AQC2021, Poster D5, Day 3 : Thursday, 6/24/2021, JST

FUJITSU

Rejection-free MCMC for QUBO Optimization and Boltzmann Sampling

Aki Dote[1,3], Jeffrey S. Rosenthal[2], Keivan Dabiri[3],
Hirotaka Tamura[4,1], Sigeng Chen[2], and Ali Sheikholeslami[3]

[1]Fujitsu Ltd., Japan
[2]Department of Statistical Sciences, University of Toronto, Canada
[3]Department of Electrical and Computer Engineering, University of Toronto, Canada
[4]DXR Laboratory Inc., Japan

Contents

FUJITSU

1. Overview

FUJITSU

- Rejection-free (RF) methods are old techniques to enhance search efficiency of Markov chain Monte Carlo (MCMC)
- Computational cost of RF is high, but *computational time can be drastically reduced using parallel architectures*, such as FPGA, GPU, and ASICS
- This work provides quasi-RF (Digital Annealer) and exact RF (future-gen DA) *algorithms for parallel architecture*
- And some numerical results
  - Sampling: Correct sampling using RF-Metropolis with multiplicities
  - Optimization: Computational time / insensibility for penalty terms

2. MCMC chain and RF chain

- Markov chain Monte Carlo (MCMC)
  - Generate stochastic sample chains that converge to target distribution
  - Inefficient if chain has many repetition by rejected trials
- Rejection-Free (RF) method [1,2]
  - Generate “chains without rejection” and “number of repetition”
  - Quite efficient if the original chain contains many repetition

reject accept

Original MCMC $\{X_t\} = \{a, a, a, b, b, b, b, b, b, b, c, c, b, b, b, \dots\}$

RF $\{J_s\} = \{a, b, c, b, \dots\}$ “jump chain” (P.9)

$\{M_s\} = \{3,7,2,3, \dots\}$ “multiplicity list” (P.11)

2. Preparation: Energy function and QUBO

FUJITSU

- Quadratic energy function:

$$E(\boldsymbol{x}) = -\frac{1}{2}\sum_{i,j=1}^{N} W_{ij}x_i x_j - \sum_{i=1}^{N} b_i x_i$$

  - $W_{ij} = W_{ji}, W_{ii} = 0, b_i$ are real constants given by the problem
  - $\boldsymbol{x} \in \{0,1\}^N$ are binary variables

- Quadratic Unconstrained Binary Optimization (QUBO)
  - Find minimum of $E(\boldsymbol{x})$ and its minimizer $\boldsymbol{x} \in \{0,1\}^N$
  - Many combinatorial optimization problems can be represented as QUBO

2. Preparation: Target distribution

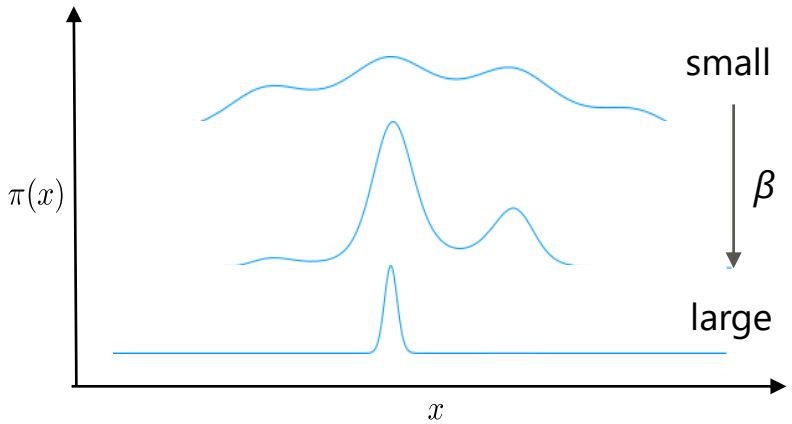
FUJITSU

- Target distribution: canonical distribution

$$\pi(\boldsymbol{x};\beta)=\frac{1}{Z(\beta)}\exp\bigl(-\beta E(\boldsymbol{x})\bigr),\qquad Z(\beta)=\sum_{\boldsymbol{x}\in\{0,1\}^N}\exp\bigl(-\beta E(\boldsymbol{x})\bigr)$$

  - $\beta = 1/T$: inverse temperature, $Z(\beta)$: partition function

- Control β for optimization
  - $\beta\to 0$: $\pi \to 1/2^N$ for all $\boldsymbol{x}$
  - $\beta\to\infty$: π is concentrated around lower $E(\boldsymbol{x})$
  - We may find minimum of $E(\boldsymbol{x})$ and its minimizer with high probability at low β *hopefully*.



