# A Biological Driven Method for Correlation Clustering

András Fülöp, *University of Debrecen, Hungary*

University of Debrecen
Faculty of Informatics

## Introduction

A lot of the widespread clustering algorithms (k-means, k-center, etc.) require the number of clusters in order to operate. However there are situations – such as clustering webpages – in which this requirement can not be fulfilled. Bansel, Blum and Chawla presented a clustering method in their paper [2] which does not specify the number of clusters. Instead it creates optimal sized groups based on similarities. For this purpose it requires a $f(x, y)$ function which defines whether $x$ and $y$ are similar $(+)$ or different $(-)$. The goal of the correlation clustering is to maximize similarity and minimize difference inside the clusters. The NP-completeness of the problem was also proven in the given paper.
Correlation clustering has many applications. It can be used for creating customer recommendation systems, where based on the consumed products, the similiar customers can be grouped together and treated in the same way. Clustering web documents thematically is also an important field of interest. Using the term frequencies the similar documents can be processed. Overlapping clusters can emerge from the data which makes the problem more challenging. [9]
If we consider the data points as nodes in the graph $G$ then we can label the edges between the nodes based on the function $f$. An edge between similar nodes is labelled with $+$, and an edge between different nodes is marked with $-$. Figure 1 shows an example of such a graph. A $-$ labelled edge inside a cluster is a *negative mistake* and a $+$ labelled edge between clusters is a *positive mistake*. Thus the goal of the correlation clustering is either minimizing difference or maximizing similarity. If we want to minimize difference, we minimize the number of $+$ labelled edges between clusters and the $-$ labelled edges inside the clusters. In order to maximize similarity, we have to maximize the $+$ edges inside the cluster and the $-$ edges between the clusters.

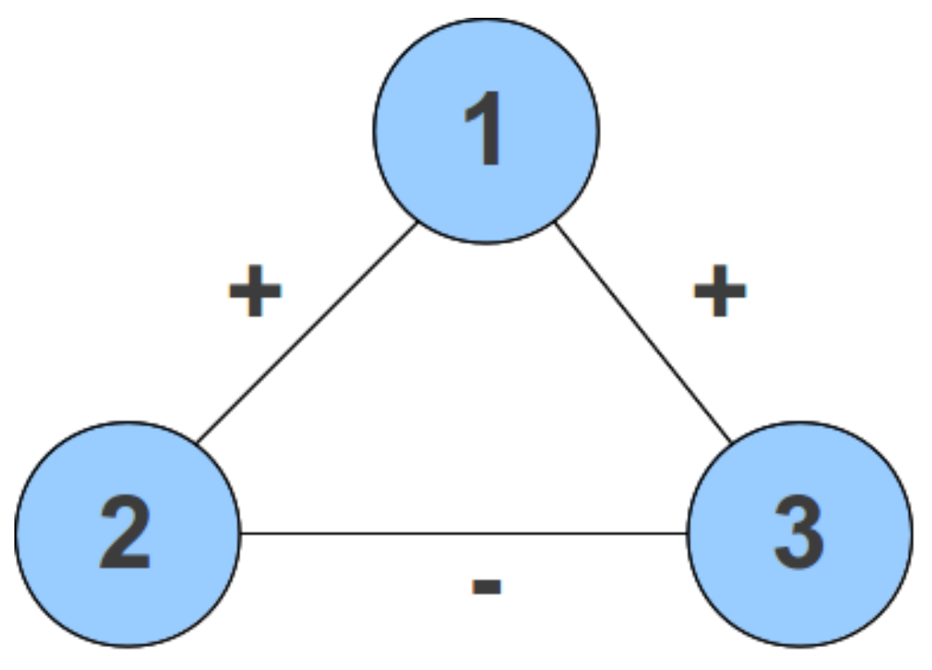

Figure 1: An erroneous triangle.

