Engineering Discrete Simulated Bifurcation for an FPGA Digital Ising Machine

Engineering Discrete Simulated Bifurcation for an FPGA Digital Ising Machine

Fabrizio Orlando*, Deborah Volpe*, Giacomo Orlandi*,
Fabrizio Riente*, Marco Vacca* and Mariagrazia Graziano†
*Department of Electronics and Telecommunications, Politecnico di Torino Italy
†Department of Applied Science and Technology, Politecnico di Torino Italy
fabrizio.orando@studenti.polito.it, deborah.volpe@polito.it, giacomo.orlandi@polito.it,
fabrizio.riente@polito.it, marco.vacca@polito.it and mariagrazia.graziano@polito.it

***Abstract*—Simulated Adiabatic Bifurcation (aSB) is a quantum-inspired algorithm that provides approximate solutions for large-scale optimization problems using the Ising model. It emulates the quantum adiabatic evolution of a network of non-linear Kerr oscillators on classical platforms. These oscillators undergo bifurcation, where each branch corresponds to a spin state, with the network creating an energy imbalance to determine the optimal solution.**

This approach is highly parallelizable, making it suitable for implementation on GPUs and FPGAs. However, classical emulation introduces analog errors, potentially compromising performance. To address this, alternative methods like ballistic (bSB) and discrete (dSB) evolutions were developed, with the ballistic approach further enhanced by thermal fluctuations (HbSB).

Our comprehensive analysis of bSB, dSB, and HbSB using benchmark Max-Cut and knapsack problems aimed to identify the best balance of speed, area, and accuracy. We found that fixed-point number representation allows more algorithm iterations within the same timeframe as floating-point representation. The proposed FPGA architecture implements dSB and its heated version, leveraging high parallelizability and efficient memory organization. This design supports larger coefficients and improves problem-solving efficiency without multipliers for matrix-vector evaluation.

Challenges include mitigating data dependency between matrix-vector multiplication and time evolution, and developing a preconditioning method for adapting problem coefficients, making the architecture suitable for real-world applications and potential ASIC release.

***Index Terms*—Ising Machines, Simulated Bifurcation, Quadratic Unconstrained Binary Optimization, quantum-inspired optimization, Ising**

I. Introduction

Simulated Adiabatic Bifurcation (aSB), a quantum-inspired algorithm introduced in [1], offers approximate solutions of large-size optimization problems written according to Ising formulation, which is equivalent to Quadratic Unconstrained Binary Optimization (QUBO) model. It emulates on classical platforms the quantum adiabatic evolution of a non-linear-Kerr-oscillators network. These oscillators exhibit a bifurcation during their evolution, and each branch can be associated with a spin state. The problem is encoded by associating an oscillator with each spin variable, and the spin interactions are expressed through the network, whose role is to create an imbalance in the energy of the systems such that the optimum of the problem corresponds to the final ground state, forcing each oscillator to choose the branch representing the spins state of the problem solution (Figure 1). A key advantage of this approach is the high level of parallelizability in simulating the system evolution, which promotes its hardware implementations such as on Graphics Processing Unit (GPU) and Field Programmable Gate Array (FPGA).

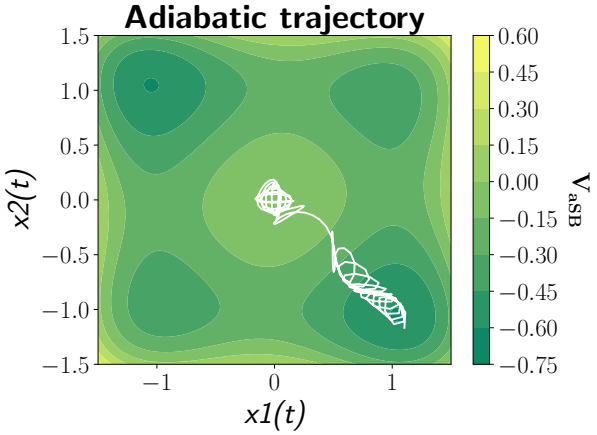

Fig. 1: Simulated Adiabatic Bifurcation (aSB) evolution for a 2-spin max-cut problem starting from the origin. The trajectories of the two oscillators' network over time are illustrated by the white lines. The potential energies at the final time are also depicted (V_{aSB}, V_{bSB} and V_{dSB}).

However, the mathematical model employed for emulating the adiabatic evolution of the system classically generates some analogue errors, potentially compromising performance. In response, alternative approaches like the ballistic (bSB) and discrete (dSB) evolution of the network were introduced in [2]. The bSB can be further enhanced by introducing a thermal fluctuation term (HbSB) for escaping from local minima [3].

