Karlsruhe Institute of Technology

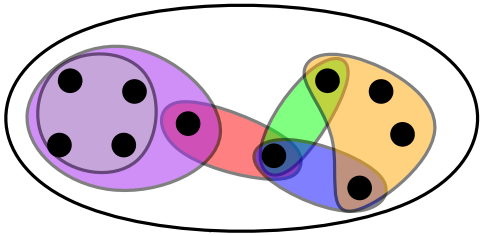
Evolutionary Hypergraph Partitioning

Presentation · December 7, 2017

Robin Andre

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

KaHyPar - Multilevel Partitioner

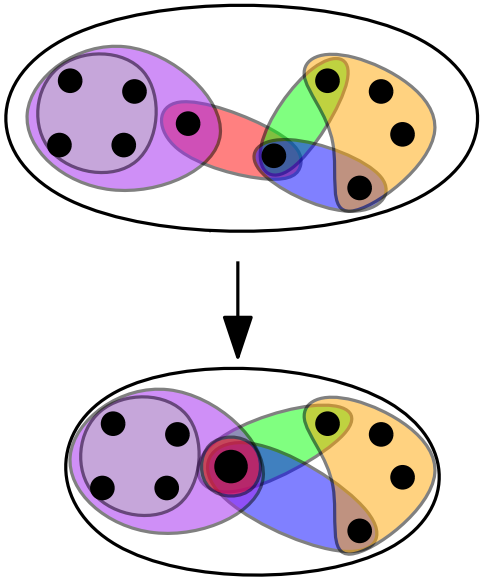


Coarsening:

- H is reduced to a smaller problem H_C
- only one node is contracted per step
- until H_C is sufficiently small

Institute of Theoretical Informatics
Algorithmics Group

KaHyPar - Multilevel Partitioner

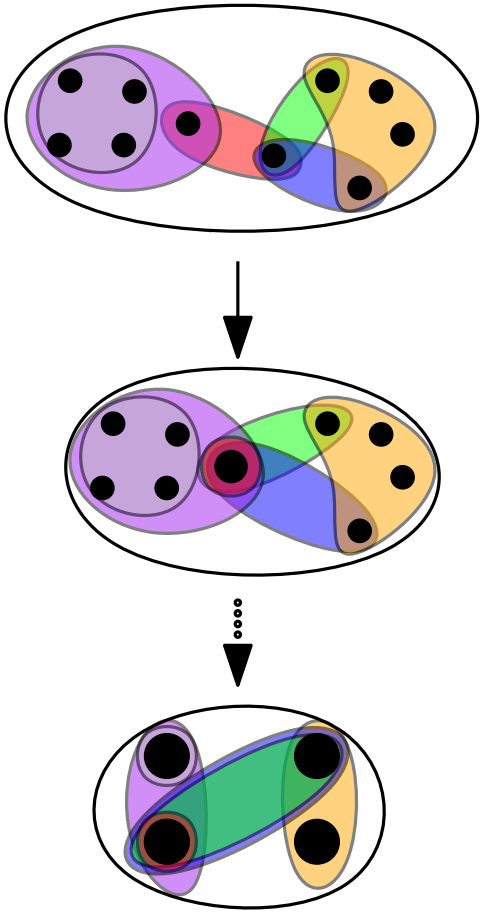


Coarsening:

- H is reduced to a smaller problem H_C
- only one node is contracted per step
- until H_C is sufficiently small

Institute of Theoretical Informatics
Algorithmics Group

KaHyPar - Multilevel Partitioner

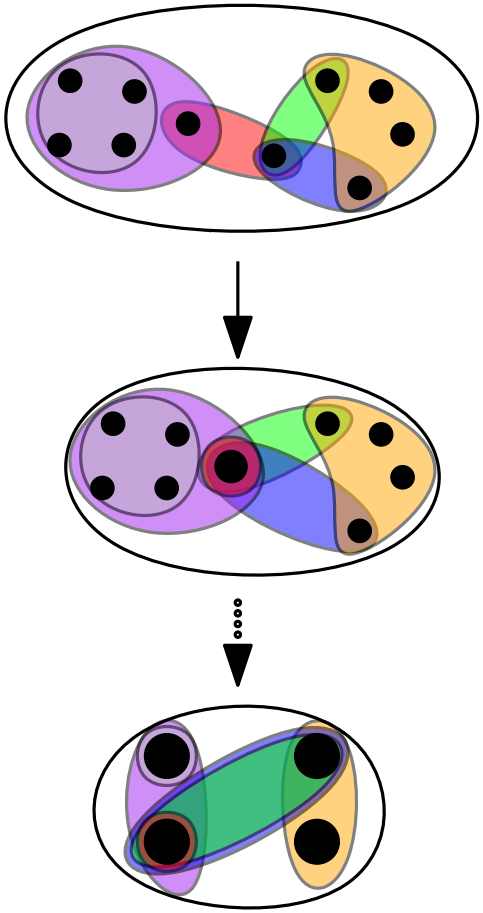


Coarsening:

- H is reduced to a smaller problem H_C
- only one node is contracted per step
- until H_C is sufficiently small

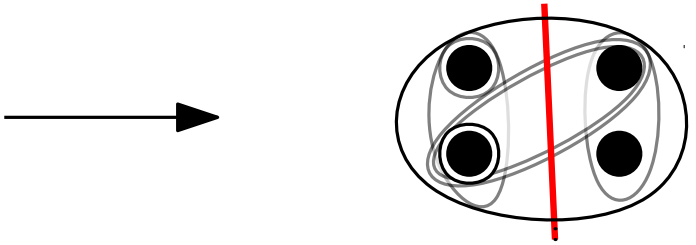
Institute of Theoretical Informatics
Algorithmics Group

KaHyPar - Multilevel Partitioner



Initial Partitioning:

- An algorithm generates an Initial Partitioning for H_C

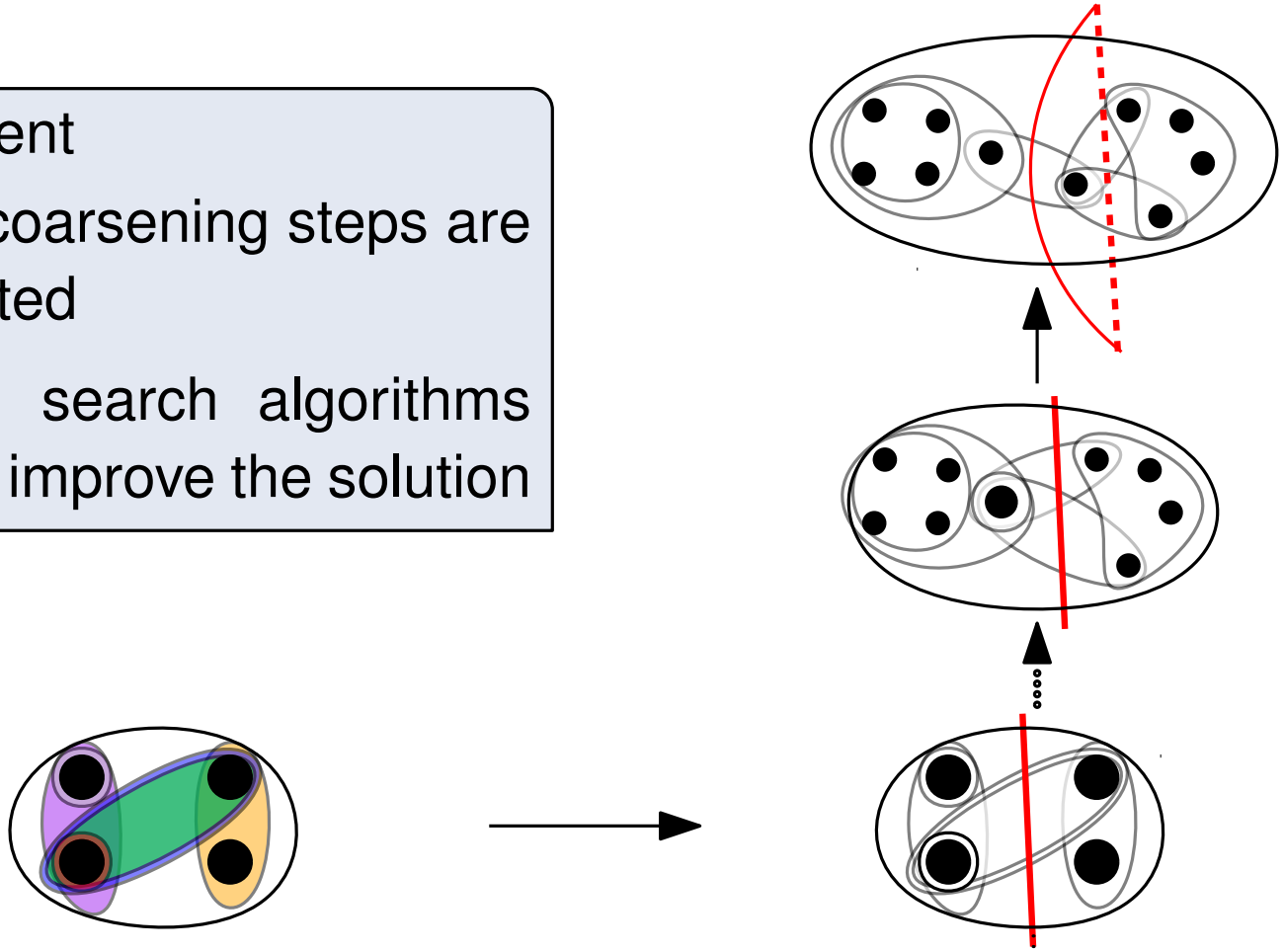


Institute of Theoretical Informatics
Algorithmics Group

KaHyPar - Multilevel Partitioner

Refinement

- The coarsening steps are reverted
- Local search algorithms try to improve the solution



