A Tutorial on Manifold Clustering using Genetic Algorithms

Hector D. Menendez

Department of Computer Science
University College London

2nd September 2015

What's Clustering?

Definition

Group Information (or Data) Blindly.

Blindly?

Clustering: Science or Art?

Ulrike von Luxburg ULRIKE.LUXBURG@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems, Tübingen, Germany

Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
Australian National University and NICTA, Canberra ACT 0200, Australia

Isabelle Guyon ISABELLE@CLOPINET.COM
ClopiNet, 955 Creston Road, Berkeley, CA 94708, USA

What do I need to cluster things?

1. Objects.

What do I need to cluster things?

1. Objects.
2. Criterion / Criteria.

What do I need to cluster things?

1. Objects.
2. Criterion / Criteria.
3. Minimization Process.

The objects

The objects are usually data.

They can be contained in a specific space or not.

The model could have all the objects or receive it in real-time.

The Criterion

It defines the final goal of the grouping process.

If defines similarity/distance among objects.

The Minimization Process

The criterion defines a Cost Function.

This Cost Function is used to guide a minimization process.

The main goal is to minimize the cost.

Clustering Approaches

Partitional: Divides the whole set of objects in Clusters with no overlapping.

Overlapping: Divides the whole set of objects in Clusters with overlapping.

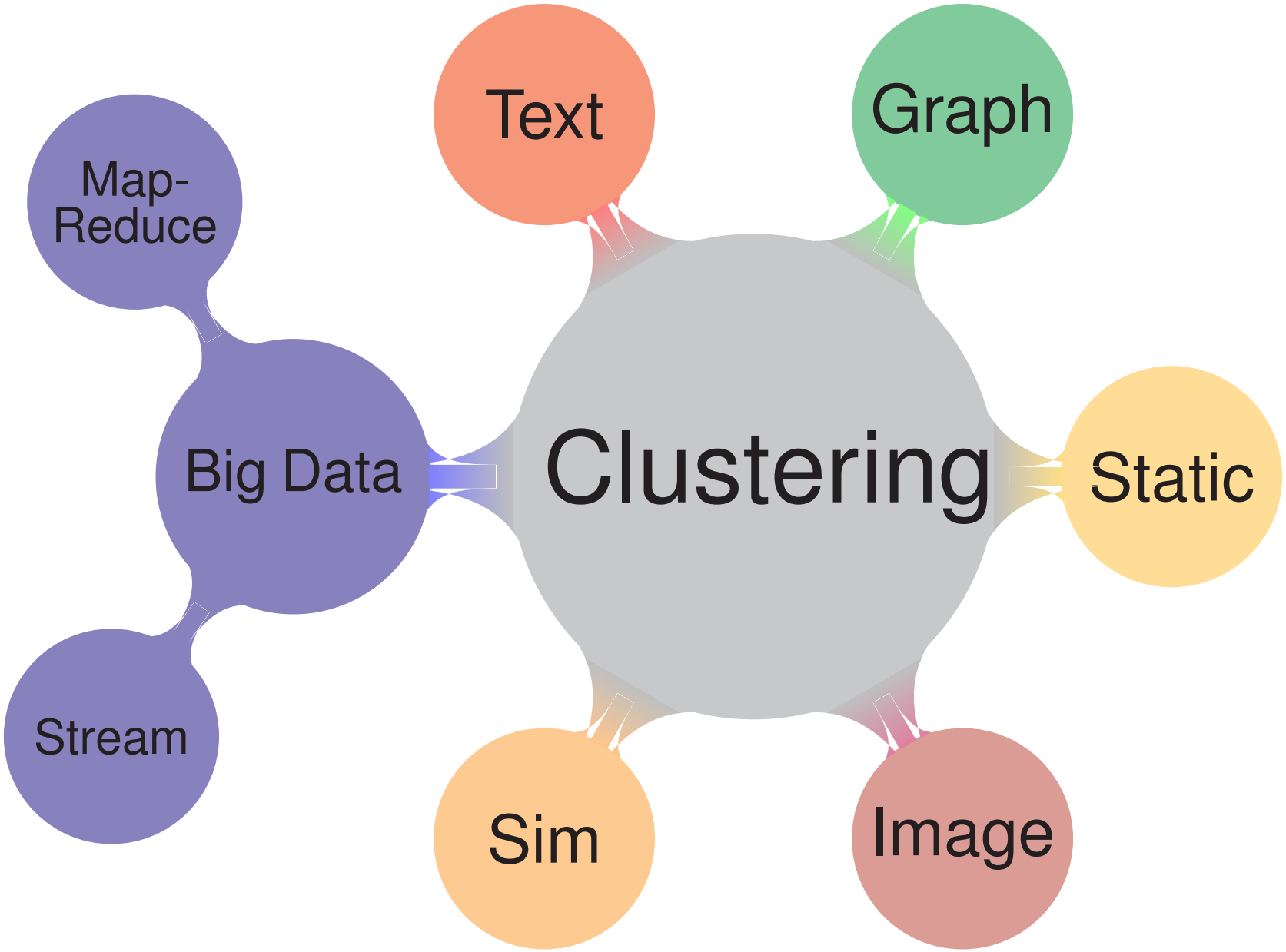
Hierarchical: Nest the clusters by hierarchical levels.

Clustering Approaches

Parametric: The Cost Function is a statistical estimator whose parameters are optimized.

Non-Parametric: There is no statistical estimators during the optimization process.

Hybrid: Combines the previous methods.


Text
Graph
Map-
Reduce
Big Data
Clustering
Static
Stream
Sim
Image

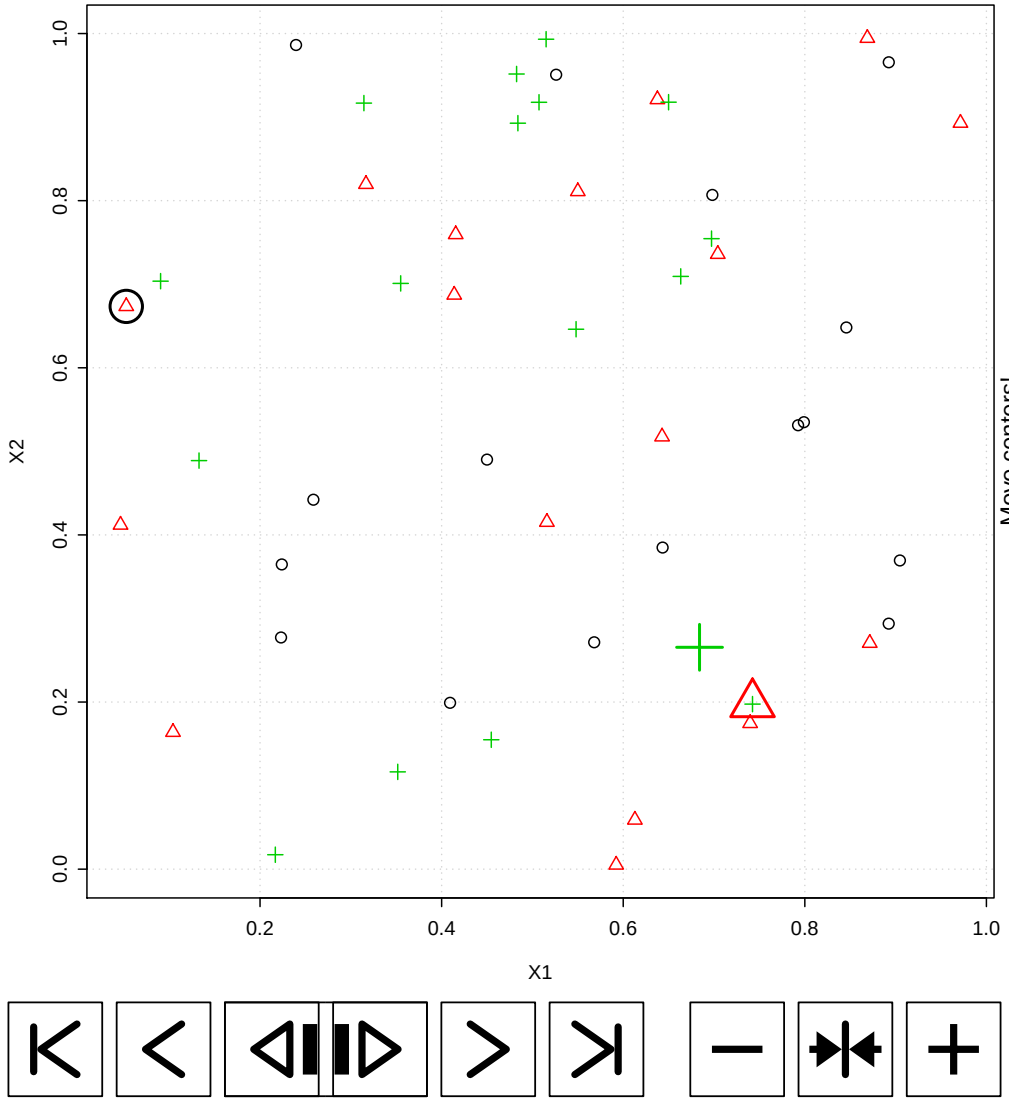
Static Data

Numerical or Categorical Data.

It can be discriminated in an Euclidean Search Space.

The most used algorithm is K-means.

K-means



Similarities

It is possible that we have no idea about the space, but we have a metric to compare objects.

Using this distance/similarity metric we can cluster the objects.

The most frequent approaches are Medoid-based approaches.

PAM

Require: k number of clusters and $D = d_{i,j}$ distance matrix between the elements of $X = x_1, \ldots, x_n$.

Ensure: $m_1, \ldots, m_k$ best k medoids and $C = \{c_1, \ldots, c_k\}$ clusters

Choose random medoids.

while iteration $<$ MaxIterations and !convCriterion() **do**

  for all $x_i \in X$ **do**

    Insert x_i in cluster c_a where m_a is the closest medoid.

  for all $c_a \in C$ **do**

    Calculate medoid $m_a = \text{mín}_i \sum d_{i,j}$ where $x_i, x_j \in c_a$.

