PPAM 2011

Parallel Version of Image Segmentation Algorithm Using Polygonal Markov Fields

Rafał Kluszczyński, *Piotr Bała*

klusi@icm.edu.pl, bala@icm.edu.pl

13th September 2011
Toruń, Poland

INNOVATIVE ECONOMY
NATIONAL COHESION STRATEGY

EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

Outline

- Introduction
- Polygonal Markov Field
- Segmentation algorithm
- Initial configuration
- Parallel version
- Scenarios and results
- Summary

Introduction

- Segmentation is one of the fundamental processes in image analysis and processes.
- Segmentation is a partition of an image elements (pixels) into homogenous regions such that:
  - every pixel belong only to one region,
  - every region fulfill homogeneity criteria,
  - any sum of 2 different regions does not fulfill homogeneity criteria.

Example of segmentation

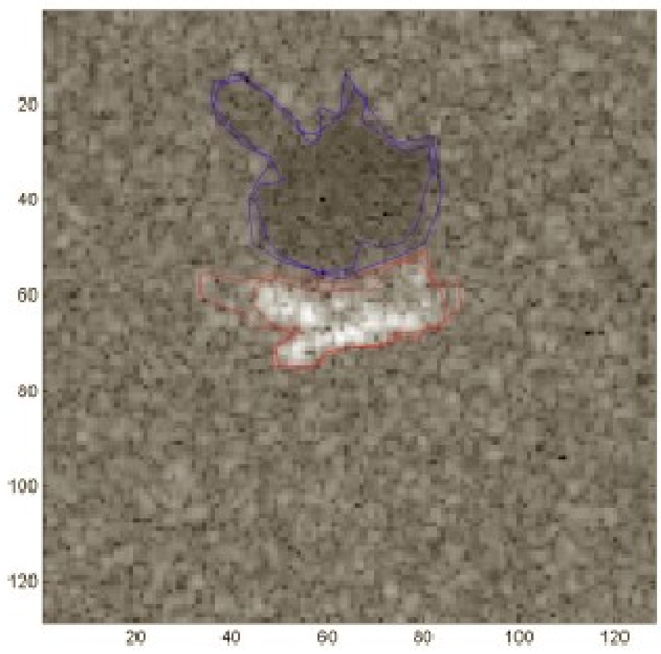
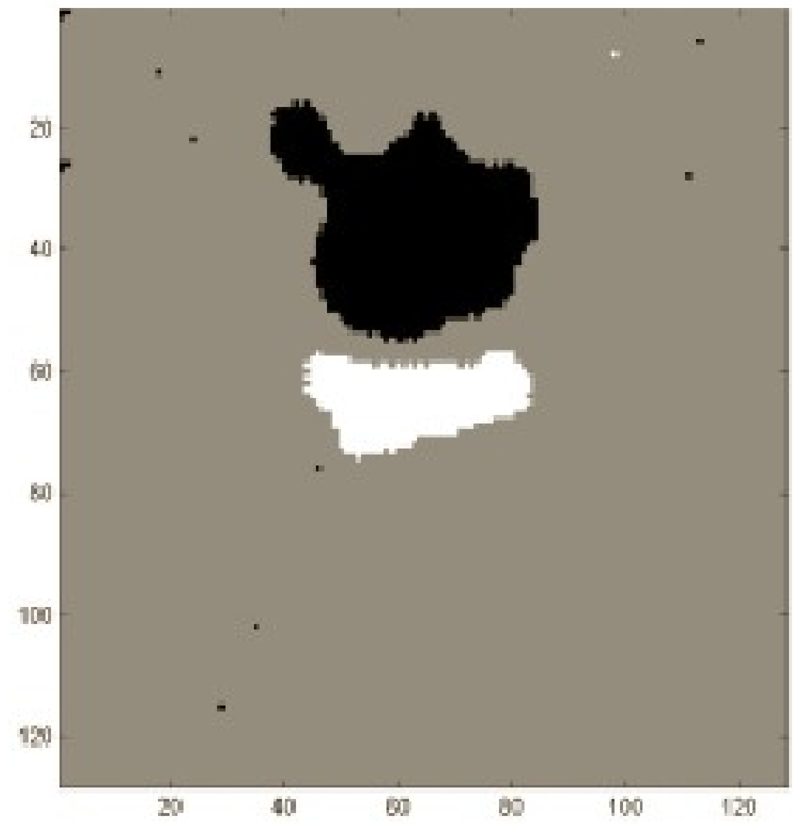


Figure 1—SAR Image of a tank and its shadow and two human segmentations (indicated by thin lines that outline the two objects). The variability of human segmentations is apparent.



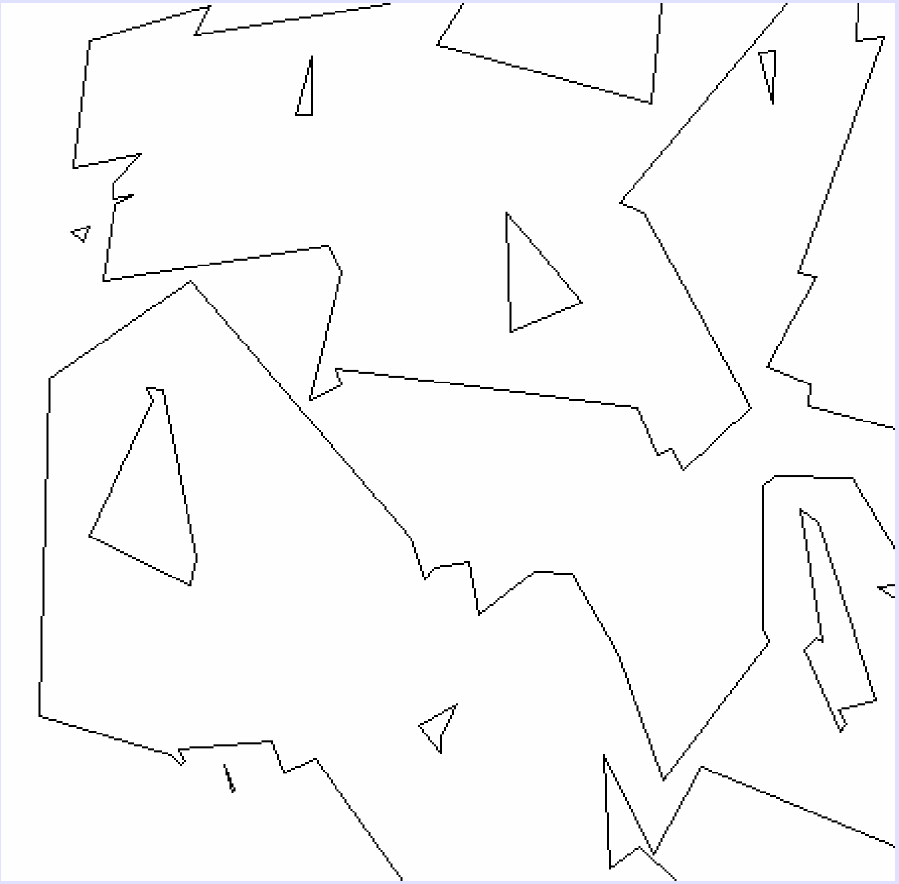
Polygonal Markov Field

- In 1982 T.Arak introduced construction of polygonal Markov field.
- In 1989 T.Arak and D.Surgailis extended this construction.
- Besides formal and probabilistic description, there exists dynamic representation of a construction of PMF.

- T.Arak (1982) *On Markovian random fields with finite number of values*.
- T.Arak, D.Surgailis (1989) *Markov fields with polygonal realisations*.