Institute of Theoretical Informatics
Algorithmics Group

Evolutionary Algorithm

KaHyPar generates multiple partitions

dynamic allocation $\delta = 15\%$

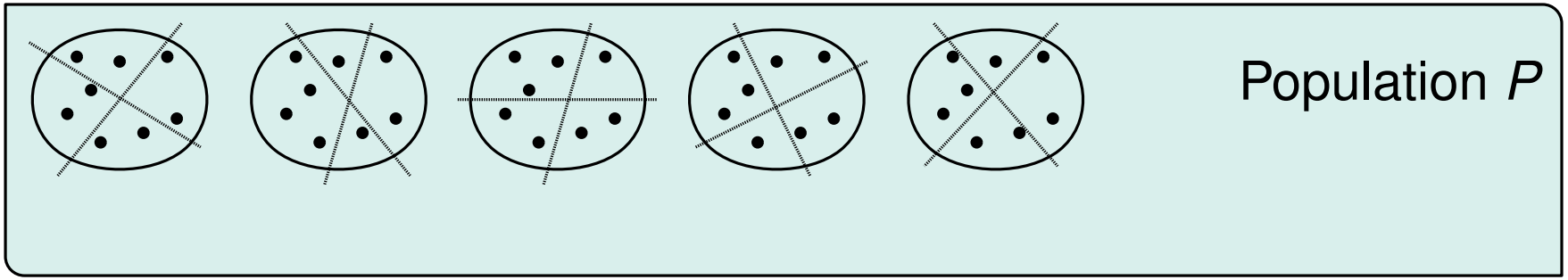
Evolutionary Algorithm

3.1s $time = 100s$ ⟶ $\sim$ 5 iterations

KaHyPar generates multiple partitions

dynamic allocation $\delta = 15\%$

Evolutionary Algorithm



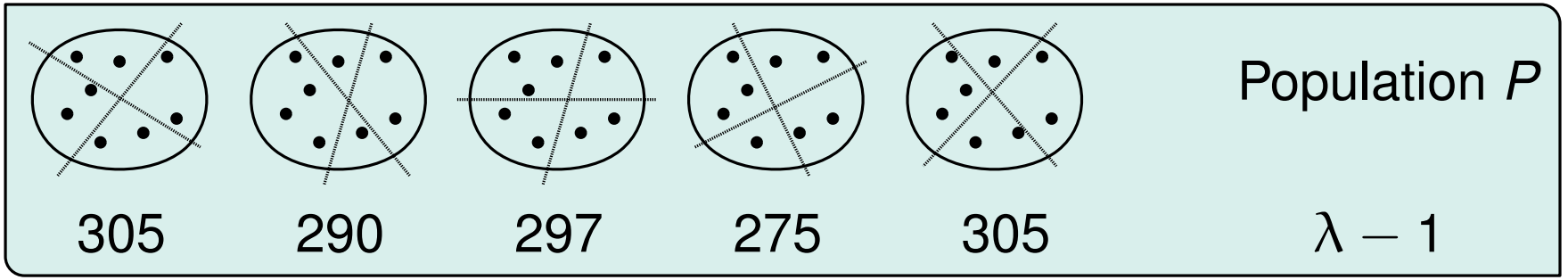
3.1s $time = 100s$ ⟶ $\sim$ 5 iterations

