Constrained optimization reformulation for thermal QUBO solvers

Edoardo Alessandroni [1 2] Sergi Ramos-Calderer [1 3] Ingo Roth [1] Emiliano Traversi [4] Michel Krispin [5] Stefan Walter [6] Martin Kliesch [5] Leandro Aolita [1]

[1]Technology Innovation Institute, Abu Dhabi, UAE [2]SISSA, Trieste, Italy [3]Universitat de Barcelona, Barcelona, Spain
[4]ESSEC Business School, Cergy-Pontoise, France [5]University of Technology, Hamburg, Germany [6]Fujitsu Germany GmbH

Technology Innovation Institute
FUJITSU
TUHH Hamburg University of Technology
Co-funded by the European Union

Abstract

Using approximate solvers to tackle constrained optimization problems requires mapping the problem to a Quadratic Unconstrained Binary Optimization (QUBO) problem, using a penalization weight M. Solvers are sensitive to the choice of this parameter, which impacts the quality of the solutions found. In this work, we propose an algorithm—applicable to various scenarios (e.g., feasibility, optimality)— to select M for a solver that samples from a thermal distribution, ensuring a given success probability based on the solver's characteristics.

Constrained optimization and the Big-M problem

We consider constrained optimization problems given by

$$\underset{\mathbf{x}\in\{0,1\}^n}{\text{minimize}}\ f(\mathbf{x}) = \mathbf{x}^t Q\mathbf{x} \quad \text{subject to} \quad A\,\mathbf{x} = \mathbf{b}, \tag{P}$$

for a problem dependent choice of $Q \in \mathbb{R}^{n\times n}$, $A \in \mathbb{Z}^{m\times n}$, and $\mathbf{b} \in \mathbb{Z}^m$.

Any constrained problem in the form of (P) can be mapped to a QUBO by promoting the constraints $A\,\mathbf{x} = \mathbf{b}$ to a quadratic penalty term, weighted by a penalization constant M. The *QUBO* formulation then reads

$$\underset{\mathbf{x}\in\{0,1\}^n}{\text{minimize}}\ \mathbf{x}^t Q\mathbf{x} + M(A\mathbf{x} - \mathbf{b})^2\,. \tag{P_M}$$

This step is crucial to control the hardness and quality of the solving procedure and leads to the *Big-M problem*: if M is too small, the constraints are insufficiently enforced, while an excessively large M may degrade solution quality and significantly increase solver run-time. [1]

Thermal solvers

The Big-M problem has been studied in [1] for exact solvers. Here, we focus on more realistic approximate solvers, specifically those that can categorized as thermal solvers:

- **Gibbs sampler**: a solver with sampling probability $p(\mathbf{x}) \sim e^{-E(\mathbf{x})\beta}$ at inverse temperature β. Mainly used as a proxy model here;
- **Simulated Annealer** (SA): Markov Chain Monte Carlo (MCMC) designed to sample from the Gibbs distribution at the final annealing temperature β;
- **Digital Annealer** (DA): MCMC similar in spirit to SA, but with efficient hardware-embedded speed-up tricks (parallel tempering, offsets, ...) [2]

Additionally, other protocols like quantum annealing can also be considered thermal solvers, under appropriate conditions. [3]

Benchmark problems

We test the proposed algorithm on the following constrained problems:

Travelling Salesman Problem (TSP)

Given a connected graph $G = (V, E)$, where the edge $e_{i,j}$ represents the cost of traveling from node i to node j, find the path that visits all the nodes in the graph, minimizing the overall total travel cost. The QUBO formulation can be written as $E(x) = E^{(o)}(x) + ME^{(p)}(x)$ where

$$E^{(o)}(x) = \sum_{t=1}^{n_v}\sum_{i\neq j=1}^{n_v} e_{i,j}x_{t,i}x_{t+1,j},$$

$$E^{(p)}(x) = \sum_{i=1}^{n_v}(1 - \sum_{t=1}^{n_v} x_{t,i})^2 + \sum_{t=1}^{n_v}(1 - \sum_{i=1}^{n_v} x_{t,i})^2.$$

Multiway Number Paritioning Problem (MNPP)

Given a set S of N positive numbers $S = \{c_1, \ldots, c_N\}$, the goal is to partition S into P disjoint subsets, such that the sums of the values in each subset are as close to each other as possible. Objective and penalization functions are

$$E^{(o)}(x) = \sum_{p=1}^{P}\left(\sum_{i=1}^{N} c_i x_{i,p} - \frac{1}{P}\sum_{i=1}^{N} c_i\right)^2,$$

$$E^{(p)}(x) = \sum_{i=1}^{N}\left(1 - \sum_{p=1}^{P} x_{i,p}\right)^2.$$

Algorithm

One algorithm, three applications: feasibility, optimality and inverse scenarios

$\beta \longrightarrow$ Given a constrained problem (P), $0 \leq \eta \leq 1$, and a thermal sampler at temperature β, find M such that the sampled solutions of the resulting QUBO reformulation are **feasible** with probability at least η. $\longrightarrow M$

