FUJITSU

shaping tomorrow with you

Analysis and Acceleration of Combinatorial Optimization Problems Including Inequality Constraints on Ising Machines

Matthieu Parizy, Fujitsu
Nozomu Togawa, Waseda University
AQC 2021 Day 4 (Poster F)

Summary

FUJITSU

- Ising machines have been successfully used to solve combinatorial problems in various fields.
- We wanted to study what was considered at the time “hard” problems for Ising machines. This led us to problems having inequality constraints.
- One such problem is the 0-1 Quadratic Knapsack Problem (QKP), which applications range from VLSI Design to budgeting problem.
- **We developed a visualization technique to analyze the bottlenecks of Ising model’s (or QUBO) formulations.**
- This provided us with hints on how to help the search for the optimal solution of QKP public benchmark with software improvement algorithm(SIA).
- We compare our results with only the 2nd gen. Digital Annealer, 2nd gen. DA + SIA and simulated annealing(SA) for baseline.
- Finally, we compare our results with the newly released 3rd gen. DA which features an inequality constraint interface.

Index

FUJITSU

Index

FUJITSU

Research Goals

FUJITSU

- We are using QKP to illustrate the challenges for Ising Machines to handle inequality constraints.
- To demonstrate how we can analyze QUBO to tune them and their behavior on an Ising machine
  - **Main Contribution: A novel energy landscape visualization technique**
- To demonstrate how analysis could also provide us with hints on developing a software improvement algorithm to enhance solutions provided by Ising machines
  - **Sub Contribution: A software improvement algorithm(SIA) able to improve overall QKP solving performance**

Index

FUJITSU

Quadratic Knapsack Problem Formulations

FUJITSU

General Formulation

$$\max_{x} \quad \sum_{i=0}^{n} \sum_{j=i}^{n} v_{ij} x_i x_j$$

$$\text{subject to:} \quad \sum_{i=0}^{n} w_i x_i \leq C \; x_i \in \{0,1\} \forall i$$

Hamiltonian Formulation (QUBO)

$$H = \underbrace{\sum_{i=0}^{n} \sum_{j=i}^{n} -v_{ij} x_i x_j}_{H_{cost}} + \underbrace{\alpha \left(\sum_{i=0}^{n} w_i x_i - C + y \right)^2}_{H_{constraint}}$$

- v_{ij} is the profit when both item i and item j are added to the knapsack.
- x_i is the binary variable representing if item i is chosen or not.
- w_i is the weight of item i, and C is the capacity of the knapsack.
- y is the auxiliary continuous variable for when the capacity is not fully used.
- α is the coefficient which to be high enough for the penalty not be violated.
- **Finding the right α balance and encoding for y is widely regarded in the community as challenging.**

Index

FUJITSU

How to find a problem's bottleneck?

- **[Answer]** Visualize a problem's energy landscape
  - It will explain why it is hard to solve
  - It will provide hints on how to make it easier to solve

