On the Definition of Dynamic Permutation Problems under Landscape Rotation

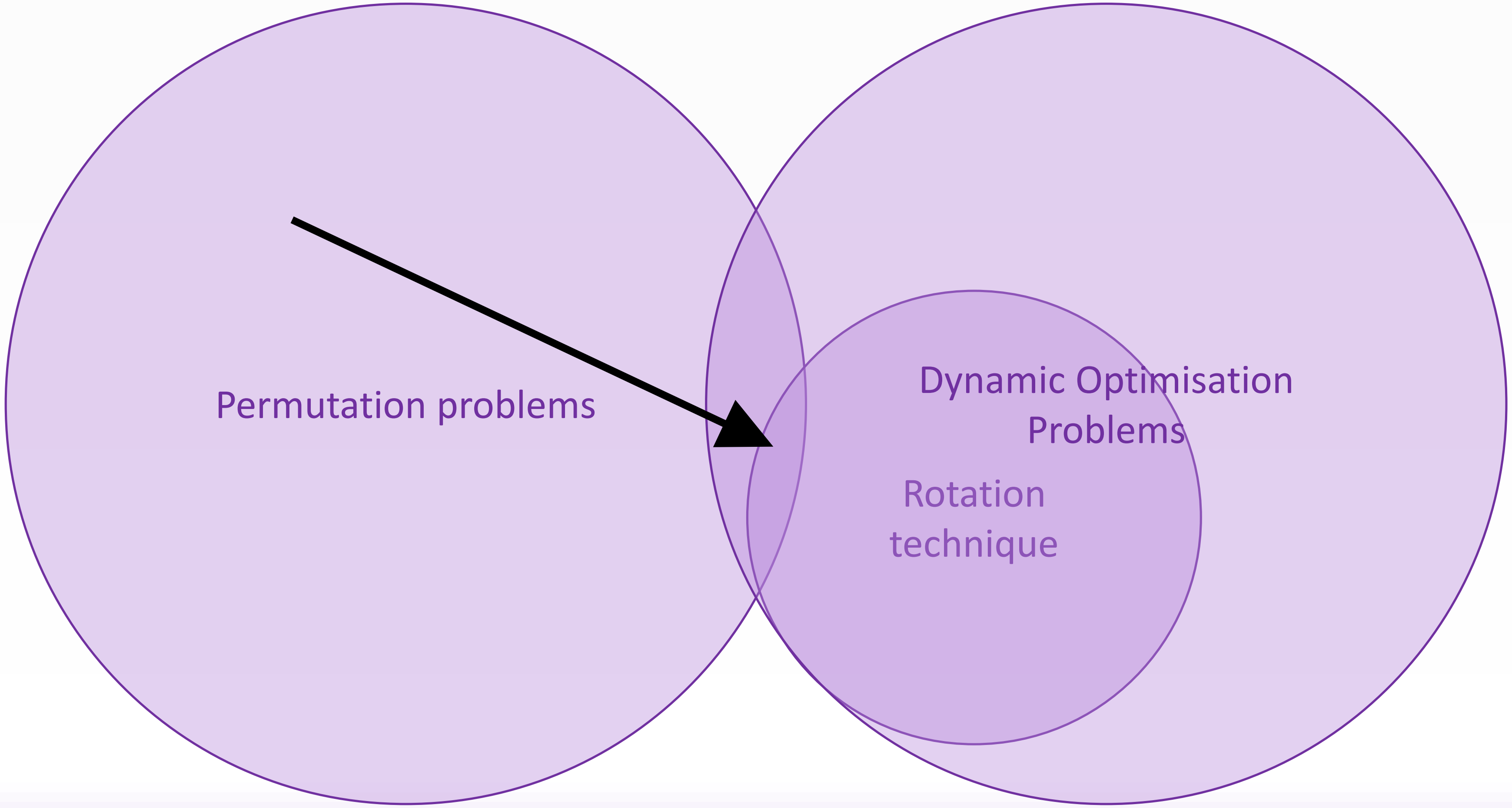
Joan Alza, Mark Bartlett, Josu Ceberio and John McCall.

Email: j.alza-santos@rgu.ac.uk

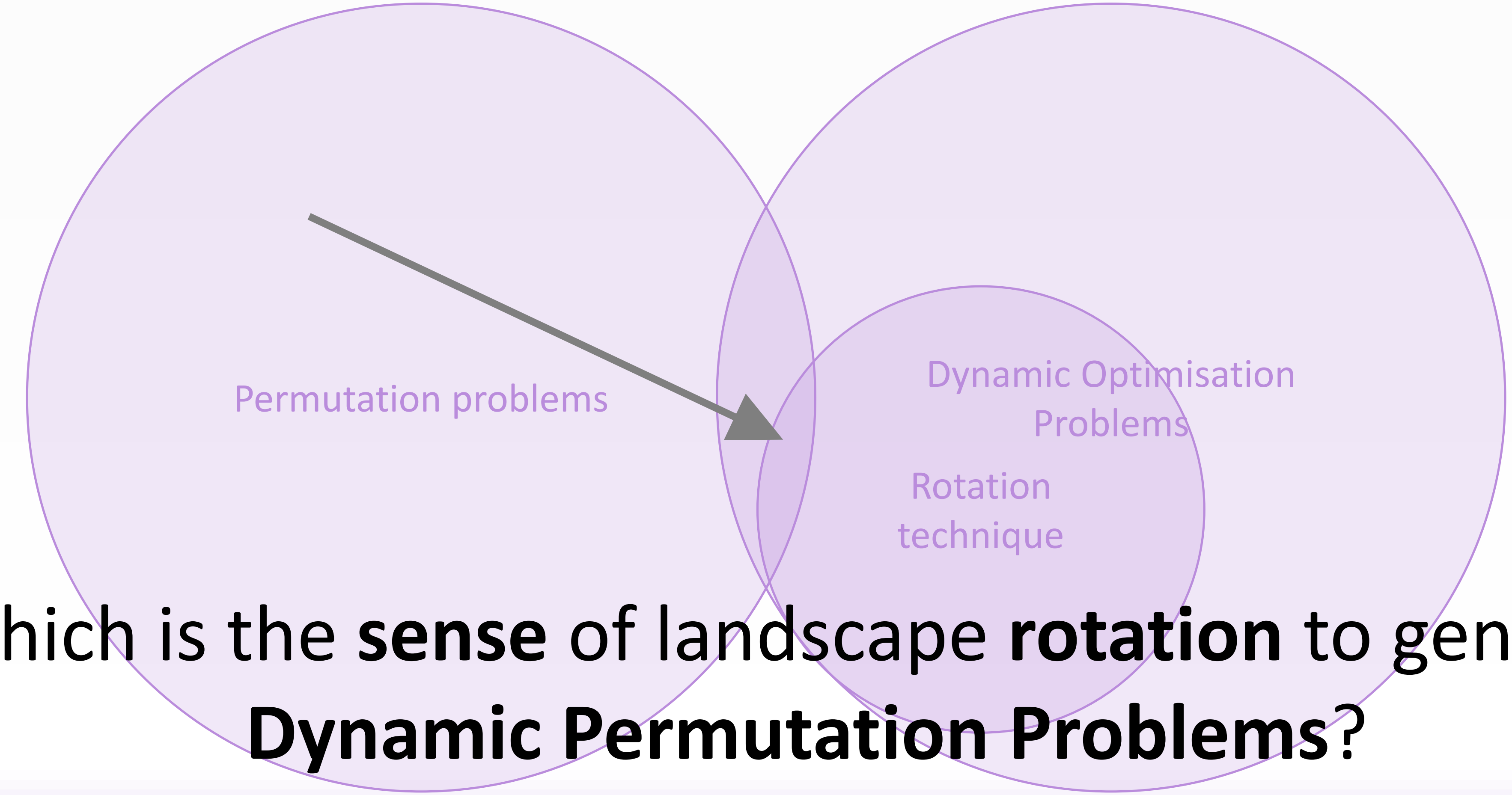
ROBERT GORDON UNIVERSITY ABERDEEN

Universidad del País Vasco
Euskal Herriko Unibertsitatea
The University of the Basque Country

Context


Permutation problems
Dynamic Optimisation Problems
Rotation technique

Context



Which is the **sense** of landscape **rotation** to generate **Dynamic Permutation Problems**?

