A Fast Max-Cut Algorithm on Planar Graphs

Frauke Liers   Gregor Pardella

Institut für Informatik
Universität zu Köln

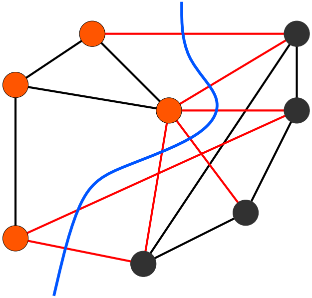
13th Combinatorial Optimization Workshop

Aussois

January 11-17, 2008

The Max-Cut Problem

A MAXIMUM CUT $\delta(Q)$ in a (weighted) graph $G = (V, E)$ is a node set $Q \subseteq V$ with maximum weight $w(\delta(Q)) = \sum_{e \in \delta(Q)} w(e)$.



The Max-Cut Problem

Complexity Status

- NP-hard in general
- poly-time solvable graph classes exist, e.g. planar graphs

The Max-Cut Problem

Complexity Status

- NP-hard in general
- poly-time solvable graph classes exist, e.g. planar graphs

Applications

- theoretical physics (e.g. Ising spin glasses)
- VIA minimization
- network flow tasks
- quadratic 0-1 optimization
- ...

We focus on planar graphs.

Solution Approaches for Planar Graphs

- nonnegative edge weights :
  - Hadlock (1975), Dorfman, Orlova (1972)

Solution Approaches for Planar Graphs

- nonnegative edge weights :
  - Hadlock (1975), Dorfman, Orlova (1972)
- arbitrarily weighted
  - Barahona (in the 1980s)
    - poly-time solvability for graphs not contractible to K_5
  - Mutzel (1990)

Solution Approaches for Planar Graphs

- nonnegative edge weights :
  - Hadlock (1975), Dorfman, Orlova (1972)
- arbitrarily weighted
  - Barahona (in the 1980s)
    - poly-time solvability for graphs not contractible to K_5
  - Mutzel (1990)
  - Shih, Wu, and Kuo (1990)
    - minimum Eulerian graph in dual
    - fastest known algorithm - $O(n^{1.5} log n)$

Solution Approaches for Planar Graphs

- nonnegative edge weights :
  - Hadlock (1975), Dorfman, Orlova (1972)
- arbitrarily weighted
  - Barahona (in the 1980s)
    - poly-time solvability for graphs not contractible to K_5
  - Mutzel (1990)
  - Shih, Wu, and Kuo (1990)
    - minimum Eulerian graph in dual
    - fastest known algorithm - $O(n^{1.5} log n)$
  - Schraudolph, Kamenetsky (2008)

General Algorithmic Scheme

Input: embedding of a weighted planar graph G
Output: MAX-CUT $\delta(Q)$ of G

```
Construct some expanded dual graph G_D
Calculate matching M in G_D
Use M to generate a MAX-CUT δ(Q) of G
return δ(Q)
```

Outline

- The New Algorithm

Outline

- The New Algorithm
- Application in Physics — 2D planar Ising spin glass

Outline

- The New Algorithm
- Application in Physics — 2D planar Ising spin glass
- Results

Preliminaries

- omit self-loops (will never be cut-edges)
- merge multiple edges to one edge

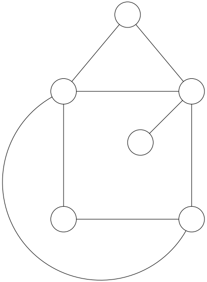
Let $G = (V, E)$ be

- simple
- connected
- planar
- real-weighted

The New Algorithm

Create Dual

- take dual edge weights from G

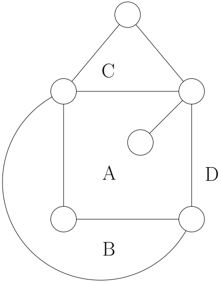


embedding of a simple planar graph (assume $w(e) = 1 \; \forall e \in E$)

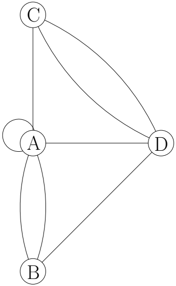
The New Algorithm

Create Dual

- take dual edge weights from G



embedding of a simple planar graph (assume $w(e) = 1 \; \forall e \in E$)

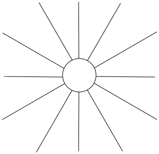


. . . and its dual graph.