KaHyPar generates multiple partitions

dynamic allocation $\delta = 15\%$

Institute of Theoretical Informatics
Algorithmics Group

Evolutionary Algorithm

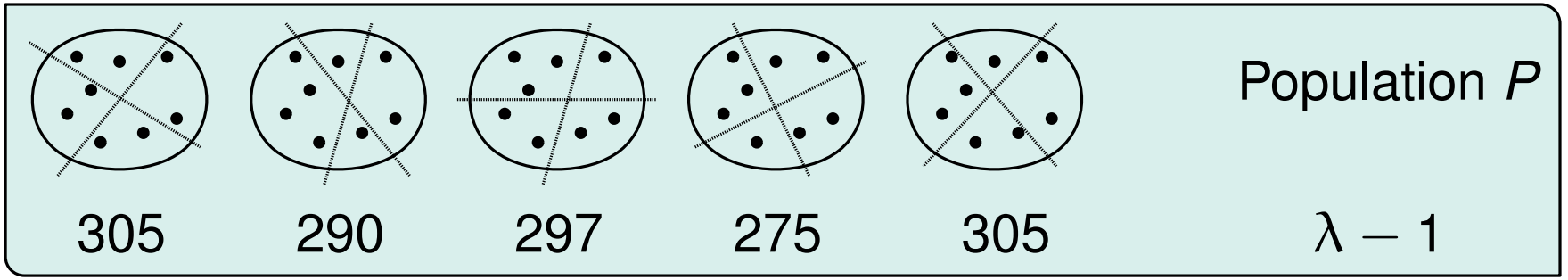


3.1s $time = 100s$ ⟶ $\sim$ 5 iterations

KaHyPar generates multiple partitions

dynamic allocation $\delta = 15\%$

Evolutionary Algorithm



3.1s $time = 100s$ ⟶ $\sim$ 5 iterations

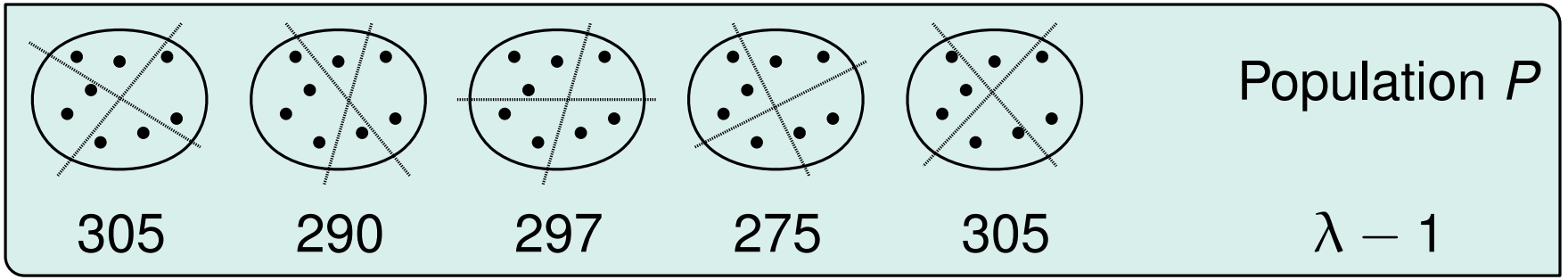
KaHyPar generates multiple partitions

dynamic allocation $\delta = 15\%$

balances time/hypergraph size

Institute of Theoretical Informatics
Algorithmics Group

Evolutionary Algorithm



3.1s $time = 100s$ ⟶ $\sim$ 5 iterations

KaHyPar generates multiple partitions

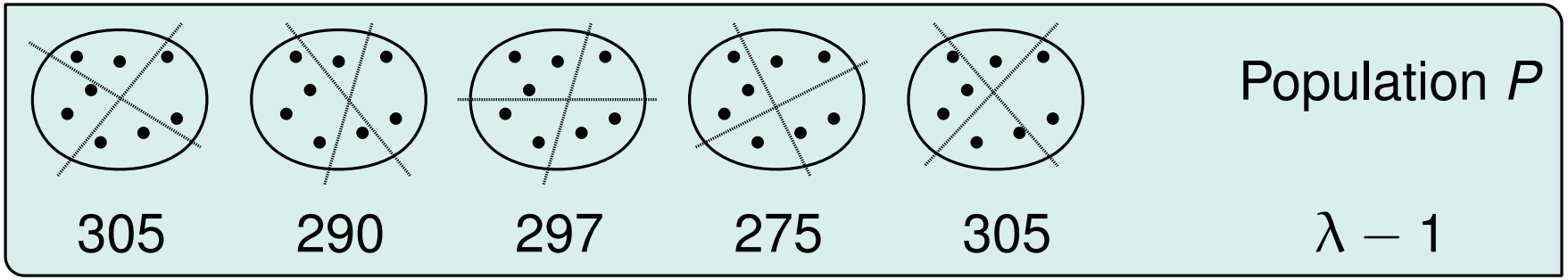
dynamic allocation $\delta = 15\%$

high quality solutions

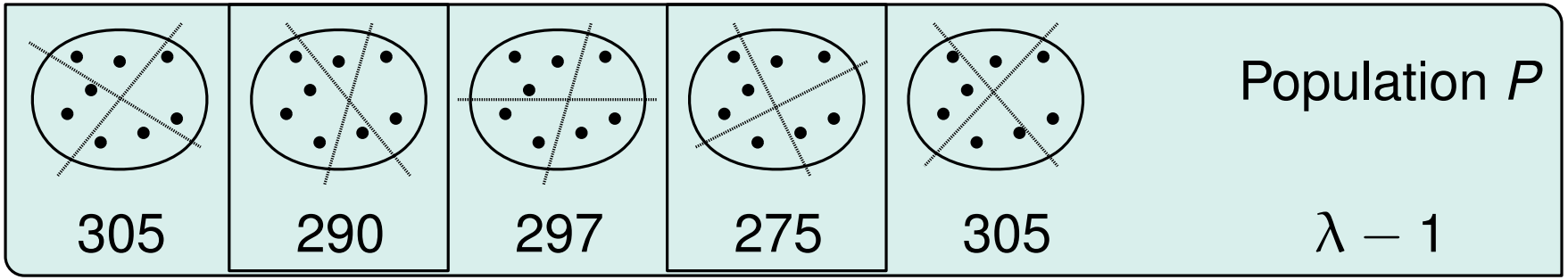
balances time/hypergraph size

Institute of Theoretical Informatics
Algorithmics Group

Tournament Selection

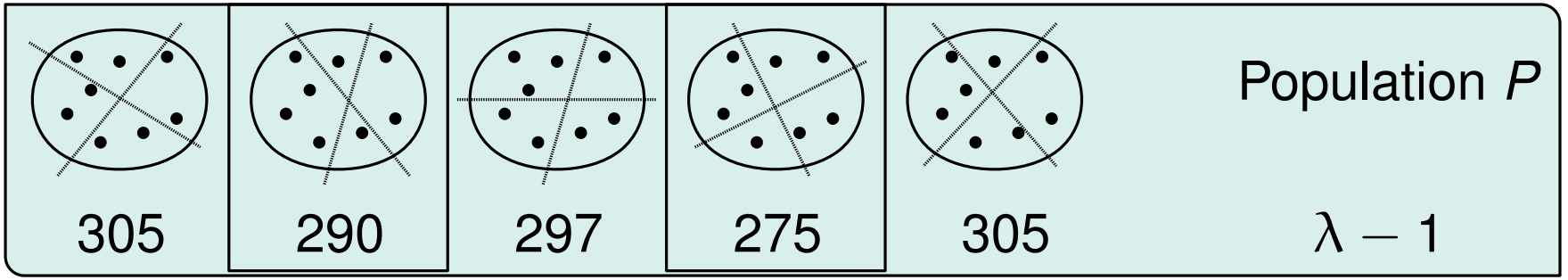


Tournament Selection



select 2 random Individuals

Tournament Selection

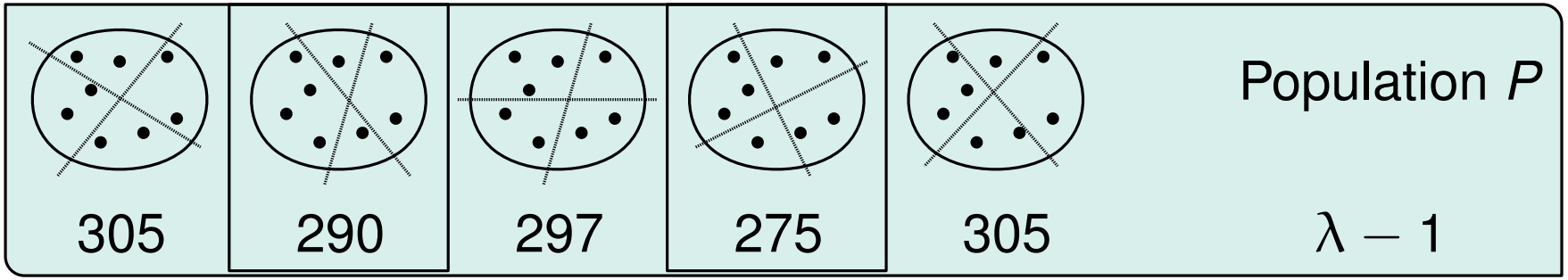


select 2 random Individuals

compare their fitness

Institute of Theoretical Informatics
Algorithmics Group

Tournament Selection

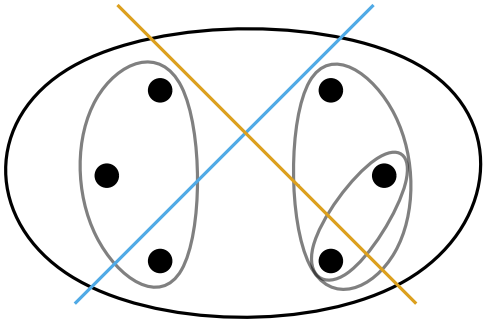


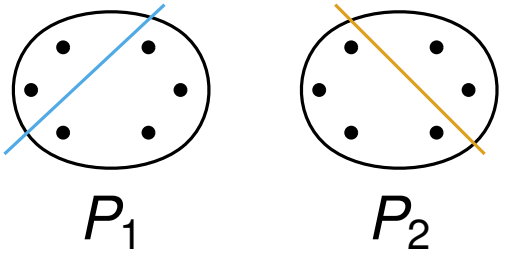
select 2 random Individuals

compare their fitness

choose the better Individual

Combine Operator

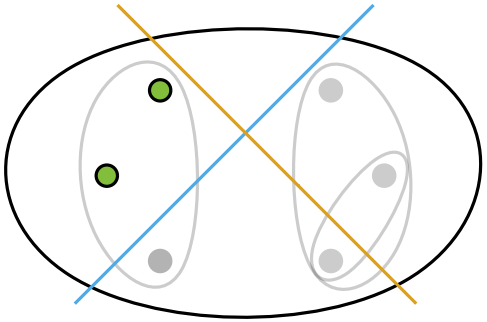


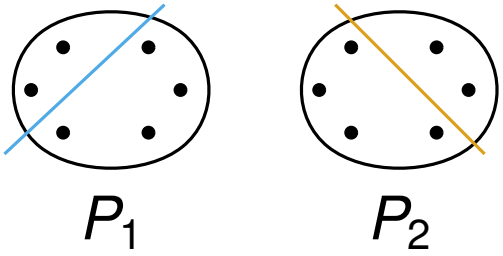


Coarsening:

- contractions must respect P_1 & P_2
- does not change solution quality

Combine Operator



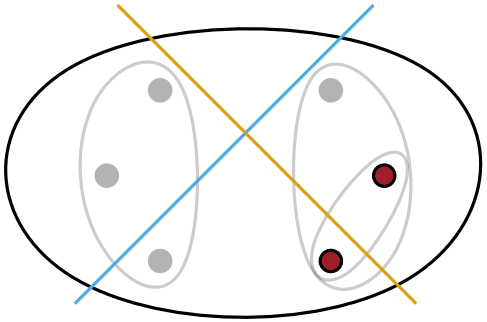


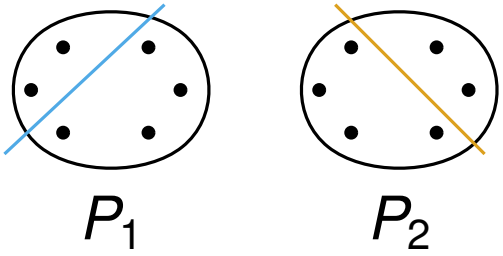
Valid Contraction

Coarsening:

- contractions must respect P_1 & P_2
- does not change solution quality

Combine Operator



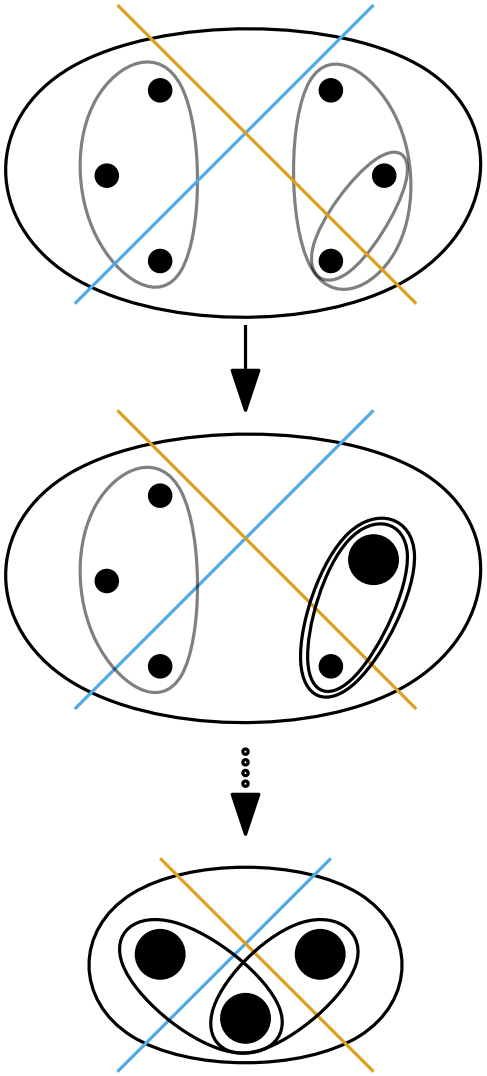


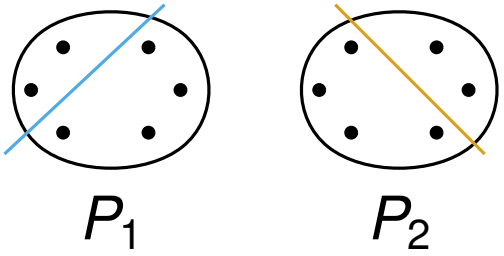
Invalid Contraction

Coarsening:

- contractions must respect P_1 & P_2
- does not change solution quality

Combine Operator

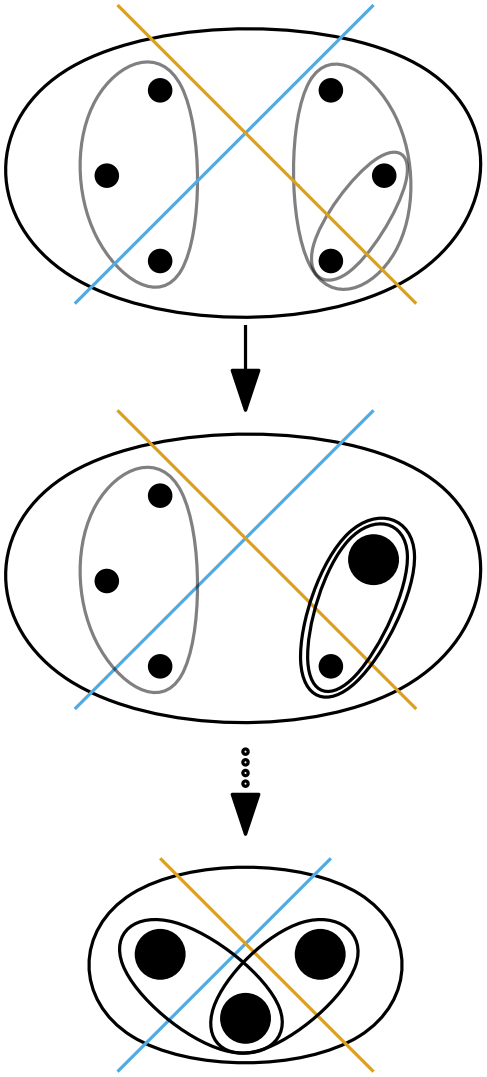


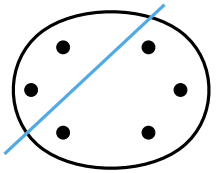


Coarsening:

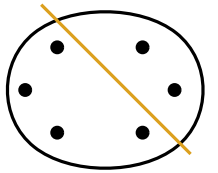
- contractions must respect P_1 & P_2
- does not change solution quality