Polygonal configurations

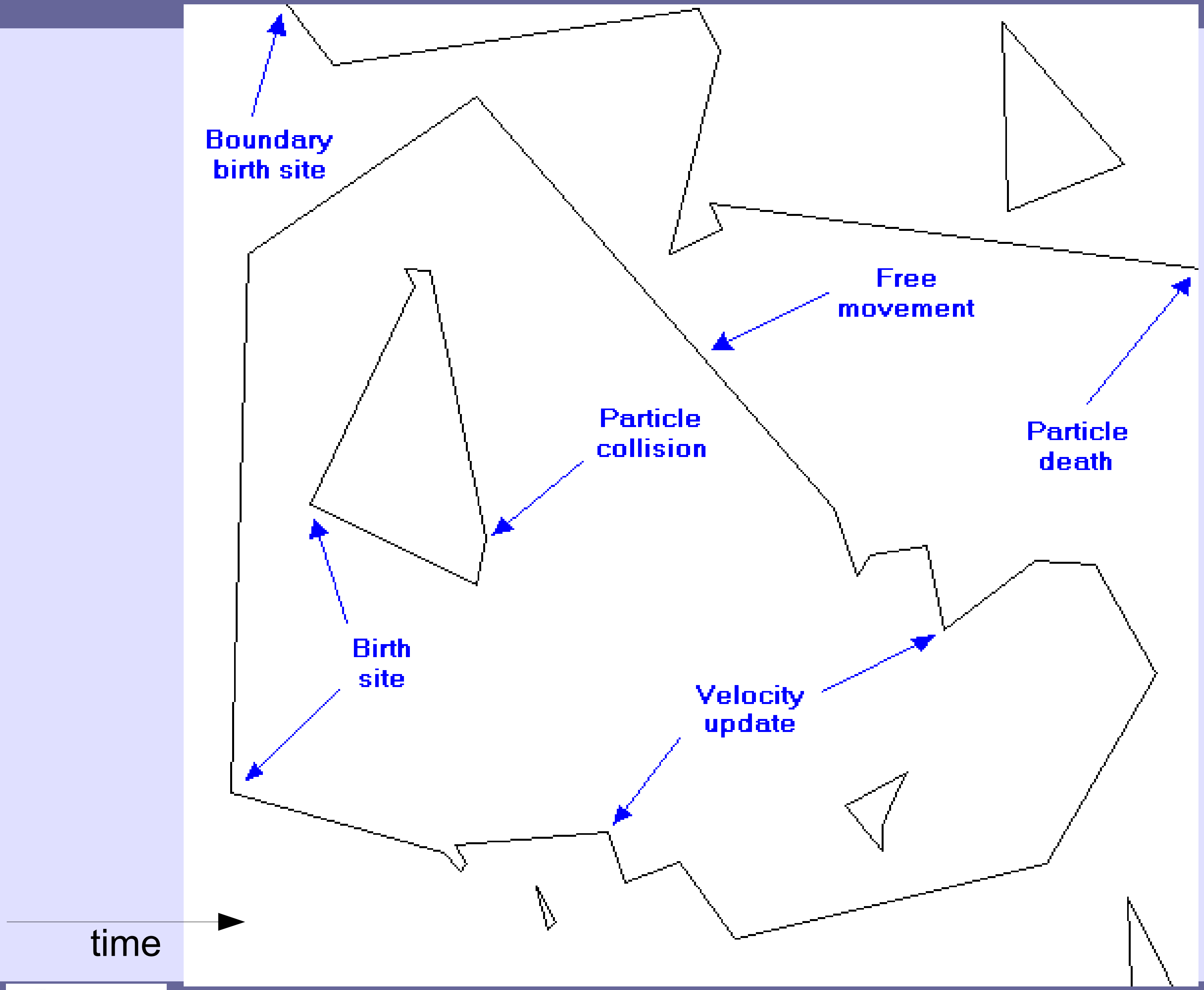
- The polygonal configurations family is a set of all the planar graphs such that:
  - the edges of a graph do not intersect,
  - all the interior vertices are of degree 2,
  - all the boundary vertices are of degree 1,
  - no two edges are collinear.



Dynamic representation

- We interpret polygonal boundaries of the field as the trace left by a particle traveling in two-dimensional time-space.
- Evolution rules:
  - Particles moves freely with constant velocity,
  - When a particle touches the boundary, it dies,
  - In case of a collision of 2 particles, both of them die,
  - The time evolution of the velocity is given by a appropriate pure-jump Markov process.

Evolution events



PMF and image processing

- In 1994 P. Clifford i G. Nicholls presented application of PMF model in image processing.
- In 2004 T. Schreiber described mechanism of configuration modification based on “disagreement loop” concept.
- In 2007 there was described sequential algorithm of image segmentation.

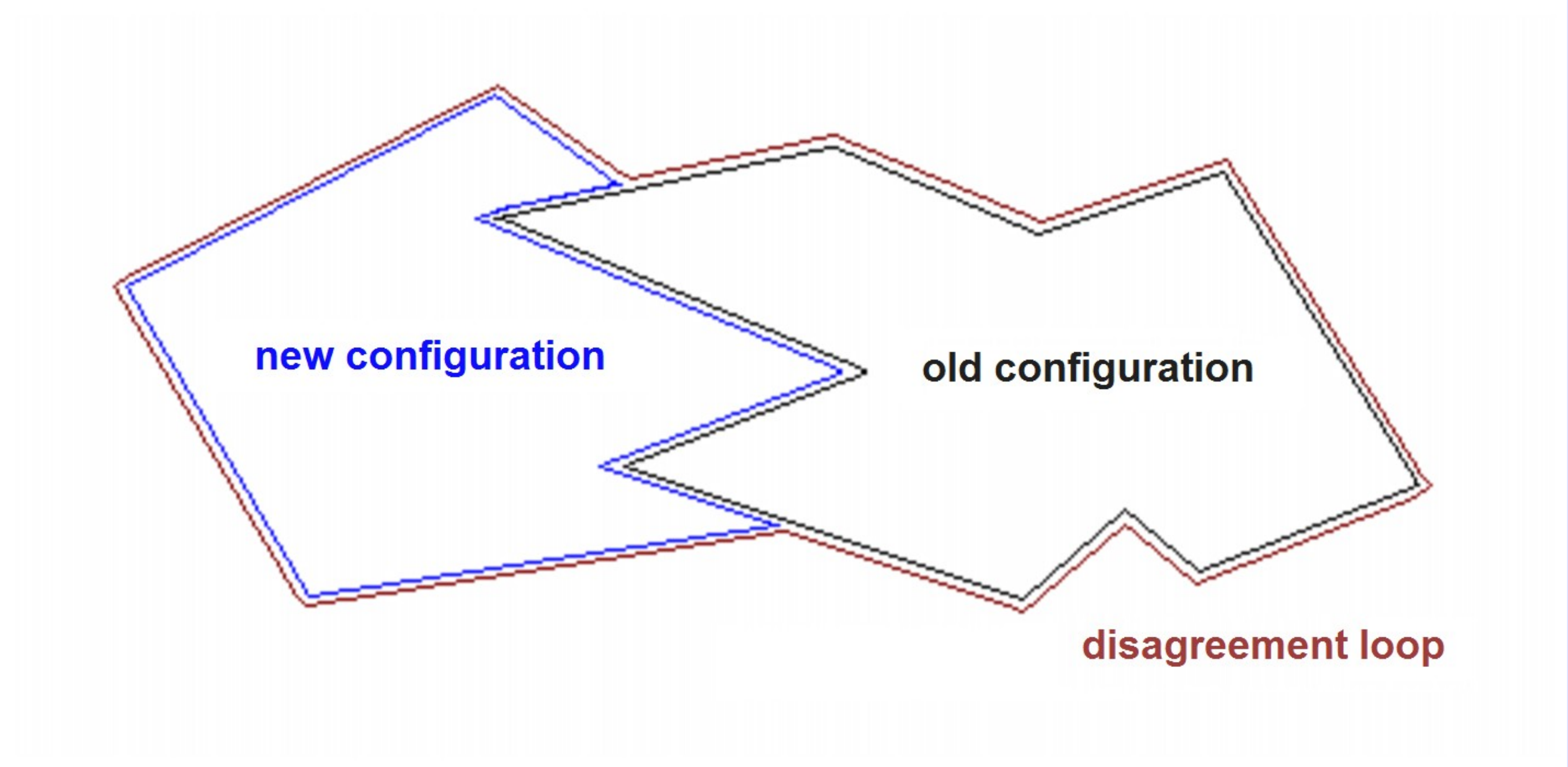
P.Clifford, G.K.Nicholls (1994) A Metropolis sampler for polygonal image reconstruction.
T. Schreiber (2004) *Random dynamics and thermodynamic limits for polygonal Markov fields in the plane*.
R.Kluszczyński, M.N.M.van Lieshout, T.Schreiber (2007) *Image segmentation by polygonal Markov fields*.

Updating mechanism

- We need a mechanism which can modify generated polygonal fields.
- Currently we have 3 main operations updating actual configuration during simulation:
  - adding a new birth site,
  - updating a velocity of a critical moment,
  - removing an existing birth site.
- Updates can be made in any direction.

Disagreement loop

- The concept of disagreement loop make the updates easy to simulate.

