Quantum annealing

faktoryzacja liczb

Autorzy: Andrzej Starzyk i Michał Szewc

Quantum annealing

To proces wyżarzania wykorzystujący “quantum fluctuations” zamiast wahań temperatury, które jest wykorzystywane w zwykłym symulowanym wyżarzaniu.

Wyżarzanie kwantowe polega na dojściu do stanu podstawowego układu poprzez ciągłą zmianę stanu.

W teorii Model gwarantuje znalezienie stanu podstawowego, ale przez zakłócenia i czynniki zewnętrzne prawdopodobieństwo maleje.

Faktoryzacja jako problem optymalizacyjny

Przedstawienie problemu jako QUBO (quadratic unconstrained binary optimization)

Postanowiliśmy użyć metody wykorzystującą zmodyfikowaną tablicę mnożenia. Znacznie zmniejsza ona liczbę zmiennych, a więc zmniejsza możliwe zakłócenia i szansę na niepoprawny wynik.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
p					1	p_2	p_1	1
q					1	q_2	q_1	1
					1	p_2	p_1	
				q_1	$p_2 q_1$	$p_1 q_1$	q_1	
			q_2	$p_2 q_2$	$p_1 q_2$	q_2		
		1	p_2	p_1	1			
carries		c_4	c_3	c_2	c_1			
$p \times q = 143$	1	0	0	0	1	1	1	1
		column 4		column 3		column 2		column 1

Faktoryzacja jako problem optymalizacyjny c.d.

Na podstawie odpowiedniego podziału tablicy na kolumny (aby zminimalizować liczbę zmiennych odpowiedzialnych za przeniesienie) tworzymy równania.

$$(p_2 + p_1 q_1 + q_2 - (c_2 \times 4 + c_1 \times 2)) \times 2 + (p_1 + q_1) = (11)_2 = 3 \quad (13)$$

$$(q_1 + p_2 q_2 + p_1 + c_2 - (c_4 \times 4 + c_3 \times 2)) \times 2 + (1 + p_2 q_1 + p_1 q_2 + 1 + c_1) = (01)_2 = 1 \quad (14)$$

$$(1 + c_4) \times 2 + (q_2 + p_2 + c_3) = (100)_2 = 4 \quad (15)$$

Faktoryzacja jako problem optymalizacyjny c.d.

Po uproszczeniu równań i sprowadzeniu do postaci z prawą stroną równą zero, tworzymy funkcję, która jest sumą kwadratów lewych stron tych warunków.

Funkcja ta osiąga minimum globalne, dla wartości odpowiadającym liczbom, których iloczyn to 143.

$$2p_2 + 2p_1q_1 + 2q_2 - 8c_2 - 4c_1 + p_1 + q_1 - 3 = 0$$

$$2q_1 + 2p_2q_2 + 2p_1 + 2c_2 - 8c_4 - 4c_3 + p_2q_1 + p_1q_2 + c_1 + 1 = 0$$

$$q_2 + p_2 + c_3 + 2c_4 - 2 = 0$$

$$\begin{aligned} f \quad = \quad & (2p_2 + 2p_1q_1 + 2q_2 - 8c_2 - 4c_1 + p_1 + q_1 - 3)^2 \\ & + (2q_1 + 2p_2q_2 + 2p_1 + 2c_2 - 8c_4 - 4c_3 + p_2q_1 + p_1q_2 + c_1 + 1)^2 \\ & + (q_2 + p_2 + c_3 + 2c_4 - 2)^2. \end{aligned}$$

Faktoryzacja jako problem optymalizacyjny c.d.

Model wyżarzania nie może mieć relacji 3 lub więcej czynników, a w wynikowej funkcji powstaną np. $c_1p_1q_1$ lub $p_1p_2q_1q_2$. Wystąpienia typu X^2 można zastąpić X, ponieważ zakres wartości zmiennych to {0, 1}.

Dla pozostałych przypadków:

$$\begin{cases} x_1x_2x_3 = \min\limits_{x_4}(x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4)) \\ -x_1x_2x_3 = -\min\limits_{x_4}(x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4)). \end{cases}$$

Zastępujemy tak pary p_1q_2,p_1q_2,p_1q_2,p_1q_2 odpowiednio zmiennymi t_1,t_2,t_3,t_4.

Przykład: $p_1p_2q_1$ zostanie zastąpione $t_1q_2 + 2(p_1q_1-2p_1t_1-2q_1t_1+3t_1)$. (pomijamy min, ponieważ minimalizujemy wynik.

Faktoryzacja jako problem optymalizacyjny c.d.

Ostatni krok: przejście z wartości {-1,1} na {0,1} podstawiając $x_i = (1 - s_i)/2$.

