Brief Tutorial on Column Generation Algorithms for the Vertex Coloring Problem

Fabio Furini

SPOC20

Outline

Dantzig-Wolfe reformulation

"Divide et Impera" in Graph Coloring

- Break up complex problems into a series of easier problems, solved in cascade!
- Dantzig-Wolfe reformulation of the (weak) compact Integer Formulations.
- Hard Combinatorial Optimization Problems, once decomposed and reformulated, become easier to tackle.
- Find effective decompositions and reformulations!
  - Vertex Coloring Problem
  - Max Coloring Problem
  - Partition Coloring Problem
  - Sum Coloring Problem
  - ...

Automatic Dantzig-Wolfe Reformulation: Overview

Potentially, every MIP model is amenable to DWR, even if its structure is not known in advance (from the modeler or from other sources).

We need to detect a structure algorithmically:

(i) which constraints of the MIP (if any) to keep in the master problem;
(ii) the number of blocks k
(iii) how to assign the remaining constraints to the different blocks.

We need to partition the set of the original constraints into one subset representing the master and several subsets representing the blocks.

- Permutation of the variables and the constraints to get an:
  - Arrowead Form
- Once the decomposition is chosen $\Rightarrow$ Branch-and-Price Algorithm

GCG - Generic Column Generation

- GCG is a generic branch-cut-and-price solver for mixed integer programs.
- It is based on the branch-and-cut-and-price framework of SCIP and is also part of the SCIP Optimization Suite.
- GCG is developed jointly by RWTH Aachen and Zuse-Institute Berlin.

[1] M. Bergner, A. Caprara, A. Ceselli, F. F., M. Lübbecke, E. Malaguti, and E. Traversi.
Automatic dantzig–wolfe reformulation of mixed integer programs.
Mathematical Programming, 149(1):391–424, 2015.

Does it work? I've tried to solve a MIPLIB instance . . .

noswot

Name	noswot
Download	noswot.mps.gz
Solution	noswot.sol.gz
Set Membership	Benchmark Tree
Problem Status	Easy
Problem Feasibility	Feasible
Originator/Contributor	J. Gregory, L. Schrage
Rows	182
Cols	128

Solving noswot with CPLEX

```
 6605742 1239766        cutoff              -41.0000      -43.0000 50153618    4.88%
 6755003 1262076      -42.9461    19        -41.0000      -43.0000 51335942    4.88%
 6896391 1280590      -43.0000    16        -41.0000      -43.0000 52543834    4.88%
Elapsed time = 538.58 sec. (168354.79 ticks, tree = 311.16 MB, solutions = 5)
Nodefile size = 183.98 MB (115.07 MB after compression)
 7045357 1301707      -42.3064    17        -41.0000      -43.0000 53726797    4.88%
 7190566 1322648        cutoff              -41.0000      -43.0000 54954083    4.88%
 7330140 1343041      -42.1052    10        -41.0000      -43.0000 56158420    4.88%
 7469709 1364126        cutoff              -41.0000      -43.0000 57393127    4.88%
 7604916 1384906        cutoff              -41.0000      -43.0000 58651552    4.88%
 7737995 1406384      -43.0000    30        -41.0000      -43.0000 59883520    4.88%
 7869081 1428551      -42.2859    13        -41.0000      -43.0000 61156619    4.88%
 8001538 1450426      -43.0000    13        -41.0000      -43.0000 62448722    4.88%
 8145241 1471495      -42.1434    25        -41.0000      -43.0000 63632808    4.88%
 8286801 1488295        cutoff              -41.0000      -43.0000 64889238    4.88%
Elapsed time = 653.42 sec. (206514.23 ticks, tree = 359.04 MB, solutions = 5)
Nodefile size = 230.98 MB (143.38 MB after compression)
 8425279 1504664      -42.6017    27        -41.0000      -43.0000 66138358    4.88%
^C
Cover cuts applied:  65
Implied bound cuts applied:  20
Flow cuts applied:  21
Mixed integer rounding cuts applied:  34
Zero-half cuts applied:  1
Gomory fractional cuts applied:  8

Root node processing (before b&c):
  Real time             =    0.02 sec. (10.71 ticks)
Parallel b&c, 8 threads:
  Real time             =  675.37 sec. (214017.86 ticks)
  Sync time (average)   =    1.74 sec.
  Wait time (average)   =    0.95 sec.
                          ------------
Total (root+branch&cut) =  675.39 sec. (214028.58 ticks)

Solution pool: 5 solutions saved.

MIP - Aborted, integer feasible:  Objective = -4.1000000000e+01
Current MIP best bound = -4.3000000000e+01 (gap = 2, 4.88%)
Solution time =  675.39 sec.  Iterations = 67328591  Nodes = 8560732 (1523579)
Deterministic time = 214030.15 ticks  (316.90 ticks/sec)

CPLEX>
```