FUJITSU

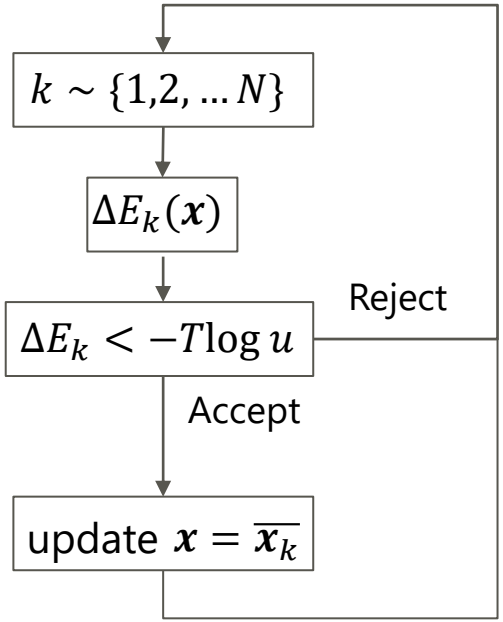
3. Speedup using parallel architecture

- Metropolis Algorithm (not parallel)
- Quasi-RF Algorithm on DA
- Exact RF Algorithm

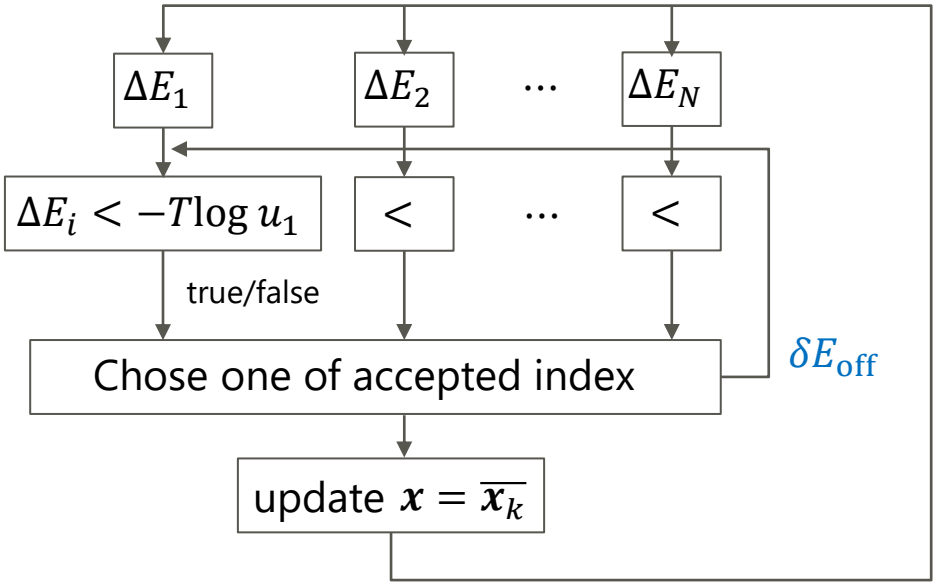
3. Metropolis algorithm

- Set arbitrary initial state $X_0 = \boldsymbol{x}_0 \in \{0,1\}^N$
- Repeat trials below as you like:
  - Propose next bit to flip: $k \sim Q_i(\boldsymbol{x}) = 1/N$
  - If $u < A_k$ $\quad u \sim \mathrm{Unif}(0,1)$
    - Accept flip: $X_{t+1} = \overline{\boldsymbol{x}_k} \equiv (x_1, x_2, \dots, 1 - x_k, \dots x_N)$
    - Else reject flip: $X_{t+1} = \boldsymbol{x}$

      → generates $\{\dots, a, a, a, a, a, \dots\}$
- Metropolis criterion:
  - $A(x'|x) = \min[1, \pi(x')/\pi(x)] = \min(1, e^{-\beta \Delta E_i})$
    - $\Delta E_i(\boldsymbol{x}) = E(\overline{\boldsymbol{x}_i}) - E(\boldsymbol{x})$: energy difference
- Not efficient for parallel architecture



3. Quasi-RF algorithm of Digital Annealer (ASIC)



Parallel trial for all neighbors

Dynamic energy offset for Quasi-RF

If no candidate to flip,
(when $\Delta E_i(x) > 0, \forall i$)
Subtract small $\delta E_{\text{off}} > 0$ from all ΔE_i's until at least one bit is accepted

