Maximum Independent Set Problem and Effective Methods for its Solving

Rasskazova VA[1]

[1] Department of Computer Modelling and Probability Theory
Moscow Aviation Institute

3rd Winter School on Data Analytics (DA 2018)
Nizhny Novgorod, 2018

Outline

Outline

1. Classic $\mathcal{NP}$-hard Problem
   - Complexity classes and their relations
   - MISP and MCP
2. Applications
3. An Exact Methods for MISP
   - Dynamic Programming
   - Bron-Kerbosh Algorithm
4. An Effective Methods for MISP
   - Variable Neighborhood Search (VNS)
   - Extended Simplicial Vertex Test (ESVT)

Class $\mathcal{NP}$

Let's consider any search-problem.

Require: I, S {an instance and a candidate solution}
Ensure: true/ false

Class $\mathcal{NP}$ (non-deterministic polynomial-time) is a class of search-problems, for which there exist polynomial-time algorithms to check a candidate solutions.

Class $\mathcal{P}$

Let's consider optimisation problem corresponding to some search-problem.

Require: I {an instance}
Ensure: min S (max S) {an optimal solution}

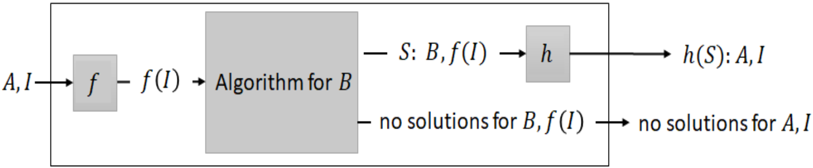
Class $\mathcal{P}$ is a class of search-problems, for which there exist polynomial-time algorithms to solve corresponding optimization problems.

Are there search-problems, for which corresponing optimisation problems cann't be solved by polynomial-time algorithm, that is

$$\mathcal{NP} \neq \mathcal{P}?$$

Class $\mathcal{NPC}$

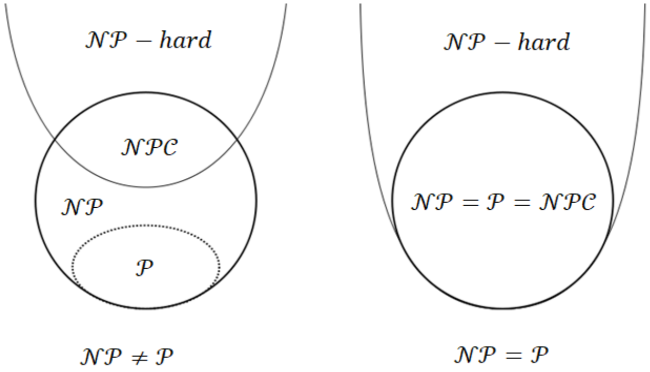
Let *f* and *h* be polynomial-time algorithms.



Class $\mathcal{NPC}$ ($\mathcal{NP}$-complete) is a class of search-problems, to each of which one can reduced any problem from $\mathcal{NP}$.

Class $\mathcal{NP} - hard$

Class $\mathcal{NP} - hard$ is a class of optimization problems, for which corresponding search-problems belong to $\mathcal{NPC}$.

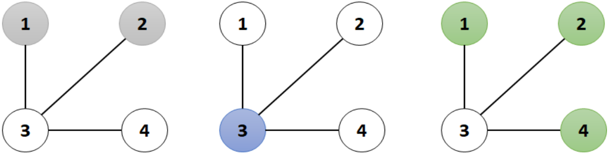


Outline

Independent Sets

Independent set (IS) of an undirected graph $G = (V, E)$ is any empty induced subgraph, that is

$$S, S \subseteq V, \text{ – is an IS, if for all } v, u \in S\colon\ (u, v) \notin E\ .$$

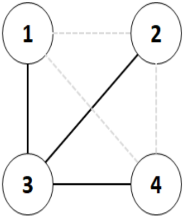


An IS: $\{1, 2\}$.
Maximal IS couldn't be extended by any vertex: $\{3\}$.
Maximum IS has the largest cardinality: $\{1, 2, 4\}$.

