Chaotic Optimization for Quadratic Assignment Problems

Tohru Ikeguchi[1], Keiichi Sato[1],

Mikio Hasegawa[2], and Kazuyuki Aihara[3]

tohru@ics.saitama-u.ac.jp.

[1] Saitama University,

[2] Communication Research Laboratory,

[3] The University of Tokyo

Introduction

Combinatorial Optimization Problems

Scheduling, Vehicle routing, Assignment problems, etc $\cdots$

$\downarrow$

Almost impossible to obtain optimal solutions

$\downarrow$

Develop effective algorithms, which offer very good near optimum solutions in a reasonable time.

Quadratic Assignment Problem

One of the most difficult NP-hard combinatorial optimization problems.

$$F(\boldsymbol{p}) = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} d_{p(i)p(j)}$$

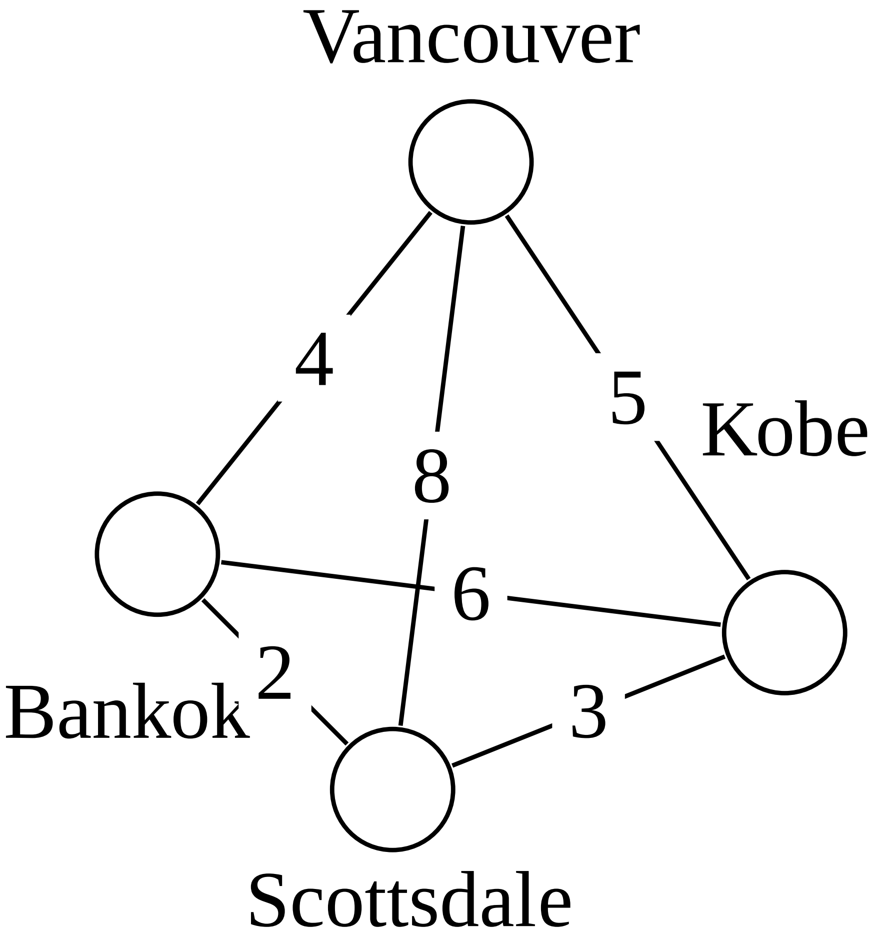
c_{ij}	:	the (i, j) th element of a flow matrix $\boldsymbol{C}$
d_{ij}	:	the (i, j) th element of a distance matrix $\boldsymbol{D}$
p_i	:	the i th element of a permutation $\boldsymbol{p}$
N	:	the size of the problem

Find a permutation $\boldsymbol{p}$, which provides the minimum value of the objective function F.

Example of QAP

1. Assign 4 factories, X, Y, Z and W, to the four cities
2. The amounts of items between 4 factories

$$
\begin{array}{c} \\ \boldsymbol{C} = \end{array}
\begin{array}{c|cccc} & X & Y & Z & W \\ \hline X & 0 & 4 & 3 & 5 \\ Y & 4 & 0 & 6 & 3 \\ Z & 3 & 6 & 0 & 4 \\ W & 5 & 3 & 4 & 0 \end{array}
$$



