The map-coloring game

Present work by A. Audoin, C. Badiola, S. Da Silva, A. Perrot, C. Ritlewski

Client : Paul Dorbec
TD supervisor : Philippe Narbel

Original Article by Tomasz Bartnicki, Jarosław Grytczuk, H. A. Kierstead and Xuding Zhu

Some graph coloring reminders

- Old problem (first results in 1852)

Some graph coloring reminders

- Old problem (first results in 1852)

- Proper coloring if any two adjacent vertices have different colors

Some graph coloring reminders

- Old problem (first results in 1852)

- Proper coloring if any two adjacent vertices have different colors

- $\chi(G)$ = least number k for which G is k-coloriable

Some graph coloring reminders

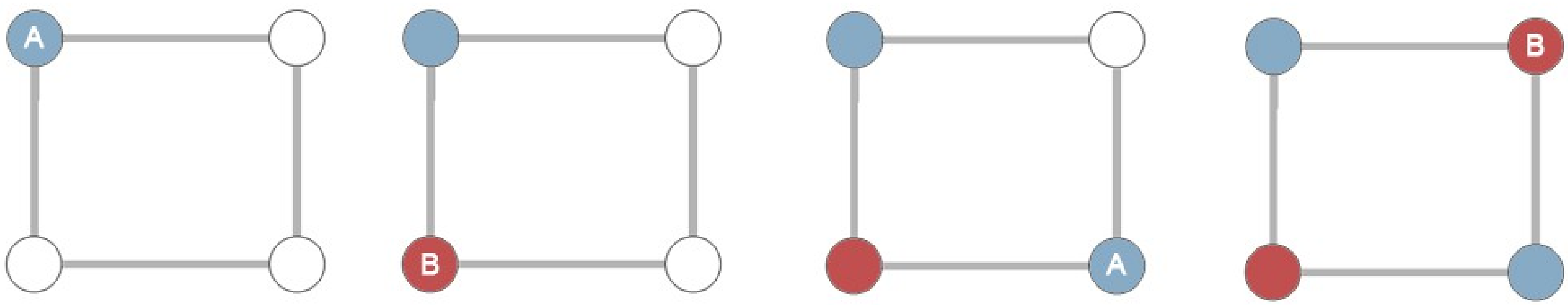
- Old problem (first results in 1852)
- Proper coloring if any two adjacent vertices have different colors
- $\chi(G)$ = least number k for which G is k-coloriable
- Deciding if a graph is k-coloriable is NP-complete

The map-coloring game

- Two players compete to color the graph
- Alice wants to achieve a proper coloring
- Bob tries to prevent Alice from winning
- The game is played with a fixed number of colors

