EuroHPC PL

EuroHPC PL

Narodowa Infrastruktura Superkomputerowa dla **EuroHPC**

Simulating and visualizing quantum architectures

Łukasz Pawela, IITiS PAN

Fundusze Europejskie
Inteligentny Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

Motivation

- Dynamically developing
- Plethora of potential applications
- Strong interest from bussiness partners

Benefit

Easier access for business and research customers to quantum architectures

Simulators: reducing the cost of solution preparation and testing

Future-proofing: the ability to prepare solutions that work on upcoming quantum architectures

Visualizations: facilitating the design of solutions, preparation and analysis of programs

Quantum annealing

Ising model

Given a function

$$H(s) = -\sum_{i\in\mathcal{I}} h_i s_i - \sum_{(i,j)\in\mathcal{I}\times\mathcal{I}} J_{ij} s_i s_j,$$

where

$$s = [s_i]_{i\in\mathcal{I}} \in \{-1, 1\}^{\mathcal{I}}, h_i \in \mathbb{R}, J_{ij} \in \mathbb{R}.$$

The goal is to find

$$s^\star = \min_s H(s).$$

The so-called ground state

Quantum adiabatic protocol

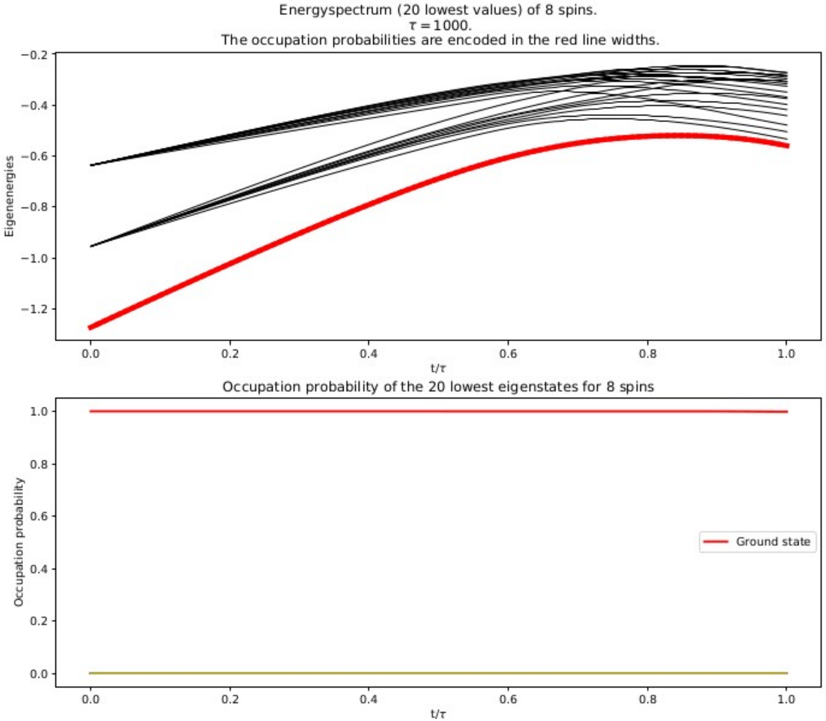
Initial Hamiltonian

$$H_0 = -\sum_{i \in \mathcal{I}} h_i \sigma_x^{(i)}.$$

Time dependent Hamiltonian

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_0 + \frac{t}{\tau} H_p$$

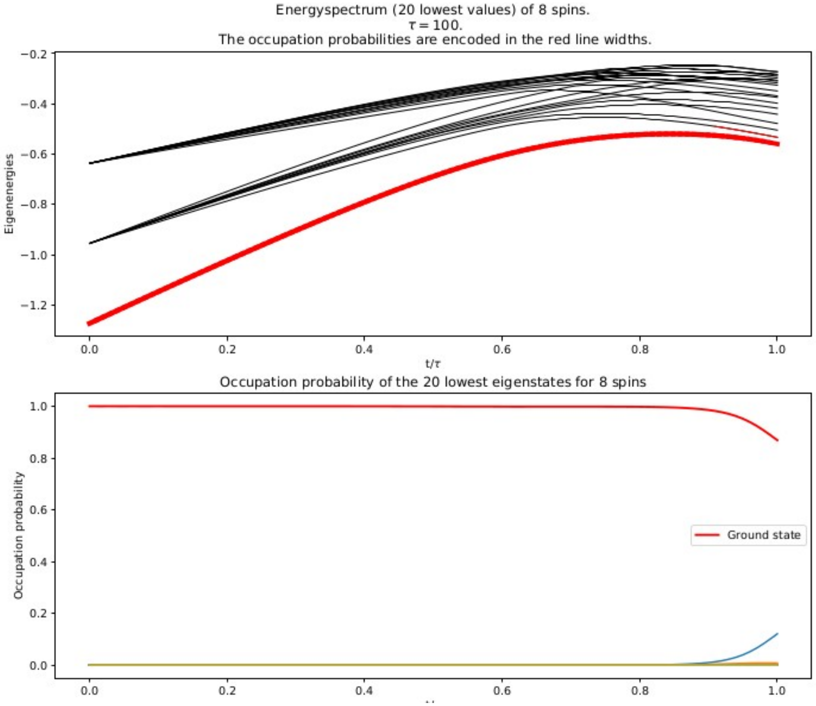
Adiabatic evolution - example



Energyspectrum (20 lowest values) of 8 spins.
$\tau = 1000$.
The occupation probabilities are encoded in the red line widths.

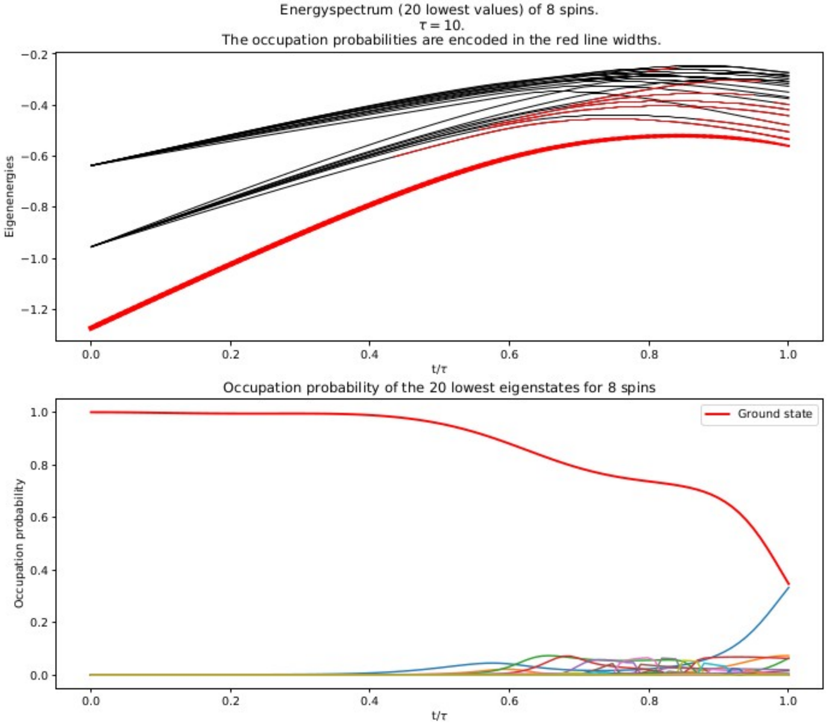
Occupation probability of the 20 lowest eigenstates for 8 spins

Adiabatic evolution - example



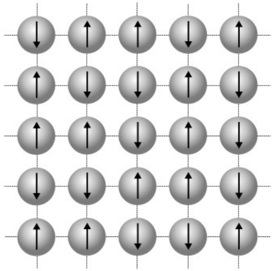
Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