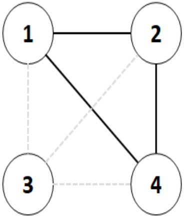
Complementary Graph

A comlementary graph $\tilde{G} = \left(V, \tilde{E}\right)$ has the same set of vertices and

$$(u, v) \in \tilde{E} \text{ if and only if } (u, v) \notin E \, .$$



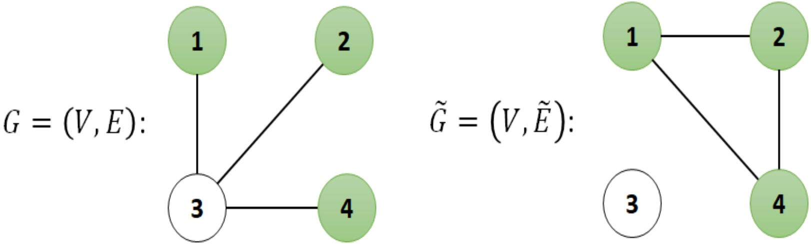
Initial graph $G = (V, E)$.



Complementary graph $\tilde{G} = (V, \tilde{E})$.

Maximum Clique

Maximum clique (MC) of an undirected graph $G = (V, E)$ is the largest (by cardinality) complete subgraph.



$$\text{MIS}\,(G) = \text{MC}\left(\tilde{G}\right) .$$

MIS Problem

Class $\mathcal{NP}$

Require: $G = (V, E)$, $S \subseteq V$.
Ensure: Is the set S an IS of the initial graph G (yes/ no)?

Class $\mathcal{NPC}$

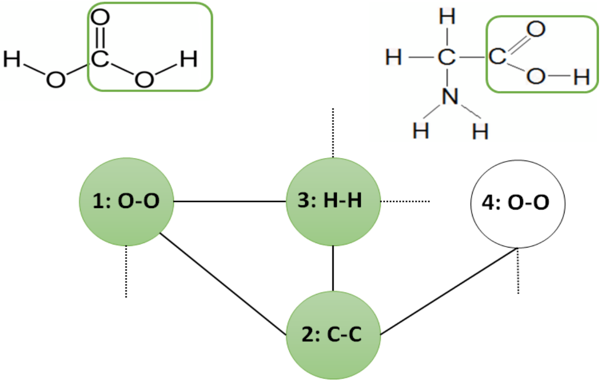
Require: $G = (V, E)$, $k \in \mathbb{Z}$.
Ensure: Is there in the initial graph G an IS S of the size k (yes/ no)?

Class $\mathcal{NP}$-hard

Require: $G = (V, E)$.
Ensure: MIS S of the initial graph G (optimal solution).

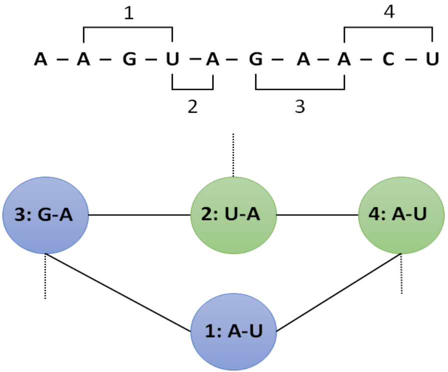
Chemistry

MC in a special graph is a maximum common substructures of molecules.



Bioinformatics

MC in a special graph is a secondary structure of RNA with the largest number of paired bases.

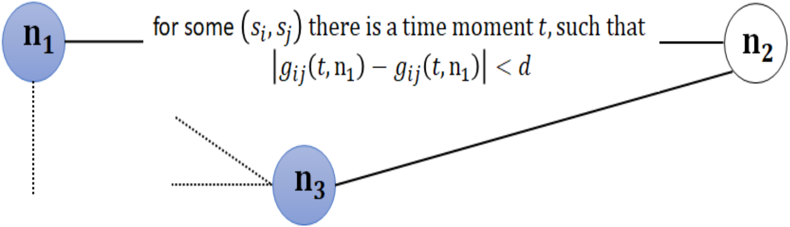


Transportation Managment

Let $\overrightarrow{G} = (S, E)$ be a railway network and let

$$\mathrm{N} = \{\mathrm{n}: \ (s_1, s_2, g_{12}(t, \mathrm{n})), (s_2, s_3, g_{23}(t, \mathrm{n})), \ldots\} \text{ —}$$

be the set of potential routes for trains moving.
Any IS of conflict-graph $G = (\mathrm{N}, \mathcal{E})$ is the set of feasible schedules for execution of transportations plan.