Permutation problems

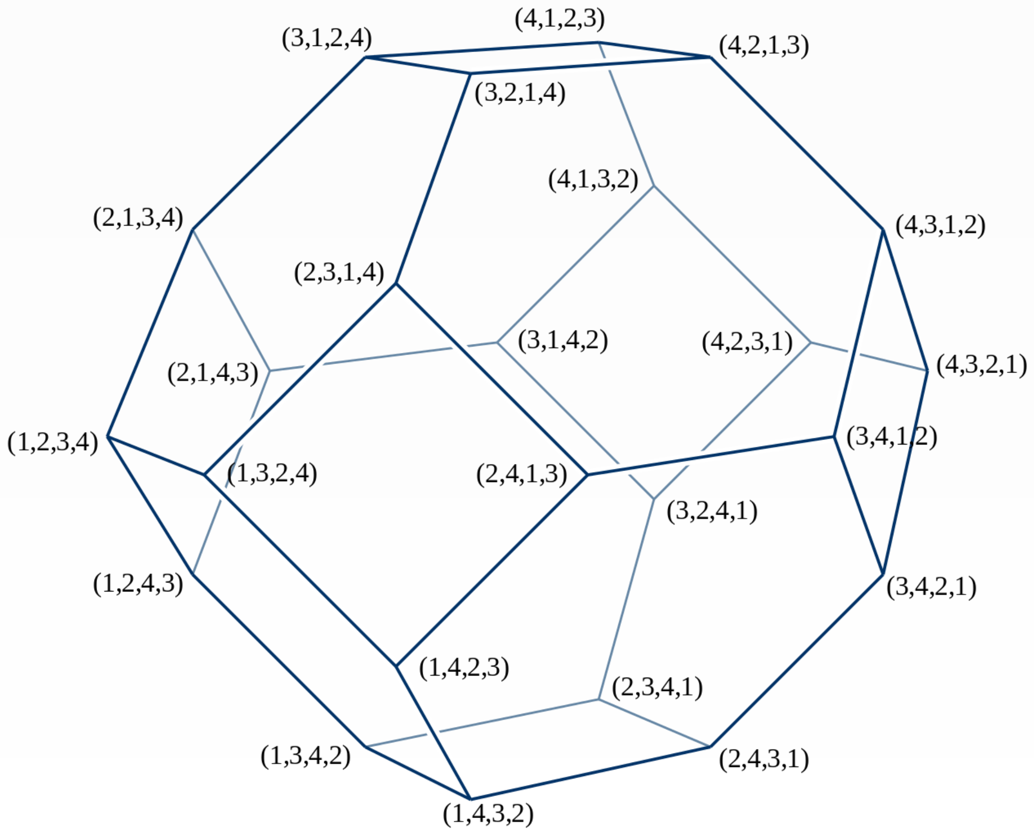
- Problems whose solutions are represented as permutations.

$$\sigma: \{1,2\ldots n\} \rightarrow \{1,2\ldots n\}$$

- ***e****:* identity permutation

Permutation problems

- Search space → n! permutations.
- Many of them are considered NP-Hard optimisation problem.



https://upload.wikimedia.org/wikipedia/commons/3/3e/Permutohedron.svg

DOPs – Definitions

- Sequence of static problems (instances) linked up by a dynamic rule.
- Problems with a time dependent parameter in the mathematical expression.
- Time-dependent problems that are solved *online,* by an algorithm, in a dynamic way as time goes by.

DOPs - Benchmarks

- Continuous domain:
  - Moving Peaks
  - GDBG
- Combinatorial domain:
  - XOR

DOPs - Benchmarks

- Continuous domain:
  - Moving Peaks
  - GDBG
- Combinatorial domain:
  - XOR

XOR

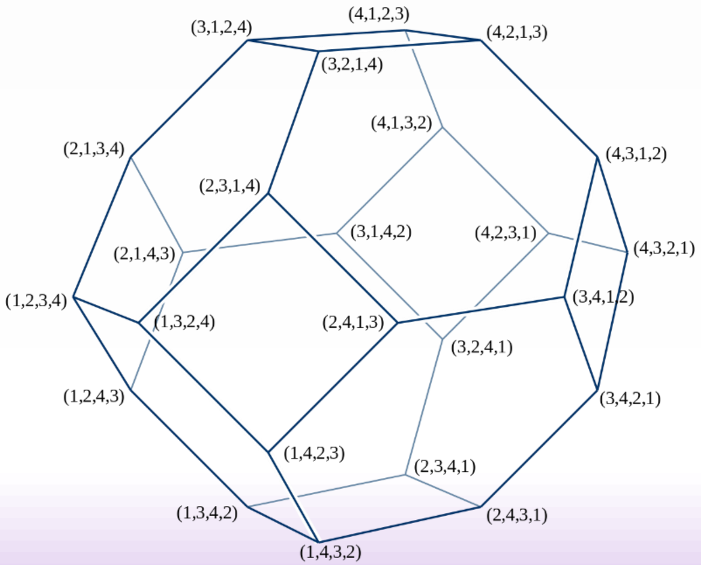
- It generates dynamic problems from any static binary problem.
- Applying an exclusive-or operator modifies the problem.
- It does not alter the search space → it just rotates the fitness landscape.

Benchmark generator - Rotation

- Adapted from the one proposed in

"A Benchmark Generator for Dynamic Permutation-Encoded Problems" (2012). M. Mavrovouniotis.

  - Modify the encoding of the problem → modify the location of the solution on the fitness landscape.



https://upload.wikimedia.org/wikipedia/commons/3/3e/Permutohedron.svg

Benchmark generator - Rotation

- Modify the encoding of the problem using permutation distance metrics:
  - **Cayley**: minimum number of swaps to convert σ into π.
    - *Maximum distance: $n – 1$.*
  - **Kendall's-τ**: minimum number of pairwise disagreement between σ and π.
    - *Maximum distance:* $\binom{n}{2}$.
  - **Ulam**: minimum number of insert operations to transform σ into e.
    - *Maximum distance: $n – 1$.*

Benchmark generator - Rotation

$\sigma = 3421 \qquad e = 1234$

- **Kendall's-τ** distance between σ and e = 5

$\sigma^{-1} = 4\mathbf{31}2 \rightarrow \mathbf{41}32 \rightarrow 14\mathbf{32} \rightarrow 1\mathbf{42}3 \rightarrow 12\mathbf{43} \rightarrow 1234 = e$

- **Cayley** distance between σ and e = 3:

$\sigma = \mathbf{3}42\mathbf{1} \rightarrow 1\mathbf{42}3 \rightarrow 12\mathbf{43} \rightarrow 1234 = e$

- **Ulam** distance between σ and e = 2:

$\sigma = 342\mathbf{1} \rightarrow 1342 \rightarrow 1234 = e$

Benchmark generator - Pattern

1. Generate a permutation u.a.r. at given distance and metric (π_i).
2. Compose with the previous permutation.

$$f(e \circ \sigma) = f(\sigma) \xrightarrow{c_1} f(\pi_1 \circ \sigma)$$

Benchmark generator - Pattern

1. Generate a permutation u.a.r. at given distance and metric (π_i).
2. Compose with the previous permutation.

$$f(e \circ \sigma) = f(\sigma) \xrightarrow{c_1} f(\pi_1 \circ \sigma) \xrightarrow{c_2} f(\pi_2 \circ \pi_1 \circ \sigma)$$

Benchmark generator - Pattern

1. Generate a permutation u.a.r. at given distance and metric (π_i).
2. Compose with the previous permutation.

$$f(e \circ \sigma) = f(\sigma) \xrightarrow{c_1} f(\pi_1 \circ \sigma) \xrightarrow{c_2} f(\pi_2 \circ \pi_1 \circ \sigma) \cdots \xrightarrow{c_k} f(\pi_k \circ \cdots \circ \pi_1 \circ \sigma)$$

Aim

- **Limited applicability**: quick and straightforward, but not realistic.

Ratify the applicability of the landscape rotation to generate Dynamic Permutation Problems.

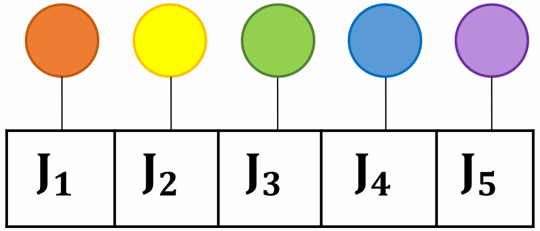
Case study

- A dynamic version of the PFSP.
- RKEDA: state-of-the-art on static PFSP.

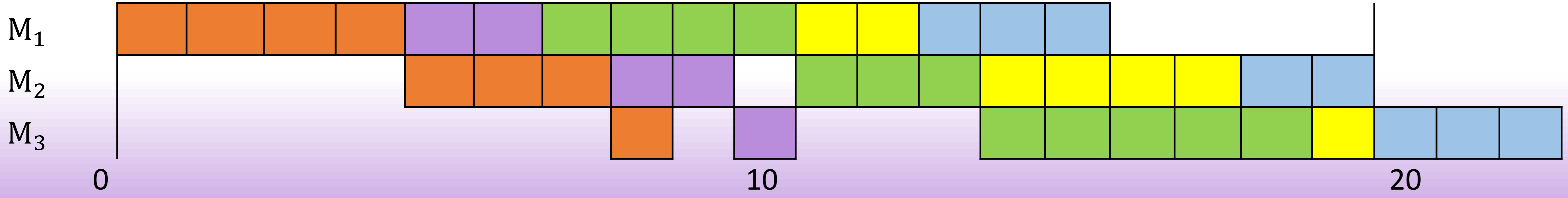
Flow Shop Scheduling Problem

- A set of *n* jobs have to be scheduled on *m* machines.
  - **Goal**: minimise the idle and waiting time.