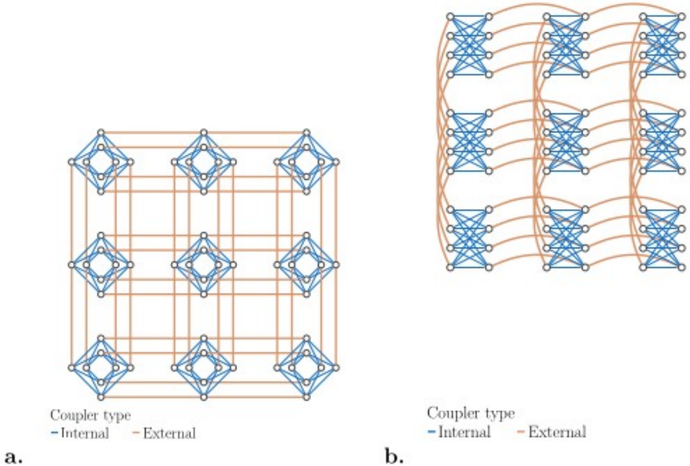
Adiabatic evolution - example



Simulating annealing of spin glasses

Spin glass simulation

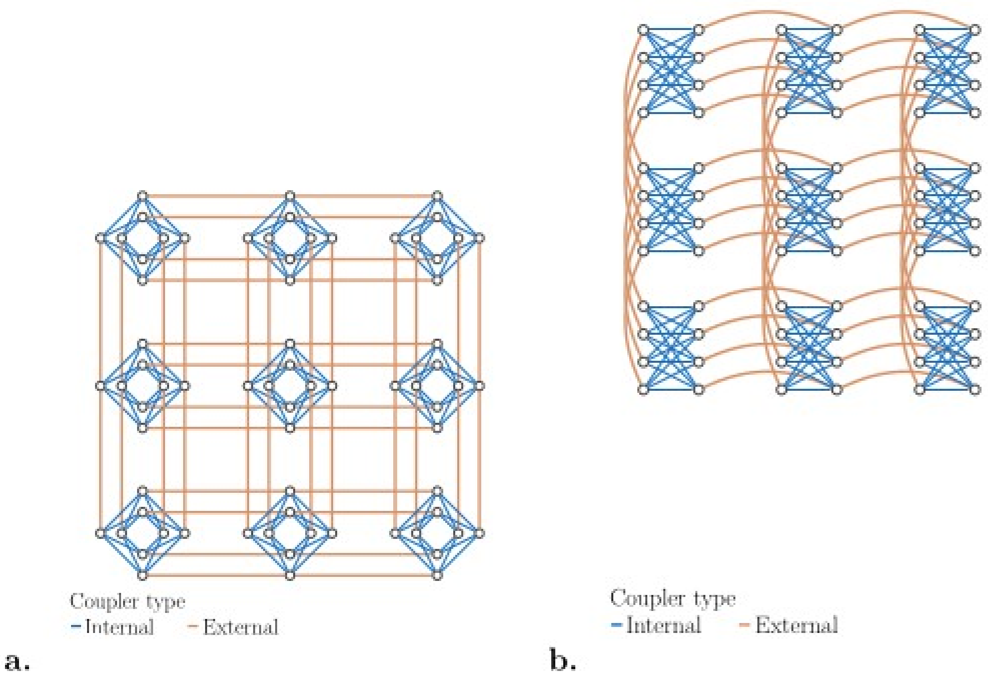




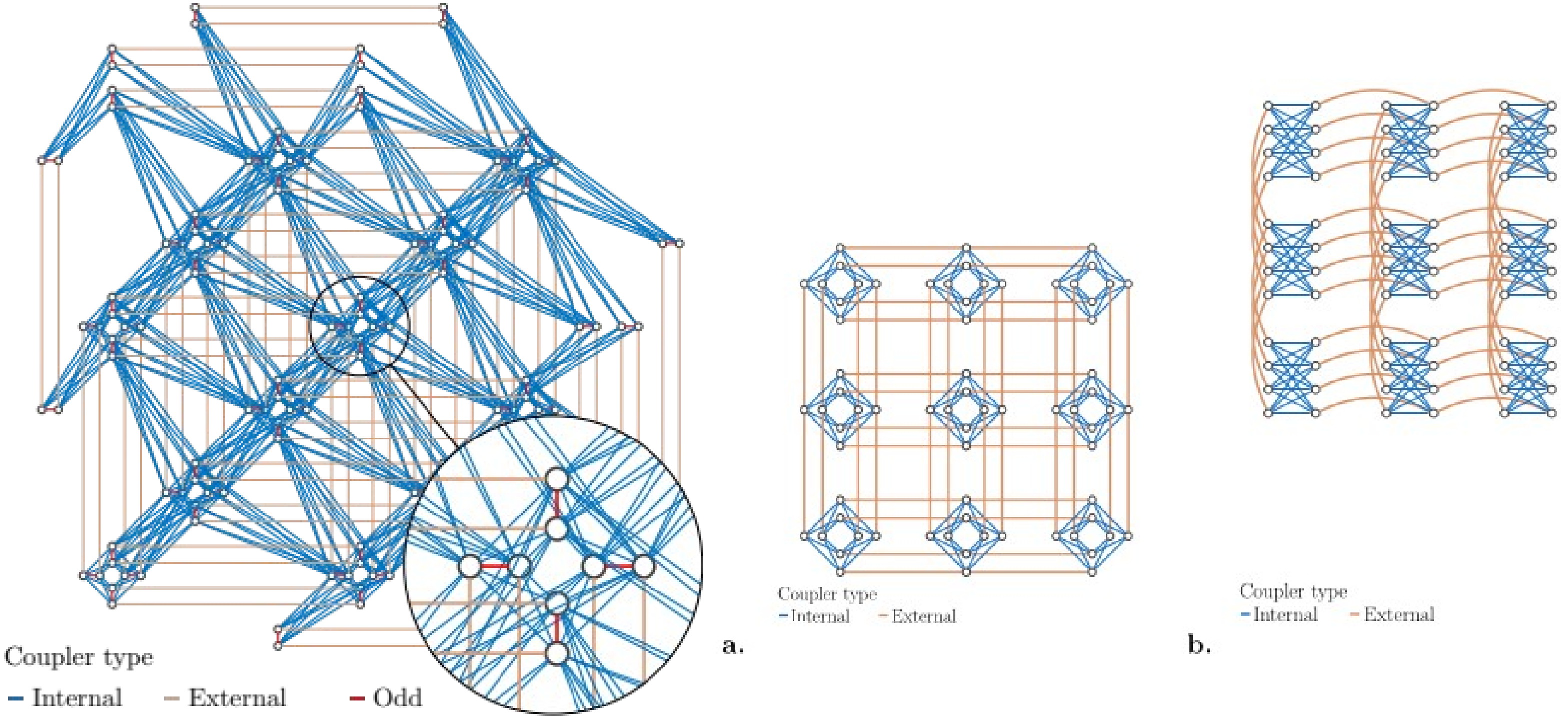
$$H(s) = -\sum_{i\in\mathcal{I}} h_i s_i - \sum_{(i,j)\in\mathcal{I}\times\mathcal{I}} J_{ij} s_i s_j ,$$

$$s = [s_i]_{i\in\mathcal{I}} \in \{-1,1\}^{\mathcal{I}},\, h_i \in \mathbb{R},\, J_{ij} \in \mathbb{R}$$