Text

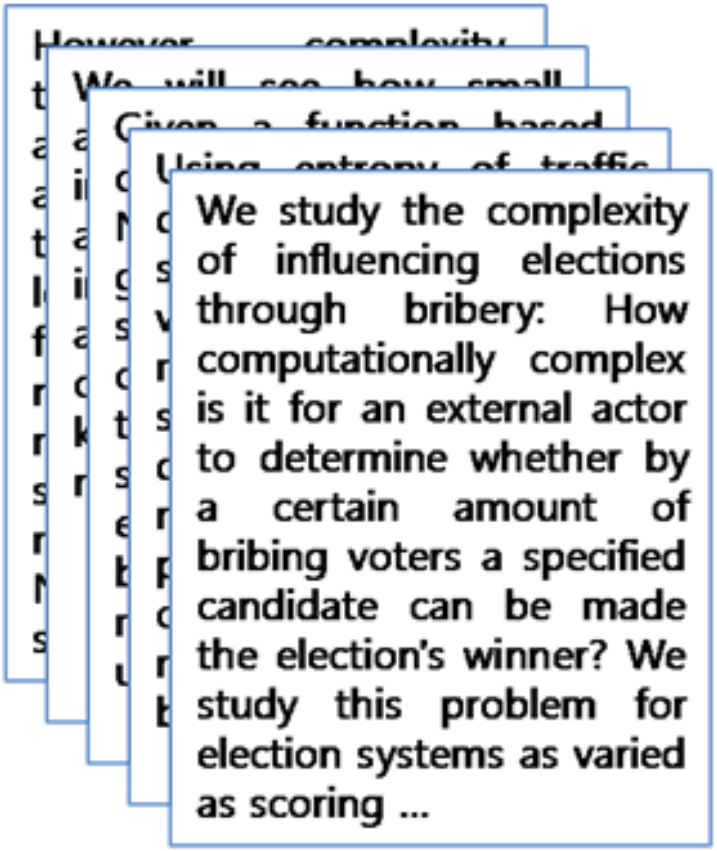
Normally used to detect topics or group documents.

It is combined with NLP techniques.

The approach usually generates a search space or a Similarity matrix.

Term-Document

Documents



Vector-space representation

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

Images

Process of detect objects in Images: Pixels or Regions considered as static data.

Group images by similarity.

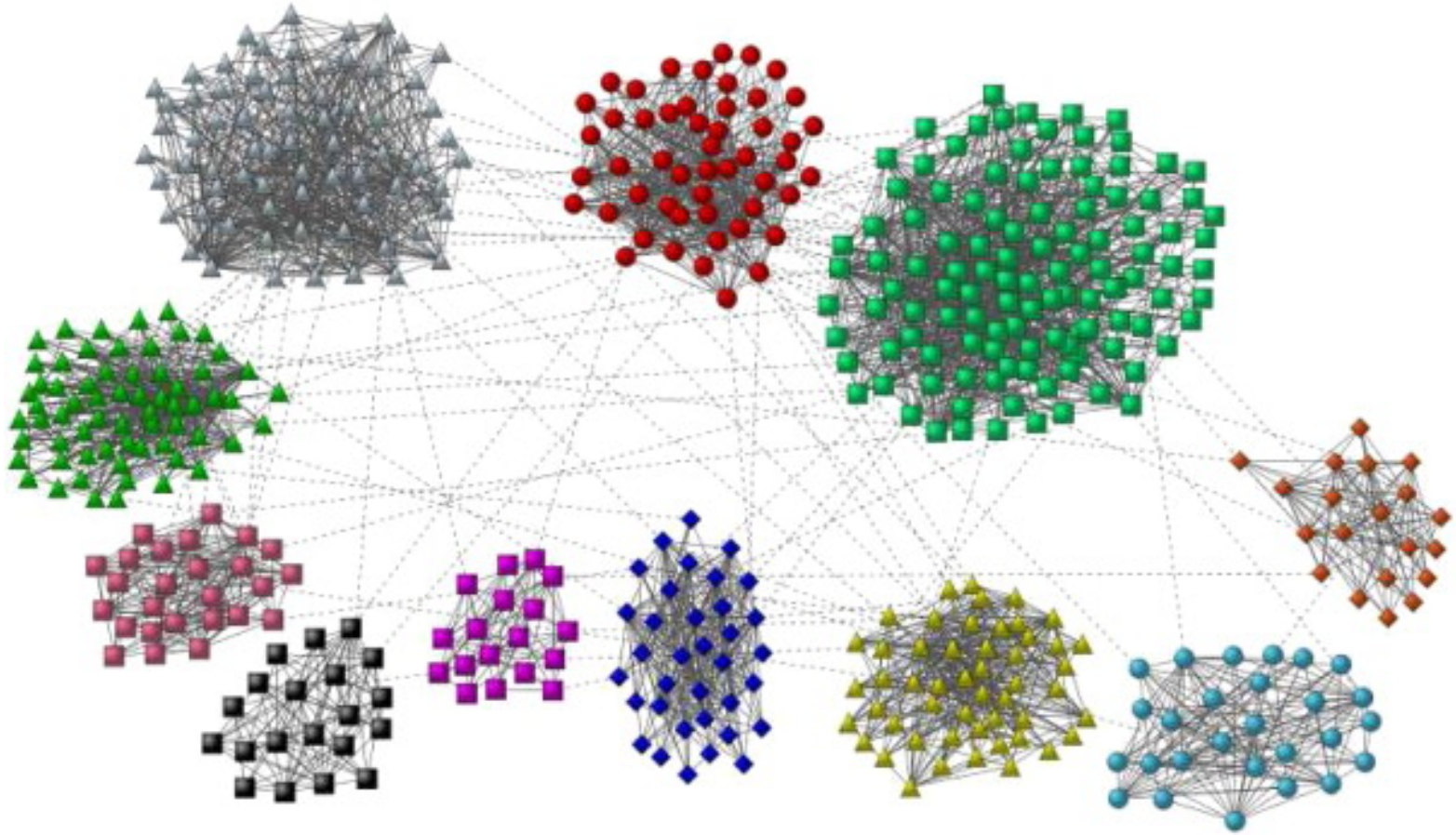
Objects

Graphs

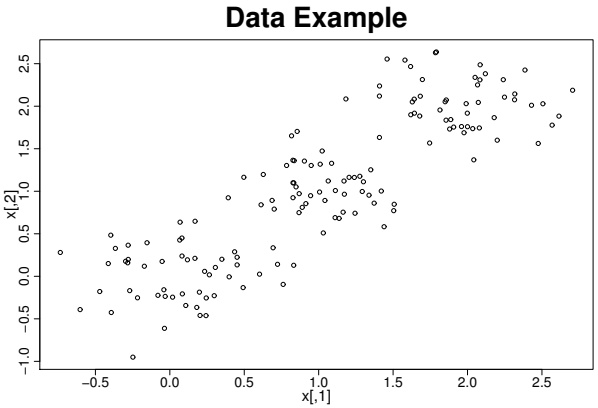
Clustering can be applied to graphs in order to group nodes by similarity: **Community Detection**.

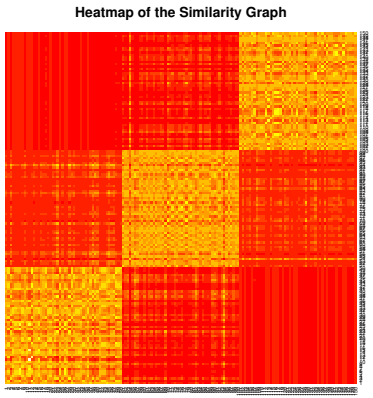
Graph Structures can help to create new types of Clustering: **Spectral Clustering**.

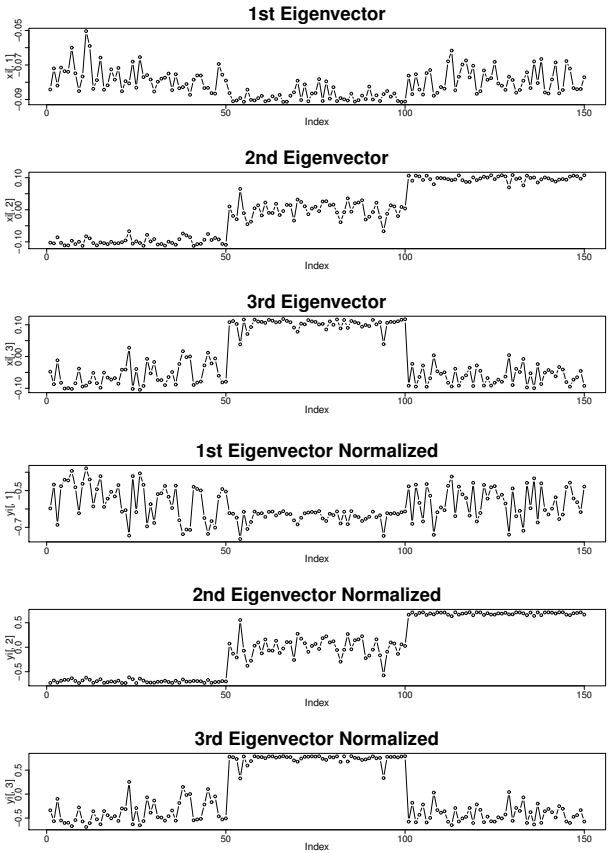
Community Detection

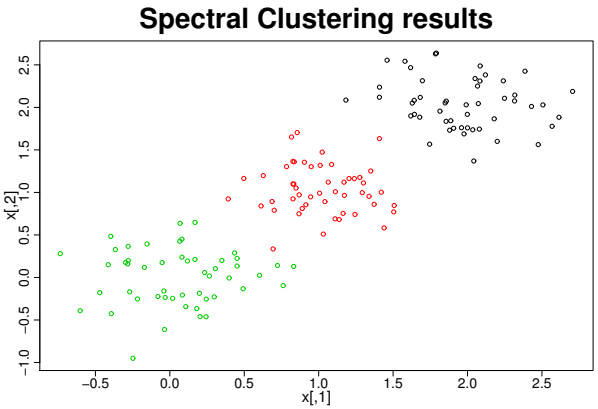


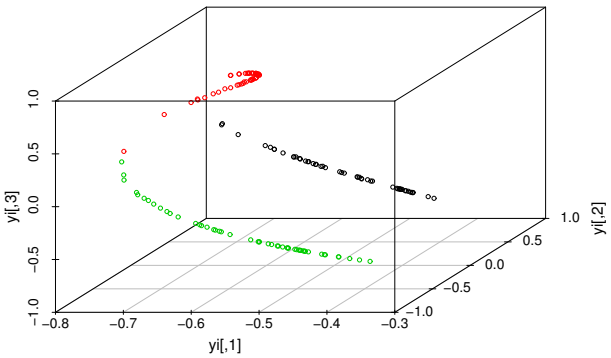
Spectral Clustering











Big Data

The Large Datasets require new strategies to apply these algorithms.