Ostateczna postać uproszczona funkcji F przyjmującej zmienne z Isingu (dla $p_1 = (1 - s_1)/2$, $p_2 = (1 - s_2)/2$, $q_2 = (1 - s_3)/2$, itd.) to:

$$
\begin{aligned}
f\prime &= (p_1, p_2, q_1, q_2, c_1, c_2, c_3, c_4, t_1, t_2, t_3, t_4) \\
&= (261s_1)/2 + (215s_2)/2 + (261s_3)/2 + (215s_4)/5 - 41s_5 - 82s_6 + 3s_7 + 6s_8 - 137s_9 - 81s_{10} - 107s_{11} - 81s_{12} + 2s_1s_2 + 79s_1s_3 \\
&\quad + (95s_1s_4)/2 + (95s_2s_3)/2 - 2s_1s_5 + 71s_2s_4 - 4s_1s_6 - 8s_2s_5 + 2s_3s_4 - 8s_1s_7 - 16s_2s_6 - 2s_3s_5 - 16s_1s_8 \\
&\quad + s_2s_7 - 4s_3s_6 - 8s_4s_5 - 148s_1s_9 + 2s_2s_8 - 8s_3s_7 - 16s_4s_6 - 84s_1s_{10} + 6s_2s_9 - 16s_3s_8 + s_4s_7 + 34s_5s_6 \\
&\quad + 6s_2s_{10} - 148s_3s_9 + 2s_4s_8 - 4s_5s_7 - 124s_2s_{11} + 6s_4s_9 - 8s_5s_8 - 8s_6s_7 - 84s_2s_{12} - 84s_4s_{10} - 8s_5s_9 - 16s_6s_8 \\
&\quad - 84s_3s_{12} - 124s_4s_{11} + s_5s_{10} - 16s_6s_9 + 34s_7s_8 + 6s_4s_{12} + 2s_5s_{11} + 2s_6s_{10} + s_5s_{12} + 4s_6s_{11} \\
&\quad - 4s_7s_{10} + 2s_6s_{12} - 8s_7s_{11} - 8s_8s_{10} - 4s_7s_{12} - 16s_8s_{11} - 8s_8s_{12} + s_9s_{11} + 794
\end{aligned}
$$

Faktoryzacja jako problem optymalizacyjny c.d.

Po przeliczeniu z funkcji z QUBO na parametry modelu Isinga otrzymujemy:

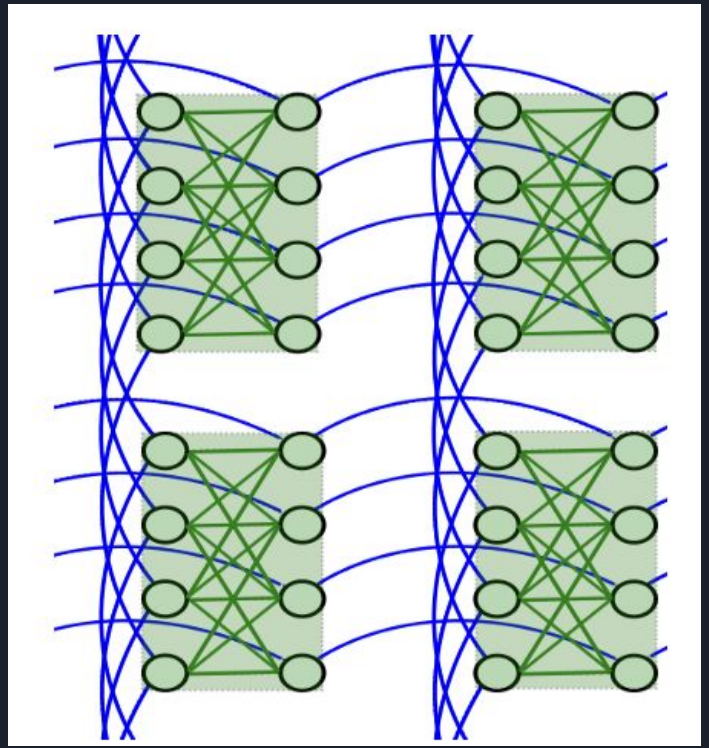
$$h = [130.5 \quad 107.5 \quad 130.5 \quad 107.5 \quad -41 \quad -82 \quad 3 \quad 6 \quad -137 \quad -81 \quad -107 \quad -81]$$