## Previous Work

In their paper [2], Bansel et al. introduced an algorithm to approximate the optimal clustering by minimizing differences in complete graphs. Their method is based on counting the *erroneous triangles* (showed in Figure 1) inside the clusters. These triangles consists of two $+$ labelled edges and a $-$ labelled edge. The algorithm is as follows: First select a node arbitrarily and all of it's positive neighbours, this will be the starting cluster. Remove a node from the cluster if a $\delta$ percentage of it's positive neighbours are outside of the cluster. Add every node to the cluster which has more positive neighbours inside the cluster than a given $\delta$ parameter. Remove the cluster's nodes from the dataset. Repeat until there are no nodes left, or if the remaining nodes do not have any positive neighbours. [3]
The minimizing difference viewpoint was extended by Demaine and Immorlica in [5]. They presented an algorithm for general graphs using linear programming and region-growing techniques. It first solves a linear program and then uses the results as weights. Greater weights represent weaker similarity. In the last step a region growing technique is used to group the nodes. [3] Other solutions are presented in [1].

## Ant System

Based on the work in [8], we decided to use a different approach on the problem. In this section we shortly discuss an optimizing algorithm called Ant System which is based on the ants' food finding method. The algorithm comes from the field of Swarm Intelligence. It was first introduced by Dorigo, Maniezzo and Colorni in a technical report [6] and it's initial name was Ant Cycle.
The algorithm was inspired by the stigmergy of the ants. Stigmergy is a way of communication between the ants while searching for an optimal route between the found food and their nest. They lay down pheromones to inform the others. Initially the ants are wandering in the environment. Once food is located by an ant it begins to lay down pheromones (Figure 2 - $1^{st}$ subfigure). If the same route is taken additional pheromones are laid down. Pheromones are laid down to other newly discovered paths (Figure 2 - $2^{nd}$ subfigure). Through positive feedback finally the optimal route is enforced (Figure 2 - $3^{rd}$ subfigure). [4]

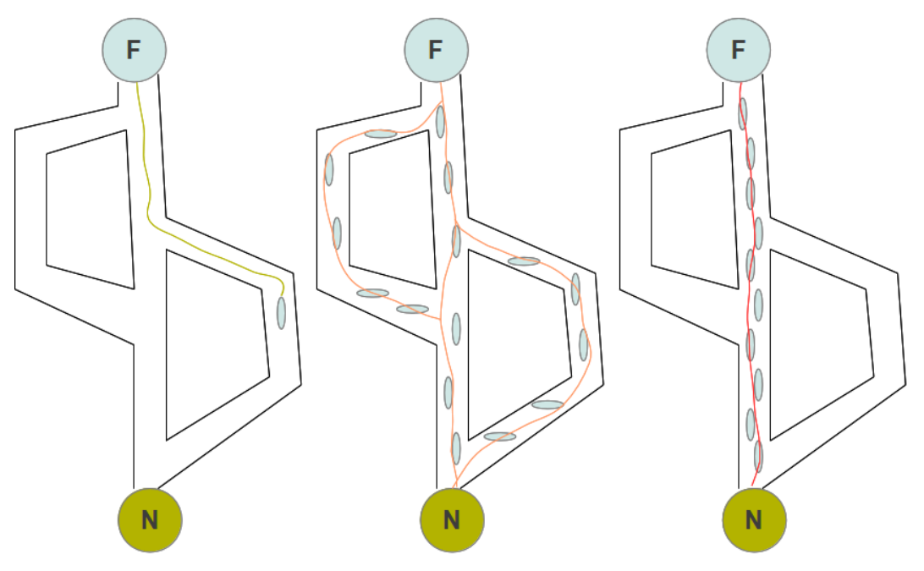

Figure 2: The route of the ants.

The method uses both heuristic and historical components in order to find the optimal solution. The candidate solutions are constructed in a probabilistic manner. The probability of selecting a component is computed from the previous results (historical) and from the actual cost (heuristic). Pheromones are layed down proportionally to the overall goodness of the constructed solution. Pheromones are also decayed to ensure that the most recent and useful information is retained.
The algorithm is as follows: First generate a solution based on heuristics and initialize pheromones. In every following turn, construct a probabilistic solution and based on the goodness of the solution, update pheromones. Repeat until the stopping criteria is reached.
The probability of selecting a component is

$$P_{i,j} = \frac{\tau_{i,j}^{\alpha} * \mu_{i,j}^{\beta}}{\sum_{k=1}^{c} \tau_{i,j}^{\alpha} * \mu_{i,j}^{\beta}}$$

where $\tau_{i,j}$ is the pheromone value for the component $(i, j)$, $\mu_{i,j}$ is the maximizing contribution to the overall score, $\alpha$ is the historical coefficient and $\beta$ is the heuristic coefficient.