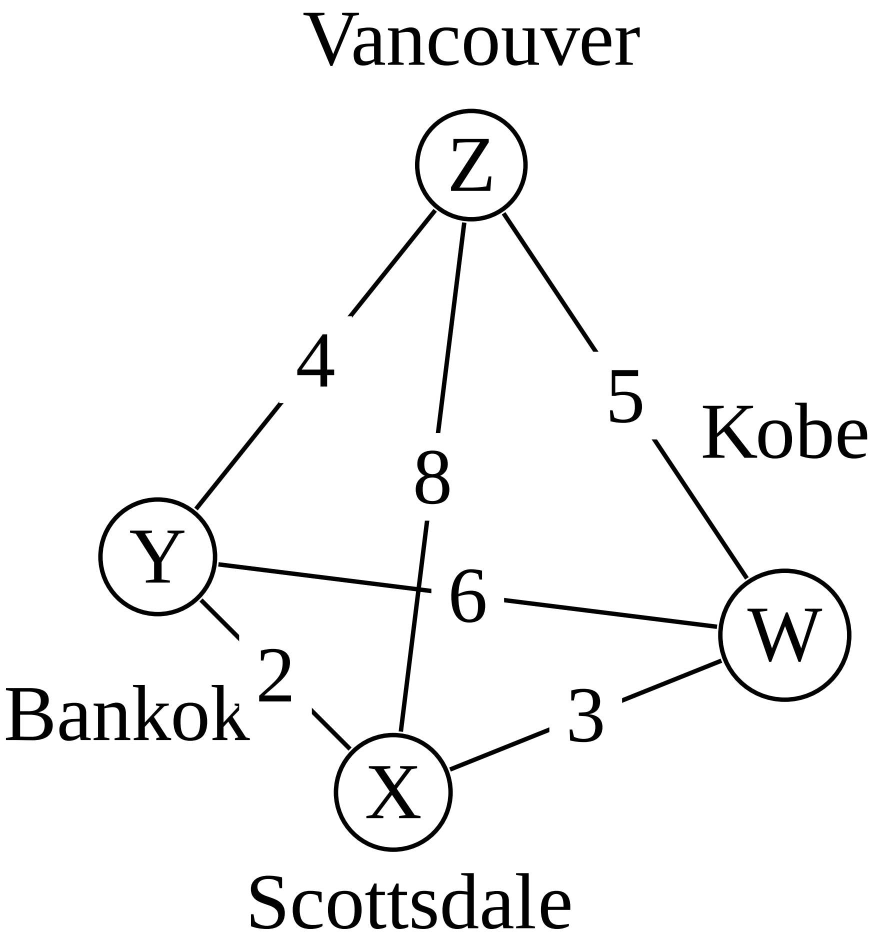
3. Distance

$$
\begin{array}{c} \\ \boldsymbol{D} = \end{array}
\begin{array}{c|cccc} & S & B & V & K \\ \hline S & 0 & 2 & 8 & 3 \\ B & 2 & 0 & 4 & 6 \\ V & 8 & 4 & 0 & 5 \\ K & 3 & 6 & 5 & 0 \end{array}
$$

Example of QAP

Assignment

X	Scottsdale
Y	Bankok
Z	Vancouver
W	Kobe

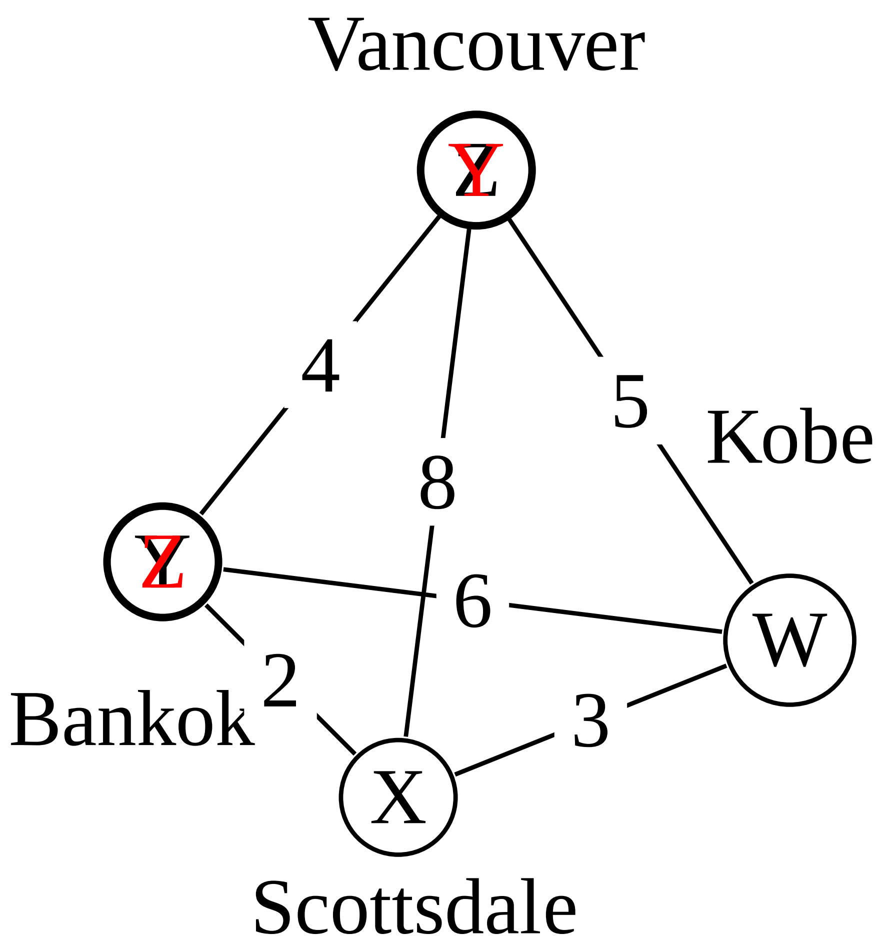


$$F(\boldsymbol{p}_1) = 4 \cdot 2 + 3 \cdot 8 + 5 \cdot 3 + 4 \cdot 6 + 3 \cdot 6 + 4 \cdot 5 = 109$$

