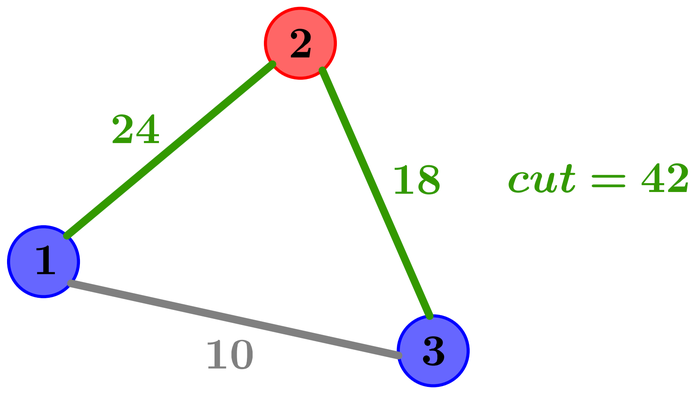
High-Performance Computing in Combinatorial Optimization

Elisabeth Gaar[4], Timotej Hrga[1], Borut Lužar[3], Janez Povh[1,2], Alen Vegi Kalamar, Angelika Wiegele[4]

University of Ljubljana, Faculty of Mechanical Engineering[1], Institute of Mathematics, Physics and Mechanics[2], Faculty of information studies in Novo mesto[3], Alpen-Adria-Universität Klagenfurt[4]

Max-Cut problem

Divide the vertices of the graph G into red and blue such that the sum on edges connecting vertices with different color is maximal.



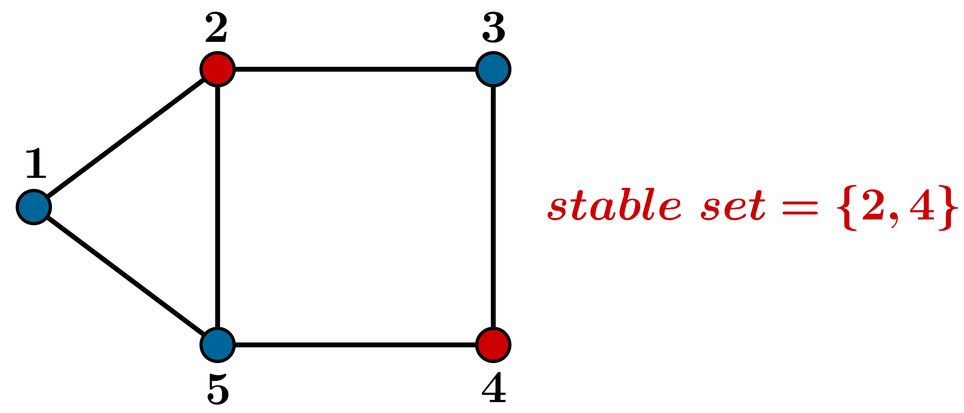
It can be posed as a binary quadratic problem

Max-Cut

$$\begin{aligned} \text{maximize} \quad & \frac{1}{4} x^T L x \\ \text{subject to} \quad & x \in \{-1, 1\}^n \end{aligned}$$

Stable Set problem

Find the maximum number of vertices such that no two are connected by an edge.



It can be expressed by an integer linear program:

Maximum Stable Set

$$\begin{aligned} \text{maximize} \quad & \sum_{v \in V} x_v \\ \text{subject to} \quad & x_v + x_u \leq 1 \quad \text{for each edge } \{u, v\} \in E, \\ & x_v \in \{0, 1\} \quad \text{for all } v \in V \end{aligned}$$

Branch & Bound parallel algorithm

MPI (Message Passing Interface) has been used to develop parallel solvers which use a Master/Slave scheme:

Algorithm for MASTER

input: graph G
output: optimal solution x
initialize:
  compute initial solution value z_{bk} and vector x_{bk};
  initialize problem list $\mathcal{Q}$;
  send initial problem to first free slave;
while *there are working slaves and* $\mathcal{Q} \neq \emptyset$ **do**
  receive message from slave;
  if *BRANCH* **then**
    generate new subproblems and put them in $\mathcal{Q}$;
  else if *PRUNE* **then**
    mark slave as free;
  else if *NEW_VALUE* **then**
    slave found better solution;
    update z_{bk} and x_{bk};
  end
  send subproblems to free slaves;
end
send message FINISH to slaves;
return x_{bk}

Algorithm for SLAVE

receive message from master;
if *FINISH* **then**
  stop;
else
  receive subproblem and best value from master;
  compute lower bound z_{bk};
  compute upper bound z_{ub};
  if $z_{bk} >$ *best value* **then**
    send message NEW_VALUE to master;
    update best value;
  end
  if $z_{ub} <$ *best value* **then**
    send message PRUNE to master;
  else
    send BRANCH to master;
  end
end

Preliminary numerical results

We present the numerical results of the parallel B&B algorithm for Max-Cut tested on graph instances from the *BiqMac Library* and of the parallel B&B algorithm for Stable set tested on *random graphs with different densities*.

The following table summarizes the execution for both serial and parallel implementations. The results show the number of processors used (#proc) and the total time spent in seconds.

graph	serial	parallel		
	#proc = 1	#proc = 24	#proc = 48	#proc = 96
	Max-Cut			
pm1d_100.0	7745	391	201	110
pm1s_100.1	404	35	34	34
pw01_100.9	383	36	33	33
g05_100.0	3633	189	103	63
w05_100.0	3799	198	108	69
	Stable set			
g150_density5	822	99	50	27
g150_density10	382	152	73	39
g150_density20	414	52	52	52
g150_density30	407	132	132	132
g150_density50	342	322	322	322
g200_density10	more than 2 hours	1347	647	300

BiqBin solver

1. **Extend** BiqMac solver for Max-Cut developed by Alpen-Adria-Universität Klagenfurt to solve

Binary Quadratic Problem

$$\begin{aligned} \text{maximize} \quad & x^T Q x + l^T x + c \\ \text{subject to} \quad & Ax = b, \\ & x \in \{0, 1\}^n \end{aligned}$$

2. **Improve** performance by parallelization.

3. **Make** new open-source solver **BiqBin** available online which will run on the *high-performance computer* owned by the Faculty of Mechanical Engineering in Ljubljana.

BiqBin — High-Performance Solver for Binary Quadratic Problems

Financed by

• Austrian Science Fund (FWF): project number I 3199-N31

FWF Der Wissenschaftsfonds.

• Slovenian Research Agency (ARRS): project number N1-0057

Slovenian Research Agency