II. Methodology and Proposed Implementation

We conducted a comprehensive analysis of bSB, dSB, and HbSB through proper software models, focusing on benchmark maxcut and knapsack problems. The analysis aimed to identify the algorithm guaranteeing the best compromise among speed, area and results accuracy for proposing a new digital architecture for FPGA and to achieve the best number

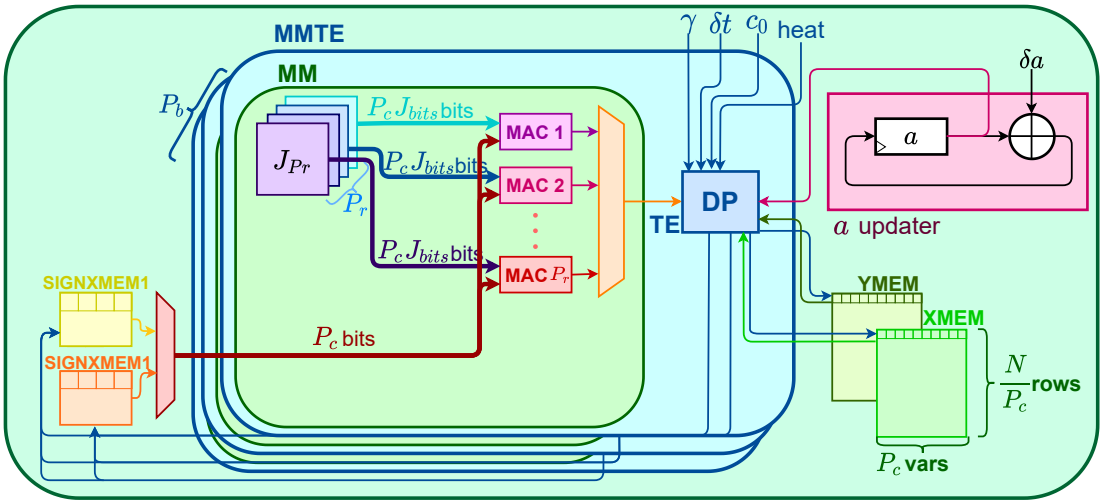

Fig. 2: High-level description of the proposed architecture. It is composed of P_b Matrix-vector Multiplication Time Evolution (MMTE) blocks evaluating the oscillator state evolution. Each of them includes a Matrix-vector Multiplication (MM) block, composed of Multiply-ACcumulate (MAC) units, allowing the parallelization of operation by a factor P_r.

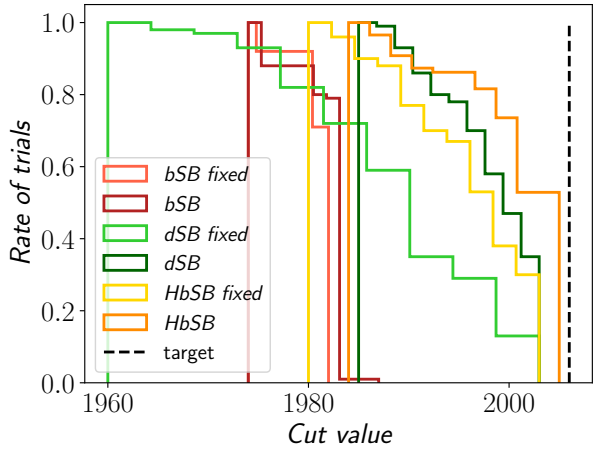

Fig. 3: Results distributions obtained with bSB, dSB, HbSB, considering both floating and fixed point number representation for the 800-spin G7 maxcut problem of the GSet.

representation. Moreover, we also analyzed the potential of introducing thermal fluctuations in dSB. Figure 3 shows the cumulative distribution obtained by considering an 800-node maxcut problem, showing that the fixed-point number representation has minimal impact on the accuracy of the outcomes, while it allows the execution of more algorithm iterations within the same time interval than its floating point number counterpart. The resulting architecture, shown in Figure 2, implements dSB, including its heated version with user-defined external parameters. The architecture structure is inspired by the ones proposed in [4] for aSB. Implementing dSB leads to several advantages that are exploited in the proposed design. In particular, with an equivalent FPGA memory capacity, the proposed architecture allows a higher level of parallelizability since, by discretizing the oscillators' state position variable, dedicated memories can concurrently provide multiple values for matrix-vector multiplication, effectively enhancing parallel processing capabilities. Moreover, it allows the resolution of problems involving larger coefficients, thanks to an efficient organization of the memory resources. For example, without involving multipliers for matrix-vector evaluation replaced with shift operations, our architecture supports 8 bits for the problem coefficients instead of the single bit of the original one. Furthermore, to enhance its applicability in real-world scenarios, the architecture efficiently manages problem coefficients associated with single-spin interactions.

III. Conclusions and Future Perspectives

Challenges for algorithmic efficiency include mitigating data dependency between Matrix-vector multiplication and Time Evolution, for example, performing some approximation in the motion equations, enabling further parallelization. Moreover, to broaden its exploitability in a real-world scenario, a preconditioning method for adapting the problem coefficients to the architecture numeric representation and choosing the algorithm parameters is needed, making possible an Application-Specific Integrated Circuit (ASIC) release.

Acknowledgements

This work was supported in part by AMD under the AMD University Program AMD University program

References

[1] H. Goto, K. Tatsumura, and A. R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems," *Science advances*, vol. 5, no. 4, p. eaav2372, 2019. https://doi.org/10.1126/sciadv.aav2372.

[2] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, "High-performance combinatorial optimization based on classical mechanics," *Science Advances*, vol. 7, no. 6, p. eabe7953, 2021. https://doi.org/10.1126/sciadv.abe79.

[3] T. Kanao and H. Goto, "Simulated bifurcation assisted by thermal fluctuation," *Communications Physics*, vol. 5, no. 1, p. 153, 2022. https://doi.org/10.1038/s42005-022-00929-9.

[4] K. Tatsumura, A. R. Dixon, and H. Goto, "Fpga-based simulated bifurcation machine," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 59–66, IEEE, 2019. https://doi.org/10.1109/FPL.2019.00019.