Exact Algorithms for the Vertex Coloring Problem

Branch and Bound:

Algorithm DSATUR, Brélaz (*Comm. ACM, 1979*);

Improvement of DSATUR: Sewell (*2nd DIMACS Implementation Challenge, 1993*).

Branch and Cut:

Méndez-Díaz, Zabala (*Disc. Appl. Math. 2006, 2008*).

Branch-and-Price:

Mehrotra, Trick (*INFORMS J. on. Computing, 1996*),

Malaguti, Monaci, T. (*Discrete Optimization, 2010*).

Gualandi, Malucelli (*INFORMS J. on. Computing, 2012*),

Maximal Clique

- **A *clique* *K* of a graph G is a complete subgraph of G.**
- **A clique is *maximal* if no vertex can be added still having a clique.**
- **The cardinality of any (maximal) clique of graph *G* represents a *Lower Bound* for the problem.**

- **The determination of the *Maximum Cardinality* (or *Maximum Weight*) *Clique* is NP-Hard.**

ILP models for VCP: Model VCP-ASSIGN

- Binary variables: $x_{ih} = \begin{cases} 1 \text{ if vertex } \boldsymbol{i} \text{ has color } \boldsymbol{h} \\ 0 \text{ otherwise} \end{cases}$ $\quad i = 1, \ldots, n \quad h = 1, \ldots, n$

  $y_h = \begin{cases} 1 \text{ if color } \boldsymbol{h} \text{ is used} \\ 0 \text{ otherwise} \end{cases}$ $\quad h = 1, \ldots, n$

$$\min \sum_{h=1}^{n} y_h \qquad (1)$$

$$\sum_{h=1}^{n} x_{ih} = 1 \qquad i = 1,..,n \qquad (2)$$

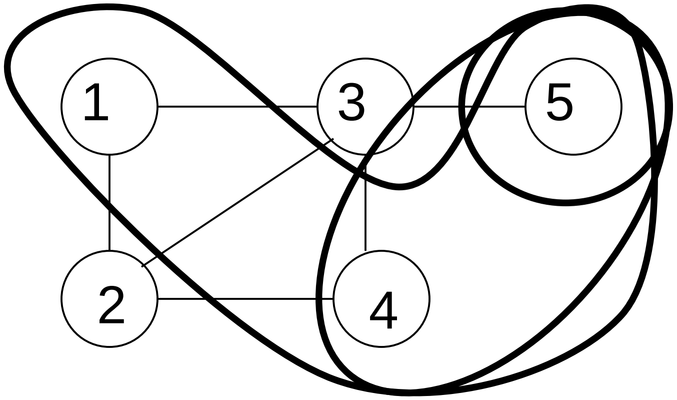
$$x_{ih} + x_{jh} \le y_h \qquad \forall i, j : (i, j) \in E \quad h = 1,\ldots,n \qquad (3)$$

$$x_{i,h} \in \{0,1\} \qquad i = 1,\ldots,n \quad h = 1,\ldots,n$$

$$y_h \in \{0,1\} \qquad h = 1,\ldots,n \qquad (4)$$

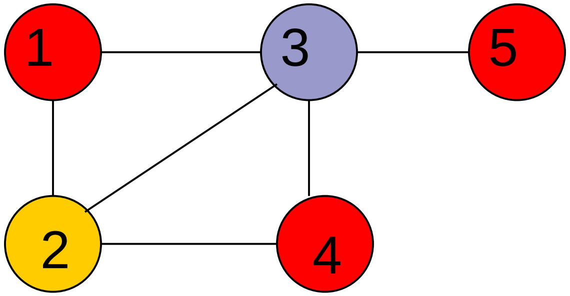
Independent Sets

- An ***Independent Set*** (or ***Stable Set***) of ***G = (V, E)*** is a subset of ***V*** such that there is no edge in ***E*** connecting a pair of vertices.
- It is ***maximal*** if no vertex can be added still having an independent set.