- Probability of acceptance for 1 trial enlarged up to *N* times [3]
- Computational cost for 1 trial is at least *N* times larger than Metropolis
- Computational *time* can be reduced to 1/*N* Limited by hardware

$+O(\log N)$ for tree reduction

3. Exact RF Metropolis on parallel architecture

FUJITSU

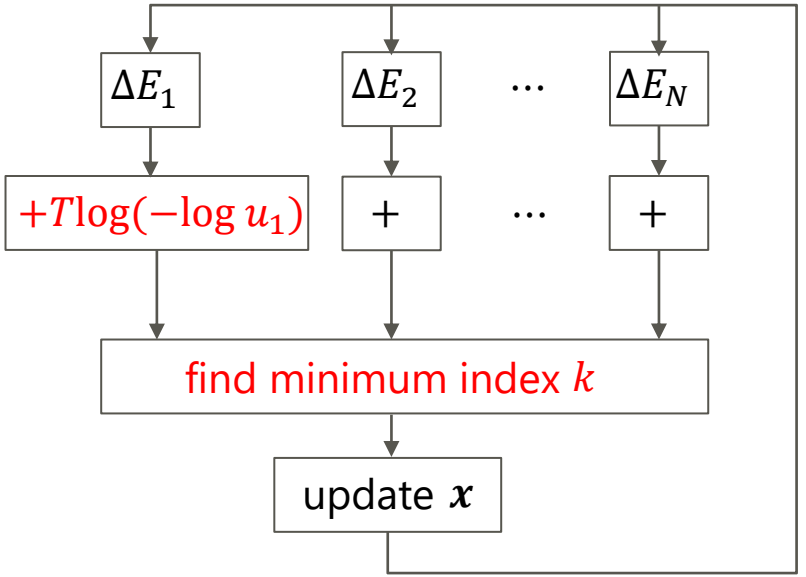
- Algorithm:
  - Select bit to flip according to

$$\tilde{T}_i(x) = \frac{A_i(x)}{\sum_{j=1}^{N} A_j(x)}, \qquad i = 1, \dots N$$

  - is mathematically equivalent [4] to

$$k = \mathrm{argmin}_i\{\max(0, \Delta E_i) + T\log(-\log u_i)\}$$

    - $u_i \in (0,1)$ are iid random number



- Advantage of exact RF algorithm
  - Enable sampling → P.11
  - Optimization: Insensitivity on penalty coefficient → P.14

FUJITSU

4. Results

- Sampling
  - Sampling and expected value using RF
- Combinatorial Optimization
  - Computational time
  - Problem with constraints – insensitivity on penalty coefficient

4.1 Sampling: multiplicity and expected value

FUJITSU

Prob. of escape: $\alpha(x) = \frac{1}{N}\Sigma_{j=1}^{N} A_i(x)$

Prob. of rejection: $1 - \alpha(x)$

Original chain: $\{X_t\} = \{a, a, a, b, b, b, b, b, b, b, c, c, b, b, b, \ldots\}$

RF: **Jump-chain (P.9) +**

Multiplicity: $M_s = t_s \sim (1-\alpha)^{t-1}\alpha$

mean value: $\langle M(x) \rangle = 1/\alpha(x)$

- Expected value of $f(\boldsymbol{x})$ is estimated with non-stochastic $\langle M \rangle = 1/\alpha(x)$ [2]

$$\langle f \rangle = \lim_{n\to\infty} \frac{\sum_{s=1}^{n} t_s f(\boldsymbol{x}_s)}{\sum_{s=1}^{n} t_s} = \lim_{n\to\infty} \frac{\sum_{s=1}^{n} f(\boldsymbol{x}_s)/\alpha(\boldsymbol{x}_s)}{\sum_{s=1}^{n} 1/\alpha(\boldsymbol{x}_s)}$$

