An overview of metaheuristics and their potential impact on clustering molecular structures

Wen Kang Lu
LXXWEN005
University of Cape Town

ABSTRACT

Molecular dynamics plays an important role in determining the shape of molecules. Clustering is often used to extract the main molecular conformations from simulation trajectories. Although traditional clustering algorithms, such as K-means, are commonly used to partition the data, these methods often get stuck on local optima. Often times, *a priori* knowledge such as cluster count must also be known, which can be prohibitive when dealing with newfound trajectories. Many metaheuristics do not suffer from these limitations. Here we review the use of genetic algorithms, simulated annealing, and artificial immune systems for clustering. We find that metaheuristics that simultaneously optimise multiple objective functions to be more promising for the purposes of clustering trajectories due to its ability to cater for properties unique to different data sets. Interestingly, we also find that the classification of 3D models through the use of a clonal selection-based algorithm draws many parallels with the clustering of molecular dynamics trajectories.

CCS CONCEPTS

• **Theory of computation** → **Unsupervised learning and clustering**; • **Applied computing** → *Chemistry*; • **Mathematics of computing** → *Combinatorial optimization.*

KEYWORDS

molecular dynamics, clustering, metaheuristics, objective functions

1 INTRODUCTION

Molecular dynamics simulates the behaviour of molecules which are allowed to interact with one another over a fixed period of time; outputting what is known as the "trajectory" of the behaviour observed. This trajectory is a record of the movement of every atom during the simulation, with respective change in positions over time calculated by solving Newton's equations of motion [29]. Since the effects of molecular interactions occur so rapidly, positions are often updated at the femtosecond timescale [34]. Therefore, trajectories containing even just a few nanoseconds worth of snapshots amount to millions of locational data to observe.

Molecular dynamics simulations play an important role for researchers interested in the effects molecular conformations can have in their applications, some of which include drug-receptor interaction and protein folding [40]. Consequently, it is important that effective methods exist to gather insight from trajectories as efficiently as possible. A common way to do so is through the use of clustering algorithms that can partition the trajectory into a set of dominant conformations that are deemed structurally unique from one other.

Using a set of unlabeled data, clustering is a form of unsupervised learning whereby items deemed similar are grouped together, with items between groups being seen as dissimilar. Many popular algorithms are used for clustering in general, let alone just for molecular dynamics. Although definitions vary, these algorithms can be broadly classified into three categories: hierarchical, partitional, and overlapping clustering [22, 36].

Hierarchical clustering can take on a divisive (top-down) approach, or an agglomerative (bottom-up) one [41]. The agglomerative version starts by initially treating each item as its own cluster before iteratively merging these clusters together on the basis of proximity. Conversely, the divisive method puts the whole set of items into one cluster before iteratively splitting them. Both methods generate a tree structure. Although efficient, these algorithms tend to be less performant than others and are sensitive to outliers within the data [40].

Partitional clustering treats the task of grouping as an optimisation problem by minimising or maximising the value of a given fitness measure. The most common example of such clustering is the K-means algorithm that attempts to minimise the intracluster distance of each point to its respective centroid [36]. Like hierarchical clustering at any given iteration, each item belongs in one and only one cluster.

Overlapping clustering, unlike hierarchical and partitional, do not produce clusters where each item belongs exclusively to just one of them. Instead, soft clusters (where items belong to one or more groups) are produced, or fuzzy ones (where each item belongs to all clusters to some level of degree) [22]. A common algorithm used for this type of clustering is the fuzzy C-means.

Shao et al. [40] made comparisons between eleven algorithms used to cluster trajectory data, concluding that no one algorithm completely dominates the others in every test, and that the user-determined values such as the cluster count or cluster diameter (depending on the algorithm), as well as the data itself can significantly alter performance. Since the choice of data to cluster cannot be changed without changing the context of the experiment itself, this requires the algorithm-specific parameters made by the user to be a perfect match for the data. This *a priori* knowledge is often not present, and so it becomes prohibitive when exploring new trajectories [34].

Metaheuristics are algorithms that are stochastic in nature, relying on the use of random initial solutions that are then improved upon iteratively through local and global search [36]. By balancing exploration of the search space and exploitation of the highest performing solutions, a metaheuristic's goal is to converge on a global optimum. Due to its use in optimisation tasks, metaheuristics used for clustering problems can be considered to be a form of partitional clustering. Similar to the K-means method, many metaheuristics

try to minimise the intracluster distances. This is not always the case, however, as we shall see with the algorithms discussed.