The update rule for the pheromones are

$$\tau_{i,j} = (1 - \rho) * \tau_{i,j} + \sum_{k=1}^{n} \Delta_{i,j}^{k}$$

where $\rho$ is the decay factor, $n$ is the number of ants and $\sum_{k=1}^{n} \Delta_{i,j}^{k}$ is the sum of $\frac{1}{\text{Cost}}$ for those solutions that include $i, j$. [4]

## Our solution

Given a $G = (V, E)$ labelled graph with similar $+$ and different $-$ nodes, let $M$ be a matrix such that $m_{i,j}$ is the label of the edge between node $i$ and $j$. It's value can be $1$ (similar), $-1$ (different) or $0$ (not connected). Table 1 shows an example of such a matrix. Let function $f : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ define a clustering. The node $i$ and $j$ are in the same cluster only if $f(i) = f(j)$. The cost function is

$$c(f, M) = -\sum_{i<j} \delta_{f(i)f(j)} m_{i,j} + \frac{1}{2} \sum_{i<j} (m_{i,j} + |m_{i,j}|) \tag{1}$$

where $\delta$ is the Kroenecker delta symbol [7]. If $G$ is complete and there are no $+$ edges the ideal clustering is that every node is a singleton cluster. In addition, if $G$ is complete and there are only $+$ edges the ideal cluster will include every node in $V$. Thus the optimal size of the cluster is determined by

$$r(q) = \max_{i} \left\{ \frac{C(i, q)}{N} \right\}$$

where $q$ is the density of positive edges, $C(i, q)$ is the number of nodes in the cluster $i$ with a given $q$ density, and $N$ is the size of the problem [7].

Table 1: Example matrix for Figure 1.

| M | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | -1 |
| 3 | 1 | -1 | 0 |

In order to solve the correlation clustering problem with the Ant System method we considered the following:

- The graph is stored as an $M$ matrix.
- We used the function in Equation 1 as cost function.
- Each ant agent starts from a random clustering, and then select the most probable cluster labels for every position.

## Results

We ran our tests on a randomly generated graph with $200$ nodes. With every different coefficient setup we ran $100$ tests for every $q$ value. The $q$ values start from $0$ and increase by $5\%$ until reaching $100\%$. The different colours represents different parameter setups. The results are comparable with the results found in [7].