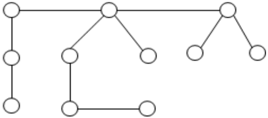


Outline

1. Classic $\mathcal{NP}$-hard Problem
   - Complexity classes and their relations
   - MISP and MCP
2. Applications
3. An Exact Methods for MISP
   - Dynamic Programming
   - Bron-Kerbosh Algorithm
4. An Effective Methods for MISP
   - Variable Neighborhood Search (VNS)
   - Extended Simplicial Vertex Test (ESVT)

Polynomial class of MISP

A tree is an undirected connected acyclic graph.

$T = (V, E)$: 

Class $\mathcal{P}$

Require: $T = (V, E), k \in \mathbb{Z}$.
Ensure: Is there in the initial tree T an IS S of the size k (yes/no)?

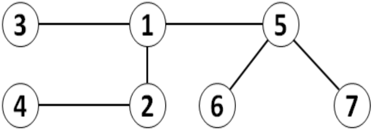
Dynamic Programming for MISP

Function MIS.Tr (v)

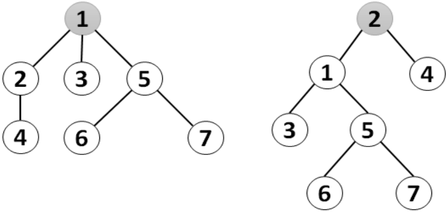
if $I[v] \neq 0$ **then**
    return $I[v]$
else
    $child.sum = 0$
    $g.child.sum = 0$
    for i=1 to child.num **do**
        $child.sum \longleftarrow child.sum + mis.tree(child[i])$
    for i=1 to g.child.num **do**
        $g.child.sum \longleftarrow g.child.sum + \text{MIS.Tr}\,(child[i])$
    $I[v] = \max\{child.sum; g.child.sum + 1\}$
    return I[v]

Dynamic Programming

Let a tree $T = (V, E)$ be given.



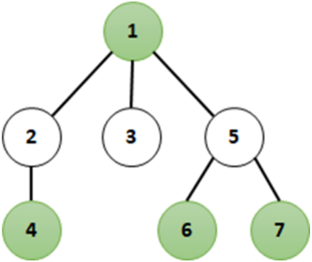
One can choose any vertex of an initial tree as a root.



Dynamic Programming

MIS.Tr (1):

$I[1] = \max\{I[2] + I[3] + I[5]; 1 + I[4] + I[6] + I[7]\}$
$\quad I[2] = \max\{[4]; 1\}$
$\quad\quad I[4] = 1$
$\quad I[2] = 1$
$\quad I[3] = 1$
$\quad I[5] = \max\{I[6] + [7]; 1\}$
$\quad\quad I[6] = 1$
$\quad\quad I[7] = 1$
$\quad I[5] = 2$
$\quad I[4] = 1$
$\quad I[6] = 2$
$\quad I[7] = 1$
I[1]=4



$\mathcal{O}(n+m)$

Outline

Bron-Kerbosh Algorithm

Bron C, Kerbosh J. (1973) Algorithm 457 — Finding all cliques of an undirected graph, Comm. of ACM, 16, pp. 575–577.

$\mathrm{M}: \ =$ the current independent set of vertices;
$\mathrm{K}: \ =$ the set of candidates which could be included in MIS under construction;
$\mathrm{P}: \ =$ the set of excluded vertices.

Bron-Kerbosh Algorithm

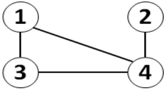
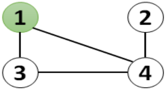
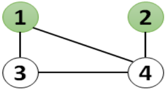
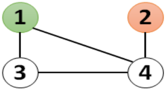
while $\mathrm{K} \neq \{\}$ or $\mathrm{M} \neq \{\}$ **do**
    if $\mathrm{K} \neq \{\}$ **then**
        $v \longleftarrow \mathrm{K.first}$
        push $v, \mathrm{M}, \mathrm{K}, \mathrm{P}$

$$\mathrm{M} \longleftarrow \mathrm{M} + \{v\}$$
$$\mathrm{K} \longleftarrow \mathrm{K} - \mathcal{N}(v) - \{v\}$$
$$\mathrm{P} \longleftarrow \mathrm{P} - \{v\}$$

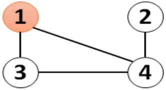
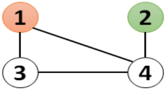
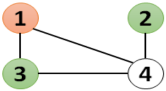
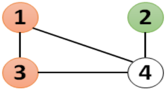
    else
        return M
        pop $v, \mathrm{M}, \mathrm{K}, \mathrm{P}$

$$\mathrm{K} \longleftarrow \mathrm{K} - \{v\}$$
$$\mathrm{P} \longleftarrow \mathrm{P} + \{v\}$$