Solving noswot with SCIP and GCG

- CPLEX in more that 600 seconds explored 8286801 nodes, 1488265 to be explored. Hundreds of cuts are generated but . . . still 4.88% of optimaltiy gap!

```
Presolving Time: 0.00
Detecting purely block diagonal structure: not found.
Detecting set partitioning master structure: found 5 blocks.
Chosen decomposition with 5 blocks of type bordered.

Discretization with continuous variables is currently not supported. The parameter setting will be ignored.

  time | node  | left  |LP iter|MLP iter|LP it/n| mem |mdpt |ovars|mvars|ocons|mcons|mcuts|confs|  dualbound   | primalbound  |  gap
   0.0s|     1 |     0 |     0 |      5 |     - |1775k|   0 | 120 |   5 | 172 |  12 |   0 |   0 |-4.300000e+01 |      --      |   Inf
  time | node  | left  |LP iter|MLP iter|LP it/n| mem |mdpt |ovars|mvars|ocons|mcons|mcuts|confs|  dualbound   | primalbound  |  gap
*  0.0s|     1 |     0 |     0 |     10 |     - |1788k|   0 | 120 |  13 | 172 |  12 |   0 |   0 |-4.300000e+01 |-5.000000e+00 | 760.00%
*P 0.0s|     1 |     0 |     0 |     10 |     - |1788k|   0 | 120 |  13 | 172 |  12 |   0 |   0 |-4.300000e+01 |-5.000000e+00 | 760.00%
Starting reduced cost pricing...
*r 0.1s|     1 |     0 |     0 |     17 |     - |1918k|   0 | 120 |  70 | 172 |  12 |   0 |   0 |-4.300000e+01 |-3.800000e+01 |  13.16%
*r 0.2s|     1 |     0 |     0 |     20 |     - |1928k|   0 | 120 |  75 | 172 |  12 |   0 |   0 |-4.300000e+01 |-4.100000e+01 |   4.88%
*r 1.0s|     1 |     0 |     0 |     23 |     - |1938k|   0 | 120 |  80 | 172 |  12 |   0 |   0 |-4.300000e+01 |-4.100000e+01 |   4.88%
   1.0s|     1 |     0 |     0 |     23 |     - |1938k|   0 | 120 |  80 | 172 |  12 |   0 |   0 |-4.100000e+01 |-4.100000e+01 |   0.00%

SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 0.98
Solving Nodes      : 1
Primal Bound       : -4.10000000000000e+01 (2 solutions)
Dual Bound         : -4.10000000000000e+01
Gap                : 0.00 %

GCG>
```

- GCG solved the instance in less than 1 second at the root node!

The Vertex Coloring Problem (VCP)

References

[1] A. Mehrotra, and M. Trick.
An exact approach for the vertex coloring problem.
INFORMS Journal on Computing, 8(4):344–354, 1996.

[2] E. Malaguti, M. Monaci, and P. Toth.
An exact approach for the vertex coloring problem.
Discrete Optim, 8:174–190, 2011.

[3] S. Gualandi and F. Malucelli.
Exact solution of graph coloring problems via constraint programming and column generation.
INFORMS Journal on Computing, 24(1):81–100, 2012.

[4] S. Held, W. Cook, and E. Sewell.
Maximum-weight stable sets and safe lower bounds for graph coloring.
Mathematical Programming Computation, 4(4):363–381, 2012.
→ https://github.com/heldstephan/exactcolors

[5] D. Morrison, E. Sewell, and S. Jacobson.
Solving the Pricing Problem in a Branch-and-Price Algorithm for Graph Coloring Using Zero-Suppressed Binary Decision Diagrams.
INFORMS Journal on Computing, 28(1): 67–82, 2016.

