Simulated bifurcation machines: combinatorial optimization accelerators based on a quantum-inspired parallelizable algorithm

Kosuke Tatsumura
Toshiba Corporation
Outline

• Introduction

• Simulated bifurcation (SB)

• Implementation & Performance

• Application

• Conclusion
Combinatorial optimization

Economically important but computationally hard

Practical problems
- decision-making, planning, etc
- Finance
- Manufacture
- Medicine
- Logistics
- Management
- Material

Combinatorial optimization
- Routing
- Allocation
- Placement
- Scheduling

Combination explosion
Problem size

Nondeterministic polynomial time (NP)-hard
Ising machine
Special-purpose computer for combinatorial optimization

Ising problem

search for the lowest-$E$ state of Ising models

$$E = - \sum_{ij} j_{ij} s_i s_j + \sum h_i s_i$$

Spin configuration, $S$

Combinatorial optimization

Ising machines

- Quantum annealer*1
- CMOS annealer*2
- Digital annealer*3

Optical Ising machines*4,5

Memristor HNN*6

+ Simulated bifurcation machine (2019)

*1 https://www.dwavesys.com/d-wave-two-system
*2 https://www.hitachi.co.jp/New/cnews/month/2019/02/0219.html

Spin: binary variable

Spin configuration, $S$
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Implementation</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantum-inspired</td>
<td>High performance</td>
<td>Very practical</td>
</tr>
<tr>
<td>Quantum bifurcation machine in a quantum principle</td>
<td>Scalable</td>
<td>edge/embedded</td>
</tr>
<tr>
<td>Simulated bifurcation algorithm in a new classical principle</td>
<td>multi-chip</td>
<td>cloud</td>
</tr>
<tr>
<td>Highly parallelizable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2023 Symposium on VLSI Technology and Circuits
Outline

• Introduction

• Simulated bifurcation (SB)

• Implementation & Performance

• Application

• Conclusion
Quantum-inspired algorithm

Quantum Bifurcation (QB) machine


Hamiltonian describing adiabatic bifurcation process in a nonlinear oscillator network

\[ H_q(t) = \hbar \sum_{i=1}^{N} \left[ \frac{K}{2} a_i^2 a_i^2 - \frac{p(t)}{2} (a_i^2 + a_i^2) + \Delta (a_i^2 a_i^2) \right] - \hbar \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} a_i^2 a_j \]

Classical Bifurcation (CB) machine

classicization of state variables

\[ H_c(x, y, t) = \frac{N}{4} \sum_{i=1}^{N} \left[ K (x_i^2 + y_i^2) - \frac{p(t)}{2} (x_i^2 - y_i^2) + \Delta (x_i^2 + y_i^2) \right] - \frac{\xi_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij} x_i x_j + y_i y_j \]

Simulated Bifurcation (SB) algorithm (2019)


```
Classicizing QB that works in a quantum principle...?
Why SB works? What principle? There was a discovery
```

2023 Symposium on VLSI Technology and Circuits
Simulated bifurcation: Why it works

New classical principle: adiabatic and ergodic search

Dynamical change of energy landscape

- A single local minimum
- Bifurcation (adiabatic process)
- Multiple local minima (target cost function)

Best solution (-1, -1)

**adiabatic search**
- Chase one of the minima

**ergodic search**
- Find a better one with a higher probability

Energy landscape ($N_{\text{spin}}=2$)

Multiple minima in the energetically allowable region

2023 Symposium on VLSI Technology and Circuits
Simulated bifurcation: How it works

“N-body”-type algorithmic structure → highly parallelizable

Example: $N_{\text{spin}} = 4000$

- Many-body interaction
  $\Delta p_i = \sum j_{ij} x_i$

- Matrix-vector multiplication (MM)

- Time evolution (TE)
  $(x, p_i) \leftarrow \text{TE}(x, p_i, \Delta p_i)$

- TE pipeline

Joint force

Matrix-vector multiplication (MM)

Positions of spin variables, $x_i$

Ising Energy

SB time step [count]

SB step iteration

Better solution

Positions of spin variables, $x_i$

SB time step [count]
### Characteristics

<table>
<thead>
<tr>
<th></th>
<th>SA</th>
<th>SB</th>
<th>R-NN</th>
<th>N-body</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structure</strong></td>
<td>simulated annealing</td>
<td>simulated bifurcation</td>
<td>recurrent neural network</td>
<td>gravitational (Coulomb)-force</td>
</tr>
<tr>
<td></td>
<td>Sequential updating</td>
<td>Parallel updating</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><img src="image1" alt="Sequential updating diagram" /></td>
<td><img src="image2" alt="Parallel updating diagram" /></td>
<td></td>
<td><img src="image3" alt="N-body diagram" /></td>
</tr>
<tr>
<td><strong>Parallelism</strong></td>
<td>$O(N)$</td>
<td>$O(N^2)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- More parallelizable
- Intensive memory access (J/W matrix (NxN matrix))
- Very similar
- More PEs per chip

