Framework for Pareto-Optimal Multimodal Clustering

Mikhail Bogatyrev[1], Dmitry Orlov[1]

[1]Tula State University, 92 Lenin ave., Tula, Russia

Abstract
The data used in Artificial Intelligence systems is often multimodal. Their representation in the form of formal contexts leads to contexts of high dimension. When constructing formal concepts and clustering on such contexts, the algorithms that are robust to the increasing dimension of contexts and capable of displaying a variety of clustering options are in demand. The modeling framework that meets these requirements is proposed. The framework uses multi-objective optimization and Evolutionary computation. The clustering results performed in the framework are compared with the known ones.

Keywords
multi-objective optimization, multimodal clustering, Pareto optimization, fact extraction.

1. Introduction

This paper is related to ongoing research in the area of applying Evolutionary computation for multi-objective optimization. It is a continuation of the research presented in [10]. In the Formal Concept Analysis (FCA), several similar tools are known [8, 9].

In FCA, the problem of multimodal clustering is solving on formal contexts and its solution depends on the dimension of the context. There are two types of solutions that are recognized as dense and non-dense clusters. Formal concepts acquired from conceptual lattice or by corresponding algorithm are dense clusters. Non-dense clusters of certain modality differ from dense ones in that their tensors may contain empty elements.

This paper describes an experimental framework for solving multi-objective optimization problems using evolutionary algorithms. Pareto-optimal solutions on formal contexts are considered here for two criteria: cluster density and volume. It is known that these criteria are contradictory. An increase in the value of one criterion is impossible without a decrease in the value of the other. The compromise solutions are needed here, which can be Pareto-optimal.

The paper is organized as follows. In the Section 2, background and related work are described. In the Section 3, proposed experimental framework is presented. The Section 4 contains some results of clustering made in the framework and compared with corresponding results acquired by the Data-Peeler algorithm. The paper ends with Conclusion section and References.

Published in Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph (Eds.): The 10th International Workshop "What can FCA do for Artificial Intelligence?", FCA4AI 2022, co-located with IJCAI-ECAI 2022, July 23 2022, Vienna, Austria, Proceedings, pp. 51–62.
okkambo@mail.ru (M. Bogatyrev)
0000-0001-8477-6006 (M. Bogatyrev); 0000-0003-0756-7491 (D. Orlov)

2. Background and Related Work

The background of this work consists of two areas. The first area is multi-objective optimization using evolutionary algorithms [11]. The second area is multimodal clustering in FCA.

2.1. Multi-objective Optimization with Evolutionary Algorithms

Multi-objective optimization is simultaneous optimization a number of objectives. Its specificity is manifested when the objectives conflict each other, i.e., improving the values of one objective, we worsen the values of another. This problem initiates the emergence of a set of compromise optimal solutions, commonly known as the Pareto-optimal solutions.

Pareto-optimal Multi-objective Optimization. The concept of Pareto optimality belongs to the mainstream in the domain of multi-objective optimization. Pareto optimality from the viewpoint of maximization optimization problem may be defined as follows. Let $\mathbb{S} = \{S_i\}$ is the set of solutions of multi-objective optimization problem, $i = 1, 2, \ldots, n, F = \{f_j\}, j = 1, 2, \ldots, m$ is the set of objective functions. Every solution is characterized by vector $\mathbf{f_i} = \{f_j(S_i)\}$. One feasible solution $\mathbf{f_i}$ is said to *dominate* another feasible solution $\mathbf{f_k}$ if and only if $f_j(S_i) \geq f_j(S_k)$ for all $j = 1, 2, \ldots, m$ and $f_j(S_i) > f_j(S_k)$ for least one objective function $f_d, d \in \{1, 2, \ldots, m\}$. A solution is said to be *Pareto optimal* if it is not dominated by any other solution. A *Pareto optimal* solution cannot be improved with respect to any objective function without worsening value at least one other objective function. The set of all feasible non-dominated solutions is referred to as the *Pareto optimal set*, and for a given Pareto optimal set, the corresponding objective function values in the objective space are called the *Pareto front*.

