Towards Tensor-Network SAT-Solvers for Quantum-Classical Workflows

Benjamin Zec*, Lukas Schmidbauer*, Maja Franz*, Wolfgang Mauerer*†
*Technical University of Applied Sciences Regensburg, Regensburg, Germany
† Siemens AG, Foundational Technology, Munich, Germany
{benjamin.zec, lukas.schmidbauer, maja.franz, wolfgang.mauerer}@othr.de

***Abstract*—Integrated HPC/QC systems aim to combine classical high-performance computing with quantum processors, but cannot be reduced to mechanisms for dispatching quantum kernels. An integrated architecture must support aspects such as observability, which cannot be implemented using QPUs alone, as well as fallback execution and cost-aware decisions on whether to replace quantum tasks with classical surrogates. Such mechanisms must be approximate or benefit from problem structure to soften the inescapable exponential classical worst-case complexity. In this work, we study tensor-network ground-state search, as such a surrogate, for optimisation problems. This combines key quantum primitives with advanced classical simulation. It provides initial empirical indicators for surrogate selection criteria, and exposes end-to-end toolchain effects that may be missed when transformation steps are studied in isolation. We compare a native polynomial unconstrained optimisation (PUBO)-to-higher-order-Ising and a quadratised quadratic unconstrained binary optimization (QUBO)-to-quadratic-Ising formulation for Max-3-SAT. Both are encoded as matrix product operator (MPO) and optimised using density matrix renormalisation group (DMRG) approaches, with simulated annealing (SA) as classical performance baseline. Our results show that quadratisation is not a neutral transformation step: auxiliary variables and pairwise couplings substantially degrade solution quality relative to the native higher-order representation, while SA matches or outperforms DMRG across all tested instances. Since the optima of Boolean satisfiability (SAT)-derived problems are classical product states, DMRG's advantages don't materialise here. These findings suggest that surrogate selection in HPC/QC runtimes must be encoding- and instance-aware and provide empirical groundwork for informed decisions on fallback strategies and architecture co-design.**

***Index Terms*—Quantum Computing, Tensor Networks, Optimization, Simulated Annealing, QC/HPC integration**

I. Introduction

While an efficient classical simulation of future application-scale quantum computers is intrinsically impossible, numerical simulation of certain aspects nonetheless plays a crucial role for tasks such as benchmarking [1], designing error-correction mechanisms [2], sub-problems task offloading [3–5], and especially systems-level integration [3, 6]: The rules of quantum mechanics make it impossible to implement classically trivial properties like observability, which must be implemented based on classical (approximative) surrogates. We use a highly performant, yet adaptable simulation method in a quantum optimisation setting to study the end-to-end integration of such mechanisms into HPC/QC platforms. Tensor networks methods, in particular DMRG, are among the most effective approaches for simulating one-dimensional quantum systems [7]. SAT problems are classically difficult optimisation tasks. By casting SAT as an Ising problem, optimal solutions can be obtained by finding the ground state (GS) of the Hamiltonian. Different transformation paths alter key structural properties such as the number of clauses and variables [8, 9]. Efficiency and convergence of tensor networks depend on such properties, and transformation specifics influence properties of the surrogate mechanism.

We study and benchmark DMRG across different SAT-to-Ising compilation paths and compare results against SA, building on the analysis of Schmidbauer *et al.* [8, 10] to evaluate the effect of encodings, discuss the impact on high-performance computing (HPC)-quantum computing (QC) hybrid workflows.

