Towards an Adaptive Encoding for Evolutionary Data Clustering

GECCO 2018

Cameron Shand[1], Richard Allmendinger[2], Julia Handl[2], & John Keane[1]

[1]*School of Computer Science, University of Manchester*
[2]*Alliance Manchester Business School, University of Manchester*

Overview

Overview

- Do we always need every decision variable for the whole search?

Overview

- Do we always need every decision variable for the whole search?

- We investigate (using an existing state-of-the-art EC algorithm):

Overview

- Do we always need every decision variable for the whole search?

- We investigate (using an existing state-of-the-art EC algorithm):
    1. When restricting at the start, can we identify during run-time that we need to expand the search space?

Overview

- Do we always need every decision variable for the whole search?

- We investigate (using an existing state-of-the-art EC algorithm):
  1. When restricting at the start, can we identify during run-time that we need to expand the search space?
  2. After expansion, can we employ strategies to focus on the new space?

Evolutionary Multi-objective Clustering

Why cluster using EAs?

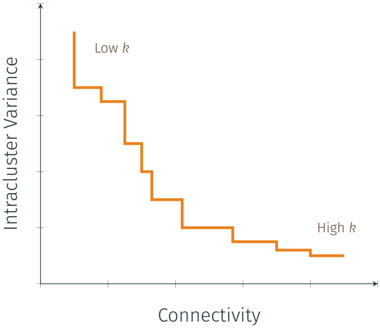
- Use multiple clustering criteria (fewer assumptions)

Why cluster using EAs?

- Use multiple clustering criteria (fewer assumptions)
- Flexibility in the representation of the problem

Why cluster using EAs?

- Use multiple clustering criteria (fewer assumptions)
- Flexibility in the representation of the problem
- Produces a set of results for additional analysis



Δ-MOCK

Representations

- The representation of the problem is key

Representations

- The representation of the problem is key
- One example for EC is the locus-based adjacency representation

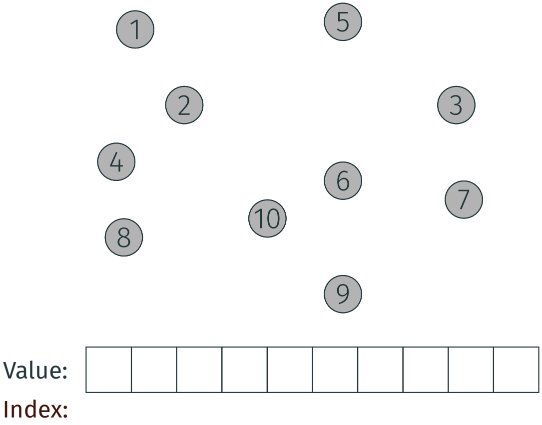
Representations

- The representation of the problem is key
- One example for EC is the locus-based adjacency representation
- Provides flexibility in representation (finds k)

Representations

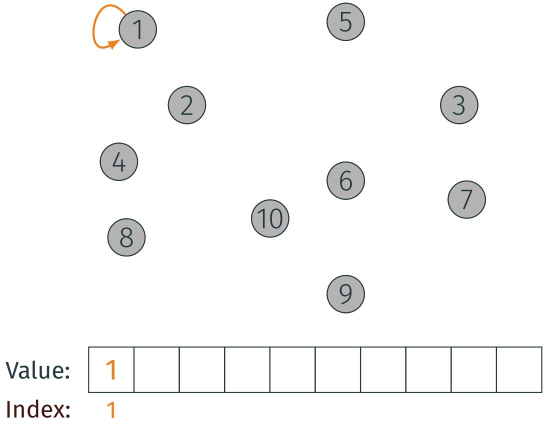
- The representation of the problem is key
- One example for EC is the locus-based adjacency representation
- Provides flexibility in representation (finds k)
- Poor scaling (genotype is of length N)

Locus-based Adjacency Representation



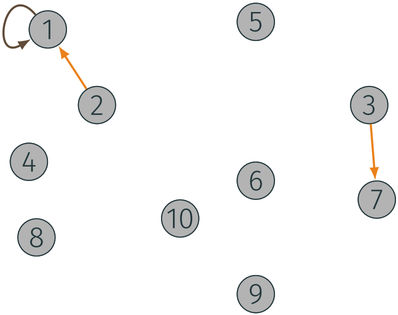
- Data points are nodes on a graph

Locus-based Adjacency Representation



- Data points are nodes on a graph
- Value (j) in gene x_i represents edge ($i \rightarrow j$)

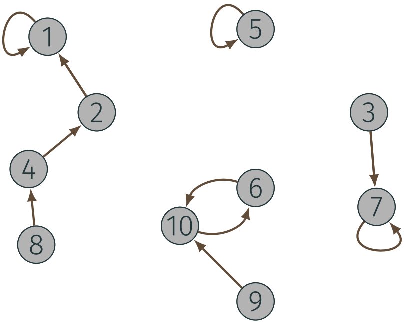
Locus-based Adjacency Representation



- Data points are nodes on a graph
- Value (j) in gene x_i represents edge ($i \rightarrow j$)

Value:	1	1	7							
Index:	1	2	3							

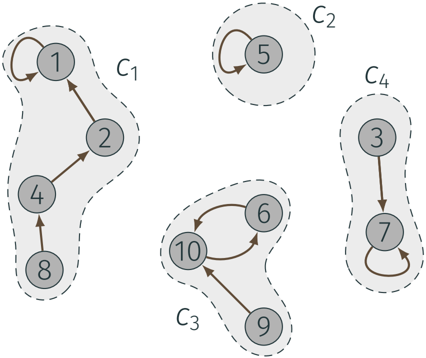
Locus-based Adjacency Representation



- Data points are nodes on a graph
- Value (j) in gene x_i represents edge ($i \rightarrow j$)

Value:	1	1	7	2	5	10	7	4	10	6
Index:	1	2	3	4	5	6	7	8	9	10

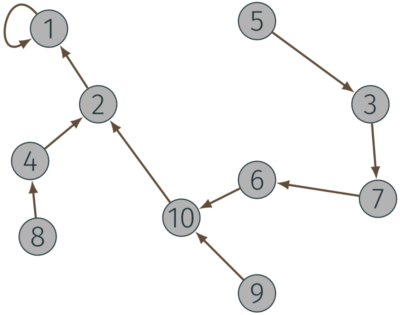
Locus-based Adjacency Representation



Value:	1	1	7	2	5	10	7	4	10	6
Index:	1	2	3	4	5	6	7	8	9	10

- Data points are nodes on a graph
- Value (j) in gene x_i represents edge ($i \rightarrow j$)
- Connected components of the graph represent clusters

