Proceedings of the Federated Conference on Computer Science and Information Systems pp. 71–74
DOI: 10.15439/2017F340
ISSN 2300-5963 ACSIS, Vol. 11

# Data Clustering with Grasshopper Optimization Algorithm

Szymon Łukasik*†, Piotr A. Kowalski*†, Małgorzata Charytanowicz*‡ and Piotr Kulczycki*†
*Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
Email: {slukasik,pakowal,mchmat,kulpi}@ibspan.waw.pl
†Faculty of Physics and Applied Computer Science
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
Email: {slukasik,pkowal,kulpi}@agh.edu.pl
‡ Institute of Mathematics and Computer Science
The John Paul II Catholic University of Lublin
Konstantynów 1 H, 20-708 Lublin, Poland
Email: mchmat@kul.lublin.pl

***Abstract*—Dividing a dataset into disjoint groups of homogeneous structure, known as data clustering, constitutes an important problem of data analysis. It can be solved with broad range of methods employing statistical approaches or heuristic procedures. The latter often include mechanisms known from nature as they are known to serve as useful components of effective optimizers. The paper investigates the possibility of using novel nature-inspired technique – Grasshopper Optimization Algorithm (GOA) – to generate accurate data clusterings. As a quality measure of produced solutions internal clustering validation measure of Calinski-Harabasz index is being employed. This paper provides description of proposed algorithm along with its experimental evaluation for a set of benchmark instances. Over a course of our study it was established that clustering based on GOA is characterized by high accuracy – when compared with standard K-means procedure.**

## I. Introduction

RECENT years brought significant advances in the field of nature-inspired optimization. Several new algorithms have been proposed – aimed at tackling both continuous, combinatorial and multiobjective optimization problems. To illustrate this fact: Evolutionary Computation Bestiary website lists over 120 optimization techniques, with almost 30 of them being developed during last three years (that is between 2015 and 2017) [1]. The emergence of diverse techniques mimicking natural phenomena brought attention – due to their efficiency – but also criticism arguing that relying on metaphors is potentially leading the area of metaheuristics away from scientific rigor [2]. Most of studied algorithms however offer high performance on known set of benchmark instances – which makes investigating their performance in real-world optimization tasks worthwhile.

Grasshopper Optimization Algorithm (GOA) is an optimization technique introduced by Saremi, Mirjalili and Lewis in 2017 [3]. It includes both social interaction between ordinary agents (grasshoppers) and the attraction of the best individual. Initial experiments performed by authors demonstrated promising exploration abilities of the GOA – and they will be further examined in the course of our study.

The goal of this contribution is to evaluate clustering method which uses GOA as the optimization strategy – aimed at minimizing the value of Calinski-Harabasz index [4] – one of internal clustering validity measures.

Cluster analysis constitutes a data mining problem of identifying homogeneous groups in data. Clustering can be perceived as combinatorial optimization problem – which is known to be NP-hard [5]. It is the reason why diverse heuristic approaches have been already used to tackle it [6], [7]. As a point of reference classic K-means [8] algorithm can be named. It is founded on minimizing the within-cluster sum of squares (WCSS) and its main drawback is a convergence to a local minimum of WCSS value – without a guarantee of obtaining the global one. That is why more up-to-date approaches are based on using metaheuristic techniques to solve clustering problem in the alternative way. Previous work in this area involve the use of – for instance – Flower Pollination Algorithm [9] and Krill Herd Algorithm [10]. The importance of clustering manifests itself through a variety of disciplines where its instances appear, e.g. in agriculture [11], automatic control [12], marketing [13] or text mining [14].

The paper is organized as follows. First, in the next Section, the general description of data clustering is given along with its formulation within the field of optimization. It is followed by the brief introduction to the Grasshopper Optimization Algorithm which is the most important component of the technique described in this paper. Section 3 explains the details of the clustering approach and subsequent part of the paper covers the results of numerical experiments along with comparative analysis. Finally general remarks regarding algorithms's features and planned further studies are under consideration.

## II. Methodological Background

### A. Data Clustering and Its Formulation in the Optimization Domain

Let us to denote $Y$ as a data matrix of $M \times N$ dimensionality. Its $N$ columns represent features describing objects. They in turn correspond to matrix rows, referred to as dataset elements or cases. The goal of clustering is to assign dataset elements $y_1, ..., y_M$ to clusters $CL_1, CL_2, ..., CL_C$.