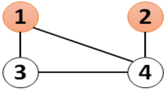
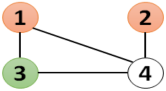
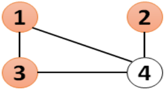
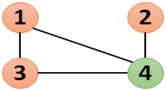
Bron-Kerbosh Algorithm

	M	K	P	v		
1	{ }	1234	{ }	1	$K \neq \{\ \}$; v=K.first	
2	1	2	{ }	2	push M, K, P; $K \neq \{\ \}$; v=K.first	
3	12	{ }	{ }		push M, K, P; $K = \{\ \}$; ► M={12}	
4	1	{ }	2		pop M, K, P (go to 2); $K = \{\ \}$; ► M={1}	

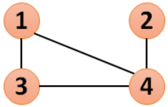
Bron-Kerbosh Algorithm

	M	K	P	v		
5	{ }	234	1	2	pop M, K, P (go to 1); K $\neq$ { }; v=K.first	
6	2	3	1	3	push M, K, P; K $\neq$ { }; v=K.first	
7	23	{ }	1		push M, K, P; K = { }; ►M={23}	
8	2	{ }	13		pop M, K, P (go to 6); K = { }; ►M={2}	

Bron-Kerbosh Algorithm

	M	K	P	v		
9	{ }	34	12	3	pop M, K, P (go to 5); K ≠ { }; v=K.first	
10	3	{ }	12		push M, K, P; K = { }; ► M={3}	
11	{ }	4	123	4	pop M, K, P (go to 9); K ≠ { }; v=K.first	
12	4	{ }	123		push M, K, P; K = { }; ► M={4}	

Bron-Kerbosh Algorithm

	M	K	P	v		
13	{ }	{ }	1234		pop M, K, P (go to 11); K = { }; M = { }	
					STOP	$O\left(3^{n/3}\right)$

All ISs of initial graph: $\{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{2,3\}$.

E. Tomita, A. Tanaka, H. Takahashi (2006) The worst-case time complexity for generating all max cliques and computation experiments. Theoretical Computer Scinces, Vol. 363, Issue 1, pp. 28–42.

Computational Results

n (number)	density (%)	$\alpha(G)$ (number)	*t* (sec)
50	20	16	50,5
50	50	11	3,6
50	70	5	0,03
100	20	20	$> 60 \cdot 60$
100	50	15	$7,2 \cdot 60$
100	70	6	0,3
150	20	23	$> 60 \cdot 60$
150	50	17	$> 60 \cdot 60$
150	70	6	1,5

Outline

Steps of the Basic VNS

Initialization. Select $\mathcal{N}_k, k = 1, \ldots, k_{max}$; find X; choose a stopping condition.
Repeat until the «stopping condition».

1. $k \longleftarrow 1$
2. **Repeat** until $k = k_{max}$.
   1. **Shaking.** X' at random from $\mathcal{N}_k(X)$.
   2. **Local Search.** X'' obtained with X' as initial solution.
   3. **Move or Not.**

      if X'' is better than X **then**
      $X \longleftarrow X''$
      $k \longleftarrow 1$
      else
      $k \longleftarrow k + 1$

Basic VNS

Let $G = (V, E)$ be given and

$C \subseteq V$: any clique of G;
$S = C$: independent set of $\tilde{G}$;
$T = V - S$: transversal of $\tilde{G}$.

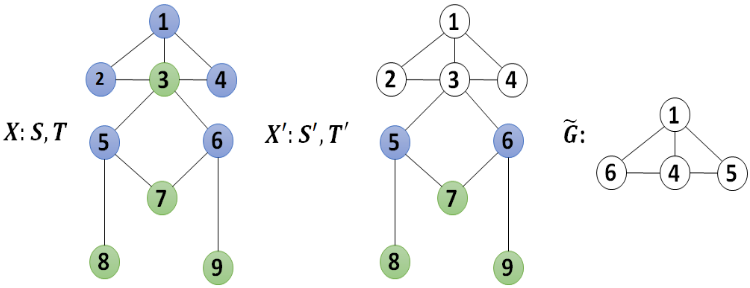
A transversal T_{opt} has a minimum cardinality and covers all vertices from S.

$$\mathcal{N}_k(X) = \{X' : \rho(X, X') = k\} \ ,$$

where $\rho(X, X')$ is the Hamming distance between corresponding 0-1 arrays for C and C'.