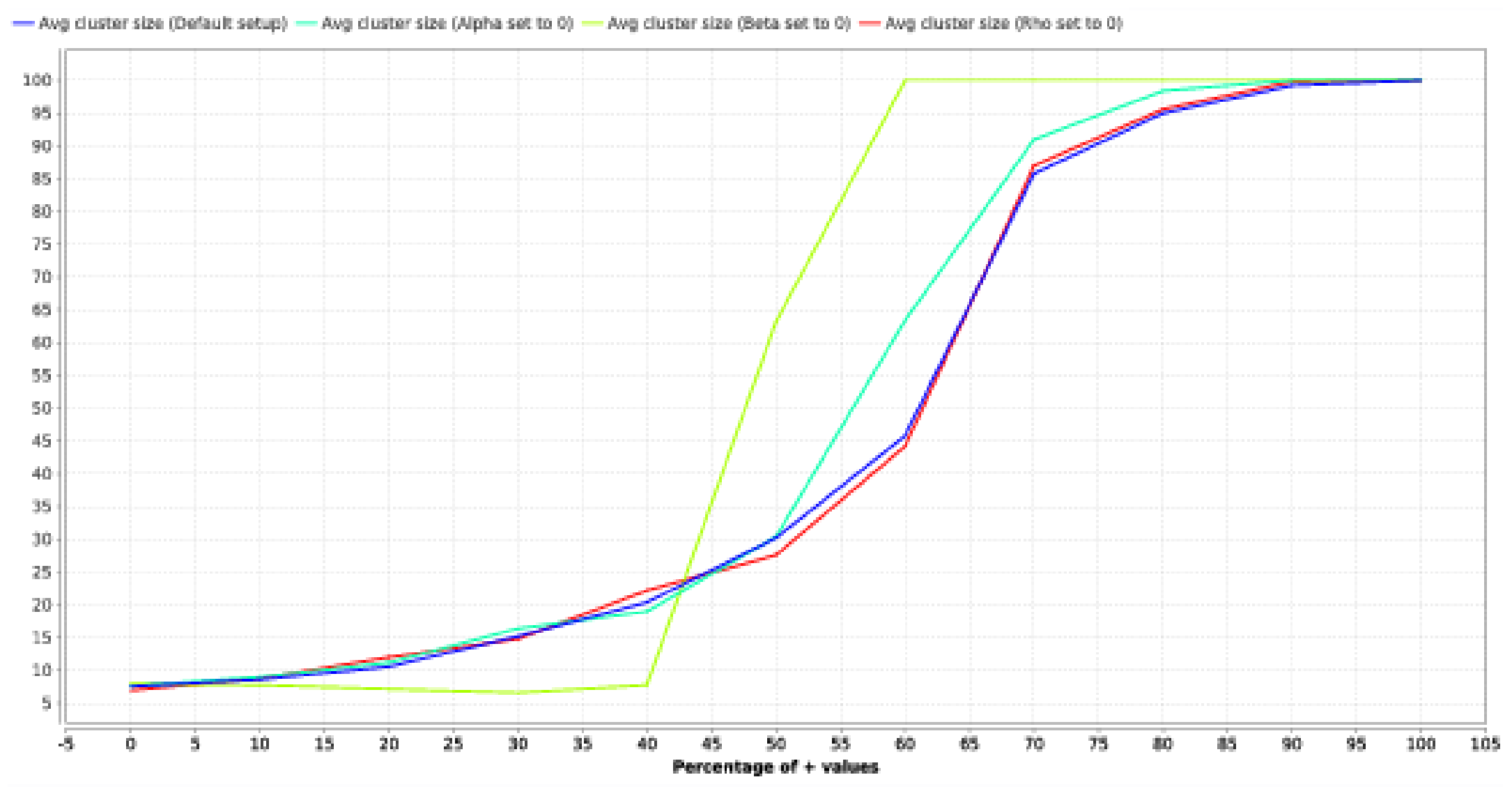

Figure 3: Average size of clusters.