Clustering remains an unsupervised learning procedure, frequently with known number of clusters $C$ being the only information available. Cluster validation constitutes a task of assessing if obtained solution reflects the structure of the data and natural groups which can be identified within its records [15]. So called external validation consists of using correct cluster labels and comparing them directly with the results of clustering whereas internal validation uses only partitioned data. Calinski-Harabasz index is representative technique of the latter. It can be written as:

$$I_{CH} = \frac{N-C}{C-1} \frac{\sum_{i=1}^{C} d(u_i, U)}{\sum_{i=1}^{C} \sum_{x_j \in CL_i} d(x_j, u_i)} \tag{1}$$

wheras $u_i \in R^N$ for non-empty cluster $CL_i$ corresponds to cluster center defined by:

$$u_i = \frac{1}{M_i} \sum_{y_j \in CL_i} y_j, \quad i = 1, ..., C \tag{2}$$

with $M_i$ being cardinality of cluster $i$ and – likewise – $U$ corresponds to the center of gravity of the dataset:

$$U = \frac{1}{M} \sum_{j=1}^{M} y_j. \tag{3}$$

Clustering solutions which describe the dataset structure will result in high value of $I_{CH}$ index. The choice of this index was motivated by our successful experiments on other heuristic algorithms using $I_{CH}$ value [10] as a key component. Also recent studies on clustering indices demonstrate its sound potential to validate clustering solutions [16].

### B. Grasshopper Optimization Algorithm

GOA represents a population-based metaheuristic which is aimed at solving continuous optimization problems, that is finding argument (solution) $x^*$ which minimizes cost function $f : S \to R$. It can be formally written as:

$$x^* = \arg\min_{x \in S} f(x), \tag{4}$$

with $S \subset R^D$. Population based heuristic algorithms solve (4) using a swarm of $P$ individual agents, in iteration $k$ of the algorithm represented by a set $\{x_p\}_{p=1}^{P}$, with $x_p = [x_{p1}, x_{p2}, ...x_{pD}]$. The important concept for the construction of this class of procedures is also a measure of closeness between two swarm members $p_1$ and $p_2$, denoted here by Euclidean distance $dist(x_{p_1}, x_{p_2})$. The best solution found by the swarm within $k$-iterations is stored as $x^*(k)$. It is also assumed here that search space $S$ is bounded and this type of constraints is represented by the values of the lower $LB_1, LB_2, ..., LB_D$ and upper bound $UB_1, UB_2, ..., UB_D$. Effectively it means that:

$$LB_d \leq x_{pd}(k) \leq UB_d \tag{5}$$

for all $k = 1, 2, ...$, $p = 1, 2, ..., P$ and $d = 1, 2, ..., D$.

Grasshopper Optimization Algorithm claims to be inspired by the social behavior of grasshoppers – insects of *Orthoptera* order (suborder *Caelifera*) [3]. Each member of the swarm constitutes a single insect located in search space $S$ and moving within its bounds. The algorithm is reported to implement two components of grasshoppers movement strategies. First it is the interaction of grasshoppers which demonstrates itself through slow movements (while in larvae stage) and dynamic motion (while in insect form). The second corresponds to the tendency to move towards the source of food. What is more deceleration of grasshoppers approaching food and eventually consuming is also taken into account.

The movement of individual $p$ in iteration $k$ (index $k$ was omitted for the sake of readability) can be written using the following equation:

$$x_{pd} = c \left( \sum_{q=1, q \neq p}^{P} c \frac{UB_d - LB_d}{2} s(|x_{qd} - x_{pd}|) \frac{x_{qd} - x_{pd}}{dist(x_q, x_p)} \right) + x_d^* \tag{6}$$

with $d = 1, 2, ..., D$. Parameter $c$ is decreased according to the formula:

$$c = c_{max} - k \frac{c_{max} - c_{min}}{K} \tag{7}$$

with maximum and minimum values – $c_{max}$, $c_{min}$ respectively – and $K$ representing maximum number of iterations serving as algorithm's termination criterion. First occurrence of $c$ in (6) reduces the movements of grasshoppers around the target – balancing between exploration and exploitation of the swarm around the target. It is analogous to the inertia weight present in the Particle Swarm Optimization Algorithm. Component $c\frac{UB_d - LB_d}{2}$, as noted in [3], linearly decreases the space that the grasshoppers should explore and exploit. Finally function $s$ defines the strength of social forces, and was established by creators of the algorithm as:

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} \tag{8}$$

with $l = 1.5$ and $f = 0.5$.