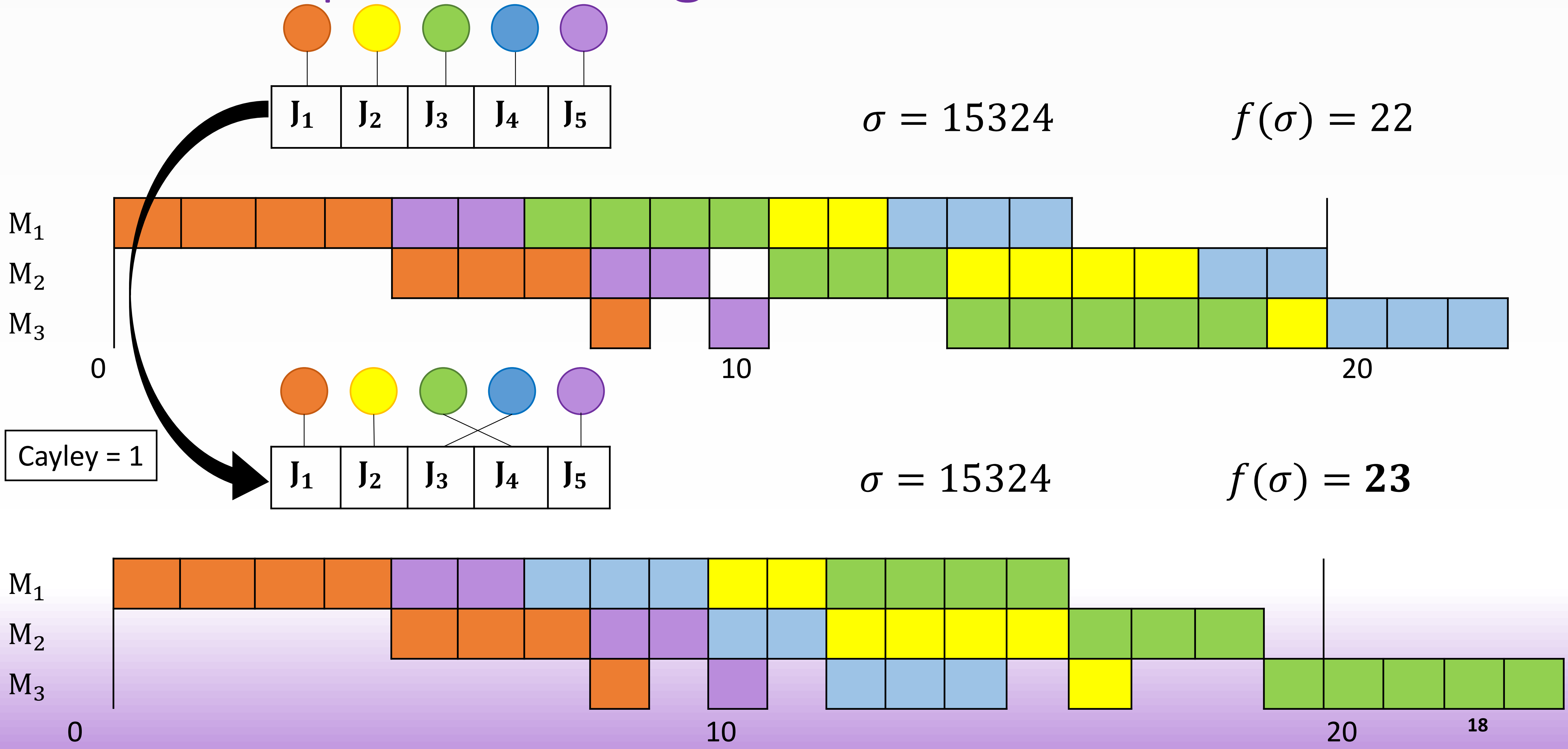
$$f(\sigma) = \sum_{i \in n} C_{\sigma(i),m}$$



$\sigma = 15324$ $f(\sigma) = 22$



Flow Shop Scheduling Problem



RKEDA

Random solutions
(0.63,0.21,0.34,0.6,0.89)
(0.3,0.57,0.45,0.43,0.14)
(0.07,0.48,0.2,0.83,0.96)
(0.26,0.38,0.44,0.34,0.27)
(0.69,0.12,0.87,0.5,0.76)

Normalise

Evaluate and sort

Normalised Population	Respective population
(0.75,0,0.25,0.5,1)	41235
(0.25,1,0.75,0.5,0)	25431
(0,0.5,0.25,0.75,1)	13245
(0,0.75,1,0.5,0.25)	14532
(0.5,0,1,0.25,0.75)	31524

Learn model

μ_1	μ_2	μ_3	μ_4	μ_5
0.33	0.5	0.41	0.58	0.66

Sample model

New samples
(0.63,0.21,0.34,0.6,0.89)
(0.29,0.44,0.39,0.55,0.65)
(0.26,0.45,0.46,0.43,0.51)
(0.36,0.67,0.32,0.74,0.76)
(0.43,0.74,0.42,0.61,0.93)

Normalise

RKEDA

Change of the landscape?

Evaluate and sort

Learn model

Sample model

μ_1	μ_2	μ_3	μ_4	μ_5
0.33	0.5	0.41	0.58	0.66

Random solutions
(0.63,0.21,0.34,0.6,0.89)
(0.3,0.57,0.45,0.43,0.14)
(0.07,0.48,0.2,0.83,0.96)
(0.26,0.38,0.44,0.34,0.27)
(0.69,0.12,0.87,0.5,0.76)

Normalised Population	Respective population
(0.75,0,0.25,0.5,1)	41235
(0.25,1,0.75,0.5,0)	25431
(0,0.5,0.25,0.75,1)	13245
(0,0.75,1,0.5,0.25)	14532
(0.5,0,1,0.25,0.75)	31524

New samples
(0.63,0.21,0.34,0.6,0.89)
(0.29,0.44,0.39,0.55,0.65)
(0.26,0.45,0.46,0.43,0.51)
(0.36,0.67,0.32,0.74,0.76)
(0.43,0.74,0.42,0.61,0.93)

Normalise

Normalise

RKEDA

Change of the landscape?

New samples
(0.36,0.67,0.32,0.74,0.76)
(0.26,0.45,0.46,0.43,0.51)
(0.43,0.74,0.42,0.61,0.93)
(0.36,0.53,0.43,0.60,0.77)
(0.29,0.44,0.39,0.55,0.65)

Reuse

aRKEDA

New samples
(0.36,0.67,0.32,0.74,0.76)
(0.26,0.45,0.46,0.43,0.51)
(0.43,0.74,0.42,0.61,0.93)
(0.36,0.53,0.43,0.60,0.77)
(0.29,0.44,0.39,0.55,0.65)

Restart

rRKEDA

Random solutions
(0.12,0.88,0.5,0.9,0.56)
(0.56,0.72,0.62,0.82,0.69)
(0.11,0.04,0.02,0.12,0.09)
(0.23,0.98,0.12,0.22,0.85)
(0.72,0.24,0.58,0.03,0.49)

Experimental setup

Parameters

Population size: 10n
Truncation size: n
Elitism criteria is used.
Number of samples: 10n
Max. generations: 100nk

Taillard's instances*

20x5, 20x10 & 20x20;
50x5, 50x10 & 50x20;
100x5, 100x10 & 100x20.

Benchmarks

Number of changes (k): 10
Periodically distributed.
30 DOPs per instance/metric.
Limitations:

- **Cayley** & **Ulam**: all distances.
- **Kendall's-τ**:
  - n=20 → all distances.
  - n=50 → from 1 to 150.
  - n=100 → from 1 to 50.

Algorithms

aRKEDA & rRKEDA.
Initial variance: 0.15

Performance measure

$$ARPD = \frac{1}{G}\sum_{i\in G}\frac{f_i(best) - Best\ known}{Best\ known}.$$

*Taillard, E. (1993). Benchmarks for basic scheduling problems. *European journal of operational research*, *64*(2), 278-285.