Metaheuristics are often nature-inspired, with many algorithms deriving its process from naturally occurring phenomena. This review explores algorithms from three different class of metaheuristics to ensure appropriate breadth regarding fundamental differences, while also ensuring depth as we analyse examples of their implementations and applications. The algorithms in question are the genetic algorithm (GA), simulated annealing (SA), and artificial immune system (AIS). These were chosen due to their popularity in recent studies, with improvements made that may particularly benefit the clustering of trajectories.

The rest of the paper is organised as follows: we shall first detail the freely available real-life data sets often used when testing metaheuristics in Section 2, followed by reviews of genetic algorithms, simulated annealing, and artificial immune systems in Sections 3, 4, and 5, respectively. Each section entails a brief exploration of an algorithms background followed by a few applications, noting any interesting and unique approaches. A discussion will then follow in Section 6, relating all the methods mentioned with molecular dynamics, and finally drawing conclusions in Section 7.

2 DATA SETS

The UCI machine learning repository is a freely available online collection of databases that contain many real-life data sets used specifically for the testing of machine learning algorithms [16]. Many of the algorithms discussed in this review utilise data sets from this repository. Therefore, we shall highlight the characteristics of each data set used.

Connectionist bench contains 208 instances of sonar signals bouncing off of metal cylinders or rocks at various angles. Each instance consists of a set of 60 real numbers. Consequently, an optimal 60 dimensional partition would contain a cluster for metal cylinder and rock.

Glass is a data set containing 6 types of glass differentiated by their oxide contents, i.e, Na, Fe, K, etc. There are 10 attributes in total, leading to an optimal 10 dimensional partition with 6 clusters.

Ionosphere consists of 351 radar returns from the ionosphere. These instances are described by 2 integer attributes per pulse number, of which 17 exist. This would lead to a 34 dimensional partition containing 2 clusters: both "good" and "bad" radar returns.

Iris is a data set containing 3 classes of iris flowers with 50 instances each. There are 4 attributes: sepal length in cm, sepal width in cm, petal length in cm, and petal width in cm. This infers an optimal partition of 3 clusters containing 150 data points in a 4 dimensional space. Along with Wisconsin Breast Cancer, these 2 data sets are most the commonly used among the algorithms reviewed.

Lung cancer describes 32 instances of pathological lung cancers that fall under 3 types, and are characterised by 57 integer-based attributes. The low amount of data points in this data set could pose a unique challenge for algorithms. An optimal partition would contain 3 clusters in a 57 dimensional space.

Wisconsin Breast Cancer is a data set that contains 699 data points albeit with a few missing values. A particular data point is considered to be benign or malignant based on 10 integer attributes. An optimal 10 dimensional partition would contain 2 clusters.

Zoo categorises 101 animals into 7 classes based on 17 attributes that are either boolean (categorical) or integer-based. Examples include hair, feathers, eggs. An optimal partition would have 7 clusters in a 17 dimensional space.

3 GENETIC ALGORITHM

The GA is considered to be the first nature-inspired metaheuristic, and it was developed by Holland et al. in 1975 [21]. The algorithm was then used to guide the first GA-based clustering algorithm proposed by Bezdek et al. [8]

By following the Darwinian evolutionary theory of "survival of the fittest", the algorithm starts by generating an initial pool of random solutions called a population, where each solution called a gene or individual is encoded into what is known as a chromosome. Researchers have utilised varying forms of encoding that fall within three main groups: binary, integer, and real.

Having determined the encoding, the first step of any generation is to select parent solutions to be used for reproduction. The expectation is that solutions of higher quality are more likely to produce offspring solutions that yield an even better fitness function (the measure that is to be maximised or minimised by the algorithm). There exist many different ways to select these parents, where the important factor to consider is the trade-off between exploration and exploitation. Should only the highest quality solutions be selected each generation without fail, the algorithm will converge on an optimum very quickly but at the cost of it likely being a local one. By selecting too many weak candidates, the algorithm will converge very poorly and likely lead to a sub-optimal solution. Once the parent solutions have been identified, two of them are probabilistically selected at a time to reproduce in a crossover step. The goal of crossover is to create two new solutions with chromosomes that bear traits from both of their parents. The final step of each generation is the mutation step which occurs with a very low probability. The use of mutations is to assist in the escaping of local optima in favour of global ones by adding this additional variability to the population.