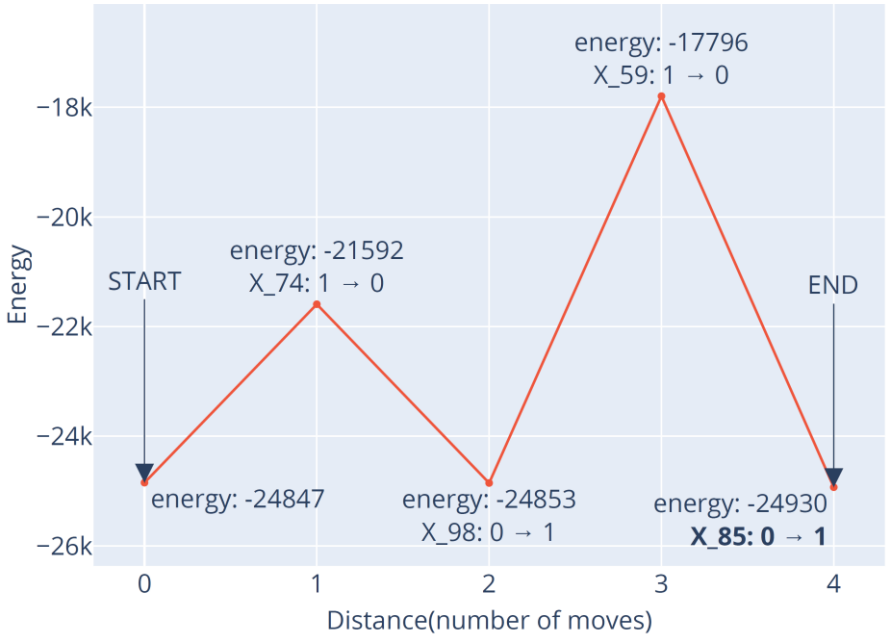


Fig: energy landscape example

Visualizing Energy Landscape

FUJITSU

- Algorithm explanation
  - Take 2 solutions of interest, which we call START and END.
    - For instance, a local minimum and an optimal solution
  - Go from START to END by greedily flips the bits of *start* which differs from *end* as to have a minimal energy increase
  - Plot on *x* the number of flips done
  - Plot on *y* the flips corresponding energy value

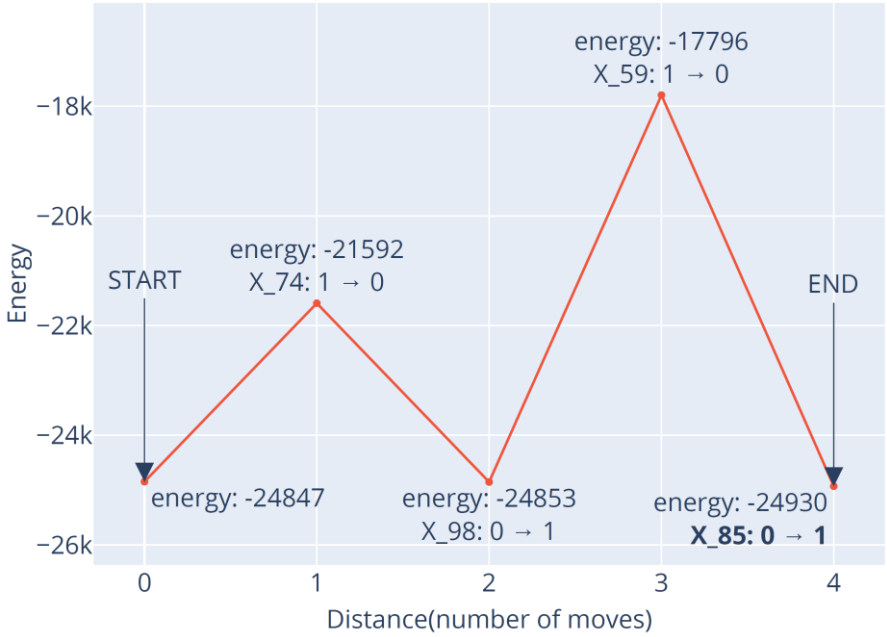


Fig: energy landscape example

QKP Energy Landscape Example

- This example comes from attempting to solve instance 100_25_1 from the QKP soutif benchmark
- α penalty coeff. value is 5.5
- y uses a OneHot encoding with range covering up to $\max w_i$ of the instance
- Some runs from 2nd gen DA converge to sub optimal solution represented as START
- Actual optimal solution is represented as END
- START and END are only 4 bit flips away from each other which represent two item swaps
- $x_{59} \Leftrightarrow x_{85}$ swap creates the biggest mountain