**SB can be accelerated by FPGAs/GPUs (not limited to special ASICs)**

**Many AI chips (AI ASSPs) are beneficial also to SB**

---

2023 Symposium on VLSI Technology and Circuits
Outline

• Introduction
• Simulated bifurcation (SB)
• Implementation & Performance
• Application
• Conclusion
FPGA-based accelerator for simulated bifurcation

Large-scale, massively parallel, and high utilization

[K. Tatsumura et al., IEEE FPL, (2019)]

<table>
<thead>
<tr>
<th>Problem</th>
<th>complete-graph MAX-CUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine size</td>
<td>4,096 spins (on Arria10 FPGA)</td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
</tr>
<tr>
<td>Pr/Pc/Pb</td>
<td>32/32/8</td>
</tr>
<tr>
<td># of MAC PEs</td>
<td>8,192</td>
</tr>
<tr>
<td>Effective activity</td>
<td>92%</td>
</tr>
<tr>
<td>Resource</td>
<td></td>
</tr>
<tr>
<td>ALM</td>
<td>40%</td>
</tr>
<tr>
<td>BRAM</td>
<td>56%</td>
</tr>
<tr>
<td>DSP</td>
<td>7%</td>
</tr>
<tr>
<td>System Clock</td>
<td>[MHz]</td>
</tr>
<tr>
<td>Fsys</td>
<td>269</td>
</tr>
</tbody>
</table>

Spatial parallelization

Temporal parallelization

#PEs > N (not achievable for SA)
Performance (2019)

Coherent Ising Machine
800 GMAC/s @ 1000 W

FPGA-SB
1,873 GMAC/s @ 49 W
(288X more energy efficient)

@all-to-all-connected
2000-spin MAX-CUT

[T. Inagaki, Science 354, 603, '16]

Performance (2021)

SB is very competitive with state-of-the-art Ising machines.

SB is very competitive with state-of-the-art Ising machines.

2nd-gen algorithm

Quasi-quantum tunneling

10X faster than 1st-gen

Competitors

SB: Simulated bifurcation
QA: Quantum annealer
CIM: Coherent Ising machine
DA: Digital annealer
SimCIM: Simulated CIM
RBM: Restricted Boltzmann machine
MA: Momentum annealing

20200-spin all-to-all-connected problem

TTS (Time-To-Solution)
Scalability (2021)

Scaling out Ising machines with full spin-to-spin connectivity

Scale-up

Locally-connected spin network

Chip 1

Partition

Chip 2

Fully-connected spin network

Our aim

Scale-out

computing chip

communication cable

Chip 1

Partition

Chip 2

[K. Tatsumura et al., Nat. Ele., (2021)]
Scalability (2021)

Multi-chip architecture based on partitioned SB

Bidirectional ring-network cluster without any centralized features

Autonomous synchronization mechanism (No clock-sharing, No central-HUB)

All chips are autonomous, homogeneous and symmetric

[K. Tatsumura et al., Nat. Ele., (2021)]
Scalability (2021)

**Strong scaling**
Increase $P_{\text{chip}}$ at a fixed problem size ($N$)

**Weak scaling**
Increase $P_{\text{chip}}$ and $N$ in the same proportion

**Computation-bound**

\[
\frac{T_{\text{computation}}}{T_{\text{communication}}} > 1
\]

Throughput enhancement to the vicinity of an ideal upper limit determined by the communication tech.