II. Foundations of Tensor Networks

Our notations follows Schollwöck [7]. A tensor is a multilinear map that generalises matrices to arbitrary rank. Higher-rank tensors arise naturally in quantum mechanics, where the state of an N-qubit system lives in a 2^N-dimensional Hilbert space and can be represented by a rank-N tensor. Tensor networks provide a graphical and computational framework for efficiently representing such high-dimensional tensors by decomposing them into networks of lower-rank tensors connected by shared indices. Consider a quantum state $|\psi\rangle$ of a system with l sites and σ_i local state spaces on sites $i = 1, ..., l$, each carrying local physical dimension d_i. The coefficients $c_{\sigma_1 \ldots \sigma_l}$ of the full state $|\psi\rangle = \sum_{\sigma_1 \ldots \sigma_l} c_{\sigma_1 \ldots \sigma_l} |\sigma_1 \ldots \sigma_l\rangle$ can be decomposed into a product of rank-3 tensors $M_1, \ldots, M_l$, yielding the matrix product state (MPS) representation $|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_l} M_1^{\sigma_1} M_2^{\sigma_2} \ldots M_l^{\sigma_l} |\sigma_1 \sigma_2 \ldots \sigma_l\rangle$, where neighbouring tensors are connected by virtual bond indices of dimension χ, also called *bond dimension*. This parameter limits the maximum representable entanglement, reducing the exponential memory scaling of the full state vector to $\mathcal{O}(ld\chi^2)$ [7].

A. Matrix Product Operator

A MPO is the tensor network representation of an operator acting on a many-body system, directly analogous to an MPS for states . A Hamiltonian $\hat{H}$, expressed as an MPO, reads $\hat{H} = \sum_{\substack{\sigma_1 \ldots \sigma_l \\ \tau_1 \ldots \tau_l}} W_1^{\sigma_1 \tau_1} \cdots W_l^{\sigma_l \tau_l} |\sigma_1 \ldots \sigma_l\rangle\langle\tau_1 \ldots \tau_l|$, where each local tensor W_i has rank 4 with dimensions (d_i, d_i, w_{i-1}, w_i) and w is the MPO bond dimension. The

physical indices τ_i and σ_i correspond to the local d-dimensional input and output state spaces of the operator, respectively.

B. Density Matrix Renormalization Group

DMRG is a variational algorithm that approximates the GS of a Hamiltonian by iteratively optimising each tensor of an MPS to minimize $\langle\psi|\hat{H}|\psi\rangle$. To efficiently evaluate $\langle\psi|\hat{H}|\psi\rangle$, left and right environment tensors are contracted recursively from the canonical-form MPS and stored, reducing the energy expectation to a local contraction at the site being optimized. DMRG optimises each site tensor M_i by performing iterative sweeps over the entire chain. During each sweep, the MPS tensors are updated sequentially using a sparse eigensolver (*e.g.*, Davidson algorithm) to solve the corresponding local eigenvalue problem. We employ the two-site DMRG variant, which jointly optimises pairs of neighbouring tensors.

III. The Benchmark Problem, SAT

The seminal NP-complete SAT problem serves as textbook benchmark for computational complexity. A SAT instance asks whether there exists an assignment of Boolean variables $\{x_1, \ldots, x_n\} = V$ that makes a given formula true. Formulas are commonly expressed in conjunctive normal form (CNF) $\bigwedge_{i=1}^{m} C_i$ such that each of the m clauses C_i is a disjunction of literals (variables or their negations, *e.g.*, $C_1 = x_1 \vee x_2 \vee \neg x_3$). In a k-SAT instance, each clause contains exactly k literals. Max-k-SAT seeks to maximise the number of satisfied clauses. The experiment uses SAT formulated as a *pseudo-boolean function (PBF)* $f : \{0,1\}^n \to \mathbb{R}$, which admits a unique multi-linear polynomial representation $f(x_1, \ldots, x_n) = \sum_{S \subseteq \{1,\ldots,n\}} \alpha_S \prod_{j \in S} x_j$, where each monomial has degree $|S|$ and coefficient $\alpha_S \in \mathbb{R}$. A PUBO problem consists of finding $\vec{x} \in \{0,1\}^n$ that minimises or maximises f of arbitrary degree, whereas QUBO restricts f to a polynomial of degree at most two. Reducing a higher-order PBF to a quadratic one by introducing auxiliary variables $\vec{y}$ that encode the higher-order terms is called quadratisation.

IV. Experimental Setup

To evaluate the impact of different problem-transformation pathways, we implemented an end-to-end execution pipeline. This experimental workflow spans from the generation of uniform Max 3-SAT instances (*i.e.*, counting satisfied clauses) to the final ground-state optimisation, as illustrated in Fig. 1. The experiments were conducted on a node equipped with two AMD EPYC 7662 (256 cores in total) and one TiB RAM.