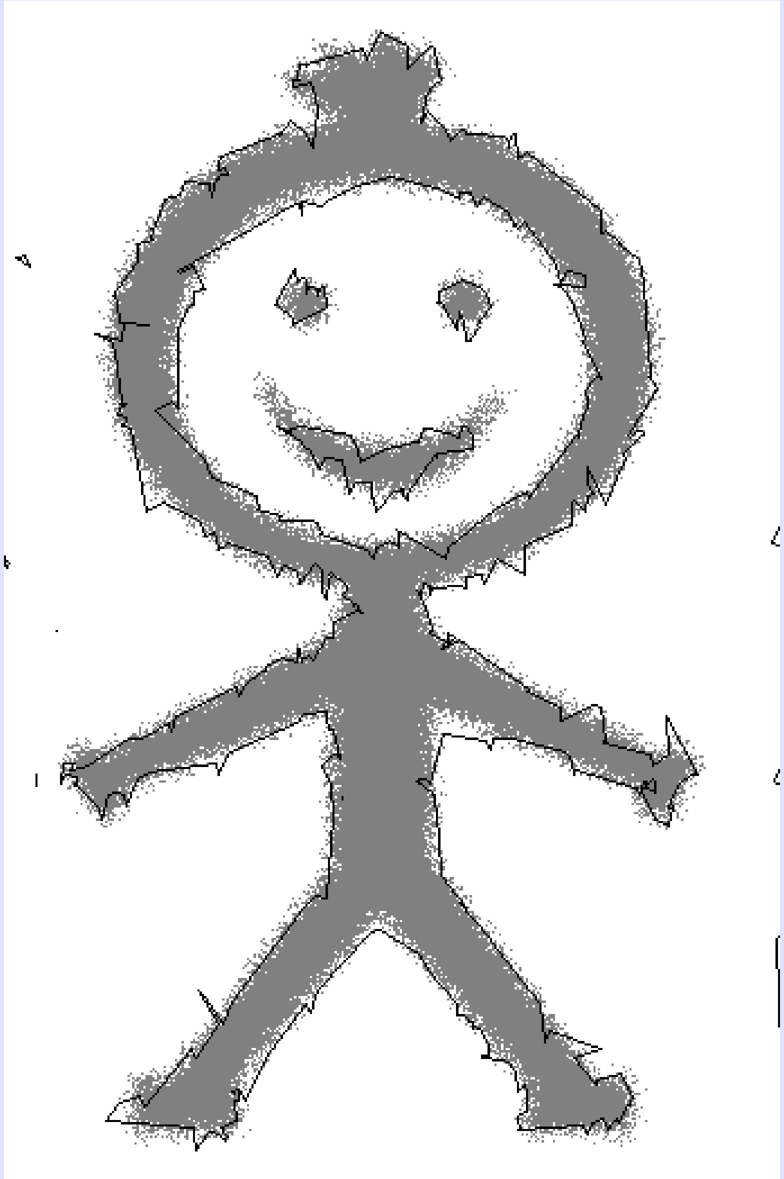


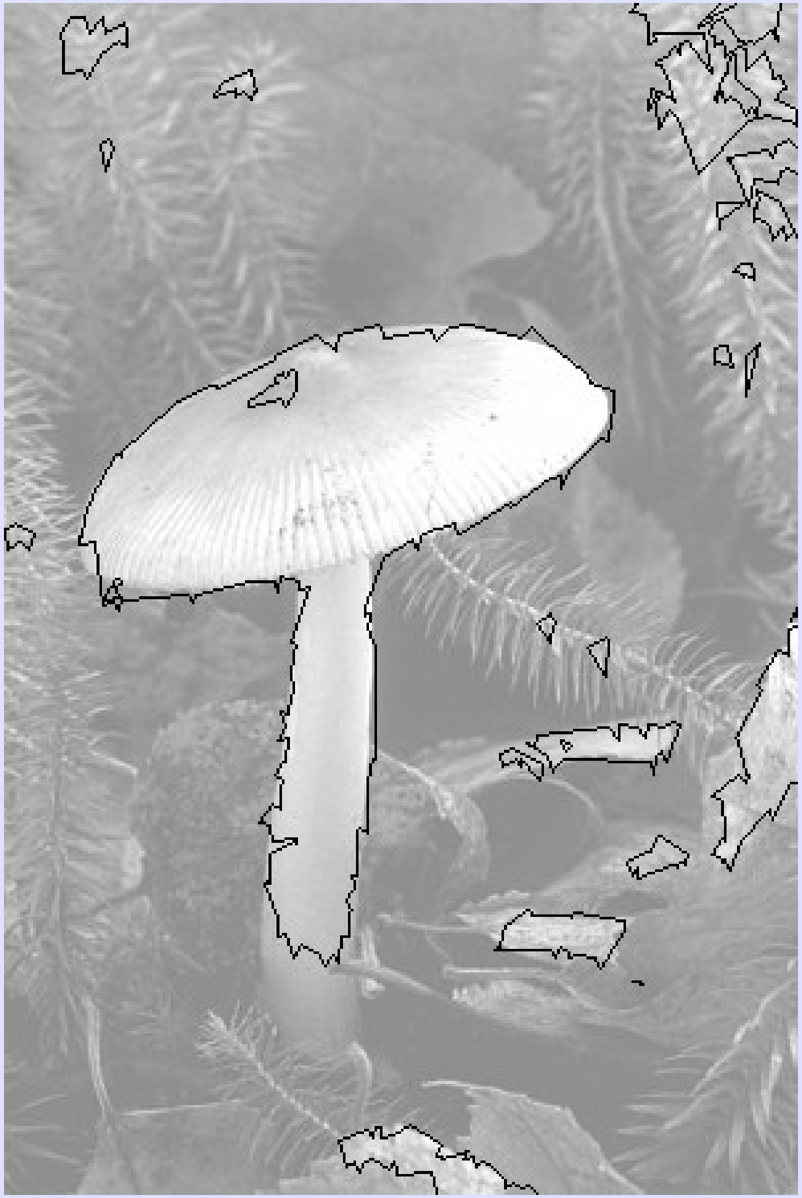
T. Schreiber (2004) *Random dynamics and thermodynamic limits for polygonal Markov fields in the plane*.

Simulated annealing

- Update operations have been combined with simulated annealing techniques to find optimal segmentation.
- After each update we determine if the new configuration describes better our image by counting misclassified pixels.
- Presented configuration can fully describe only binary images, but it works also in case of greylevel pictures.