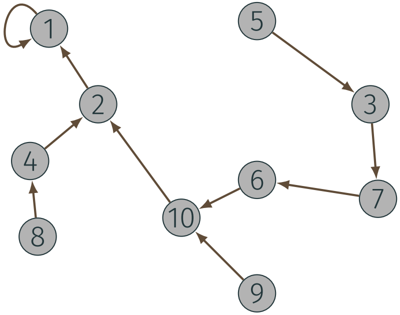
Interesting Links



1	1	7	2	3	10	6	4	10	2
1	2	3	4	5	6	7	8	9	10

- First determine the MST

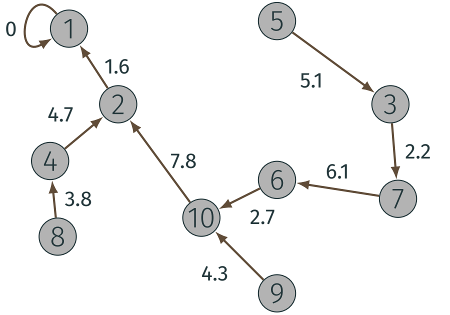
Interesting Links



1	1	7	2	3	10	6	4	10	2
1	2	3	4	5	6	7	8	9	10

- First determine the MST
- Some links are more relevant than others

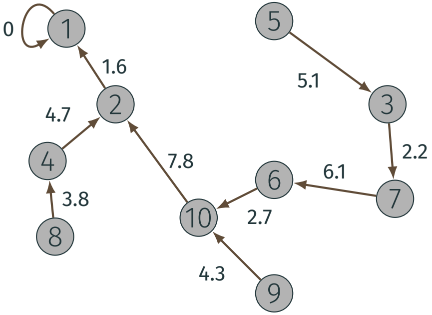
Interesting Links



1	1	7	2	3	10	6	4	10	2
1	2	3	4	5	6	7	8	9	10

- First determine the MST
- Some links are more relevant than others
- Calculate *degree of interestingness* (*DI*) for each link in MST

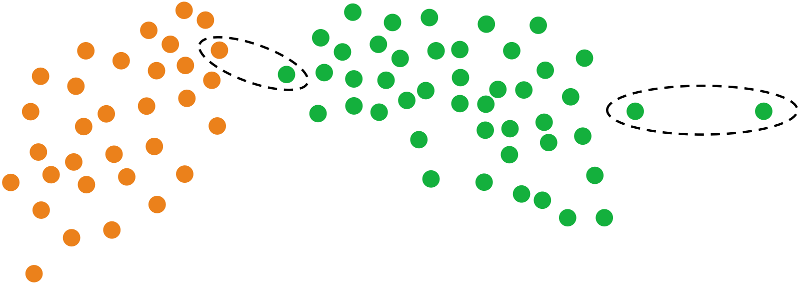
Interesting Links



1	1	7	2	3	10	6	4	10	2
1	2	3	4	5	6	7	8	9	10

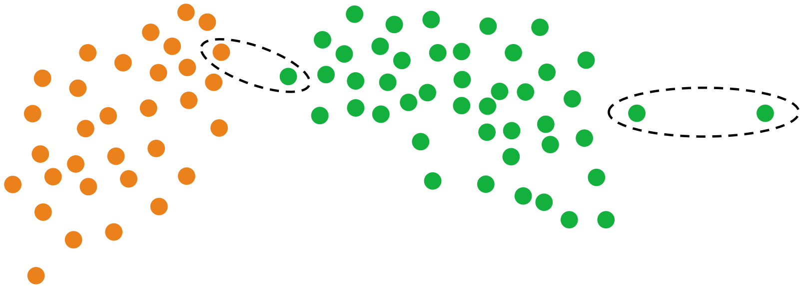
- First determine the MST
- Some links are more relevant than others
- Calculate *degree of interestingness* (*DI*) for each link in MST
- Restrict search to most interesting links

What is an interesting link?



$$DI(i \to j) = \min\left\{nn_i(j), nn_j(i)\right\} + \frac{\sigma(i,j)}{\sigma_{max}}$$

What is an interesting link?

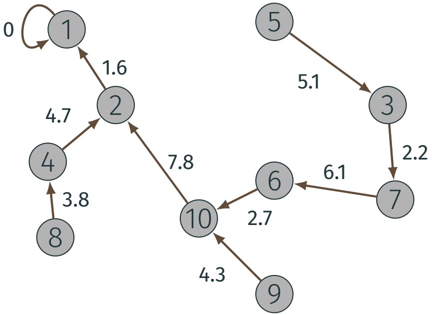


Mutual *nn* ranking

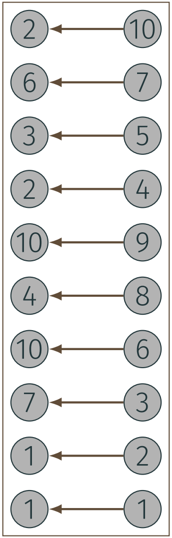
$$DI(i \to j) = \min\left\{nn_i(j), nn_j(i)\right\} + \frac{\sigma(i,j)}{\sigma_{max}}$$