Example of QAP

Assignment

X	Scottsdale
Z	Bankok
Y	Vancouver
W	Kobe

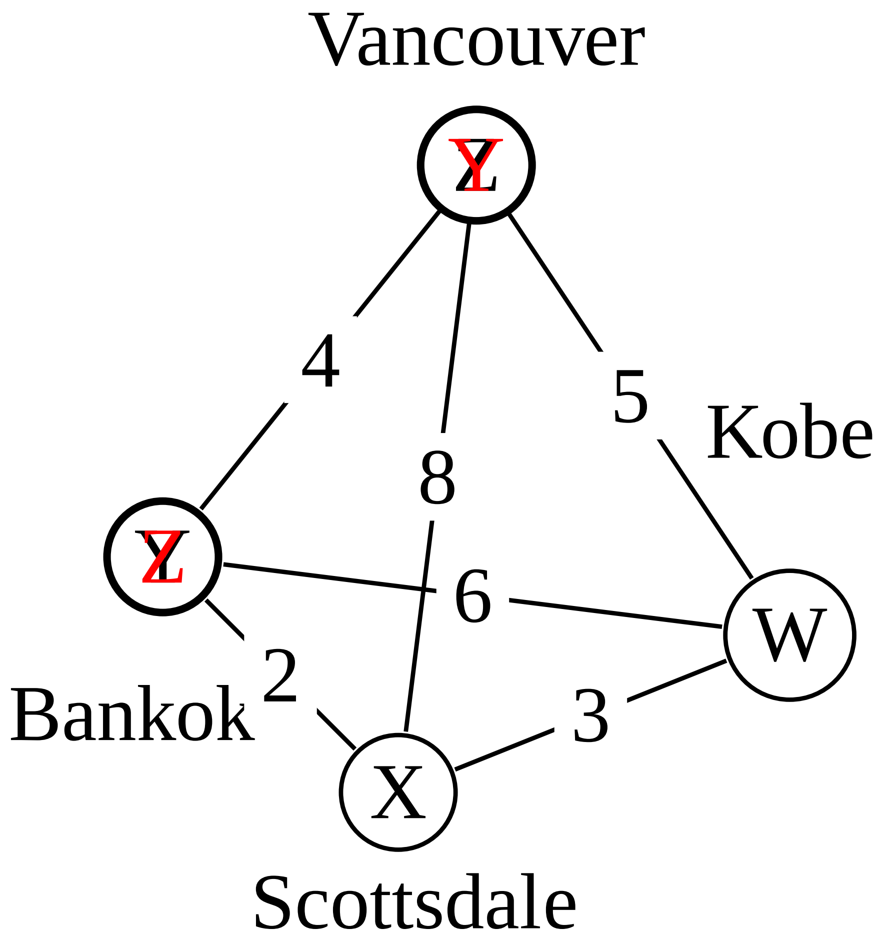


$$F(\boldsymbol{p}_1) = 4 \cdot 2 + 3 \cdot 8 + 5 \cdot 3 + 4 \cdot 6 + 3 \cdot 6 + 4 \cdot 5 = 109$$
$$F(\boldsymbol{p}_2) = 4 \cdot 8 + 3 \cdot 2 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 5 + 4 \cdot 6 = 116$$

Example of QAP

Assignment

X	Scottsdale
Z	Bankok
Y	Vancouver
W	Kobe



$$F(\boldsymbol{p}_1) = 4 \cdot 2 + 3 \cdot 8 + 5 \cdot 3 + 4 \cdot 6 + 3 \cdot 6 + 4 \cdot 5 = 109$$
$$F(\boldsymbol{p}_2) = 4 \cdot 8 + 3 \cdot 2 + 5 \cdot 3 + 6 \cdot 4 + 3 \cdot 5 + 4 \cdot 6 = 116$$

$$F(\boldsymbol{p}_1) < F(\boldsymbol{p}_2)$$

One of the conventional method

Approach based on the gradient dynamics of
mutual connection neural networks
(Hopfield – Tank neural network approach)

- Good news
  We can obtain good solutions (possibly optimum) if we can decide connection weights with appropriate initial conditions.

Coding

- Solving an N-size problem, $N \times N$ neurons are prepared.
- The (i, m)-th neuron firing ($x_{im} = 1$) assigns the i-th element to the j-th element.

	S	B	V	K
X	○	○	○	○
Y	○	○	○	○
Z	○	○	○	○
W	○	○	○	○

Coding

- Solving an N-size problem, $N \times N$ neurons are prepared.
- The (i, m)-th neuron firing ($x_{im} = 1$) assigns the i-th element to the j-th element.

	S	B	V	K
X	1	0	0	0
Y	0	1	0	0
Z	0	0	1	0
W	0	0	0	1

$1 \longrightarrow$ Firing, $0 \longrightarrow$ Resting.

X : Scottsdale, Y : Bankok, Z : Vancouver and W : Kobe.

Coding

- Solving an N-size problem, $N \times N$ neurons are prepared.
- The (i, m)-th neuron firing ($x_{im} = 1$) assigns the i-th element to the j-th element.

	S	B	V	K
X	0	0	1	0
Y	0	1	0	0
Z	0	0	0	1
W	1	0	0	0

$1 \longrightarrow$ Firing, $0 \longrightarrow$ Resting.

X : Vancouver, Y : Bankok, Z : Kobe and W : Scottsdale.

Connection Weights

$$
\begin{aligned}
F(x) &= A\sum_{i=1}^{N}(\sum_{m=1}^{N} x_{im} - 1)^2 + B\sum_{m=1}^{N}(\sum_{i=1}^{N} x_{im} - 1)^2 \\
&+ \sum_{i=1}^{N}\sum_{m=1}^{N}\sum_{j=1}^{N}\sum_{n=1}^{N} c_{ij} d_{mn} x_{im} x_{jn}, \\
E(x) &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{m=1}^{N}\sum_{j=1}^{N}\sum_{n=1}^{N} w_{im;jn} x_{im} x_{jn} + \sum_{i=1}^{N}\sum_{m=1}^{N} \theta_{im} x_{im} \\
w_{im;jn} &= -2\{A(1-\delta_{mn})\delta_{ij} + B\delta_{mn}(1-\delta_{ij}) + c_{ij} d_{mn}\}
\end{aligned}
$$

A, B : positive constants, δ_{ij} : Kronecker's delta

One of the conventional method

Approach based on the gradient dynamics of mutual connection neural networks (Hopfield – Tank neural network approach)

- Good news
  We can obtain good solutions (possibly optimum) if we can decide connection weights with appropriate initial conditions.