Experimental setup

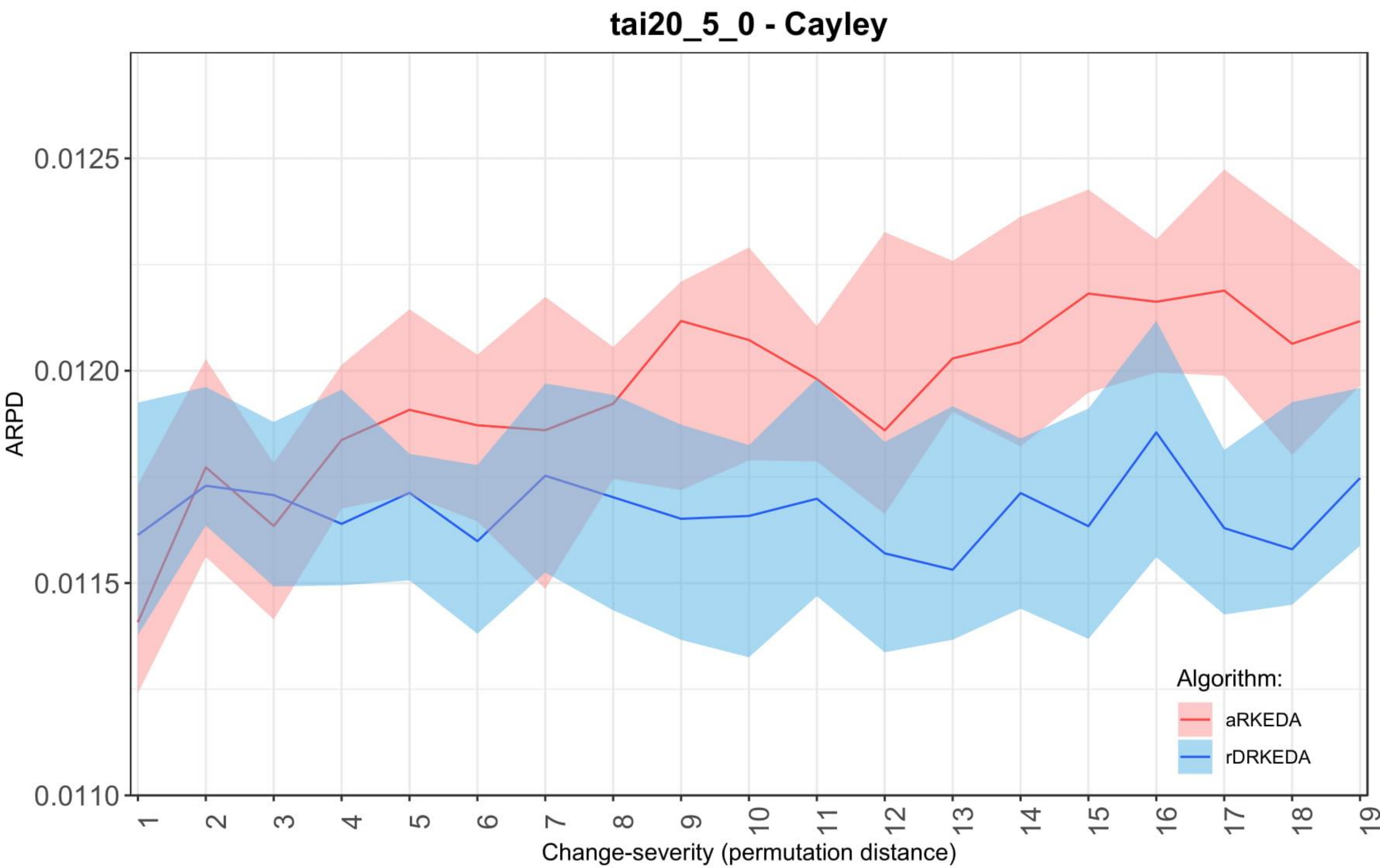
Performance measure
$$ARPD = \frac{1}{G}\sum_{i \in G}\frac{f_i(best) - Best\ known}{Best\ known}.$$

Definition: *elusiveness.*

Being ***P*** *a* ***series of static problems, A*** *an* ***algorithm*** *and* ***A****^{r} the* ***restarting*** *version of the algorithm. Then, we say that* ***P is elusive iff***

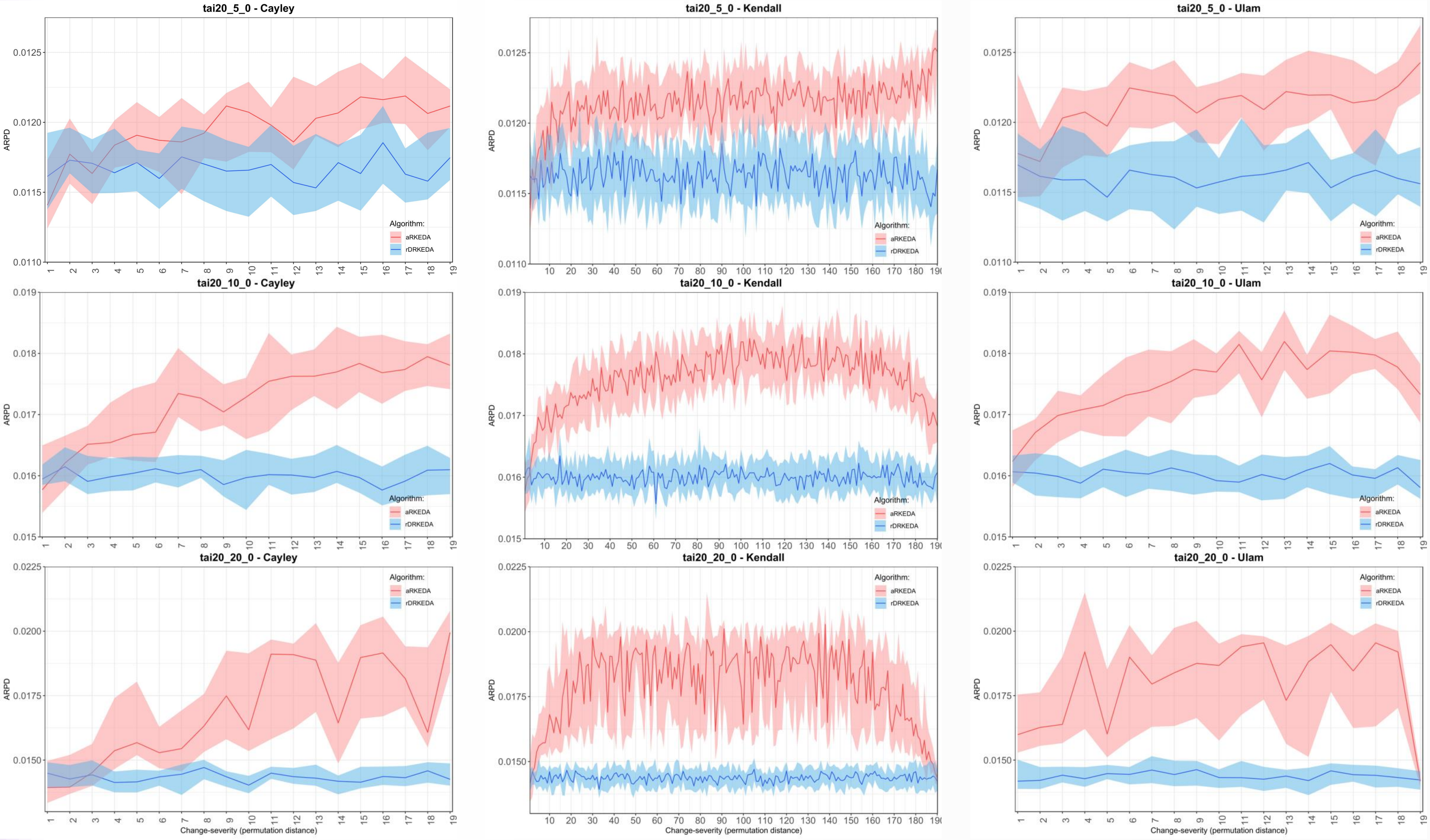
$$\boldsymbol{E[m(A^r, P) - m(A, P)] \leq \tau}.$$