Standardised
Euclidean distance

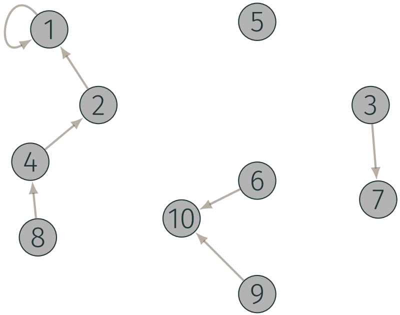
Reduced Encoding



1	1	7	2	3	10	6	4	10	2
1	2	3	4	5	6	7	8	9	10



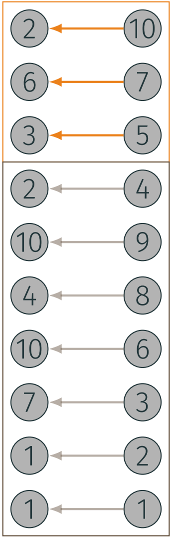
Reduced Encoding



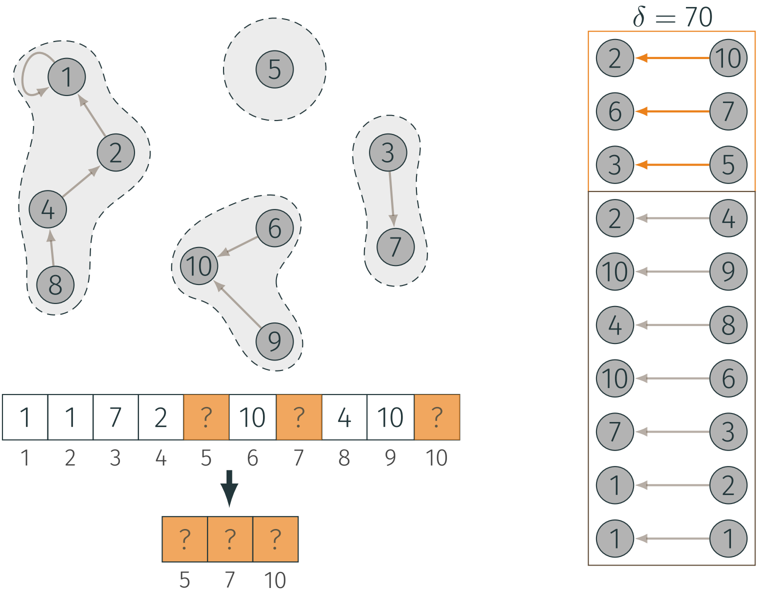
1	1	7	2	?	10	?	4	10	?
1	2	3	4	5	6	7	8	9	10

?	?	?
5	7	10

$\delta = 70$

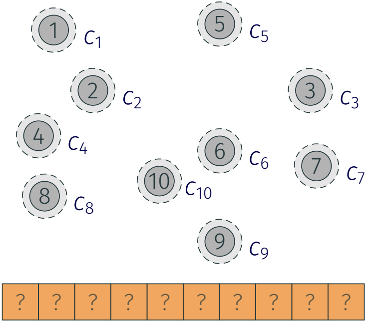


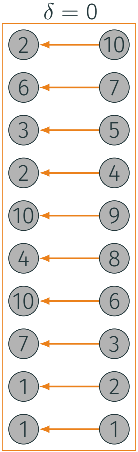
Reduced Encoding



The Extremes of δ

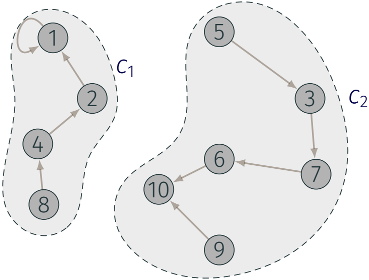
With a very low δ, the genotype and thus search space are large but not restrictive





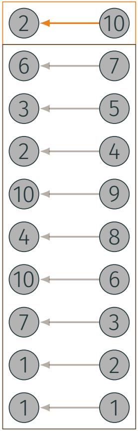
The Extremes of δ

With a very high δ, the optimisation problem can become trivial and meaningless



1	1	7	2	3	10	6	4	10	?

$\delta = 90$



The Role of δ

- Previous work[1] shows that δ can both reduce computation time and improve performance by focusing the search
- The optimal value is different for each dataset
- To avoid tuning, we can adapt this parameter

[1] Mario Garza-Fabre, Julia Handl, and Joshua Knowles. 2017. An Improved and More Scalable Evolutionary Approach to Multiobjective Clustering. IEEE Transactions on Evolutionary Computation V (2017)

Adapting δ

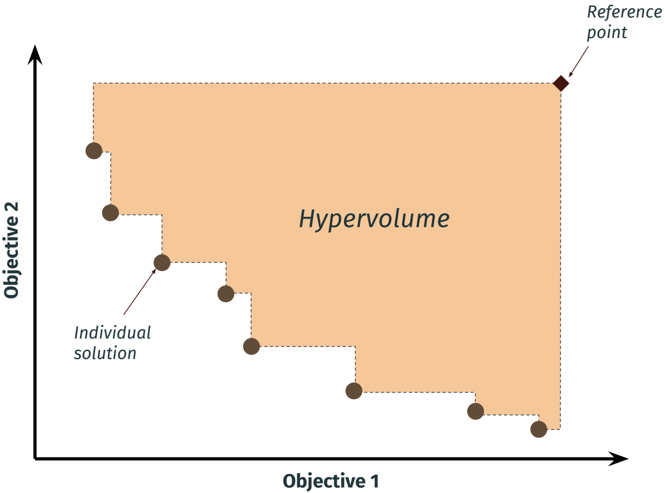
How do we adapt δ?

1. Identify that δ needs to change (and trigger this)

How do we adapt δ?

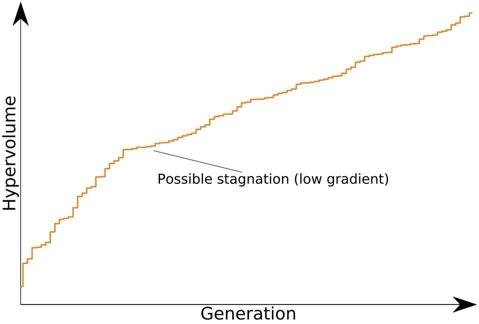
1. Identify that δ needs to change (and trigger this)
2. Explore the new space rapidly (avoiding previously explored space)

Identifying Convergence



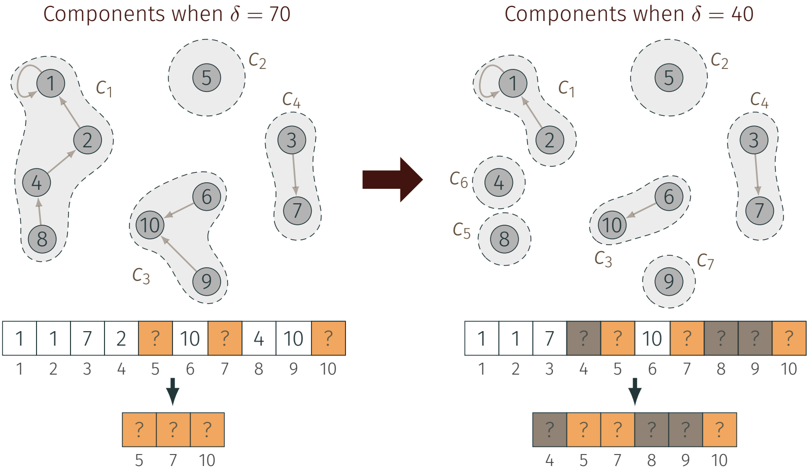
Triggering a Change in δ

- Trigger method identifies if δ should be changed
- Hypervolume indicates stagnation: current δ too restrictive