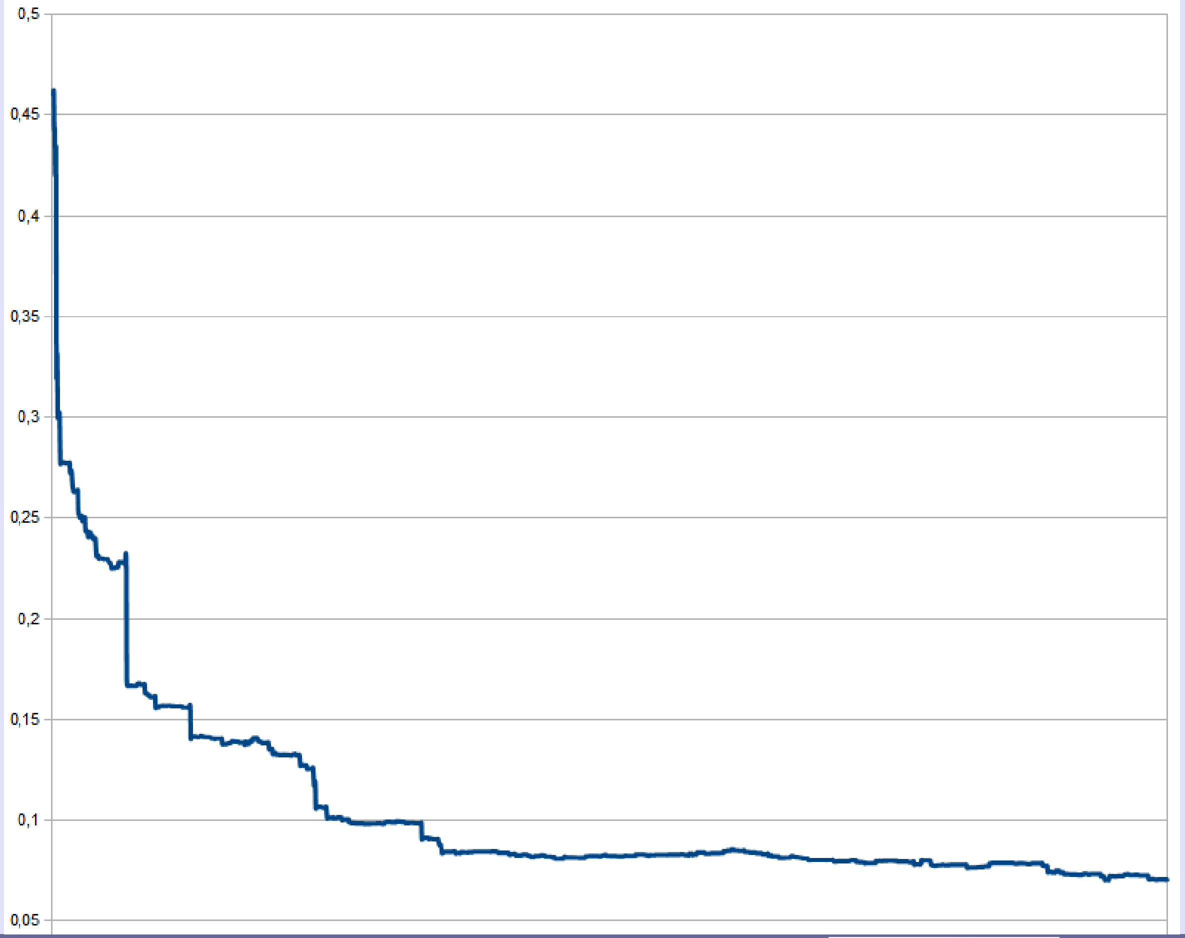
Segmentation examples





Convergence

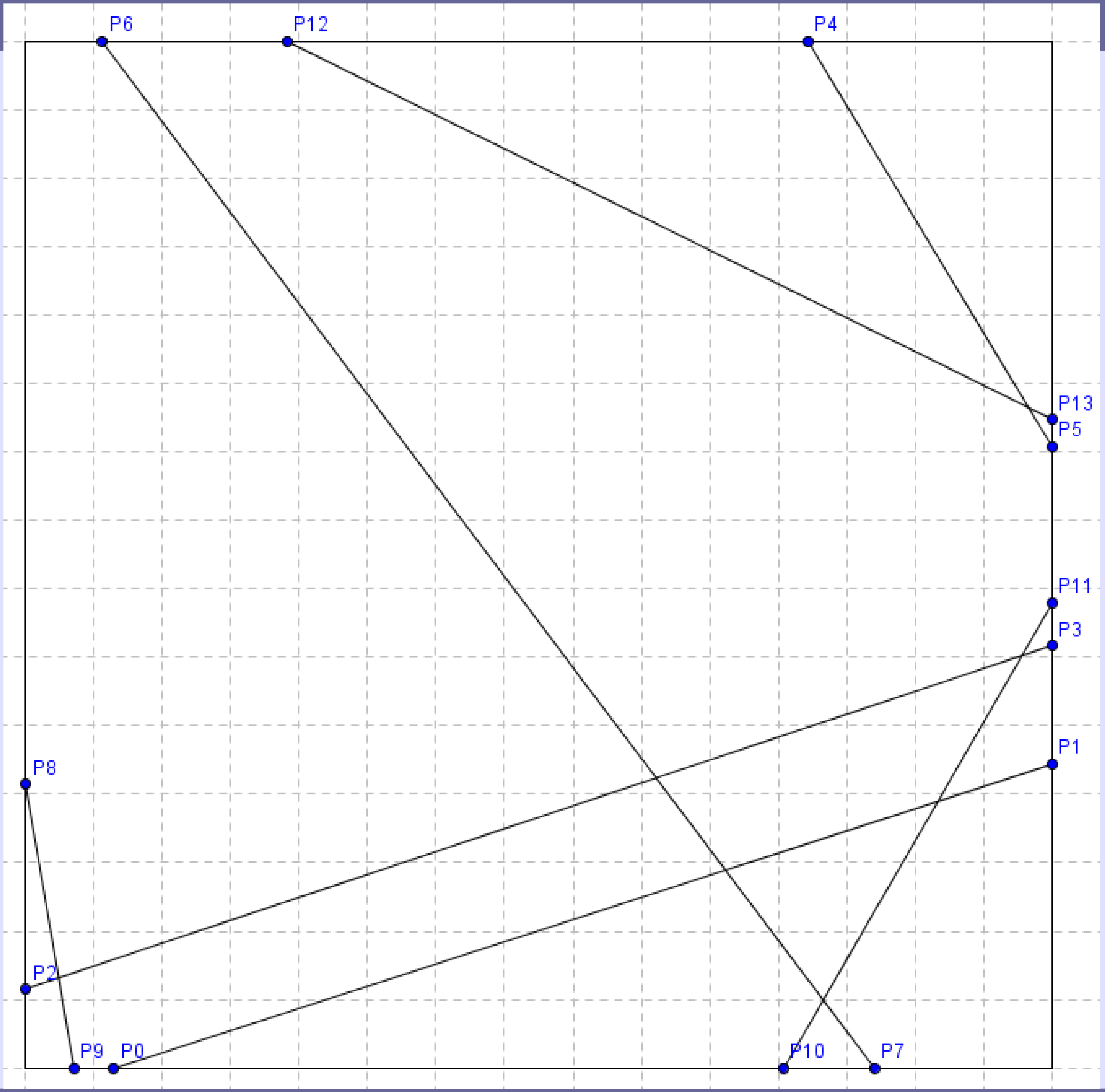
- First version of an algorithm started with a randomly generated configuration.
- There was no use of any image data.



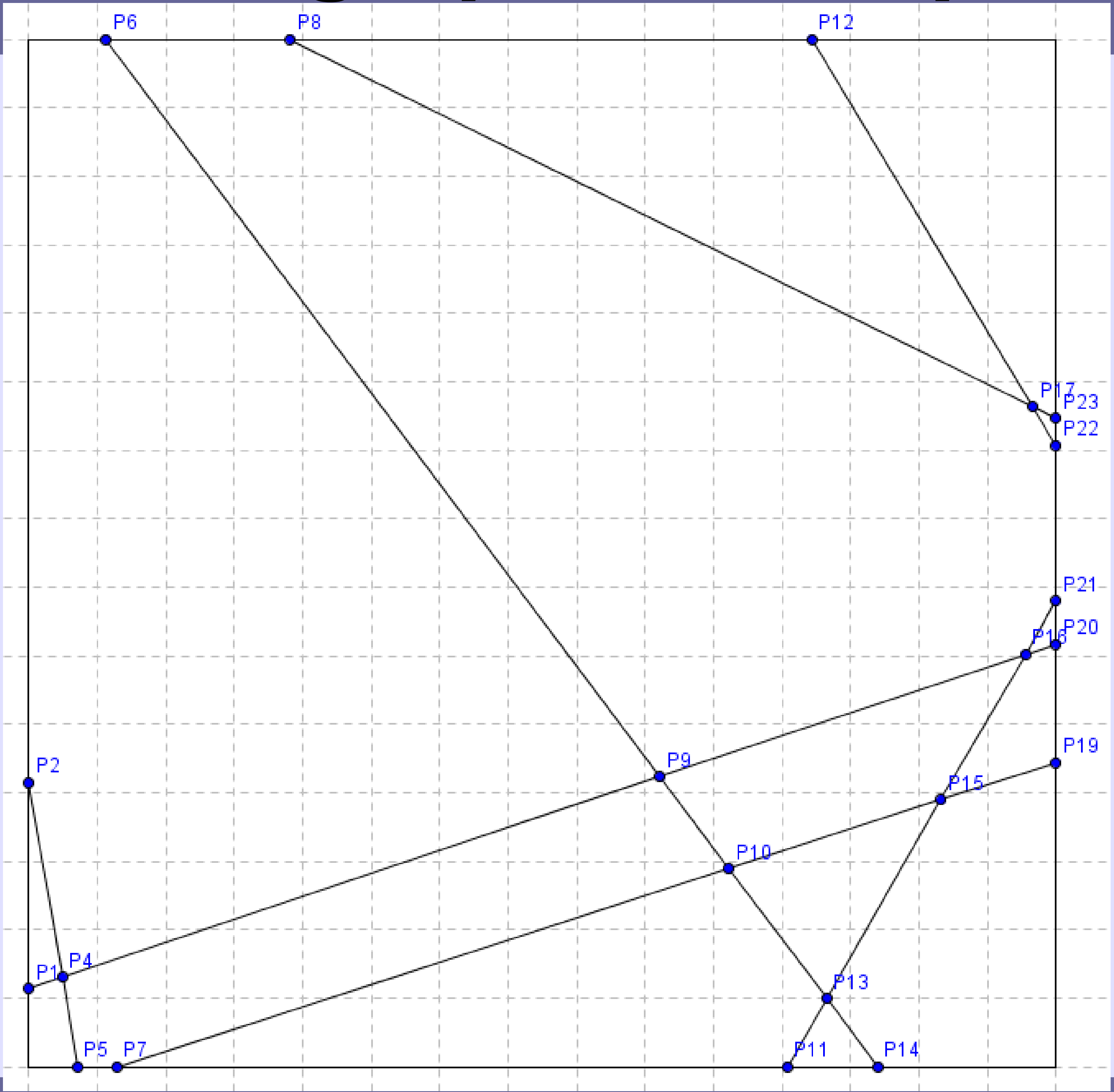
Initial PMF algorithm

- In order to speedup an initial phase we implement an algorithm of generating initial configuration based on image data:
  - Generate random lines,
  - Translate lines into planar graph,
  - Generate dual graph,
  - Determine colors of dual graph vertices,
  - Merge regions of two adjacent dual graph vertices with the same color,
  - Adapt resulting graph to polygonal Markov field structure.