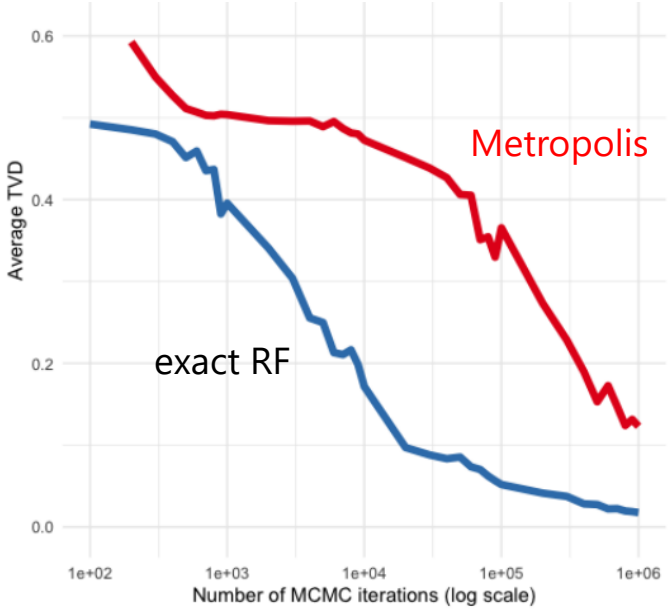
stochastic value

fixed value at each $\boldsymbol{x}$

4.1 Sampling: 4x4 lattice Ising model

Energy function: $E(\boldsymbol{\sigma}) = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j, \quad \boldsymbol{\sigma} \in \{\pm 1\}^{n^2}$

magnetization: $m = \sum_i^{n^2} \sigma_i$



Total variation distance of magnetization

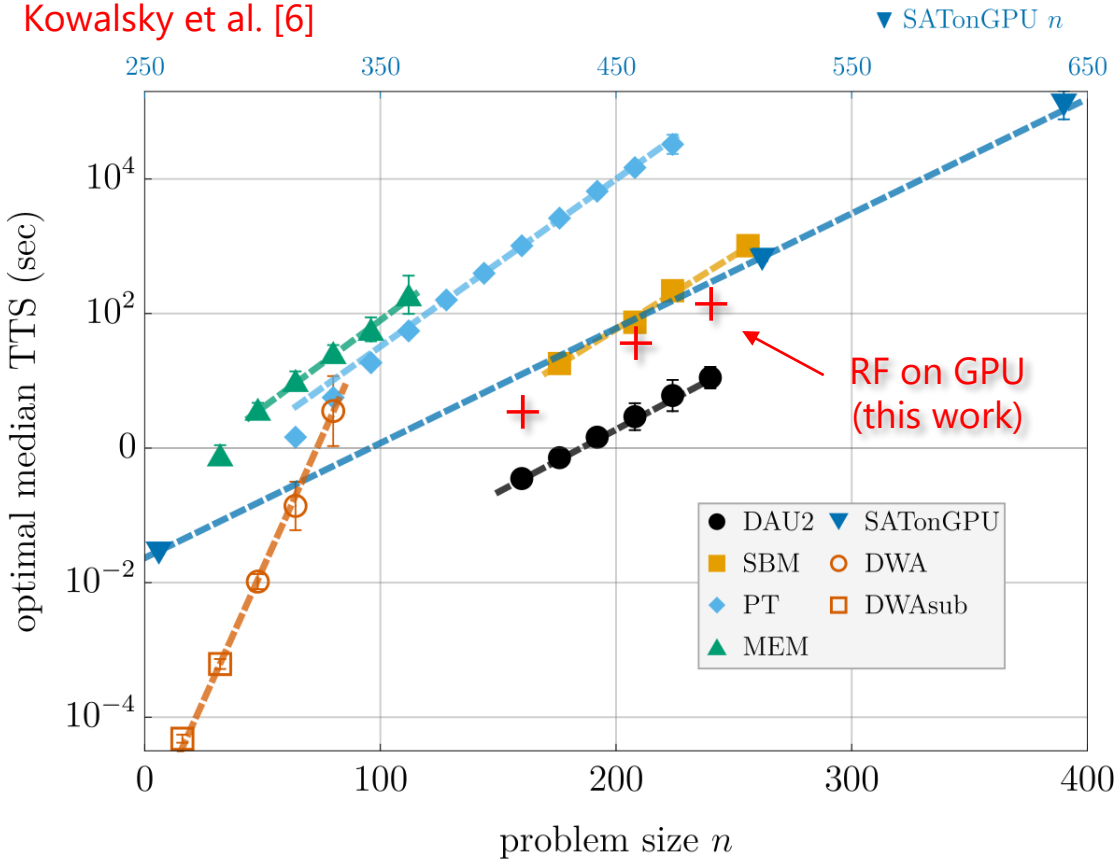
- At low temperature: T=1.0
- Median of 100 runs with different random seed for each algorithm (w/o PT) [2]

■ Exact RF generates correct weighted samples with less iterations

4.2 Optimization time: 3-regular-3-XORSAT

FUJITSU

Kowalsky et al. [6]