Evolutionary algorithms belong to the Evolutionary computation, the set of global optimization methods that use the *evolution of solutions*. The first known evolutionary algorithm is genetic algorithm, which realizes a probabilistic optimization technique based on the biological principles of evolution:

- encoding every solution as the string of symbols from certain alphabet (*chromosome*);
- using of a set (*population*) of solutions that evolves to one solution or to a subset of solutions corresponding to the extreme value of the certain quality criterion;
- applying various types of selecting the better solutions and (genetic) operators for manipulating solutions in the form of *mutation* and *crossover* of chromosomes.

The first two features of genetic and evolutionary algorithms determine their effectiveness in solving multi-objective optimization problems.

Encoding solutions as chromosomes allows one to simulate solutions of various problems. For example in clustering, chromosomes can directly represent clusters.

Using population of chromosomes is suitable for creating Pareto optimal solutions. There are several well-known multi-objective evolutionary algorithms (MOEAs) focused on obtaining Pareto optimal solutions: Niched Pareto Genetic Algorithm (NPGA), Strength Pareto Evolutionary Algorithm (SPEA), Non-dominated Sorting Genetic Algorithm (NSGA), and others reviewed in [11].

The mentioned algorithms are used in solving various problems of multi-objective optimization in engineering, business and science. They are also used in data clustering [12]. In this work, the use MOEAs in multimodal clustering of formal contexts represents their new application.

2.2. Multimodal Clustering Problem in FCA

In FCA, multimodal clustering is formulated as follows.

If $R \subseteq D_1 \times D_2 \times \cdots \times D_n$ is a relation on data domains $D_1, D_2, \ldots, D_n$ then *formal context* is an $n+1$ set:

$$\mathbb{K} =< K_1, K_2, \ldots, K_n, R > \tag{1}$$

where $K_i \subseteq D_i$. *Multimodal clusters* on the context (1) are *n* – sets

$$\mathbb{C} =< X_1, X_2, \ldots, X_n > \tag{2}$$

which have the closure property [4]:

$$\forall u = (x_1, x_2, \ldots, x_n) \in X_1, X_2, \ldots, X_n, u \in R \tag{3}$$

$\forall j = 1, 2, \ldots, n, \forall x_j \in D_j \setminus X_j < X_1, \ldots, X_j \cap \{x_1\}, \ldots, X_n >$ does not satisfy (3).

A multimodal cluster is a subset in the form of combinations of elements from different sets K_i. It is also defined as a closed *n*-set [3] since the closure property (3) provides its "self-sufficiency": it cannot be enlarged without violating closure property.

Formal concepts on multimodal formal context are those multimodal clusters where *for all* $u = (x_1, x_2, \ldots, x_k) \in X_1, X_2, \ldots, X_k, u \in R$ and *k* is maximally possible. In other words, they are the largest possible *k*-dimensional hypercubes completely filled with units. The concept of the density of a multimodal cluster is introduced in FCA and formal concepts are interpreted as absolutely dense clusters [3].

There are some practical arguments in favour of studying multimodal clusters as none dense concepts additionally to studying formal concepts. Non-dense multimodal clusters can contain important information. For example, the very fact that there are certain data instances in a subset of a cluster may be an indicator of the importance of this fact. If the cluster is not dense, then to find the rest of the data that is combined with the found instance, one need to refer to the formal context. However, the search in this case will be limited by the size of the found cluster.

The Need of Multi-objective Optimization. In addition to the density of clusters, their other characteristics of *volume, modality, diversity* and *coverage* have been introduced [6]. These characteristics illustrate the quality of multimodal clustering and in some cases help to interpret the contents of clusters.

Having a set of clustering quality parameters, the multimodal clustering problem is formulated as an optimization problem in which the extremum of the criterion based on mentioned characteristics is searched for [5, 6]. In fact, some of these characteristics, for example, the volume of clusters and their density, form conflicting criteria.