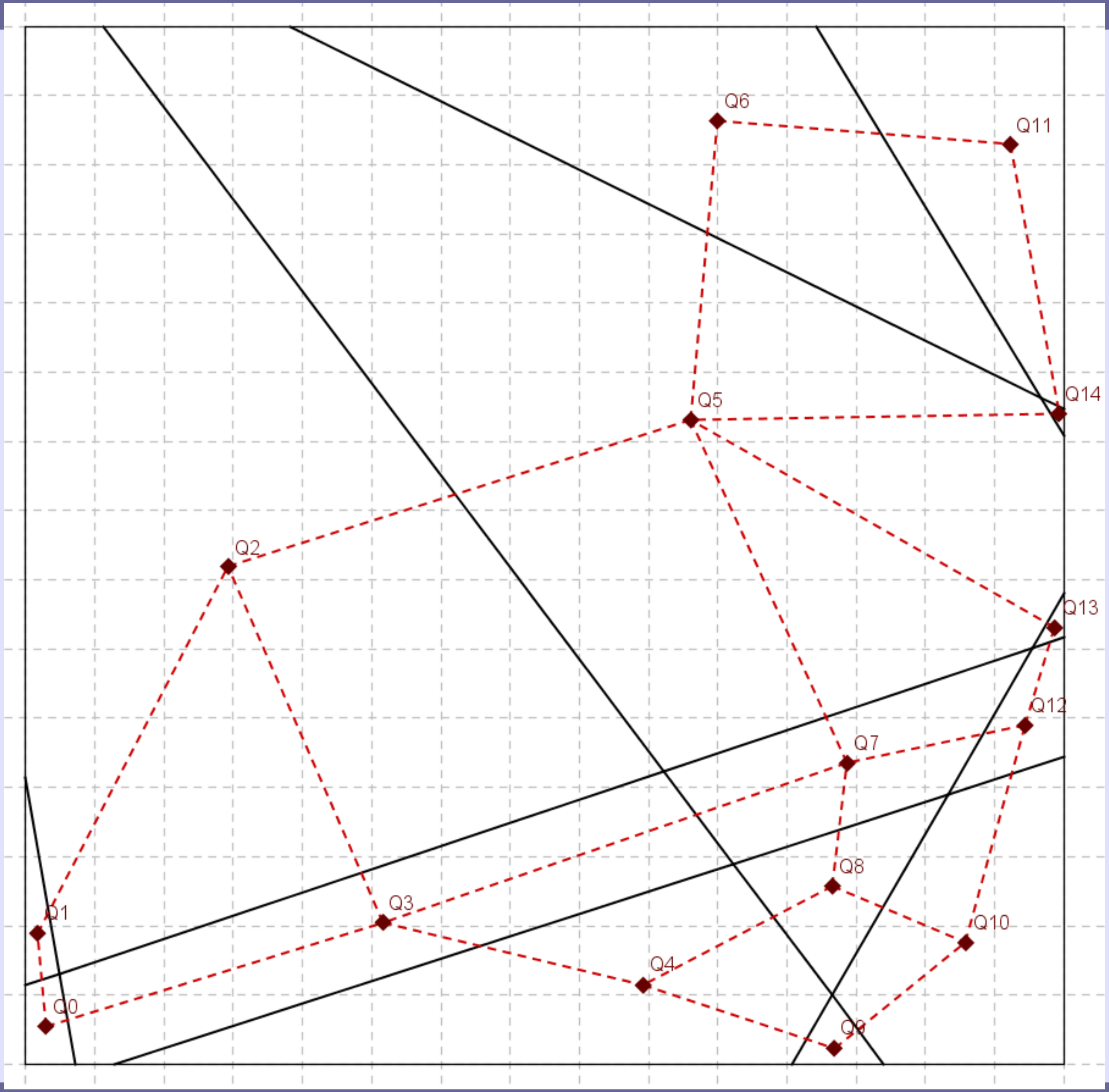
Randomly chosen lines



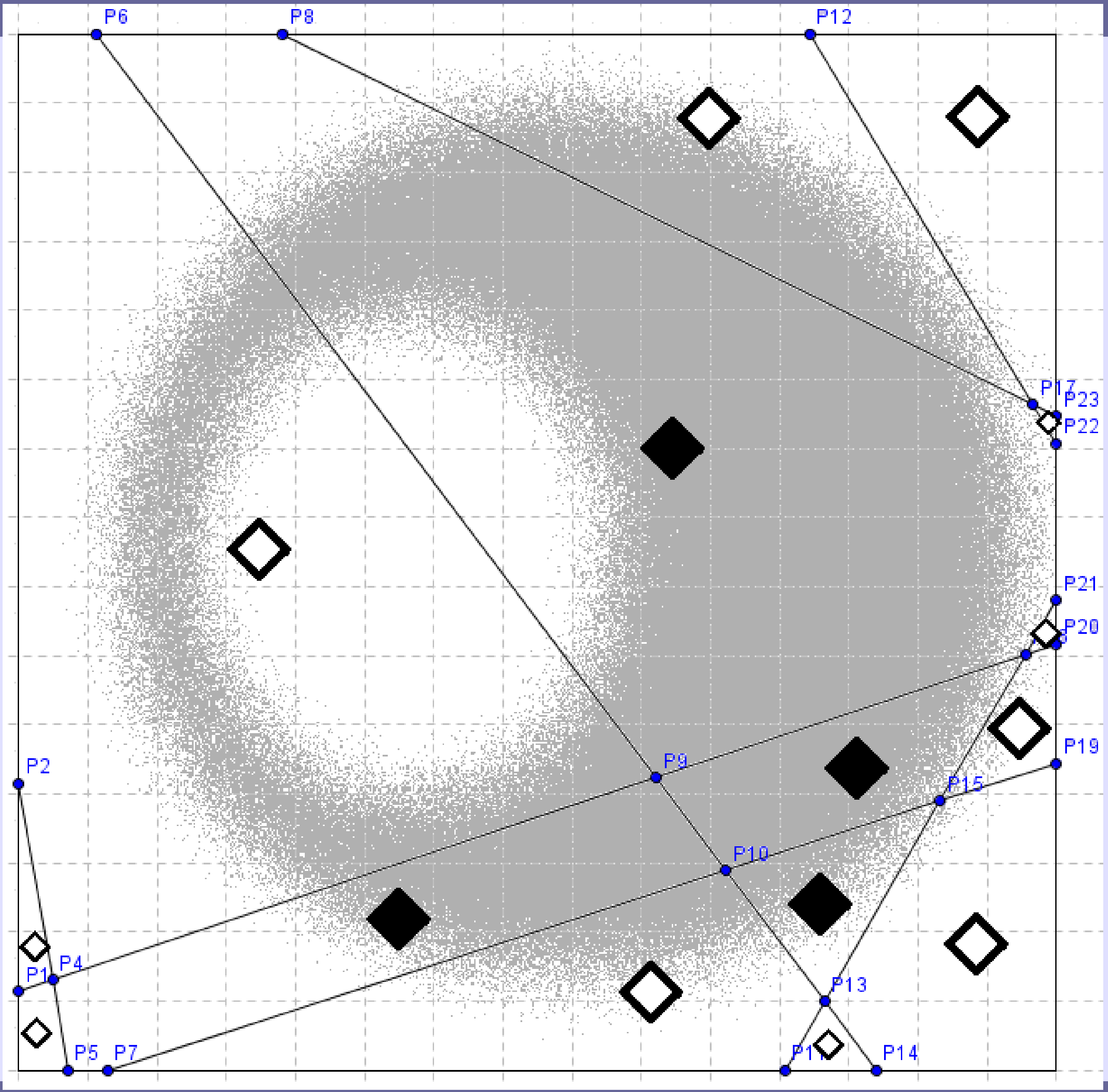
Planar graph descritpion



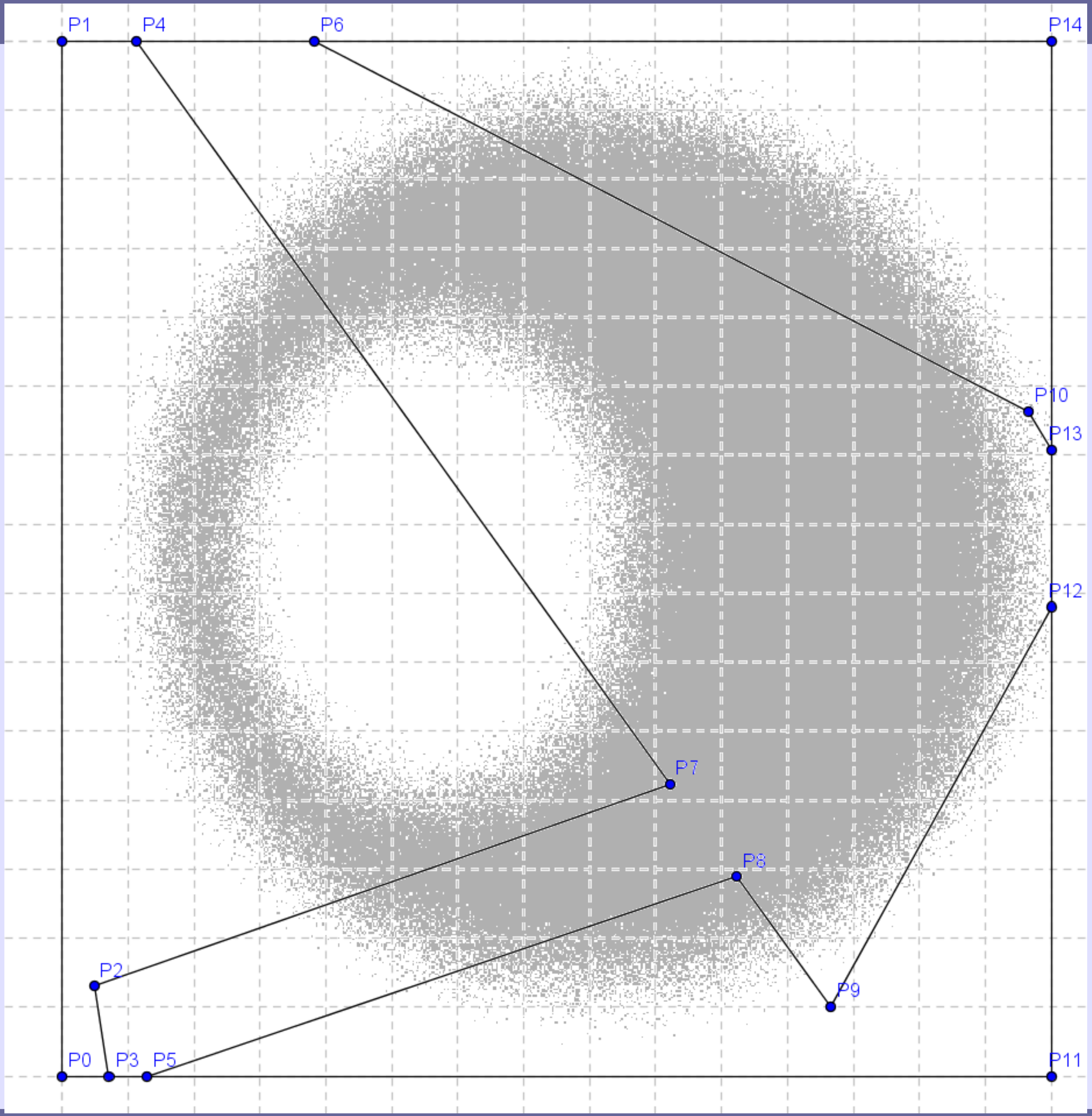
Dual graph



Coloring of dual graph



Merging phase

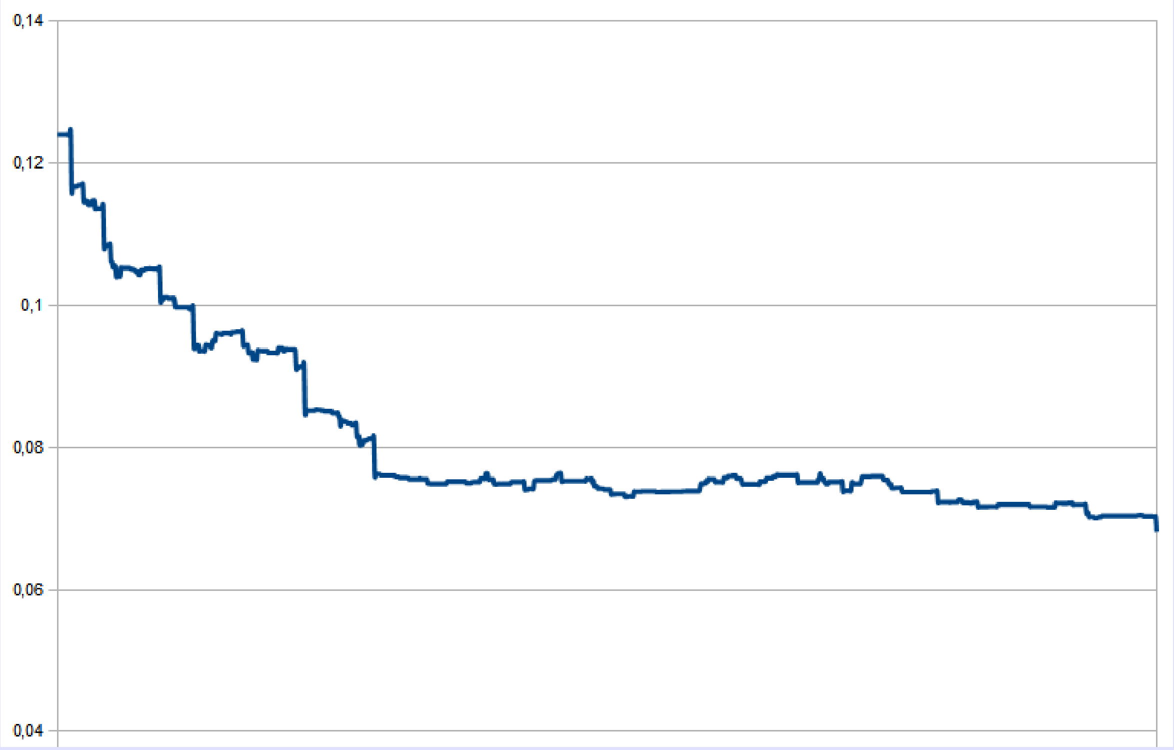


Complexity

- Complexity of initial PMF algorithm is:

$$O(\ (n+k)\ log(n+k)\ \ +\ \ m\ ).$$

- An algorithm shortened initial phase of segmentation.



Multicore processors

- Simulated annealing is a powerful global optimization technique, which can be computationally expensive.
- Nowadays, multicore processors are available at homes.

http://www.intel.com

Parallel SA

- Algorithms based on SA method can be speed up in parallel environment.
- There are 2 different approaches:
  - single-trial parallelism,
  - multiple-trial parallelism.
- In case of single-trial parallelism implementation and speedup is highly depended on a problem.