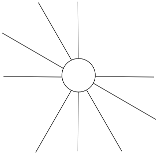
The New Algorithm

Split Nodes

Split each node $v \in V_D$ with $deg(v) > 4$ into $\lfloor (deg(v) - 1)/2 \rfloor$ nodes, connect them by a simple path. New edges receive weight zero.



even degree node



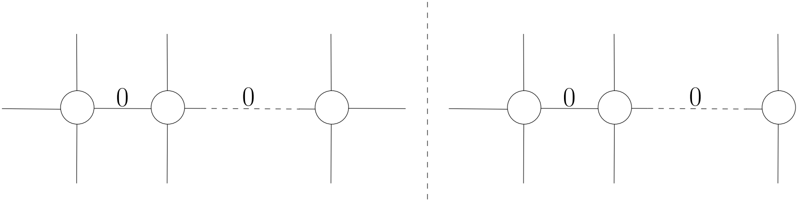
odd degree node

The New Algorithm

Split Nodes

Split each node $v \in V_D$ with $deg(v) > 4$ into $\lfloor (deg(v) - 1)/2 \rfloor$ nodes, connect them by a simple path. New edges receive weight zero.

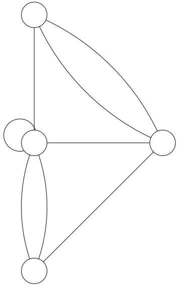
Result of a splitting operation



even degree node

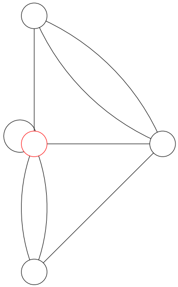
odd degree node

The New Algorithm

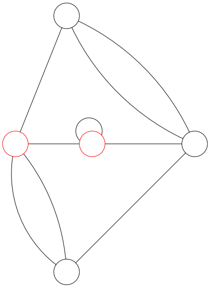


The dual graph.

The New Algorithm

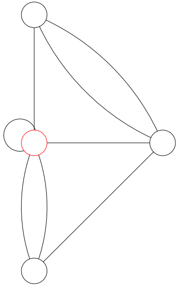


The dual graph.

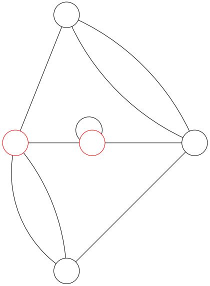


The split dual graph.

The New Algorithm



The dual graph.



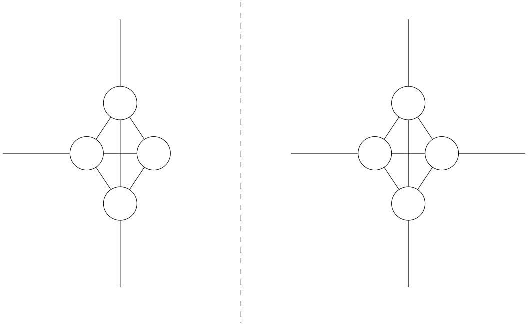
The split dual graph.

Each node now has degree three or four.

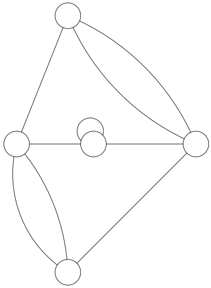
The New Algorithm

Expand Graph

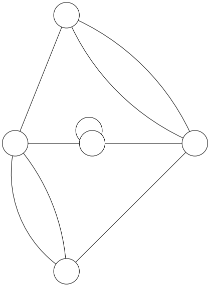
Each node $v \in V_D$ is expanded to a K_4 subgraph (Kasteleyn city). New edges receive weight zero.