Constant-efficiency scaling at the maximized computation parallelism (at the strong scaling limit)

Throughput 

\[
\begin{align*}
\text{Throughput} & = T_{\text{computation}} \\
N & = 16,384 \\
P_{\text{chip}} & = 8 \\
\text{Throughput} & = 10,870 \text{ GMAC/s} \\
\end{align*}
\]

\[
\begin{align*}
\text{Throughput} & = T_{\text{communication}} \\
N & = 131,072 \\
P_{\text{chip}} & = 64 \\
\text{Throughput} & = 89,319 \text{ GMAC/s} \\
\end{align*}
\]

Experiment

Simulator
Outline

• Introduction

• Simulated bifurcation (SB)

• Implementation & Performance

• Application

• Conclusion
Application of SBMs

edge/embedded

Lower latency

FPGA cluster

\[
\begin{bmatrix}
\Delta p_1 & \Delta p_2 \\
\end{bmatrix}
= 
\begin{bmatrix}
J_{11} & J_{12} \\
J_{21} & J_{22}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\]

Single-shot processing

High-speed real-time systems

Financial trading

Autonomous control

cloud

Larger problem

GPU cluster

\[
\begin{bmatrix}
\Delta p_1, \Delta p_1 \\
\Delta p_2, \Delta p_2
\end{bmatrix}
= 
\begin{bmatrix}
J_{11} & J_{12} \\
J_{21} & J_{22}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\]

Batch processing

Addressing complex/global issues

Drug design

Planning

Toshiba’s website “SQBM⁺™”
Enabling NP-hard optimization in real-time systems

Must respond within critically defined time constraints
→ Enabling rational judgment based on combinatorial optimization

Financial trading system

<table>
<thead>
<tr>
<th>Stock</th>
<th>ASK</th>
<th>Price</th>
<th>BID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock A</td>
<td>1,000</td>
<td>4,250</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4,249</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4,248</td>
<td>1,200</td>
<td></td>
</tr>
</tbody>
</table>

Stock B

<table>
<thead>
<tr>
<th>ASK</th>
<th>Price</th>
<th>BID</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>6,381</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6,380</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6,379</td>
<td>500</td>
</tr>
</tbody>
</table>

Trading system A

System B

BBO changed

Autonomous control

Sensing → Understanding → Judging/Planning → Control

Less than 100 ms latency (periodic operation)

FPGA-based SBMs

Ultralow latency (sub-msec)

Deterministic latency

1. Embeddable

2. Custom I/F

3. Custom circuit  (No software interrupt)
Trading system for cross-currency arbitrage

Optimal path search in a directed graph (a typical combinatorial problem)

**Market Graph**

- Currency, $i$
- Exchange rate, $r_{ij}$

**Arbitrage Problem**

Find a closed path that maximizes the profit

**Cost function**

$$ \text{Profit} = \prod_{i,j \in \text{path}} r_{ij} $$

**Constraint**

Must be a closed path

**Ising (QUBO) formulation**

$$ C' = \prod r_{i,j} b_{i,j} $$

$$ w_{i,j} = -\log r_{i,j} $$

$$ C = \sum w_{i,j} b_{i,j} $$

$$ P = \sum_{i} \sum_{j \neq i'} b_{i,j} b_{i,j'} + \sum_{j} \sum_{i \neq i'} b_{i,j} b_{i',j} + \sum_{i} \left( \sum_{j} b_{i,j} - \sum_{j} b_{j,i} \right)^2 + \sum_{i,j} b_{i,j} b_{j,i} $$

[K. Tatsumura et al., IEEE ISCAS., (2020)]
**Trading system for cross-currency arbitrage**

An end-to-end FPGA-based arbitrage system

---

**Arbitrage system**

1. **Custom I/F (feed handler)** captures market feeds at unscheduled intervals
2. **Exchange rate manager** updates an NxN wij matrix, outputs all weights in a single clock
3. **SB accelerator** searches for an optimal path from all possible paths
4. **Trading engine** prepares order packets
5. **Custom I/F (line handler)** issues the order packets

---

**Market packet**

**Order packet**

**FPGA**

**10Gb Ethernet cable**

---

[K. Tatsumura et al., IEEE ISCAS, (2020)]
Trading system for cross-currency arbitrage

<30 μs system-wide latency & 91% Top-1 probability

Exchange rates on Jan. 2nd, 2019

Profit rates for arbitrage paths

Solution accuracy

System-wide response time:
27.5us (on average over 1000 packets)
Conclusion

Simulated bifurcation (SB):
quantum-inspired, highly-parallelizable algorithm for combinatorial optimization

Simulated bifurcation machines (SBM, HW implementation):
efficiently implemented with FPGAs/GPUs, very practical (no refrigerator, no laser)
high performance, very competitive with state-of-the-art Ising machines
embeddable, customizable (FPGA), scalable (FPGA cluster, GPU cluster)
prefers memory-rich architectures, affinity to AI chips

Innovative applications:
Edge(FPGA):
real-time systems that make a rational judgment based on combinatorial optimization
Cloud(GPU):
enabling large/complex combinatorial optimization that was previously impossible
References