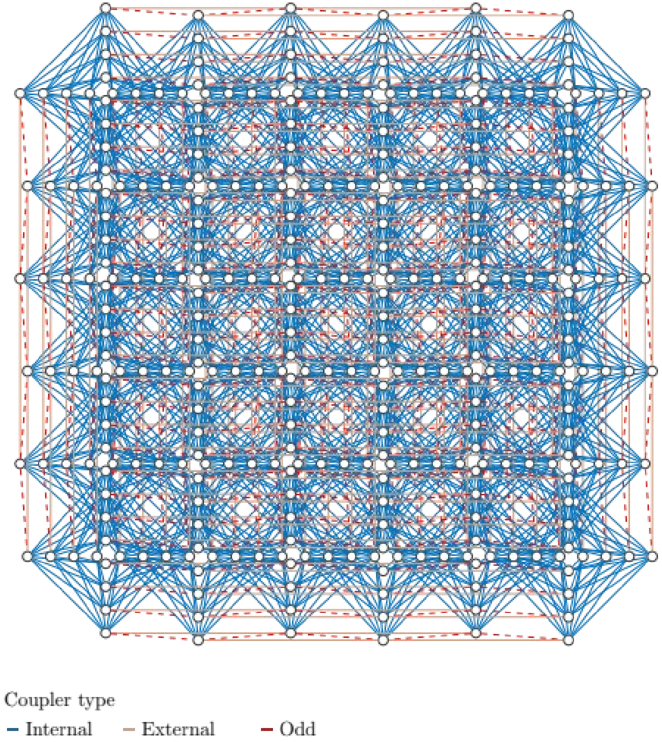
$$s^\star = \arg\min_s H(s).$$


Coupler type
Internal
External
a.
Coupler type
Internal
External
b.