3.1 Applications

Agustín-Blas et al. [2] developed a genetic algorithm (GGA) with a variable length chromosome and modified crossover and mutation operators that follow Falkenauer's grouping-based methodologies [17]. The probabilities of crossover and mutation are high in the beginning of the algorithm to promote exploration and lowers as generations pass to converge on a global optimum. The algorithm also makes use of the DB Index [11] whose value is not monotonic for varying number of clusters, allowing it to determine the optimal amount of clusters to be used. The algorithm was tested against K-means and DBSCAN using artificial data consisting of two-dimensional spherical, structured and unbalanced data with 300, 400, and 200 data points respectively. The authors also tested with the UCI data sets Iris and Wine. Using the Rand Index [37], the GA was found to outperform both K-means and DBSCAN in all tests. A summary of the algorithm, along with the others discussed in this review, can be found in Table 1.

An automatic GA called AGCUK using a unique mutation operator was designed by Liu et al. [30]. Two types of solution are present in any given generation: the best individual with the correct number of clusters (determined by the DB Index) and the rest of the population. Non-best solutions are subject to a division-absorption mutation whereby individuals are randomly partitioned or merged. A noising method is also used to explore the search space further, probabilistically accepting poor individuals and rejecting good ones to escape local optima. The algorithm was tested using a subset of the 100 synthetic two-dimensional data proposed by Lin et al. [28]. Comparisons were made against four other genetic algorithms provided by Bandyopadhyay and Maulik [5, 6], Lai [26], and Lin et al. [28]. It was found that the AGCUK method outperforms the other algorithms in most cases, exhibiting lower misclassification rates using the optimal amount of clusters. The Breast Cancer data set was also used to test the algorithms. Even though the correct number of clusters were found by AGUCK and Bandyopadhyay's method, no algorithm correctly partitioned the data. Liu et al. specifically point out that the use of only one validity index makes it difficult when clustering different data sets. It was also noted that although the time complexity of AGCUK was comparable to the other three, the noising method may stochastically cause the algorithm to converge very late due to its tolerance for weaker individuals.

He and Tan [20] tested a two-stage genetic clustering algorithm (TGCA) that can optimise the number of clusters and find the best partition of a data set. The algorithm continuously looks for both by using the CH Index [9] based on the internal cluster coherence and the external cluster separation. The selection and mutation probabilities vary with the number of clusters, with stage one searching for the number of clusters and stage two locally searching for the best cluster center. Parent selection made use of roulette wheel selection where solutions are chosen with a probability proportional to their objective function. Using artificial data and the UCI data sets Iris, Glass, Breast, Connectionist Bench, ionosphere, Zoo, and Lung Cancer, the algorithm was compared to a hierarchical agglomerative K-means hybrid, an automatic spectral algorithm [42], and standard genetic K-means algorithm. Although the two-stage GA performed better across the board, its search ability was reduced when the dimensionality of the data was high. Specifically, the Connectionist Bench, Ionosphere, and Lung Cancer data sets (with 60, 34, and 57 dimensions respectively) imposed a challenge to the two-stage GA.

4 SIMULATED ANNEALING

Simulated annealing is a physical-based algorithm developed by Kirkpatrick et al. in 1983 [23]. By mimicking the process of annealing, whereby solids are heated to a sufficiently high temperature to then be cooled down, SA algorithms find their solutions using the same paradigm. The algorithm optimises a single solution which contrasts the other population-based algorithms discussed in this review.

Having initialised a random starting solution, the system iteratively compares this solution to a neighbouring one achieved through mutation. A neighbouring solution yielding a better fitness function will always replace the current solution. When faced with a poor individual, however, it is selected based on a probability function that takes into account how much worse the new solution is, as well as the current temperature of the system. A high temperature increases the probability of a worse solution being chosen for the sake of exploring the search space and escaping local optima, which is especially important given the single solution approach of traditional SA. After each iteration, the temperature of the system decreases by a factor often called the cooling rate. Therefore, as the temperature lowers, the system becomes more strict regarding whether a worse solution may replace the existing one. A SA-based clustering method was first introduced by Selim and Alsutan in 1991 [39].