Shaking

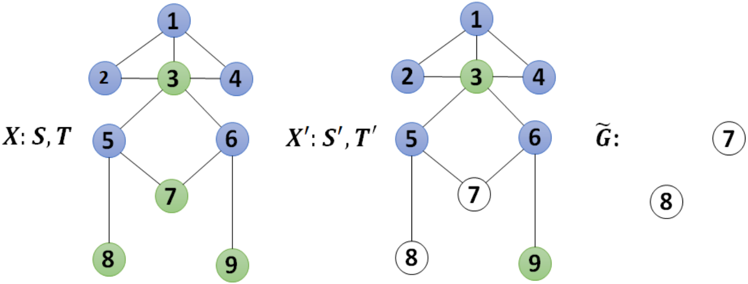
Let X and $k = 1$ are given.



$X' \longleftarrow Shaking(X, 1),$
$X'' \longleftarrow Local\ Search(\tilde{G}, X').$

Shaking

Let X and $k = 2$ are given.



$X' \longleftarrow Shaking(X, 2),$
$X'' \longleftarrow Local\ Search(\tilde{G}, X').$

Local Search

1. **Solution Test**

   if $\tilde{V} = \{\}$ **then**

   stop

2. **Add Step**

   $v \longleftarrow \min degree(\tilde{G})$ {using some tie-breaking rule}

   $C \longleftarrow C + \{v\}$

   $T \longleftarrow T + \left\{u \colon (u, v) \in \tilde{E}\right\}$

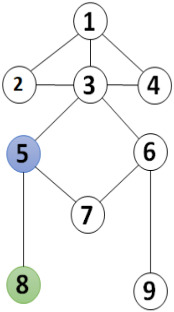
3. **Subproblem Reduction**

   $\tilde{V} \longleftarrow \tilde{V} - \{v, u\}$

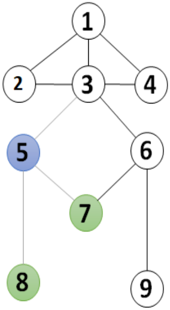
   $\tilde{E} \longleftarrow \tilde{E} - \{(w, u)\}$ {e$\in$ $\tilde{E}$ if and only if both endpoints belong to $\tilde{V}$}

   return to **Solution Test**

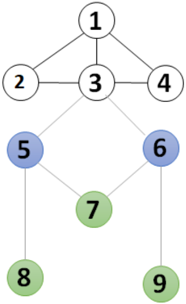
Greedy Selection Rule

	C	T	v	$\tilde{V}$	$\tilde{E}$
	{ }	{ }	8	123456789	(1,2), (1,3), (1, 4), (2, 1), (2, 3) (3, 1), (3, 2), (3, 4), (3, 5), (3, 6) (4, 1), (4, 3) (5, 3), (5, 7), (5, 8) (6, 3), (6, 7), (6, 9) (7, 5), (7, 6) (8, 5) (9, 6)
	8	5			

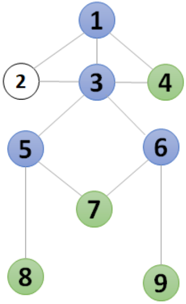
Greedy Selection Rule

	C	T	v	$\widetilde{V}$	$\widetilde{E}$
	8	5	7	1234679	(1,2), (1,3), (1, 4), (2, 1), (2, 3) (3, 1), (3, 2), (3, 4), (3, 6) (4, 1), (4, 3) (6, 3), (6, 7), (6, 9) (7, 6) (9, 6)
	87	56			

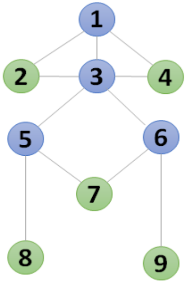
Greedy Selection Rule

	C	T	v	$\widetilde{V}$	$\widetilde{E}$
	87	56	9	12349	(1,2), (1,3), (1, 4) (2, 1), (2, 3) (3, 1), (3, 2), (3, 4) (4, 1), (4, 3)
	879	56			

Greedy Selection Rule

	C	T	v	$\tilde{V}$	$\tilde{E}$
	879	56	4	1234	(1,2), (1,3), (1, 4) (2, 1), (2, 3) (3, 1), (3, 2), (3, 4) (4, 1), (4, 3)
	8794	5613			

Greedy Selection Rule

	C	T	v	$\widetilde{V}$	$\widetilde{E}$
	8794	5613	2	2	(1,2), (1,3), (1, 4) (2, 1), (2, 3) (3, 1), (3, 2), (3, 4) (4, 1), (4, 3)
	87942	5613			
	87942	5613		{ }	{ }
	stop				

VNS for MCP

Initialization
Neighborhood structures: $k_{max} = 3$;
Initial solution X_0: $C = \{\}, T = \{\}$;
Stopping condition: 2 iterations between 2 improvements.