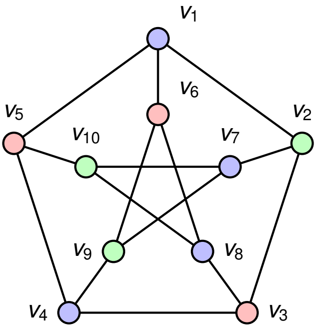
...

The Vertex Coloring Problem (VCP)

Given a graph $G = (V, E)$, the VCP asks for a partition of the vertex set

$$C = \{S_1, S_2, \ldots, S_k\},$$

with the minimum number of colors, such that vertices linked by an edge receive different colors.



$$S_1 = \{v_1, v_4, v_7, v_8\}$$

$$S_2 = \{v_2, v_9, v_{10}\}$$

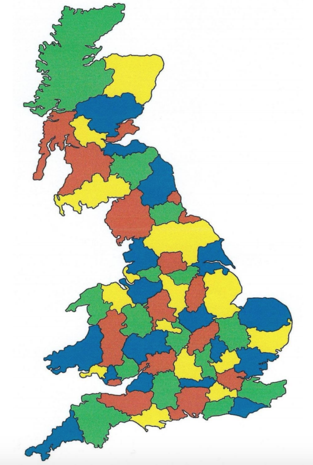
$$S_3 = \{v_3, v_6, v_5\}$$

chromatic number $\rightarrow \chi(G) = 3$

- A coloration C is a partition a of vertices into stables sets of G
- Clique number $\rightarrow \omega(G) = 2 \leq \chi(G)$

Origins and applications

- Colour the map of *UK*, in such a way that no two *counties touching* with a common stretch of boundary are given the *same colour*, by using the smallest number of colours.
- The *Four Color Conjecture* was proposed by *Francis Guthrie* in 1852



Theorem (Appel and Haken (1976))

Given any separation of a plane into contiguous regions, producing a figure called a map, no more than four colours are required to color the regions of the map so that no two adjacent regions have the same color.

- It was the first major theorem to be proved using a computer.

Origins and applications

1. problem (i): assign frequencies to broadcast stations in such a way that:
   - interfering stations use different frequencies;
   - the total number of used frequencies is minimized.

2. problem (ii): assign exams to time slots in such a way:
   - every student can do the exams of the courses he is taking;
   - the total number of used time slots is minimized.

3. problem (iii): assign platforms to trains in such a way that:
   - if the arrival times overlap, the trains cannot use the same platform;
   - the total number of used platforms is minimized.

How difficult is the VCP in practice?

Theorem (Garey and Johnson (1979))

The Vertex Coloring Problem is NP-Hard.

- Some NP-Hard problems can be solved to optimality for instances of reasonable size:
  - TSP — thousands of vertices (Branch-and-Cut Algorithms)
  - BPP — up to 1000 items (Branch-and-Price Algorithms)
  - VRP — up to 200 customers (Branch-and-Price Algorithms)
- VCP is really difficult from a practical viewpoint: it cannot be consistently solved to optimality for graphs with more than $\approx$ 150 vertices.
- The state-of-the-art algorithms for the VCP are based on Column Generation!

A natural compact Integer Linear Programming (ILP) Formulation

Two sets of binary variables

$$y_c = \begin{cases} 1 & \text{if color } c \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{vc} = \begin{cases} 1 & \text{if vertex } v \text{ has color } c \\ 0 & \text{otherwise,} \end{cases}$$

- given an an upper bound $m \leq n$ on the chromatic number

ILP Formulation

$$\min_{x,y \in \{0,1\}} \sum_{c=1}^{m} y_c$$

$$\sum_{c=1}^{m} x_{vc} = 1 \qquad v \in V$$

$$x_{vc} + x_{uc} \leq y_c \qquad vu \in E$$

$$c = 1, \ldots, m$$

Very weak and symmetric formulation!

A natural compact Integer Linear Programming (ILP) Formulation

Two sets of binary variables

$$y_c = \begin{cases} 1 & \text{if color } c \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{vc} = \begin{cases} 1 & \text{if vertex } v \text{ has color } c \\ 0 & \text{otherwise,} \end{cases}$$

- given an an upper bound $m \leq n$ on the chromatic number

ILP Formulation

$$\min_{x,y \in \{0,1\}} \sum_{c=1}^{m} y_c$$

$$\sum_{c=1}^{m} x_{vc} = 1 \qquad v \in V$$

$$x_{vc} + x_{uc} \leq y_c \qquad vu \in E$$

$$c = 1, \ldots, m$$

Very weak and symmetric formulation!