One of the conventional method

Approach based on the gradient dynamics of mutual connection neural networks (Hopfield – Tank neural network approach)

- Good news
  We can obtain good solutions (possibly optimum) if we can decide connection weights with appropriate initial conditions.
- Bad news

1. Difficult to apply to large scale size problems
   N-size problem $\rightarrow N^2$ neurons $\rightarrow N^4$ connections
2. Unfeasible solutions
3. Local minimum problem

Chaos for combinatorial optimization

Chaos for combinatorial optimization

- Chaotic dynamics for escaping from local minima.

Chaos for combinatorial optimization

- Chaotic dynamics for escaping from local minima.
- Fluctuation in a state space possibly leads to an optimum solution.

Chaos for combinatorial optimization

- Chaotic dynamics for escaping from local minima.
- Fluctuation in a state space possibly leads to an optimum solution.

Mutual connection NN $\rightarrow$ CNN for TSPs

Nozawa,1992

Yamada et al, 1993

Chen & Aihara, 1995

Ishii et al., 1997

Chaos for combinatorial optimization

- Chaotic dynamics for escaping from local minima.
- Fluctuation in a state space possibly leads to an optimum solution.

Mutual connection NN $\rightarrow$ CNN for TSPs

Nozawa,1992

Yamada et al, 1993

Chen & Aihara, 1995

Ishii et al., 1997

- CNN for QAPs

Chaotic neural network

$$y_{im}(t+1) = k y_{im}(t) + \sum_{j=1}^{N} \sum_{n=1}^{N} w_{im;jn} x_{jn}(t) - \alpha f(y_{im}(t)) + a_{im}$$

$y_{im}(t)$	:	the internal state of the (i, m) th neuron at t
$w_{im;jn}$	:	the connection weight
f	:	output function, $f(z) = 1/(1 + \exp(-z/\epsilon))$
k	:	the decay parameters for the refractoriness
α, a, ϵ	:	parameters

Use CNN to solve QAPs

$$F(x) = A\sum_{i=1}^{N}(\sum_{m=1}^{N} x_{im} - 1)^2 + B\sum_{m=1}^{N}(\sum_{i=1}^{N} x_{im} - 1)^2 + \sum_{i=1}^{N}\sum_{m=1}^{N}\sum_{j=1}^{N}\sum_{n=1}^{N} c_{ij}d_{mn}x_{im}x_{jn},$$

$$w_{im;jn} = -2\{A(1-\delta_{mn})\delta_{ij} + B\delta_{mn}(1-\delta_{ij}) + \frac{c_{ij}d_{mn}}{q}\}$$

q : a normalization parameter

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
					x_{1m}
					x_{2m}
					x_{3m}
					x_{4m}
					x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
					x_{1m}
					x_{2m}
					x_{3m}
					x_{4m}
					x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
					x_{1m}
					x_{2m}
					x_{3m}
					x_{4m}
					x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0					x_{2m}
0					x_{3m}
0					x_{4m}
0					x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0					x_{2m}
0					x_{3m}
0					x_{4m}
0					x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0					x_{2m}
0					x_{3m}
0					x_{4m}
0					x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0				x_{2m}
0	1	0	0	0	x_{3m}
0	0				x_{4m}
0	0				x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0				x_{2m}
0	1	0	0	0	x_{3m}
0	0				x_{4m}
0	0				x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0				x_{2m}
0	1	0	0	0	x_{3m}
0	0				x_{4m}
0	0				x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0		0		x_{2m}
0	1	0	0	0	x_{3m}
0	0		0		x_{4m}
0	0	0	1	0	x_{5m}