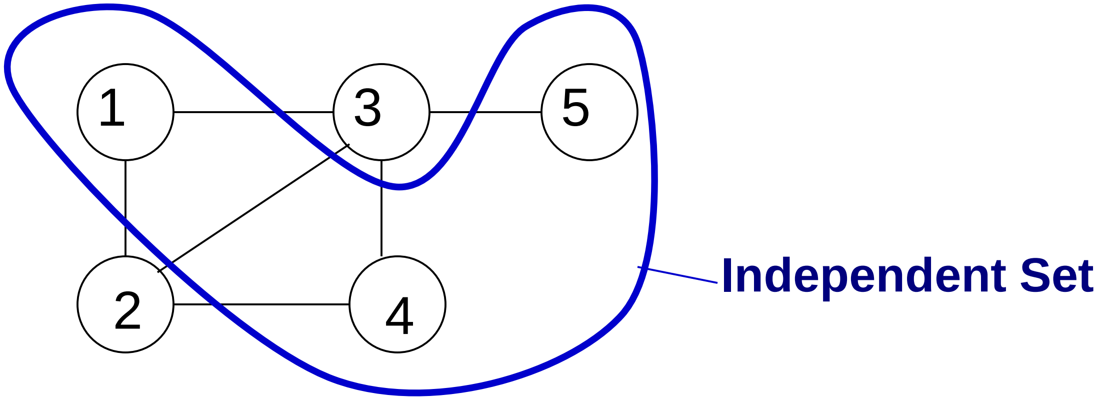
For VCP: all the vertices of an independent set can have the same color

Feasible coloring **->** **partitioning of the graph into independent sets.**

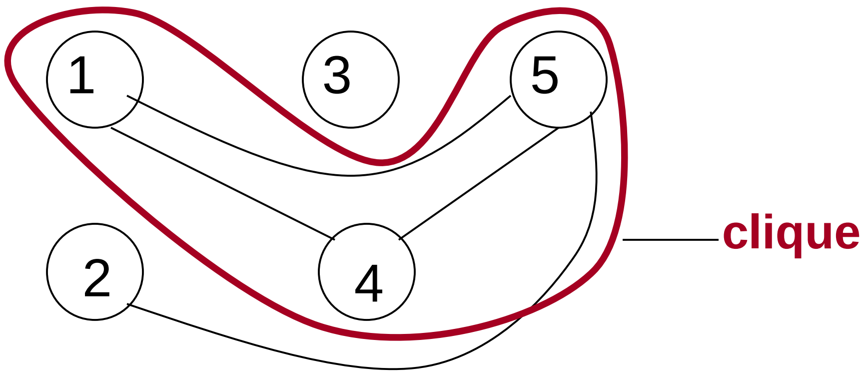


Independent Sets and Cliques

- **Given a graph $G = (V, E)$**



Its "complementary graph" $\overline{G} = (V, \overline{E})$, with $\overline{E} = \{(i, j): (i, j) \notin E\}$



independent set of G → clique of $\overline{G}$ (and viceversa)

clique of G → independent set of $\overline{G}$ (and viceversa)

Set Covering Formulation SC -VCP

$$\min \sum_{s \in S} x_s \qquad (1)$$

s.t.

$$\sum_{s: i \in s} x_s \geq 1 \qquad \forall i \in V \qquad (2)$$

$$x_s \in \{0,1\} \qquad \forall s \in S \qquad (3)$$

- ***S*** **can be defined as the family of all the *maximal Independent Sets* (or *Stable Sets*) of graph *G*.**
- **The *LP Relaxation* of this formulation leads to *tight lower bounds,* and *symmetry* in the solution is *avoided*, but the *number of* maximal independent sets (i.e. the number of "variables" or "columns") can grow *exponentially* with the number of vertices *n.***

Set Covering Formulation SC-VCP
Branch-and-Price Algorithms

SC-VCP: Master Problem

- ***LP Relaxation of SC-VCP:*** **exponentially many variables *(columns, independent sets) .***
- ***Column Generation procedure:***

Solve the *LP Relaxation* of the SC-VCP formulation by considering a subset of independent sets (columns): *Restricted Master Problem (RMP);*

Detect possible negative "reduced cost" columns by solving the corresponding "*Pricing Problem*", add them to the *RMP* and iterate.

Pricing Problem

- c_i **is the *optimal dual variable* associated with the *i*-th "covering constraint" in the SC-VCP formulation (*weight of vertex i*).**

The Pricing Problem requires the solution of a *Maximum Weighted Independent Set Problem (MWISP)* (NP-Hard).

