An algorithm for the Potts model with many states

Diana Fanghänel Frauke Liers

University of Cologne

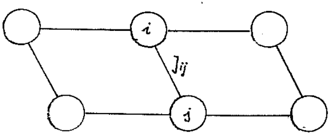
CompPhys08, Leipzig

Potts model

The Potts model is a spin model with

- n spins,
- the spins i, j are coupled with coupling strength J_{ij},
- each spin can take q different states $\sigma_i \in \{0, 1, \ldots, q-1\}$,
- Hamiltonian

$$H(\sigma) = -\sum_{<i,j>} J_{ij}\delta(\sigma_i - \sigma_j).$$



Reformulations [Juhász, Rieger, Iglói (2001)]

The partition function (with $\beta = 1/(k_0 T)$) is

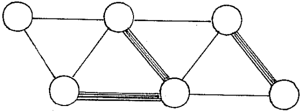
$$Z = \sum_{\sigma} \exp(-\beta H(\sigma)).$$

With $w_{ij} = \log_q(\exp(\beta J_{ij}) - 1)$ for all $(i,j) \in E(G)$ it is

$$Z = \sum_{F \subseteq E(G)} q^{c_G(F) + \sum_{(i,j)\in F} w_{ij}} = \sum_{F \subseteq E(G)} q^{f(F)}.$$

$c_G(F)$: number of connected components of $G(F) = (V, F)$ and

$$f(F) = c(F) + \sum_{(i,j)\in F} w_{ij}.$$



Approximation of the partition function

$$\begin{aligned} \text{Let } f^* &= \max_{F \subseteq E(G)} f(F) \quad \text{and} \\ \Psi_G^* &= \text{set of optimum solutions.} \end{aligned}$$

$$\text{Then,} \quad q^{f^*} |\Psi_G^*| \leq Z = q^{f^*} \big(|\Psi_G^*| + \sum_{F \subseteq E(G) \setminus \Psi_G^*} q^{f(F) - f^*} \big).$$

For large numbers q it holds

$$\ln Z \approx f^* \ln q + \ln |\Psi_G^*|.$$

Problem formulation

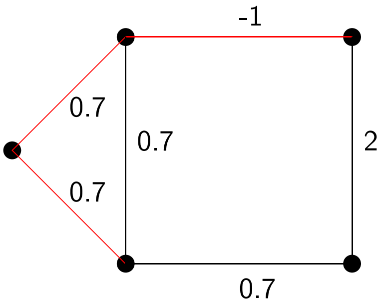
We want to solve the problem

$$\max\{f(A) = c_G(A) + w(A) : \ A \subseteq E\}, \tag{1}$$

where

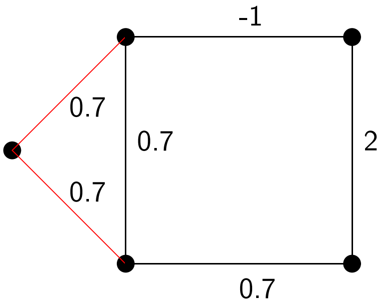
$G = (V, E)$	...	is a simple graph
$w_e \in \mathbb{R}$	...	are the edge weights
$w(A) = \sum_{e \in A} w_e$	...	is the weight of all edges in A
$c_G(A)$	...	is the number of connected components of $G(A) = (V, A)$.

A small example



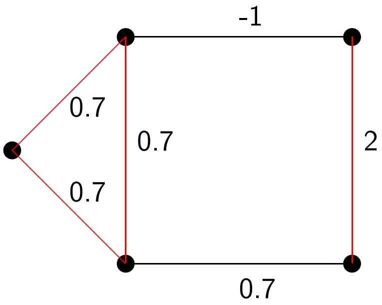
- $w(A_1) = 0.4,\ c_G(A_1) = 2,\ f(A_1) = 2.4$

A small example



- $w(A_2) = 1.4,\ c_G(A_2) = 3,\ f(A_2) = 4.4$

A small example



- $w(A^*) = 4.1,\ c_G(A^*) = 2,\ f(A^*) = 6.1$

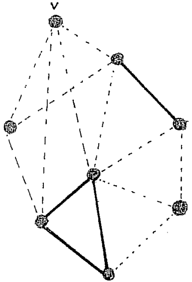
Idea of the algorithm

Anglés d'Auriac, Iglói, Preissmann, Sebő (2002)

Let $A \subseteq E$ be optimal for $G(U)$, $U \subseteq V$. Let $v \in E \backslash U$. Then there exists an edge set $A^* \supseteq A$ that is optimal for $G(U \cup \{v\})$.