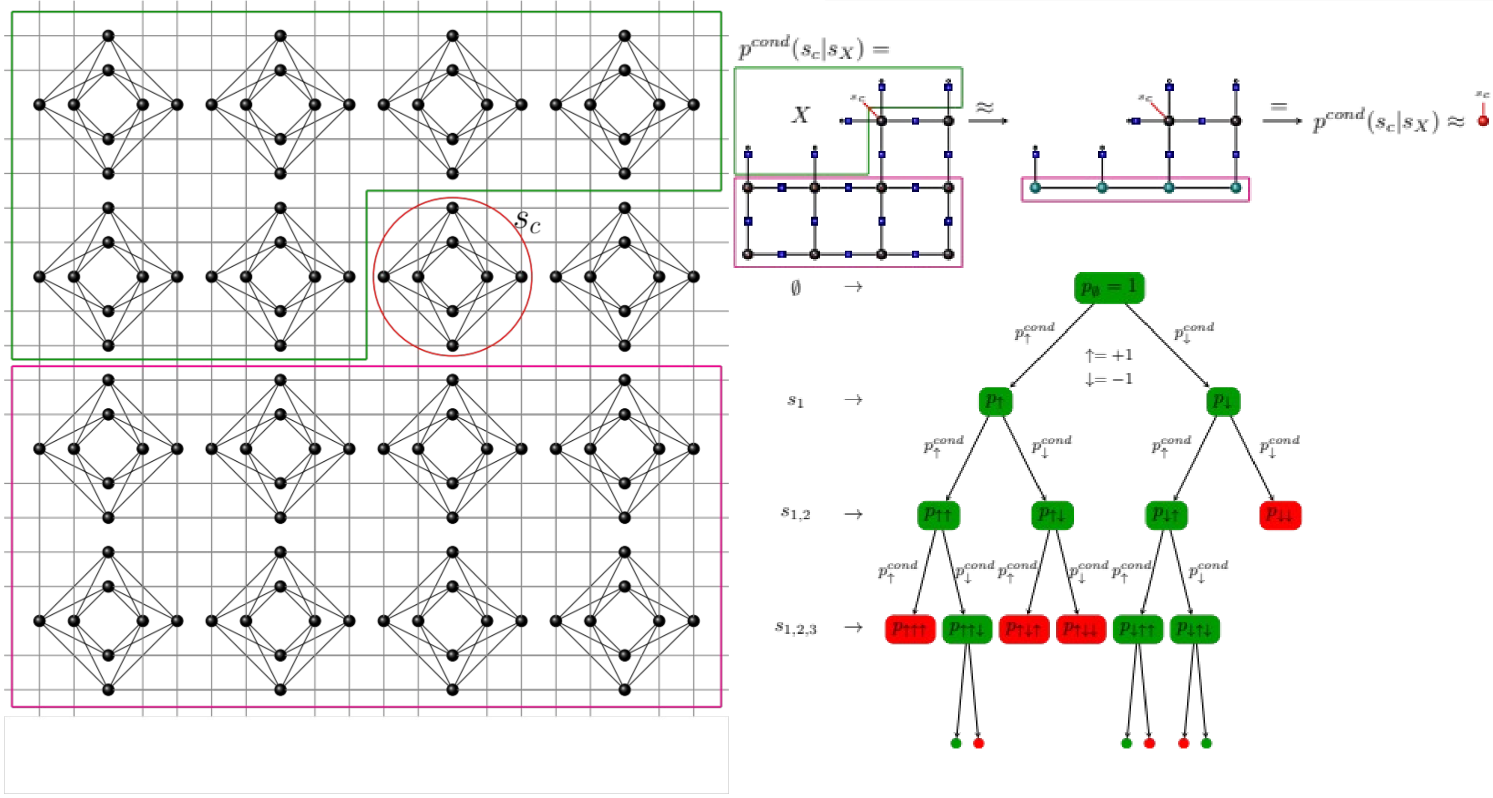
D-Wave Pegasus



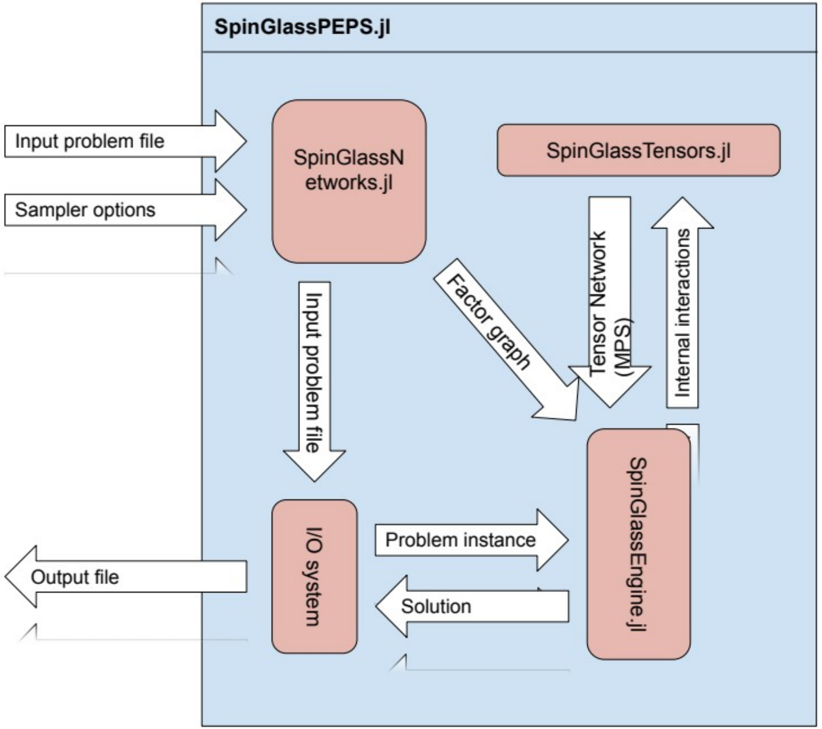
D-Wave Zephyr



Tensor networks

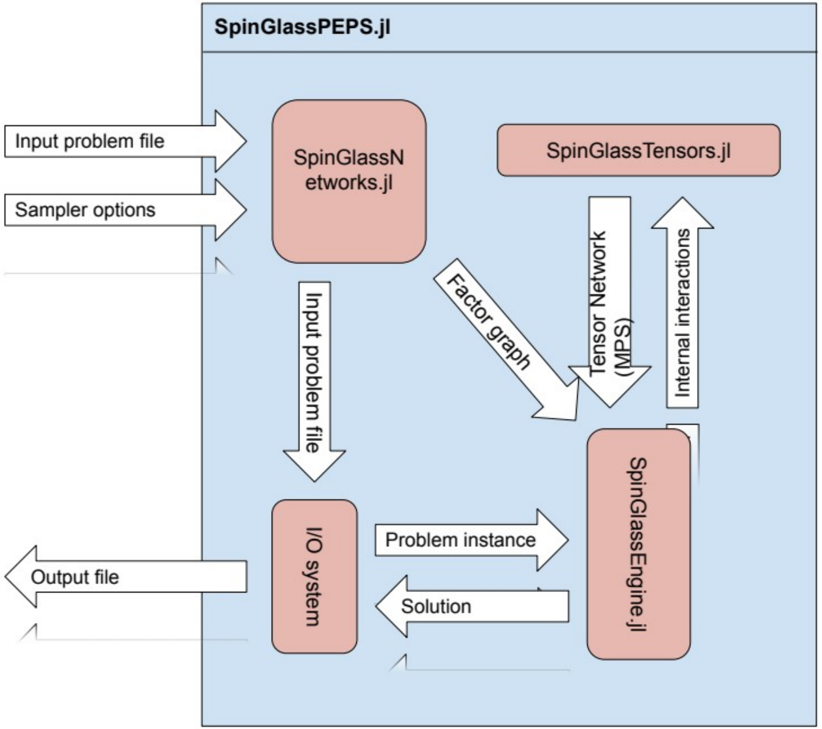

s_c
$p^{cond}(s_c|s_X) =$
X
s_c
$\approx$
s_c
$=$
$p^{cond}(s_c|s_X) \approx$
s_c
$\emptyset$ →
$p_\emptyset = 1$
$p^{cond}_\uparrow$
$p^{cond}_\downarrow$
$\uparrow = +1$
$\downarrow = -1$
s_1 →
$p_\uparrow$
$p_\downarrow$
$p^{cond}_\uparrow$
$p^{cond}_\downarrow$
$p^{cond}_\uparrow$
$p^{cond}_\downarrow$
$s_{1,2}$ →
$p_{\uparrow\uparrow}$
$p_{\uparrow\downarrow}$
$p_{\downarrow\uparrow}$
$p_{\downarrow\downarrow}$
$p^{cond}_\uparrow$
$p^{cond}_\downarrow$
$p^{cond}_\uparrow$
$p^{cond}_\downarrow$
$p^{cond}_\uparrow$
$p^{cond}_\downarrow$
$s_{1,2,3}$ →
$p_{\uparrow\uparrow\uparrow}$
$p_{\uparrow\uparrow\downarrow}$
$p_{\uparrow\downarrow\uparrow}$
$p_{\uparrow\downarrow\downarrow}$
$p_{\downarrow\uparrow\uparrow}$
$p_{\downarrow\uparrow\downarrow}$

Simulation of spin glass annealing



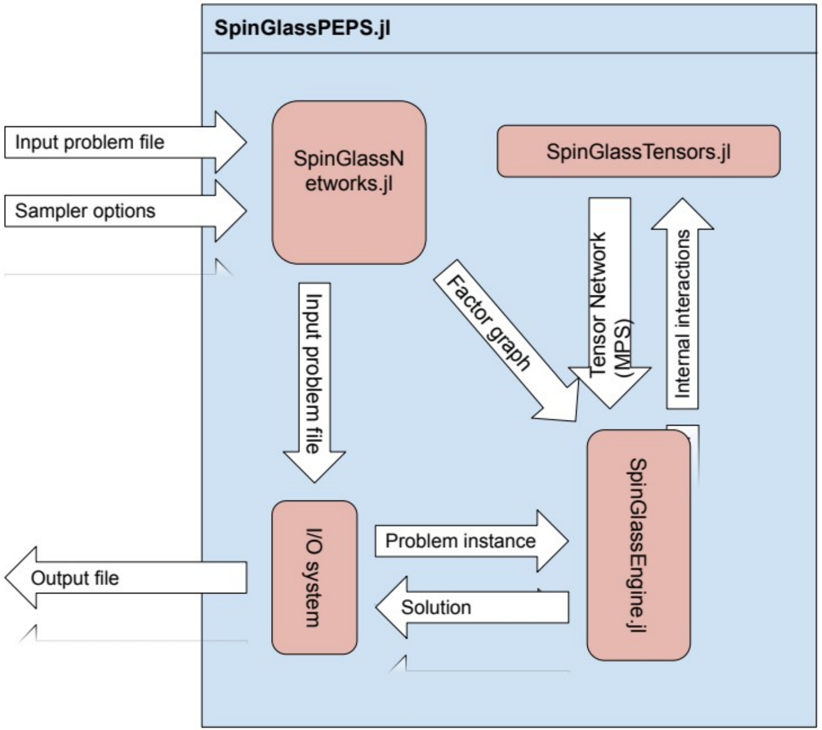
SpinGlassTensors.jl

- Tensor structures
- Opeation on tensors
- GPU utilization



SpinGlassNetworks.jl

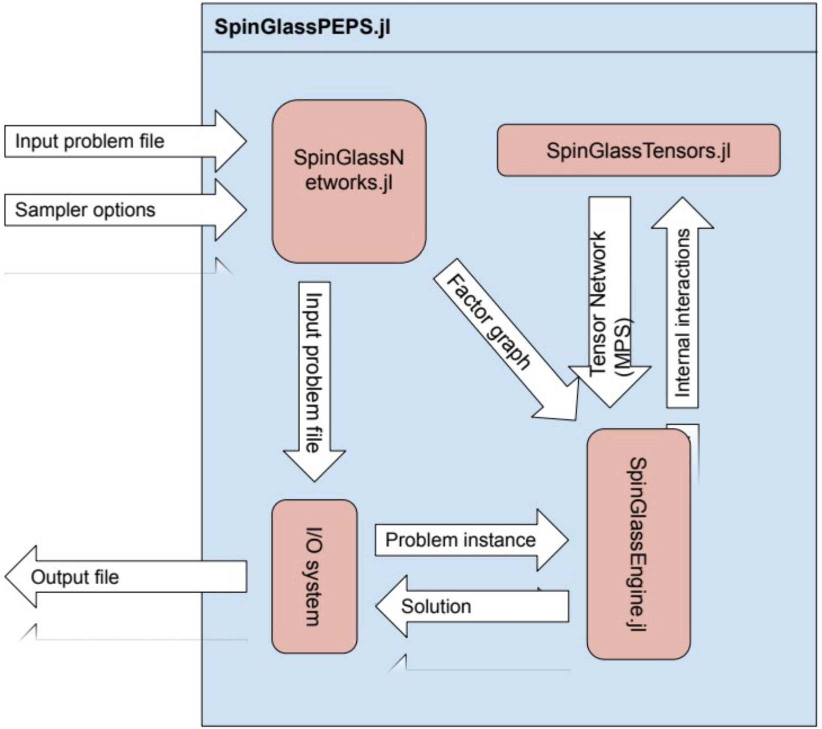
- Network graphs
- Factor graphs
- Basic I/O