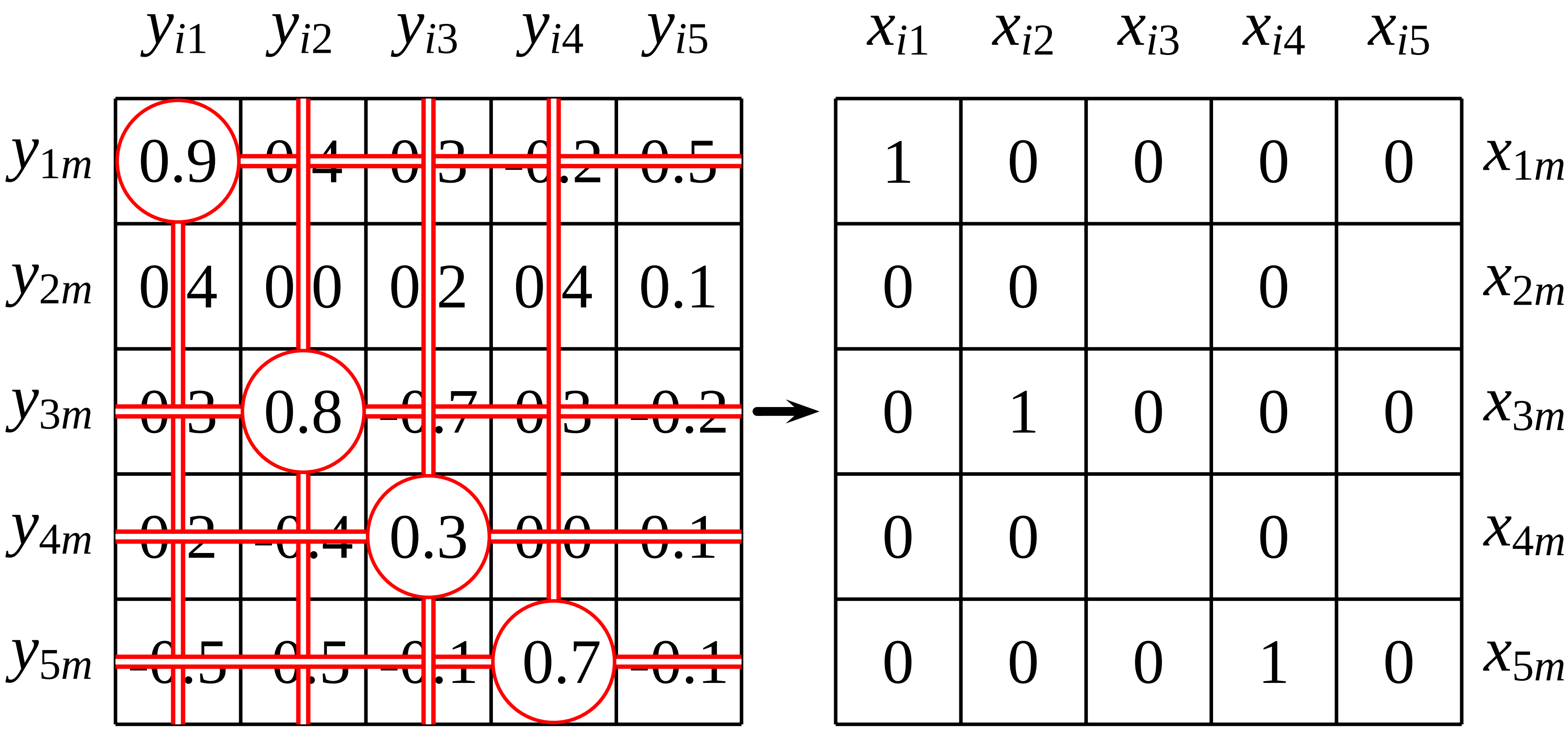
Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0		0		x_{2m}
0	1	0	0	0	x_{3m}
0	0		0		x_{4m}
0	0	0	1	0	x_{5m}

Definition of firing



Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0	0	0		x_{2m}
0	1	0	0	0	x_{3m}
0	0	1	0	0	x_{4m}
0	0	0	1	0	x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0	0	0		x_{2m}
0	1	0	0	0	x_{3m}
0	0	1	0	0	x_{4m}
0	0	0	1	0	x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0	0	0	1	x_{2m}
0	1	0	0	0	x_{3m}
0	0	1	0	0	x_{4m}
0	0	0	1	0	x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0	0	0	1	x_{2m}
0	1	0	0	0	x_{3m}
0	0	1	0	0	x_{4m}
0	0	0	1	0	x_{5m}

Definition of firing

	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}
y_{1m}	0.9	0.4	0.3	-0.2	0.5
y_{2m}	0.4	0.0	0.2	0.4	0.1
y_{3m}	0.3	0.8	-0.7	0.3	-0.2
y_{4m}	0.2	-0.4	0.3	0.0	0.1
y_{5m}	-0.5	0.5	-0.1	0.7	-0.1

→

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	
1	0	0	0	0	x_{1m}
0	0	0	0	1	x_{2m}
0	1	0	0	0	x_{3m}
0	0	1	0	0	x_{4m}
0	0	0	1	0	x_{5m}

This coding scheme always offers feasible solutions.

↓

Higher solvable performance

Benchmark problems

- From QAPLIB
  (URL: http://www.opt.math.tu-graz.ac.at/qaplib/)

1. Nug

$$N = 12, 14, 15, 16, 17, 20, 21, 22, 24, 25, 27, 30$$

2. Had

$$N = 12, 14, 16, 18, 20$$

Chaotic Dynamics and Solvable Performance

We evaluate the solvable performance on benchmark problems by calculating

1. Lyapunov dimension
2. sum of positive Lyapunov exponents
3. the number of positive Lyapunov exponents

for 1,000 parameter sets (a, k).

$$3 \leq a \leq 8, 0.8 \leq k < 1$$

$$A = B = 0.5, \alpha = 1.01, \epsilon = 0.02,$$

Estimating Lyapunov Exponents

$$
\begin{aligned}
\boldsymbol{y}(t+1) &= \boldsymbol{F}(\boldsymbol{y}(t)) \\
\boldsymbol{y}(t+1) + \Delta\boldsymbol{y}(t+1) &= \boldsymbol{F}(\boldsymbol{y}(t) + \Delta\boldsymbol{y}(t)) \\
\Delta\boldsymbol{y}(t+1) &= \boldsymbol{DF}(\boldsymbol{y}(t))\Delta\boldsymbol{y}(t)
\end{aligned}
$$

$\boldsymbol{DF}(\boldsymbol{y}(t))$: Jacobian matrix at $\boldsymbol{y}(t)$

$$
\boldsymbol{y}(t) = \begin{pmatrix} y_{11}(t) \\ y_{12}(t) \\ y_{13}(t) \\ \cdots \\ y_{NN}(t) \end{pmatrix} \in \mathbf{R}^{N^2}
$$

QR decomposition

$$
\begin{aligned}
\boldsymbol{DF}(\boldsymbol{y}(0)) &= \boldsymbol{Q}_1\boldsymbol{R}_1 \\
\boldsymbol{DF}(\boldsymbol{y}(1))\boldsymbol{Q}_1 &= \boldsymbol{Q}_2\boldsymbol{R}_2 \\
\boldsymbol{DF}(\boldsymbol{y}(2))\boldsymbol{Q}_2 &= \boldsymbol{Q}_3\boldsymbol{R}_3 \\
&\cdots \\
\boldsymbol{DF}(\boldsymbol{y}(t))\boldsymbol{Q}_t &= \boldsymbol{Q}_{t+1}\boldsymbol{R}_{t+1} \\
&\cdots
\end{aligned}
$$

$$\lambda_i = \lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n}\log|R_t^{ii}|$$

where R_t^{ii} is the i-th diagonal element of $\boldsymbol{R}_t$.