4.1 Applications

Bandyopadhyay [3] proposed a single-objective, fuzzy clustering algorithm (SA-RJMCMC) that made use of the homogenous Reversible Jump Markov Chain Monte Carlo (RJMCMC) kernel [18]. In short, RJMCMC is used to determine the optimal number of clusters during mutation which minimises the cluster validity index. In this case, Bandyopadhyay opted to use the fuzzy XB Index [43]. The key advantages SA-RJMCMC has over the traditional fuzzy C-means algorithm is its ability to determine the right number of clusters to use for a given data set, as well as its ability to escape local optima. The algorithm was compared to the standard fuzzy C-means method using 3 sets of artificial data and 2 real-life sets Breast Cancer and Kala Azar [33]. Data points ranged from 68 to 900, and dimensions ranged from 2 to 9. Although results were similar between the 2 methods, the traditional fuzzy C-means approach required multiple runs as it got stuck at local optima, further highlighting this weakness. The algorithm was also used to cluster satellite images and was compared to a GA-based fuzzy algorithm. SA-RJMCMC was found to provide more optimal number of clusters while also differentiating the landscape better.

With data sets becoming more and more complex, single-objective algorithms may no longer be sufficient when determining the best partition for the problem at hand. The previous algorithms discussed only optimise one function like the DB or XB index. Multi-objective optimisation tries to simultaneously optimise many objective functions or validity indices, acknowledging the fact that one function may not be able to capture all properties a data set contains. By doing so, multiple global optima may now be present, with the group of such solutions called the Pareto-optimal set. A solution is said to be Pareto-optimal if there exists no change that would improve one index without another index deteriorating. Pareto-optimal solutions are also non-dominated solutions. A solution is said to be dominated by another should the latter solution yield better objective functions across the board.

Bandyopadhyay et al. [7] proposed a multi-objective simulated annealing-based clustering algorithm called AMOSA. Using the concept of dominance, non-dominated solutions were stored in an archive. This archive eventually becomes the Pareto-optimal set where each individual is a global optimum, varying in the number of clusters and layout of each partition. A key feature of this algorithm not present in most multi-objective evolutionary algorithms is its non-zero probability of accepting a dominated solution, allowing for better exploration of the search space. The algorithm has both a binary and real encoded variation, with both being tested

against two multi-objective evolutionary algorithms NGSA-II [15] and PAES [24]. The data used were a collection of test problems derived from other studies [13, 14]. PAES was found to perform poorly in general with AMOSA performing better than NSGA-II in most cases. Most notably, AMOSA provided more distinct solutions in its set of Pareto-optimal solutions, and it was found to perform significantly better when using many objective functions.

Another multi-objective SA-based clustering method was proposed by Acharya et al. to specifically cluster tissue samples for cancer diagnosis [1]. This algorithm was specifically used to cluster tissue samples for cancer diagnosis. AMOSA was used for its search capabilities, optimising the XB Index, FCM Index and PBM Index [4] simultaneously. Interestingly, the algorithm has a choice of 3 mutation operators when generating a neighbouring solution. Mutation 1 changes the cluster center by some small value, and mutations 2 and 3 decrease and increase the size of the string encoding by one respectively, i.e., the current partition shifts its current cluster configuration or changes its number of clusters by one. The algorithm was tested using three (non-UCI) real-life cancer data sets: Brain Tumor (42 samples, 5 dimensions), Adult Malignancy (190 samples, 14 dimensions), and Small Round Blood Cell Tumors (63 samples, 4 dimensions). The algorithm was compared to many others, most notably K-means, hierarchical average linkage, and self organising maps. Using the Adjusted Rand Index and Classification Accuracy as evaluation metrics, the AMOSA-based algorithm performed the best across all three data sets.

5 ARTIFICIAL IMMUNE SYSTEM

The artificial immune system contains represents a category of algorithms based on different immunology theories. Based on the survey on nature-inspired metaheuristics done by Nanda and Panda [36], the clonal selection algorithm (CSA) approach proposed by de Castro and Zuben [12] seems to be the most popular for the purposes of machine learning and optimisation. As such, we will be exploring applications of AIS algorithms that specifically make use of the CSA approach.

The clonal selection algorithm is based on a theory developed by Burnet in the mid 1900's. The algorithm makes use of the principle where only antibodies (solutions) that recognise the present antigen (problems) are allowed to proliferate in the system. An initial population of antibodies is generated, and a calculation is made with each antibody to check their affinity with the antigen. This affinity, in the form of an objective function(s) in clustering problems, tells us how fit a given antibody is for the problem at hand. The N most fit individuals are cloned a number of times which is proportional to their respective affinity - an antibody with a high affinity yields more clones. The clones are then subjected to what is known as affinity maturation. This is the process of mutating each clone at a rate inversely proportional to its affinity, i.e., a clone with a low affinity is mutated more often than a clone with high affinity. The clone with the best affinity after the mutations is then stored in a memory pool. The usage of the memory pool, as we shall see, is different across studies. A portion of the weakest antibodies in the initial population is then replaced by new randomly generated antibodies in a step called receptor editing, with the goal of diversifying the population. This process repeats until the termination criterion is fulfilled.