The Linear Programming Relaxation has optimal solution value 2

$$y_1 = 1, y_2 = 1$$

$$y_c = 0 \qquad c = 3, \dots, m$$

$$x_{v1} = \frac{1}{2}, x_{v2} = \frac{1}{2} \qquad v \in V$$

$$x_{vc} = 0 \qquad v \in V, c = 3, \dots, m$$

$$\underbrace{\sum_{c=1}^{m} x_{vc}}_{=1} + \underbrace{\sum_{c=1}^{m} x_{uc}}_{=1} \leq \sum_{c=1}^{m} y_c \quad \rightarrow \quad \sum_{c=1}^{m} y_c \geq 2$$

Every (fract.) solution with $\alpha \leq n$ colors has $\binom{n}{\alpha} \alpha!$ equivalent solutions!

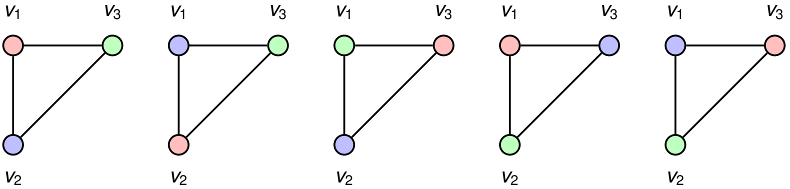
$\alpha = n = 3 \rightarrow 5$ equivalent solutions

The Linear Programming Relaxation has optimal solution value 2

$$y_1 = 1, y_2 = 1$$
$$y_c = 0 \qquad c = 3, \dots, m$$
$$x_{v1} = \frac{1}{2}, x_{v2} = \frac{1}{2} \qquad v \in V$$
$$x_{vc} = 0 \qquad v \in V, c = 3, \dots, m$$

$$\underbrace{\sum_{c=1}^{m} x_{vc}}_{=1} + \underbrace{\sum_{c=1}^{m} x_{uc}}_{=1} \leq \sum_{c=1}^{m} y_c \quad \rightarrow \quad \sum_{c=1}^{m} y_c \geq 2$$

Every (fract.) solution with $\alpha \leq n$ colors has $\binom{n}{\alpha}\, \alpha!$ equivalent solutions!



$\alpha = n = 3 \rightarrow 5$ equivalent solutions

Dantzig-Wolfe Reformulation

Minkowski-Weyl Theorem. Every polyhedron can be represented:

- by outer descriptions (intersection of finitely many affine halfspaces)
- by inner descriptions (Minkowski sum of a polytope and a finitely generated cone)

So a polyhedron $P = \{x : Ax \leq b\}$ can be then expressed as:

$$P = \left\{ x : x = \sum_p p\, \lambda_p + \sum_r r\, \mu_r,\ \sum_p \lambda_p = 1,\ \lambda_p \geq 0,\ \mu_r \geq 0 \right\}$$

where p are the extreme points and r are the extreme rays of P.

For the VCP, we reformulate the following sets of constraints (polytope):

$$P_c = \left\{ x, y \in \{0,1\} : \quad x_{vc} + x_{uc} \leq y_c, vu \in E \right\} \qquad c = 1, \ldots, m$$

$\rightarrow$ if $y_c = 1$, it is the stable set polytope!

Dantzig-Wolfe Reformulation

Minkowski-Weyl Theorem. Every polyhedron can be represented:

- by outer descriptions (intersection of finitely many affine halfspaces)
- by inner descriptions (Minkowski sum of a polytope and a finitely generated cone)

So a polyhedron $P = \{x : Ax \leq b\}$ can be then expressed as:

$$P = \left\{ x : x = \sum_p p\, \lambda_p + \sum_r r\, \mu_r,\ \sum_p \lambda_p = 1,\ \lambda_p \geq 0,\ \mu_r \geq 0 \right\}$$

where p are the extreme points and r are the extreme rays of P.

For the VCP, we reformulate the following sets of constraints (polytope):

$$P_c = \left\{ x, y \in \{0,1\} : \quad x_{vc} + x_{uc} \leq y_c, vu \in E \right\} \qquad c = 1, \ldots, m$$

$\rightarrow$ if $y_c = 1$, it is the stable set polytope!