$$J = \begin{bmatrix}
2 & 79 & 47.5 & -2 & -4 & -8 & -16 & -148 & -84 & 0 & 0 \\
 & 47.5 & 71 & -8 & -16 & 1 & 2 & 6 & 6 & -124 & -84 \\
 & & 2 & -2 & -4 & -8 & -16 & -148 & 0 & 0 & -84 \\
 & & & -8 & -16 & 1 & 2 & 6 & -84 & -124 & 6 \\
 & & & & 34 & -4 & -8 & -8 & 1 & 2 & 1 \\
 & & & & & -8 & -16 & -16 & 2 & 4 & 2 \\
 & & & & & & 34 & 0 & -4 & -8 & -4 \\
 & & & & & & & 0 & -8 & -16 & -8 \\
 & & & & & & & & 0 & 1 & 0 \\
 & & & & & & & & & 0 & 0 \\
 & & & & & & & & & & 0
\end{bmatrix}$$

Symulowanie kwantowego wyżarzania

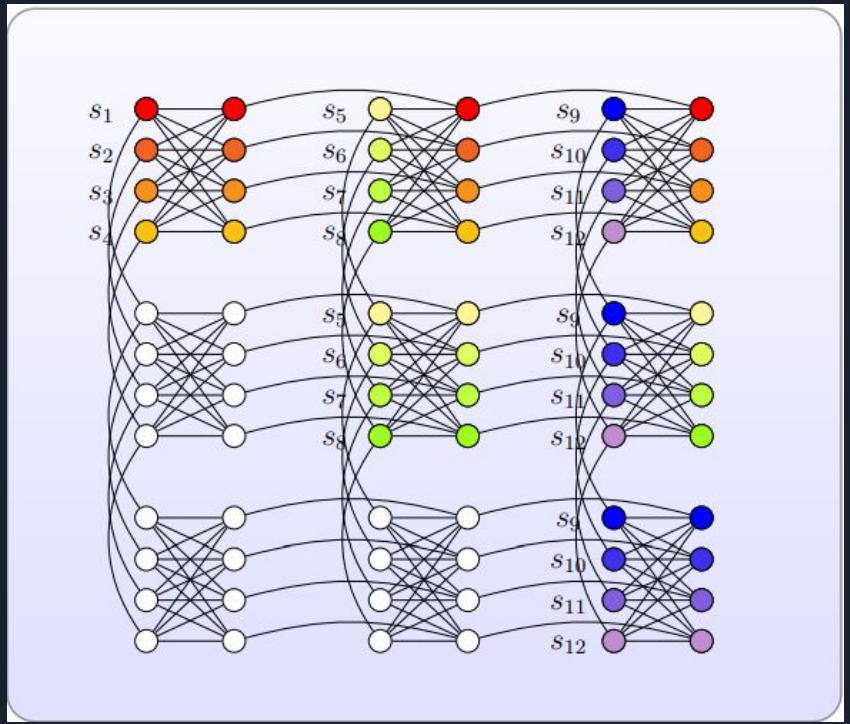
Mając parametry, możemy przejść do symulowania kwantowego wyżarzania.

Tworzymy wirtualną sieć, symulującą maszynę D-Wave. Przez ograniczenia związane z sąsiedztwem kubitów, konieczne jest użycie 4-krotnie więcej kubitów niż zmiennych (każda para zmiennych posiada reprezentację w kubitach, które ze sobą sąsiadują).



Embedding

- Struktura grafu reprezentującego problem
- Graf docelowy / architektura komputera kwantowego
- Odwzorowanie pierwszego grafu na drugi