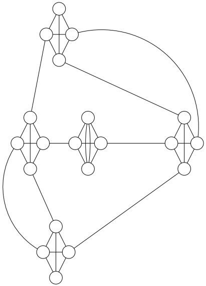


The New Algorithm



The New Algorithm

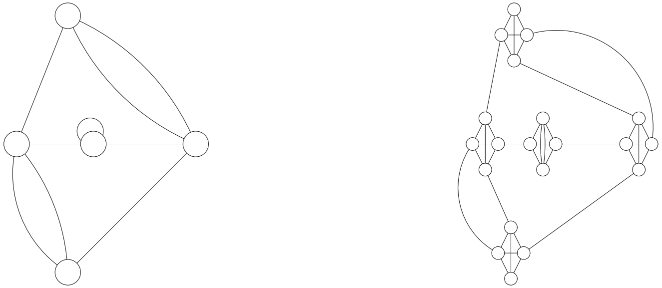




The New Algorithm

Match Edges

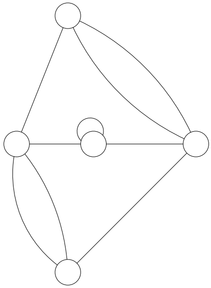
Determine minimum-weight perfect matching on the transformed dual graph.
(MAX-CUT: negate weights)

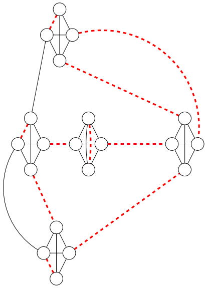


The New Algorithm

Match Edges

Determine minimum-weight perfect matching on the transformed dual graph.
(MAX-CUT: negate weights)

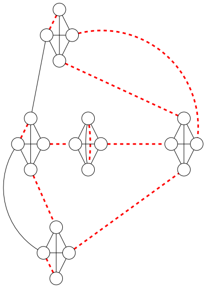




The New Algorithm

Shrink Nodes

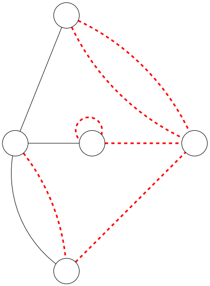
Shrink back all nodes (and edges) created in previous steps, and keep track of matched dual edges.



The New Algorithm

Shrink Nodes

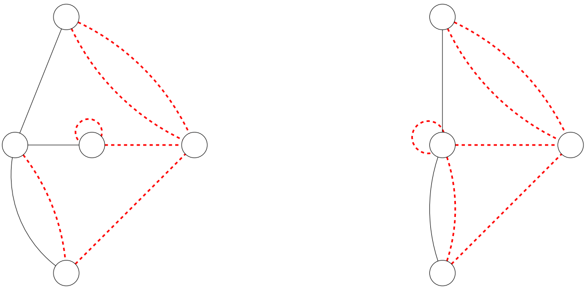
Shrink back all nodes (and edges) created in previous steps, and keep track of matched dual edges.



The New Algorithm

Shrink Nodes

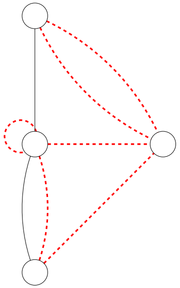
Shrink back all nodes (and edges) created in previous steps, and keep track of matched dual edges.

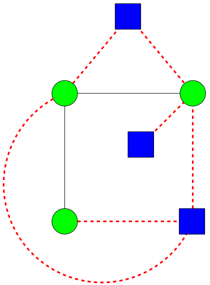


The New Algorithm

Cut

Eulerian subgraphs in dual $\Leftrightarrow$ cut G





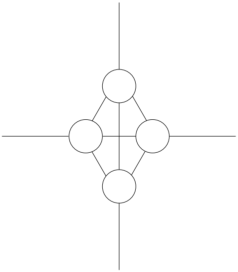
Correctness and Running Time

Correctness

$\Rightarrow$ optimal matching

$\Leftrightarrow$ optimal Eulerian subgraphs

$\Leftrightarrow$ optimal cut



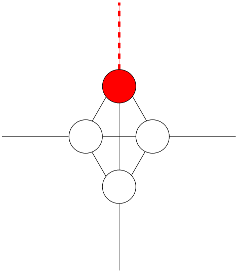
Correctness and Running Time

Correctness

$\Rightarrow$ optimal matching

$\Leftrightarrow$ optimal Eulerian subgraphs

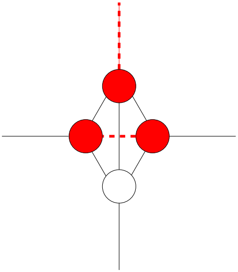
$\Leftrightarrow$ optimal cut



Correctness and Running Time

Correctness

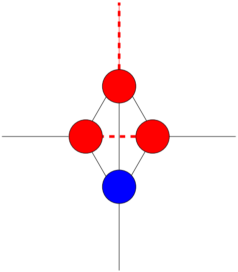
- $\Rightarrow$ optimal matching
- $\Leftrightarrow$ optimal Eulerian subgraphs
- $\Leftrightarrow$ optimal cut