Lyapunov dimension

- Lyapunov dimension

$$D_L = j + \frac{\sum_{i=1}^{j} \lambda_i}{\lambda_{j+1}}, \; j = \max_k \{ \sum_{i=1}^{k} \lambda_i > 0 \}$$

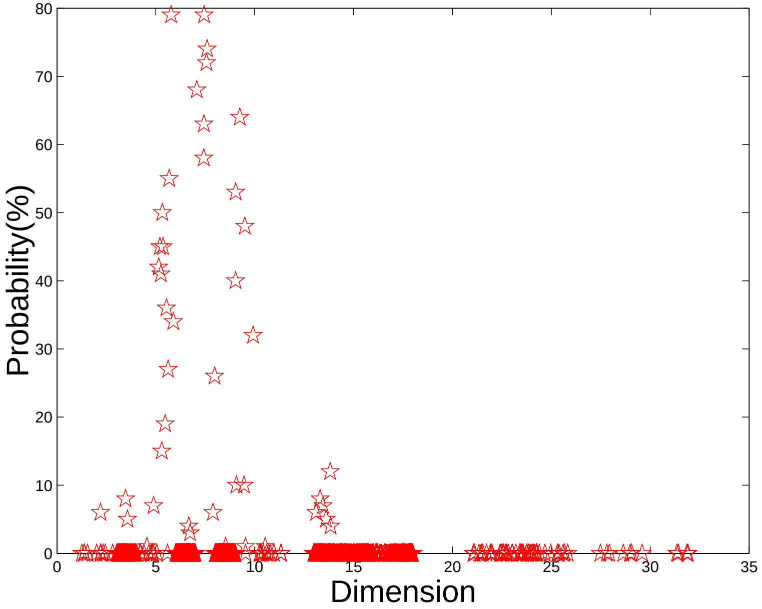
- # of positive Lyapunov exponents p

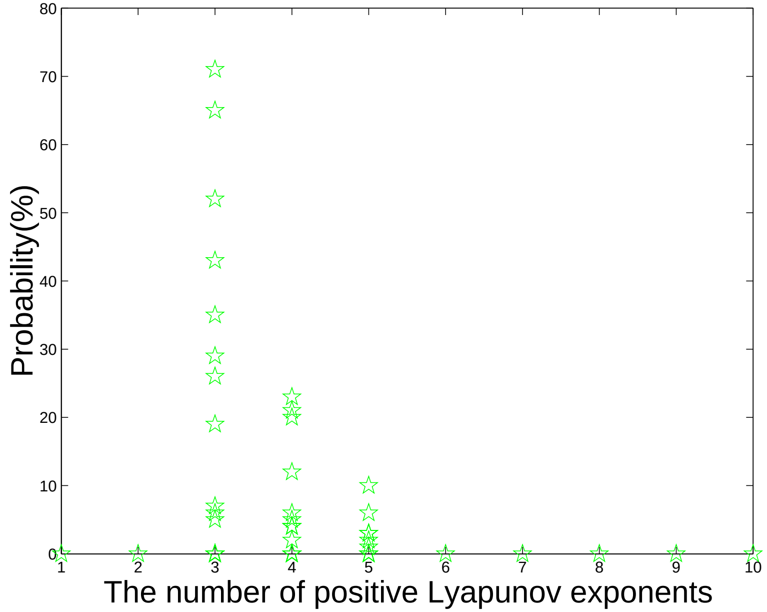
$$\lambda_1 > \lambda_2 > \cdots \lambda_p > 0 > \lambda_{p+1} > \cdots \lambda_{N^2}$$