A sample game

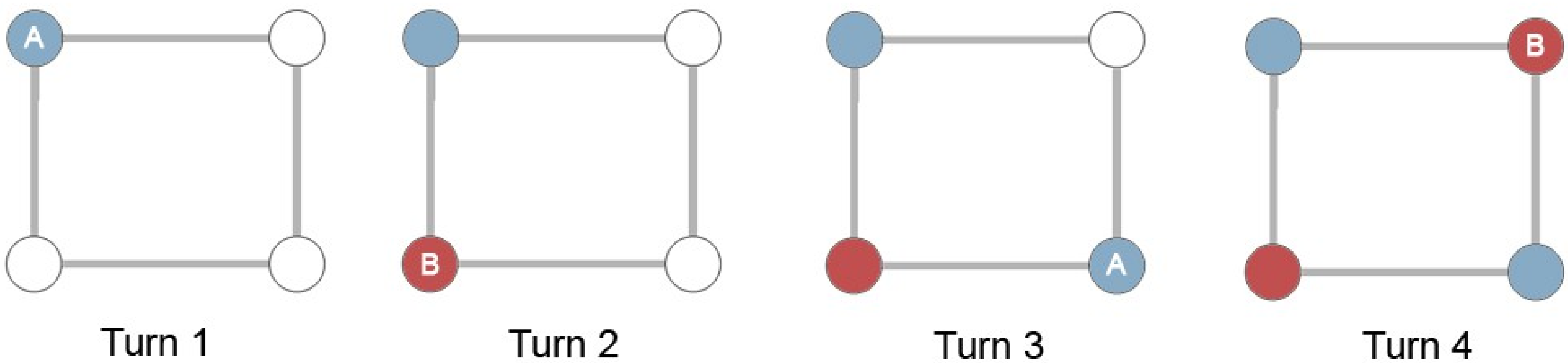
Game 1 : Alice won with 2 colors



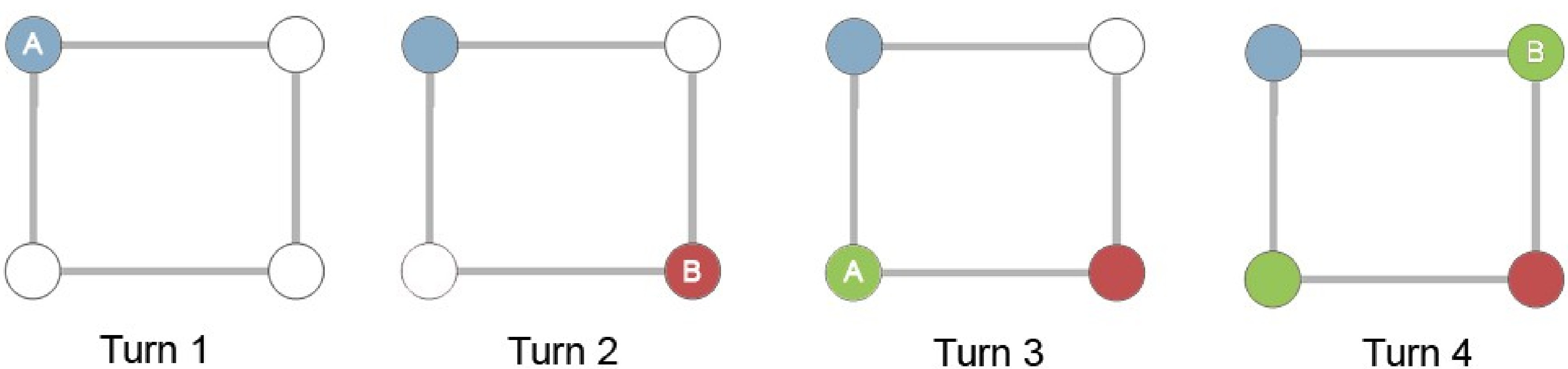
Turn 1 Turn 2 Turn 3 Turn 4

A sample game

Game 1 : Alice won with 2 colors



Game 2 : Alice can't win with 2 colors but she can win with 3 colors

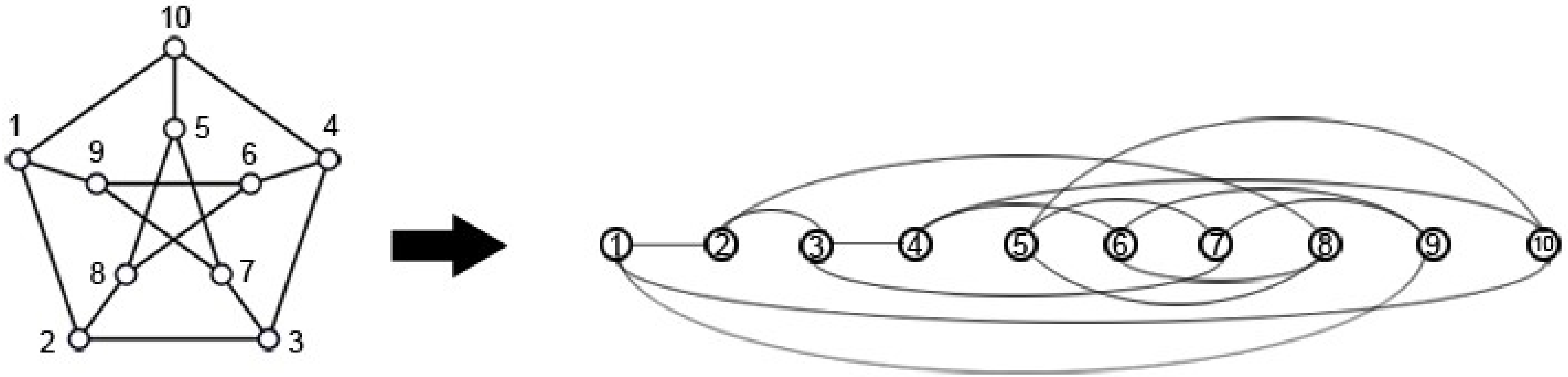


Game chromatic number

- χ_g (G) is the least number of colors for which Alice has a winning strategy regardless of Bob's strategy
- Several strategies have been created to find upper bounds to this number