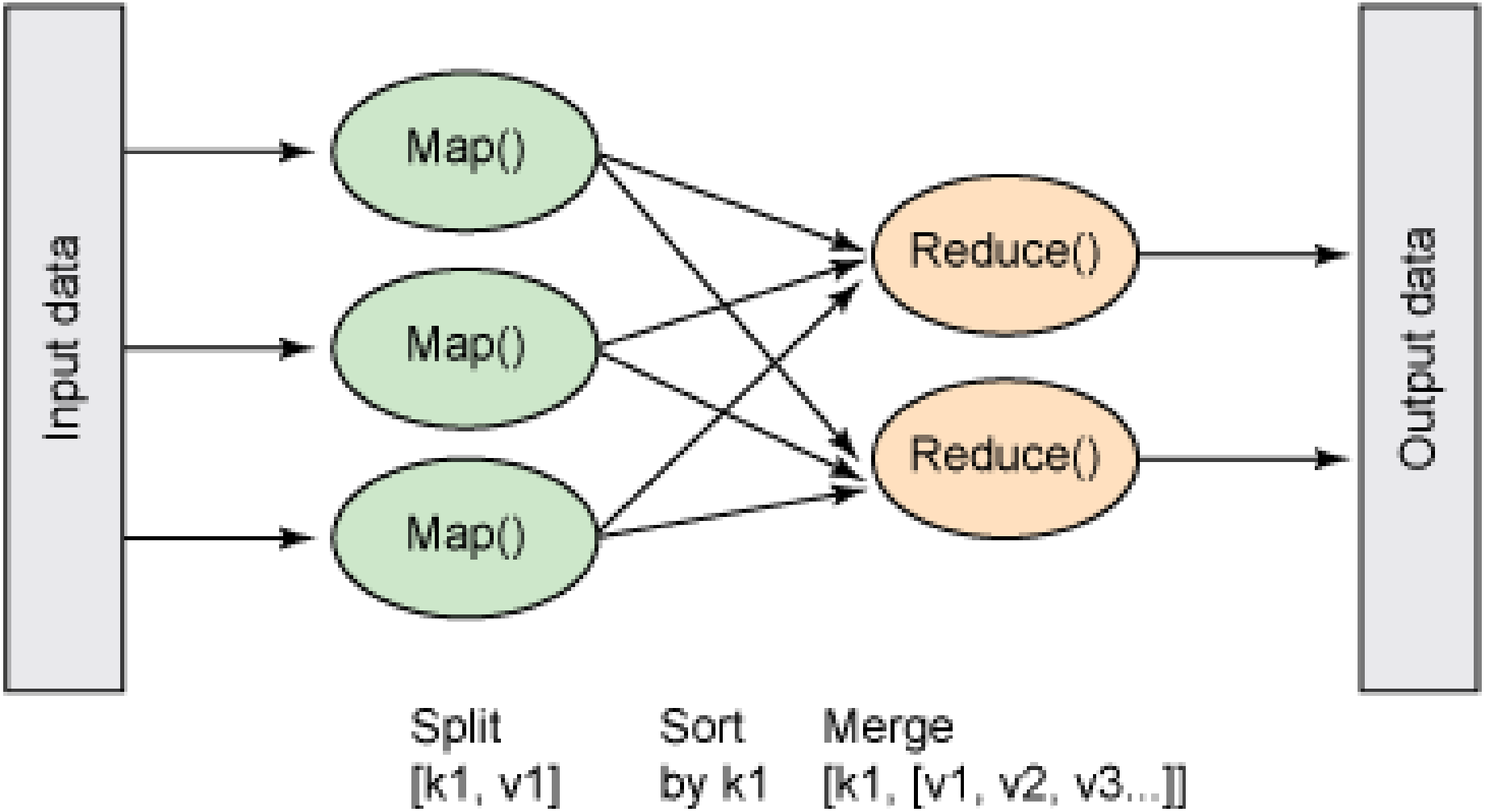
We will focus on two approaches: Large Datasets and Real-Time data.

Large Data

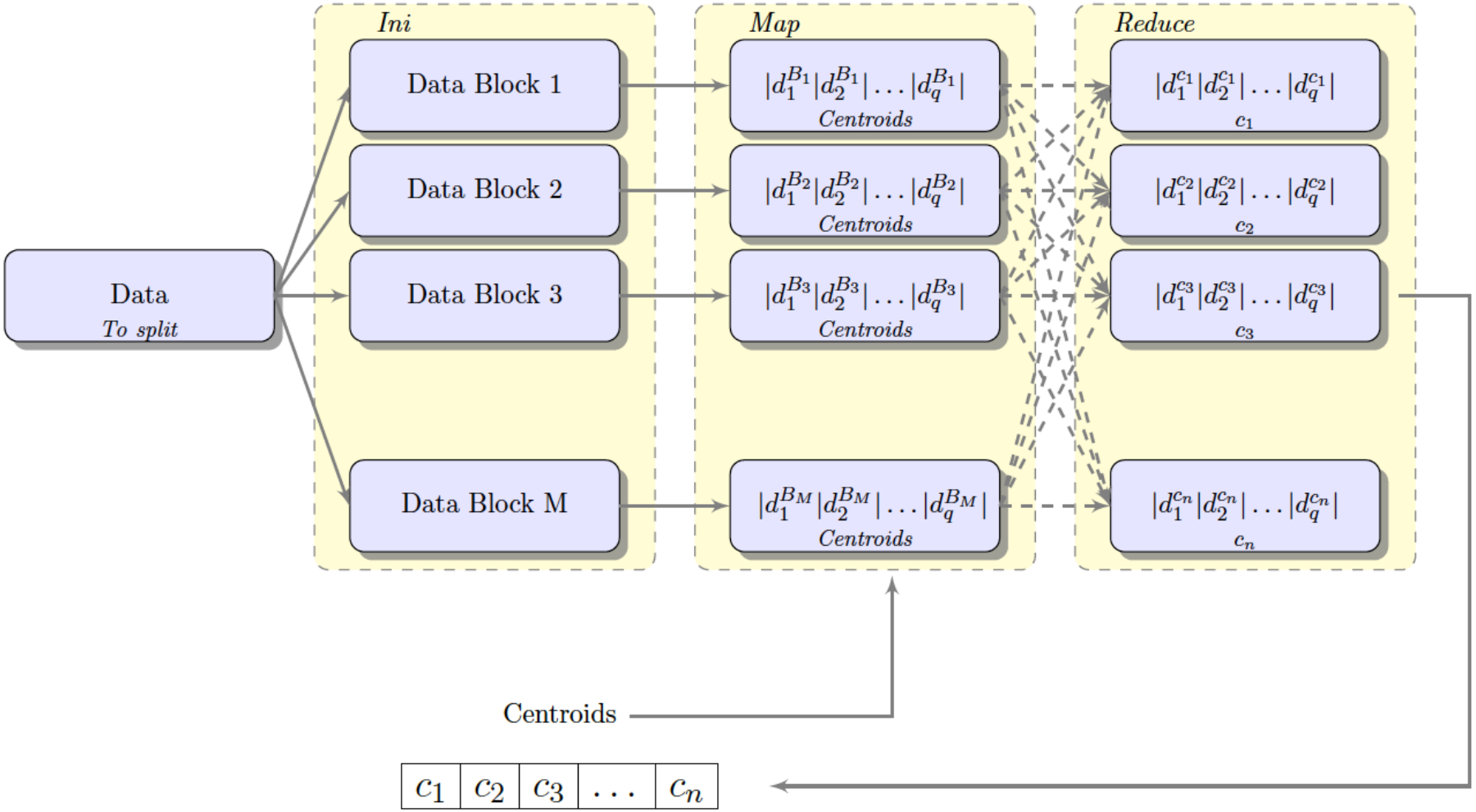
Large data can be processed parallel.

We will focus on two approaches: Large Datasets and Real-Time data.

Map Reduce


Input data
Map()
Map()
Map()
Reduce()
Reduce()
Output data
Split
[k1, v1]
Sort
by k1
Merge
[k1, [v1, v2, v3...]]

K-means Map Reduce



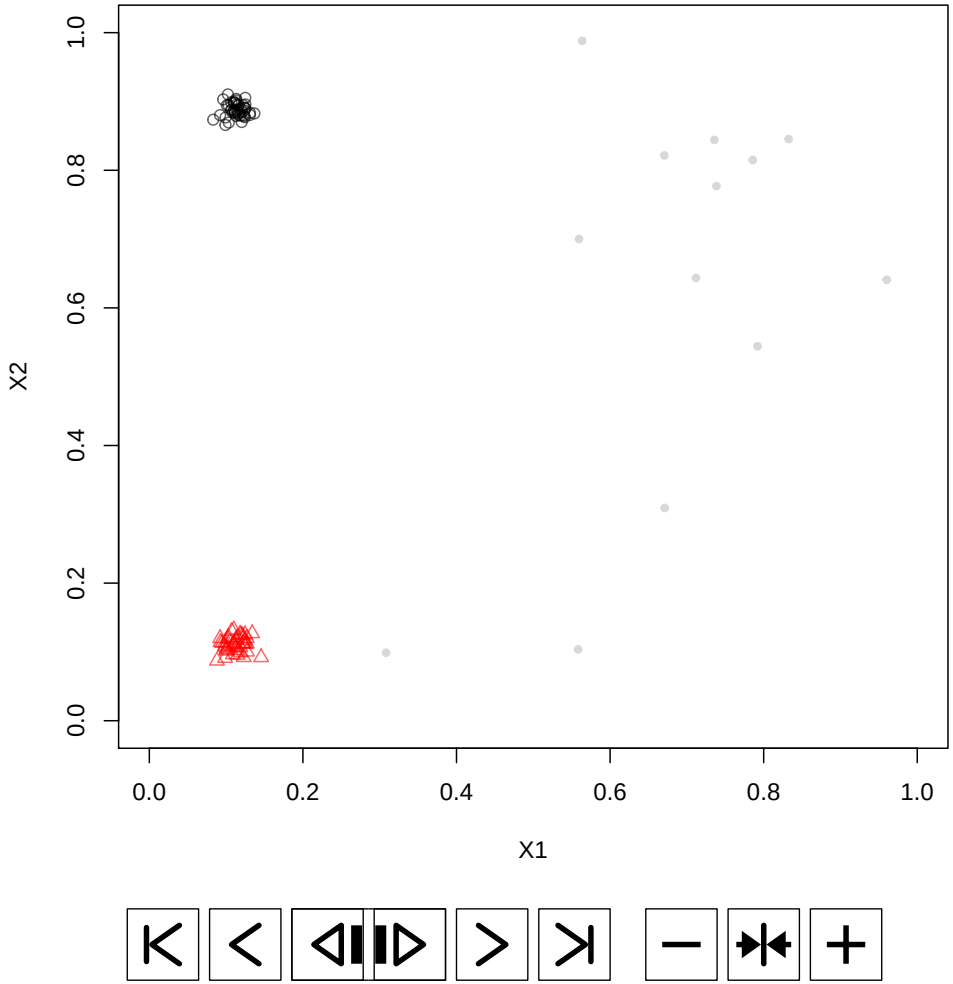
Real-Time Data

Stream data can be analysed in Real-Time using clustering.