Results



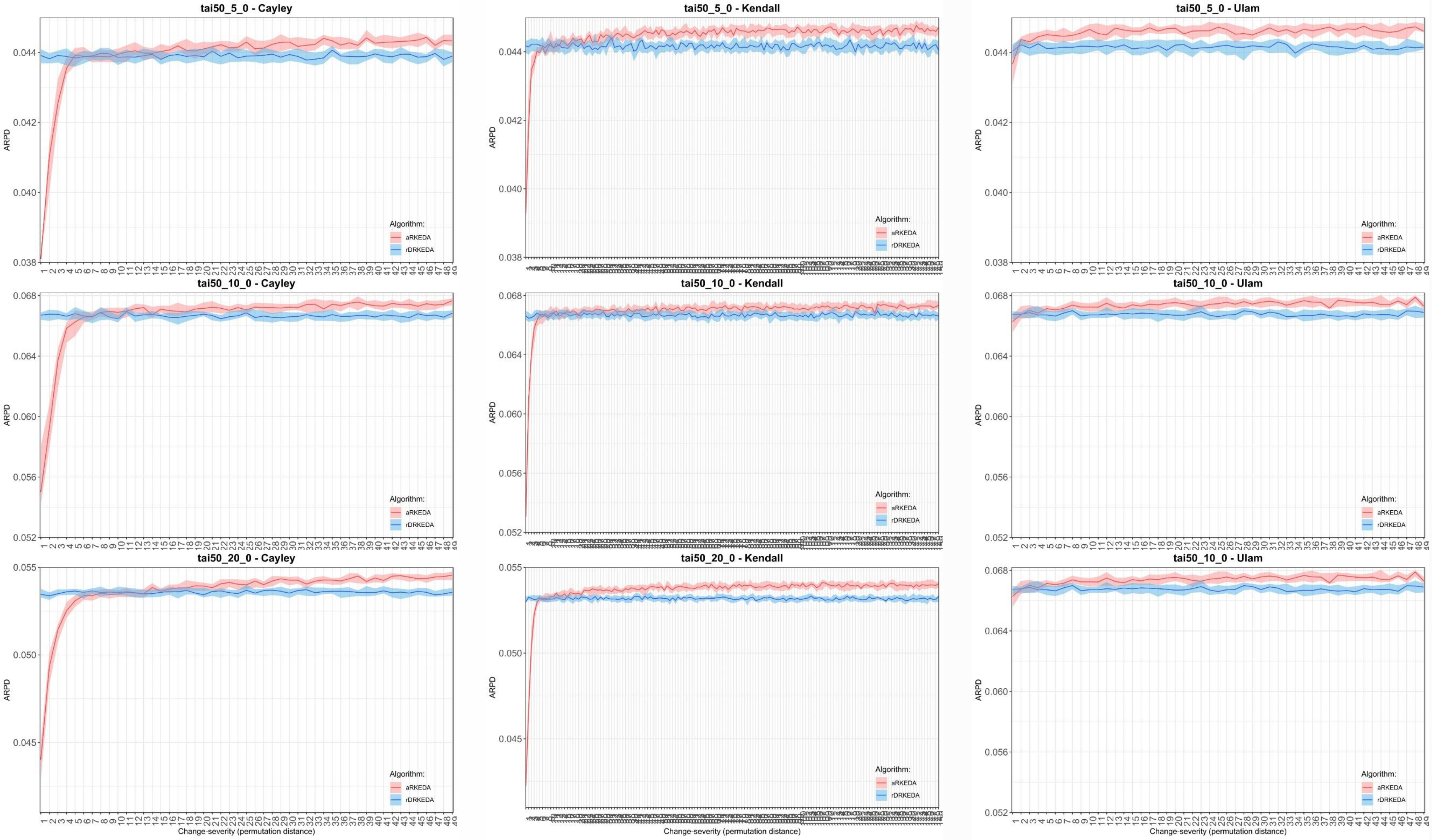
tai20_5_0 - Cayley
ARPD
0.0125
0.0120
0.0115
0.0110
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA

Results



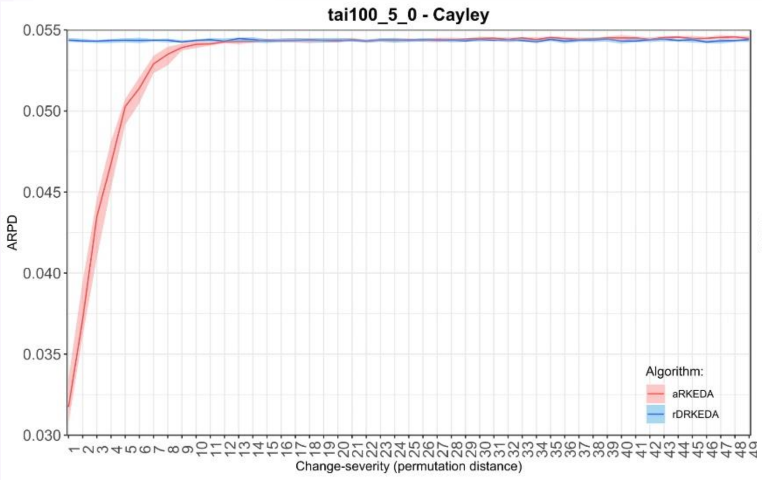
tai20_5_0 - Cayley
tai20_5_0 - Kendall
tai20_5_0 - Ulam
tai20_10_0 - Cayley
tai20_10_0 - Kendall
tai20_10_0 - Ulam
tai20_20_0 - Cayley
tai20_20_0 - Kendall
tai20_20_0 - Ulam
ARPD
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA

Results

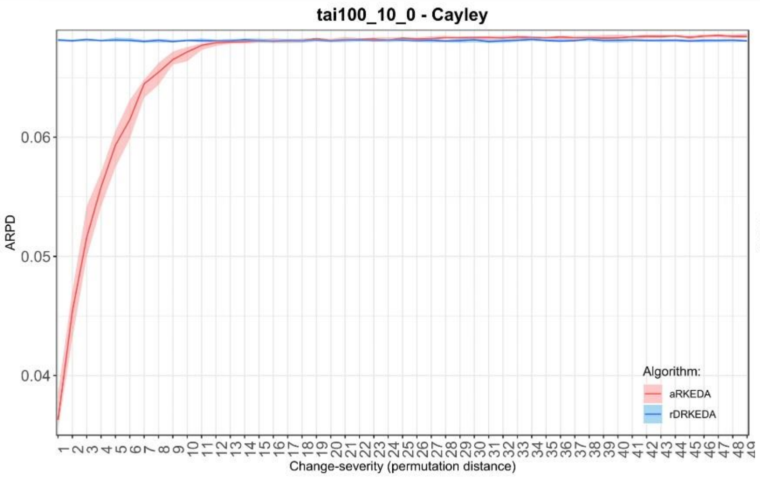


tai50_5_0 - Cayley
tai50_5_0 - Kendall
tai50_5_0 - Ulam
tai50_10_0 - Cayley
tai50_10_0 - Kendall
tai50_10_0 - Ulam
tai50_20_0 - Cayley
tai50_20_0 - Kendall
tai50_10_0 - Ulam
ARPD
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA

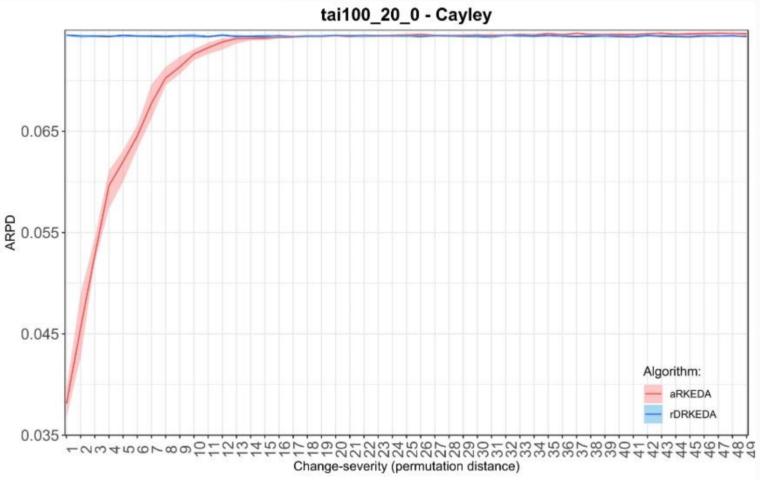
Results



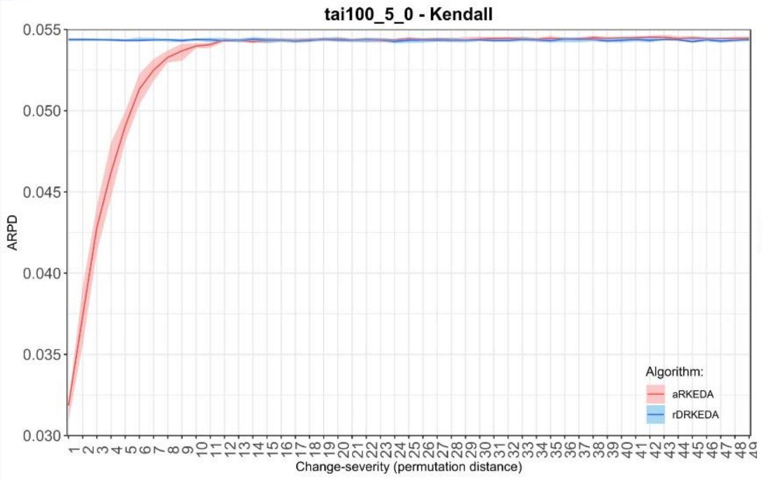
tai100_5_0 - Cayley
ARPD
0.055
0.050
0.045
0.040
0.035
0.030
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA



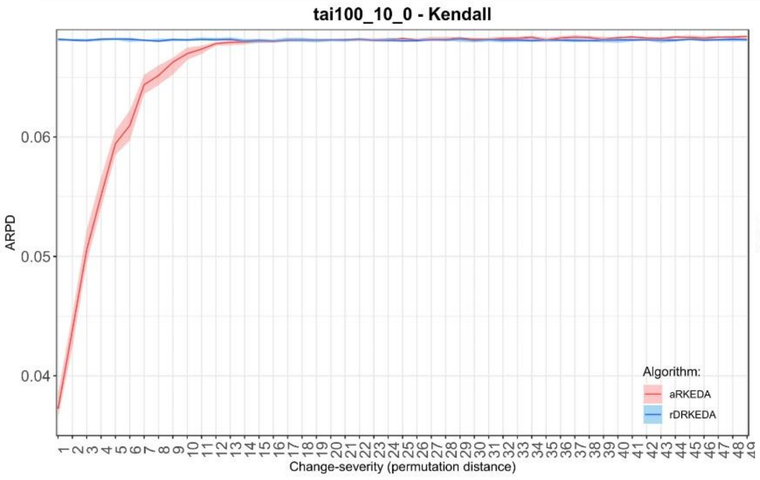
tai100_10_0 - Cayley
ARPD
0.06
0.05
0.04
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA