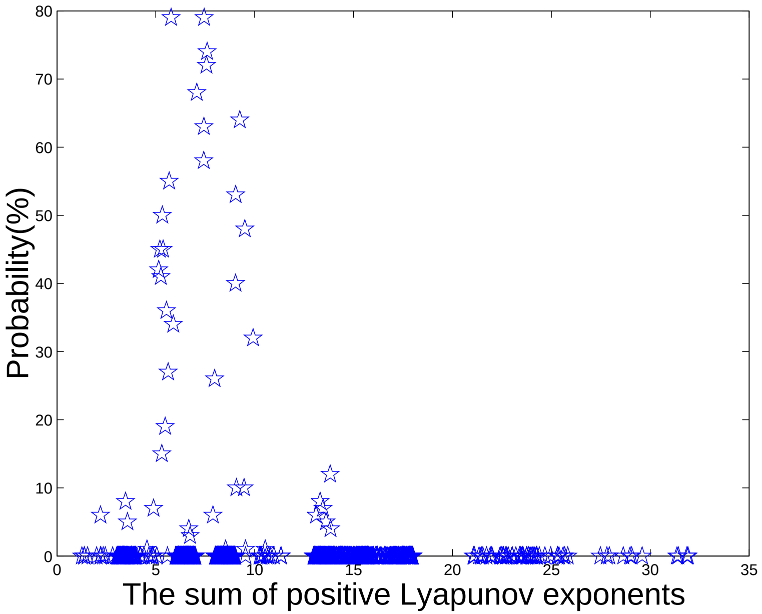
- Sum of positive Lyapunov exponents S

$$S = \sum_{i=1}^{p} \lambda_i$$

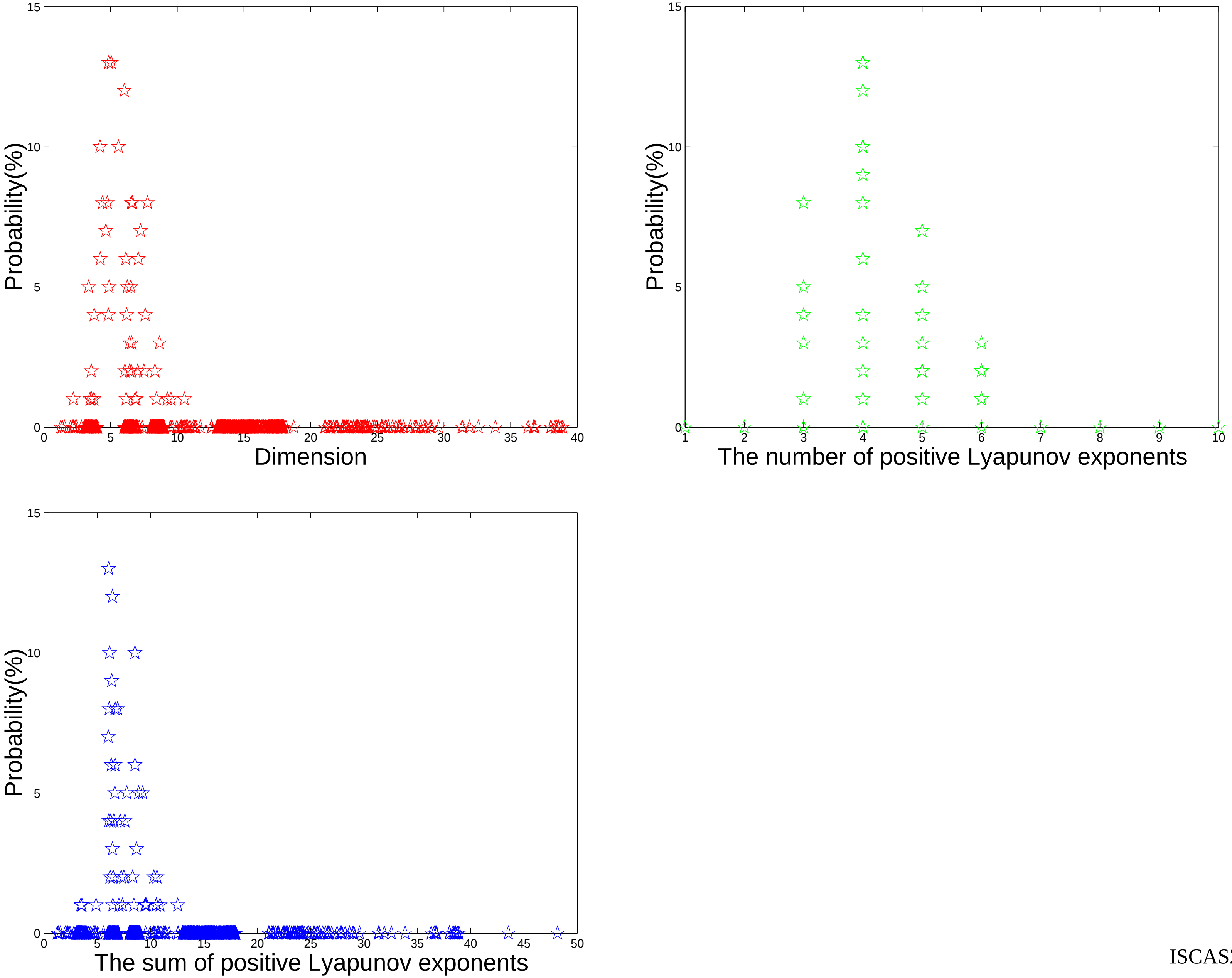
Results for Nug12 (578)







Results for Nug16 (1240)



How to find a good parameter set?

How to find a good parameter set?

1. Robust application for

   | different type problems |, | larger size problems |

How to find a good parameter set?

1. Robust application for

different type problems, larger size problems

↓

How to find a good parameter set?

1. Robust application for

different type problems, larger size problems

↓

2 Firing rates F_R of CNN that solves an N-size problem

How to find a good parameter set?

1. Robust application for

different type problems , larger size problems

↓

2 Firing rates F_R of CNN that solves an N-size problem

- $F_B = \dfrac{1}{N} = \dfrac{N}{N^2}$

How to find a good parameter set?