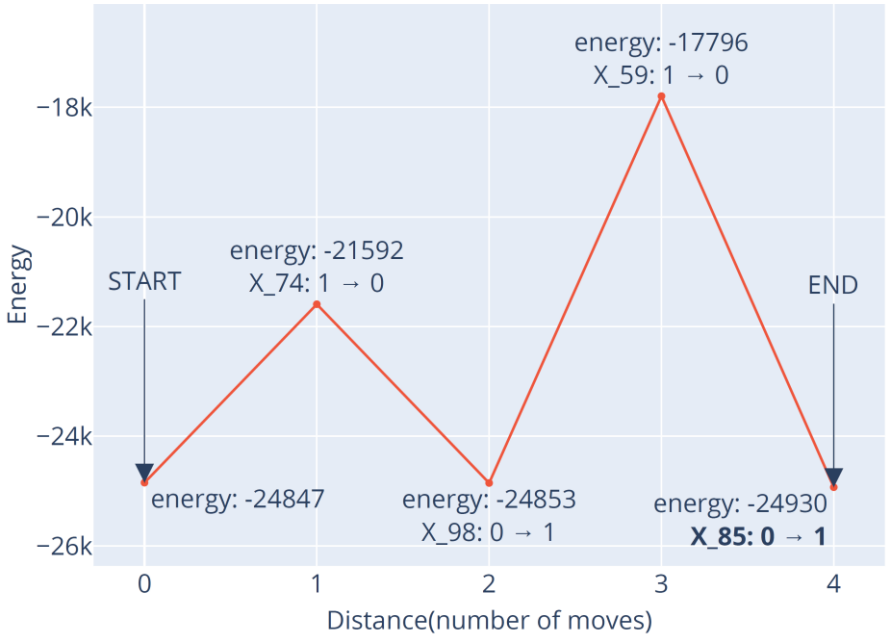


Fig: energy landscape example for QKP soutif instance r_100_25_1

Where does the $x_{59} \Leftrightarrow x_{85}$ mountain comes from?

FUJITSU

- x_{59} is lighter than x_{85} and has a best cost performance which makes it easier to choose/hard to remove
- We observe QUBO linear values order is higher than profit values
- Therefore, filling the knapsack has a higher priority than maximizing profit
- We need an efficient way to swap heavy items
  - They are hard to swap due to high temperature needed.
  - At high temperature small items also move which make heavy item swap trials difficult

item cost performance

QUBO linear values

	$\sum_j v_{ij}$	w_i	$\frac{\sum_j v_{ij}}{w_i}$	$Q_{i,i}$
mean	1,222.4	26.1	99.0	-293,731.6
std	255.7	13.6	175.9	151,532.1
min	728.0	1.0	18.2	-558,250.0
Q1	1,051.2	15.0	**31.9**	**-426,778.0**
med	1,212.0	25.5	45.8	-288,142.2
Q3	1,368.2	38.0	87.9	-170,362.5
max	2,206.0	50.0	1,201.0	-11,434.5
x_{59}	1668	36	46.3	-404,712.0
x_{85}	1162	38	**30.6**	**-426,778.0**

Solution Improvement Algorithm (SIA)

- SIA is executed after obtaining solutions from 2nd gen. DA
- Solution improvement is done by looking for the best possible items swaps in/out of the knapsack, which do not violate the constraint
- Since calculating all combinations energy impact is intensive, we consider only removing items with the lowest cost performance(RPD).
- In the example, we set SIA to target the two worse RPD items for swapping.