Therefore, multimodal clustering on formal context may be formulated as a multi-objective optimization problem.

There are directions in FCA in which the construction of multimodal clusters is associated with the solution of optimization problems [6, 7].

3. Experimental Framework

Consider the main functional elements of the proposed framework.

3.1. Evolutionary Multi-Objective Algorithm for Multimodal Clustering

The basis of our system is evolutionary multi-objective algorithm for multimodal clustering. The algorithm uses Evolutionary computation. Evolutionary approach is applied in Pareto-optimal optimization.

Our algorithm is based on the NSGA-II algorithm [13], which was adapted for clustering. We also expanded it with functions for visualization Pareto fronts.

The algorithm is shown on the Fig. 1. As any evolutionary algorithm, it contains functions being characteristic for genetic algorithms.

doSelection function realizes selection chromosomes according to the selection method. There are *proportional, random universal, tournament* and *truncation* selection methods realized in the algorithm. The specific selection method is picked through the user interface.

The *doMultipleCrossover* function, in addition to performing a crossover, accesses the original tensor in order to filter out the wrong chromosomes. We have also provided the crossover mode which is performed only in certain sections of chromosomes.

Encoding scheme. Encoding chromosomes is core element in evolutionary algorithms. After analyzing the several variants of chromosome encoding [12], we settled on the binary scheme organized according to the principle "one chromosome – one cluster". If formal context has modality n then a chromosome has n modal sections. In the sections, a number of gene is the number of an element of corresponding set in multimodal context. The units in the chromosome representing the cluster denote the elements included in this cluster. This binary encoding scheme is not compact because for large contexts with high modality the chromosomes will be very long. Nevertheless, in the task of clustering, it is much more convenient to work with such chromosomes than with chromosomes with more compact length. Explicit representation of clusters in the form of separate chromosomes does not require additional computations, which are necessary for other encodings. In addition, handling large binary strings is not a problem.

Algorithm 1 Evolutionary multi-objective clustering algorithm

Input: *tensor* is multidimensional context as the set of *n* samples on the axes of measurements;
Parameters:
sizePop is the size of population of chromosomes;
numpoints is the number of points of crossover;
mutationRate is the probability of mutation;
crossoverRate is the probability of crossover;
limitPop is the maximal number of populations;
countPop is the number of steps of evolution;
popFitness is the value of the fitness function for the entire population.
historyPop it stores all the populations
Output: *clusters* is the set of clusters in the form of a set of subsets.

```
    population ⟵ createPopulation[tensor, sizePop] creating a population of chromosomes chrom
while countPop ≤ limitPop do
    for all chrom do
        clusterDensity[chrom, tensor]
        clusterVolume[chrom, tensor]
        fitnessFunction[chrom, tensor]
    end
    doSelection[chrom, popFitness]
    doMultipleCrossover[{chrom1, chrom2}, numpoints, tensor]
    doMutation[chrom, mutationRate, tensor]
    popFitness[population] calculating the value of the fitness function for the entire population.
    combPop ⟵ historyPop ∪ population provides the elitism of the best chromosomes
    {front, rest} ⟵ survivorSelection[combPop, popSize] obtaining front and rear chromosomes
    historyPop ⟵ historyPop ∪ front replenishment with front-end chromosomes
    visualizePop[front, rest] visualization of the Pareto front
    for all chrom in front do
        {clusters} ⟵ getSubTensorChrom[chrom, tensor]  formation of front clusters from a tensor
    end
end
```

Figure 1: Evolutionary multi-objective algorithm for multimodal clustering.

The following characteristics of multimodal clusters are used in the clustering algorithm.

