Position Guided Tabu Search for Graph Coloring

Daniel Porumbel, Jin Kao Hao, Pascale Kuntz

January 29, 2009

LERIA, University of Angers
France

LINA,University of Nantes
France

Main Problem

- Typical Local Search:
  - ''trying to find the top of Mount Everest in a thick fog while suffering from amnesia''—as expressed in a leading Artificial Intelligence book [Russell & Norvig. Artificial Intelligence: A Modern Approach, 1995.]
  - thick fog = no visibility over more than one step
  - amnesia = no long term memory
- Objective of Position-Guided Local Search (PGTS):
  - Use long-term memory to continuously guide the search process towards as-yet-unexplored regions

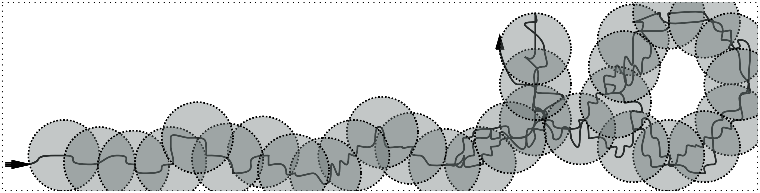
Objective of the new PGTS algorithm

1. Memorizing the whole exploration path
   - Complete recording (of each configuration) IMPOSSIBLE

2. Avoid already-explored regions

Objective of the new PGTS algorithm

1. Memorizing the whole exploration path
   - Coarse-grained recording (sphere par sphere) POSSIBLE:



2. Avoid already-explored spheres (orient toward as-yet-unvisited spheres)

Solution Sketch Position-Guided Local Search

1. The exploration process: Tabu Search (TS)
2. The learning process:
    - New concepts:
        - distance between configurations = minimal number of TS moves
        - a sphere of a configuration = all configurations within a radius
    - LEARNING: records all visited spheres
    - ACTION: discourage the main search process from revisiting a recorded sphere

Graph coloring

- Graph *K-coloring problem*: decide if there is a vertex coloring with K colors with no two adjacent vertices of the same color
- The general coloring problem: find the chromatic number χ, i.e. the minimum K such that graph G is K-colorable
- Our main search process: Tabucol—a classical Tabu Search (TS) algorithm for graph *K-coloring*,
  - numerous versions developed since 1987 [Hertz & Werra, Using Tabu Search Techniques for Graph Coloring]

Tabu Search algorithm for graph coloring

- Given a graph $G(V, E)$:
  - a coloring = a $|V|$-array, each position is a color assignment
  - neighborhood = a color change of a conflicting vertex
  - Tabu Search (TS) moves from coloring to coloring by changing a color
  - Each color change has to be not-Tabu, i.e. not performed in the *near past*
- The Tabu List prevents the search process to repeat recent moves
  - a move is re-performed only if it was not performed during the last T_ℓ iterations
  - shorter T_ℓ = more repetitive moves = stronger intensification

General Ideas

Typical Tabu Search (TS):

+++ helps the search process escape a local optimum

— — always attracted to a limited set of basins of attraction, it can often get locked looping from one local optimum to another.

Position-Guided Tabu Search:

+++ uses all advantages of TS (e.g. the local optimum escape mechanism)

+++ uses global positioning knowledge to guide the search process to avoid looping between local local optima

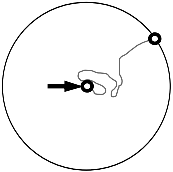
Search Space Distance Metric

Distance between colorings C_a and C_b:

- The minimal nr. of moves needed by TS to arrive from C_a to C_b $\sim$
- The set-theoretic partition distance:
  - a K coloring is a partition of V into K classes
  - how many vertices need to change their class in C_a to obtain C_b
  - first computing algorithms described since 1981 [Day W.H.E., The complexity of computing metric distances between partitions]
- Very fast computation time $(O(|V|))$:
  - We use a Las Vegas algorithm very effective in practice—we have a special article dedicated to it. [Porumbel, Hao and Kuntz, Submitted paper: A fast algorithm for computing the partition distance]

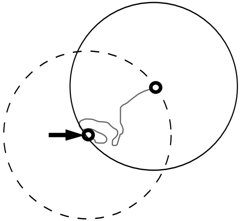
Tabu Search + Learning Process

- The sphere of a coloring C is the set of colorings situated at a distance of less than $R = 10\%|V|$ from C
- The *learning process* records all spheres explored by the *search process*
  - $R = 1 \iff$ all individual colorings are recorded
  - $R = |V| \iff$ one sphere encloses the whole search space