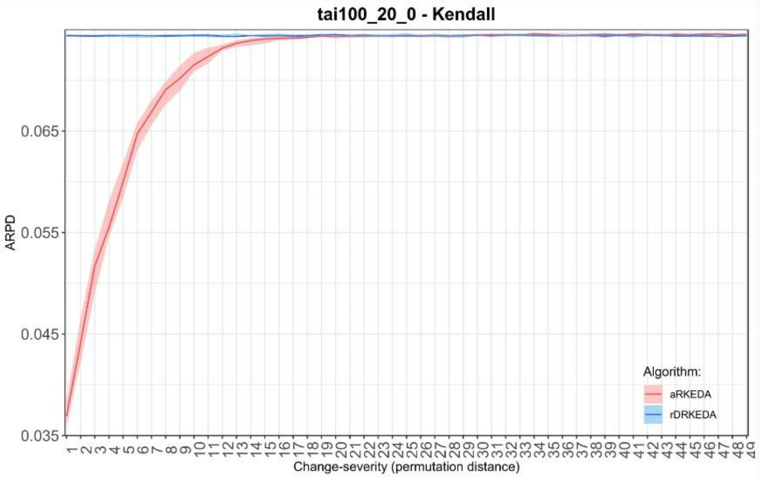
tai100_20_0 - Cayley
ARPD
0.065
0.055
0.045
0.035
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA



tai100_5_0 - Kendall
ARPD
0.055
0.050
0.045
0.040
0.035
0.030
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA



tai100_10_0 - Kendall
ARPD
0.06
0.05
0.04
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA



tai100_20_0 - Kendall
ARPD
0.065
0.055
0.045
0.035
Change-severity (permutation distance)
Algorithm:
aRKEDA
rDRKEDA

Results

Jobs	Cayley			Kendall's-τ			Ulam		
20	2	1	2	2	3	3	0	0	0
50	7	7	12	14	7	8	0	2	1
100	19	15	20	16	19	26	-	-	-
Machines	5	10	20	5	10	20	5	10	20

Number of times in which aRKEDA outperformed rRKEDA.

Results

Jobs	Cayley			Kendall's-τ			Ulam		
20	10%	5%	10%	1.05%	1.58%	1.58%	0%	0%	0%
50	14%	14%	24%	1.14%	0.57%	0.65%	0%	4%	2%
100	19%	15%	20%	0.32%	0.38%	0.52%	-	-	-
Machines	5	10	20	5	10	20	5	10	20

Percentage in which the generated problem should be considered DOP.

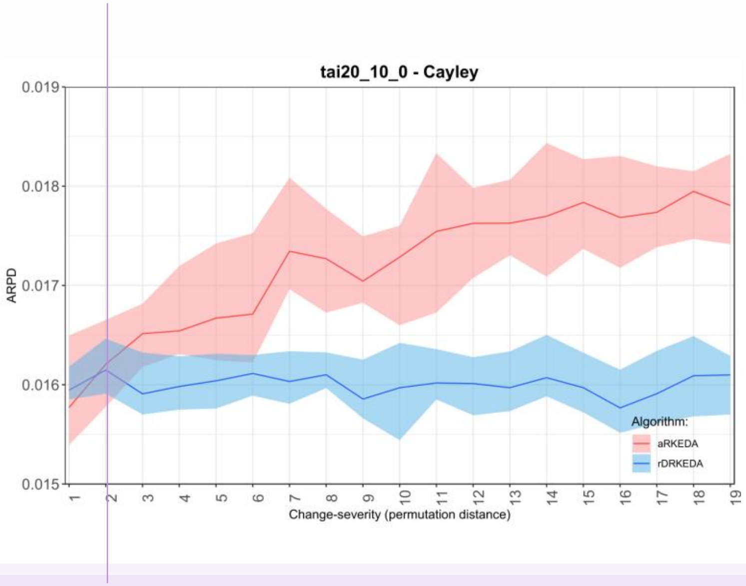
Discussion

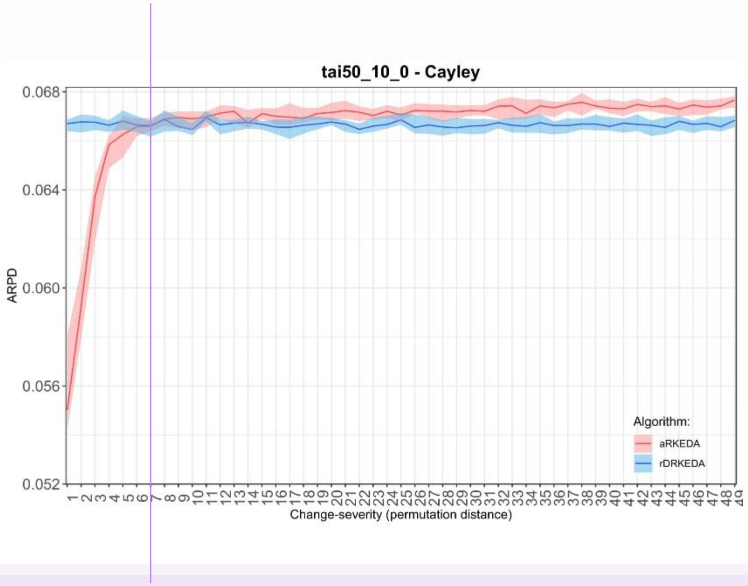
- Restarting the algorithm best option almost always.
  - Surprisingly, only in few cases is beneficial reusing previous knowledge.

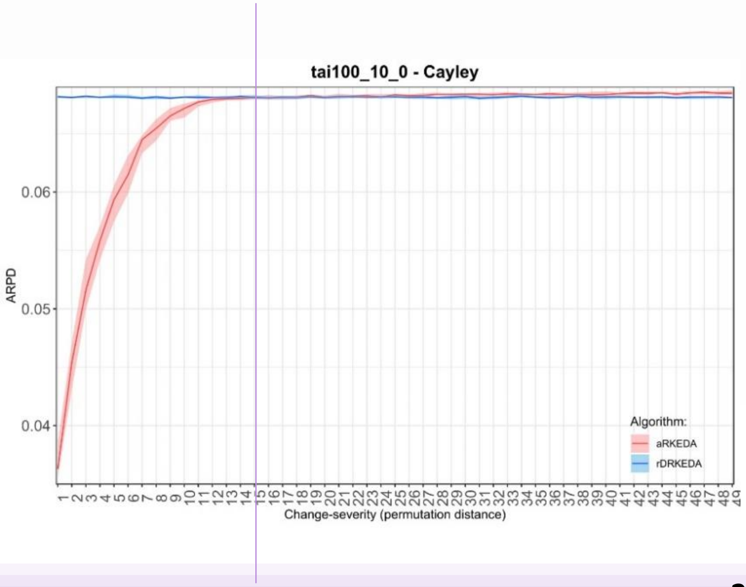
Jobs	Cayley			Kendall's-τ			Ulam		
20	10%	5%	10%	1.05%	1.58%	1.58%	0%	0%	0%
50	14%	14%	24%	1.14%	0.57%	0.65%	0%	4%	2%
100	19%	15%	20%	0.32%	0.38%	0.52%	-	-	-
Machines	5	10	20	5	10	20	5	10	20

Discussion

- The **increase** of the **problem size extends** the **preference of** using the **aRKEDA** for slightly changing problem.