- Time to solution (TTS) of RF is competitive to QUBO solvers
  - RF with naïve PT
  - Prototype on 1GPU (RTX2080ti)
  - × 10 slower than DAU2 (ASIC)

[Caution]
- Results other than RF is cited from the paper [6]
- where each data point are median of 100 random instances
- We solves **only one instance** for each size n

4.2 Optimization: Search efficiency with penalty

FUJITSU

QUBO solver

$$E(\boldsymbol{x}) = C(\boldsymbol{x}) + \lambda V(\boldsymbol{x})$$

cost　　penalty term

・$\lambda > 0$ should be enough large to $x_{\min} = \min\{E(x)\}$ satisfy $V(x_{\min}) = 0$

Problem with constraints

minimize: $C(\boldsymbol{x})$
subject to: $V(\boldsymbol{x}) = 0$

・$V(\boldsymbol{x}) \geq 0$
・feasible states: $\boldsymbol{x}^* \in \{\boldsymbol{x} | V(\boldsymbol{x}) = 0\}$

- Optimization speed of RF is not sensitive for λ.
  - Metropolis: $A_i(x^*) = \exp[-\beta(\Delta C_i + \lambda \Delta V_i)]$ decreases as λ grows → slow
  - RF: $\widetilde{T}_i(x^*) = A_i / \Sigma_j A_j$ is invariant for λ
    - In the case of strict and uniform penalty: $\Delta V_i(x^*) = V(\overline{x_i^*}) - V(x^*) = c > 0, \forall i \in N(x^*)$

4.2 Optimization: Quadratic Assignment Problem

FUJITSU

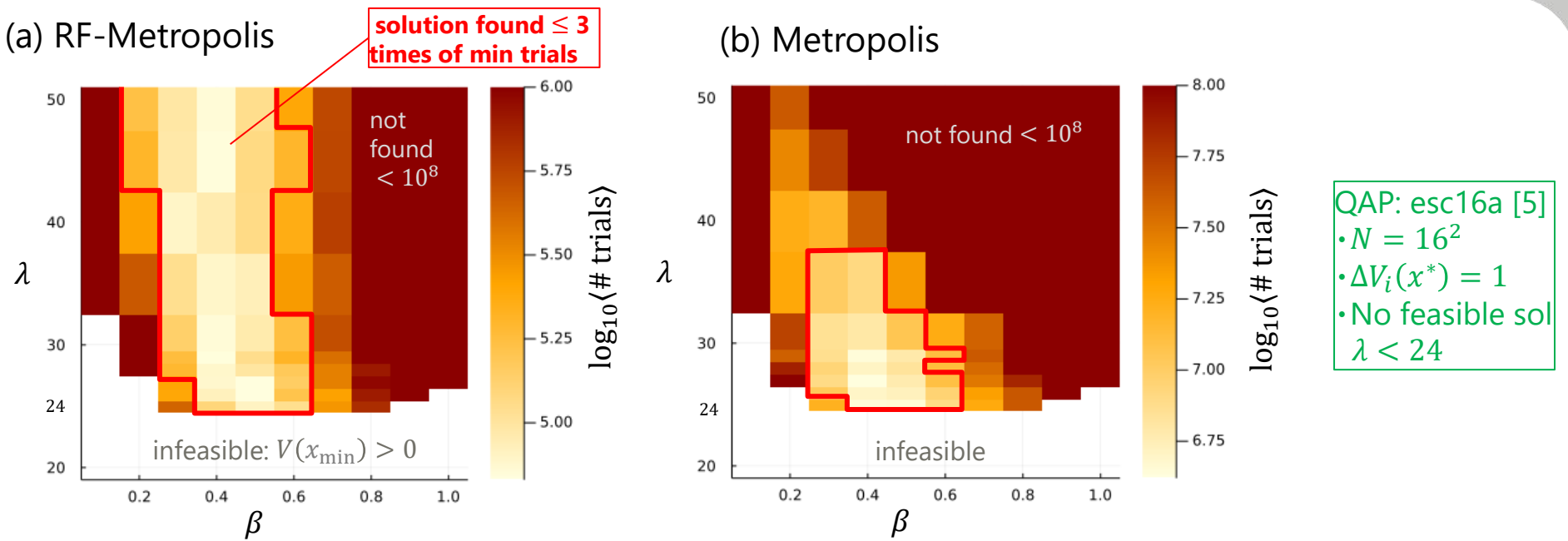


Fig. Number of trials to reach the known best solution at fixed λ and β (median of 100runs)

- RF found known solution faster in wider ranges of λ and β (red area) with 1/100 trials of Metropolis

5. Conclusion

FUJITSU

- Parallel RF algorithm can recover canonical distribution correctly with (expected value of) multiplicities
- RF-Metropolis algorithm is usable for solving combinatorial optimization problems
  - Especially for the problems with strict local minima
- Computational time of RF methods can be reduced drastically using parallel architecture while maintaining its high search efficiency

Thank you!