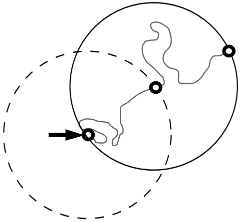
Tabu Search + Learning Process

- The sphere of a coloring C is the set of colorings situated at a distance of less than $R = 10\%|V|$ from C
- The *learning process* records all spheres explored by the *search process*
  - $R = 1 \iff$ all individual colorings are recorded
  - $R = |V| \iff$ one sphere encloses the whole search space



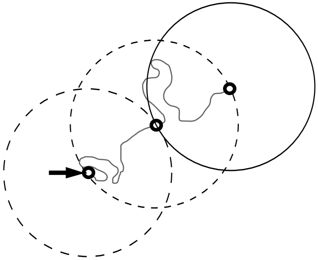
Tabu Search + Learning Process

- The sphere of a coloring C is the set of colorings situated at a distance of less than $R = 10\%|V|$ from C
- The *learning process* records all spheres explored by the *search process*
  - $R = 1 \iff$ all individual colorings are recorded
  - $R = |V| \iff$ one sphere encloses the whole search space



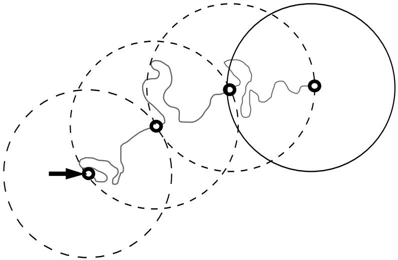
Tabu Search + Learning Process

- The sphere of a coloring C is the set of colorings situated at a distance of less than $R = 10\%|V|$ from C
- The *learning process* records all spheres explored by the *search process*
  - $R = 1 \iff$ all individual colorings are recorded
  - $R = |V| \iff$ one sphere encloses the whole search space



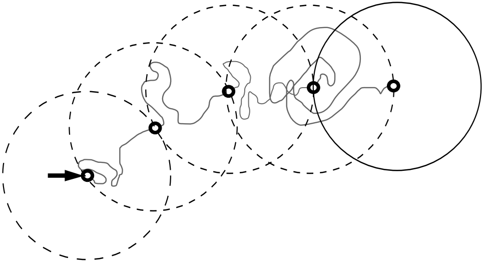
Tabu Search + Learning Process

- The sphere of a coloring C is the set of colorings situated at a distance of less than $R = 10\%|V|$ from C
- The *learning process* records all spheres explored by the *search process*
  - $R = 1 \iff$ all individual colorings are recorded
  - $R = |V| \iff$ one sphere encloses the whole search space



Tabu Search + Learning Process

- The sphere of a coloring C is the set of colorings situated at a distance of less than $R = 10\%|V|$ from C
- The *learning process* records all spheres explored by the *search process*
  - $R = 1 \iff$ all individual colorings are recorded
  - $R = |V| \iff$ one sphere encloses the whole search space

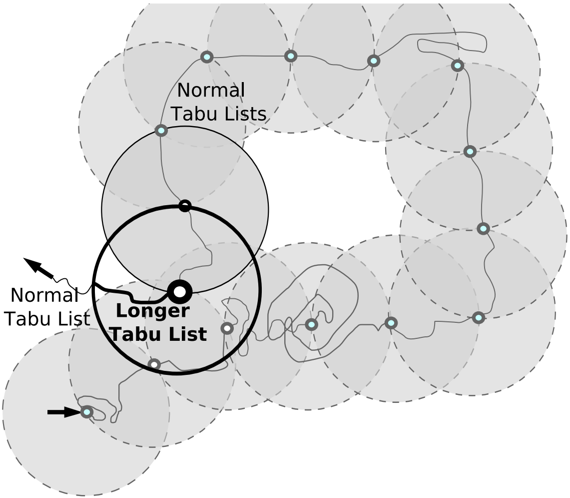


Tabu Search + Learning + Using learned information

- All visited spheres are recorded in an archive
- As soon as the search process visits a coloring covered by a previously-visited sphere

  $\Longrightarrow$
- PGTS starts a diversification phase
  - the Tabu list length is increased
  - this makes the next moves more diverse, less repetitive
  - it directs the search process towards other regions