The algorithm needs to update its structure with every instance.

These algorithms are fast and keep as less as possible information about past instances.

Stream



Bio-Inspired Computation

Bio-inspired computation is based on natural behaviours.

Some examples are: Particle Swarm Optimization, Ant Colony Optimization, Artificial Bee Colony.

We focus on Evolutionary Algorithms.

Evolutionary Algorithms

Evolutionary Algorithms are based on the evolution of a chromosome population.

These algorithms combine an **encoding**, **operations** and a **fitness** function to guide the search.

There are several different approaches based on evolutionary computation.

Types of Evolutionary Algorithms

Simple Genetic Algorithms.

Multi-Objective Genetic Algorithms.

Evolutionary Strategies.

Island Models.

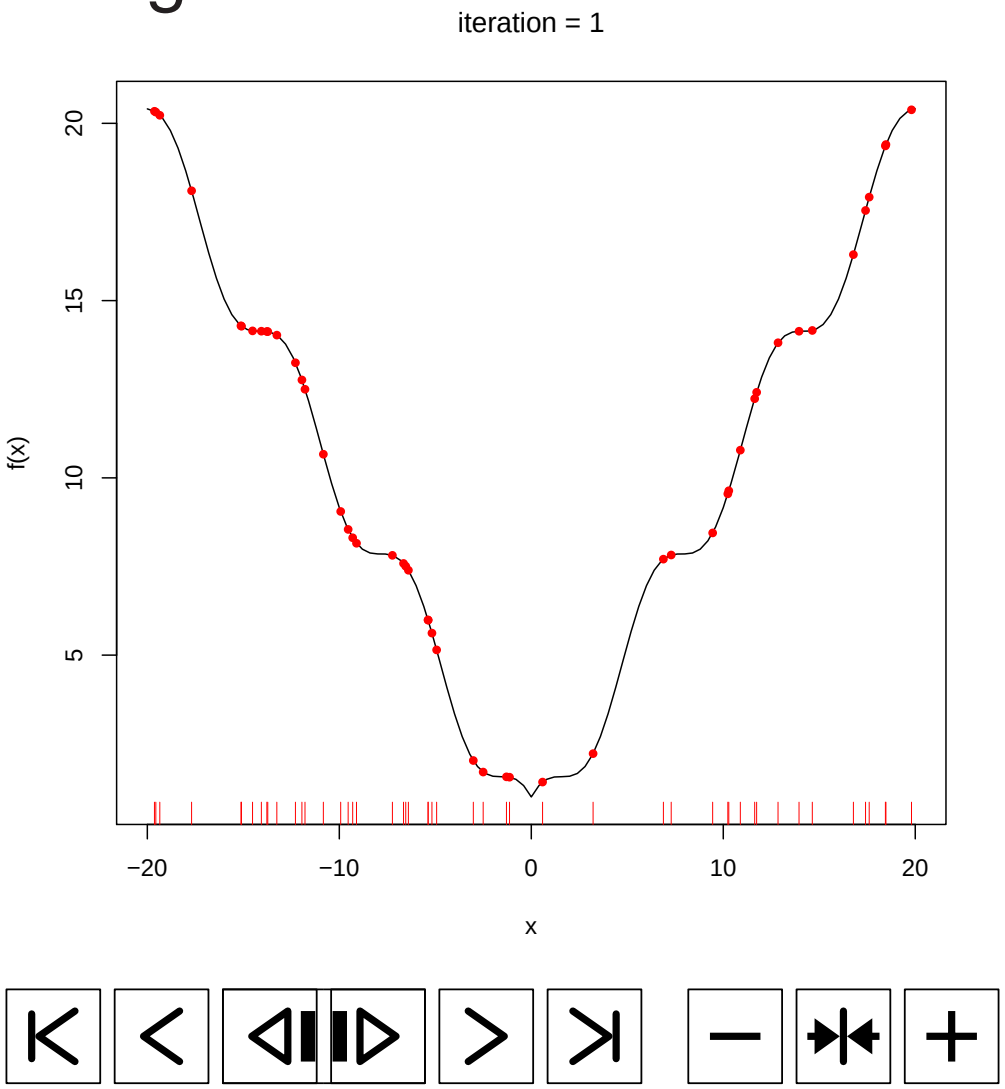
Simple Genetic Algorithms

It is divided in three parts: Encoding, Operations and Fitness.

They are applied for optimization problems.

The idea is to encode a set of possible solutions as the population and evolve them in order to find the best solution according to the fitness function.

Simple Genetic Algorithms



Multi-Objective Genetic Algorithms

Different objectives are combined during the search.

A Pareto Front of solutions is generated.

Evolutionary Strategies

The chromosomes evolves according to the position in the search space.

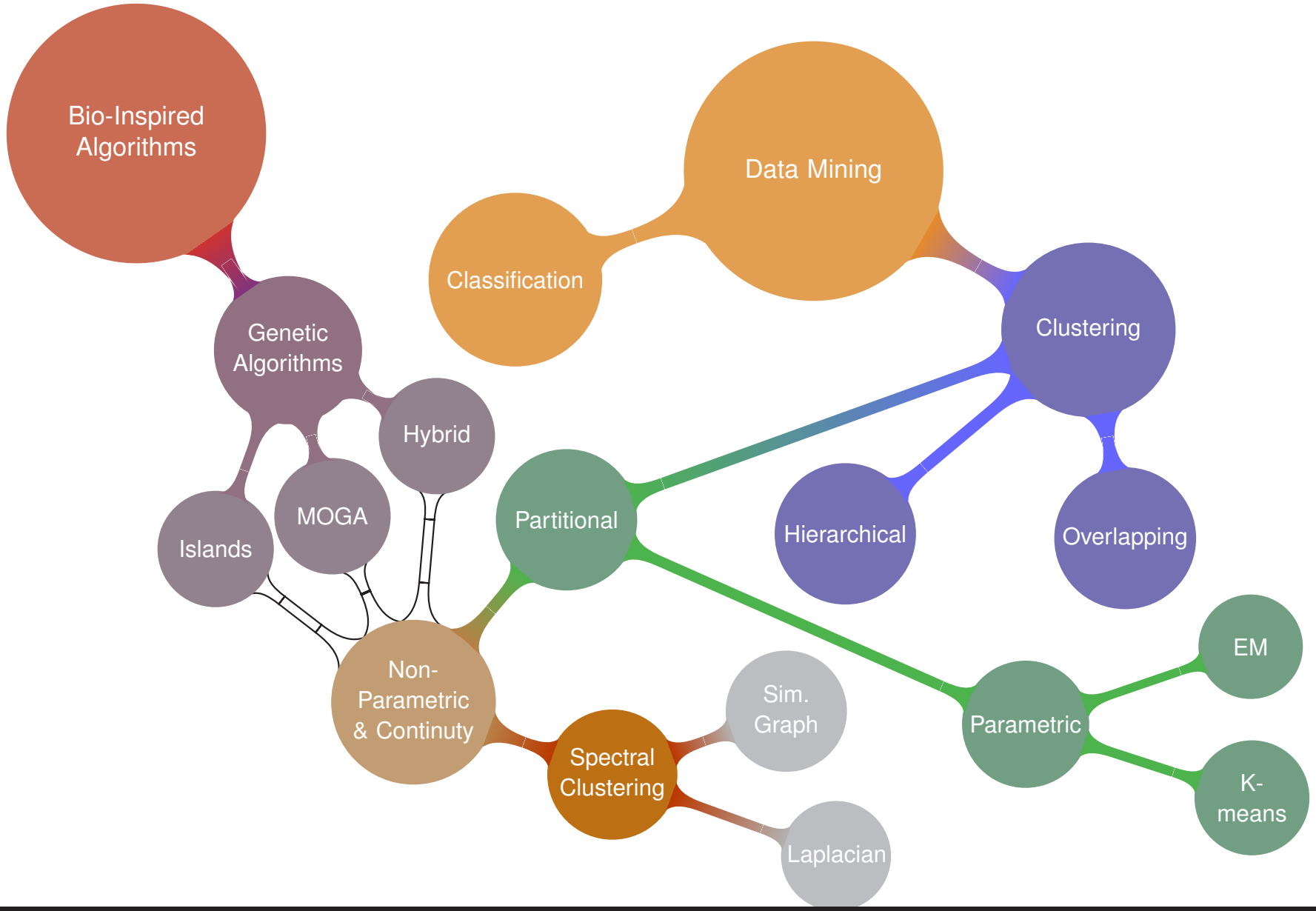
The evolutionary process is also adaptive.

Island-based Algorithms

Create islands that share information in two steps.

Each island performs a micro search.

The general model performs a macro search.


Bio-Inspired Algorithms
Data Mining
Classification
Clustering
Genetic Algorithms
Hybrid
MOGA
Islands
Partitional
Hierarchical
Overlapping
EM
Non-Parametric & Continuty
Sim. Graph
Parametric
Spectral Clustering
K-means
Laplacian

GAs in Manifold Identification

Create a Genetic Algorithm for Manifold Identification.

Compare it with Spectral Clustering.

Encodings

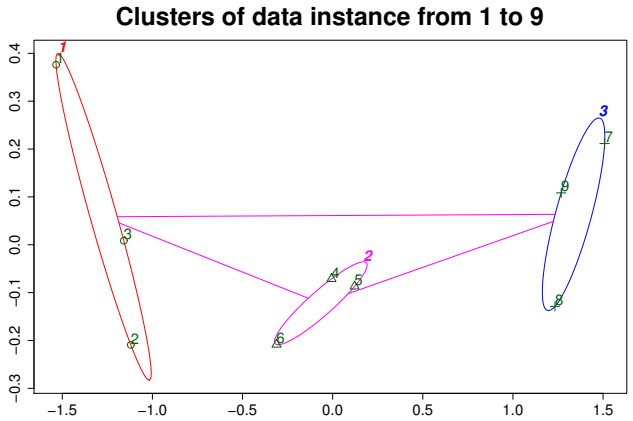
nodes

	1	2	3	4	5	6	7	8	9
Chromosome 1	1	1	1	2	2	2	3	3	3

(a) Label-based

	Cluster 1	Cluster 2	Cluster 3
Chromosome 1	$\{1, 2, 3\}$	$\{4, 5, 6\}$	$\{7, 8, 9\}$

(b) Medoid-based



Operations: Selection y Reproduction

Selection: $(\mu + \lambda)$.

Reproduction: Tournament selection of individuals.