Shaking
Random tie-breaking rule.

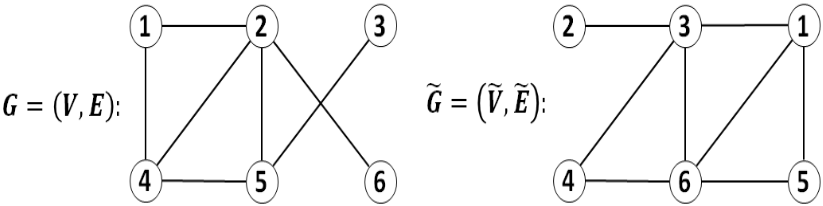
Move or Not
Local solution X'' is better than current solution X if and only if

$$\left|C\left(X''\right)\right| > |C(X)| \ ,$$

where $C(X)$ and $C\left(X''\right)$ are the corresponding cliques.

VNS for MCP

Let's consider an arbitrary graph $G = (V, E)$ and corresponding complementary graph $\tilde{G} = \left(\tilde{V}, \tilde{E}\right)$.



VNS for MCP

VNS step		C	T	$\widetilde{V}$	$\widetilde{E}$
Initialization	X	{ }	{ }	123456	(1,3) (1,5) (1,6) (2,3) (3,1) (3,2) (3,4) (3,6) (4,3) (4,6) (5,1) (5,6) (6,1) (6,3) (6,4) (6,5)
k=1					
Shaking (X,1)	X'	3	1246	5	{ }
Local Search (X')	X''	35	1246	{ }	{ }
Move or not	X	35	1246		
k=1					

VNS for MCP

VNS step		C	T	$\tilde{V}$	$\tilde{E}$
Shaking $(X,1)$	X'	5	16	234	(2,3) (3,2) (3,4) (4,3)
Local Search (X')		52	163	4	{ }
	X''	524	163	{ }	{ }
Move or not	X	524	163		
k=1					

VNS for MCP

VNS step		C	T	$\widetilde{V}$	$\widetilde{E}$
	X	524	163		
Shaking (X,1)	X'	24	63	15	(1,5) (5,1)
Local Search (X')	X''	241	635	{ }	{ }
Move or not	X	524	163		
k=2					

VNS for MCP

VNS step		C	T	$\widetilde{V}$	$\widetilde{E}$
Shaking (X,2)	X'	2	3	1456	(1,5) (1,6) (4,6) (5,1) (5,6) (6,1) (6,4) (6,5)
Local Search (X')		24	36	15	(1,5) (5,1)
	X''	245	361	{ }	{ }
Move or not	X	524	163		
k=3					
1-st iteration between 2 improvements					

VNS for MCP

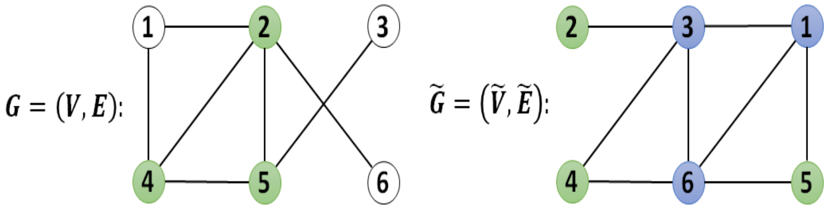
VNS step		C	T	$\tilde{V}$	$\tilde{E}$
k=1					
	X	524	163		
Shaking $(X,1)$	X'	54	136	2	{ }
Local Search (X')	X''	542	136	{ }	{ }
Move or not	X	524	163		
k=2					

VNS for MCP

VNS step		C	T	$\tilde{V}$	$\tilde{E}$
Shaking (X,2)	X'	4	36	125	(1,5) (5,1)
Local Search (X')		42	36	15	(1,5) (5,1)
	X''	421	365	{ }	{ }
Move or not	X	524	163		
k=3					
2-d iteration between 2 improvements					
STOP					

VNS for MCP

MC, IS and transversal of the initial and corresponding complementary graphs found by VNS.