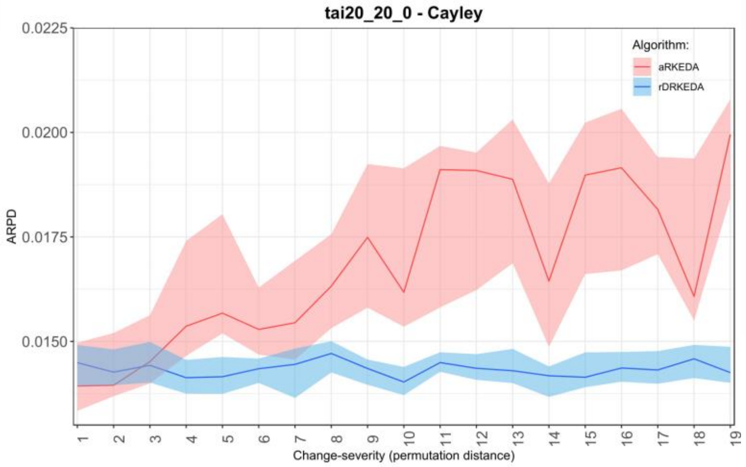


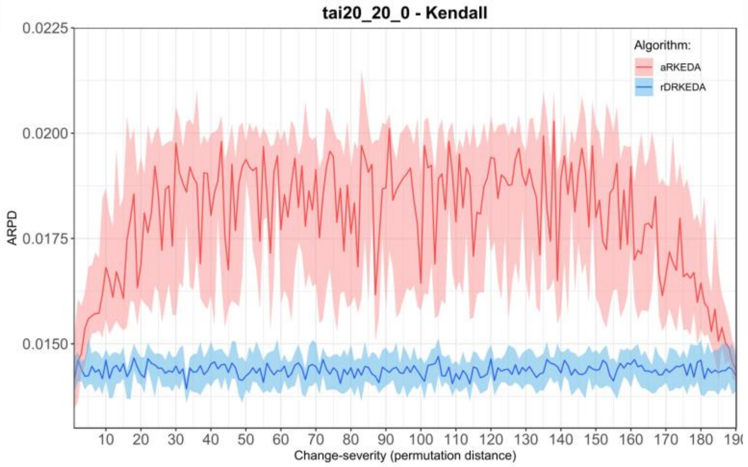


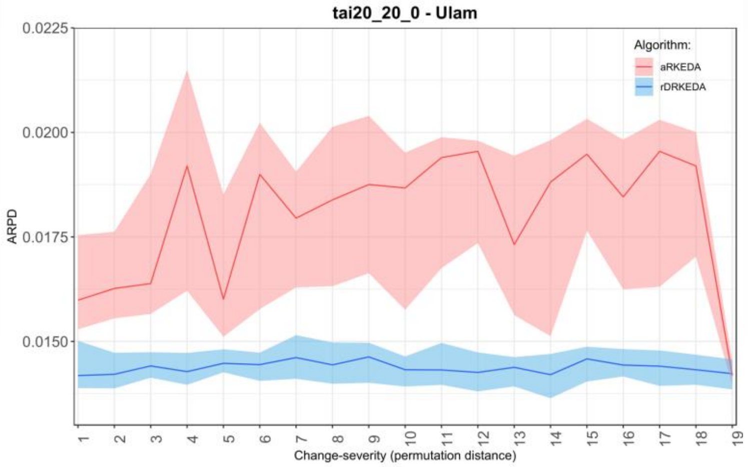


Discussion

- The **increase** of the **machines** produces a **chaotic behaviour** of aRKEDA**.**
  - Kendall's-τ and Ulam metrics → arc shape on 20x20.







Discussion

Jobs	Ulam		
20	0%	0%	0%
50	0%	4%	2%
100	-	-	-
Machines	5	10	20

Ulam metric has aggressive behaviour for rotation. Why?

σ = 342**1** → 1342 → 1234 = e

Discussion

Jobs	Ulam		
20	0%	0%	0%
50	0%	4%	2%
100	-	-	-
Machines	5	10	20

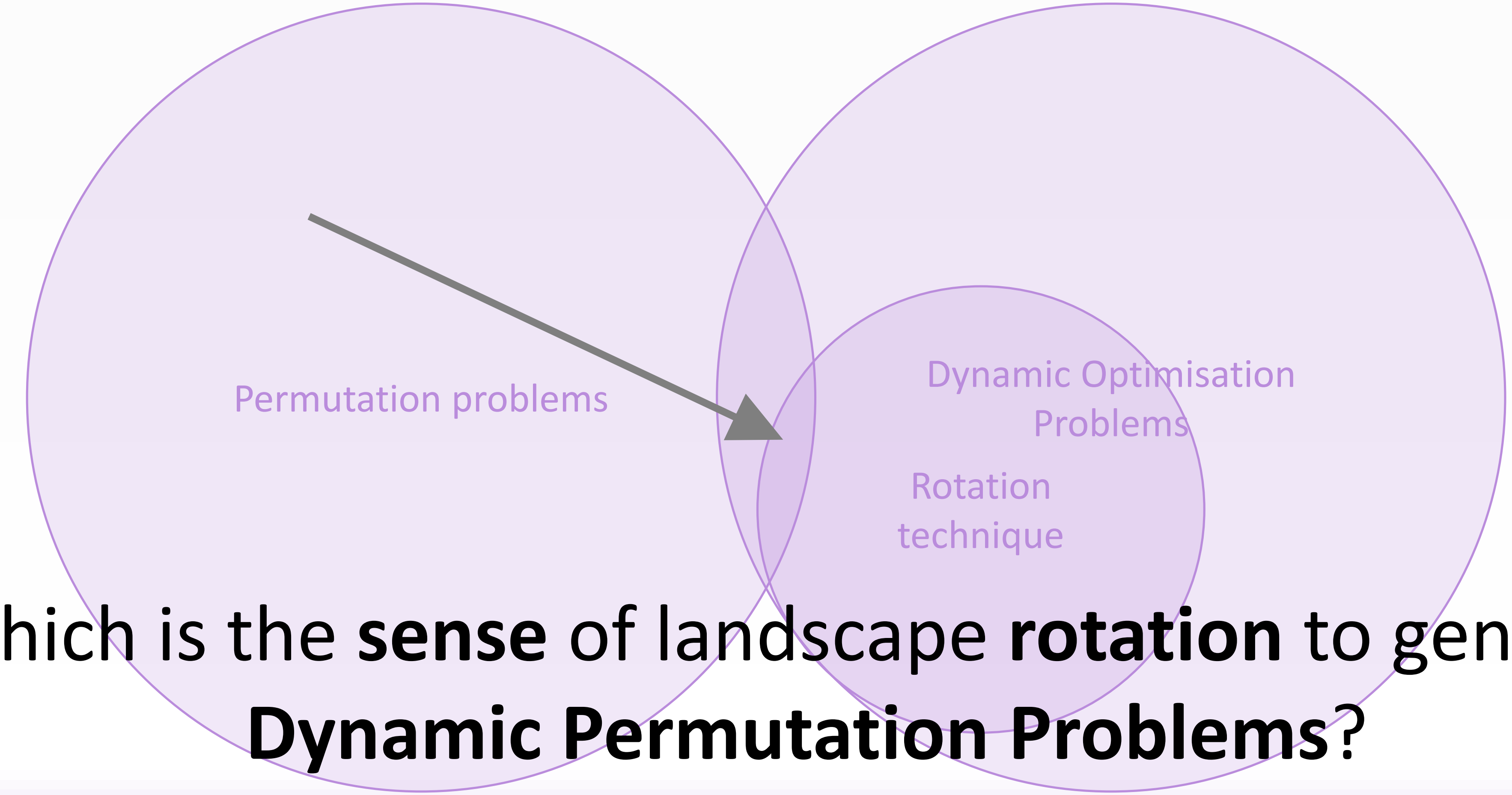
Ulam metric has aggressive behaviour for rotation. Why?

σ = 3421 → 1342 → 1234 = e

Conclusions

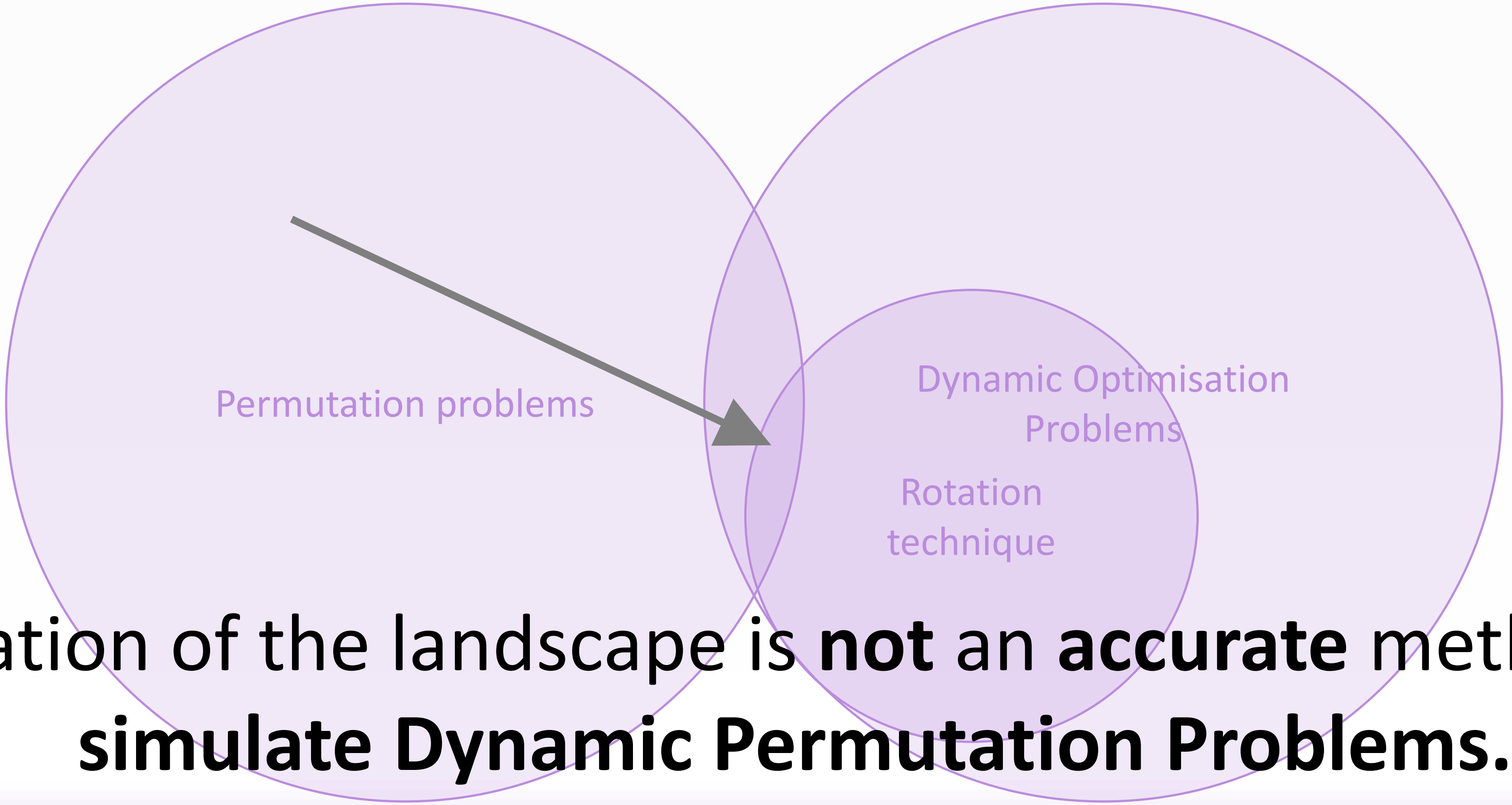
The **definition** of Dynamic Permutation Problems should be **extended**, specially concerning the severity of the change and the algorithm.