Combine Operator





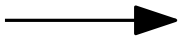
P_1

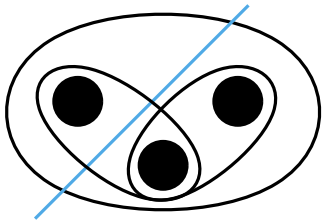


P_2

Initial Partitioning:

- Use the better parent partition (P_1)
- Maintains solution quality

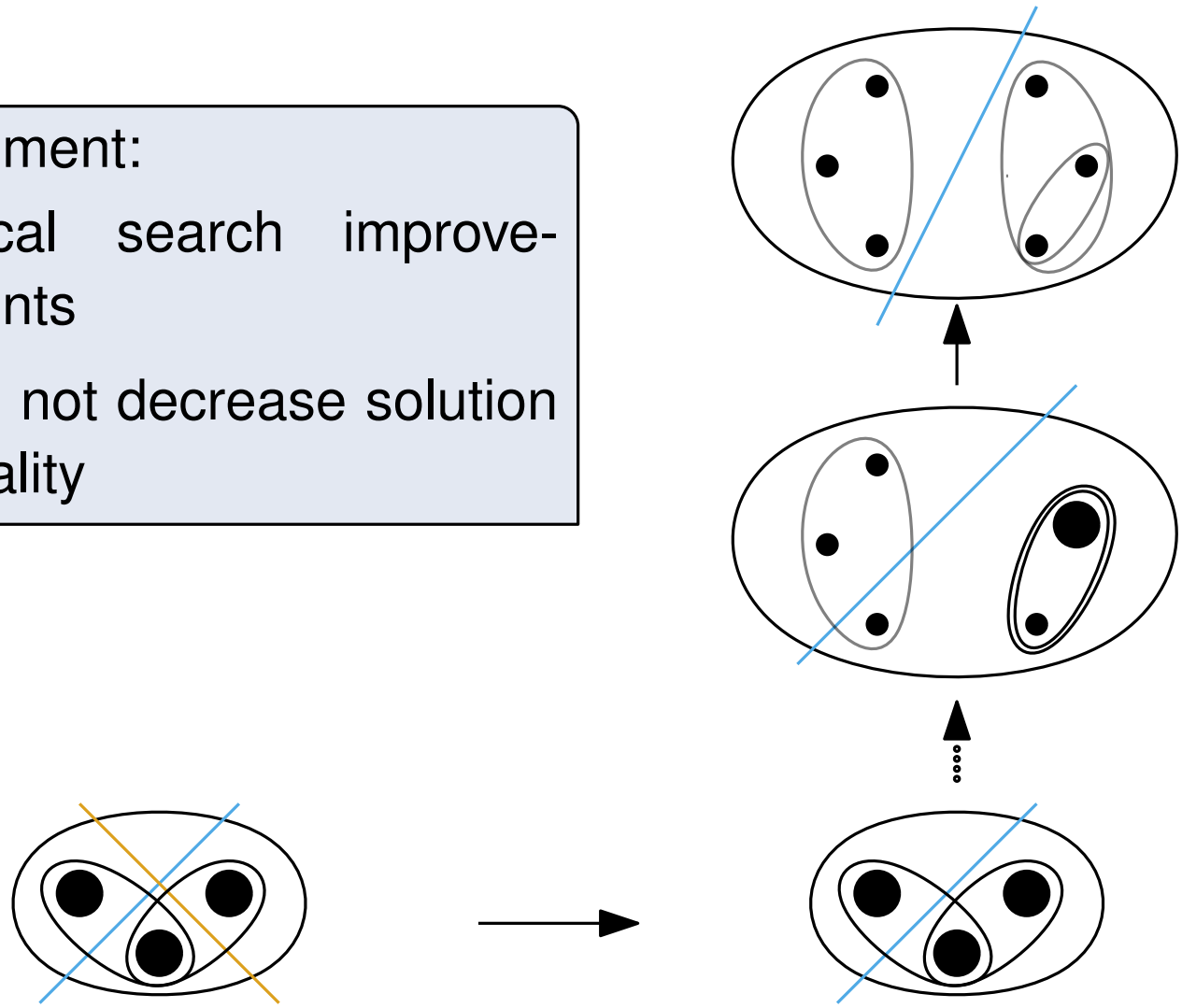




Combine Operator

Refinement:

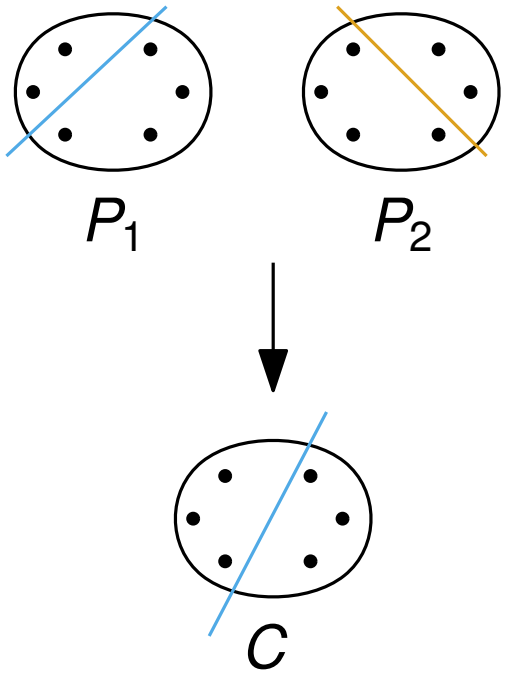
- Local search improvements
- will not decrease solution quality



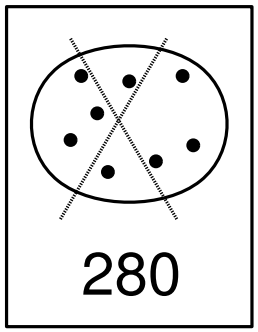
Combine Operator

Benefits:

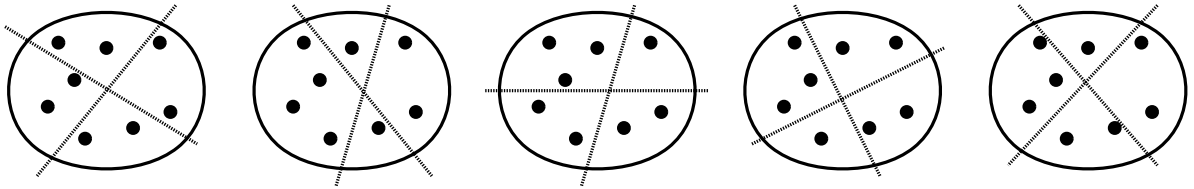
- local search
- structure preservation



Replacement Strategy



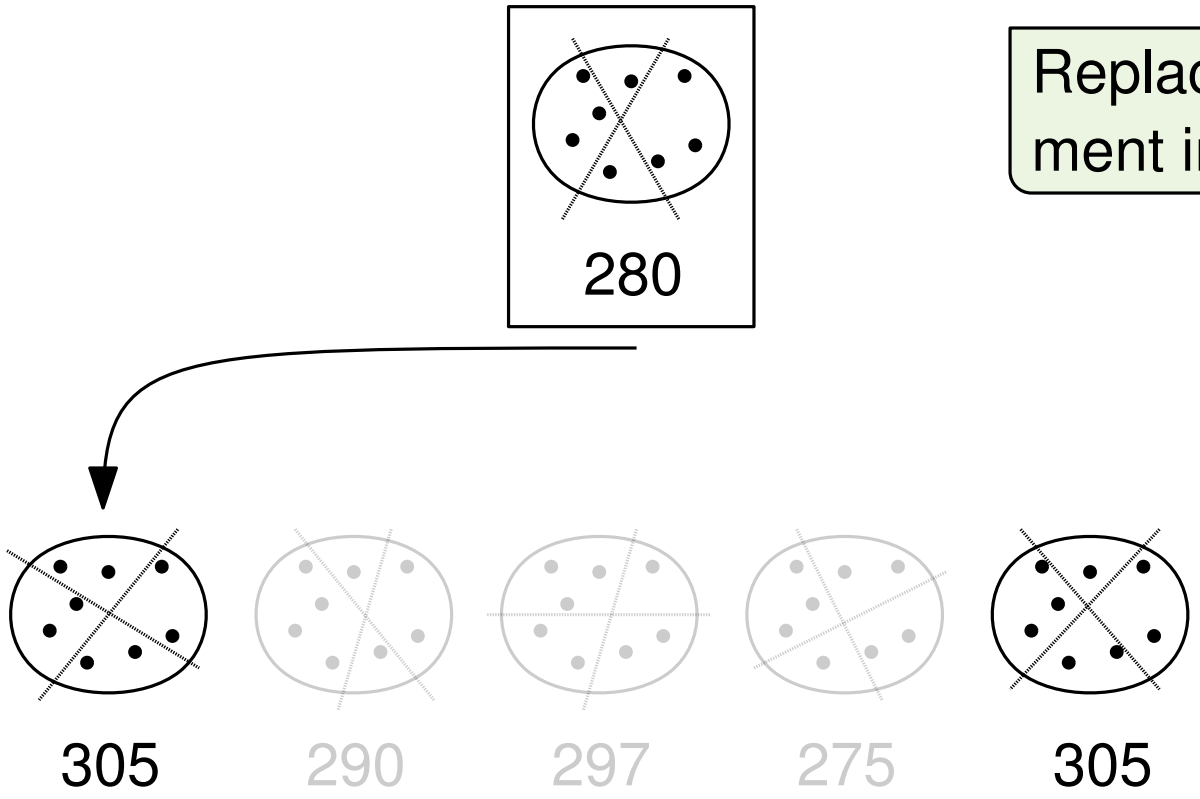
New individual has to be placed in P



Population P

$\lambda - 1$

Replacement Strategy

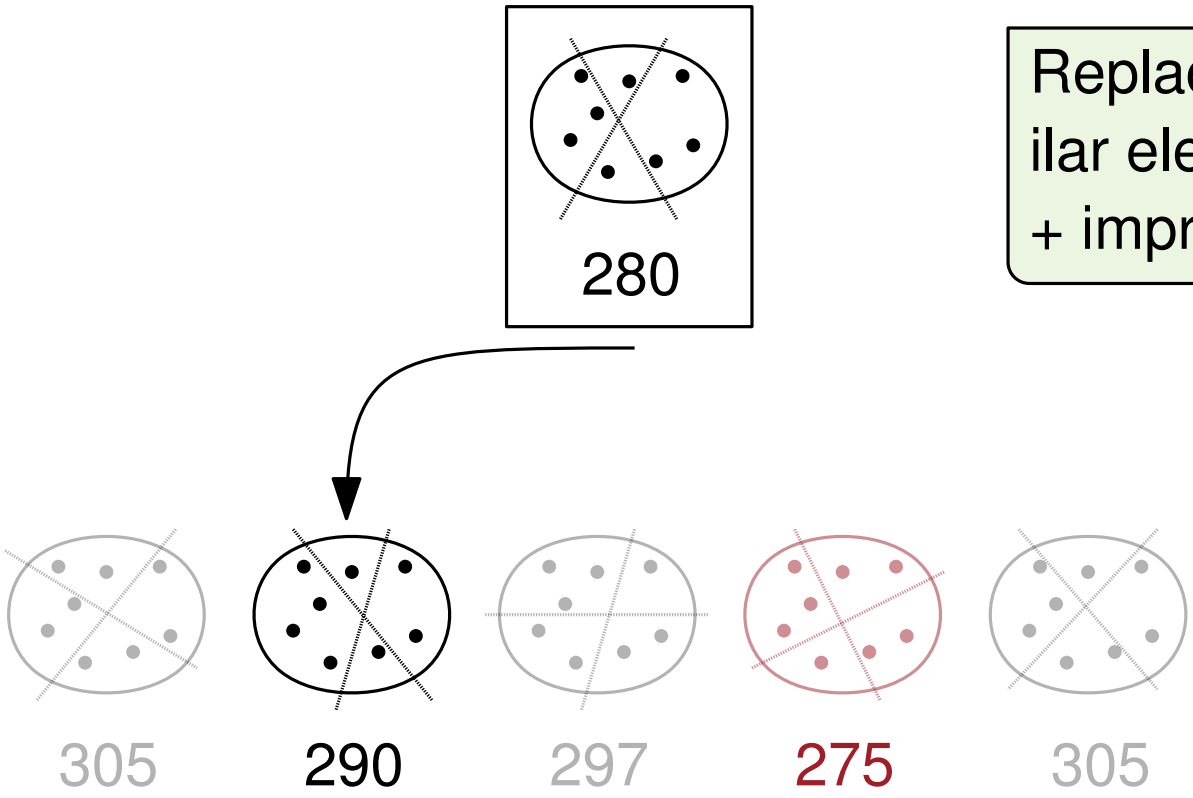


Replace the worst element in P

Population P

$\lambda - 1$

Replacement Strategy

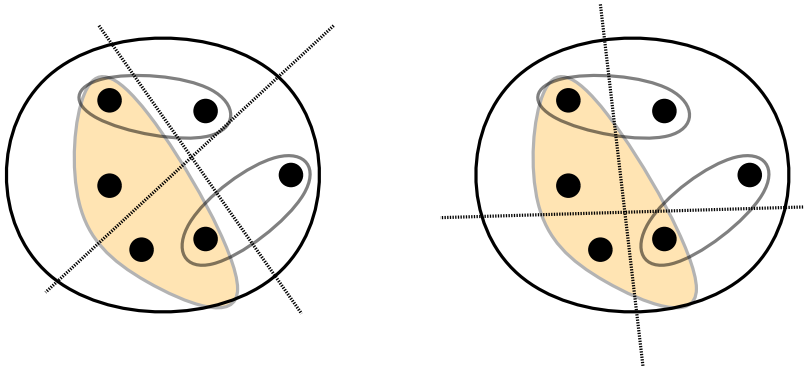


Replace the most similar element in P
+ improving quality

Population P

$\lambda - 1$

Diversity

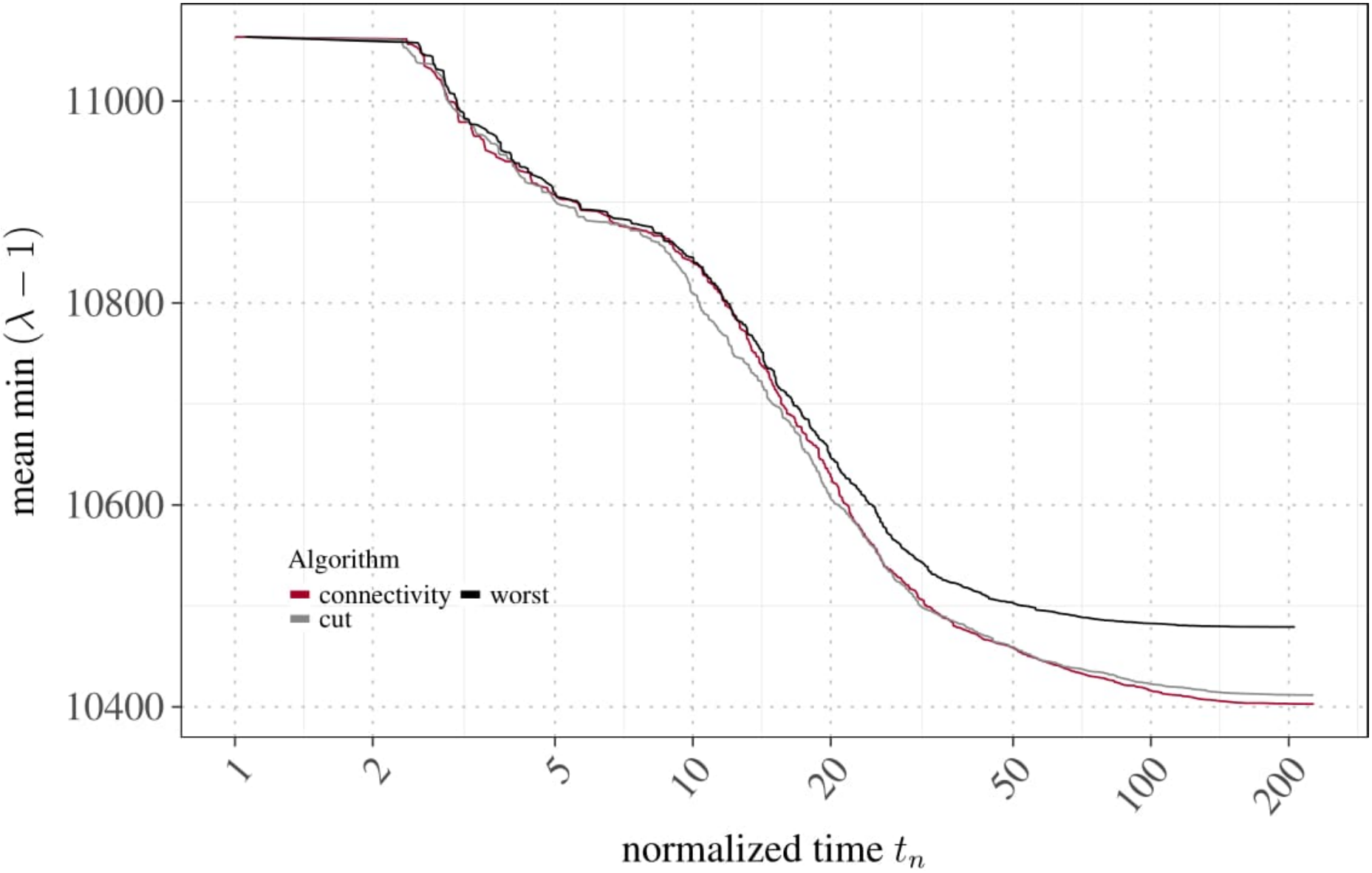


			difference
cut	1, 1, 1	1, 1, 1	0,0,0 = 0
connectivity	2,2,2	2,3,2	0,1,0 = 1

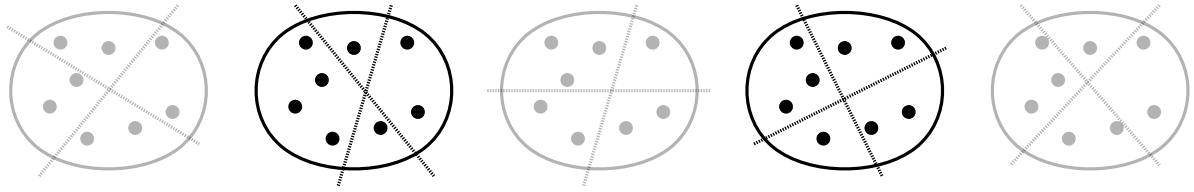
There are two approaches for difference :