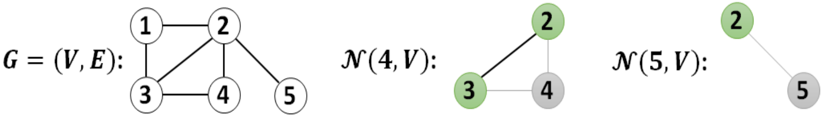
Outline

Simplicial Vertex Test (SVT)

A vertex v, $v \in V$ of a graph $G = (V, E)$ is a simplicial vertex, if its neighborhood is a complete subgraph.

We denote with $k[v]$ a number of vertices in a neighborhood of simplicial vertex v, that is

$$k[v] = |\mathcal{N}(v, V)|\,,\ \langle \mathcal{N}(v, V) \rangle_G \text{ — complete subgraph }.$$



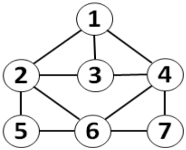
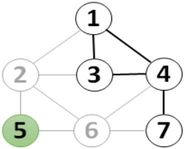
$k[4] = 2;$
$k[5] = 1.$

SVT for MISP

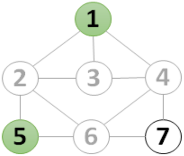
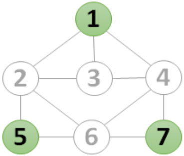
Require: $G = (V, E)$
Ensure: $S, S \subseteq V$

1: $V_0 \longleftarrow V$
   $S \longleftarrow \{\}$
   $E_0 \longleftarrow E$
2: **if** $V_0 = \{\}$ **then**
3:     **stop**
       return S
4: **for all** $v \in V_0$ **do**
5:     **if** v — is a simplicial vertex **then**
6:         $S \longleftarrow S + \{v\}$
           $V_0 \longleftarrow V_0 - \{v\} - \mathcal{N}(v, V_0)$ {neighborhood of v in a subgraph induced by V_0}
           $E_0 \longleftarrow E_0 - \{(u, v)\} - \{(u, w) \colon w \in \mathcal{N}(v, V_0)\}$
           go to step 2
7:     **else**
8:         **stop**
           return S, V_0

SVT for MISP

G_0	Step	S	V_0	E_0
	Initialization	{ }	123456	(1,2) (1,3) (1,4) (2,1) (2,3) (2,5) (2,6) (3,1) (3,2) (3,4) (4,1) (4,3) (4,6) (4,7) (5,2) (5,6) (6,2) (6,4) (6,5) (6,7) (7,4) (7,6)
	Search	5	1347	(1,3) (1,4) (3,1) (3,4) (4,1) (4,3) (4,7) (7,4)

SVT for MISP

G_0	Step	S	V_0	E_0
	Search	51	7	{ }
	Search	517	{ }	{ }
STOP				

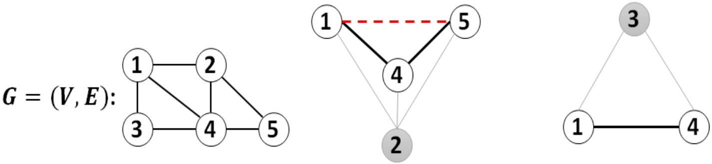
Extended SVT (ESVT)

Proposition

If for S found by SVT $V_0 = \{\}$, than S — MIS of $G = (V, E)$.

Let's denote with $m[v]$ a number of edges missed in a neighborhood of this vetrex to be a complete subgraph, that is

$$m[v] = C^2_{k[v]} - |E \cup E(\langle \mathcal{N}(v, V) \rangle_G)| \ .$$



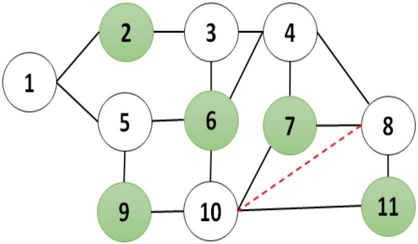
$k[2] = 3, m[2] = 1 \quad k[3] = 2, m[3] = 0$

ESVT

Require: $G = (v, E)$
Ensure: $S \subseteq V, Est$

1: $V_0 \longleftarrow V$
   $S \longleftarrow \{\}$
   $Est \longleftarrow 0$
2: **while** $V_0 \neq \{\}$ **do**
3:   **for all** $v \in V_0$ **do**
4:     calculate $k[v]$
       calculate $m[v]$
5:   $v_0 \longleftarrow MinMaxParam(v)$
     $S \longleftarrow S + \{v_0\}$
     $Est \longleftarrow Est + m[v_0]$
     $V_0 \longleftarrow V_0 - \{v_0\} - \mathcal{N}(v_0, V_0)$