Extreme points $\longrightarrow p \in EP$

$$\bar{x}^p_{vc} = \begin{cases} 1 & \text{if vertex } v \text{ has color } c \text{ in } p \\ 0 & \text{otherwise} \end{cases} \qquad \bar{y}^p_i = \begin{cases} 1 & \text{if color } c \text{ is used in } p \\ 0 & \text{otherwise} \end{cases}$$

Relation between the original variables and the new ones:

$$x_{vc} = \sum_{p \in EP} \bar{x}^p_{vc} \, \lambda^c_p \qquad v \in V, c = 1, \dots, m$$

$$y_c = \sum_{p \in EP} \bar{y}^p_c \, \lambda^c_p \qquad c = 1, \dots, m$$

Example of extreme points with 8 vertices (color last position):

- vertices 1, 3, 8 and color used

$$[1, 0, 1, 0, 0, 0, 0, 1|1]$$

- color not used

$$[0, 0, 0, 0, 0, 0, 0, 0|0]$$

Extreme points $\longrightarrow p \in EP$

$$\bar{x}^p_{vc} = \begin{cases} 1 & \text{if vertex } v \text{ has color } c \text{ in } p \\ 0 & \text{otherwise} \end{cases} \qquad \bar{y}^p_i = \begin{cases} 1 & \text{if color } c \text{ is used in } p \\ 0 & \text{otherwise} \end{cases}$$

Relation between the original variables and the new ones:

$$x_{vc} = \sum_{p \in EP} \bar{x}^p_{vc} \lambda^c_p \qquad v \in V, c = 1, \ldots, m$$

$$y_c = \sum_{p \in EP} \bar{y}^p_c \lambda^c_p \qquad c = 1, \ldots, m$$

Example of extreme points with 8 vertices (color last position):

- vertices 1, 3, 8 and color used

$$[1, 0, 1, 0, 0, 0, 0, 1 | 1]$$

- color not used

$$[0, 0, 0, 0, 0, 0, 0, 0 | 0]$$

Exponential-size formulation

Now by using inner description of the "stable set" constraints we obtain:

$$\min_{\lambda \in \{0,1\}} \quad \sum_{c=1}^{m} \sum_{p \in EP} \bar{y}_c^p \, \lambda_p^c$$

$$\sum_{c=1}^{m} \sum_{p \in EP} \bar{x}_{vc}^p \, \lambda_p^c = 1 \qquad v \in V$$

$$\sum_{p \in EP} \lambda_p^c = 1 \qquad c = 1, \ldots, m$$

- The extreme points in which $\bar{y}_c^p = 0$ can be removed
- The colors are identical (same set of extreme points)

$$\lambda_p = \sum_{c=1}^{m} \lambda_p^c$$

- After the removal of some variables the "convex combination" constraints become $\leq$ and they can be dropped (due to the objective function)

Exponential-size formulation

Now by using inner description of the "stable set" constraints we obtain:

$$\min_{\lambda \in \{0,1\}} \quad \sum_{c=1}^{m} \sum_{p \in EP} \bar{y}_c^p \, \lambda_p^c$$

$$\sum_{c=1}^{m} \sum_{p \in EP} \bar{x}_{vc}^p \, \lambda_p^c = 1 \qquad v \in V$$

$$\sum_{p \in EP} \lambda_p^c = 1 \qquad c = 1, \ldots, m$$

- The extreme points in which $\bar{y}_c^p = 0$ can be removed
- The colors are identical (same set of extreme points)

$$\lambda_p = \sum_{c=1}^{m} \lambda_p^c$$

- After the removal of some variables the "convex combination" constraints become $\leq$ and they can be dropped (due to the objective function)

Exponential-size formulation

$$\min_{\lambda \in \{0,1\}} \sum_{p \in EP : \bar{y}^p_c = 1} \underbrace{\left(\sum_{c=1}^{m} \lambda^c_p \right)}_{\lambda_p}$$

$$\sum_{p \in EP} \bar{x}^p_{vc} \underbrace{\left(\sum_{c=1}^{m} \lambda^c_p \right)}_{\lambda_p} = 1 \qquad v \in V$$

by replacing the variables we obtain:

$$\min_{\lambda \in \{0,1\}} \sum_{p \in EP} \lambda_p$$

$$\sum_{p \in EP} \bar{x}^p_{vc} \lambda_p = 1 \qquad v \in V$$

by relaxing the integrality condition on the variables $\rightarrow \lambda \geq 0$, we obtain the fractional chromatic number $\chi_f(G)$