- count the different cut edges
- count the different blocks of cut edges

Test Results

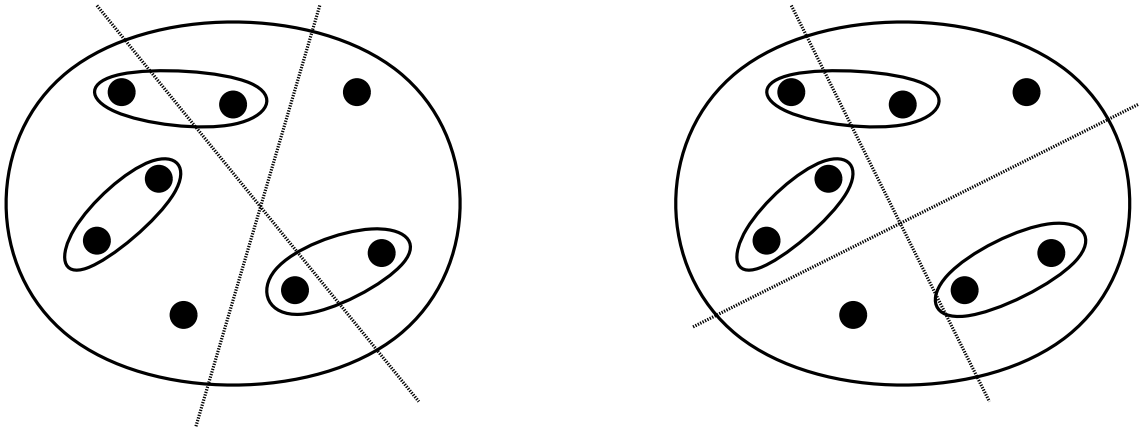


Edge Frequency Multicombine



We inspect the $\sqrt{|P|}$ best individuals of P

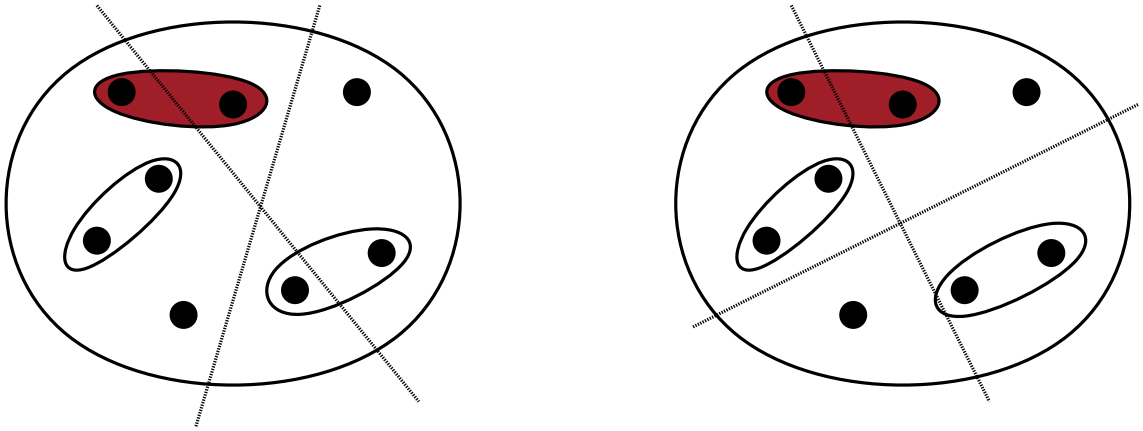
Edge Frequency Multicombine



The frequency of each hyperedge is analyzed

Institute of Theoretical Informatics
Algorithmics Group

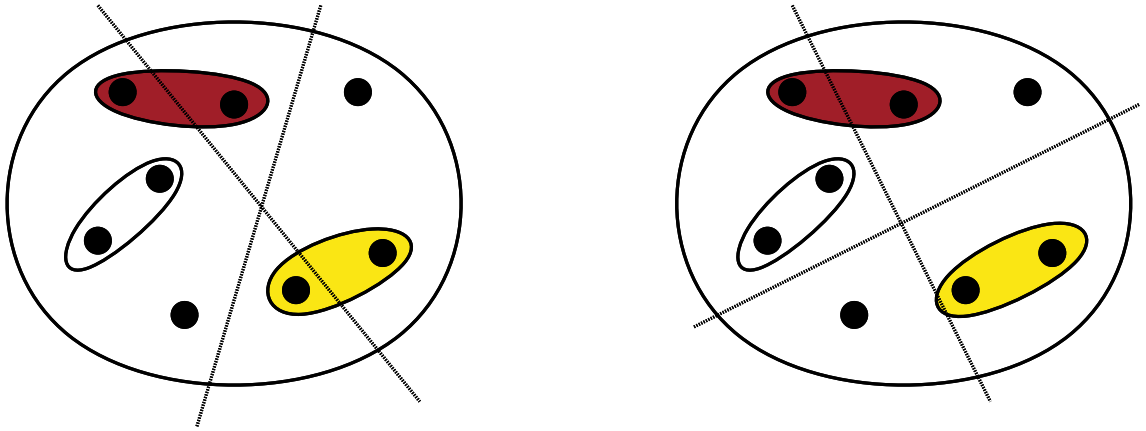
Edge Frequency Multicombine



The frequency of each hyperedge is analyzed

Institute of Theoretical Informatics
Algorithmics Group

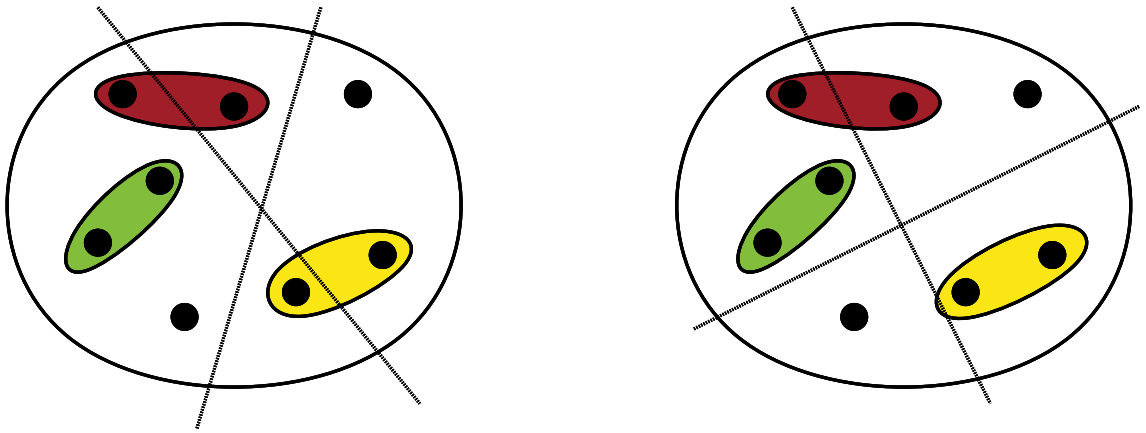
Edge Frequency Multicombine



The frequency of each hyperedge is analyzed

Institute of Theoretical Informatics
Algorithmics Group

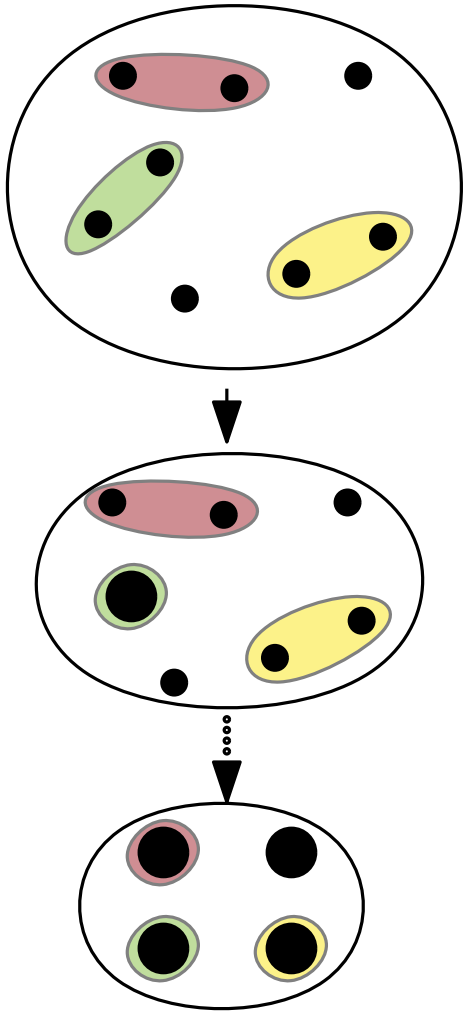
Edge Frequency Multicombine



The frequency of each hyperedge is analyzed

Institute of Theoretical Informatics
Algorithmics Group

Edge Frequency Multicombine



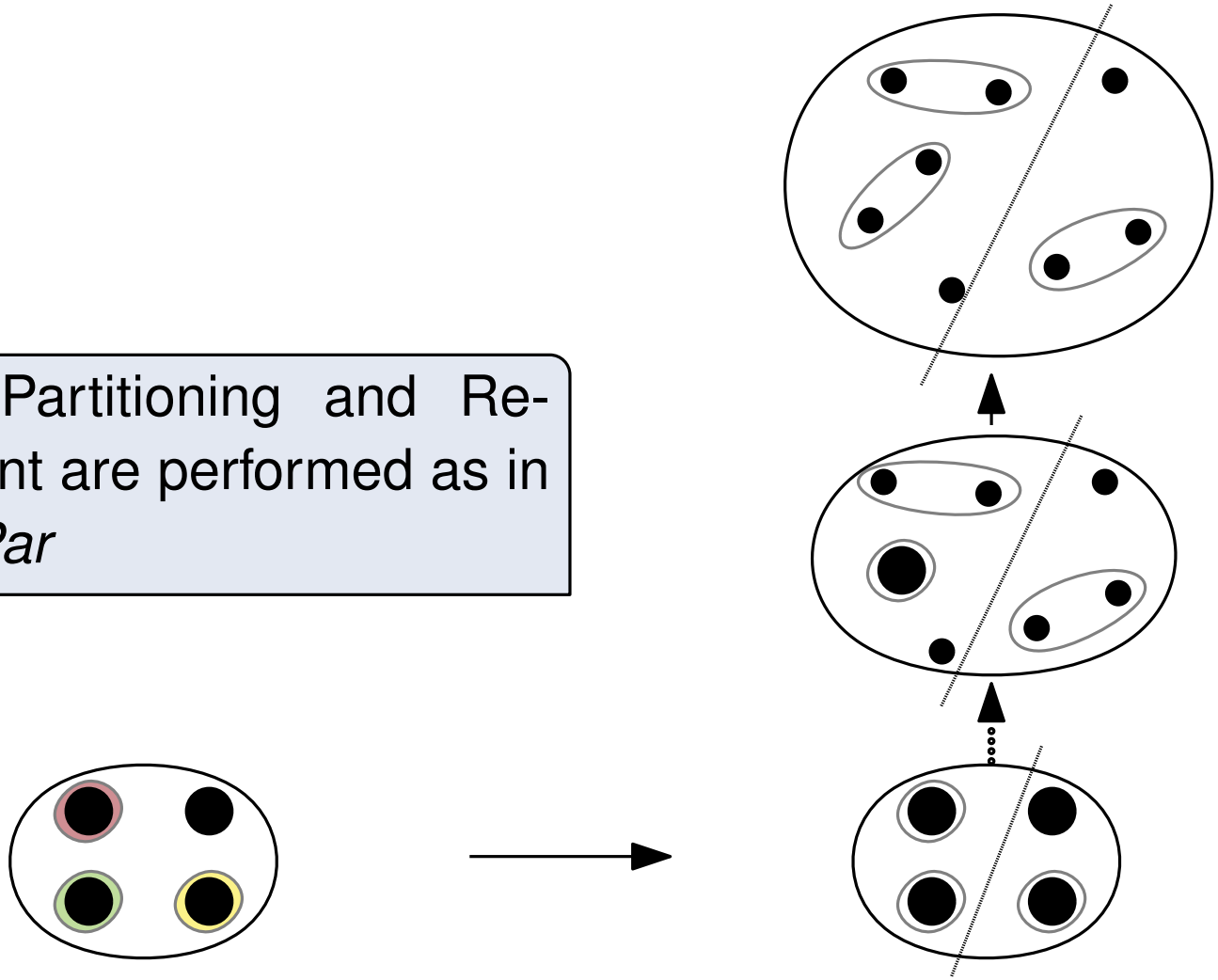
Coarsening:

- the frequency of e discourages contracting e
- low frequency edges are contracted first

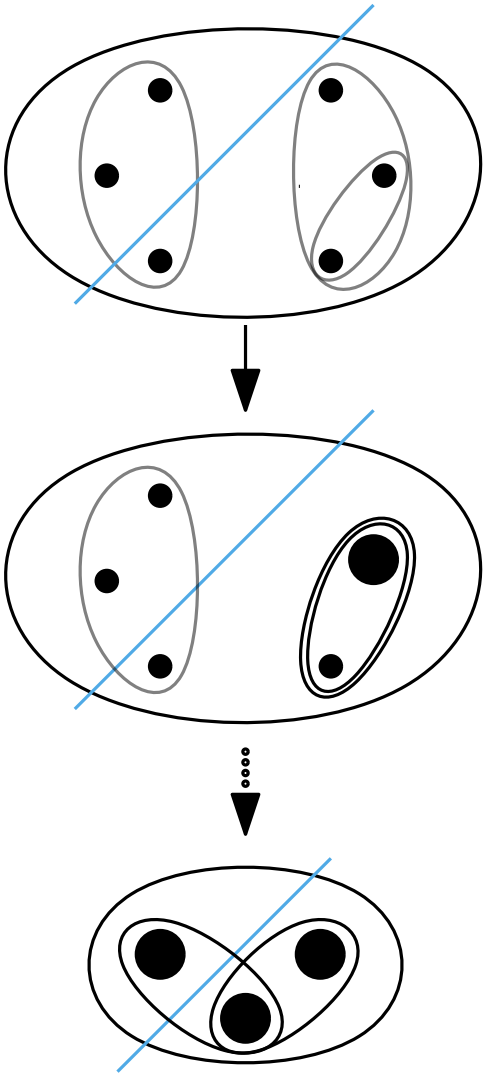
Institute of Theoretical Informatics
Algorithmics Group

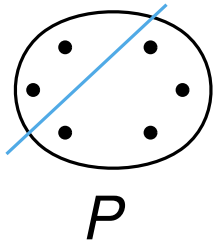
Edge Frequency Multicombine

Initial Partitioning and Refinement are performed as in *KaHyPar*



V-Cycle (+ New Initial Partitioning)

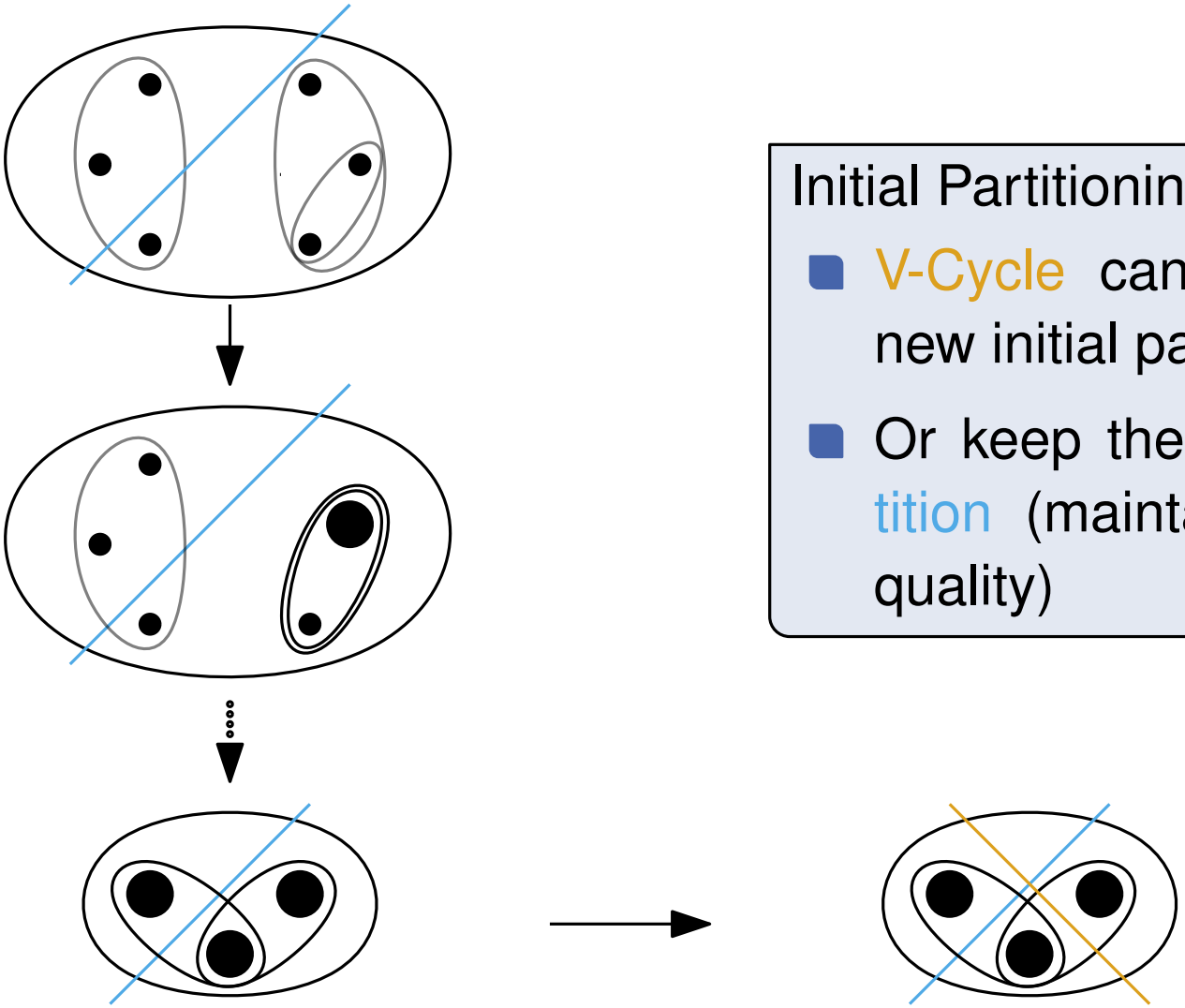




Coarsening:

- contractions must respect P
- does not change solution quality

V-Cycle (+ New Initial Partitioning)



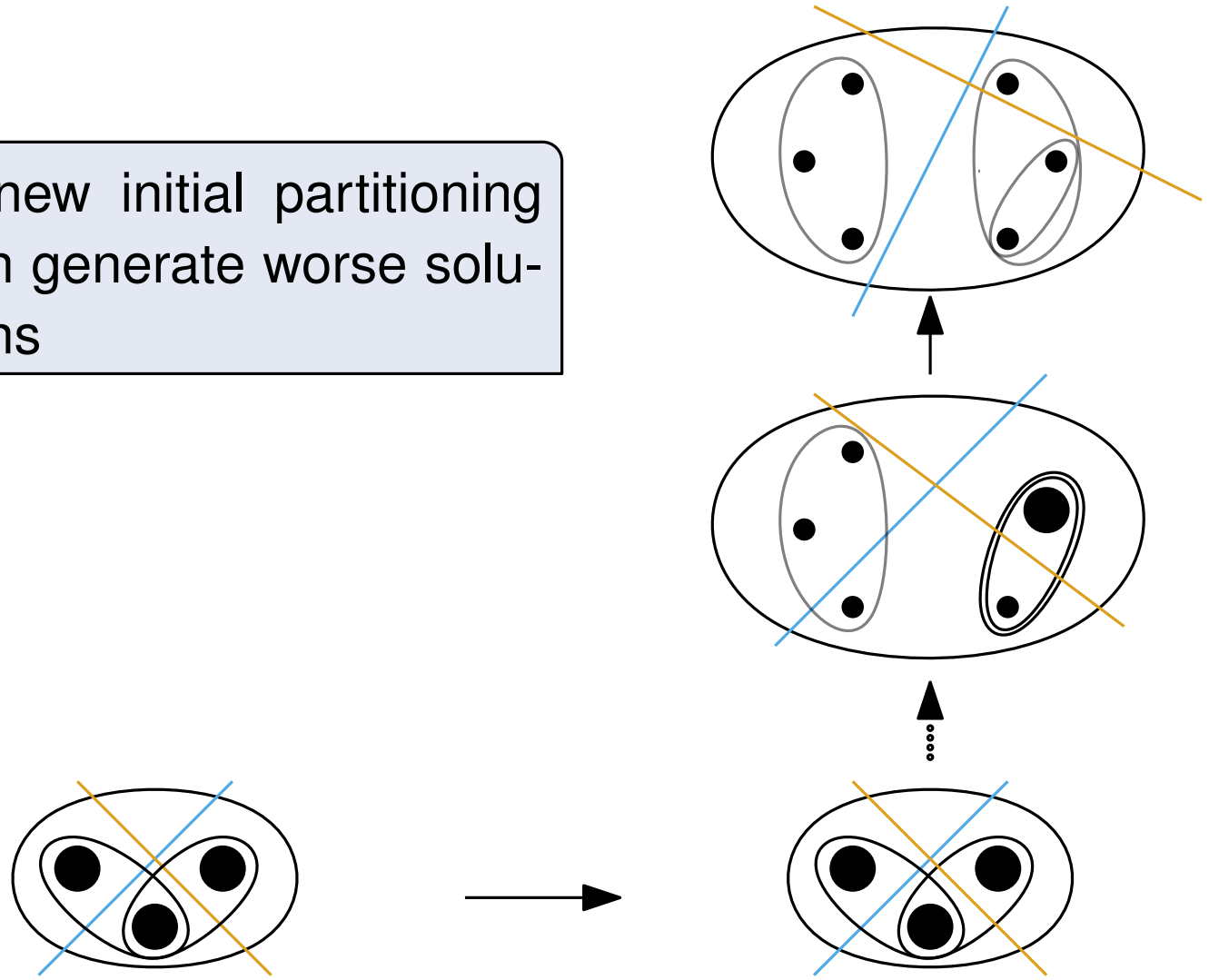
Initial Partitioning:

- V-Cycle can generate a new initial partitioning
- Or keep the current partition (maintains solution quality)

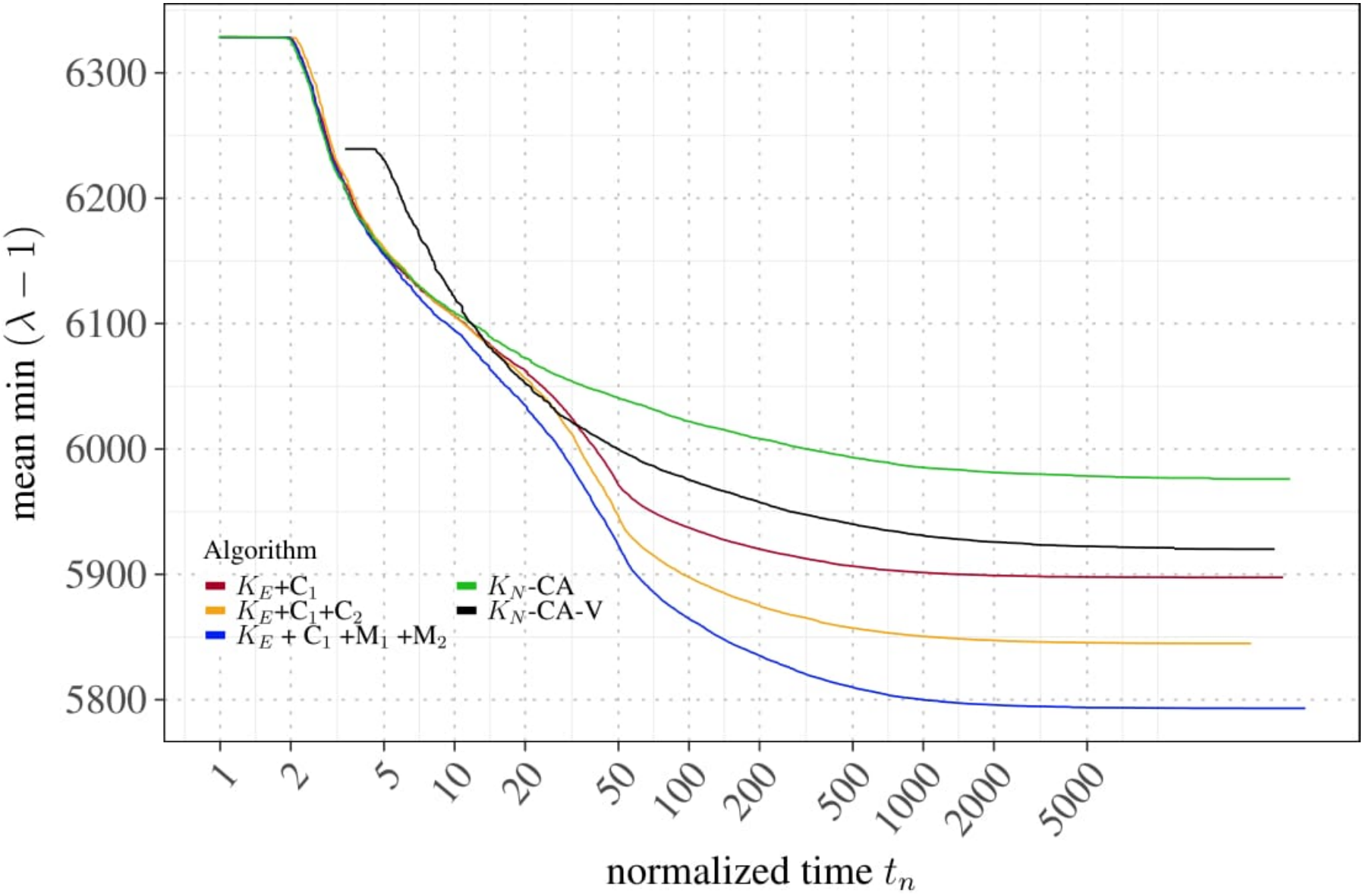
Institute of Theoretical Informatics
Algorithmics Group

V-Cycle (+ New Initial Partitioning)

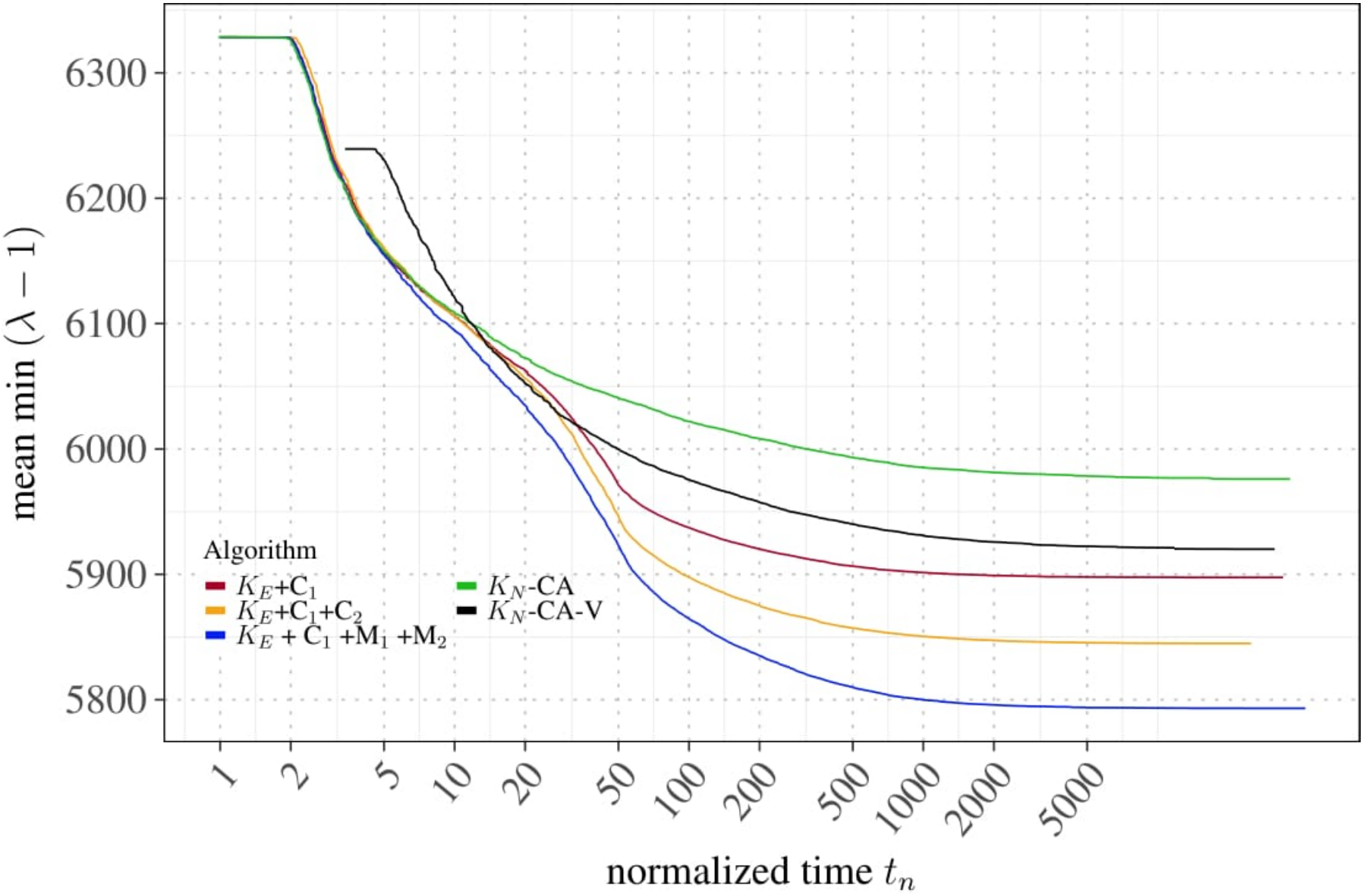
- A new initial partitioning can generate worse solutions



Results


mean min ($\lambda - 1$)
6300
6200
6100
6000
5900
5800
1
2
5
10
20
50
100
200
500
1000
2000
5000
normalized time t_n
Algorithm
K_E+C$_1$
K_E+C$_1$+C$_2$
K_E + C$_1$ +M$_1$ +M$_2$
K_N-CA
K_N-CA-V

Results


mean min ($\lambda - 1$)
6300
6200
6100
6000
5900
5800
1
2
5
10
20
50
100
200
500
1000
2000
5000
normalized time t_n
Algorithm
K_E+C$_1$
K_E+C$_1$+C$_2$
K_E + C$_1$ +M$_1$ +M$_2$
K_N-CA
K_N-CA-V