Let W be a set of components of $G' = (U \cup \{v\}, A)$ with $v \in W$. Then adding all edges with endnodes in different components of W changes the function value by

$$w(E(W)) + 1 - |W|.$$



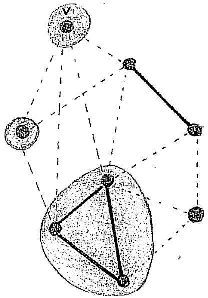
Idea of the algorithm

Anglés d'Auriac, Iglói, Preissmann, Sebő (2002)

Let $A \subseteq E$ be optimal for $G(U)$, $U \subseteq V$. Let $v \in E \backslash U$. Then there exists an edge set $A^* \supseteq A$ that is optimal for $G(U \cup \{v\})$.

Let W be a set of components of $G' = (U \cup \{v\}, A)$ with $v \in W$. Then adding all edges with endnodes in different components of W changes the function value by

$$w(E(W)) + 1 - |W|.$$



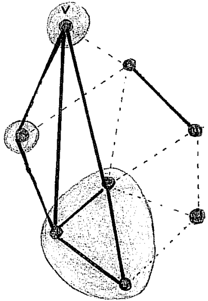
Idea of the algorithm

Anglés d'Auriac, Iglói, Preissmann, Sebő (2002)

Let $A \subseteq E$ be optimal for $G(U)$, $U \subseteq V$. Let $v \in E \setminus U$. Then there exists an edge set $A^* \supseteq A$ that is optimal for $G(U \cup \{v\})$.

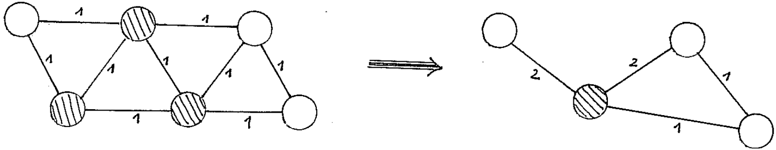
Let W be a set of components of $G' = (U \cup \{v\}, A)$ with $v \in W$. Then adding all edges with endnodes in different components of W changes the function value by

$$w(E(W)) + 1 - |W|.$$



Shrinking

We can shrink the connected components of $G(U)$, i.e., the computed edges.



$$\begin{array}{rcl} X & \Rightarrow & x \\ \{(v,u) : u \in X\} & \Rightarrow & (v,x) \text{ for all } v \in V \setminus X \text{, weights are added} \end{array}$$

We can shrink edges e with weight $w_e \geq 1$.

The basic algorithm of Anglés d'Auriac et al.

Input: A graph $G = (V, E)$ with $w_e \in (0, 1)$ for all $e \in E(G)$.

Output: An optimal solution A^* of G.

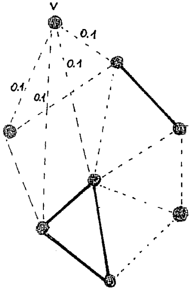
1. Set $U := \emptyset$ and $A := \emptyset$.
2. Choose a vertex $v \in V \setminus U$.
3. Determine a set of nodes $W \subseteq U \cup \{v\}$, $v \in W$ that minimizes $|W| - 1 - w(E(W))$ by computing a minimum $s - t$ cut in a graph $D_{U,v}$.
4. Set $U := U \cup \{v\}$, shrink the vertex set W in the graph G, set $U := U/W$ and update the computed edge set A.
5. If $U \neq V(G)$ go to Step 2.; else output $A^* = A$ and STOP.

Consider all incident edges of some node v. If the sum of their weights is smaller than 1, then these incident edges are not in an optimal edge set.
All nodes $v \in V$ with $w(\delta(v)) \leq 1$ are successively deleted in G.

Another idea is to recognize the case $W = \{v\}$ early.