Tabu Search + Learning + Using learned information



PGTS Advantages and Disadvantages

+ + + New regions are discovered at all stages of the execution

+ + + For small graphs, is possible to enumerate all spheres: their number varies from $K^{|V|}$ ($R = 1$) to 1 ($R = |V|$)

− − − An overhead is introduced by the learning process

PGTS Advantages and Disadvantages

- + + + New regions are discovered at all stages of the execution
- + + + For small graphs, is possible to enumerate all spheres: their number varies from $K^{|V|}$ ($R = 1$) to 1 ($R = |V|$)
- − − − An overhead is introduced by the learning process

PGTS Advantages and Disadvantages

$+++$ New regions are discovered at all stages of the execution

$+++$ For small graphs, is possible to enumerate all spheres: their number varies from $K^{|V|}$ $(R = 1)$ to 1 $(R = |V|)$

$---$ An overhead is introduced by the learning process

Learning process overhead

- PGTS can concentrate on the deep configurations, those that pose looping problems
- The value of R controls sphere sizes:
  - Larger R = less spheres = smaller overhead
  - We use $R = 10\%$: we performed an analysis showing that the local optima visited during a certain period are grouped in clusters of diameter $10\%|V|$.
- The distance is calculated very fast

Comparison Tabu Search

- Phase 1: PGTS is equivalent to TS in the beginning, while there are few or no recorded spheres
- Phase 2: The recoded spheres become important:
  - TS starts re-exploring the same spheres over and over again
  - PGTS keeps finding new regions at all times

Instance		PGTS		Basic TS	
Graph	K	Success rate	Time [h]	Success rate	Time [h]
*dsjc*250.5	28	10/10	< 1	10/10	< 1
*dsjc*500.5	48	2/10	35	0/10	–
*dsjc*1000.1	20	2/10	9	0/10	–
*dsjc*1000.5	87	5/10	28	0/10	–
*dsjc*1000.9	224	8/10	24	2/10	44
*flat*300_28_0	29	7/10	8	0/10	–
*le*450_25*c*	25	4/10	11	3/10	7
*le*450_25*d*	25	2/10	19	2/10	12
*flat*1000_76_0	86	3/10	33	0/10	–
*r*1000.1*c*	98	10/10	< 1	10/10	< 1

Table: Comparison PGTS vs. TS for a time limit of 50 hours.

Comparison Best Algorithms

- PGTS finds all solutions found by other local searches
- PGTS competes well also with population-based genetic algorithms

Graph	χ, K^*	PGTS	VSS [1] 2008	PCol [2] 2008	ACol [3] 2008	MOR [4] 1993	GH [5] 1999	MMT [6] 2008
*dsjc*250.5	?, 28	*28*	-	-	28	28	28	28
*dsjc*500.5	?, 48	*48*	48	48	48	49	48	48
*dsjc*1000.1	?, 20	*20*	20	20	20	21	20	20
*dsjc*1000.5	?, 83	87	88	88	84	88	83	83
*dsjc*1000.9	?, 224	*224*	224	225	224	226	224	226
*le*450.25*c*	25, 25	*25*	26	25	26	25	26	25
*le*450.25*d*	25, 25	*25*	26	25	26	25	26	25
*flat*300.28	28, 32	29	29	28	31	31	31	31
*flat*1000.76	76, 82	86	87	87	84	89	83	82
*r*1000.1*c*	?, 98	*98*	—	98	-	98	—	98

[1] Hertz et. al. Variable space search for graph coloring, [2] Blöchliger & Zufferey. A graph coloring heuristic using partial solutions and a reactive tabu scheme, [3] Galinier et. al. An adaptive memory algorithm for the k-coloring problem, [4] C. Morgenstern. Distributed coloration neighborhood search, [5] Galinier & Hao. Hybrid Evolutionary Algorithms for Graph Coloring, [6] Malaguti et. al. A Metaheuristic Approach for the Vertex Coloring Problem,

Conclusion

- We devised a local search algorithm that records its path to assure diversification
  - It often finds the best-known solution for graph coloring
  - It can methodically enumerate all visited spheres
  - It can perform a complete exploration given enough time
- The principle can be used for any problem where there is a distance between configurations
- PGTS completely solves diversification issues. We mixed it with an intensification algorithm:
  - It enabled finding for the first time a solution with 223 colors for the well studied graph *dsjc*1000.9