5.1 Applications

Kuo et al. [25] proposed a CSA-based algorithm (AISK) that integrated the K-means algorithm to perform an additional local search of each clone after the affinity maturation step. The antibodies were real-encoded and centroid-based, allowing the usage of Gaussian mutation which slightly alters each cluster centroid by a Gaussian random variable. The proposed algorithm is single-objective, where the affinity is based on minimising the intra-cluster distance. The memory cells are simply part of the population with no special characteristics other than not being able to be replaced via receptor editing. Like the traditional CSA, mutations occur at a rate inversely proportional to the fitness level of each solution. The algorithm was applied to 4 UCI benchmark data sets: Iris, Wine, Glass, and Breast Cancer. Dimensions ranged from 4 to 13, and data points 150 to 683. When compared to particle swarm optimisation (PSO), a swarm intelligence-based metaheuristic, the AISK performed better with higher accuracy rates across all data sets. The convergence rate was also found to be faster with AISK, highlighting its strong search capabilities. Although AISK requires the number of clusters to be known *a priori*, Kuo et al. also provided a two-stage variant that uses ART2 [10] in the first stage which finds the optimal number of clusters, performing AISK in the second stage. This method was used for customer clustering whereby the data set contained 698 data points and 3 dimensions. The experiment resulted in ART2+AISK performing better and converging faster than ART2+PSO.

A CSA-based clustering algorithm named MOCLONAL was proposed by Nanda and Panda [35]. Contrasting the AISK algorithm by Kuo et al., MOCLONAL is inherently automatic, multi-objective, and uses the K-means algorithm only to generate the initial population. Integer encoding in the form of a locus-based adjacency scheme [19] is used, optimising 2 objective functions: variance and connectivity. Variance is a measure of the intra-cluster spread of the data while connectivity measures the degree to which nearby data is placed within the same cluster. Since MOCLONAL is multi-objective and follows the same concept of domination as AMOSA, the memory pool in this case is used to store the non-dominated solutions found. Therefore, as with other multi-objective algorithms, MOCLONAL outputs a Pareto-optimal set of results. Unlike traditional CSA, mutation rates for antibodies occurred at a rate depending on the amount of data points and the size of the antibody population. The algorithm was tested on one synthetic data set containing 250 two-dimensional data points that overlap in 5 spherical clusters. MOCLONAL was also tested on the UCI Iris and Lung Cancer data sets, comparing performance across all data sets with MOCK - a multi-objective evolutionary algorithm [19]. Using the Minkowski score as done by Saha and Bandyopadhyay [38], Nanda and Panda is able to select a single solution out of the Pareto-optimal set, which may be useful when only one result is required. It is noted, however, that this is only applicable when the true cluster for at least a portion of the data is known beforehand. As the Minkowski score is the difference between the obtained partition compared to the true clusters, this value is to be minimised. It was found that

Table 1: Summary of Algorithms

Algorithm	Author/s	Metaheuristic Class	Objectivity	Data Sets Used
GGA [2]	Agustín-Blas et al.	GA	Single	3 artificial, Iris, Wine
AGCUK [30]	Liu et al	GA	Single	50 artificial, Wisconsin Breast Cancer
TGCA [20]	He and Tan	GA	Single	4 artificial, Iris, Glass, Wisconsin Breast Cancer, Connectionist Bench, Ionosphere, Zoo, Lung Cancer
SA-RJMCMC [3]	Bandypadhyay	SA	Single	3 artificial, Wisconsin Breast Cancer, Kala Azar, IRS-1A
AMOSA [7]	Bandyopadhyaya et al.	SA	Multi	7 artificial
AMOSA'[1]	Acharya et al.	SA	Multi	Brain Tumor, Adult Malignancy, Small Round Blood Cell Tumor
AISK [25]	Kuo et al.	CSA	Single	Iris, Wine, Glass, Wisconsin Breast Cancer
MOCLONAL [35]	Nanda and Panda	CSA	Multi	1 artificial, Iris, Lung Cancer
AIS-based [27]	Li et al.	CSA	Single	DSR472, DBD438, SIL300, RSH136, nedtORIGINAL544

MOCLONAL performed better in the synthetic and Iris data sets while matching MOCK in Lung Cancer. The synthetic data caused the greatest disparity between the two, indicating that MOCLONAL may be better suited for overlapping data. Both algorithms, however, did not accurately determine the correct number of clusters for Lung Cancer and had high MS values (0.79 each). The algorithm was also used to classify actions of 3D human models. The algorithm was able to determine the 2 clusters by itself, comprising of normal and aggressive human behaviour.