- $y_i = 1\{$ **if vertex *i* is in the independent set}, = 0 otherwise**

$$\max \sum_{i=1}^{n} c_i \, y_i$$

$$y_i + y_j \leq 1 \qquad \forall i, j : (i, j) \in E$$

$$y_i \in \{0,1\} \qquad i = 1,\ldots,n$$

Pricing Problem (MWISP) (2)

- c_i **is the optimal dual variable associated with the *i*-th "covering constraint" in the SC-VCP formulation.**
- $y_i = 1\{$ **if vertex *i* is in the independent set}, = 0 otherwise**

$$\max \sum_{i=1}^{n} c_i \, y_i$$

$$y_i + y_j \le 1 \qquad \forall i, j : (i, j) \in E$$

$$y_i \in \{0,1\} \qquad i = 1, \ldots, n$$

If the optimal solution value (global weight) is greater than 1, then an independent set (column, variable) with negative reduced cost has been found. Add it to the RMP.

Branch-and-Price Algorithm 1 (Gualandi, Malucelli; *INFORMS J. C. 2012*)

- ***Column Generation procedure:***

  detection of possible negative reduced cost columns by alternatively solving:

1) Exact Algorithm CLIQUER (maximum weighted clique problem; Ostergard, *Discr. Appl. Math. 2002*).

2) Heuristic Algorithm QUALEX-MS (maximum weighted clique problem; Busygin, *Discr. Appl. Math. 2006*).

2) Constraint Programming Algorithm (weighted version of the max-clique problem algorithm proposed by Fahle, *Proc. Ann. Eur. Symp. Algorithms, Springer, 2002).*

Branch-and-Price Algorithm 2

(Malaguti, Monaci, T.; *Discrete Optimization 2010*)

- ***Column Generation procedure:***

  detect possible negative reduced cost columns by solving *MWISP*, by using:

1) a Tabu Search heuristic algorithm (*TS*) which produces maximum weighted independent sets;

2) an ILP Solver (CPLEX 10.2), if the algorithm *TS* fails in finding negative reduced cost columns.

Branch-and-Cut Algorithm
Méndez-Díaz, Zabala (*Disc. Appl. Math. 2006, 2008*)

- ***Improvement of the VCP-ASSIGN Formulation by adding new valid inequalities.***

- ***Initialization Phase:***
- ***Preprocessing procedure*** **to reduce the number of vertices to be considered.**
- ***Initial Upper Bound*** **computed through the execution of algorithm DSATUR with a short time limit (5 seconds).**
- ***Initial Lower Bound*** **computed by finding a maximal clique through a greedy algorithm.**

- ***New Branching Rules.***

Computational Results for the Exact Approaches

- ***Algorithm DSATUR*: Brélaz (*Comm. ACM, 1979*),**
  Improved by Sewell (*2nd DIMACS Implem. Challenge 1993*)
  Implemented by Mehrotra, Trick (*INFORMS J. on C., 1996*)
- ***Branch and Cut Algorithm BC-COL (with the stronger lower bounding procedures)*:**
  Méndez-DÍaz, Zabala (*Disc. Appl. Math. 2006, 2008*)
- ***Branch-and-Price Algorithm G-M*: Gualandi, Malucelli (*INFORMS J. on. Computing, 2012*).**
- ***Branch-and-Price Algorithm M-M-T*: Malaguti, Monaci, T. (*Discrete Optimization, 2010*).**
- **Comparable CPU "time limit" for each instance (by considering the different computer speeds).**

DIMACS Benchmark Instances

Johnson, Trick, 2nd DIMACS Implementation Challenge 1993

- ***DIMACS benchmark graph instances*** **consist of several graph classes used for evaluating the performance of VCP algorithms:**

  - random graphs: DSJC_*n.x*;
  - geometric random graphs: DSJR_*n.x;* r_*n.x*;
  - quasi-random graphs: flat_*n.x*;
  - artificial graphs: le_*n.x*; latin_square_*10*;
  Queen_*rn.rn; myciel_k*
  - real-world application-related graphs.

VCP: Exact Algorithms

	Mendez-Diaz, Zabala (2006, 2008)		Gualandi-Malucelli (2012)	M-M-T (2010)
	BC-COL	DSATUR		
common instances (w.r.t. M-M-T)	66	66	39	-
proven optimal solutions	8	1	15	29 (19)
only algorithm	1	0	2	13 (6)
Lower Bound				
LB > LB (MMT)	9	1	2	-
LB = LB (MMT)	43	23	27	-
LB < LB (MMT)	14	42	10	-
global LB gap w.r.t. MMT	92	326	*	-
Upper Bound				
UB < UB (MMT)	0	0	0	-
UB = UB (MMT)	24	18	27	-
UB > UB (MMT)	42	48	12	-
global UB gap w.r.t. MMT	333	304	*	-