1. Robust application for

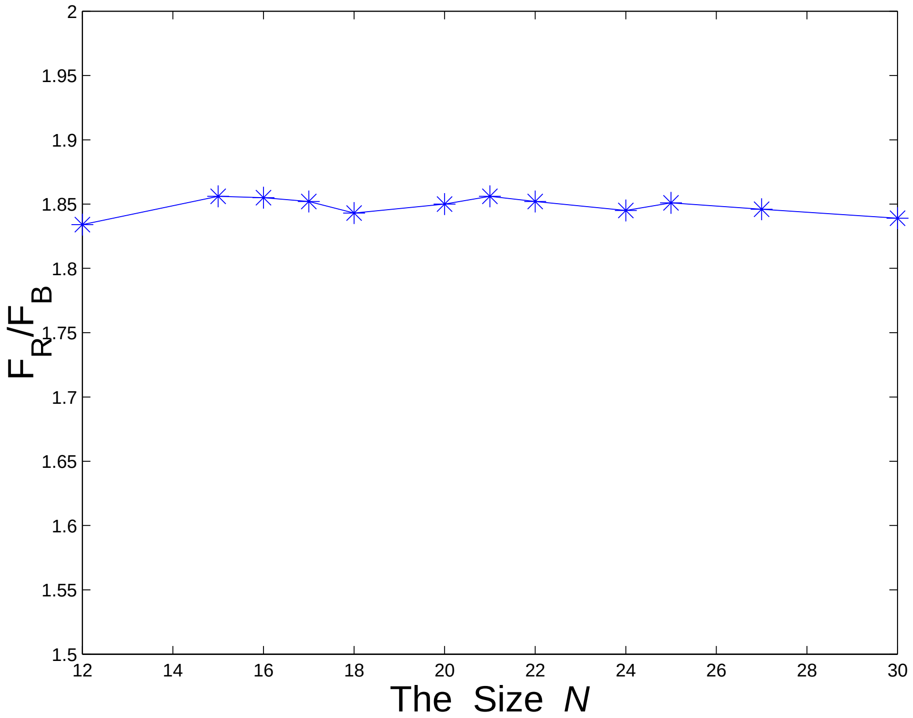
   different type problems, larger size problems

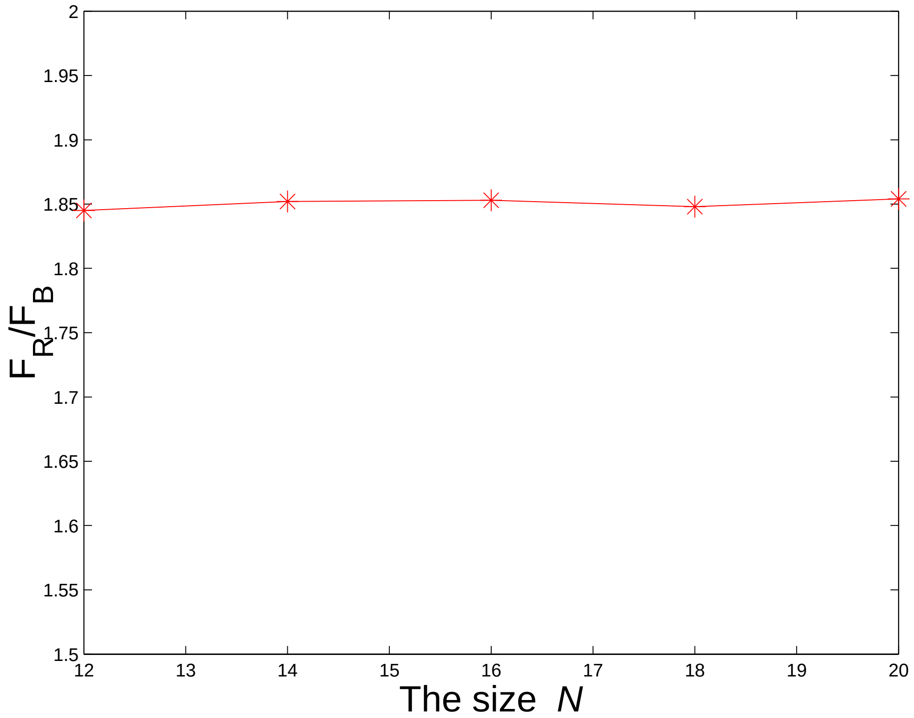
   ↓

2 Firing rates F_R of CNN that solves an N-size problem

- $F_B = \frac{1}{N} = \frac{N}{N^2}$
- the relation between N and $\frac{F_R}{F_B}$

N vs F_R/F_B





Good solutions are obtained when $F_R \approx 1.85 F_B$

Possibility to set good parameter values of CNN by observing firing rates

Conclusions

CNN is applied for solving QAPs.

Conclusions

CNN is applied for solving QAPs.

1. Evaluate chaotic dynamics and solvable performance

$$\text{high performance} \Longleftrightarrow \begin{cases} D_L & \text{small} \\ p = \sum \lambda_i, \lambda_i > 0 & \text{small} \\ S = \sum^{p} \lambda_i & \text{small} \end{cases}$$

Conclusions

CNN is applied for solving QAPs.

1. Evaluate chaotic dynamics and solvable performance

$$\text{high performance} \iff \begin{cases} D_L & \text{small} \\ p = \sum \lambda_i, \lambda_i > 0 & \text{small} \\ S = \sum^{p} \lambda_i & \text{small} \end{cases}$$

$\longrightarrow$ "Shy" search $\Leftrightarrow$ CNN for Associative memories

Conclusions

CNN is applied for solving QAPs.

1. Evaluate chaotic dynamics and solvable performance

$$\text{high performance} \iff \begin{cases} D_L & \text{small} \\ p = \sum \lambda_i, \lambda_i > 0 & \text{small} \\ S = \sum^{p} \lambda_i & \text{small} \end{cases}$$

$\longrightarrow$ "Shy" search $\Leftrightarrow$ CNN for Associative memories

2. How to set parameters of CNN for robust applications?
   (a) Observe the firing rate F_R
   (b) Set parameters at $F_R \approx 1.85/N$