To sum up GOA written using pseudocode and symbols introduced in the paper and taking into account all important elements – like initialization or calculation of the best solution – is presented as Algorithm 1.

## III. GOA-based Clustering Technique

Using any heuristic optimization algorithm requires choosing proper solution representation. In the case of clustering it is natural to represent solution as a vector of cluster centers $x_p = [u_1, u_2, ..., u_C]$. Consequently the dimensionality $D$ used

**Algorithm 1** Grasshopper Optimization Algorithm

$k \leftarrow 1$, $f(x^*(0)) \leftarrow \infty$ {initialization}
**for** $p = 1$ to $P$ **do**
  $x_p(k) \leftarrow$ Generate_Solution$(LB, UB)$
**end for**
{find best}
**for** $p = 1$ to $P$ **do**
  $f(x_p(k)) \leftarrow$ Evaluate_quality$(x_p(k))$
  **if** $f(x_p(k)) < f(x^*(k-1))$ **then**
    $x^*(k) \leftarrow x_p(k)$
  **else**
    $x^*(k) \leftarrow x^*(k-1)$
  **end if**
**end for**
**repeat**
  $c \leftarrow$ Update_c$(c_{max}, c_{min}, k, K_{max})$
  **for** $p = 1$ to $P$ **do**
    {move according to formula (6)}
    $x_p(k) \leftarrow$ Move_Grasshopper$(c, UB, LB, x^*(k))$
    {correct if out of bounds}
    $x_p(k) \leftarrow$ Correct_Solution$(x_p(k), UB, LB)$
    $f(x_p(k)) \leftarrow$ Evaluate_quality$(x_p(k))$
    **if** $f(x_p(k)) < f(x^*k)$ **then**
      $x^*(k) \leftarrow x_p(k)$, $f(x^*k) \leftarrow f(x_p(k))$
    **end if**
  **end for**
  **for** $p = 1$ to $P$ **do**
    $f(x_p(k+1)) \leftarrow f(x_pk)$, $x_p(k+1) \leftarrow x_p(k)$
  **end for**
  $f(x^*(k+1)) \leftarrow f(x^*k)$, $x^*(k+1) \leftarrow x^*(k)$
  $k \leftarrow k + 1$
**until** $k < K$
**return** $f(x^*(k))$, $x^*(k)$

in the description of GOA, in the case of data clustering problem, is equal to $C * N$.

Another important aspect is choosing proper tool of assessing the quality of generated solutions. Here an idea already presented in [9] is implemented. After assigning each data element $y_i$ to the closest cluster center the solution $x_p$ (representing those centers) is evaluated according to the formula:

$$f(x_p) = \frac{1}{I_{CH,p}} + \#_{CL_{i,p}=\emptyset,\, i=1,\ldots,C}. \tag{9}$$

It is equivalent to adding to the inverse value of Calinski-Harabasz index – calculated for solution $p$ – the number of empty clusters identified in $x_p$ clustering solution written above as $\#_{CL_{i,p}=\emptyset,\, i=1,\ldots,C}$. The idea behind appending the second component in (9) is penalizing solutions which do not include desirable number of clusters.

## IV. Experimental Evaluation

Evaluating clustering algorithms is in essence a difficult task due to unsupervised character of this problem. It is usually approached by performing cluster analysis on the labeled dataset containing the information about assignment of data elements to classes. Subsequently, clustering solution understood as a set of cluster indexes provided for all data points should be compared with a set of class labels. Such a comparison can be done with the use of Rand index [17], external validation index which measures similarity between cluster analysis solutions. It is characterized by a value between 0 and 1. Low value of $R$ suggests that the two clusterings are different and 1 indicates that they represent exactly the same solution – even when the formal indexes of clusters are mixed.

As a point of reference for evaluating performance of clustering methods classic K-means algorithm is being used. It is also the case of this contribution. For the experiments we used a set of benchmark datasets – based on real-world examples taken from the UCI Machine Learning Repository [18]. In the same time a set of standard synthetic clustering benchmark instances known as S-sets was used [19].