We provide an extensive reproduction package (link in pdf) that allows for extending our work.

A. SAT-Instance Generation

We construct each clause by randomly sampling $k = 3$ literals from literal pool $L = V \cup \{\neg x : x \in V\}$, where V denotes the total set of variables and $\neg x$ defines the negation of x. This sampling process is repeated m times to yield the total number of clauses. The resulting clause-to-variable ratio

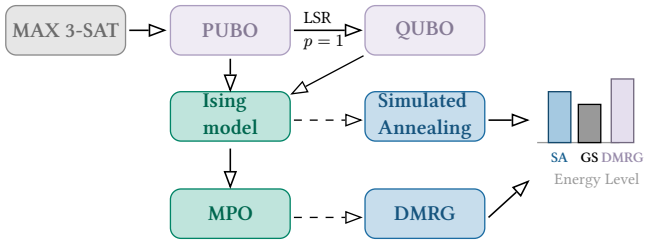

Fig. 1. We benchmark two transformation paths: MAX 3-SAT cast into PUBO, Ising and then MPO (path one) and MAX 3-SAT cast into PUBO, quadratised to QUBO, Ising and then MPO. We compare SA and DMRG ground-state deviations for the Ising and MPO models, respectively.

$\alpha = m/n$ varies across tested configurations. For small n, the number of distinct 3-literal clauses is bounded by $\binom{2n}{3}$, so instances with m approaching this bound yields high clause degeneracy.

B. Max 3-SAT to PUBO Transformation

Consider a Max-k-SAT formula in CNF containing m clauses C_i with $k = 3$ literals each. Using a standard algebraic mapping, encoding positive literals as x_i and negated literals $\neg x_i$ as $(1 - x_i)$. Summing over individual clause functions yields the global objective function for the PUBO problem $f_{\text{SAT}}(\vec{x}) = \sum_{i=1}^{m} f_{C_i}(\vec{x})$. Since $f_{C_i}(\vec{x}) = 1$ if C_i is satisfied and 0 otherwise, $s = f_{\text{SAT}}(\vec{x})$ evaluates to the number of satisfied clauses. Casting Max-3-SAT as maximisation problem where perfect satisfiability yields $s = m$. Expanding 3-SAT polynomials yields non-linear monomials up to degree 3 [8].

C. PUBO to QUBO

To transform a higher-order PUBO f_p into a QUBO f_q, we use the local structure reduction (LSR) algorithm [8], an iterative quadratisation technique governed by a free parameter $p \in [0,1]$ that manages the trade-off between the number of introduced auxiliary variables and the connectivity of the resulting interaction graph. A higher value of p minimises the number of auxiliary variables at the expense of a denser degree-2 interaction matrix in f_q. As each additional introduced variable doubles the search space, we fix $p = 1$ in our automated pipeline used in our experiments.

D. Ising to MPO Transformation

To evaluate the optimisation performance via DMRG, PUBO and QUBO formulations are mapped onto their respective Ising representations and encoded as an MPO. For the quadratised QUBO pipeline, the substitution yields a standard quadratic Ising Hamiltonian consisting of (at maximum) two-body Pauli-Z coupling terms. Conversely, the Hamiltonian in the PUBO pipeline inherits higher-order terms as multi-body spin interactions. Both Hamiltonians are explicitly constructed as MPOs according to Sec. II-A. Auxiliary variables required to reduce higher-order terms to quadratic forms increase the effective problem dimension and introduce additional coupling terms not present in the native formulation [11].