$\beta, E_f \longrightarrow$ Given a constrained problem (P), $0 \leq \eta \leq 1$, energy E_f and a thermal sampler at temperature β, find M such that the sampled solutions of the resulting QUBO reformulation are **feasible, with energy at most E_f,** with probability at least η. $\longrightarrow M$

$M \longrightarrow$ Given a constrained problem (P), $0 \leq \eta \leq 1$, and penalty weight M, find β such that using a thermal sampler at temperature β on the resulting QUBO reformulation returns **feasible** solutions with probability at least η. $\longrightarrow \beta$

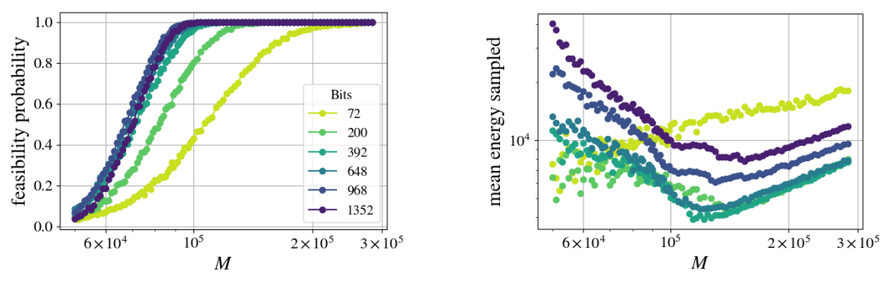
Algorithm: `M_for_ThermalSampler`$(E^{(o)}, E^{(p)}, E_f, \beta, \eta, v_{\max})$

input: Objective and penalization functions: $E^{(o)}$, $E^{(p)}$
Maximum desired energy of solutions: E_f
Inverse temperature: β
Minimal success probability: η
Maximum violation v: $v_{\max}$

1 $\tilde{p}_{\text{pen}} \leftarrow$ Estimate first few violation spectral densities $(E^{(p)}, v_{\max})$
2 $E_{\text{LB}} \leftarrow$ Compute objective lower bound via convex relaxation $(E^{(o)})$
3 $C_{feas} \leftarrow$ Uniformly sample from the feasible subspace $(E^{(o)}, E^{(p)})$
4 $g(M, \beta) \leftarrow$ Build analytically known function $(\beta, E_f, \eta, E_{\text{LB}}, C_{feas}, \tilde{p}_{\text{pen}})$
5 $M^* \leftarrow$ Find numerically (unique) zero of g
6 **return** M^*

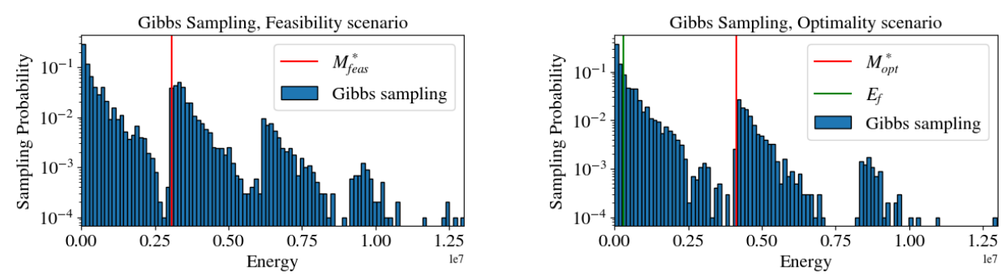
Results

Persistence of the Big M problem for DA: energy degradation



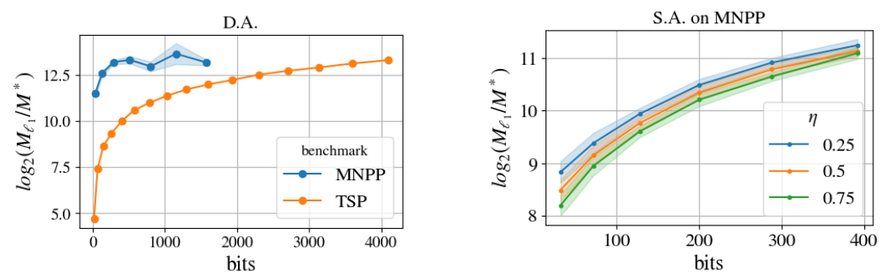
With larger M, the quality of the samples quickly deteriorates (right) after reaching feasibility (left).

Toy example



Demonstrative example: an instance addressed in the feasibility (optimality) scenario on the left (right), yielding a success probability of 0.68 $(0.56) \geq 0.5 = \eta$.

Iterations saved for binary search



Binary search is used to find target parameters, with fewer iterations when starting closer to the goal. Compared to naive choice $M_{\ell_1}(\beta) = ||Q||_{\ell_1} + \frac{1}{\beta}(n\ln(2) - \ln(\eta))$, using M^* saves $\log_2(M_{\ell_1}/M^*)$ iterations.

References

[1] Alessandroni et al., arXiv:2307.10379 (2023). [2] Y. Kao, arXiv/2311.05196 (2023). [3] Benedetti et al., arXiv:1510.07611 (2015).

ealessan@sissa.it