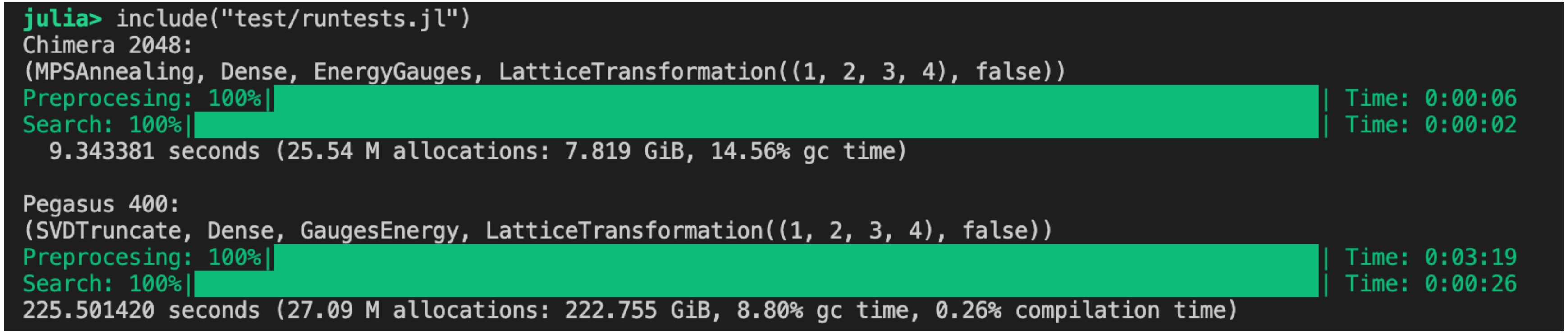
Fundusze Europejskie Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Rozwoju Regionalnego

SpinGlassEngine.jl

- Main algorithms
- Efficient simulation of Chimera
- Possibility to simulate more complex structures



SpinGlassEngine.jl

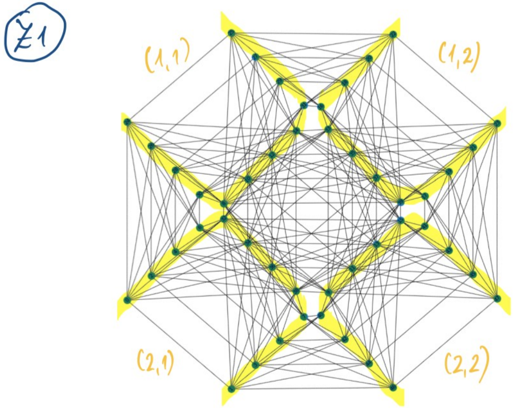


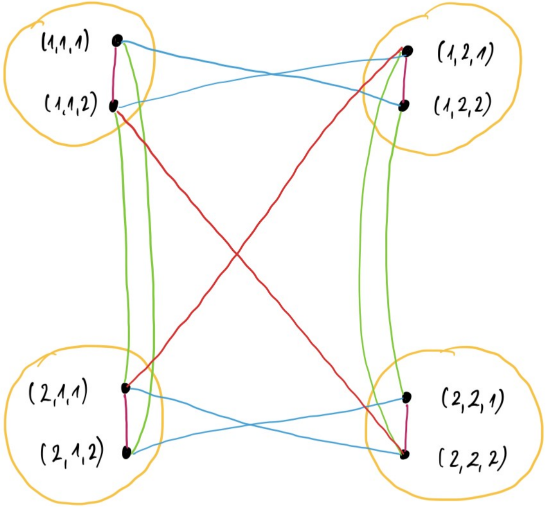
```
julia> include("test/runtests.jl")
Chimera 2048:
(MPSAnnealing, Dense, EnergyGauges, LatticeTransformation((1, 2, 3, 4), false))
Preprocesing: 100%|██████████████████████████████████████████| Time: 0:00:06
Search: 100%|██████████████████████████████████████████████████| Time: 0:00:02
  9.343381 seconds (25.54 M allocations: 7.819 GiB, 14.56% gc time)

Pegasus 400:
(SVDTruncate, Dense, GaugesEnergy, LatticeTransformation((1, 2, 3, 4), false))
Preprocesing: 100%|██████████████████████████████████████████| Time: 0:03:19
Search: 100%|██████████████████████████████████████████████████| Time: 0:00:26
225.501420 seconds (27.09 M allocations: 222.755 GiB, 8.80% gc time, 0.26% compilation time)
```

SpinGlassEngine.jl

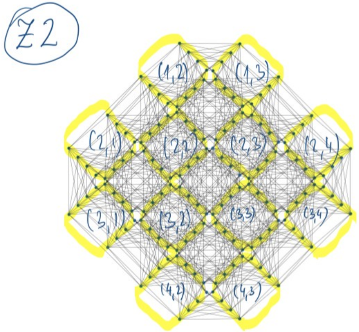
Solution: sparse tensor networks on the GPU