Conclusions



Which is the **sense** of landscape **rotation** to generate **Dynamic Permutation Problems**?

Conclusions



Rotation of the landscape is **not** an **accurate** method **to simulate Dynamic Permutation Problems.**

Thank you

for listening!

On the Definition of Dynamic Permutation Problems under Landscape Rotation

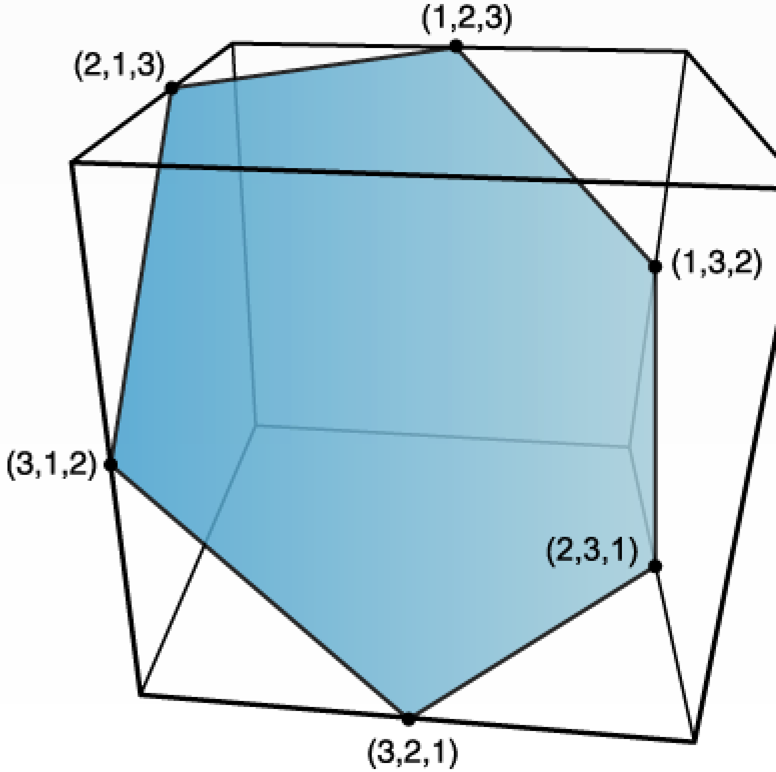
Joan Alza, Mark Bartlett, Josu Ceberio and John McCall

Email: j.alza-santos@rgu.ac.uk

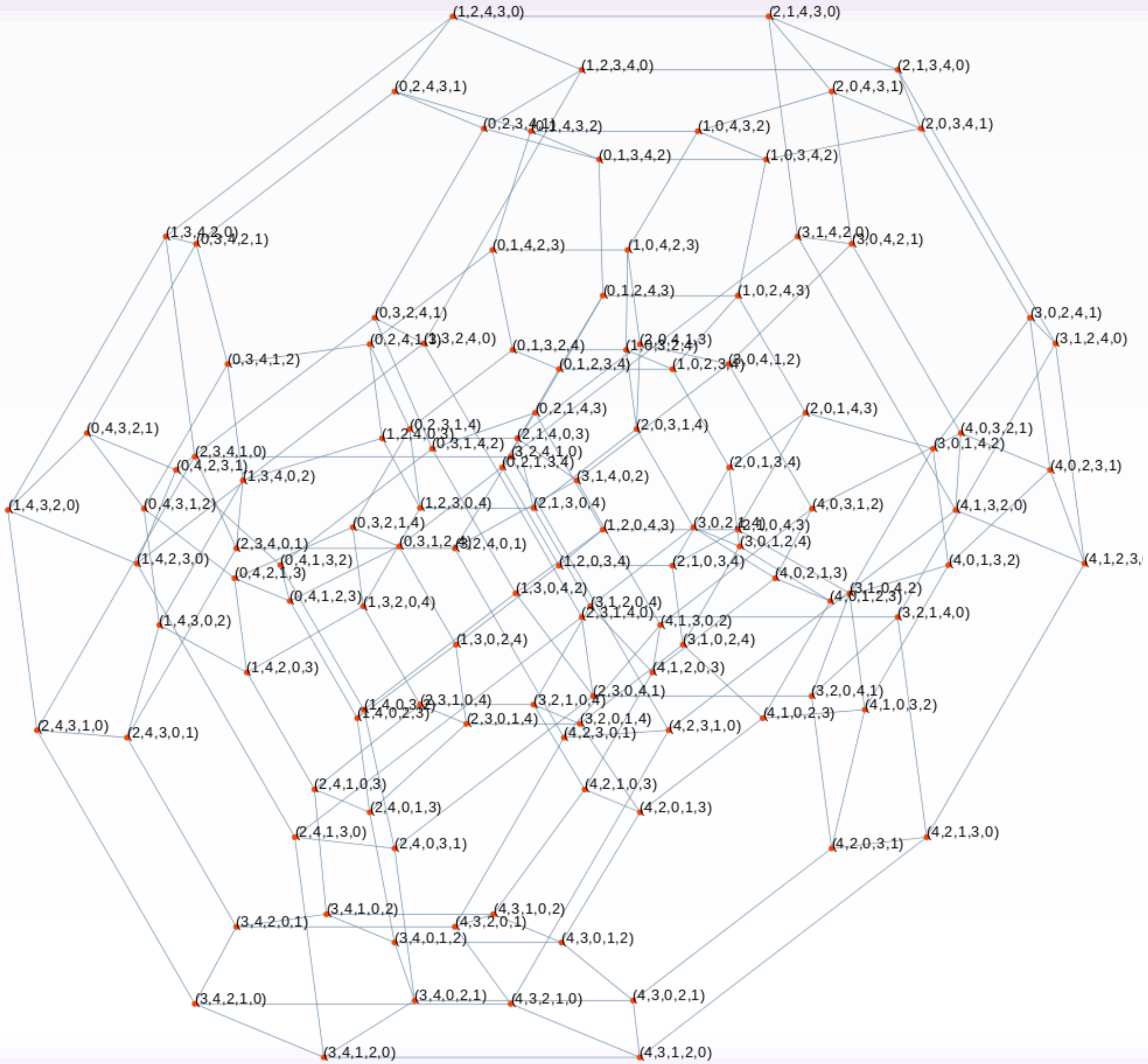
ROBERT GORDON UNIVERSITY ABERDEEN

Universidad del País Vasco
Euskal Herriko Unibertsitatea
The University of the Basque Country

Permutohedron



https://commons.wikimedia.org/wiki/File:Permutohedron_order_3.svg



https://en.wikipedia.org/wiki/Permutohedron#/media/File:Omnitruncated_5Cell_as_Permutohedron.svg

Kendall's-τ

σ = 3421 e = 1234

Pairs	Disagreements
$1 \prec 2$	X
$1 \prec 3$	X
$1 \prec 4$	X
$2 \prec 3$	X
$2 \prec 4$	X
$3 \prec 4$	

5

$$\sigma^{-1} = 4312 \xrightarrow{1} 4132 \xrightarrow{2} 1432 \xrightarrow{3} 1423 \xrightarrow{4} 1243 \xrightarrow{5} 1234 = e$$