ESVT for MISP



1	2	3	4	5	6	7	8	9	10	11	S	Est
[2,1]	[2,1]	[3,2]	[4,4]	[3,3]	[4,5]	[3,2]	[3,2]	[2,1]	[4,6]	[2,1]	11	1
[2,1]	[2,1]	[3,2]	[3,2]	[3,3]	[3,2]	[2,1]		[1,0]			11,9	1
[1,0]	[2,1]	[3,2]	[3,2]		[2,0]	[2,1]					11,9, 6	1
[1,0]	[1,0]					[0,0]					11, 9, 6, 2	1
						[0,0]					11, 9, 6, 2, 7	1

VNS with SVT

Initialization $X \longleftarrow SVT(X_0)$;
Repeat until the **Stopping condition.**

1. $k \longleftarrow 1$
2. **Repeat** until $k = k_{max}$.
   1. **Shaking** using random tie-breaking rule.
   2. **Local Search**
      $X'' \longleftarrow SVT(X')$
   3. **Move or Not**

Computational Results (VNS)

	GA1		GA2		VNS		BR
	|C|	Time	|C|	Time	|C|	Time	|C|
MANN_a27	126	2,55	126	8	126	0,07	126*
MANN_a45	343	34,38	343,59	383	344,5	20,79	345*
brock200_2	11	0,30	9,92	17	11,3	20,79	12*
brock200_4	16	0,25	16,04	24	16,9	6,8	17*
brock400_2	24	2,57	24	79	27,4	57,19	29*
brock400_4	24	1,59	24,13	12	33	36,90	33*
brock800_2	19	4,77	18,47	47	21	11,78	21
brock800_2	19	10,28	19,05	550	21	11,78	21
hamming8-4	16	0,33	16	1	16	0,01	16*
hamming10-4	33	15,68	39,18	46	40	0,26	40
keller4	11	0,18	11	1	11	0,01	11*
keller5	25	11,96	25,76	54	27	0,38	27

Computational Results (VNS)

	GA1		GA2		VNS		BR
	|C|	Time	|C|	Time	|C|	Time	|C|
p_hat300-1	8	0,63	7,69	2	8	0,02	8*
p_hat-2	25	0,93	25	1	25	0,01	25*
p_hat300-3	36	0,37	35,85	1	36	0,07	36*
p_hat700-1	9	4,74	9,45	180	11	0,52	11*
p_hat700-2	44	11,34	44	3	44	0,05	44*
p_hat700-3	62	4,32	62	3	62	0,06	62
p_hat1500-1	10	12,53	10,10	107	12	415,68	12*
p_hat1500-2	59	56	65	10	65	76,3	65
p_hat1500-3	92	56,24	93,44	860	94	0,58	94

C. Aggarwal, Orlin, R. Tai, Optimized crossover for the maximum independent set proble. Oper. Res. 45 (1997) 226-234.

Computational Results (ESVT)

	IncMaxCLQ	BBMCX	ESVT		BR
	Time	Time	|C|	Time	|C|
MANN_a81			1099	696,870972	1100*
brock400_2	259,9	132	22	0,024	29
brock400_4	197,7	20,2	22	0,04	33
brock800_2		3568	18	0,19	24
brock800_4		2532	17	0,191	26
hamming8-4			16	0,011	16
p_hat300-1			8	0,038	8
p_hat300-2			25	0,029	25
p_hat300-3	0,87	0,531	34	0,015	36

Computational Results (ESVT)

	IncMaxCLQ	BBMCX	ESVT		BR
	Time	Time	|C|	Time	|C|
p_hat700-1			8	0,363	11
p_hat700-2	1,25	1,42	43	0,691	44
p_hat700-3	357,4	718	61	0,144	62*
C125.9			33	0,001	34*
C250.9	333,2	1144	42	0,005	44*
DSJC1000_5	261	211	13	0,529	15

Li C et al. Combining MaxSAT Reasoning and Incremental Upper Bound for the Maximum Clique Problem.

San Segundo P et al. Infra-chtomatic Bound for the exact Clique. Comp. Oper. Research. 2015. 64, pp.293-303.

THANK YOU FOR ATTENTION!!