Cluster density and volume. For a cluster (2) its density is defined as

$$d(\mathbb{C}) = \frac{|R \cap (X_1 \times X_2 \times \cdots \times X_n)|}{|X_1| \times |X_2| \times \cdots \times |X_n|} \tag{4}$$

and volume of a cluster has the following form

$$\mathrm{v}(\mathbb{C}) = |X_1| \times |X_2| \times \cdots \times |X_n| \tag{5}$$

Cluster density and volume are contradictory criteria for cluster quality. A large and dense cluster is interesting because combinations of elements of its subsets set a property that manifests itself on a large number of elements and, possibly, means a regularity. However, often the clustered data is sparse and the existence of large and dense clusters on them is unlikely. Therefore, when selecting clusters, a trade-off between density and volume is provided by the algorithm.

Coverage and diversity. These two cluster characteristics were discussed and defined for triclustering problem in [6]. They also have generalized for multimodal clustering. Coverage is defined as a fraction of the tuples of the context included in at least one of the multimodal clusters. This can be defined by analogy with the definition in [6]:

$$\sigma(\Omega) = \sum_{(x_1,x_2,\ldots,x_n)\in R} [(x_1, x_2, \ldots, x_n) \in \bigcup_{(x_1,x_2,\ldots,x_n)\in\Omega} (x_1 \times x_2 \times \cdots \times x_n)/|R|], \quad (6)$$

where Ω is a set of multimodal clusters.

The data of the sets that make up the cluster modalities have different meanings. Sometimes it is important to control the coverage of the context by some subset of the cluster. In this case, in the expression (6), instead of a whole tuple $(x_1, x_2, \ldots, x_n)$ one of its elements is used.

The definition of cluster diversity given in [6] is valid for multimodal clusters:

$$\tau(\Omega) = 1 - \frac{\sum_j \sum_{i<j} \gamma(\Omega_i, \Omega_j)}{\frac{|\Omega|(|\Omega|-1)}{2}}, \quad (7)$$

where $\gamma(\Omega_i, \Omega_j)$ is an intersection function which is equal to 1 if clusters Ω_i, Ω_j intersect at least one of their subsets and 0 otherwise.

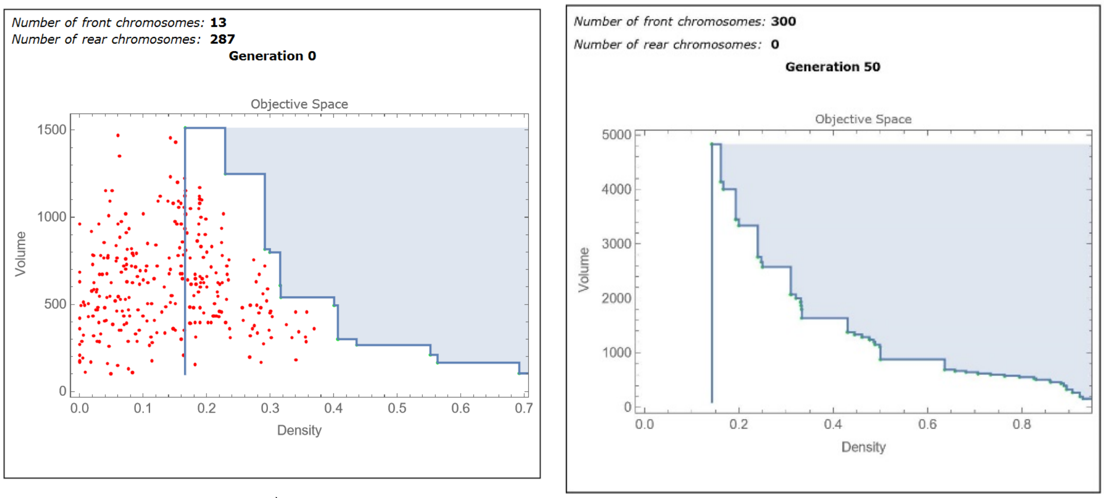
Elitist Nondominated Sorting. Evolution of solutions in the evolutionary algorithm is performed by applying genetic operators of selection, mutation and crossover to chromosome population. If the probability of mutation and crossover is high enough and the crossover is not tied to the peculiarities of chromosome encoding, then the algorithm performs random uncontrolled walks in the search space. By this way, the algorithm may explore most of the search space to find the global extremum of the fitness function. However, such walks reduce the convergence of the algorithm and, in principle, do not exclude its cycling in the regions of local extrema. Moreover, when calculating the Pareto front, random walks can lead to a "loss of the front", when the constructed Pareto front is destroyed at the next step of evolution. To exclude such phenomena we apply *elitism* [12, 13]. Elitism may be considered as an operator which preserves the better of parent and child solutions (or populations or Pareto fronts) so that a previously found better solution is never deleted. In the case of Pareto optimization, elitism is associated with dominance, and it is necessary to preserve not individual solutions, but, if possible, the entire front. In the MOEAs, elitism is realized as *nondominated sorting* [13].