Exponential-size formulation

$$\min_{\lambda \in \{0,1\}} \sum_{p \in EP : \bar{y}_c^p = 1} \underbrace{\left(\sum_{c=1}^{m} \lambda_p^c \right)}_{\lambda_p}$$

$$\sum_{p \in EP} \bar{x}_{vc}^p \underbrace{\left(\sum_{c=1}^{m} \lambda_p^c \right)}_{\lambda_p} = 1 \qquad v \in V$$

by replacing the variables we obtain:

$$\min_{\lambda \in \{0,1\}} \sum_{p \in EP} \lambda_p$$

$$\sum_{p \in EP} \bar{x}_{vc}^p \lambda_p = 1 \qquad v \in V$$

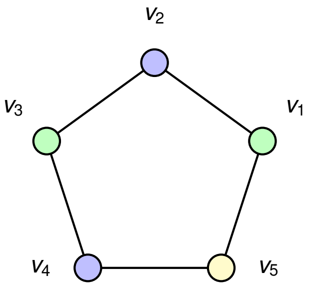
by relaxing the integrality condition on the variables $\rightarrow \lambda \geq 0$, we obtain the fractional chromatic number $\chi_f(G)$

Exponential-size formulation: example

Let $\mathscr{S}$ represent the set of incidence vectors of all stable sets of G:

$$\mathscr{S} = \left\{x \in \{0,1\}^n : x_u + x_v \leq 1, uv \in E\right\}$$

- $\mathscr{S} = \left\{ \underbrace{\{v_1, v_3\}}_{S_1}, \underbrace{\{v_1, v_4\}}_{S_2}, \underbrace{\{v_2, v_4\}}_{S_3}, \underbrace{\{v_2, v_5\}}_{S_4}, \underbrace{\{v_3, v_5\}}_{S_5} \right\}$



$$\begin{array}{llr}
\min_{\lambda \geq 0} & \lambda_{S_1} + \lambda_{S_2} + \lambda_{S_3} + \lambda_{S_4} + \lambda_{S_5} & \\
& \lambda_{S_1} + \lambda_{S_2} = 1 & (v_1) \\
& + \lambda_{S_3} + \lambda_{S_4} = 1 & (v_2) \\
& \lambda_{S_1} + \lambda_{S_5} = 1 & (v_3) \\
& + \lambda_{S_2} + \lambda_{S_3} = 1 & (v_4) \\
& + \lambda_{S_4} + \lambda_{S_5} = 1 & (v_5)
\end{array}$$

cycle C of size 5

$\omega(C) = 2, \chi(C) = 3$

- $\lambda^*_{S_1} = \lambda^*_{S_2} = \lambda^*_{S_3} = \lambda^*_{S_4} = \lambda^*_{S_5} = \frac{1}{2} \rightarrow \chi_f(G) = 2.5$

Pricing and Branching

Column Generation

- Restricted Mater Problem

$$\min_{\lambda \geq 0} \sum_{S \in \mathscr{S}} \lambda_S$$
$$\sum_{S \in \mathscr{S} : v \in S} \lambda_S \geq 1 \quad v \in V$$

- Dual Problem

$$\max_{\pi \geq 0} \sum_{v \in V} \pi_v$$
$$\sum_{v \in S} \pi_v \leq 1 \quad S \in \mathscr{S}$$

- Given an opt. sol. (λ^*, π^*) of the (RMP), find a stable set $S^* \in \mathscr{S}$:

$$\sum_{v \in S^*} \pi^*_v > 1$$

- Pricing: *Max Weight Stable Set Problem (MWSSP)*

$$\alpha(G, \pi^*) = \max \sum_{v \in V} \pi^*_v \, x_v$$
$$x_u + x_v \leq 1 \quad uv \in E$$
$$x_v \in \{0, 1\} \quad v \in V.$$

MWSSPs are solved by means of specialized B&B algorithms!

- Main Idea! Given a valid lower bound on MWSSP of value q, we can partition V into two disjoint sets of vertices

$$P \text{ and } B = V \setminus P \text{ such that } \alpha(G[P], \pi^*) \leq q$$

  Branching is necessary on the vertices in B only!

