540-20613-2, 2008.
[3] http://en.ncbi.org//DNA,_RNA_and_proteins:_The_three_essential_macromolecules_of_life#DNA.2C_RNA_and_proteins.
[4] Seung-Hyun Lee and Chengde Mao, DNA Nanotechnology, Techniques essay, University, USA, 2004.
[5] J. Goewaracki et. al., Optimization of tile sets for DNA self assembly, JACS, pp. 31-39, 2011.
[6] Scott Summers, “Universality of Algorithmic Self Assembly“ Thesis, California 2010.
[7] David Doty, “Theory of Algorithmic Self Assembly” Review, 2012.
[8] Mathew J. Patitz, “An Introduction to Tile-Based Self-Assembly and a Survey of Recent Results”, Dept of CS, Arksana University, USA, 2012.
[9] http://wiki.answers.com/Q/What_macromolecules_make_up_DNA
[10] N. Jonoksa, N. C. Seeman, “ Computing by Molecular Self assembly” Interface Focus, 2012.
[11] http://www.algoritmicselfassembly/wiki/html
[12] Harish Chandran, “ Thesis on DNA Self Assembly”, Dept of CS, Duke University, USA, 2012.
[13] Stochastic simulations, Application to molecular networks May 10, 2007.
[14] M. R. Lakin et. al., “Visual DSD”, Bioinformatics Application note, Vol. 22(11), pp. 3211 – 3214, 2011.
[15] http://lepton.research.microsoft.com/webdnaWireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification , IEEE Std. 802.11, 1997.