3.2. Framework Realization

Considered framework is currently realized as desktop PC application with the use of some elements of Wolfram Mathematica™ environment. We use several Mathematica kernels for parallel computation. Since parallelization is natural for evolutionary algorithms, it can be realized on Mathematica kernels and helps to increase computing performance.

Java technology is also applied in the framework. Json data format is used for representing multidimensional formal contexts. We also plan to apply Java in future Web realization of the framework.

The framework uses program interface (API) Mathematica – Java and user interface with visualization Pareto fronts during evolution of computation.



a) b)

Figure 2: The initial (a) and final (b) Pareto fronts visualizations.

Fig. 2 illustrates the evolution of solutions in the evolutionary algorithm. The area of the search space in the initial generation on the Fig. 2 a) was expanded in the final generation on the Fig. 2 b).

4. Experiments

To demonstrate functionalities of the framework we present some results of multimodal clustering on the several data sets.

Data sets. The first data set contains data about offenses committed by juveniles [14]. We selected this data set to be able to compare our results with the results of triclustering performed by FCA algorithm of Data-Peeler [4]. This data set contains 30 objects which are the offense names, 7 attributes being the age group (m/f) which had the most amount of the certain sort offense, and 23 conditions being the years when offenses took place. Tricontext is presented as 690 incidents in the form {*offense name, age group, year*}. There are 79 formal concepts acquired from the context by Data-Peeler algorithm.

Other data sets are five tensors of dimensions from 2 to 6 generated randomly on the set $\{1, 2, \ldots, 10\}$. They are used in experiments to study the scaling of the algorithm.

Comparison with Data-Peeler. Using the juvenile offenses data set from [14] we have the possibility to compare the results of evolutionary clustering with the results of acquiring formal concepts from this data set performed by Data-Peeler algorithm [4]. The results of the comparison are as follows.

Juvenile offenses data have a feature that manifests itself in the multimodal clustering. All formal concepts found by Data-Peeler algorithm, with the exception of concepts having empty subsets of elements, contain unique attribute values denoting the gender and age of juveniles. Most of these concepts, namely 46 ones, contain attribute m_17 denoting boys of 17 years old. On the Fig. 3 a) there are examples of three such formal concepts. They are intersecting over subsets of objects and conditions and may be united into clusters having a larger volume and containing the same information as the set of small clusters.

Our algorithm, following the principle of increasing the volume of the cluster, finds namely these, coarse-grained clusters. Example of such cluster is shown on the Fig. 3 b). It demonstrates the fact that 17 years old boys had all types of offenses at all the years of observation.