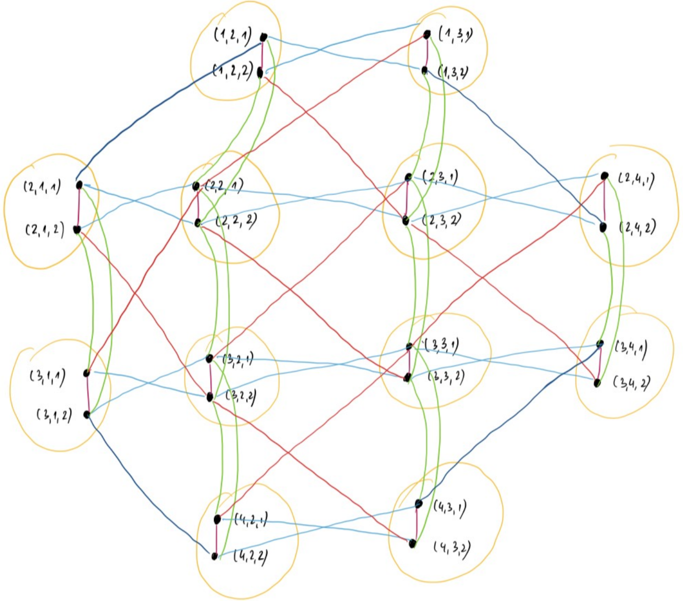




SpinGlassEngine.jl

Solution: sparse tensor networks on the GPU



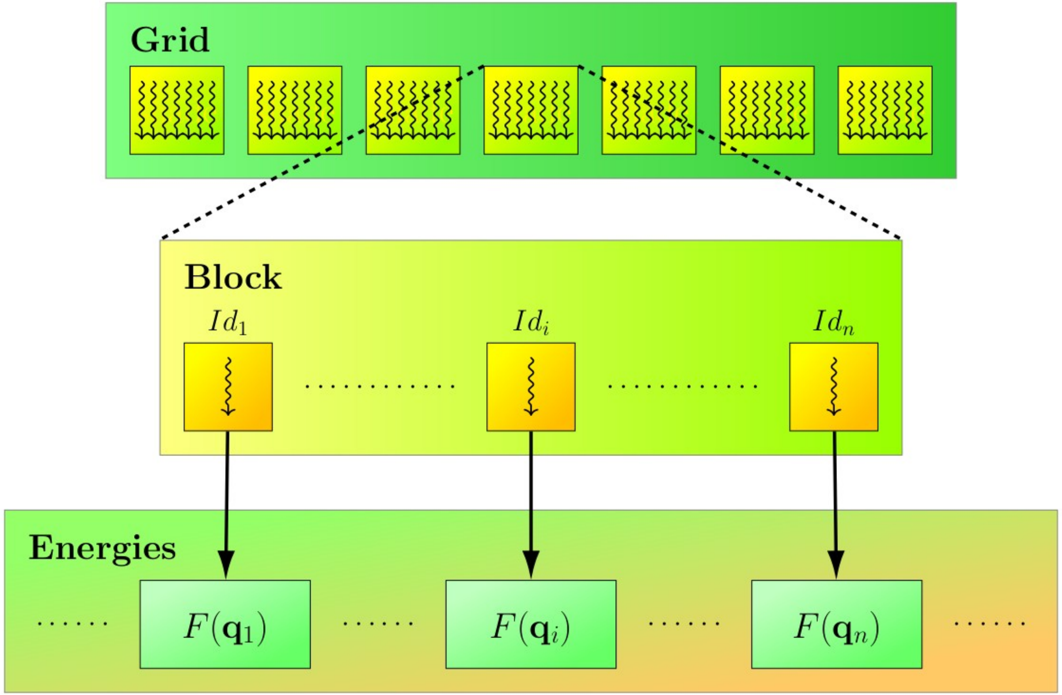


Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Bruteforce search

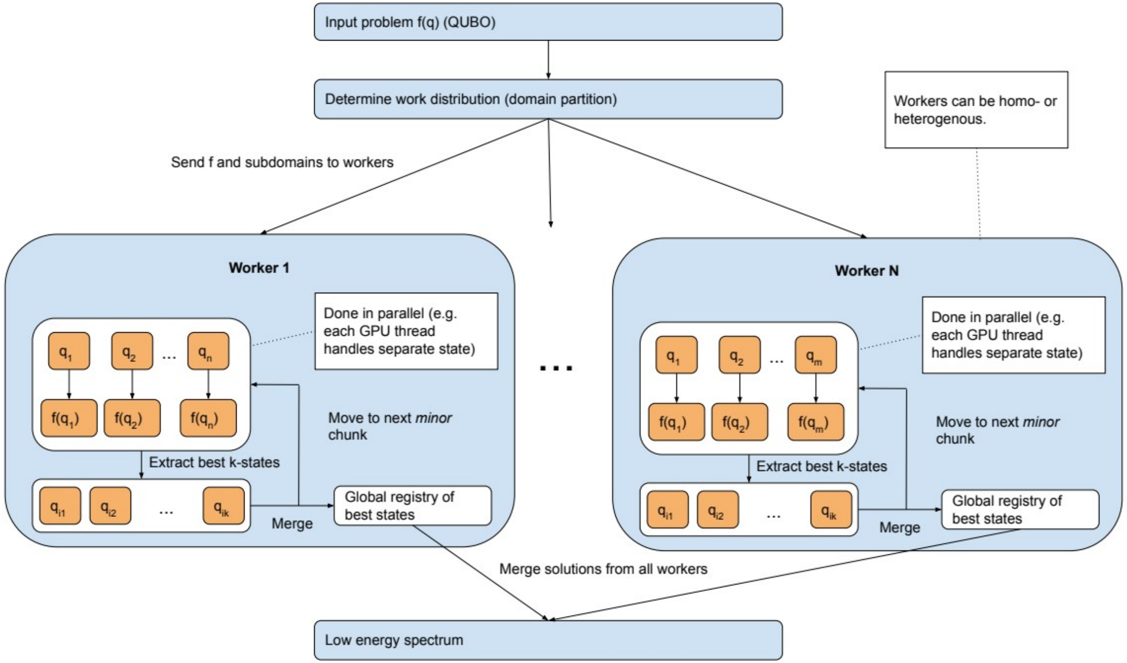
Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Bruteforce search



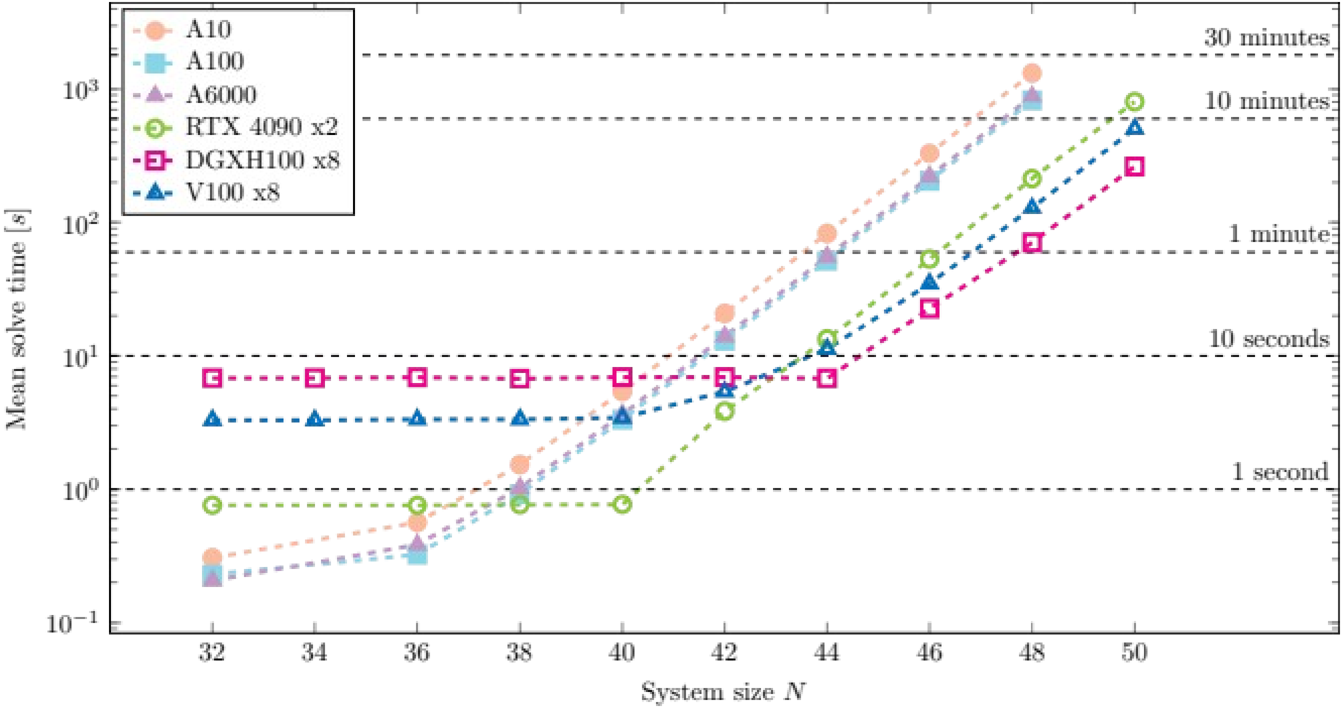
SpinGlassExhaustive.jl, omnisolver-bruteforce

- Parallel GPU implementation
- Two appraoches:
    - naive brute-force
    - Gray codes
- Main problem: parallel Sorting on the GPU


Input problem f(q) (QUBO)
Determine work distribution (domain partition)
Workers can be homo- or heterogenous.
Send f and subdomains to workers
Worker 1
q_1
q_2
...
q_n
$f(q_1)$
$f(q_2)$
$f(q_n)$
Done in parallel (e.g. each GPU thread handles separate state)
Move to next *minor* chunk
Extract best k-states
q_{i1}
q_{i2}
...
q_{ik}
Merge
Global registry of best states
...
Worker N
q_1
q_2
...
q_m
$f(q_1)$
$f(q_2)$
$f(q_m)$
Done in parallel (e.g. each GPU thread handles separate state)
Move to next *minor* chunk
Extract best k-states
q_{i1}
q_{i2}
...
q_{ik}
Merge
Global registry of best states
Merge solutions from all workers
Low energy spectrum

Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Execution times



Dynamical systems

Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

SpinGlassDynamics.jl

- Solving Ising instances via dynamical systems
- Main libraries:
  - DynamicalSystems.jl
  - DifferentialEquations.jl

$$y_i(t_{k+1}) = y_i(t_k) + \left\{ -\left[a_0 - a(t_k)\right] x_i(t_k) + c_0 \sum_{j=1}^{N} J_{i,j} \operatorname{sgn}\left[x_j(t_k)\right] \right\} \Delta_t \tag{17}$$

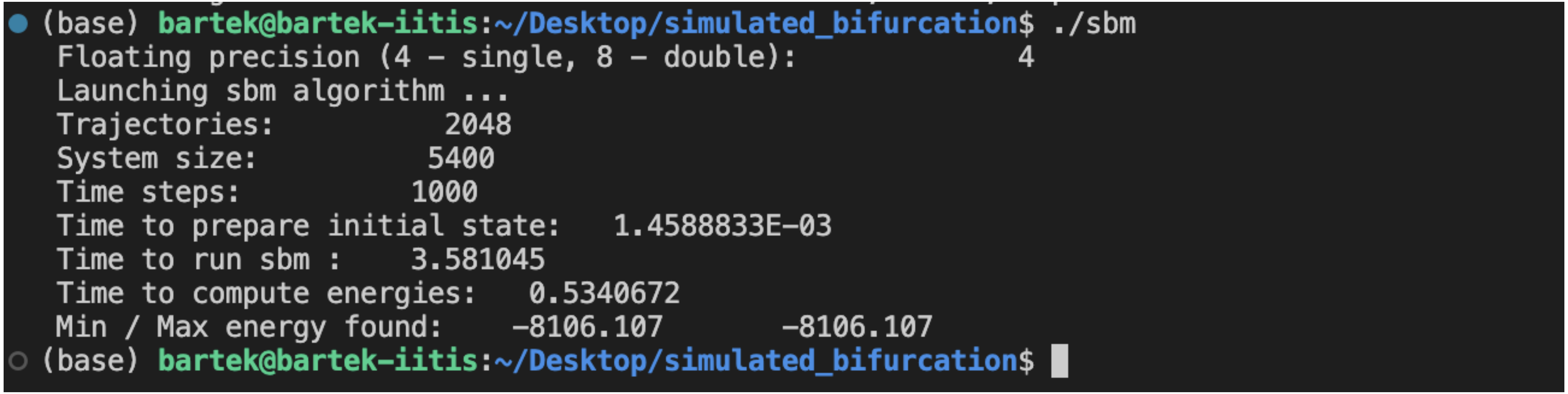
$$x_i(t_{k+1}) = x_i(t_k) + a_0 y_i(t_{k+1}) \Delta_t \tag{18}$$

Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

SpinGlassDynamics.jl

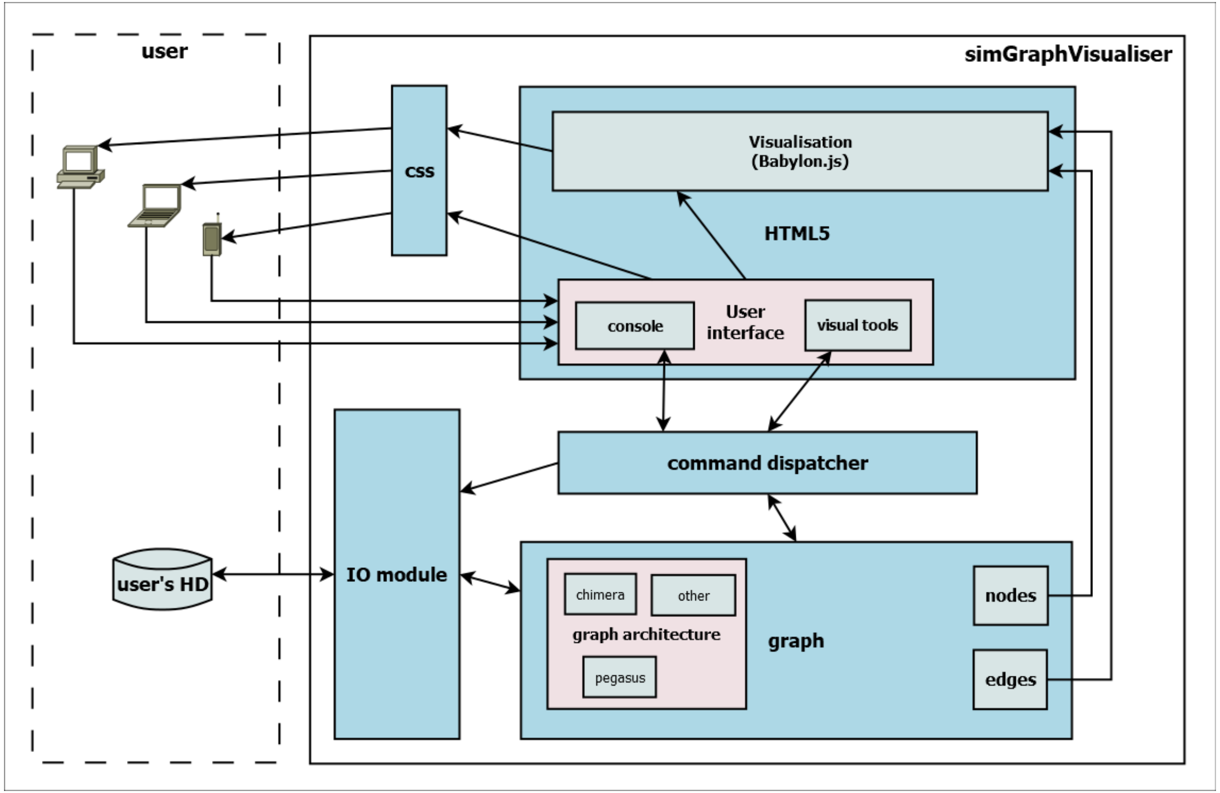
$$y_i(t_{k+1}) = y_i(t_k) + \left\{ -\left[a_0 - a(t_k)\right] x_i(t_k) + c_0 \sum_{j=1}^{N} J_{i,j} \,\mathrm{sgn}\left[x_j(t_k)\right] \right\} \Delta_t \tag{17}$$

$$x_i(t_{k+1}) = x_i(t_k) + a_0\, y_i(t_{k+1})\, \Delta_t \tag{18}$$



```
● (base) bartek@bartek-iitis:~/Desktop/simulated_bifurcation$ ./sbm
   Floating precision (4 - single, 8 - double):             4
   Launching sbm algorithm ...
   Trajectories:           2048
   System size:           5400
   Time steps:           1000
   Time to prepare initial state:   1.4588833E-03
   Time to run sbm :     3.581045
   Time to compute energies:   0.5340672
   Min / Max energy found:    -8106.107        -8106.107
○ (base) bartek@bartek-iitis:~/Desktop/simulated_bifurcation$ 
```