- Instead of computing $\alpha(G[P], \pi^*)$, strong MWSSP upper bounds are obtained via feasible dual solutions:

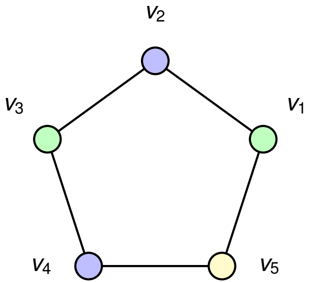
$$\begin{aligned} \alpha(G, \pi^*) \leq \min \sum_{K \in \tilde{\mathscr{K}}} \rho_K & & \\ \sum_{K \in \tilde{\mathscr{K}} : v \in K} \rho_K \geq \pi^*_v & \quad & v \in V, \\ \rho_K \geq 0 & \quad & K \in \tilde{\mathscr{K}}. \end{aligned}$$

  where $\tilde{\mathscr{K}}$ is a subset of the cliques of the graph.

- If $\tilde{\mathscr{K}}$ is a vertex disjointed clique partition $\rightarrow$ Max Coloring Upper Bound!

Infra-chromatic Bounding Functions (Main Idea)

Partial MAX-SAT Bound



cycle C of size 5
$\omega(C) = 2$, $\chi(C) = 3$

- Hard Clauses (non-edges)

  $h_1 \equiv \bar{x}_1 \vee \bar{x}_3,\ h_2 \equiv \bar{x}_1 \vee \bar{x}_4$

  $h_3 \equiv \bar{x}_2 \vee \bar{x}_4,\ h_4 \equiv \bar{x}_2 \vee \bar{x}_5,\ h_5 \equiv \bar{x}_3 \vee \bar{x}_5$

- Soft Clauses (colors)

  $s_1 \equiv x_1 \vee x_3,\ s_2 \equiv x_2 \vee x_4,\ s_3 \equiv x_5$

- Unit Literal Propagation

  $x_5 = 1 \rightarrow x_2 = 0\ (h_4) \rightarrow x_4 = 1\ (s_2)$

  $x_5 = 1 \rightarrow x_3 = 0\ (h_5) \rightarrow x_1 = 1\ (s_1)$

- Inconsistency!
  $\rightarrow h_2 \quad \text{core} \quad \{s_1, s_2, s_3\}$
- Stronger Bound
  $\rightarrow \chi(C) > 3 - 1 = 2 \geq \omega(C)$

Some Computational Results

	$\|V\|$	$\|E\|$	$\mu(G)$	$\lceil \chi_f(G) \rceil$	BBMCW t[s] tot	BBMCW t[s] pricing	MWSS t[s] tot	MWSS t[s] pricing
flat300_28_0	300	21695	0.48	28 (28)	**136.06**	**118.29**	881.03	863.05
r1000.5	1000	238267	0.48	234 (234)	**268.40**	**211.79**	2556.25	2508.37
r250.5	250	14849	0.48	65 (65)	**3.88**	**3.54**	6.41	6.15
DSJR500.5	500	58862	0.47	122 (122)	**21.35**	**18.67**	94.86	93.04
DSJR500.1c	500	121275	0.97	85 (85)	**8.73**	**8.43**	40.27	39.97
DSJC125.5	125	3891	0.50	16 (17)	**2.36**	**1.85**	3.83	3.33
DSJC250.9	250	27897	0.90	71 (72)	**5.07**	**4.47**	9.40	8.93
queen10_10	100	1470	0.30	10 (11)	**3.19**	**2.64**	4.92	4.37
queen11_11	121	1980	0.27	11 (11)	**9.20**	**8.21**	13.87	12.98
queen12_12	144	2596	0.25	12 (12)	**41.51**	**39.63**	67.42	65.60
queen13_13	169	3328	0.23	13 (13)	**234.69**	**231.10**	303.05	299.73
queen14_14	196	4186	0.22	14 (14)	**1564.04**	**1558.14**	1922.45	1916.36

Table 1: Comparing the performance of BBMCW and MWSS as pricing algorithms in computing the fractional cromatic number $\chi_f(G)$.

[1] P. San Segundo, F. F. and J. Artieda.
A new branch-and-bound algorithm for the Maximum Weighted Clique Problem.
Computers & Operations Research , 110:18 – 33, 2019.