Operations: Crossover

	nodes 1	2	3	4	5	6	7	8	9
Chromosome 1	1	1	1	**2**	**2**	**2**	**3**	3	3
Chromosome 2	1	2	1	**2**	**3**	**2**	**3**	3	3
New Chromosome 1	1	1	1	**2**	**3**	**2**	**3**	3	3
New Chromosome 2	1	2	1	**2**	**2**	**2**	**3**	3	3

(a) Crossover Label-based

	Clusters 1	Clusters 2	Clusters 3
Chromosome 1	$\{1, 2, 3\}$	$\{4, 5, 6\}$	$\{7, 8, 9\}$
Chromosome 2	$\{2, 3\}$	$\{5, 6, 7\}$	$\{1, 4, 8, 9\}$
Intersection	$\{2, 3\}$	$\{5, 6\}$	$\{8, 9\}$
New Chromosome 1	$\{2, 3, \mathbf{4}\}$	$\{5, 6, \mathbf{1}\}$	$\{8, 9, \mathbf{7}\}$
New Chromosome 2	$\{2, 3, \mathbf{1}, \mathbf{7}\}$	$\{5, 6, \mathbf{4}\}$	$\{8, 9\}$

(b) Crossover Medoid-based

Operations: Mutation

Mutation: It is **adaptive**. The mutation probability depends on their assignation probability to the current cluster. For each encoding there are different mutations:

Label-based: Randomly modifies the value of an allele.

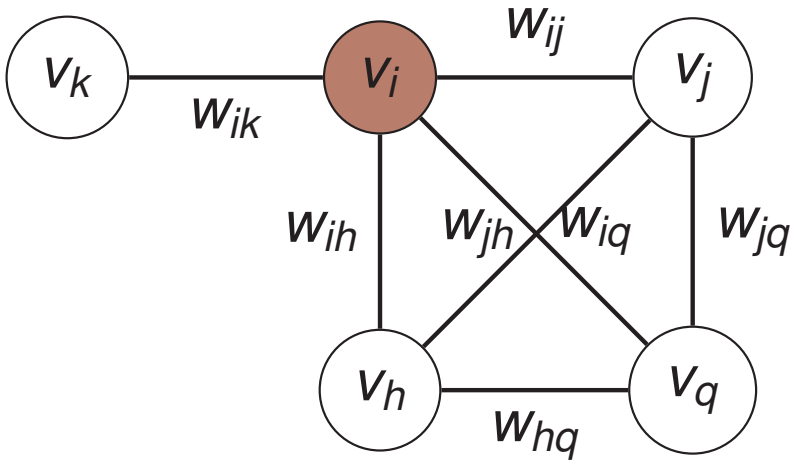
Medoid-based: Moves an allele from a cluster to another.

Operations: Mutation example

	nodes								
	1	2	3	4	5	6	7	8	9
Chromosome	**1**	1	1	2	2	2	**3**	3	3
	↓						↓		
Chromosome Mutated	**2**	1	1	2	2	2	**2**	3	3

	Cluster 1	Cluster 2	Cluster 3
Chromosome	$\{1, 2, \mathbf{3}\}$	$\{4, 5, 6\}$	$\{7, \mathbf{8}, 9\}$
Chromosome Mutated	$\{1, 2\}$	$\{4, 5, 6, \mathbf{8}\}$	$\{7, 9, \mathbf{3}\}$

Fitness Function: Weight Clustering Coefficient



$$C_i^w = \frac{\sum_{j,h} \frac{w_{ij}+w_{ih}}{2} a_{ij} a_{ih} a_{jh}}{S_i(k_i - 1)}$$

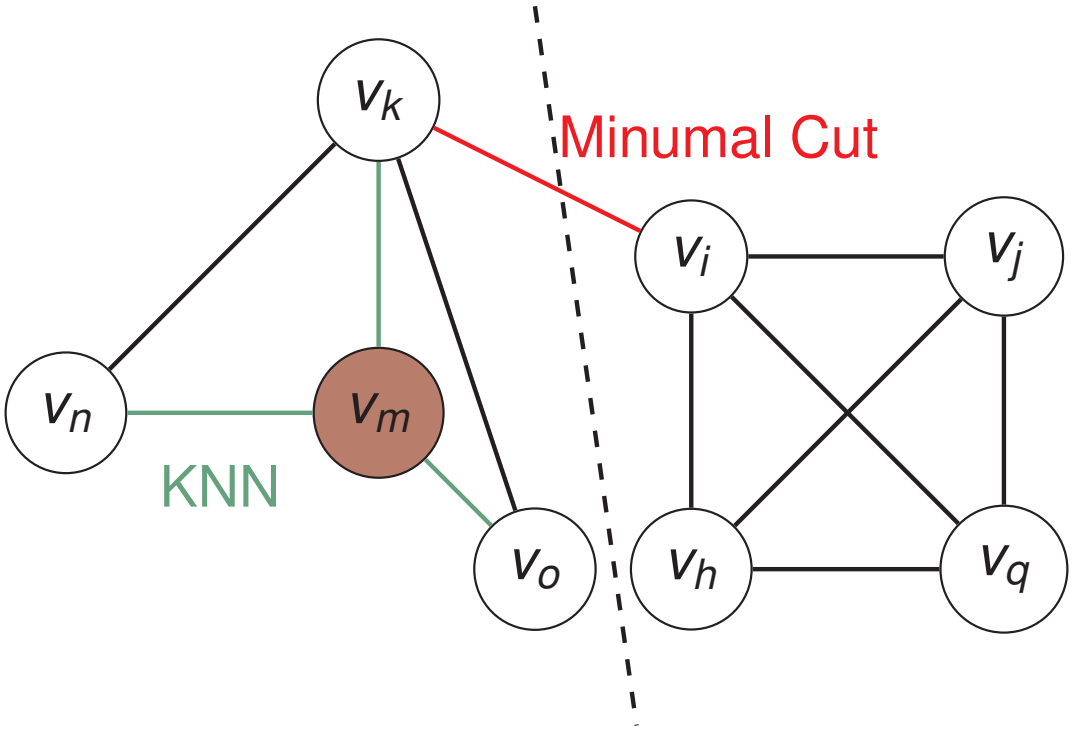
La Función Fitness: KNN-Minimal Cut

Combines K-Nearest Neighbourhood (KNN) and Minimal Cut.

KNN guarantees the data continuity.

MinCut guarantees the clustering separation.

Fitness Function: KNN-Minimal Cut



$$KNN = \sum_{x \in C} \frac{|\{y | y \in \Gamma(x) \wedge y \in C_x\}|}{|\Gamma(x)|} \quad MC = \sum_{x \in C} \frac{\sum_{y \notin C_x} w_{xy}}{|\{y | y \notin C_x\}|}$$

$$\frac{KNN}{|C|} \times \left(1 - \frac{MC}{|C|}\right)$$

The robustness problem

GCC is focus on improving the **robustness**.

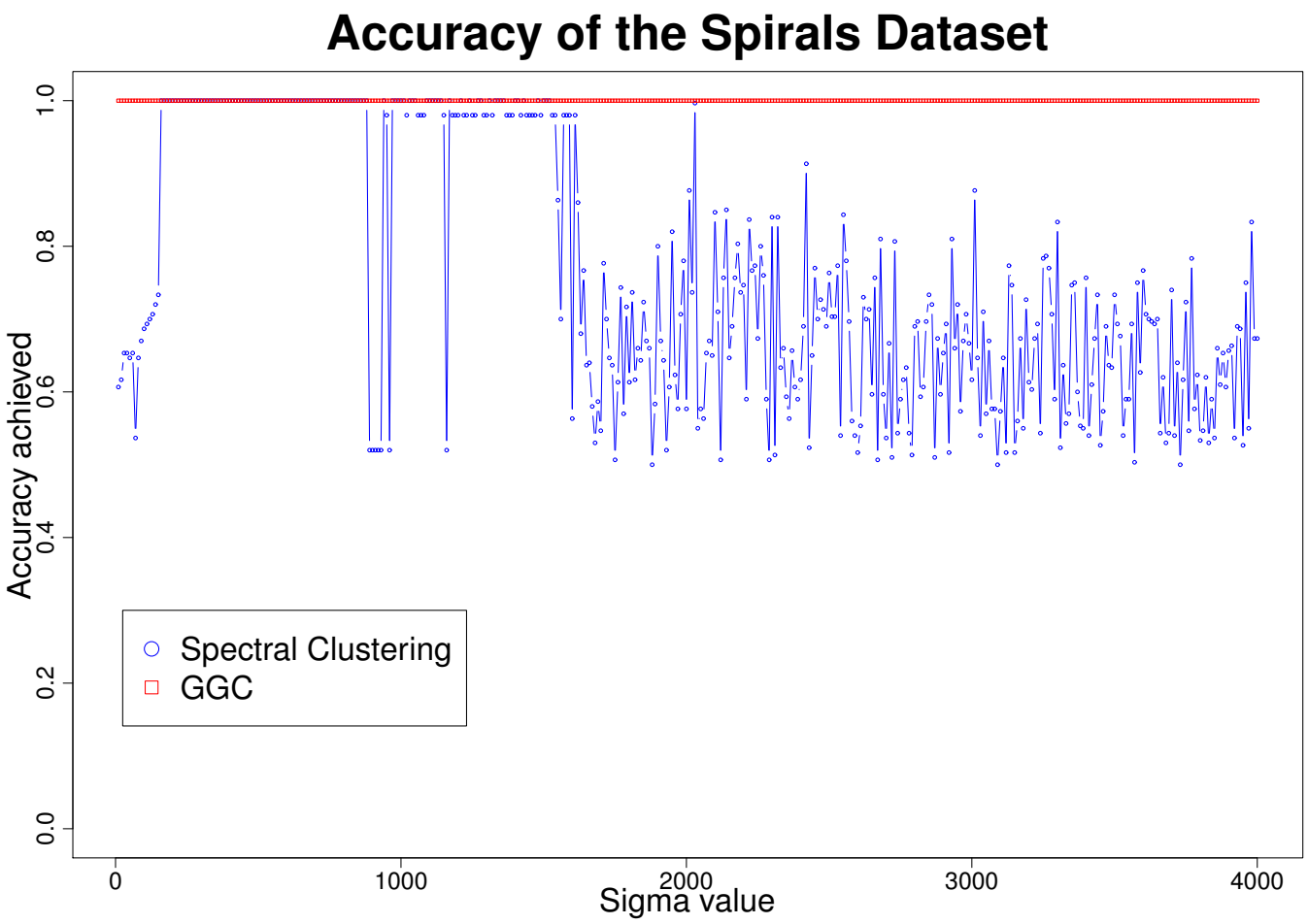
Given three objects: o_1, o_2, o_3, and their distances $d_{12}^{M_1}$, $d_{23}^{M_1}$, $d_{13}^{M_1}$, $d_{12}^{M_2}$, $d_{23}^{M_2}$, $d_{13}^{M_2}$ where M_1 y M_2 are metrics, when they satisfy:

$$d_{12}^{M_1} < d_{23}^{M_1} < d_{13}^{M_1}$$

$$d_{12}^{M_2} < d_{23}^{M_2} < d_{13}^{M_2}$$

The the clustering algorithm results have to be the same (except for 0 or infinite distance)

Robustness of SC and GGC



Less Memory, More Accuracy

Use multi-objective search to improve the results.