Objects: Lock Aggravated Assault All Other Offenses Drive Under the Influence Drug Abuse Violations Drunkenness Embezzlement Forgery and Counterfeiting Gambling Liquor Laws Manslaughter by Negligence Murder and Nonnegligent Homicide Offenses Against the Family and Children Robbery Stolen Property Weapons
Attributes: Lock m_17
Conditions: Lock 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014

Objects: Lock Aggravated Assault All Other Offenses Drive Under the Influence Drug Abuse Violations Drunkenness Embezzlement Forgery and Counterfeiting Gambling Liquor Laws Manslaughter by Negligence Murder and Nonnegligent Homicide Robbery Stolen Property Suspicion Vagrancy Weapons
Attributes: Lock m_17
Conditions: Lock 1994 1995 1996 1997 1998 1999 2000 2002 2003 2016

Objects: Lock Aggravated Assault All Other Offenses Drive Under the Influence Drug Abuse Violations Drunkenness Embezzlement Forgery and Counterfeiting Gambling Liquor Laws Manslaughter by Negligence Murder and Nonnegligent Homicide Robbery Stolen Property Suspicion Weapons
Attributes: Lock m_17
Conditions: Lock 1994 1995 1996 1997 1998 1999 2000 2002 2003 2004 2005 2006 2008 2009 2016

a)

Density:	1.
Volume:	330
Individual coverage:	0.068323
Set #1 (15):	1) Aggravated Assault 2) All Other Offenses 3) Drive Under the Influence 4) Drug Abuse Violations 5) Drunkenness 6) Embezzlement 7) Forgery and Counterfeiting 8) Fraud 9) Gambling 10) Liquor Laws 11) Manslaughter by Negligence 12) Murder and Nonnegligent Homicide 13) Robbery 14) Stolen Property 15) Weapons
Set #2 (1):	1) m_17
Set #3 (22):	1) 1994 2) 1995 3) 1997 4) 1998 5) 1999 6) 2000 7) 2001 8) 2002 9) 2003 10) 2004 11) 2005 12) 2006 13) 2007 14) 2008 15) 2009 16) 2010 17) 2011 18) 2012 19) 2013 20) 2014 21) 2015 22) 2016

b)

Figure 3: a): Three formal concepts from 46 ones acquired by Data-Peeler and containing only boys of 17 years old (m_17) as attributes; b): the coarse-grained dense cluster that equivalent to these formal concepts.

Guided evolution. Our algorithm finds coarse-grained clusters of high density, which at the same time have the maximum volume. However, formal concepts of small volume containing no more than one element in one or several subsets are of particular interest. Among the formal concepts in the juvenile offenses data set there is the following one: Runaway, f_16, (2007, 2009, 2010). This cluster cannot be found when the algorithm is configured to the maximum density and volume of clusters. It is necessary either to look for clusters with a minimum volume and maximum density, or to use the evolution control tools inherent in the algorithm. In our case, we supplement the principle of nondominated sorting with additional conditions for the preservation of chromosomes containing only one unit in the first and second sections. This requires running a separate experiment, in which previously found clusters may be lost, but the desired clusters of small volume are found, reflecting the unique features of the data. Fig. 4 illustrates the result of applying guided evolution on the previous example of a single formal concept.

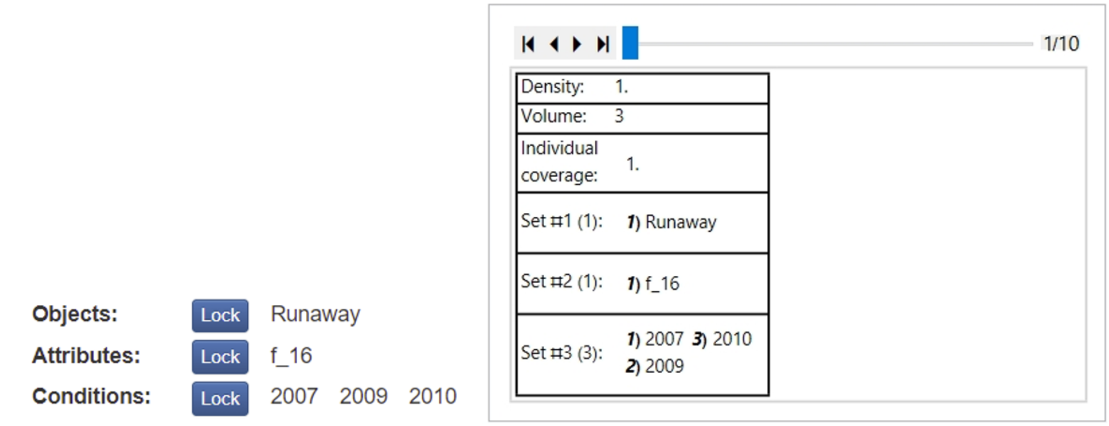


Figure 4: Formal concept and dense cluster of small volume.