Assume $\delta_U(v) = \{(v, u) \in E : \ u \in U\}$ and $w(\delta_U(v)) \leq 1$. Then $W = \{v\}$ is optimal for

$$\begin{aligned} & |W| - 1 - w(E(W)) \to \min \\ & s.t. \quad W \subseteq U \cup \{v\}, \ v \in W. \end{aligned}$$



Subgraphs

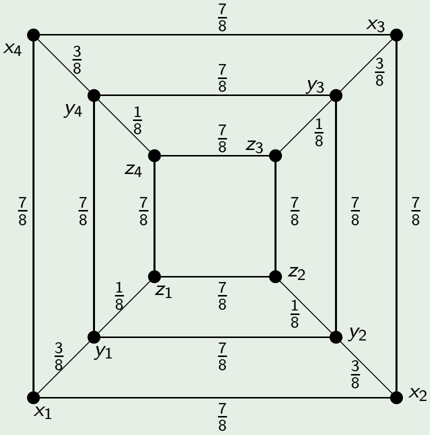
Let $U \subseteq V$ such that the edge set E_U of $G(U)$ is optimal for it. Then $E_U \subseteq A^*$ for some $A^* \in \Psi^*_G$.

We can shrink the edge set E_U ,i.e. U, in the graph.

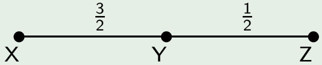
Let C be a chordless cycle with $n \geq 3$ vertices and edge weights $w_e \in (0, 1)$ for all edges $e \in E(C)$. If $\sum_{e \in E(C)}(1 - w_e) \leq 1$, then $E(C)$ is optimal for C.

- We found similar results for forests, subgraphs of wheels, complete graphs and special grids.

Example



shr(G):



An improved algorithm

Input: $G = (V, E)$ with $w_e \in (0, 1)$ for all edges $e \in E(G)$.

1. Set $G := G(V \backslash notU)$, $U := \emptyset$ and $A := \emptyset$.
2. Choose a vertex $v \in V \backslash U$.
3. While you find a set $W \subseteq U \cup \{v\}$, $v \in W$, such that the induced edge set $E(W)$ is optimal for $G(W)$, shrink W in $U \cup \{v\}$ and G to node v and update A.
4. Determine $W \subseteq U \cup \{v\}$ that solves $\min\{|W| - 1 - w(E(W)) : \; W \subseteq U \cup \{v\}, \; v \in W\}$. Shrink W in $U \cup \{v\}$ and G. Update A and U.
5. If $U \neq V(G)$ go to Step 3.; otherwise **Output:** The optimum solution $A^* = A$.

Test instances

- Tests on:
  - toroidal square $L \times L$ grids with weights $w_1 \in (0,1)$ and $w_2 = 1 - w_1$ where $p\%$ of the edges have weight w_1,
  - random graphs with n nodes, m edges
- OGDF library
- Minimum cuts: fast implementation of the Goldberg-Tarjan Algorithm (mincut package of Jünger, Rinaldi and Thienel)

Results

- $L = 128,\ w_1 = 0.2$:

p	80	70	60	50	40	30	20
Basic algorithm	136.7	142.8	150.5	115.6	34.0	16.4	12.9
Improved Method	0.5	4.9	19.5	19.4	5.2	3.3	2.2
Heuristic	0.5	4.6	7.6	3.9	4.5	3.3	2.2

- $w_1 = 0.2,\ p = 50\%$:

L	32	64	128	256	512
Basic algorithm	0.55 sec	6 sec	2min	40min	-
Improved Method	0.19 sec	1.2 sec	20sec	6min	1h40min
Heuristic	0.01 sec	0.3 sec	4 sec	1min	15min

	L	p=80	p=70	p=60	p=50	p=40	p=30
Impr. Meth.	128	9.24	17.26	28.55	74.95	8.09	3.38
Basic Alg.	128	142.64	149.96	150.52	279.99	15.08	13.09
Impr. Meth.	256	118.55	252.29	473.60	2116.18	81.03	32.09
Basic Alg.	256	2248.08	2418.49	2614.13	8454.78	219.73	208.77

Table: Solution times in seconds for grids with $w_1 = 0.4$

	n	$m = 2n$	$m = 3n$	$m = 4n$	$m = 5n$
Improved Method	10.000	8.45	20.39	28.92	39.51
Basic algorithm	10.000	22.67	38.72	53.16	67.65
Improved Method	25.000	56.30	156.42	245.02	341.10
Basic algorithm	25.000	146.76	263.78	379.83	502.49
Improved Method	50.000	404.303	929.7325	1318.943	1786.865
Basic algorithm	50.000	697.6215	1297.199	1904.5525	2584.315

Table: Solution times in seconds for random graphs.

Thank you for your attention!