TABLE I: Characteristics of investigated datasets

| Dataset | *M* | *N* | *C* | Dataset | *M* | *N* | *C* |
|---|---|---|---|---|---|---|---|
| *glass* | 214 | 9 | 6 | *yeast* | 1484 | 8 | 10 |
| *wine* | 178 | 13 | 3 | *s1* | 5000 | 2 | 15 |
| *iris* | 150 | 4 | 3 | *s2* | 5000 | 2 | 15 |
| *seeds* | 210 | 7 | 3 | *s3* | 5000 | 2 | 15 |
| *heart* | 270 | 13 | 2 | *s4* | 5000 | 2 | 15 |

Table I provides the description of the datasets used in the numerical experiments. It contains properties like dataset size $M$, dimensionality $N$ and the number of classes $C$ – used as desired number of clusters for the grouping algorithms.

To evaluate clustering methods they were run 30 times with mean and standard deviation values of Rand index – $\overline{R}$ and $\sigma(R)$ – being recorded. For GOA-based algorithm a population of $P = 20$ swarm members was used. Algorithm terminates when $C * N * 1000$ cost function evaluations were performed. It is a standard strategy for evaluating metaheuristics – making the length of search process dependent on data dimensionality.

First default values of all GOA parameters were used, with $c = 0.00001$. It means that $c$ quickly approaches values close to zero. Summary of obtained results for this case is provided in Table II. It is easy to observe that GOA-based clustering outperforms K-means on the majority of the datasets – it is also less prone to getting stuck in local minima (it is indicated by the fact that it is less stable in terms of performance). We studied also the effect of using alternative values for parameter $c_{min}$ (using $c_{max} = 1$ seems natural for the construction of normalized "schedule"). Table III provides the results of these experiments. First, we have used fixed values for $c_{min}$ – higher than the one suggested by creators of the algorithm. This approach brings clearly very positive results. For most of datasets the performance of clustering algorithm has improved (as indicated by bold font). Especially the value $c_{min} = 0.001$ seems to be functioning very well.

We have also studied the possibility of using random values of $c$ in the interval $[0, 1]$. It is a common strategy of "embed-

TABLE II: K-means vs GOA-based clustering (with default parameter values)

| | K-means clustering | | GOA clustering (default $c_{min} = 0.00001$) | |
|---|---|---|---|---|
| | $\overline{R}$ | $\sigma(R)$ | $\overline{R}$ | $\sigma(R)$ |
| *glass* | 0.619 | 0.061 | 0.643 | 0.035 |
| *wine* | 0.711 | 0.014 | 0.730 | 0.000 |
| *iris* | 0.882 | 0.029 | 0.892 | 0.008 |
| *seeds* | 0.877 | 0.027 | 0.883 | 0.004 |
| *heart* | 0.522 | 0.000 | 0.523 | 0.000 |
| *yeast* | 0.686 | 0.033 | 0.676 | 0.034 |
| *s1* | 0.980 | 0.009 | 0.990 | 0.006 |
| *s2* | 0.974 | 0.010 | 0.984 | 0.006 |
| *s3* | 0.954 | 0.006 | 0.960 | 0.005 |
| *s4* | 0.944 | 0.006 | 0.951 | 0.003 |

TABLE III: Impact of parameter $c$ on the performance of GOA-based clustering

| | chaotic $c$ | | $c_{min} = 0.001$ | | $c_{min} = 0.1$ | |
|---|---|---|---|---|---|---|
| | $\overline{R}$ | $\sigma(R)$ | $\overline{R}$ | $\sigma(R)$ | $\overline{R}$ | $\sigma(R)$ |
| *glass* | 0.630 | 0.034 | **0.652** | **0.033** | 0.651 | 0.034 |
| *wine* | **0.730** | **0.000** | **0.730** | **0.000** | **0.730** | **0.000** |
| *iris* | **0.894** | **0.016** | **0.895** | **0.008** | 0.891 | 0.009 |
| *seeds* | 0.881 | 0.005 | 0.881 | 0.005 | 0.882 | 0.004 |
| *heart* | 0.522 | 0.000 | **0.523** | **0.000** | **0.523** | **0.000** |
| *yeast* | 0.669 | 0.036 | **0.690** | **0.029** | **0.690** | **0.025** |
| *s1* | 0.987 | 0.006 | **0.991** | **0.005** | **0.991** | **0.006** |
| *s2* | 0.982 | 0.005 | **0.985** | **0.005** | **0.986** | **0.005** |
| *s3* | 0.957 | 0.005 | **0.960** | **0.004** | **0.961** | **0.004** |
| *s4* | 0.949 | 0.003 | **0.951** | **0.003** | **0.951** | **0.003** |