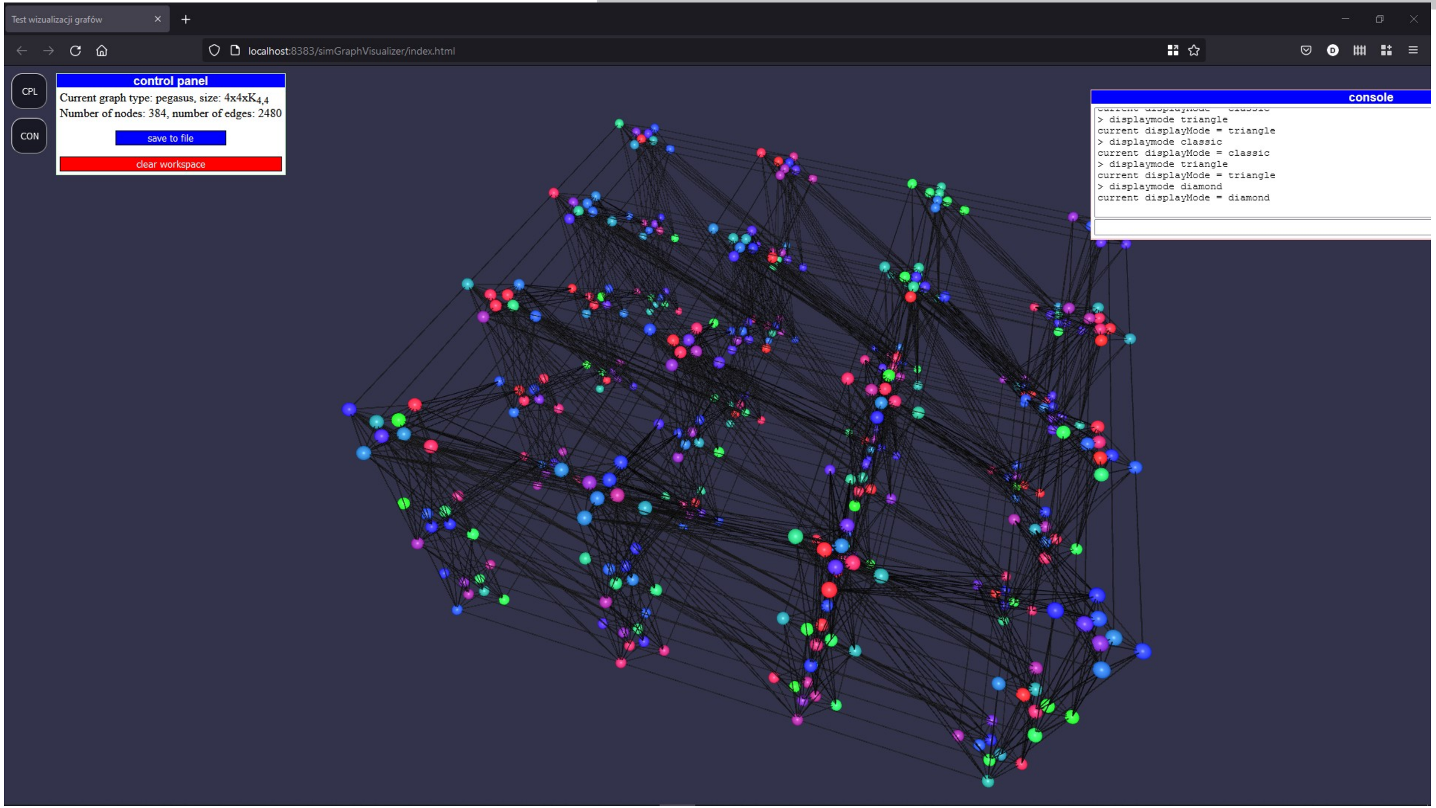
Visualization



Fundusze Europejskie
Inteligentny Rozwój

Rzeczpospolita Polska

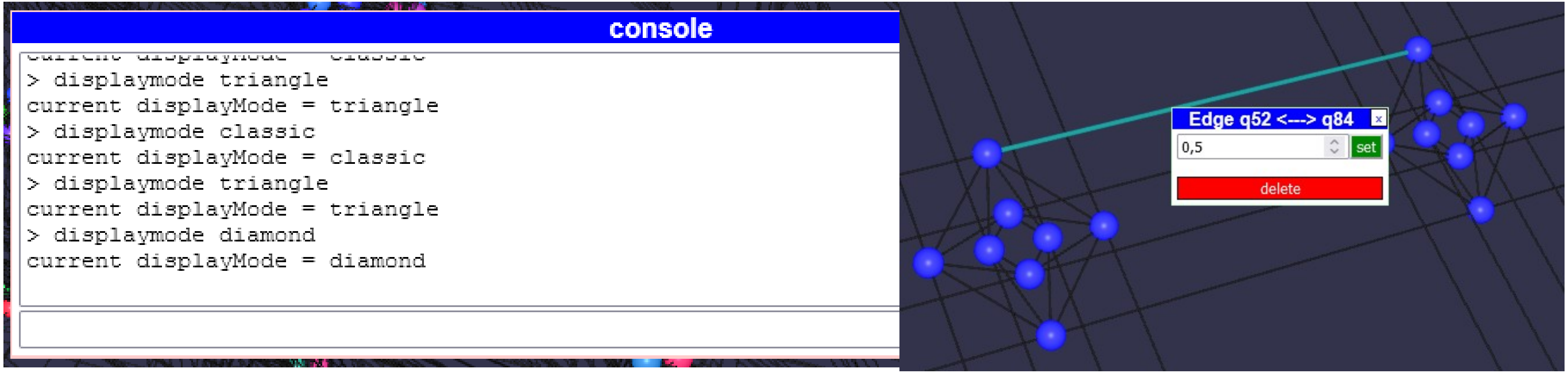
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego



Current graph type: pegasus, size: $4x4xK_{4,4}$

Number of nodes: 384, number of edges: 2480

```
current displayMode = classic
> displaymode triangle
current displayMode = triangle
> displaymode classic
current displayMode = classic
> displaymode triangle
current displayMode = triangle
> displaymode diamond
current displayMode = diamond
```

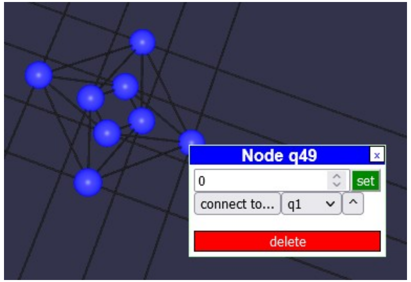


console

```
current displayMode = classic
> displaymode triangle
current displayMode = triangle
> displaymode classic
current displayMode = classic
> displaymode triangle
current displayMode = triangle
> displaymode diamond
current displayMode = diamond
```

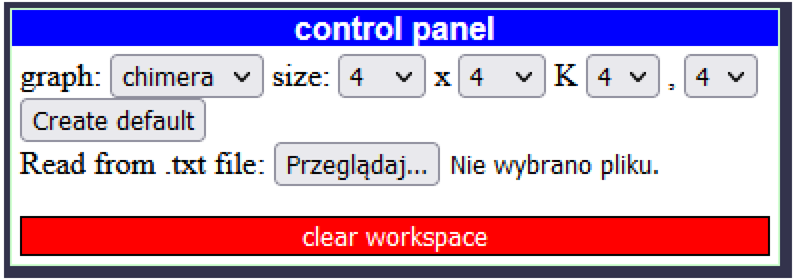
Edge q52 <---> q84

0,5



Node q49

0



control panel

graph: chimera size: 4 x 4 K 4 , 4

Read from .txt file: Przeglądaj... Nie wybrano pliku.

THANK YOU

Repositories

- https://github.com/euro-hpc-pl/SpinGlassPEPS.jl
- https://github.com/euro-hpc-pl/omnisolver-bruteforce
- https://github.com/euro-hpc-pl/SpinGlassDynamics.jl

Publication:
arXiv:2112.11131

Fundusze Europejskie
Inteligentny Rozwój
Rzeczpospolita Polska
Unia Europejska
Europejski Fundusz Rozwoju Regionalnego