Correctness and Running Time

Correctness

- $\Rightarrow$ optimal matching
- $\Leftrightarrow$ optimal Eulerian subgraphs
- $\Leftrightarrow$ optimal cut



Correctness and Running Time

transformation can be done in linear time

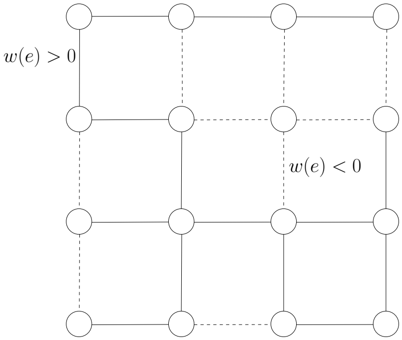
$\Rightarrow$ running time depends on the matching:
$O(n^{1.5} logn)$ (with Planar Separator Theorem)

Shih, Wu, and Kuo vis-à-vis the New Algorithm

	Shih, Wu, and Kuo (sharp bounds)	new algorithm (upper bounds)
$\|V\|$	$14n - 28$	$8n - 16$
$\|E\|$	$21n - 42$	$15n - 30$

expanded dual graph size

2D Planar Ising Spin Glasses



ground state energy

$$\min \; -\sum_{e \in E} w(e) + 2 \sum_{e \in \delta(Q)} w(e)$$

with $Q \subseteq V$.

Traditional Approaches

- exact algorithm by Bieche et al. (1980)
- exact algorithm by Barahona (1982)

also solve the problem via matching

Traditional Approaches

- exact algorithm by Bieche et al. (1980)
- exact algorithm by Barahona (1982)

also solve the problem via matching

Popular heuristic variant of the approach by Bieche et al.

- thin graph by deleting edges with weight $> c_{max}$

Traditional Approaches

- exact algorithm by Bieche et al. (1980)
- exact algorithm by Barahona (1982)

also solve the problem via matching

Popular heuristic variant of the approach by Bieche et al.

- thin graph by deleting edges with weight $> c_{max}$
- often yields high-quality heuristic
  - Palmer, and Adler (1999) 1801^2 nodes
  - Hartmann, and Young (2001) 480^2 nodes

Results with New Algorithm

2D planar Ising spin glasses

$\lvert V_{grid} \rvert$	average time	memory
100^2	<1	159 MB
150^2	2	159 MB
250^2	<10	163 MB
500^2	110	333 MB
1000^2	1200	995 MB
1500^2	5280	2.05 GB
2000^2	($\sim$ 4h) 14524	3.57 GB
3000^2	($\sim$ 17h) 61167	7.83 GB

- MIN-CUT calculations (using Blossom IV)
- uniform distributed $\pm J$ edge weights
- running times in seconds

Results with New Algorithm

TSPLIB - Delaunay triangulated point sets

instance name	$|V|$	$|E|$	time (sec)
pla85900	85900	257604	($\sim$ 2.8h) 10248.70
pla33810	33810	101367	390.50
usa13509	13509	40503	169.73
brd14051	14051	42128	140.49
d18512	18512	55510	86.50
pla7397	7397	21865	15.11
rl11849	11849	35532	9.87
rl5934	5934	17770	4.56
fnl4461	4461	13359	3.21
rl5915	5915	17728	2.84

- MAX-CUT calculations (using Blossom IV)
- euclidean distances as edge weights

Results with New Algorithm

USA road networks - DIMACS

instance ($\|V\|$, $\|E\|$)	time (sec)
USA-road-d.FLA (1,070,376, 2,712,798)	394937 ($\sim$ 4.5d)
USA-road-d.NW (1,207,945, 2,840,208)	168239 ($\sim$ 2d)
USA-road-d.NY (264,346, 793,002)	117997 ($\sim$ 1.3d)
USA-road-d.BAY (321,270, 800,172)	90486 ($\sim$ 1d)
USA-road-d.COL (435,666, 1,057,066)	32227 ($\sim$ 0.3d)

- MAX-CUT calculations (using Blossom IV)
- euclidean distances as edge weights

Thank you very much for your attention!