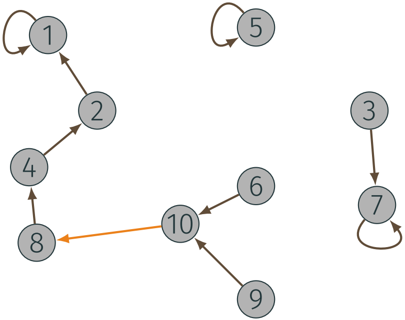


Decreasing δ

If hypervolume indicates stagnation, we need to expand the search space



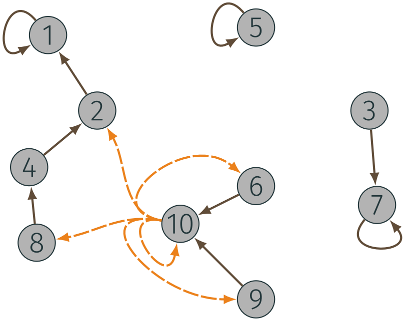
Neighbourhood-biased Mutation Operator



1	1	7	2	5	10	7	4	10	8

- Highlighted link successfully undergoes mutation

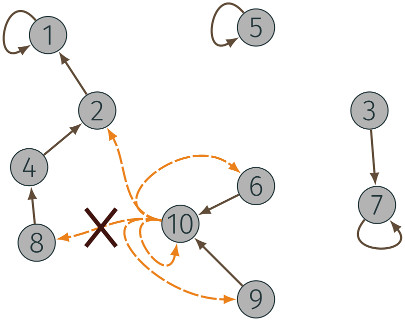
Neighbourhood-biased Mutation Operator



1	1	7	2	5	10	7	4	10	?

- Highlighted link successfully undergoes mutation
- Possible replacements from links to $L = 5$ nearest neighbours (inc. self-connecting)

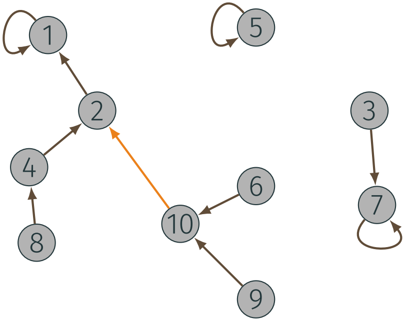
Neighbourhood-biased Mutation Operator



1	1	7	2	5	10	7	4	10	?

- Highlighted link successfully undergoes mutation
- Possible replacements from links to $L = 5$ nearest neighbours (inc. self-connecting)

Neighbourhood-biased Mutation Operator



1	1	7	2	5	10	7	4	10	2

- Highlighted link successfully undergoes mutation
- Possible replacements from links to $L = 5$ nearest neighbours (inc. self-connecting)
- New link is randomly selected (exc. previous)

Exploring the new search space

Upon changing δ, several search strategies are tested to explore the new space

Exploring the new search space

Upon changing δ, several search strategies are tested to explore the new space

- Triggered hypermutation[1] (2 methods):

[1] Helen G Cobb. 1990. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report (1990)

Exploring the new search space

Upon changing δ, several search strategies are tested to explore the new space

- Triggered hypermutation[1] (2 methods):
    1. Hypermutation rate is applied to all genes (TH_{all})
    2. This rate is applied only to the new genes (TH_{new})

[1] Helen G Cobb. 1990. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report (1990)

Exploring the new search space

Upon changing δ, several search strategies are tested to explore the new space

- Triggered hypermutation[1] (2 methods):
    1. Hypermutation rate is applied to all genes (TH_{all})
    2. This rate is applied only to the new genes (TH_{new})
- Fair mutation[2] (FM)

[1] Helen G Cobb. 1990. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report (1990)

[2] Richard Allmendinger and Joshua Knowles. 2010. Evolutionary optimization on problems subject to changes of variables. Lecture Notes in Computer Science 6239 LNCS, PART 2 (2010)

Exploring the new search space

Upon changing δ, several search strategies are tested to explore the new space

- Triggered hypermutation[1] (2 methods):
    1. Hypermutation rate is applied to all genes (TH_{all})
    2. This rate is applied only to the new genes (TH_{new})
- Fair mutation[2] (FM)
- Δ-MOCK's initialisation routine (RO)

[1] Helen G Cobb. 1990. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report (1990)

[2] Richard Allmendinger and Joshua Knowles. 2010. Evolutionary optimization on problems subject to changes of variables. Lecture Notes in Computer Science 6239 LNCS, PART 2 (2010)

Exploring the new search space

Upon changing δ, several search strategies are tested to explore the new space

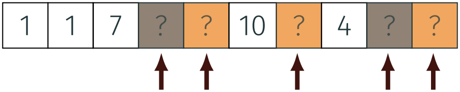
- Triggered hypermutation[1] (2 methods):
  1. Hypermutation rate is applied to all genes (TH_{all})
  2. This rate is applied only to the new genes (TH_{new})
- Fair mutation[2] (FM)
- Δ-MOCK's initialisation routine (RO)
- No additional changes (control method) (CO)

[1] Helen G Cobb. 1990. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report (1990)

[2] Richard Allmendinger and Joshua Knowles. 2010. Evolutionary optimization on problems subject to changes of variables. Lecture Notes in Computer Science 6239 LNCS, PART 2 (2010)

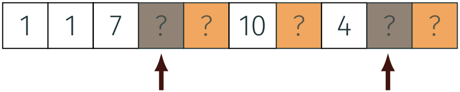
Triggered Hypermutation

TH_{all}



Hypermutation rate
applied to all genes
in reduced genotype

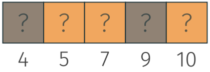
TH_{new}



Hypermutation rate
applied to new genes
in reduced genotype

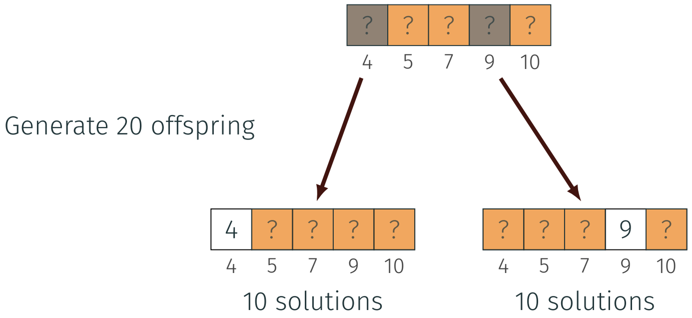
Fair Mutation

- Aim is to explore new solutions for each of the new genes



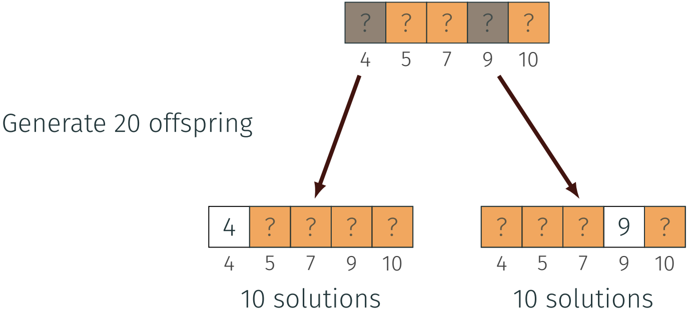
Fair Mutation

- Aim is to explore new solutions for each of the new genes
- Generate offspring where equal portion have one of the new genes set to self-connecting link