Ryan/Foster branching rule

Basic idea. At each node of the branching tree select two vertices $v, u \in V$:

$$\sum_{S \in \mathscr{S} : u \in S, v \in S} \lambda^*_S = \gamma, \;\; \gamma \text{ is fractional}$$

Then two branching nodes are created as follows:

1) vertices v and u take the same color

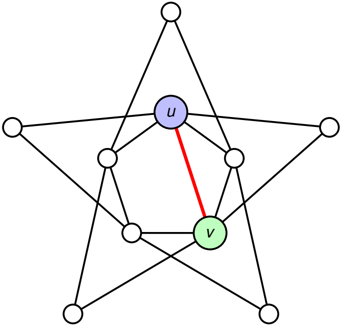
2) vertices v and u take different colors

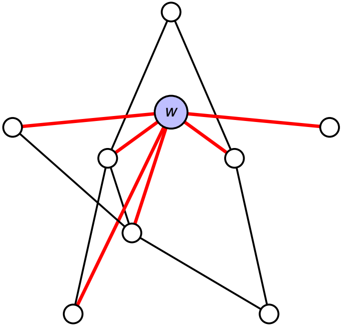
- This branching rule is complete (Zykov (49), Barnhart et al. (98)). Since the master constraint matrix A is a 0-1 matrix, if a basic solution to $A\,\lambda^* = 1$ is fractional, then there exist two rows (vertices) u and v of the master problem such that:

$$0 < \sum_{S \in \mathscr{S} : u \in S, v \in S} \lambda^*_S < 1$$

- It preserve the same pricing algorithm! Only minor graph modifications are necessary.

Example: *u* and *v* fractionally colored





- The first subproblem graph is obtained by adding the edge *uv* which forces these vertices to take different colors.
- The second subproblem graph is obtained by merging the two vertices into a new vertex *w* (connected to all the neighbours of *u* and *v*). This forces the two vertices to take the same color.

Conclusions and Future Lines of Search

- The Vertex Coloring Problems and its variants are very challenging problems. The state-of-the-art exact approaches are branch-and-price algorithms
- There is still a large space for improvements since only instances with up to 100 vertices can be effectively solved
- To the best of my knowledge, no branch-and-cut-and-price algorithms have been developed for the VCP
- Some techniques to accelerate the column generation phase can be designed, e.g., stabilization, smoothing, strong branching, column enumeration and columns pools, pricing relaxations etc. etc. . . . . . .
- I have not mentioned other exact approaches for the VCP like e.g., other compact ILP formulations, branch-and-cut algorithms, combinatorial branch-and-bound algorithm like DSATUR-B&B . . . . . . etc. etc. . . . . . .

Conclusions and Future Lines of Search

- The Vertex Coloring Problems and its variants are very challenging problems. The state-of-the-art exact approaches are branch-and-price algorithms

- There is still a large space for improvements since only instances with up to 100 vertices can be effectively solved

- To the best of my knowledge, no branch-and-cut-and-price algorithms have been developed for the VCP

- Some techniques to accelerate the column generation phase can be designed, e.g., stabilization, smoothing, strong branching, column enumeration and columns pools, pricing relaxations etc. etc. ......

- I have not mentioned other exact approaches for the VCP like e.g., other compact ILP formulations, branch-and-cut algorithms, combinatorial branch-and-bound algorithm like DSATUR-B&B ...... etc. etc. ......

(some of my coloring) References

[1] F. F. and E. Malaguti.
Exact weighted vertex coloring via branch-and-price.
Discrete Optimization, 9(2):130–136, 2012.

[2] D. Cornaz, F. F., and E. Malaguti.
Solving vertex coloring problems as maximum weight stable set problems.
Discrete Applied Mathematics, 217(Part 2):151 – 162, 2017.

[3] F. F., V. Gabrel, and I. Ternier.
An improved dsatur-based branch-and-bound algorithm for the vertex coloring problem.
Networks, 69(1):124–141, 2017.

[4] F. F., E. Malaguti, and A. Santini.
An exact algorithm for the partition coloring problem.
Computers & Operations Research, 92:170 – 181, 2018.

[5] D. Cornaz, F. F., E. Malaguti, and A. Santini.
A note on selective line-graphs and partition colorings.
Operations Research Letters, 2019.