Multiple-trial parallelism

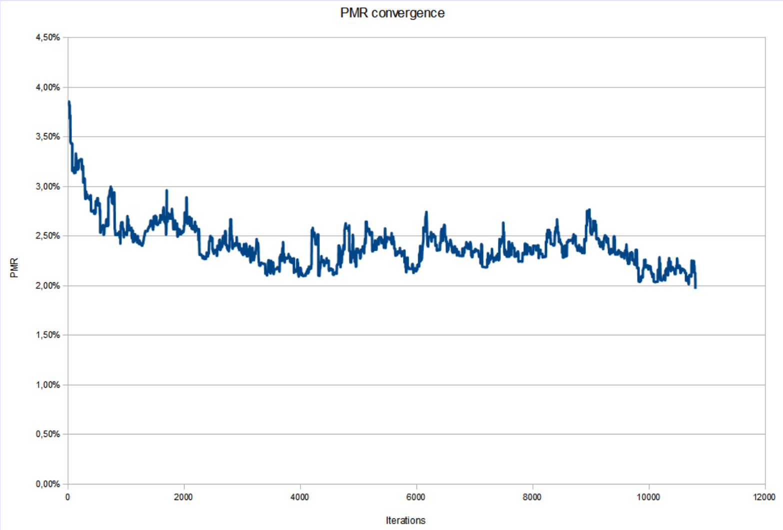
- In case of multiple-trial parallelism we used “parallel tempering” method.
- Method used in many applications, like molecular dynamics.
- The key concept is to simulate simultaneously multiple instances of solutions.
- There exists additional movement which swaps two different solutions (states) between threads (chains).

Scenarios of parallel SA

- In case of multiple-trial parallelism we tested following scenarios of exchanging replicas:
  - Exchange with a probability,
  - Weighted replication with fixed step,
  - Weighted replication with increasing step,
  - Replication of the best partial solution with increasing step.

Example used in tests

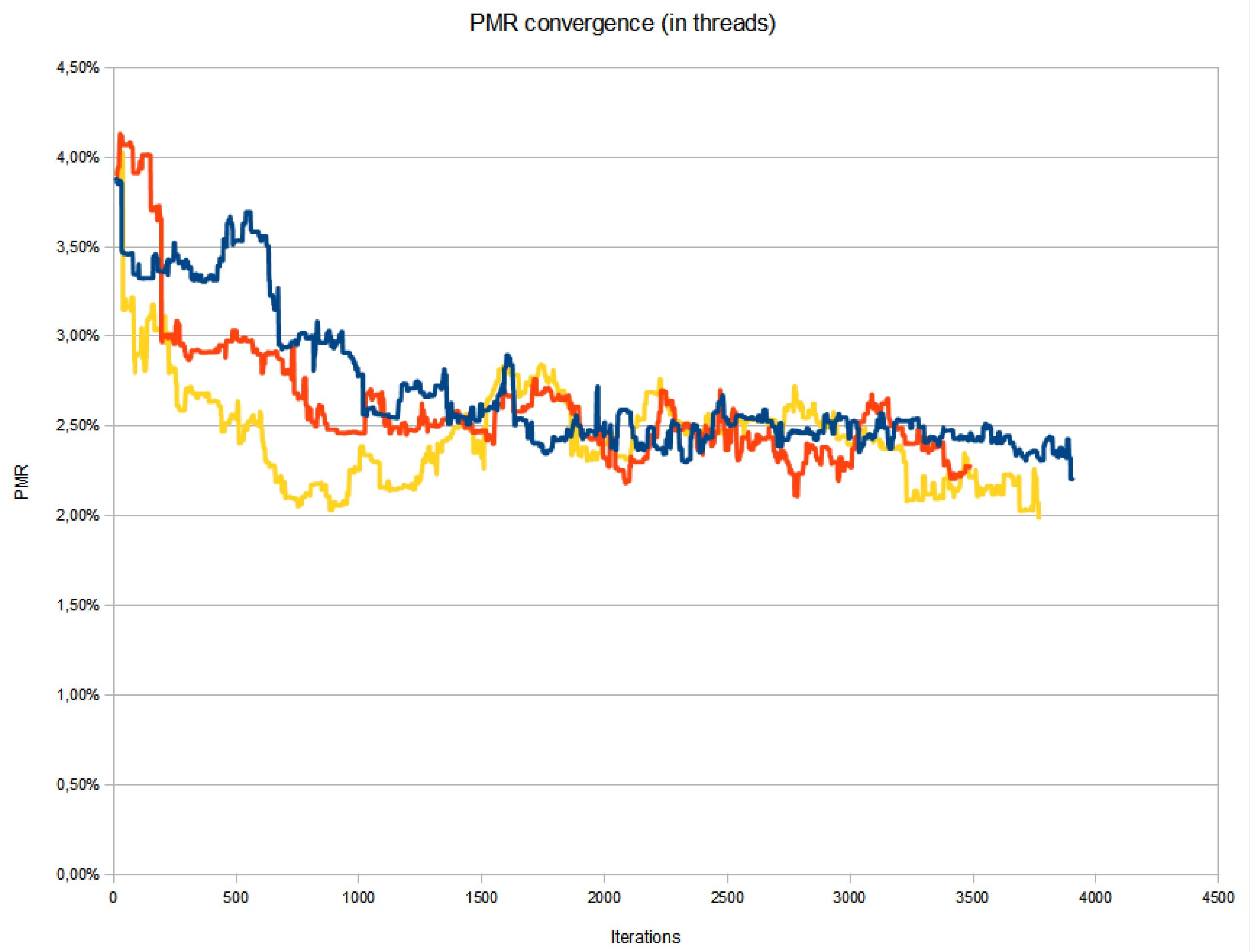
- To compare different scenarios of exchange we used a cherry image.