Fair Mutation

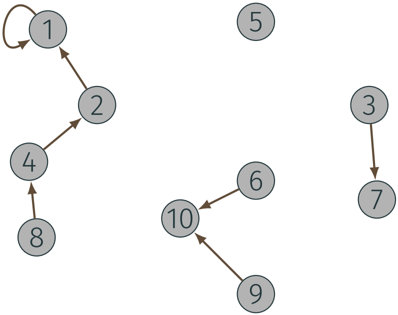
- Aim is to explore new solutions for each of the new genes
- Generate offspring where equal portion have one of the new genes set to self-connecting link
- Permits exploration of new component combinations



Reinitialised Offspring

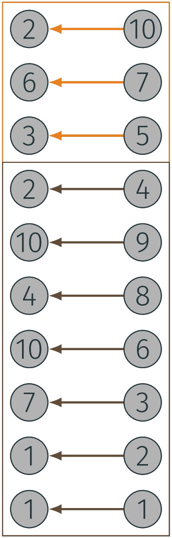
- Randomly select a subset of our most interesting links in the MST (bound by δ) to remove
- A new link is then randomly selected (similar to mutation) to replace it

Reinitialised Offspring

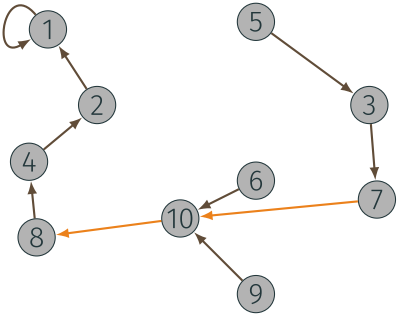


1	2	3	4	5	6	7	8	9	10
1	1	7	2	?	10	?	4	10	?

$\delta = 70$

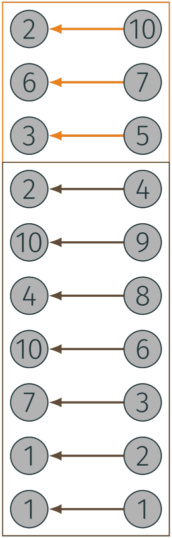


Reinitialised Offspring

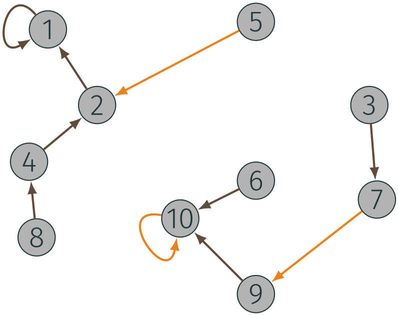


1	1	7	2	3	10	10	4	10	8
1	2	3	4	5	6	7	8	9	10

$\delta = 70$

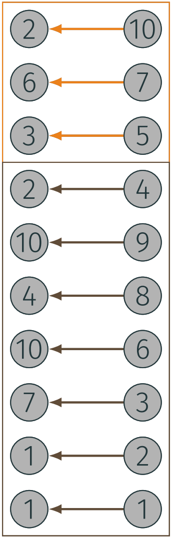


Reinitialised Offspring



1	1	7	2	2	10	9	4	10	10
1	2	3	4	5	6	7	8	9	10

$\delta = 70$



Experiments

Experimental Aims

- The aim was to show whether adapting δ would:
    1. Recover performance (ARI) when starting with a restrictive δ value

Experimental Aims

- The aim was to show whether adapting δ would:
  1. Recover performance (ARI) when starting with a restrictive δ value
  2. When compared to Δ-MOCK, if at least similar performance could be achieved with less computation time

Experimental Setup

- Compare hypervolume trigger method with two control methods to specify when we decrease δ:

Experimental Setup

- Compare hypervolume trigger method with two control methods to specify when we decrease δ:
  1. Random: Random numbers signify when to change δ
  2. Interval: As above, but numbers are taken at regular intervals to ensure adequate time at each encoding length

Experimental Setup

- Compare hypervolume trigger method with two control methods to specify when we decrease δ:
    1. Random: Random numbers signify when to change δ
    2. Interval: As above, but numbers are taken at regular intervals to ensure adequate time at each encoding length
- Each of the 3 above trigger methods were run with all 5 search strategies (TH_{all}, TH_{new}, FM, RO, CO) on all data 30 times

Experimental Design

$$\begin{tikzcd}
\delta_{High} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i1} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i2} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i3} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{Low}
\end{tikzcd}$$

Two control methods (random and interval) have exactly 5 levels of resolution

Experimental Design

$$\begin{tikzcd}
\delta_{High} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i1} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i2} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i3} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{Low}
\end{tikzcd}$$

Two control methods (random and interval) have exactly 5 levels of resolution

$$\begin{tikzcd}
\delta_{High} \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{i1} \arrow[d, bend left, "\text{Trigger}"] \\
\vdots \arrow[d, bend left, "\text{Trigger}"] \\
\delta_{Low}
\end{tikzcd}$$

Hypervolume trigger method may have fewer triggers, but cannot decrease beyond δ_{Low}