References

[1] Bortz, A. B., Kalos, M. H., & Lebowitz, J. L. (1975). A new algorithm for Monte Carlo simulation of Ising spin systems. *Journal of Computational Physics*, 17(1), 10-18.

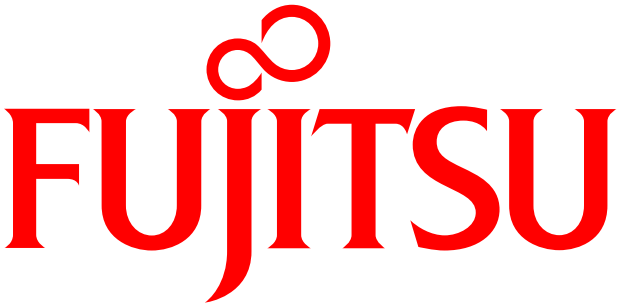
[2] Rosenthal, J. S., Dote, A., Dabiri, K., Tamura, H., Chen, S., & Sheikholeslami, A. (2021). Jump Markov chains and rejection-free Metropolis algorithms. *Computational Statistics*, 1-23.

[3] Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., & Katzgraber, H. G. (2019). Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics*, 7, 48.

[4] Efraimidis, P. S., & Spirakis, P. G. (2006). Weighted random sampling with a reservoir. *Information Processing Letters*, *97*(5), 181-185.

[5] QAPLIB: https://coral.ise.lehigh.edu/data-sets/qaplib/

[6] Kowalsky, M., Albash, T., Hen, I., & Lidar, D. A. (2021). 3-Regular 3-XORSAT Planted Solutions Benchmark of Classical and Quantum Heuristic Optimizers. arXiv preprint arXiv:2103.08464.



shaping tomorrow with you

FUJITSU

Appendix: interesting theoretical results [2]

- RF has its own stationary distribution
- RF is a MCMC that breaks detailed balance condition
- Modification of exchanging probability of Parallel tempering

Appendix: Stational distribution of RF

FUJITSU

- RF has its own Stationary distribution: $\tilde{\pi}(x) = c\alpha(x)\pi(x)$
  - Normalization factor: $c^{-1} = \sum_{x\in\{0,1\}^N} \alpha(x)\pi(x)$

- Satisfy (global) balance condition → converge to $\tilde{\pi}(x)$

$$\sum_{x\in N(x')} \tilde{P}(x'|x)\tilde{\pi}(x) = \tilde{\pi}(x')$$

  - Not satisfy detailed balance condition: $\tilde{P}(x'|x)\tilde{\pi}(x) \neq \tilde{P}(x|x')\tilde{\pi}(x')$

- Jump-chain is Markov chain
  - Irreducibility and aperiodicity is satisfied if original MCMC holds them

Appendix: Swap probability of PT with RF

FUJITSU

- Parallel Tempering(PT)
  - Run independent simulations at different temperature
  - Swap temperatures between runs with probability

$$P_{\mathrm{swap}}(x, \beta; x', \beta') = \min\left(1, \frac{\pi(x; \beta')\pi(x'; \beta)}{\pi(x; \beta)\pi(x'; \beta')}\right)$$

- PT with RF methods
  - Swap probability should be modified with $\tilde{\pi}$ as:

$$\tilde{P}_{\mathrm{swap}}(x, \beta; x', \beta') = \min\left(1, \frac{\tilde{\pi}(x; \beta')\tilde{\pi}(x'; \beta)}{\tilde{\pi}(x; \beta)\tilde{\pi}(x'; \beta')}\right) = \min\left(1, \boxed{\frac{\alpha(x; \beta')\alpha(x'; \beta)}{\alpha(x; \beta)\alpha(x'; \beta')}} \frac{\pi(x; \beta')\pi(x'; \beta)}{\pi(x; \beta)\pi(x'; \beta')}\right)$$

  - Naïve application of PT causes large bias for small system [2]