In a study performed by Li et al. [27], a CSA-inspired clustering algorithm was proposed for the classification of 3D models. The algorithm's novelty lied within its ability to not only cluster 3D models with an empty database (unsupervised), but also recognise new incremental models and cluster them with previously seen models. This is possible due to the algorithm's particular use of memory cells, allowing it to reuse previous antibodies with the best affinity for new antigens. In this case, antigens are the different 3D models it has to cluster, and the affinity is determined by the Euclidean distance between antibody and antigen (intracluster distance). When a model is presented to the algorithm, the model is considered as "seen" if it matches one of the clusters already present in the memory pool via a recognising scope - a function that determines the similarity to an existing cluster. In this case, the memory cell is updated with the new model in mind, adjusting its recognising scope for further new models. If there is no match, the new model is considered as "unseen" and becomes the base for a new cluster within the memory pool through the standard use of affinity maturation, and clonal selection procedures. Although the number of clones produced is proportional to the affinity of each antibody, the mutation rate seems to be completely random. The algorithm was tested with different sets of 3D models, with dimensions ranging from 136 to 544. When compared to ASCAR [32] and CAIOC+Kmeans [31], the CSA-based algorithm performed better overall with the highest consistency, exhibiting above 90% accuracy for all three cases.

6 DISCUSSION

The methods explored in this paper all have the intrinsic ability of being able to find the optimal number of clusters without *a priori* knowledge of the data set. This is achieved either through two-stage approaches such as He and Tan's GA, or continuously throughout the algorithm like in MOCLONAL. This alone should make metaheuristics more compelling to use when clustering molecular dynamics trajectories as little may be known about the behaviour of the molecules simulated. Both single-objective and multi-objective methods were covered. Although single-objective algorithms are easier to understand and implement due to not having to worry about simultaneous objective function optimisation and Pareto-optimal sets, it is also less capable because of the very same reason when handling different data sets. Therefore, multi-objective algorithms will generally be preferred for the purpose of clustering molecular dynamics trajectories.

With every metaheuristic, the balance between exploration and exploitation must be considered during its development. An imbalance between the two can lead to wasted computational resources due to slow convergence or sub-optimal solutions due to premature

convergence. For genetic algorithms, this balance is mainly brought about by ensuring its selection, crossover and mutation methods allow for both performant and weaker solutions for reproduction. To converge in a reasonable time, adaptive probabilities for these steps can be taken, as with Agustín-Blas et al.'s approach. Simulated annealing presents a unique case whereby only a single solution is optimised in any given iteration. This inherently hinders its scope when exploring the search space. Multi-objective methods, as proposed by Bandyopadhyay et al., somewhat overcome this limitation via the use of Pareto-optimal sets in which multiple solutions of equal solution quality are presented to the user via the optimisation of multiple validity indices. For the clonal selection-based algorithms, exploitation is achieved through the cloning of solutions while exploration is performed by affinity maturation and receptor editing. Traditionally, the rate of maturation for CSA-based algorithms is often inversely proportional to the affinity of each antibody, presenting a bias towards solutions that display a higher fitness level. Different approaches have been taken, however, like with Nanda and Panda's MOCLONAL.

While most of the algorithms reviewed focus on outputting a single partition or Pareto-optimal set of partitions for a single problem, Li et al.'s clonal selection-based clustering algorithm provides a set of classes for different problems in the form of 3D models. Even though the algorithm is single-objective, adaptations could possibly be made to optimise multiple objective functions. The method can almost be seen as a hybrid between clustering and classification whereby new unrecognised models generate a new class, and recognised models are added to these existing classes. Interestingly, this can be paralleled with the clustering of molecular dynamics. Each frame from a trajectory can be seen as a 3D model, and while each frame consists of the same atoms to be examined, significantly different conformations of these atoms can be translated as unrecognised models while similar conformations can be said to belong to one of the existing clusters of models. This makes Li et al.'s approach particularly appealing when compared the other algorithms.