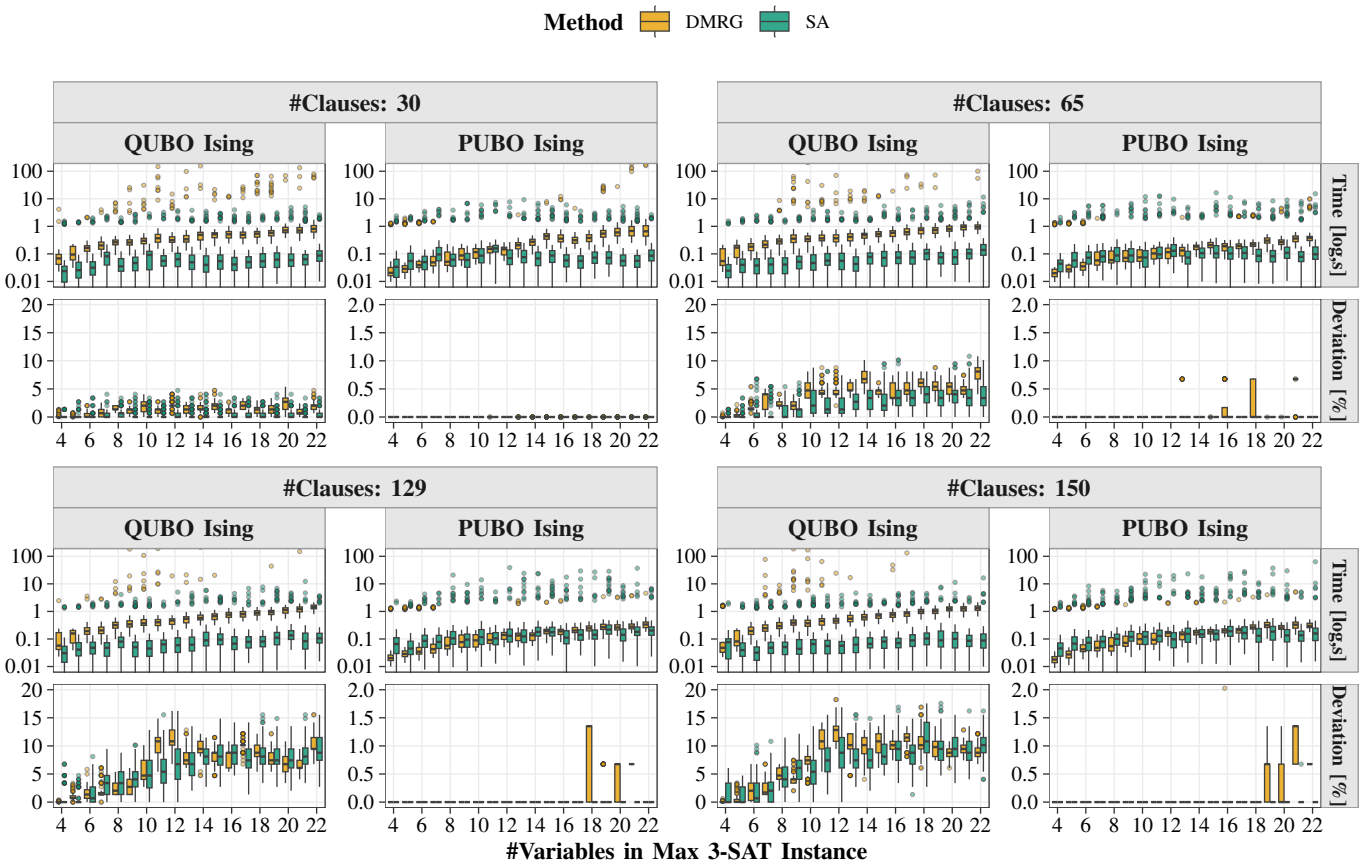

Fig. 2. Optimisation performance of SA (green) and DMRG (yellow), grouped by number of SAT variables n and clause count m. Upper panels show runtime per instance: SA is consistently faster and DMRG scales with n in both encodings, where outliers reflect instances with increased bond dimension χ, driving up contraction cost. Lower panels show deviation from the ground state: the PUBO encoding keeps deviations near zero throughout, while QUBO deviations grow monotonically with n and m and outliers reflect individual instances trapped in local minima due to insufficient bond dimension χ.