Exchange with a probability

- The movement of exchange between two different was set by a probability:

p = 0.001



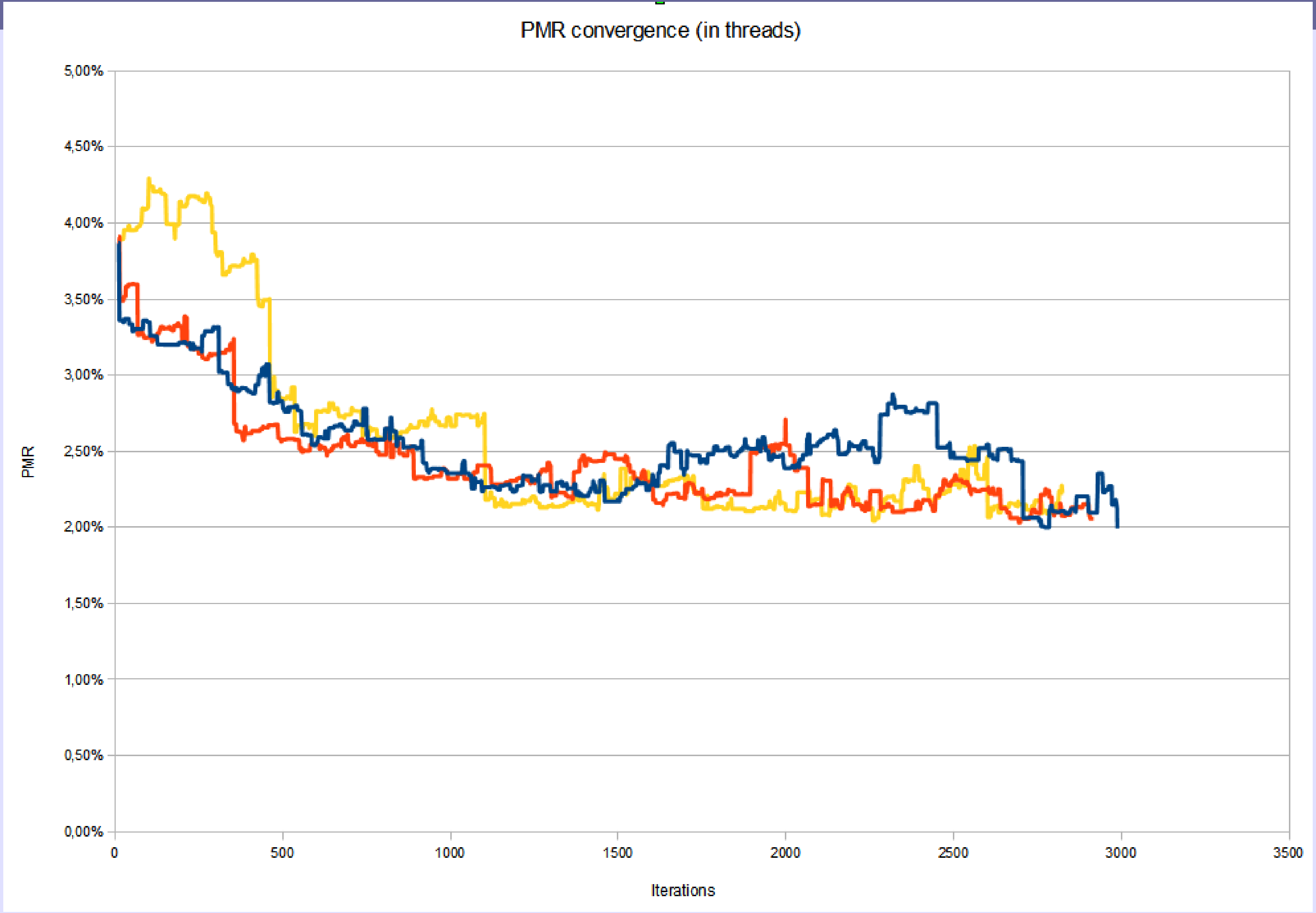
Weighted replication

- Every *N* iterations we synchronize threads and replace the solutions with one randomly chosen.
- We choose replica *i* with a probability:

$$p_i = \frac{\exp(E_i - E_m)}{\sum_k \exp(E_k - E_m)}$$

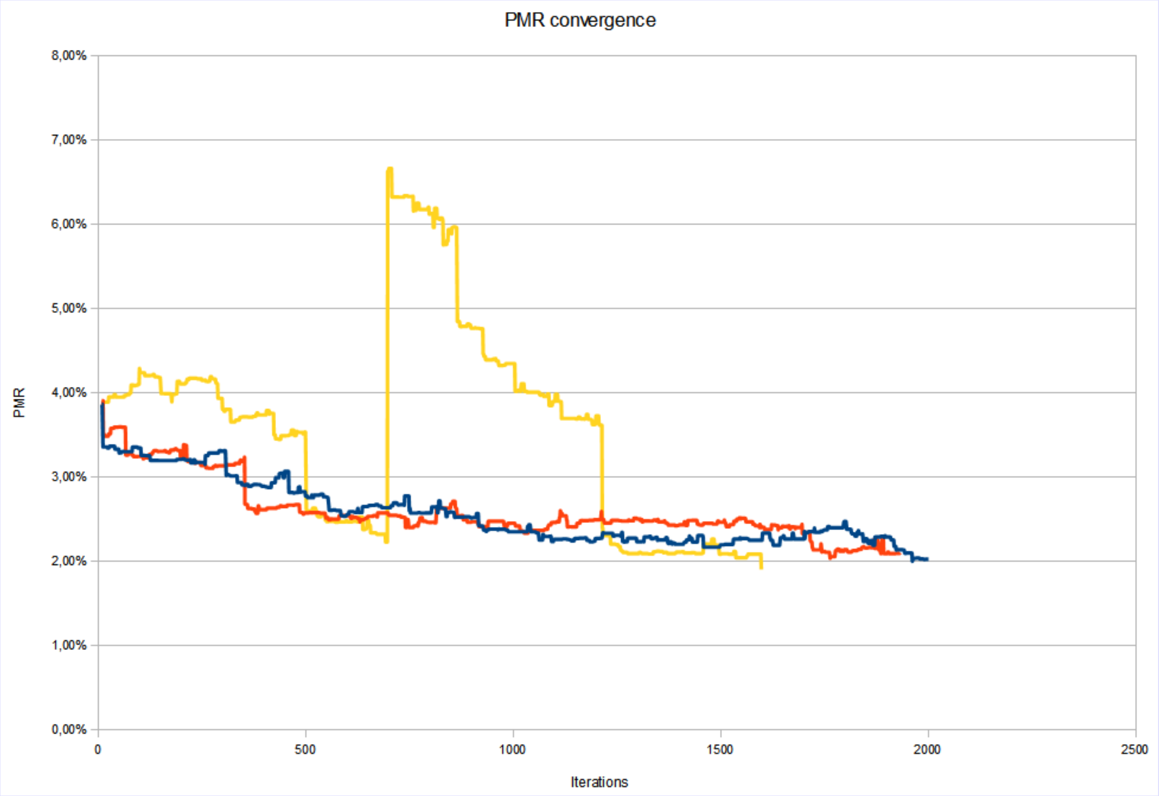
where $E_m = min_k(E_i)$

Weighted replication – fixed step



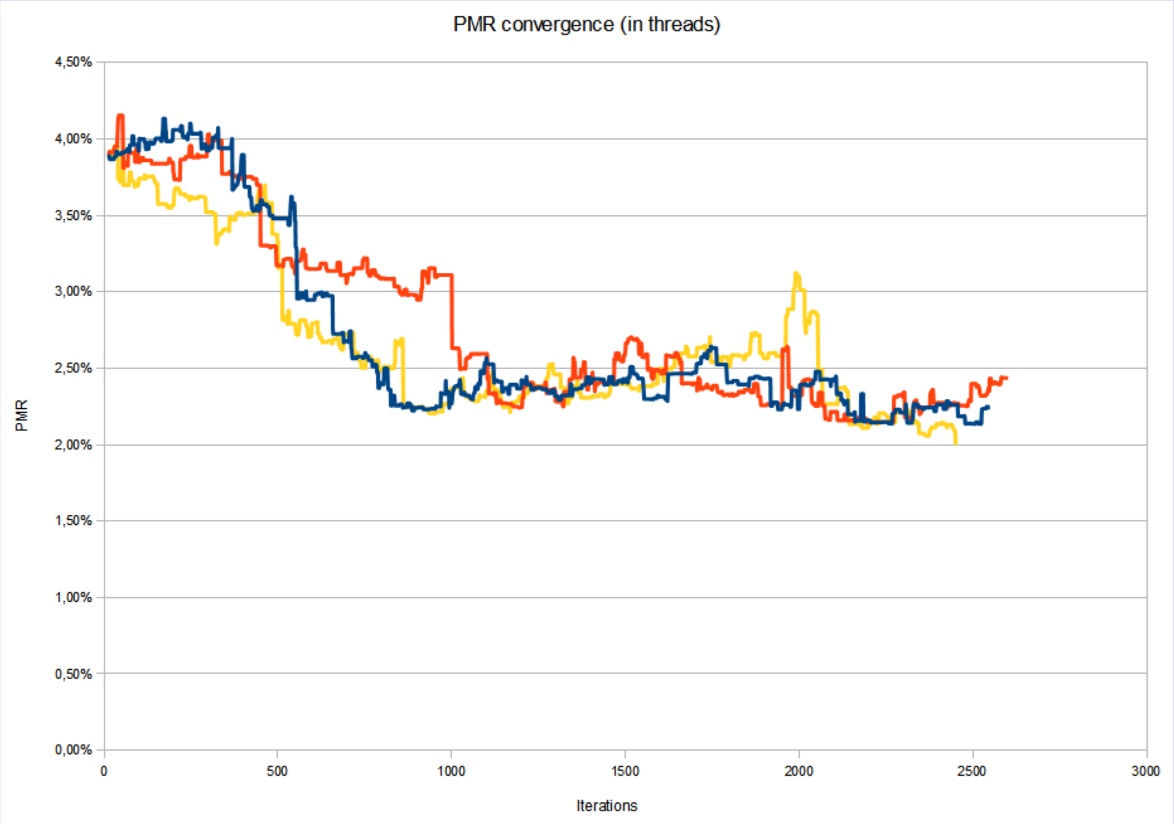
Replication with increasing step

- In case of SA simulations, increasing period of synchronization is more suitable.
- We used linear increase: $750 + 1000 * i$ ($i = 0, 1, ...$).



Minimum propagation

- We choose the best partial solution along the replicas and replace the others with it.
- Synchronization with linear increasing step.



Comparison of scenarios

Scenario	1 thread	2 threads	3 threads	4 threads	Work
Single thread run	14776	-	-	-	14776
Exchange with a probability	3236	3159	3190	3220	12805
Weighted replication – fixed step	2618	2652	2611	2609	10490
Weighted replication – increasing step	2495	2521	2517	2533	10066
Best partial solution replication	2214	2235	2267	2235	8950

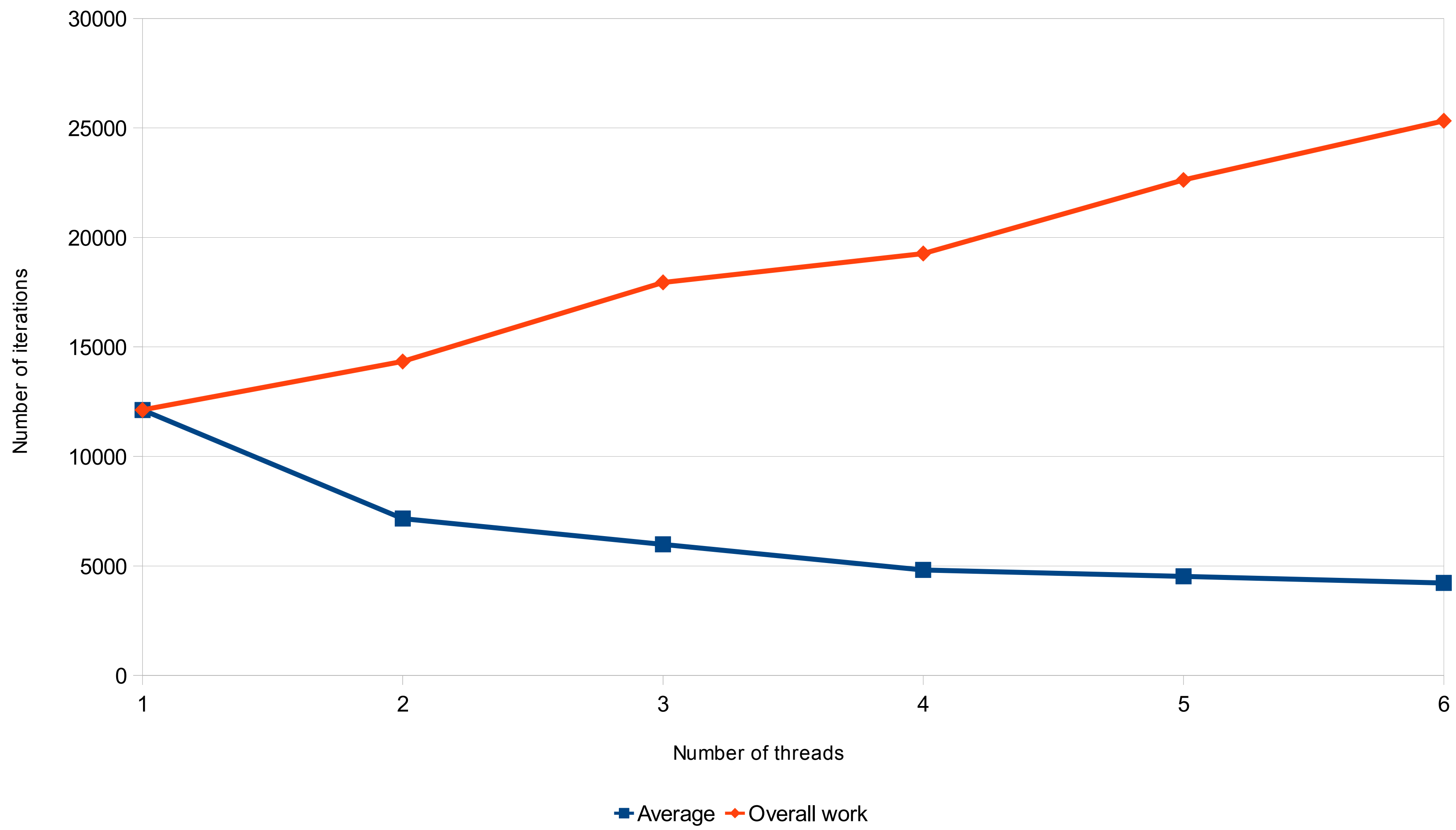
– Average values of 30 segmentations (*PMR* = 2%)

Comparison for #threads

#	Th1	Th2	Th3	Th4	Th5	Th6	Work
1	12119	-	-	-	-	-	12119
2	7159	7173	-	-	-	-	14332
3	5977	5992	5977	-	-	-	17946
4	4814	4820	4824	4808	-	-	19266
5	4530	4527	4523	4522	4523	-	22625
6	4225	4224	4209	4233	4221	4216	25328

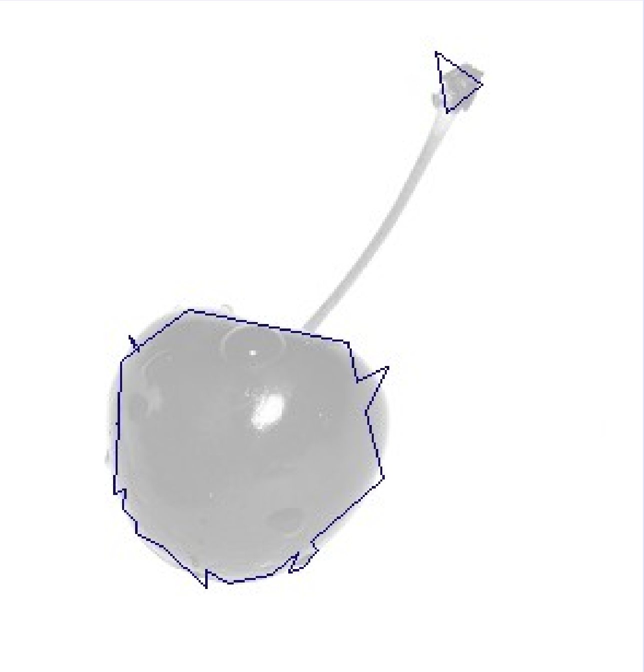
– Average values of 100 segmentations (*PMR* = 1.8%)

Average iterations number

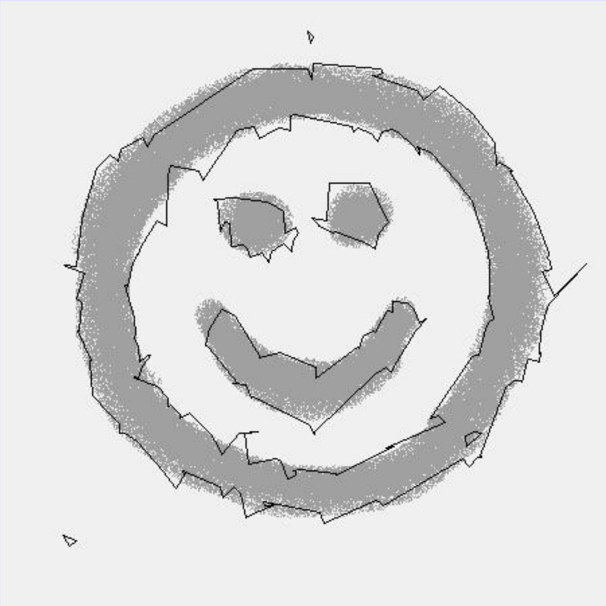


Summary

- General multiple-trial parallelism of SA can speedup calculations.
- Scenarios of synchronization do matter and affect execution time.
- Increasing number of threads:
  - decreases execution time,
  - increases asymptotically overall work.



Thank you for your attention!



This research was supported in part by PL-Grid Infrastructure.
http://www.plgrid.pl/