Scaling the algorithm. The NSGA-II algorithm has computational complexity of $O(M * N^2)$ where M is the number of objectives and N is the population size [13]. In the problem of multimodal clustering on formal contexts, it is useful to estimate the performance of algorithms depending on the dimension of the formal context. For our algorithm, in which the dimension of the context determines the size of the chromosomes, such estimates are especially important.

Fig. 5 shows the results of testing algorithm on the randomly generated formal contexts having dimensions from 2 to 6. The population size was constant equal to 100 chromosomes, mutation probability was 0.01.

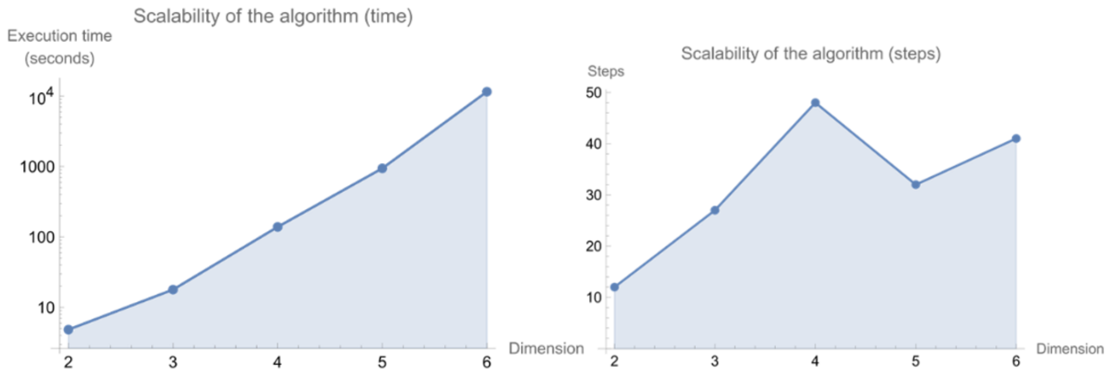


Figure 5: The dependence of the execution time and the number of steps required to construct the Pareto front on the dimension of the formal context.

The execution time is not an objective performance characteristic of algorithms, especially for evolutionary algorithms which have a population size as a parameter. However, we use the execution time to roughly estimate the scaling of the algorithm. The dependence of the execution time on the dimension of the formal context on the Fig. 5 is approximately estimated as $10^(k-2)$ where k is dimension on formal context.

The number of steps required to construct the Pareto front has no explicit dependence on the dimension of the context. This corresponds to the well-known properties of evolutionary algorithms, in which the number of evolution steps randomly depends on several parameters of the algorithm.

5. Conclusion

Evolutionary algorithms have certain advantages in implementing Pareto-optimal solutions to multi-objective optimization problems. Among them, there is one important which consists in the fact that the presence of a population of solutions supported by the algorithm allows one to naturally organize the formation of the Pareto front.

In this paper, we propose two innovations, which may be interested in FCA community.

The first innovation is the application of multi-objective optimization for the construction of multimodal clusters on formal contexts.

The second innovation is the ability to control the process of building multimodal clusters through the use of an evolutionary optimization algorithms.

In the future research, we plan to perform a deeper comparison of the work of the evolutionary algorithm with other well-known FCA algorithms [3].

Also we plan to explore several other encodings in evolutionary algorithm to exclude the appearance of extra chromosomes in the population.

We hope that the modeling framework proposed here would be useful for the FCA community.