7 CONCLUSIONS

Clustering structural conformations from molecular dynamics trajectories is important for researchers to efficiently gather insights from them. Traditional clustering methods such as K-means are often used to do so even though many require *a priori* knowledge about the trajectories, and produce poor clusters due to getting stuck in local optima. Metaheuristics have been explored as an alternative due to many implementations that automatically find the optimal number of clusters while also escaping from local optima.

The genetic algorithm, simulated annealing, and artificial immune system (clonal selection-based) were found to provide promising results in their respective studies. The implementations all provided approaches that allowed for the automatic optimising of cluster count, with single and multi-objective methodologies. Multi-objective algorithms may be more beneficial for the clustering of different trajectories as conformations can vary greatly between them. By simultaneously optimising multiple objective functions to account for the different properties of each trajectory, better clusters may be obtained than that of single-objective methods.

Li et al.'s clonal selection-based algorithm is also of particular interest due the similarities of its use case in analysing 3D models. Although it is single-objective, the method could possibly be adapted to cater for multiple objective functions which may make it even more suitable for MD clustering.

REFERENCES

[1] Sudipta Acharya, Sriparna Saha, and Yamini Thadisina. 2016. Multiobjective Simulated Annealing-Based Clustering of Tissue Samples for Cancer Diagnosis. *IEEE Journal of Biomedical and Health Informatics* 20, 2 (2016), 691–698.

[2] L.E AgustıN-Blas, S Salcedo-Sanz, S Jiménez-Fernández, L Carro-Calvo, J Del Ser, and J.A Portilla-Figueras. 2012. A new grouping genetic algorithm for clustering problems. *Expert Systems With Applications* 39, 10 (2012), 9695–9703.

[3] S Bandyopadhyay. 2005. Simulated annealing using a reversible jump Markov chain Monte Carlo algorithm for fuzzy clustering. *IEEE Transactions on Knowledge and Data Engineering* 17, 4 (2005), 479–490.

[4] Sanghamitra Bandyopadhyay. 2012. *Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications.* (1st ed.. ed.). Vol. 9783642324512. 1–262 pages.

[5] S Bandyopadhyay and U Maulik. 2001. Nonparametric genetic clustering: comparison of validity indices. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 31, 1 (2001), 120–125.

[6] Sanghamitra Bandyopadhyay and Ujjwal Maulik. 2002. Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern Recognition* 35, 6 (2002), 1197–1208.

[7] S Bandyopadhyay, S Saha, U Maulik, and K Deb. 2008. A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation* 12, 3 (2008), 269–283.

[8] J.C Bezdek, S Boggavarapu, L.O Hall, and A Bensaid. 1994. Genetic algorithm guided clustering. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence.* IEEE, 34–39 vol.1.

[9] T Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics* 3, 1 (1974), 1–27. http://www.tandfonline.com/doi/abs/10.1080/03610927408827101

[10] G.A. Carpenter and S. Grossberg. 1988. Art 2: Self-organization of stable category recognition codes for analog input patterns. *Proceedings of SPIE - The International Society for Optical Engineering* 848 (1988), 272–280.

[11] David L Davies and Donald W Bouldin. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1, 2 (1979), 224–227.

[12] L.N de Castro and F.J Von Zuben. 2002. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation* 6, 3 (2002), 239–251.

[13] Kalyanmoy. Deb. [n. d.]. *Multi-objective optimization using evolutionary algorithms* (paperback ed. ed.). John Wiley Sons, Chichester ;.

[14] Kalyanmoy Deb. 1999. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation* 7, 3 (1999), 205–30.

[15] K Deb, A Pratap, S Agarwal, and T Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[16] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[17] Emanuel Falkenauer. 1994. A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems. *Evolutionary Computation* 2, 2 (1994), 123–144.

[18] Peter J. Green. 1995. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika* 82, 4 (1995), 711–732.

[19] J Handl and J Knowles. 2007. An Evolutionary Approach to Multiobjective Clustering. *IEEE Transactions on Evolutionary Computation* 11, 1 (2007), 56–76.

[20] Hong He and Yonghong Tan. 2012. A two-stage genetic algorithm for automatic clustering. *Neurocomputing* 81 (2012), 49–59.

[21] John Henry Holland et al. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press.