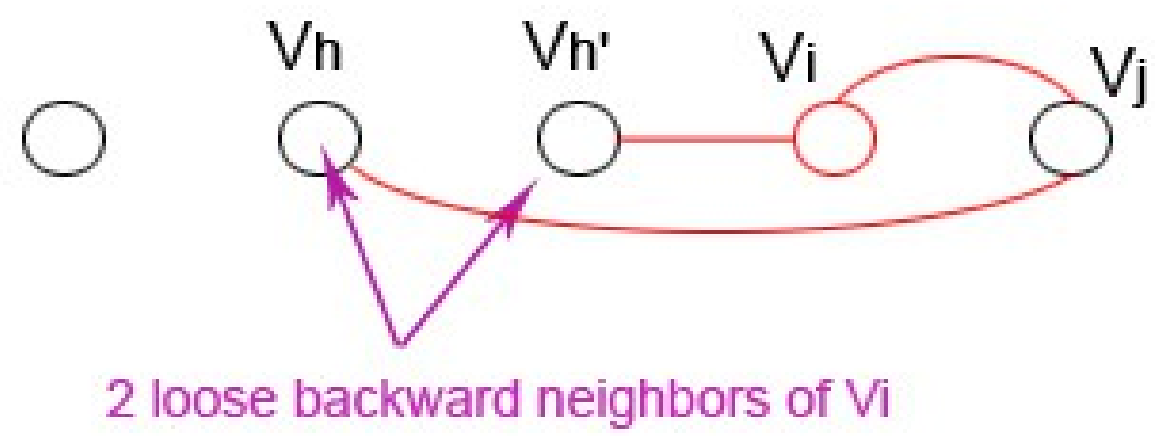
First Strategy

- Based on a graph ordering satisfying $col_2(G)$



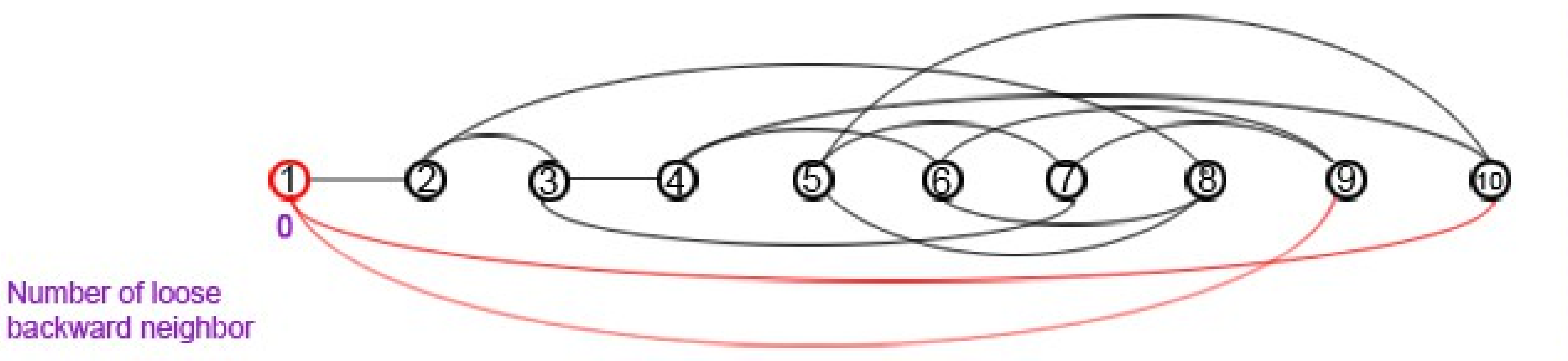
First Strategy

- Based on a graph ordering
- Loose backward neighbors :