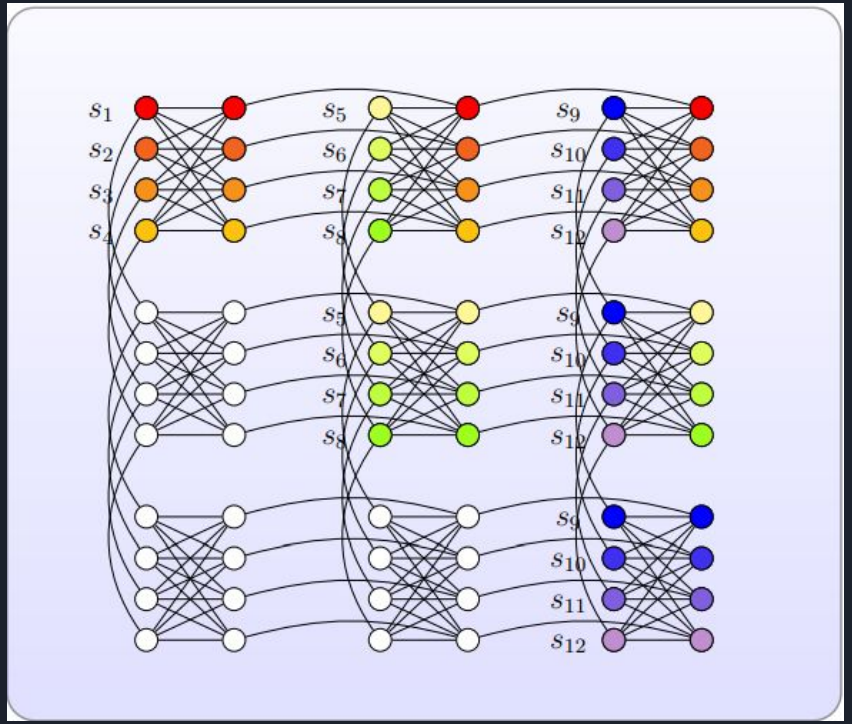


Symulowane wyżarzanie vs symulacja modelu Isinga

- Malejąca temperatura
- Formalne zdefiniowanie problemu (QUBO)
- Embedding problemu w grafie roboczym
- qubity vs +/- 1

Równoległe symulowane wyżarzanie

- Podział node'ów pomiędzy procesy
- Wyznaczenie danych, którymi procesy muszą się dzielić
- Synchronizacja komunikacji



Wyniki 😕

Instance	SplitPos	Model	No. of variables	$\|h\|$ range	$\|J\|$ range	Frequency[a)]
143=11×13	[1,3,5]	[33]	12	1-137	1-148	100%
		Proposed	5	1-120	1-96	100%
221=13×17	[1,4,6]	[33]	15	1-462.5	1-520	100%
		Proposed	9	1-186	1-130	100%
247=13×19	[1,4,6]	[33]	15	1-458	1-520	100%
		Proposed	10	1-1050	1-888	100%
323=17×19	[1,4,6]	[33]	20	1-532.5	1-448	95.2%
		Proposed	13	1-159	1-152	100%
437=19×23	[1,4,6]	[33]	20	1-551.5	1-448	99.8%
		Proposed	13	1-181	1-130	100%
589=19×31	[1,4,7]	[33]	19	1-757.5	1-616	100%
		Proposed	12	1-386.5	1-400	100%
667=23×29	[1,4,6]	[33]	20	1-577.5	1-448	85.5%
		Proposed	12	1-122	1-136	100%
899=29×31	[1,4,6]	[33]	20	1-550.5	1-448	89.9%
		Proposed	13	1-195	1-152	100%
989=23×43	[1,4,7]	[33]	24	1-1292	1-1172	90.4%
		Proposed	16	1-581	1-504	100%
1073=29×37	[1,4,7]	[33]	24	1-1235	1-1172	16.5%
		Proposed	18	1-884	1-856	100%
1517=37×43	[1,4,7]	[33]	29	1-1530.5	1-1464	4.2%
		Proposed	22	1-1504	1-1392	92.3%
2449=31×79	[1,4,7]	[33]	28	1-1512.5	1-1212	17.9%
		Proposed	23	1-1837.5	1-1408	84.3%
59989=251×239	[1,4,7,10,13]	[33]	59	1-2947	1-1832	1.4%
		Proposed	52	1-2948	1-1792	4%
376289=659×571	[1,5,8,11,14,17]	[33]	95	1-7505	1-4848	0.2%
		Proposed	90	1-11624	1-4592	0.7%
1005973=997×1009	[1,4,7,10,13,16]	[33]	96	1-5977.5	1-2744	0.2%
		Proposed	89	1-7032	1-2812	0.9%

a) Frequency denotes the percentage of correct solutions produced by the algorithm.

Bibliografia

- 1804.02733 Shuxian Jiang1, Keith A. Britt, Alexander J. McCaskey, Travis S. Humble, and Sabre Kais “Quantum Annealing for Prime Factorization”
- Quantum Annealing for Prime Factorization | Scientific Reports Shuxian Jiang, Keith A. Britt, Alexander J. McCaskey, Travis S. Humble & Sabre Kais “Quantum Annealing for Prime Factorization”
- Intro to Quantum Annealing. Let’s demystify this new buzzword… | by Dominic Plein | Medium Dominic Plein “Intro to Quantum Annealing”
- Simulated Annealing Tutorial
- D-Wave QPU Architecture: Topologies — D-Wave System Documentation documentation
- WangChun Peng, BaoNan Wang, Feng Hu, YunJiang Wang, XianJin Fang3 XingYuan Chen, and Chao Wang “Factoring larger integers with fewer qubits via quantum annealing with optimized parameters”

Bibliografia

- “Next-Generation Topology of D-Wave Quantum Processors” Kelly Boothby, Paul Bunyk, Jack Raymond, Aidan Roy