[22] E.R Hruschka, R.J.G.B Campello, A.A Freitas, and A.C.P.L.F de Carvalho. 2009. A Survey of Evolutionary Algorithms for Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 39, 2 (2009), 133–155.

[23] S. Kirkpatrick, Jr. Gelatt, C.D., and M.P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220 (1983).

[24] Joshua D. Knowles and David W. Corne. 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation* 8, 2 (2000).

[25] R. J Kuo, N. J Chiang, and Z.-Y Chen. 2014. Integration of Artificial Immune System and K-Means Algorithm for Customer Clustering. *Applied Artificial*

Intelligence 28, 6 (2014), 577–596. http://www.tandfonline.com/doi/abs/10.1080/08839514.2014.923167

[26] Chih-Chin Lai. 2005. A Novel Clustering Approach using Hierarchical Genetic Algorithms. *Intelligent Automation Soft Computing* 11, 3 (2005), 143–153.

[27] Xianghua Li, Chao Gao, Tianyang Lv, and Li Tao. 2012. A dynamic clustering algorithm based on artificial immune system for analyzing 3D models. In *2012 8th International Conference on Natural Computation*. IEEE, 854–858.

[28] Hwei-Jen Lin, Fu-Wen Yang, and Yang-Ta Kao. 2005. An efficient GA-based clustering technique. *Tamkang Journal of Science and Engineering* 8, 2 (2005), 113.

[29] Erik Lindahl. 2015. Molecular dynamics simulations. *Methods in molecular biology (Clifton, N.J.)* 1215 (2015), 3–26.

[30] Yongguo Liu, Xindong Wu, and Yidong Shen. 2011. Automatic clustering using genetic algorithms. *Appl. Math. Comput.* 218, 4 (2011), 1267–1279.

[31] T. Lv, S. Huang, X. Zhang, and Z.-X. Wang. 2006. A robust hierarchical clustering algorithm and its application in 3D model retrieval. In *First International Multi-Symposiums on Computer and Computational Sciences, IMSCCS'06*, Vol. 2. 560–567.

[32] Tian-yang Lv, Yu-hui Xing, Shao-bing Huang, Zheng-xuan Wang, and Wan-li Zuo. 2005. An auto-stopped hierarchical clustering algorithm for analyzing 3D model database. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 601–608.

[33] U Maulik and S Bandyopadhyay. 2003. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Transactions on Geoscience and Remote Sensing* 41, 5 (2003), 1075–1081.

[34] Ryan Melvin, Ryan Godwin, Jiajie Xiao, William Thompson, Kenneth Berenhaut, and Freddie Salsbury. 2016. Uncovering Large-Scale Conformational Change in Molecular Dynamics without Prior Knowledge. *Journal of Chemical Theory and Computation* 12, 12 (2016), 6130–6146. http://search.proquest.com/docview/1880017835/

[35] S. J Nanda and G Panda. 2012. Automatic clustering using MOCLONAL for classifying actions of 3D human models. In *2012 IEEE Symposium on Humanities, Science and Engineering Research*. IEEE, 945–950.

[36] Satyasai Jagannath Nanda and Ganapati Panda. 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation* 16 (2014), 1–18.

[37] William M Rand. 1971. Objective Criteria for the Evaluation of Clustering Methods. *J. Amer. Statist. Assoc.* 66, 336 (1971), 846–850. http://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356

[38] Sriparna Saha and Sanghamitra Bandyopadhyay. 2009. A new multiobjective simulated annealing based clustering technique using symmetry. *Pattern Recognition Letters* 30, 15 (2009), 1392–1403.

[39] Shokri Z Selim and K1 Alsultan. 1991. A simulated annealing algorithm for the clustering problem. *Pattern recognition* 24, 10 (1991), 1003–1008.

[40] Jianyin Shao, Stephen W Tanner, Nephi Thompson, and Thomas E Cheatham. 2007. Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms. *Journal of chemical theory and computation* 3, 6 (2007), 2312–2334.

[41] Viviana Siless, Ken Chang, Bruce Fischl, and Anastasia Yendiki. 2018. AnatomiCuts: Hierarchical clustering of tractography streamlines based on anatomical similarity. *NeuroImage* 166 (2018), 32–45.

[42] Tao Xiang and Shaogang Gong. 2008. Spectral clustering with eigenvector selection. *Pattern Recognition* 41, 3 (2008), 1012–1029.

[43] X.L. Xie and G. Beni. 1991. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 8 (1991), 841–847.