E. Solver Configuration and Hyperparameter Benchmarking

To evaluate optimisation performance and eliminate initialisation bias, each Max 3-SAT instance is evaluated over 100 independent trials using randomised hyperparameters in both solvers. For the DMRG framework, we used a vanilla two-site setup, as it provides a conservative lower bound on what a more specialised tensor-network approach might achieve. Each trial starts from a random MPS ansatz, randomly varying the number of sweeps between 5 and 13, drawing the truncation error cutoff $\epsilon \in \{10^{-9}, 10^{-11}, 10^{-13}, 10^{-14}\}$, and sampling a terminal maximum bond dimension $m_{\text{max}} \in \{20, 40, 50, 100, 200, 300\}$. To ensure cross-encoding comparability, the ground-state energy of each QUBO Ising Hamiltonian is corrected by adding the Ising constant. The SA baseline executes a localised spin-flip kernel with a stochastical schedule for each trial. The initial temperature T_0 is drawn from $[5.0, 150.0]$, the final stop temperature $T_{\text{min}} \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, the cooling rate $\alpha \in [0.985, 0.999]$, and the Monte Carlo steps per temperature step from $[30, 500]$. Upon completion, the results of all 100 runs per instance are aggregated and compared against the exact global minimum extracted from the PUBO energy landscape and its computational time per instance.

V. Results

Fig. 2 summarises the optimization performance achieved by DMRG and SA.

a) Deviation from ground state: As the number of variables n in SAT increases, Ising formulations diverge. In the Quadratic Ising (*i.e.*, QUBO) regime, the median deviation for both solvers grows monotonically with n, rising from below 1% at small instance sizes to deviations exceeding 10% to 15% for $n \geq 12$ in the facets where $m \in \{129, 150\}$. This performance follows instance-dependent solution quality drops, observed for QUBO-encoded 3-SAT on quantum annealing platforms [12]. The interquartile range and high-deviation outliers both widen with n, indicating that larger QUBO instances are not only solved less accurately on average but also less reliably across independent trials. In contrast, for both solvers in the higher-order Ising case, the corresponding boxes remain close to zero deviation across the entire tested range of variables (note the chosen scale in $[0, 2]$ for this encoding). Localised deviations and minor outliers for DMRG within the higher-order pipeline only begin to manifest at the absolute highest complexity bounds where $m \geq 129$ and $n \geq 18$. The gap between the two encodings widens with instance size, since the QUBO pipeline's auxiliary-variable

overhead and pairwise-interaction density both scale with m. Comparing the two solvers directly within the quadratic encoding, SA matches or slightly outperforms DMRG for the majority of instances, and this performance gap widens as n and m increase, particularly past $n = 12$ where the upper whiskers in the DMRG case extend significantly higher than those in SA.

b) Runtime: Runtime measurements show a similar asymmetry: SA remains consistently faster than DMRG across all configurations. DMRG's runtime increases steadily with n in both encodings, exceeding $10s$ for moderate sizes and approaching $120s$ for the densest $m = 150$ instances. Notably, this runtime growth is not confined to the quadratic encoding as in the higher-order panels, where deviation remains close to zero across nearly the entire tested range, DMRG's runtime nevertheless scales with n in essentially the same manner as in the quadratic case.

This indicates that DMRG's cost is driven by MPO contraction and variational sweeping as n grows. These combined metrics demonstrate that DMRG's defining strength, namely efficiently representing many-body entanglement at bounded bond dimension, provides no advantage in this experiment, where GSs are inherently classical product states. Taken together, these results suggest that for SAT-derived combinatorial problems, the PUBO representation outperforms the QUBO representation in both runtime and ground state solution quality, since the auxiliary variables and additional coupling terms introduced during quadratisation appear to affect DMRG performance.

VI. Discussion and Outlook

Our principal technical result is that quadratisation is not a transparent compilation step. For the studied Max-3-SAT instances, the native PUBO encoding yields markedly better GS search results than the corresponding QUBO formulation, while SA matches or outperforms DMRG. Tensor-network methods thus offer no immediate advantage for this problem class, whose optima are classical product states. Spending additional classical computational resources may reduce the implementation overhead of DMRG, as tensor contractions and eigensolver steps admit distributed-memory and multi-GPU acceleration [13]. This would not, however, remove the encoding-induced difficulties. This distinction matters directly for integrated HPC/QC systems. Such systems require more than the dispatch of quantum kernels: they must decide when a quantum task should be replaced by a classical surrogate, and must expose sufficient information to support debugging and scheduling [14]. Our results show that this decision must be encoding- and instance-aware. Performance models proposed in Refs [15, 16] provide a natural mechanism: measurements such as those reported here can serve as baseline data for predicting solution quality and cost, and hence for selecting between SA, DMRG, and quantum execution. Within such a runtime, DMRG is best regarded not as a universally superior solver, but as a bounded classical surrogate. Where tractable, it can be a key component in implementing full (state) observability and provide an independently reproducible reference for debugging quantum subroutines [17], which is intrinsically unachievable with a pure quantum approach (including under noisy intermediate-scale quantum (NISQ) imperfections that complicate direct diagnosis [18]). It can also support approximate fallback execution when quantum resources are unavailable.