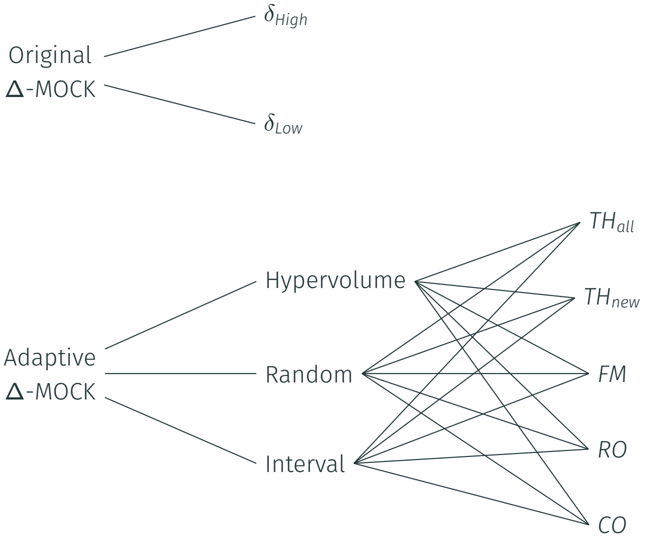
Experimental Setup

Original
Δ-MOCK

δ_{High}

δ_{Low}

Experimental Setup

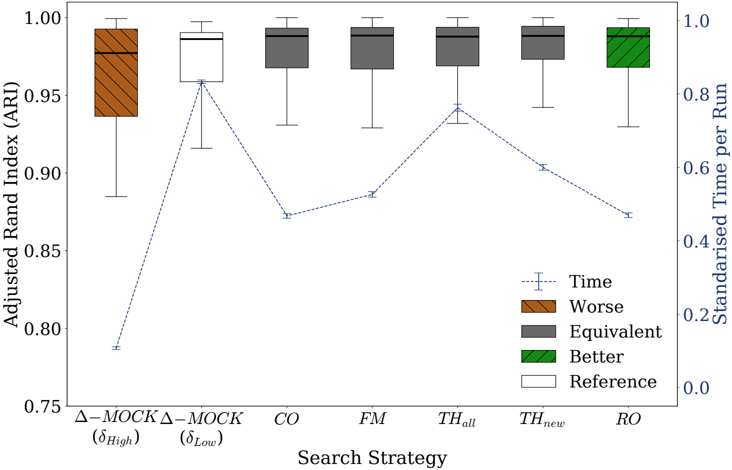


Datasets

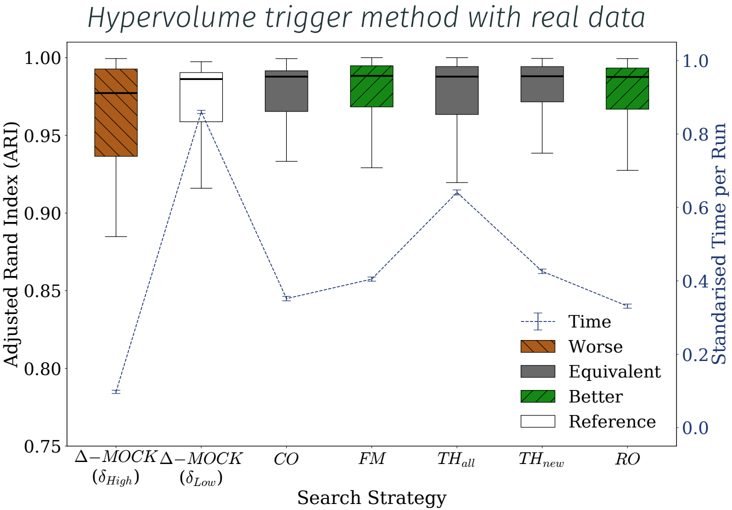
Dataset Type	# Datasets	# Clusters	# Dimensions	# Examples
Real	8	{10, 11, 12}	2	26,739 – 34,654
Synthetic	35	{10, 20, 40, 60, 80, 100, 120}	{20, 50, 100, 150, 200}	1,951 – 9,574

Results

Random trigger method with real data



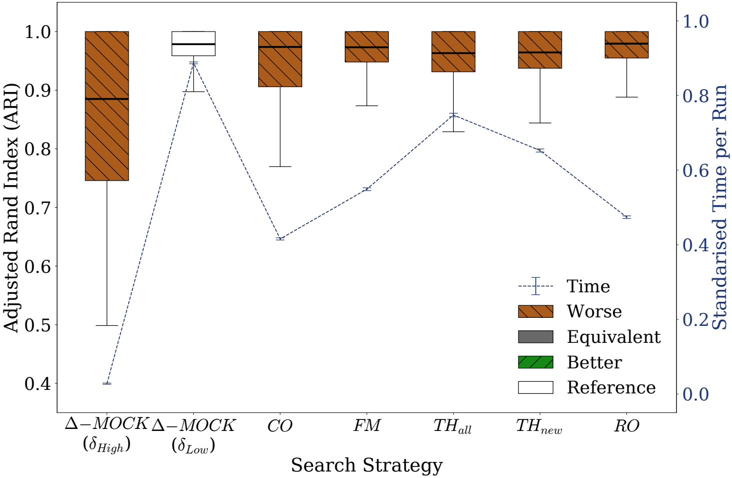
Results



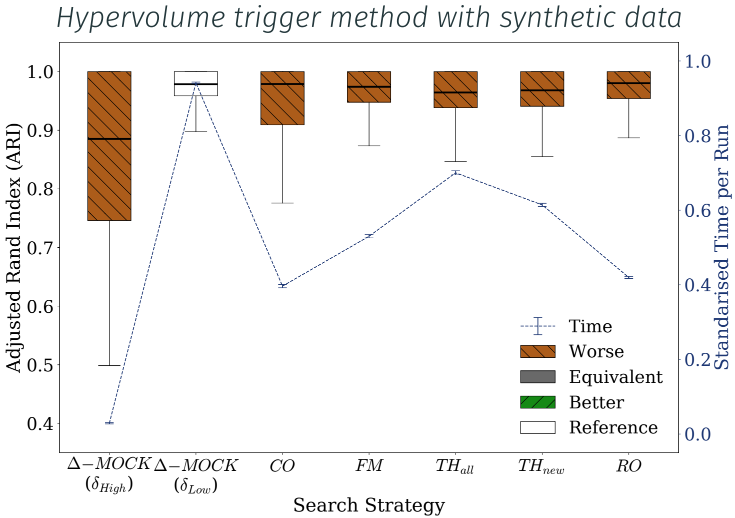
Hypervolume trigger method with real data
Adjusted Rand Index (ARI)
Standarised Time per Run
1.00
0.95
0.90
0.85
0.80
0.75
1.0
0.8
0.6
0.4
0.2
0.0
Time
Worse
Equivalent
Better
Reference
$\Delta{-}MOCK$ (δ_{High})
$\Delta{-}MOCK$ (δ_{Low})
CO
FM
TH_{all}
TH_{new}
RO
Search Strategy

Results

Random trigger method with synthetic data



Results


Hypervolume trigger method with synthetic data
Adjusted Rand Index (ARI)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Standarised Time per Run
1.0
0.8
0.6
0.4
0.2
0.0
Time
Worse
Equivalent
Better
Reference
$\Delta\text{−}MOCK$ (δ_{High})
$\Delta\text{−}MOCK$ (δ_{Low})
CO
FM
TH_{all}
TH_{new}
RO
Search Strategy

Results Summary

- *RO* search strategy is the most robust and fastest of the strategies

Results Summary

- *RO* search strategy is the most robust and fastest of the strategies
- Hypervolume trigger method appears effective and conservative

Results Summary

- *RO* search strategy is the most robust and fastest of the strategies
- Hypervolume trigger method appears effective and conservative
- Adapting δ is less effective for smaller datasets

Conclusions and Future Work

Issues/Future Work

- Not fully adaptive: δ can only be decreased

Issues/Future Work

- Not fully adaptive: δ can only be decreased
- Mutation operator bias put some search strategies at a disadvantage