Reduce the memory usage reduceing the Similarity graph.

MOGGC Motivation

GGC shows some **problems** according to the **Memory Usage**.

The **convergence** of GGC can also be **improved** through a MOGA algorithm.

The algorithm can employ other graph representation to obtain the same results.

MOGGC Algorithm

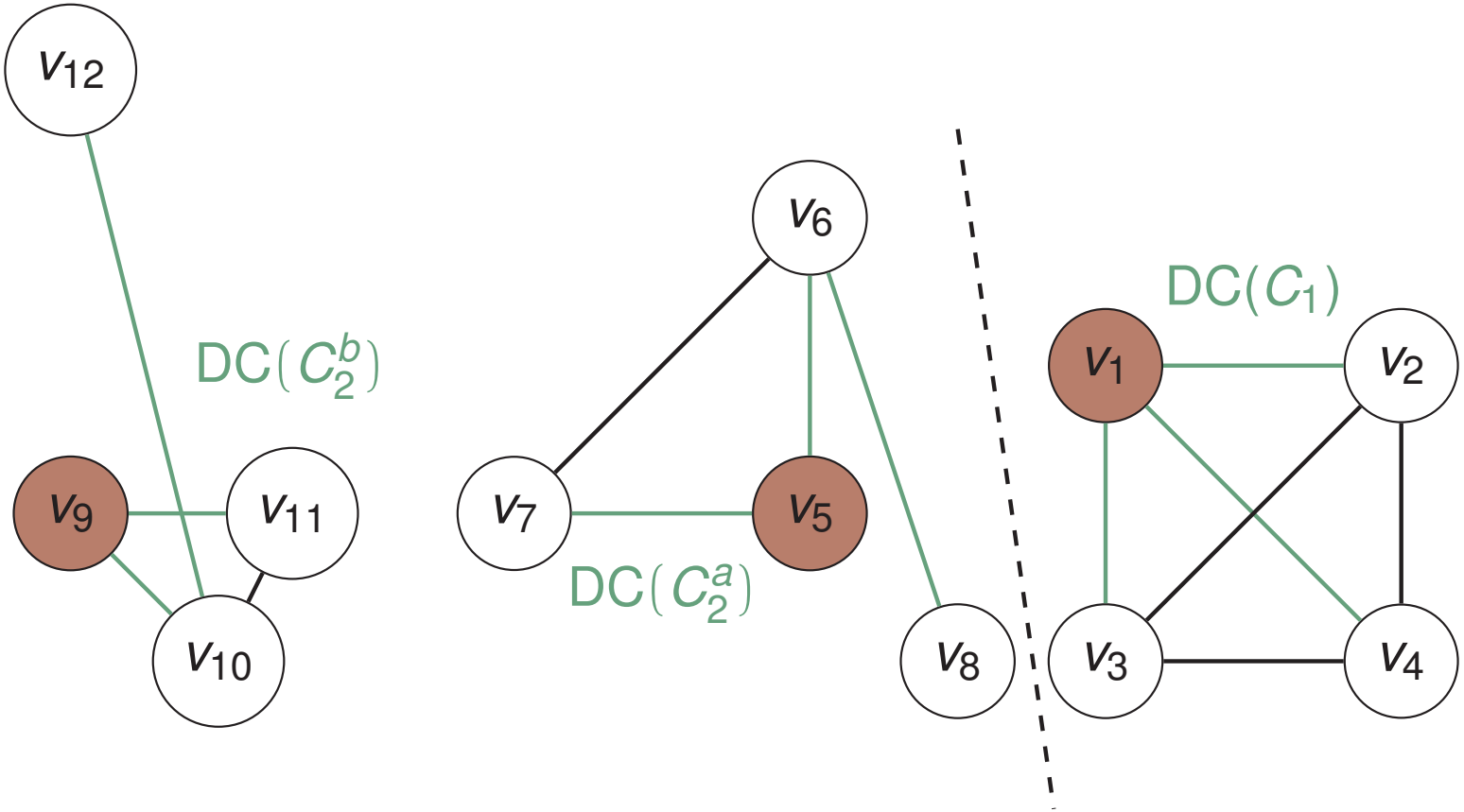
It uses the **same initial parameters** than GGC and the same encoding.

The MOGA algorithm used to guide the algorithm is **SPEA2**.

It used a **K-Similarity Graph** which is a KxN matrix where K is constant and N is the number of elements. It **grows linearly** instead of exponentially.

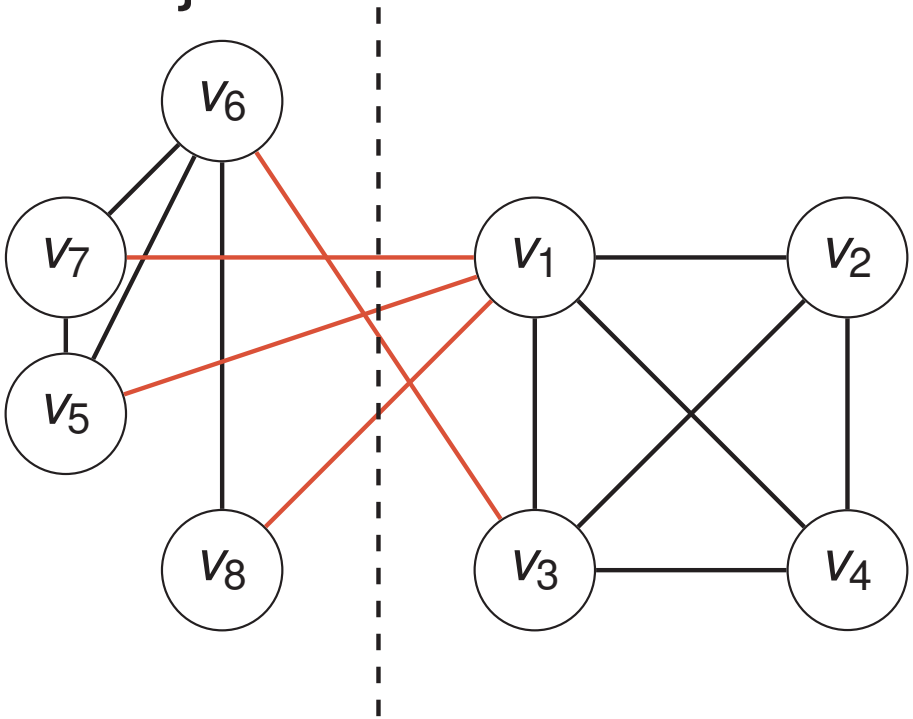
The fitness functions have been modified to improve the data continuity and **cluster separation**.

MOGGC Fitness Objectives: DC



Data Continuity Degree: It calculates the total edges sum for each minimal spanning tree of each connected component of the K-Graph.

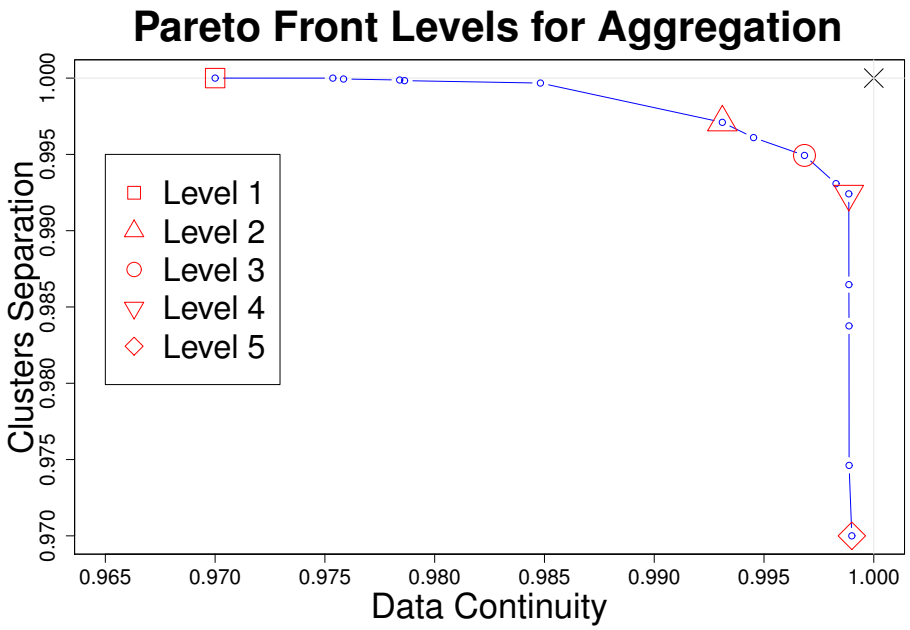
MOGGC Fitness Objectives: CS

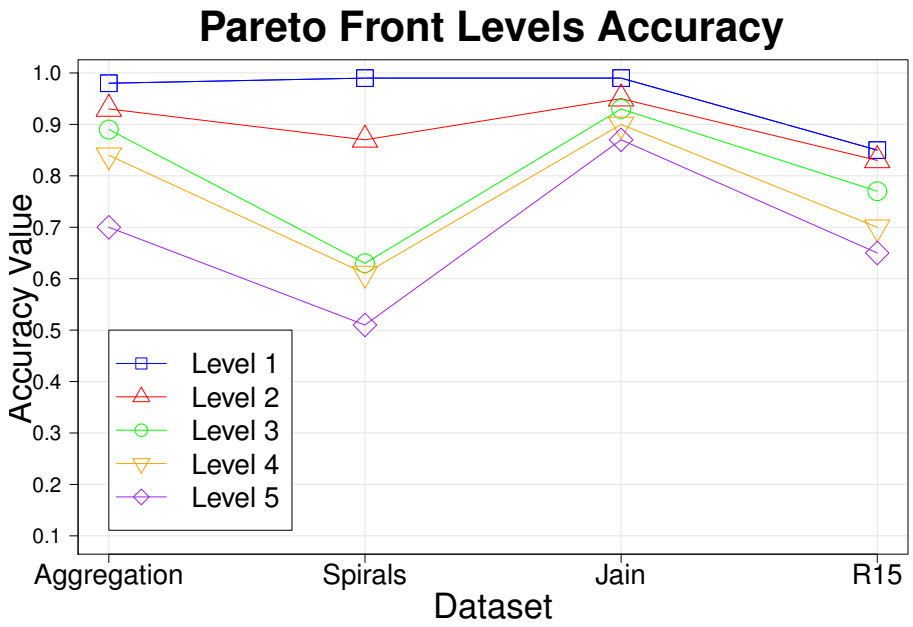


$$\frac{\sum_{v_i \in C} \frac{\sum_{v_j \in G}\{w_{ij} |\ v_j \notin C\}}{|G| - |C|}}{|C|}$$

Clusters Separation: It calculates the arithmetic average value of the edge weights between the different clusters.