Acknowledgments

The reported study was funded by Russian Foundation of Basic Research, the research project № 20-07-00055 and RFBR and Tula Region research project № 19-47-710007. Our special thanks is for Robin Gruna for visualization of Pareto fronts.

References

[1] Voutsadakis, G. Polyadic concept analysis. – Order. Vol. 19 (3). Pp. 295–304. (2002).

[2] Ganter, Bernhard; Stumme, Gerd; Wille, Rudolf, eds., Formal Concept Analysis: Foundations and Applications, Lecture Notes in Artificial Intelligence, No. 3626, Springer-Verlag. Berlin (2005)

[3] Dmitry V. Gnatyshak, Dmitry I. Ignatov, Sergei O. Kuznetsov: From Triadic FCA to Triclustering: Experimental Comparison of Some Triclustering Algorithms. In: Proceedings of the Tenth International Conference on Concept Lattices and Their Applications (CLA'2013), La Rochelle: Laboratory L3i, University of La Rochelle, pp. 249-260. (2013).

[4] Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed Patterns Meet N-ary Relations. In: ACM Trans. Knowl. Discov. Data. 3, 1, Article 3, 36 p. (2009)
[5] Ignatov D. I., Semenov A., Komissarova D. V., Gnatyshak D. V. Multimodal Clustering for Community Detection, in: Formal Concept Analysis of Social Networks / Ed. by R. Missaoui, S. Kuznetsov, S. Obiedkov. Springer, P. 59-96. (2017)
[6] Ignatov D. I., Gnatyshak D. V., Kuznetsov, S. O., Mirkin, B., G.: Triadic Formal Concept Analysis and triclustering: searching for optimal patterns. Mach. Learn. 101:271–302. (2015)
[7] Mirkin, B. G., Kramarenko, A. V.: Approximate bicluster and tricluster boxes in the analysis of binary data. In: Rough sets, fuzzy sets, data mining and granular computing, LNCS, Vol. 6743, pp. 248–256. (2011)
[8] Ignatov D. I., Egurnov D. Triclustring Toolbox, in: Supplementary Proceedings ICFCA 2019 Conference and Workshops Vol. 2378. CEUR Workshop Proceedings, 2019. P. 65-69. (2019)
[9] Neznanov A., Ilvovsky D., Kuznetsov S. FCART: A New FCA-based System for Data Analysis and Knowledge Discovery, in: Contributions to the 11th International Conference on Formal Concept Analysis. Dresden : Qucoza,. P. 31-44. (2013)
[10] Mikhail Bogatyrev, Dmitry Orlov and Tatyana Shestaka. Multimodal Clustering with Evolutionary Algorithms. Proc. of the 9th Int. Workshop "What can FCA do for Artificial Intelligence?" co-located with the 30th Int. Joint Conference on Artificial Intelligence (IJCAI 2021). Montréal, Québec, Canada, August 21, 2021. CEUR Proceedings, Vol. 2972. Pp. 71-86. (2021)
[11] Li, K., Wang, R., Zhang, T., et al. Evolutionary many-objective optimization: A comparative study of the state-of-the-art. IEEE Access, vol. 6, pp. 26194-26214. (2018)
[12] Hruschka E., Campello R., Freitas A., de Carballo A.: A Survey of Evolutionary Algorithms for Clustering. IEEE Transactions on Evolutionary Computation. V. 39. P. 133–155. (2009)
[13] Deb, A. Pratap, Agrawal, S. and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation, vol. 6, pp. 182–197. (2002)
[14] Kis, L.L., Sacarea, C., Troanca, D.: FCA Tools Bundle - a Tool that Enables Dyadic and Triadic Conceptual Navigation. In: Proceedings of the 5th International Workshop “What can FCA do for Artificial Intelligence?” Co-located with the European Conference on Artificial Intelligence, The Hague, The Netherlands, 30 August 2016, pp. 42–50 (2016)