Issues/Future Work

- Not fully adaptive: δ can only be decreased
- Mutation operator bias put some search strategies at a disadvantage
- Effectiveness of *RO* strategy indicates crossover should be investigated

Conclusions

- An adaptive encoding can reduce computation and focus the search

Conclusions

- An adaptive encoding can reduce computation and focus the search
- The hypervolume can be used to identify when to expand the search space

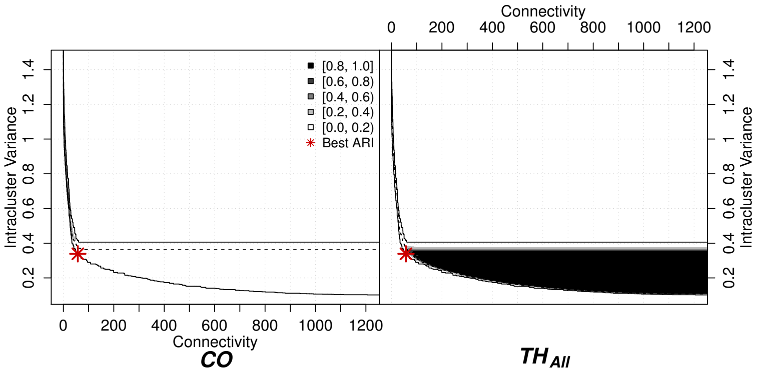
Conclusions

- An adaptive encoding can reduce computation and focus the search
- The hypervolume can be used to identify when to expand the search space
- With an appropriate strategy, performance can be maintained even when starting with a harmfully restrictive search space

Thank you! Questions?

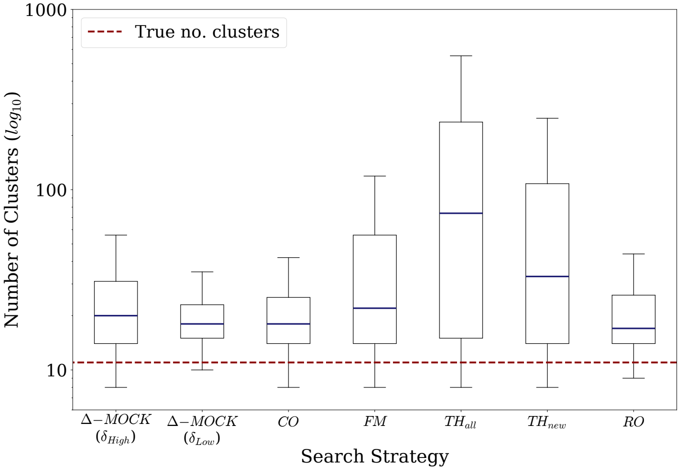
Mutation Bias

Mutation operator bias towards optimisation of the intracluster variance



Mutation Bias

The bias affects the quality of the Pareto front and search strategies



Objectives

Intra-cluster Variance

$$var(\mathcal{C}) = \frac{1}{N} \sum_{c \in \mathcal{C}} v(c) \qquad \text{where } v(c) = \sum_{i \in c} \sigma(i, \mu_c)^2$$

Objectives

Intra-cluster Variance

$$var(\mathcal{C}) = \frac{1}{N}\sum_{c\in\mathcal{C}} v(c) \qquad \text{where } v(c) = \sum_{i\in c}\sigma(i,\mu_c)^2$$

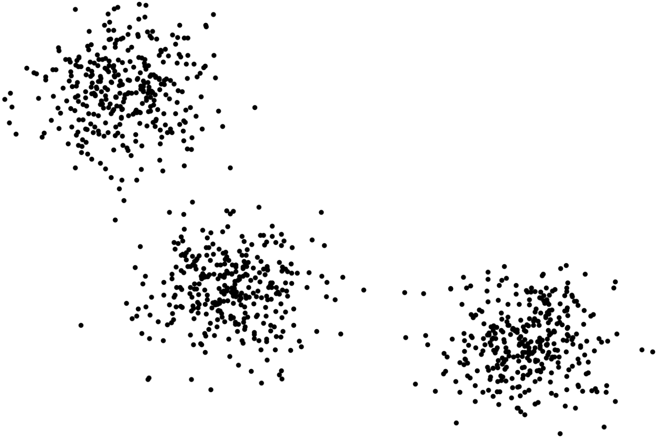
Connectivity

$$cnn(\mathcal{C}) = \sum_{i=1}^{N}\sum_{l=1}^{L}\rho(i,l)$$

$$\text{where } \rho(i,l) = \begin{cases} \frac{1}{l}, & \text{if } \nexists\, c \in \mathcal{C} \mid i \in c \wedge nn_{il} \in c; \\ 0, & \text{otherwise.} \end{cases}$$

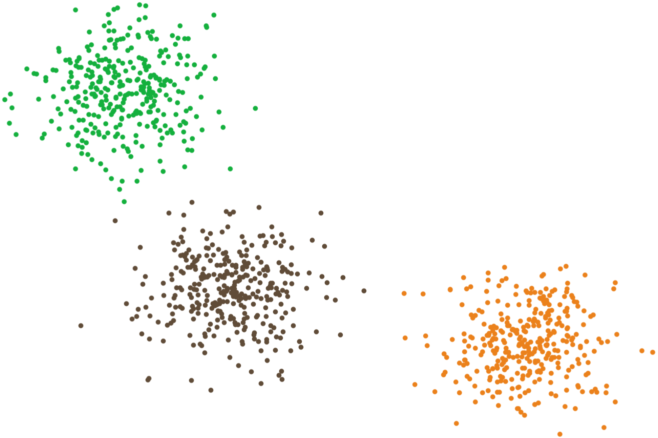
Clustering Subjectivity - A Toy Example

It is easy for humans to identify number of clusters (k) in toy data



Clustering Subjectivity - A Toy Example

With the exact k, a simple dataset is easy for methods such as KMeans



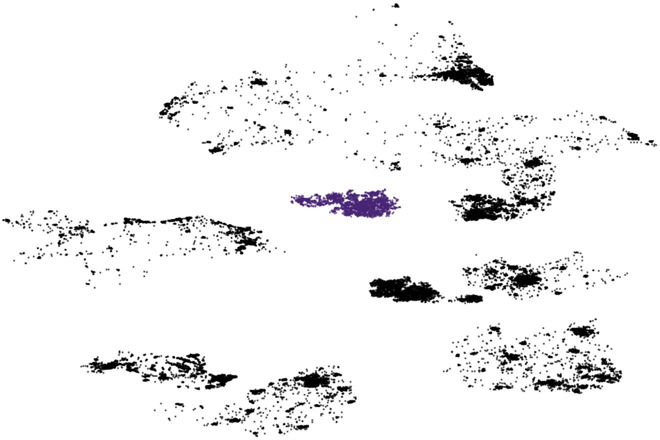
Real-world dataset example - how many clusters are there?