MOGGC Fitness Objectives: Pareto





Find the Best K

Include the Number of Clusters in the Search.

Goal: Find the best number of Manifolds.

CEMOG Motivation

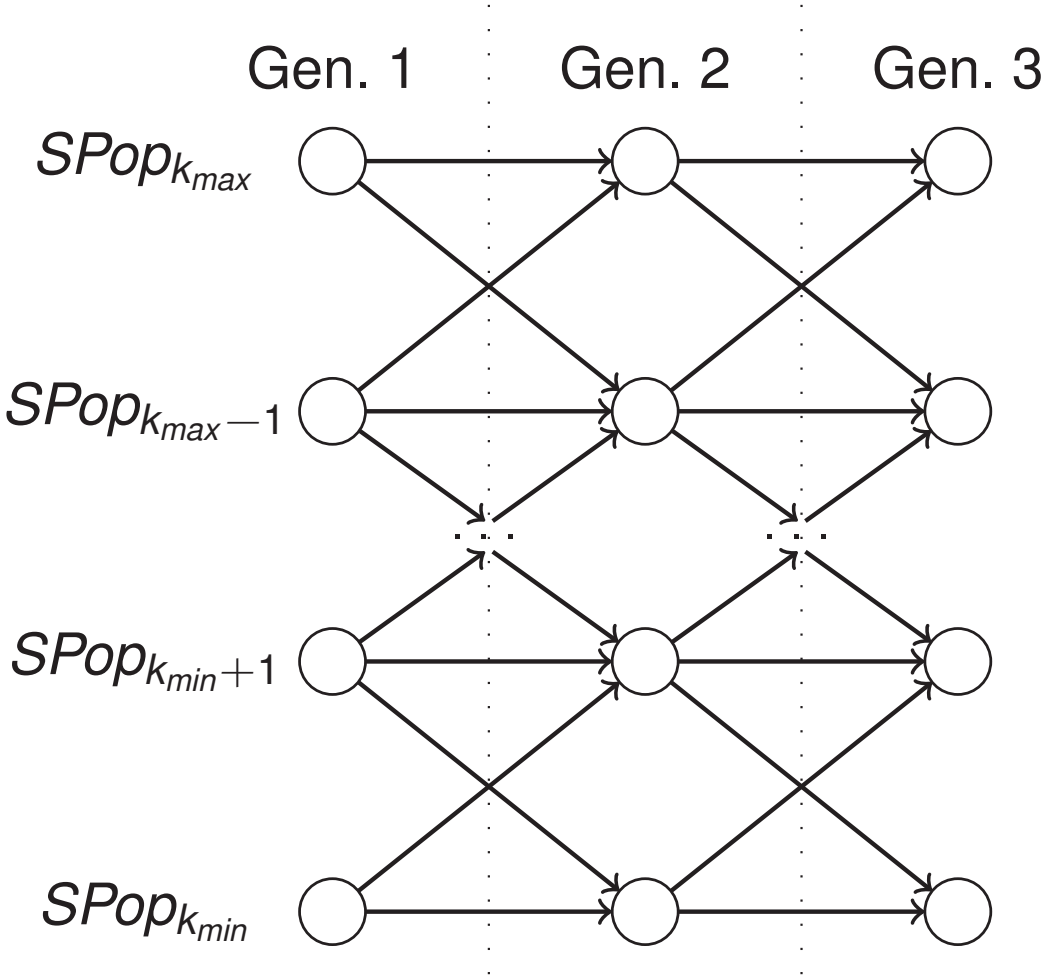
CEMOG was designed in order to include the number of clusters in the genetic search.

The algorithm is based on Co-Evolutionary algorithm.

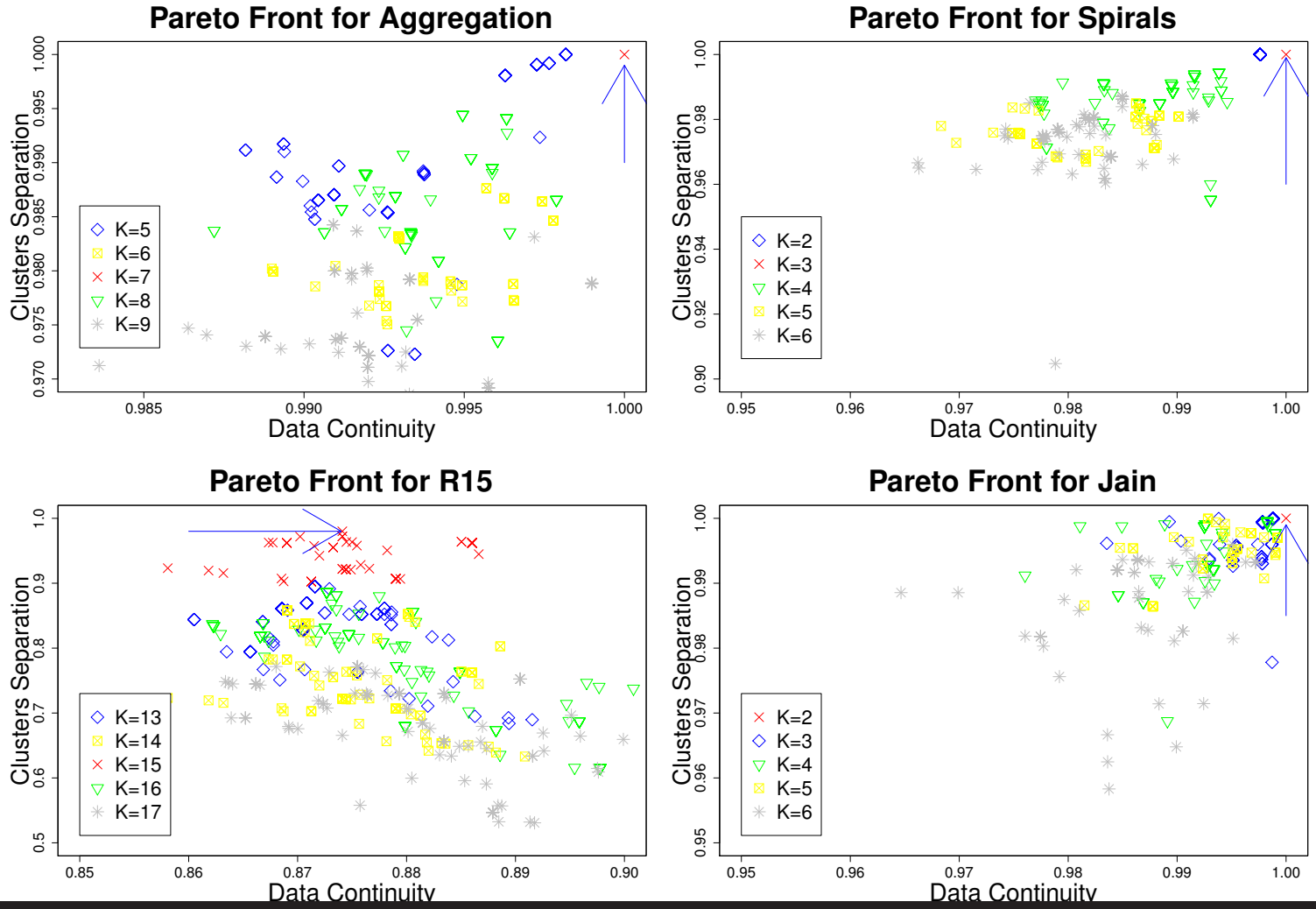
It is mainly divided in two kind of evolutions:

- Macro-evolution: the **evolution** of the different islands **as a whole**.
- Micro-evolution: the **evolution of each island** as an independent population. It is performed **through MOGGC**.

CEMOG Algorithm: Macro-Evolution



Pareto


Pareto Front for Aggregation
Clusters Separation
Data Continuity
K=5
K=6
K=7
K=8
K=9
Pareto Front for Spirals
Clusters Separation
Data Continuity
K=2
K=3
K=4
K=5
K=6
Pareto Front for R15
Clusters Separation
Data Continuity
K=13
K=14
K=15
K=16
K=17
Pareto Front for Jain
Clusters Separation
Data Continuity
K=2
K=3
K=4
K=5
K=6

Large Data: Sub-sampling

Choose relevant points to estimate the model.

Goal: improve the performance of the whole analysis.

Nyström Method and GANY

$$W = \begin{pmatrix} w_{1,1} & \dots & w_{1,N} \\ \vdots & \ddots & \vdots \\ w_{N,1} & \dots & w_{N,N} \end{pmatrix}$$

Full-Matrix

$$W = \begin{pmatrix} w_{1,1} & \dots & w_{1,N} \\ \vdots & \ddots & \vdots \\ w_{N,1} & \dots & w_{N,N} \end{pmatrix} \xrightarrow{\text{Eigenvectors Extraction}} UU^T$$

Equivalent Eigenvectors

n samples

$$(A|B) = \left(\begin{array}{ccc|ccc} a_{1,1} & \dots & a_{1,n} & b_{1,1} & \dots & b_{1,N-n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} & b_{n,1} & \dots & b_{n,N-n} \end{array}\right) \xrightarrow[\text{approximation}]{\text{Nyström extension}} \mathcal{U}\mathcal{U}^T$$

GANY Algorithm

0.1	0.5	0.1	-0.7	-0.1	-0.6	-0.8	0.6	0.6

Initialization: **Uniform** initialization. The **Nyström method** generates a normalized search space.

Selection-Reproduction: **Elitism** selection and **Tournament** reproduction.

Crossover: Exchanges strings.

Mutation: **Uniform** random mutation of the centroids coordinates.