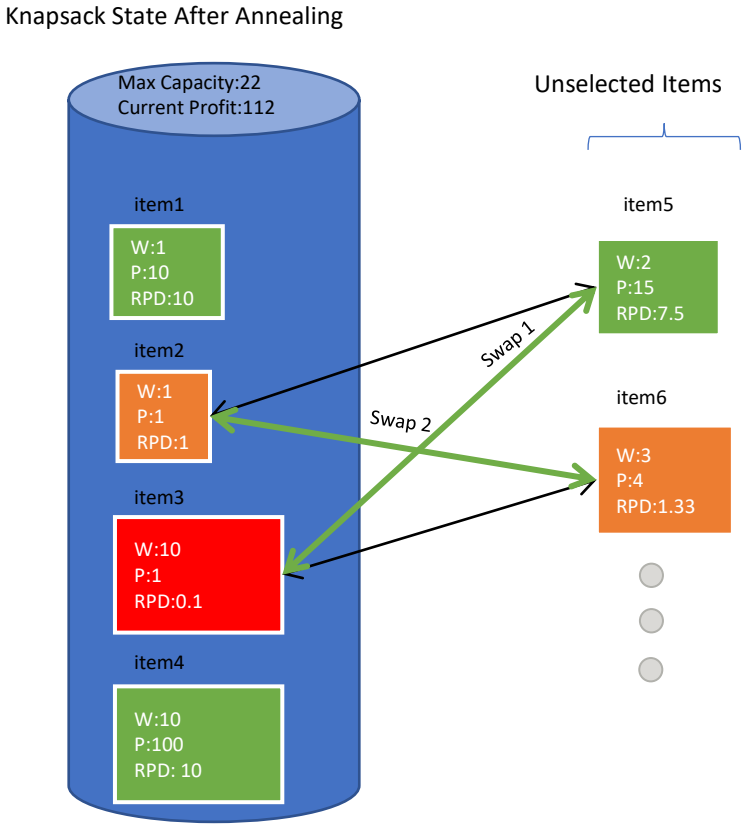


Fig: SIA toy example with a non quadratic knapsack

Index

FUJITSU

Experiments

FUJITSU

- Benchmark description
  - We use the soutif QKP benchmark
  - It is composed of 3 size of instances:100,200, 300.
  - 4 kinds of density for the profit matrix for each size, 25%,50%,75% and 300%
  - For each combination of density and size, 10 random instances were generated.
  - Weights values have a normal distribution between 1 and 50
  - Profit values have a normal distribution between 1 and 100
  - Capacity takes a random value between 50 and $\sum w_i$
- 2^{nd} gen DA setting
  - We use parallel tempering with 26 replicas, Tmax= 9000, Tmin = 0 with a temperature exchange every 100 iterations. We do 750M iterations runs which is equivalent to 40s.
- SIA settings
  - We set it to target the worst 15 items RPD wise for swapping. For 2^{nd} Gen DA + SIA runs, DA runs 20s and SIA 20s
- SA settings
  - For SA, we use D-Wave neal sampler, with a number of steps of 2.75M which also take 40s

Results

FUJITSU

Table: Results summary

solver	success%
Neal	0.0%
2nd Gen DA	6.7%
2nd Gen DA+SIA	60.7%

- We measure over 20 random seed the average % of chance of finding the optimal solution over all available instances (success%).
- Both Neal and 2^{nd} gen DA perform poorly.
- Whereas 2^{nd} gen DA+SIA performs much better.

3rd Gen DA Trial

FUJITSU

- In April, a 3rd gen DA was release featuring a linear inequality interface.
- It can handle constraints of the form
  - $w_0x_0 + w_1x_n \ldots + w_nx_n \leq C$
  - This is our case for QKP
- We ran the same experiment with it

3rd Gen DA Results

FUJITSU

solver	success%
Neal	0.0%
2nd Gen DA	6.7%
2nd Gen DA+SIA	60.7%
3rd Gen DA	100%

All instances could be solved with 3rd gen DA. in less than 5 seconds

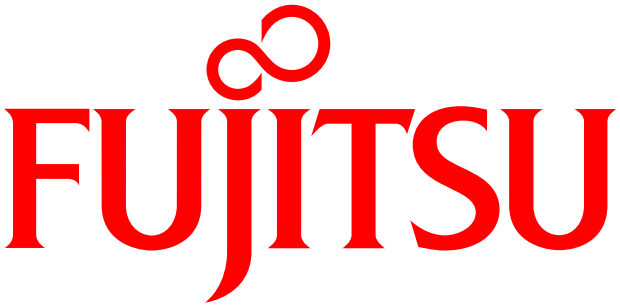
Index

FUJITSU

Conclusion

FUJITSU

- We proposed a novel and simple visualization technique.
- It helped us identify bottlenecks when solving QUBO with inequality constraints.
- It gave us hints to circumvent the heavy item removal bottleneck with SIA
- Finally, we illustrated how inequality constraint dedicated handling can bring tremendously better results on the 3rd gen DA



shaping tomorrow with you