ding" chaotic behavior into metaheuristic – which should result in enriching the search behavior [20]. In this case this approach does not work well. A decrease in the algorithm performance was predominantly observed. Still, such a "chaotic-enhanced" GOA-based clustering algorithm outperforms K-means in the most of investigated data mining cases.

## V. Conclusion

The paper proposes new clustering approach based on recently introduced Grasshopper Optimization Algorithm. Besides the description of the method the results of its experimental evaluation were also discussed. It was established that GOA-based approach offers high performance with respect to the standard K-means algorithm, both in terms of average quality of solutions and their stability. We also examined the impact of important algorithm's parameter – namely value of $c$. Possibility of using both fixed values for the lower bound of $c$ (alternative to the default $c_{min} = 0.00001$) as well as random strategy (which proved to be mostly unsuccessful) were inspected.

Further studies within the scope of this paper should include more detailed analysis of the impact of population size and coefficient $c$ on the quality of obtained solutions. The importance of the first aspect stems from the fact that the algorithm is characterized by quadratic time complexity with regards to the population size. It essentially means that choosing proper, compact $P$ value is important for the success of GOA-based optimization. Choosing the right scheme of $c$ alteration seems also of great importance. Therefore the idea of using alternative function to the standard linearly decreasing one should be explored.

## References


[1] "Evolutionary computation bestiary," http://conclave.cs.tsukuba.ac.jp/research/bestiary/, accessed May 06 2017.

[2] K. Sörensen, "Metaheuristics - the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.

[3] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Advances in Engineering Software*, vol. 105, pp. 30 – 47, 2017.

[4] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[5] W. J. Welch, "Algorithmic complexity: three np- hard problems in computational statistics," *Journal of Statistical Computation and Simulation*, vol. 15, no. 1, pp. 17–25, 1982.

[6] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Applied Soft Computing*, vol. 10, no. 1, pp. 183 – 197, 2010.

[7] J. Senthilnath, S. Omkar, and V. Mani, "Clustering using firefly algorithm: Performance study," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 164 – 171, 2011.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66*, 1967, pp. 281–297.

[9] S. Łukasik, P. A. Kowalski, M. Charytanowicz, and P. Kulczycki, "Clustering using flower pollination algorithm and calinski-harabasz index," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 2724–2728.

[10] P. A. Kowalski, S. Łukasik, M. Charytanowicz, and P. Kulczycki, "Clustering based on the krill herd algorithm with selected validity measures," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2016, pp. 79–87.

[11] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Żak, "Complete gradient clustering algorithm for features analysis of X-Ray images," in *Information Technologies in Biomedicine*, ser. Advances in Intelligent and Soft Computing, E. Piętka and J. Kawa, Eds. Springer Berlin Heidelberg, 2010, vol. 69, pp. 15–24.

[12] S. Łukasik, P. Kowalski, M. Charytanowicz, and P. Kulczycki, "Fuzzy models synthesis with kernel-density-based clustering algorithm," in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on*, vol. 3, Oct 2008, pp. 449–453.

[13] H. Müller and U. Hamm, "Stability of market segmentation with cluster analysis - a methodological approach," *Food Quality and Preference*, vol. 34, pp. 70 – 78, 2014.

[14] C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Springer US, 2012, pp. 77–128.

[15] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[16] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243 – 256, 2013.

[17] H. Parvin, H. Alizadeh, and B. Minati, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, 1971.

[18] "UCI machine learning repository," http://archive.ics.uci.edu/ml/, accessed May 10 2017.

[19] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognition*, vol. 39, no. 5, pp. 761 – 775, 2006.

[20] A. Kaveh, *Chaos Embedded Metaheuristic Algorithms*. Cham: Springer International Publishing, 2014, pp. 369–391.