Future work should turn this characterisation into explicit policies for surrogate selection, cost-aware offloading, and observability hooks in HPC-QC runtimes. The broader lesson is that compiler and runtime design must account for the complete transformation path: representation choices can dominate solver behaviour and should be treated as first-class decisions.

Acknowledgements This work was supported by the German Federal Ministry of Research, Technology and Space (BMFTR), funding program 'Research Program Quantum Systems', grant number 13N17387, the High-Tech Agenda of the Free State of Bavaria, and the German Research Foundation, grant MA 9739/1-1.

References

[1] F. Arute, K. Arya, R. Babbush, et al., Quantum supremacy using a programmable superconducting processor, *Nature*, 2019.
[2] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, *Nature*, 2017.
[3] A. Elsharkawy, X.-T. M. To, P. Seitz, et al., Integration of quantum accelerators with high performance computing—a review of quantum programming tools, *ACM TQC*, 2025.
[4] E. Kaya, J. Echavarria, M. N. Farooqi, et al., A software platform to support disaggregated quantum accelerators, in *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2024.
[5] M. Schulz, M. Ruefenacht, D. Kranzlmuller, et al., Accelerating hpc with quantum computing: It is a software challenge too, *Computing in Science and Engineering*, 2022.
[6] R. Ramsauer and W. Mauerer, Towards system-level quantum-accelerator integration, in *QCE*, IEEE, 2025.
[7] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Annals of Physics*, 2011.
[8] L. Schmidbauer, E. Lobe, I. Schaefer, et al., It's quick to be square: Fast quadratisation for quantum toolchains, *ACM TQC*, 2026.
[9] T. Gabor, M. L. Rosenfeld, C. Linnhoff-Popien, et al., How to approximate any objective function via quadratic unconstrained binary optimization, in *SANER*, IEEE, 2022.
[10] L. Schmidbauer and W. Mauerer, Sat strikes back: Parameter and path relations in quantum toolchains, in *QSW*, IEEE, 2025.
[11] J. D. Biamonte, Nonperturbative k-body to two-body commuting conversion hamiltonians and embedding problem instances into ising spins, *Physical Review A*, 2008.
[12] T. Gabor, S. Zielinski, S. Feld, et al., Assessing solution quality of 3sat on a quantum annealing platform, in *Quantum Technology and Optimization Problems*. Springer International Publishing, 2019.
[13] R. Levy, E. Solomonik, and B. K. Clark, Distributed-memory dmrg via sparse and dense parallel tensor contractions, in *SC20: International Conference for HPC, Netw., Storage and Analysis*, IEEE, 2020.
[14] A. Delgado and P. Date, Defining quantum-ready primitives for hybrid hpc-qc supercomputing: A case study in hamiltonian simulation, *Frontiers in Computer Science*, 2025.
[15] S. Thelen and W. Mauerer, Predict and conquer: Navigating algorithm trade-offs with quantum design automation, in *QCE*, IEEE, 2025.
[16] S. Thelen, H. Safi, and W. Mauerer, Approximating under the influence of quantum noise and compute power, in *QCE*, IEEE, 2024.
[17] D. Rovara, L. Burgholzer, and R. Wille, A framework for debugging quantum programs, in *QSW*, IEEE, 2025.
[18] F. Greiwe, T. Krüger, and W. Mauerer, Effects of imperfections on quantum algorithms: A software engineering perspective, in *QSW*, IEEE, 2023.