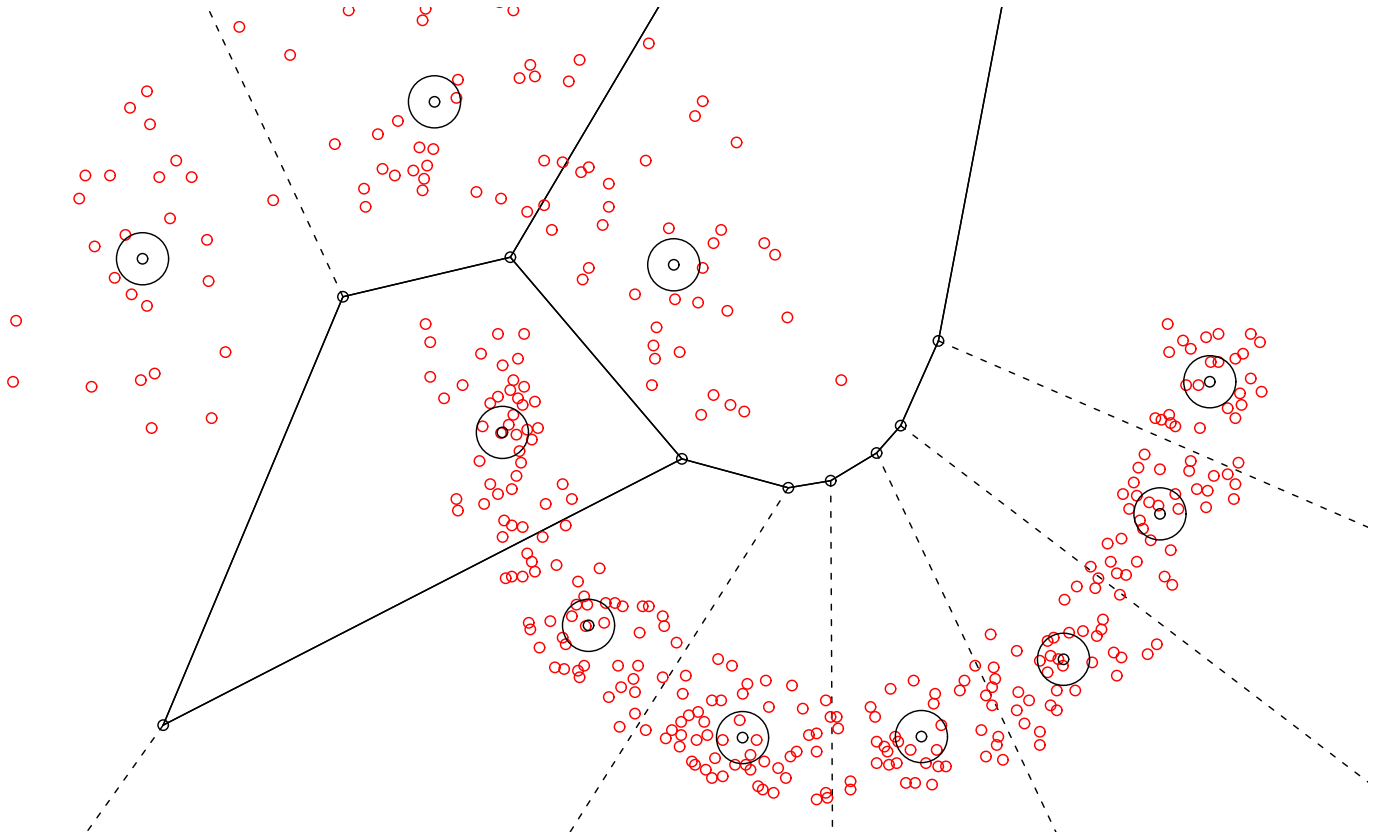
Large Data: Reduce the Space

We need to zoom out the space.

Goal: analyze the data in a general way.

Reducing the space: Clustering as Manifolds

Voronoi Tessellation for Jain

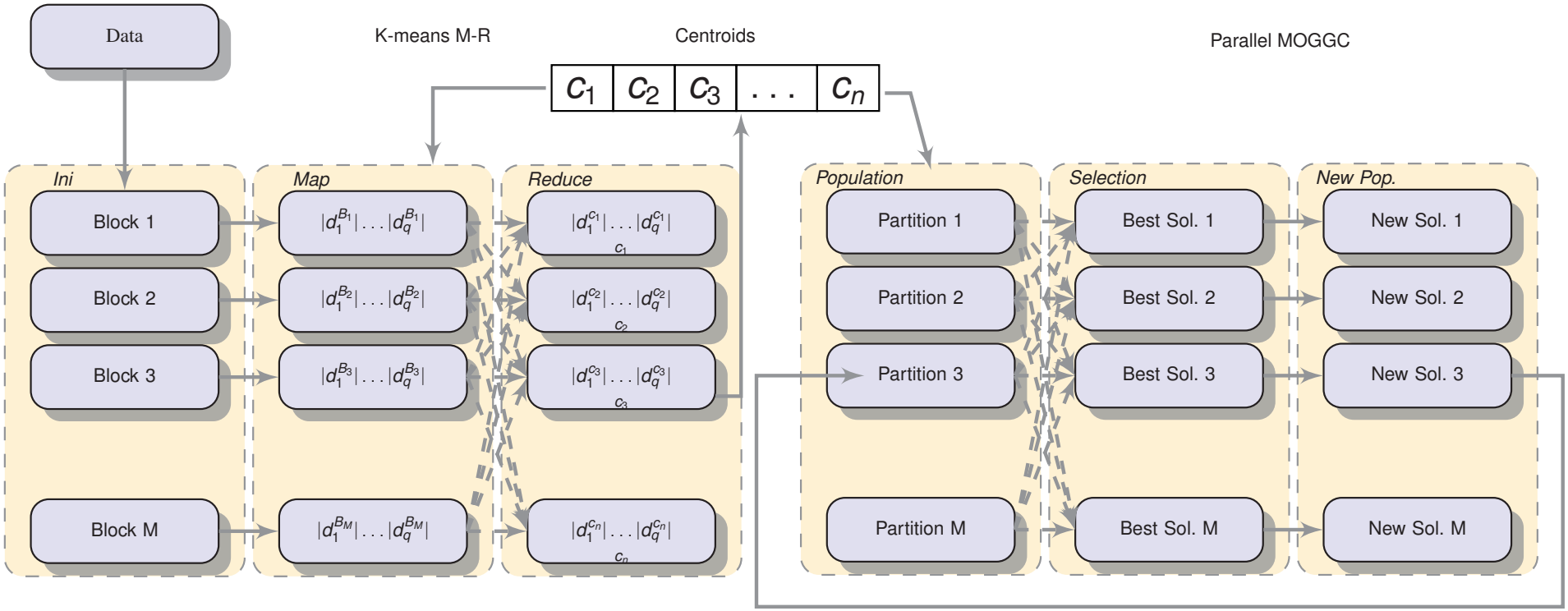


MOGCLA Motivation

MOGCLA tries to **optimize** the **set of centroids** using a search in two levels:

1. The **micro** search implements a **centroid selection**.
2. The **macro** search **joins the regions** to form the manifolds.

MOGCLA Algorithm



Online Clustering

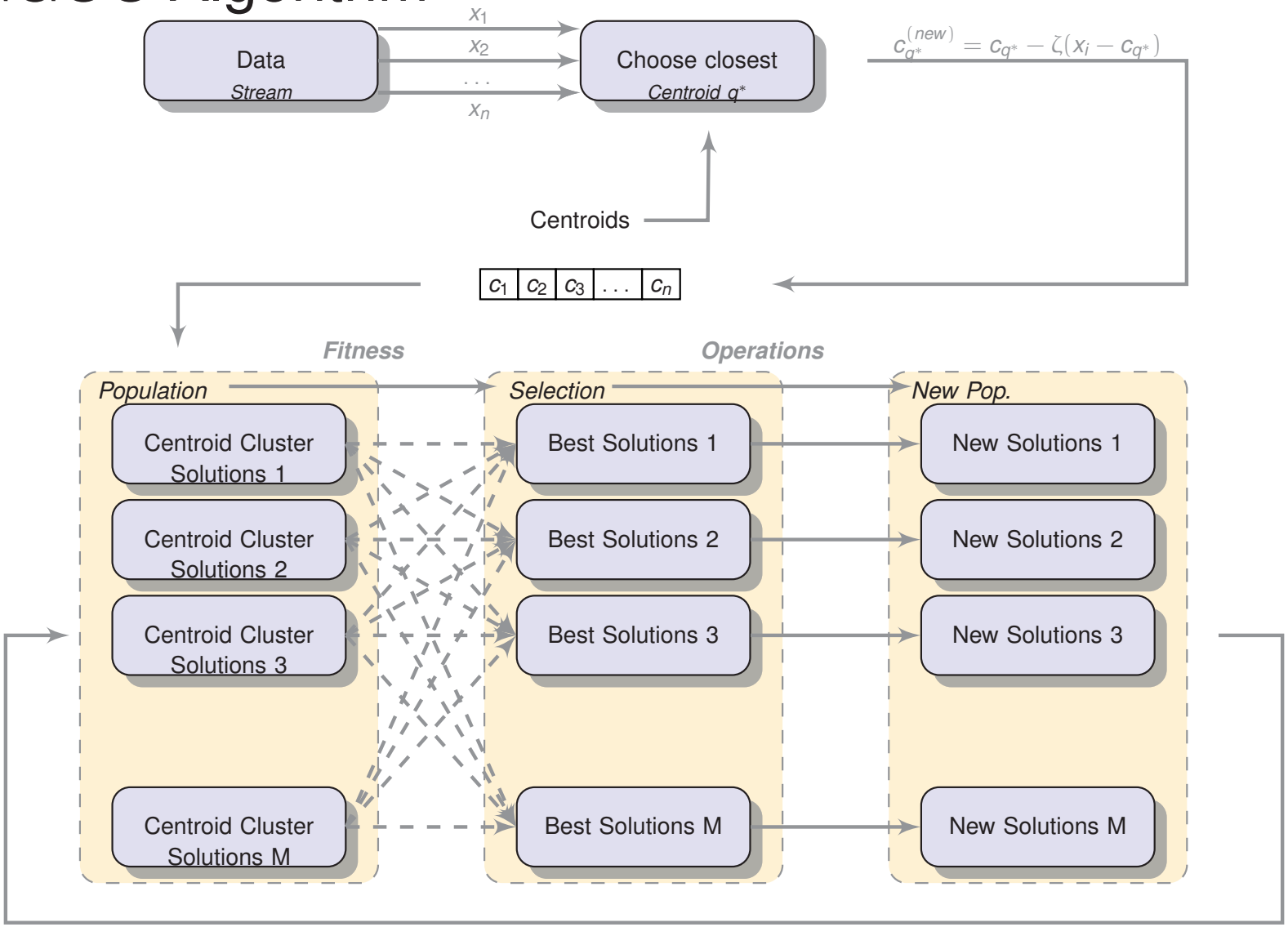
Data Real-time analysis.

Goal: construct the manifold with the data online.

MOGGOC Motivation

- The new MOGGOC algorithm tries to analyse the data in an online way in order to generate a **real-time analysis**.
- This algorithm follows the same criteria than the previous one, however, using online clustering.
- It has been compared with online approaches.

MOGGOC Algorithm



Conclusions

Evolutionary Computation shows promising approaches for Manifold Clustering.

There are still some problems that need to be solve: like **manifold intersection**, improve the **number of Manifold selection** and more **stream-based memory**.

A Tutorial on Manifold Clustering using Genetic Algorithms

Hector D. Menendez

Department of Computer Science
University College London

2nd September 2015