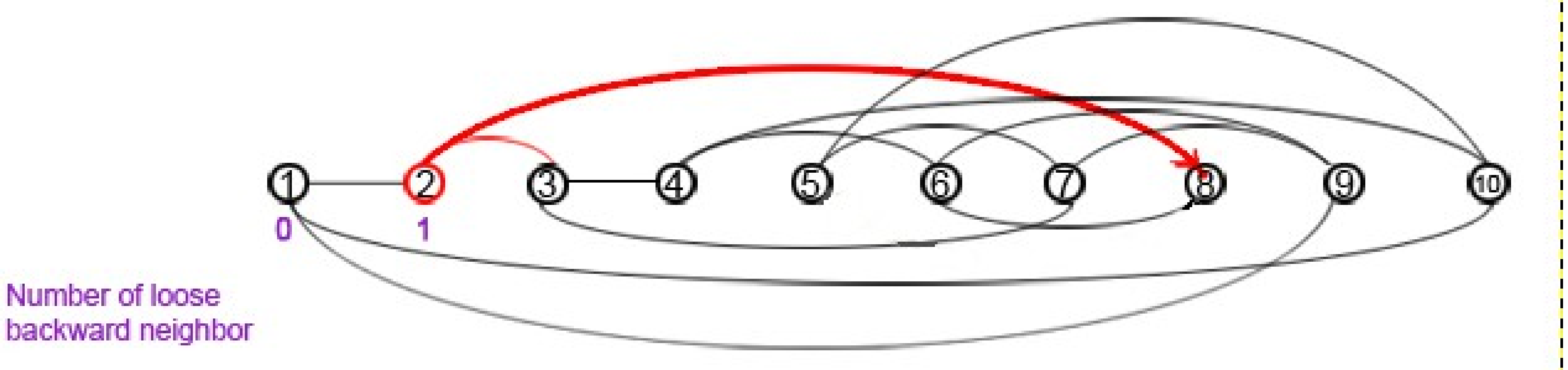
First Strategy

- Based on a graph ordering
- Loose backward neighbors



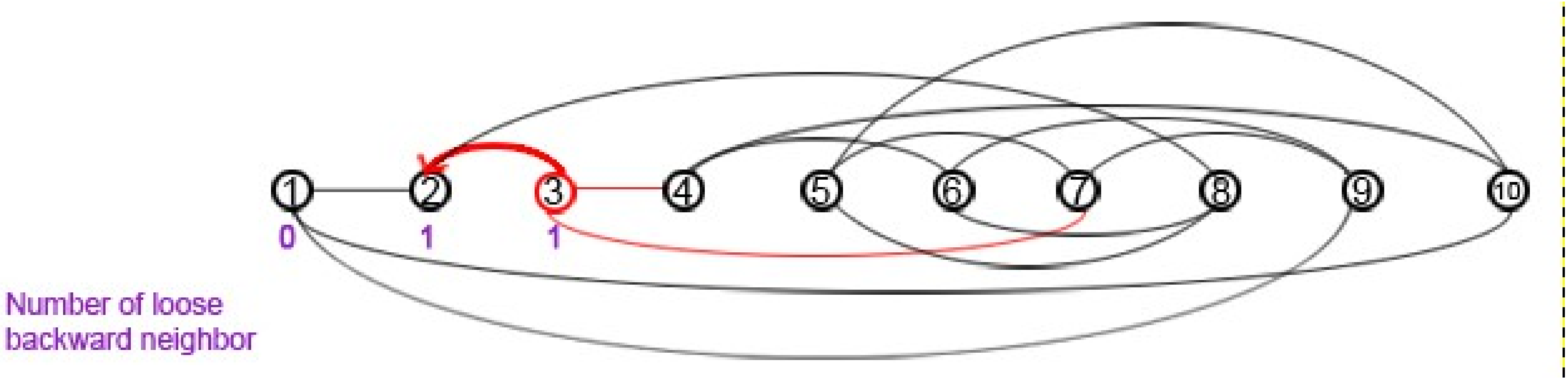
First Strategy

- Based on a graph ordering
- Loose backward neighbors



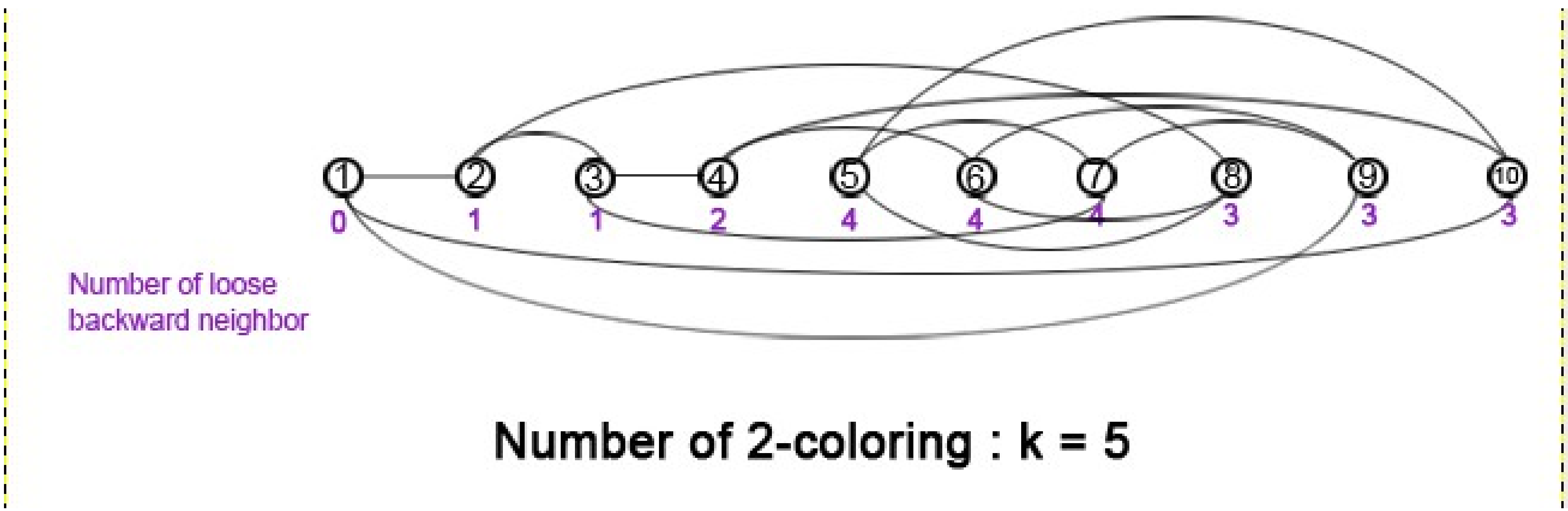
First Strategy