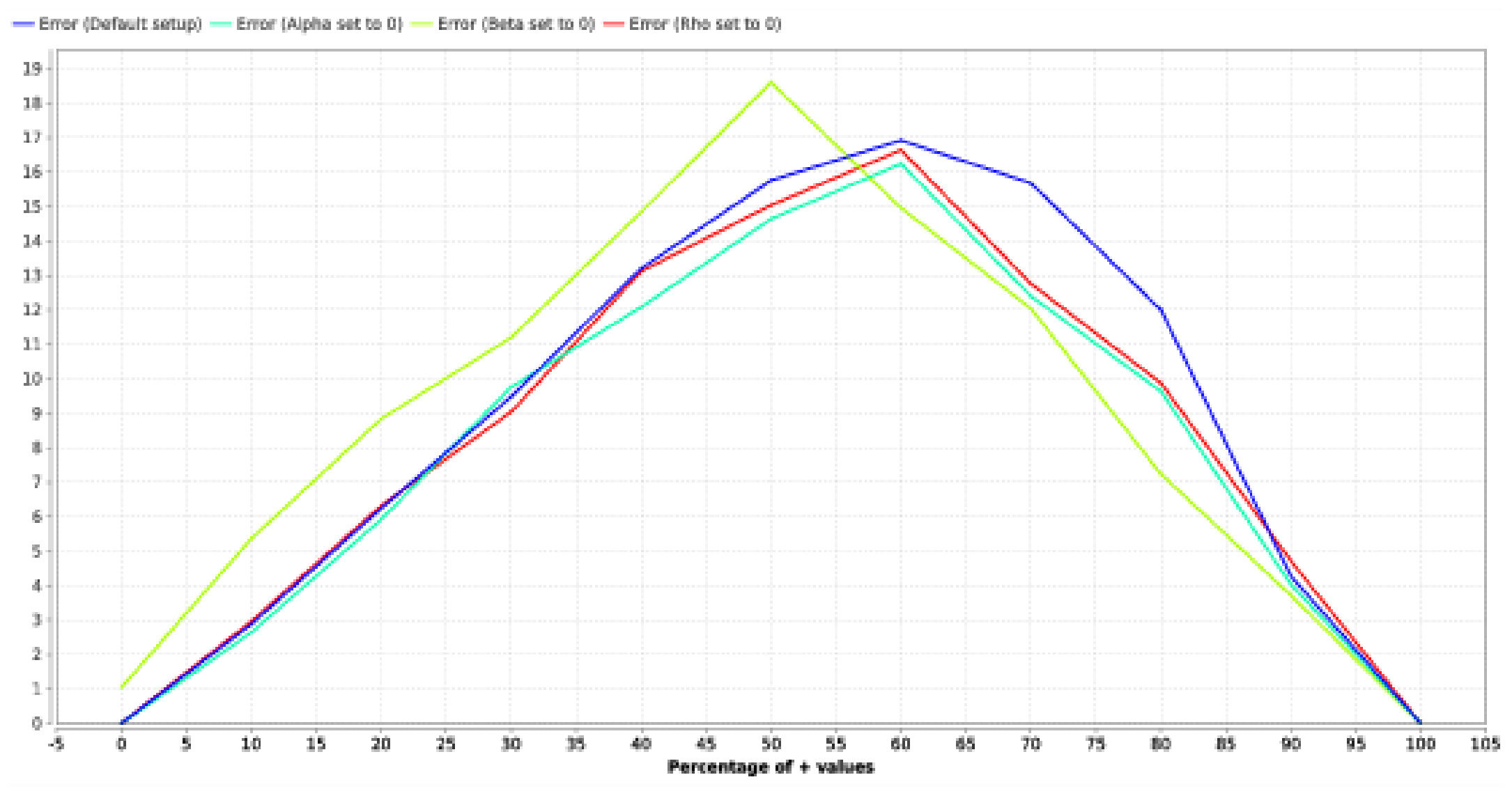

Figure 4: Average error.

## References

[1] Achtert, Elke; Böhm, Christian; Kriegel, Hans-Peter; Kröger, Peer; Zimek, Arthur. *Robust, Complete, and Efficient Correlation Clustering.* SIAM Proceedings, 2007.

[2] Bansal, Nikhil; Blum, Avrim; Chawla, Shuchi. *Correlation clustering.* Machine Learning Journal (Special Issue on Theoretical Advances in Data Clustering), 2004.

[3] Becker, Hila. *A Survey of Correlation Clustering.* COMS E6998: Advanced Topics in Computational Learning Theory, 2005.

[4] Brownlee, Jason. *Clever Algorithms: Nature-Inspired Programming Recipes.* Lulu Enterprises, 2011.

[5] Demaine, Erik D.; Immorlica, Nicole. *Correlation clustering with partial information.* Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2003.

[6] Dorigo, Marco; Maniezzo, V.; Colorni, Alberto. *Positive feedback as a search strategy.* Technical report, Ipartimento di Elettronica, Politecnico di Milano, 1991.

[7] Néda, Z.; Sumi, R.; Ercsey-Ravasz, M.; Varga, M.; Molnár, B. and Cseh, Gy. *Correlation clustering on networks.* Journal of Physics A: Mathematical and Theoretical, 2009.

[8] Liu, Xiaoyong; Fu, Hiu. *An Effective Clustering Algorithm With Ant Colony.* Journal of Computers, vol. 5, no. 4, 2010.

[9] Zimek, Arthur. *Correlation Clustering.* Thesis, Ludwig-Maximilians University, Munich, 2008.

Author: András Fülöp, Faculty of Informatics, University of Debrecen
H-4032 Debrecen, Kassai út 26., Hungary. E-mail: fulibacsi@gmail.com