Clustering Subjectivity - A Real Example

Even in 2D there's uncertainty

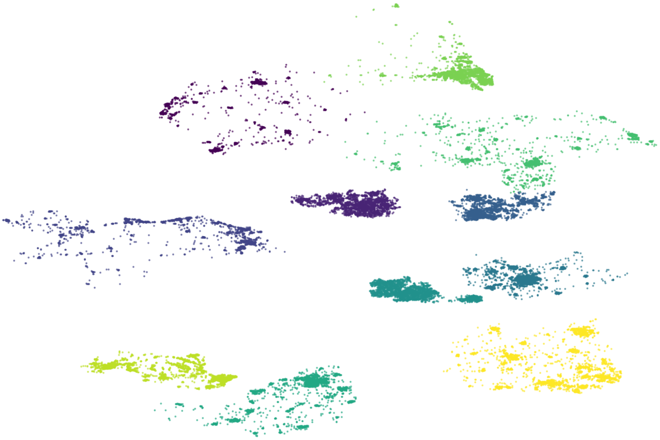
Clustering Subjectivity - A Real Example

Some clusters are obvious to humans and most apporaches



Clustering Subjectivity - A Real Example

Using all true labels, we can see that there are 11 clusters

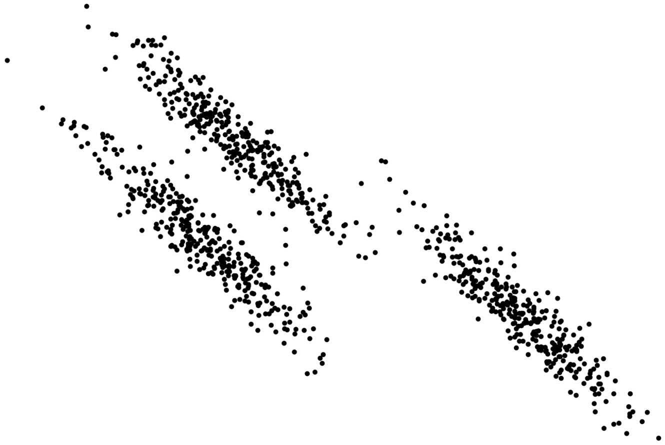


Clustering Subjectivity - A Real Example

Without this ground truth, is 11 easy to guess?

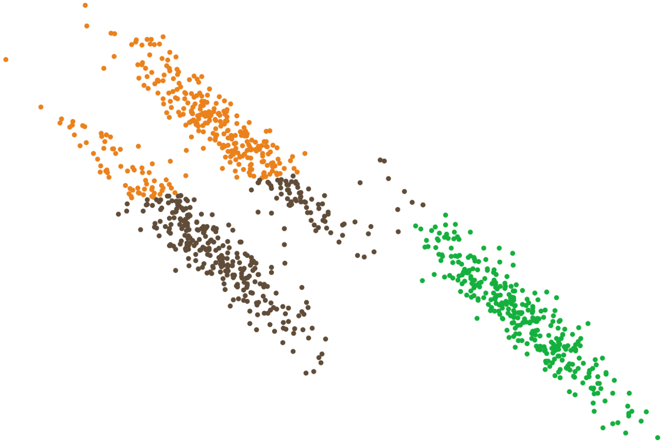
The need for multiple clustering criteria

Each criterion (e.g. intracluster variance) makes an assumption



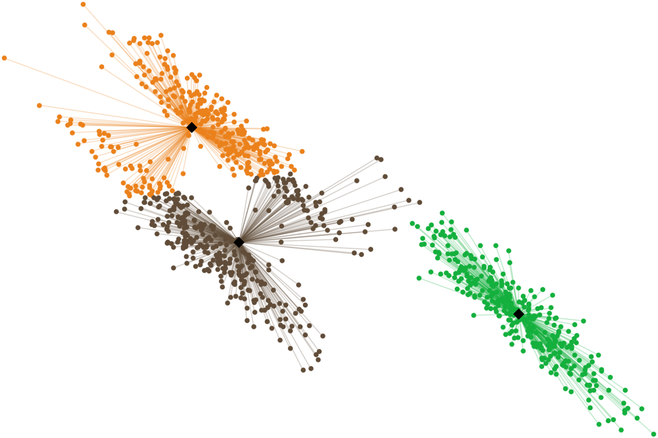
The need for multiple clustering criteria

Even knowing $k = 3$, this dataset is impossible for this criteria



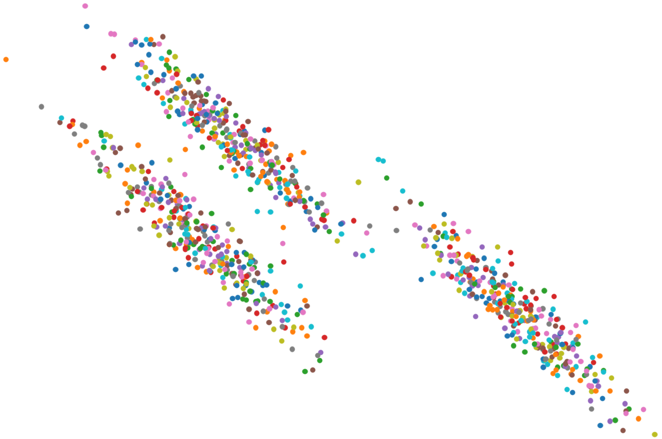
Objectives – Intracluster Variance

Intracluster variances minimises the distance from all data points to its centroid



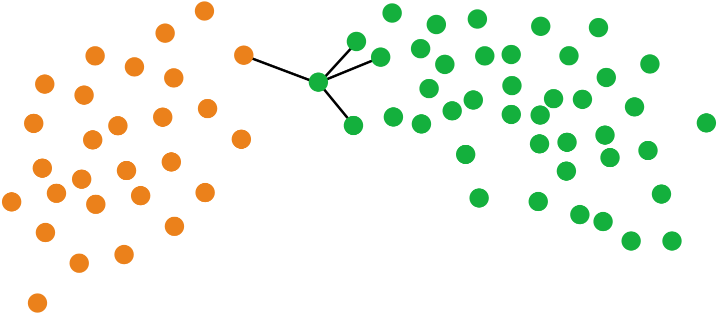
Objectives – Intracluster Variance

Ultimately, this is minimised when k equals number of data points (N)



Objectives – Connectivity

Optimising connectivity penalises differences in cluster assignment to each point's local neighbourhood



Objectives – Connectivity

Connectivity is minimised when $k = 1$