- Based on a graph ordering
- Loose backward neighbors



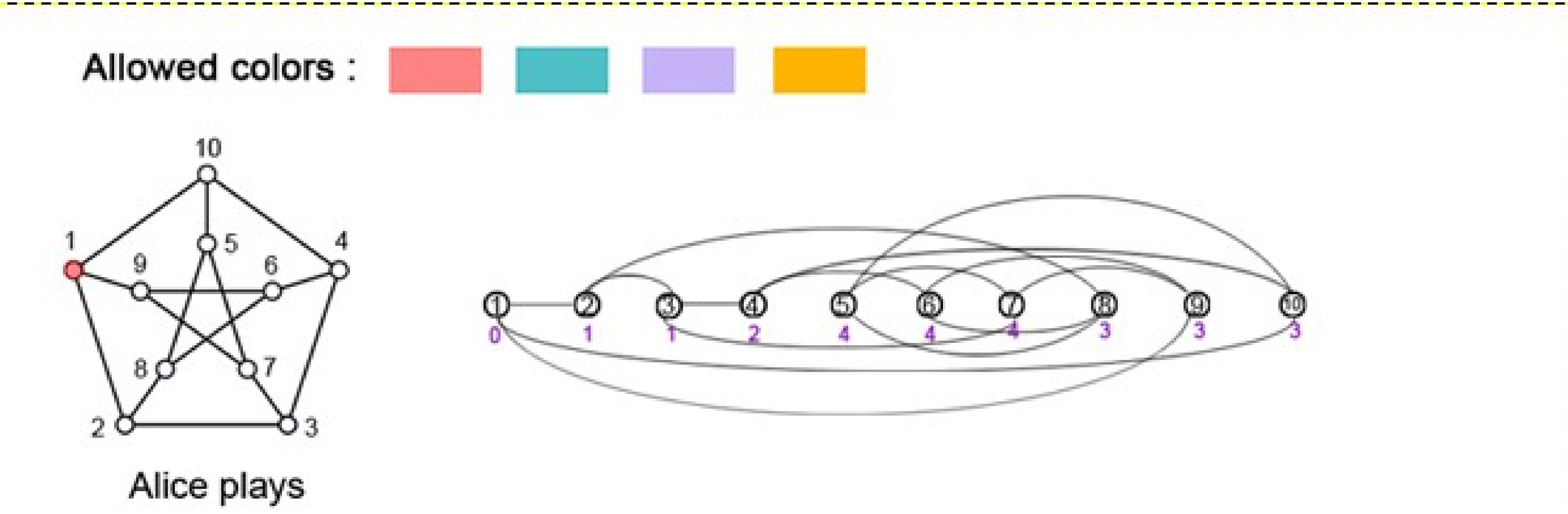
First Strategy

- Based on a graph ordering
- Loose backward neighbors



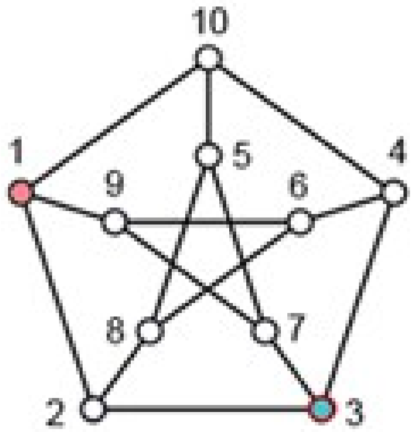
First Strategy

- Based on a graph ordering
- Loose backward neighbors
- Alice's strategy:

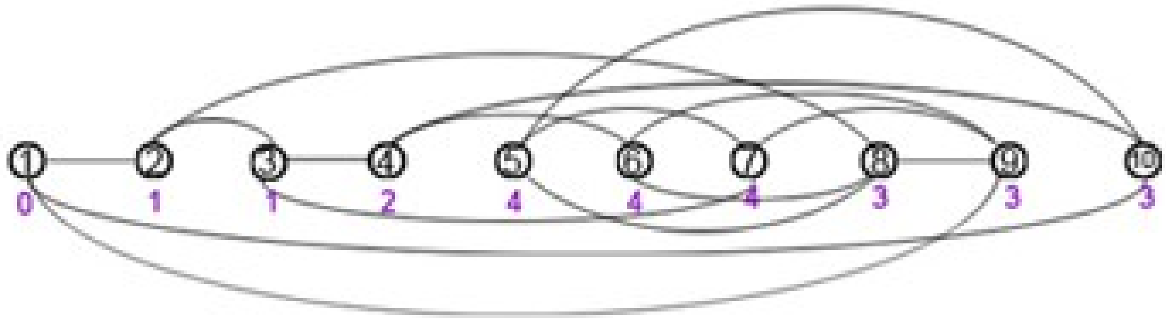


First Strategy

- Based on a graph ordering
- Loose backward neighbors
- Alice's strategy:

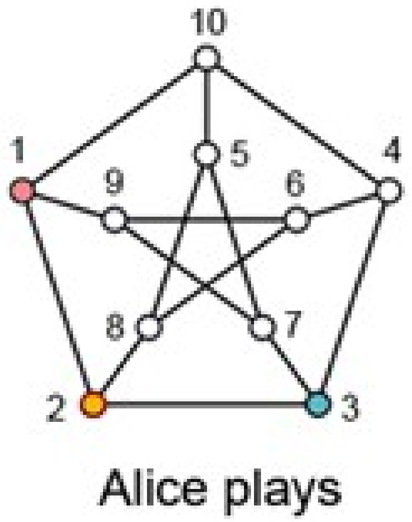


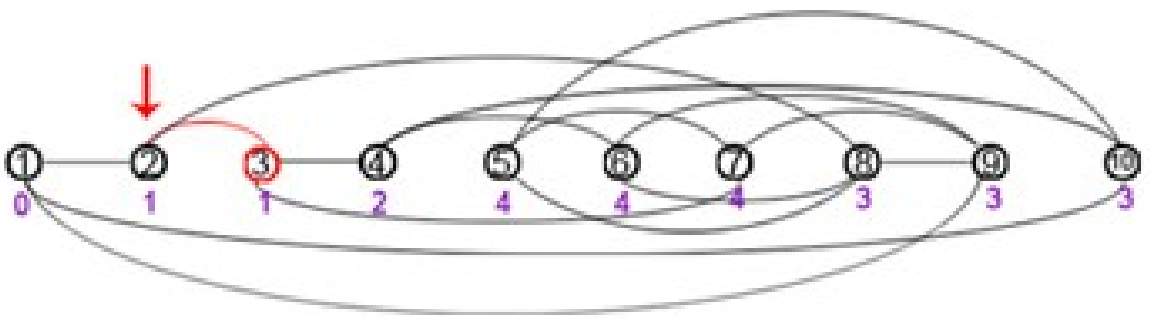
Bob plays



First Strategy

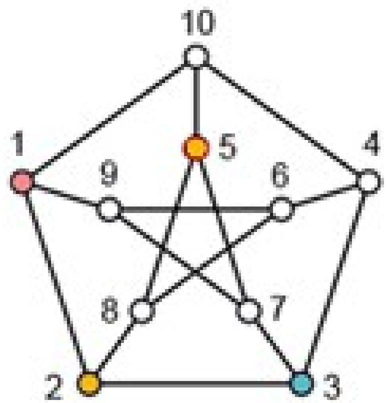
- Based on a graph ordering
- Loose backward neighbors
- Alice's strategy:



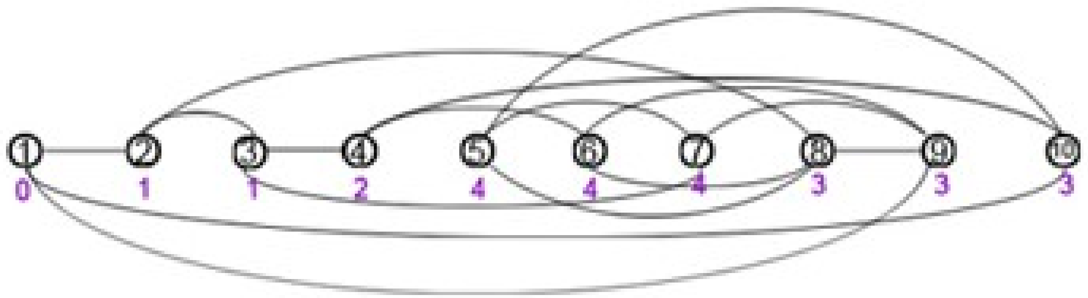


First Strategy

- Based on a graph ordering
- Loose backward neighbors
- Alice's strategy:

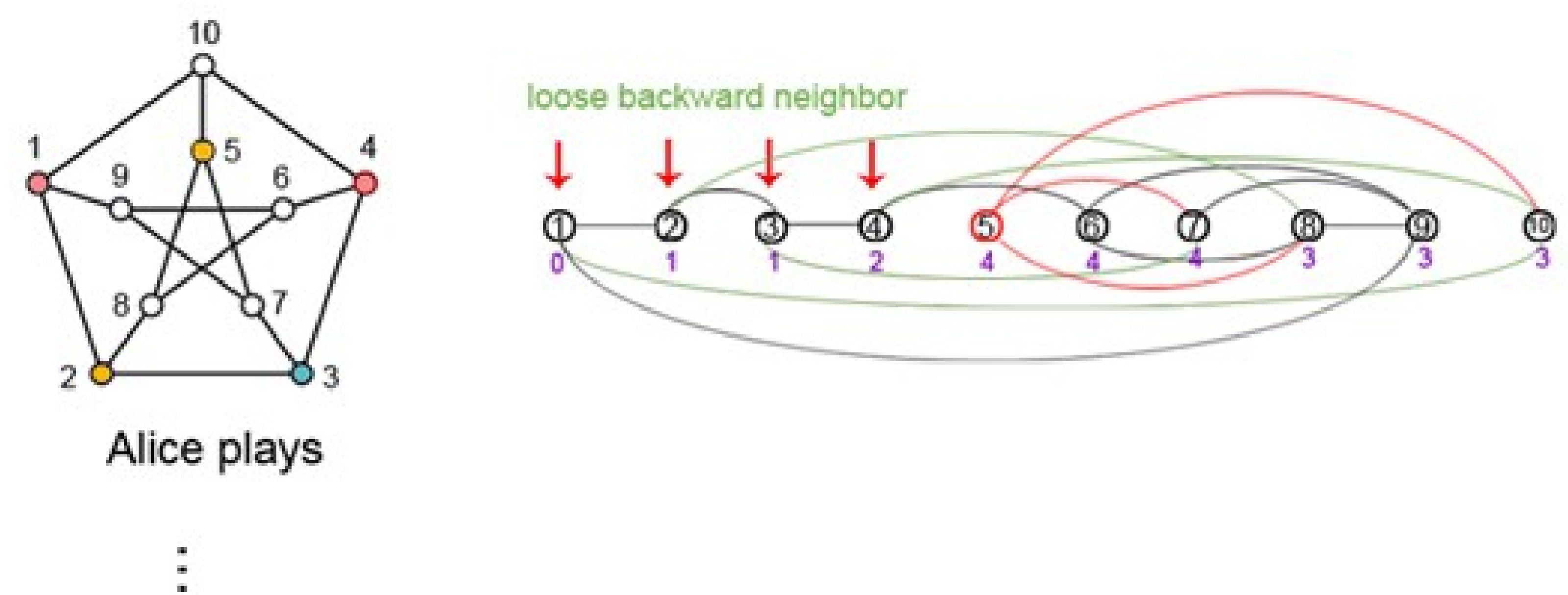


Bob plays



First Strategy

- Based on a graph ordering
- Loose backward neighbors
- Alice's strategy:



Upper bounds

- First strategy: $\chi_g(G) \leq \chi(G) \times (1 + col_2(G))$

Upper bounds

- First strategy: $\chi_g(G) \leq \chi(G) \times (1 + col_2(G))$
- Second strategy: $\chi_g(G) \leq a(G) \times (a(G) + 1)$
- Third strategy: $\chi_g(G) \leq 3 \times col_2(G) - 1$
- Fourth strategy: $\chi_g(G) \leq 18$ for planar graphs

Monte-Carlo Tree Search

- Based on Monte-Carlo methods
- 4 steps:

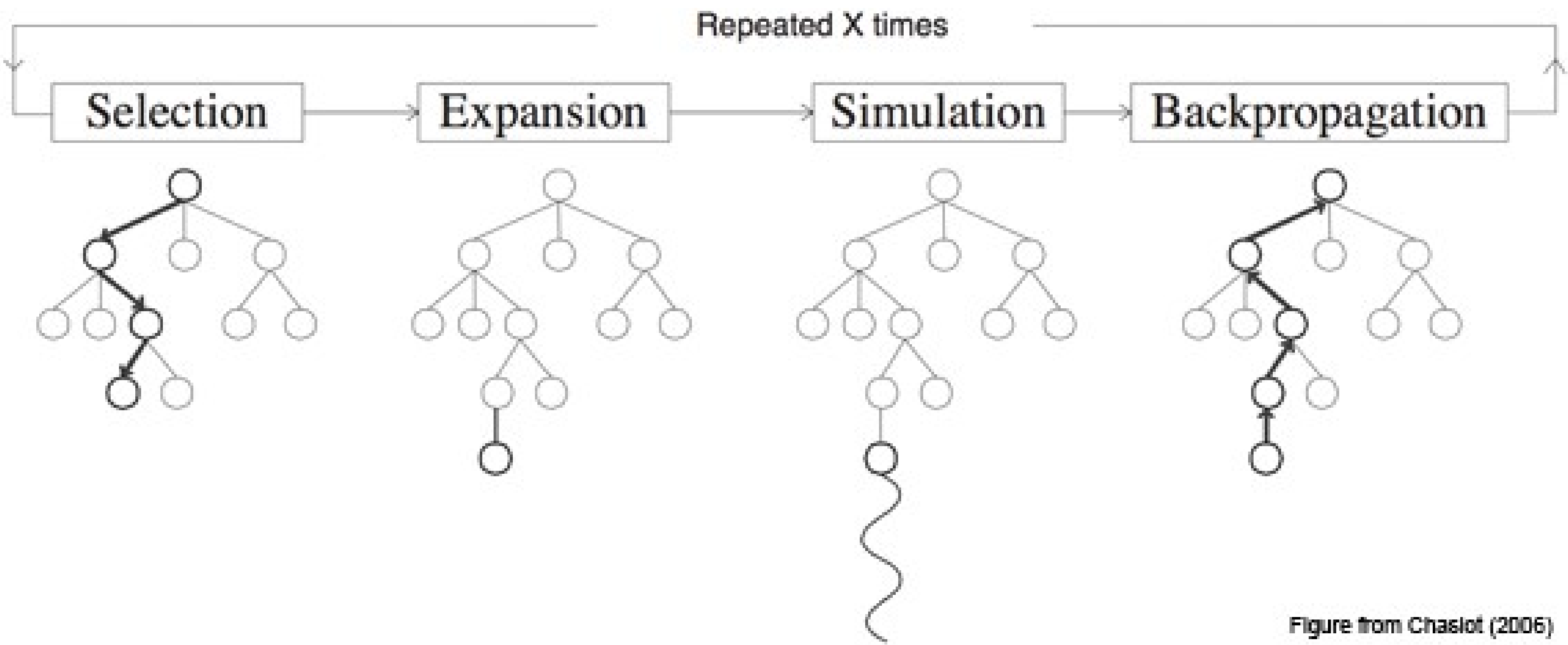


Figure from Chaslot (2006)

UCB-1

- Multi-armed bandit algorithm
- Computes an upper confidence bound for each move
- Selects the move whose value + bound is the highest
- The bound is reduced with each simulation

Work

- Various particular graphs
- Parameterizable program
- 1K simulations per choice
- 1K games per graph per color

Results

Graph	Chromatic number	Strategy 1	Strategy 2	Strategy 3	Experimental results
Chain (odd)	2	6	6	5	3
Chain (even)	2	6	6	5	3
Cycle (odd)	3	12	12	8	3
Cycle (even)	2	8	12	8	3
Grid 2x5	2	6	12	5	3

Results

Graph	Chromatic number	Strategy 1	Strategy 2	Strategy 3	Experimental results
Grid 5x5	2	10	12	11	5
Grid 5x5 tor	3	5	5	5	5
Binary tree	2	6	6	5	4
Petersen	3	18	12	14	4
Icosahedron	4	32	20	20	5
Grotzsch	4	?	?	?	4

Conclusion

- Better results than theoretical bounds
- Better theoretical model?

- Experimental biases:
  - Alice and Bob have the same strength
  - Adaptive AI versus strategies of the article

Prospect

- Better performances:
  - Parallelization
  - Better selection algorithms
- Strategies for Bob
- Independently change Alice's and Bob's strength