Using Tabu Search to Solve University Timetable Problem

Chongfeng Ling[1] Prof. M.B.N. (Thijs) Kouwenhoven[1]

[1]Xi'an Jiaotong-Liverpool University

May 18, 2022

Content

1. Background
2. Problem Description
   - Elements
   - Constraints
   - Graph Coloring
3. Methodology
   - Tabu Search
   - TABCOL Algorithm
4. Experiences
   - Assumptions
   - Examples
   - Data Generation
   - Numerical Results
5. Conclusion

Table of Contents

1. Background
2. Problem Description
   - Elements
   - Constraints
   - Graph Coloring
3. Methodology
   - Tabu Search
   - TABCOL Algorithm
4. Experiences
   - Assumptions
   - Examples
   - Data Generation
   - Numerical Results
5. Conclusion

Why timetable problem matters?

- Importance
- Complexity
- Periodicity

Table of Contents

1 Background

2 Problem Description
- Elements
- Constraints
- Graph Coloring

3 Methodology
- Tabu Search
- TABCOL Algorithm

4 Experiences
- Assumptions
- Examples
- Data Generation
- Numerical Results

5 Conclusion

What is the timetable problem in educational institute?

Educational Timetable Problem

Timetable problem is defined as an allocation to arrange elements into space and time subjects to constraints such that satisfies a set of desirable objectives as many as possible [4].

	Monday	Tuesday
9:00		MTH302-Lecture-D1/1 Xin He 1-6, 8-14: Online
9:30		
10:00		
10:30		

Figure: Personal timetable in XJTLU.

Elements

- Course

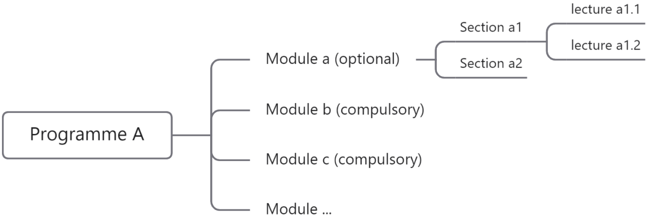

Figure: Elements in course design.

- Teacher and Student
- Classroom
- Time

Constraints

Hard Constraints

- Student overlap
- Teacher overlap
- Classroom overlap
- ...

Soft Constraints

- Geographical constraint
- Compactness constraint
- Student preferences
- ...

Constraints

Hard Constraints

- Student overlap
- Teacher overlap
- Classroom overlap
- ...

Soft Constraints

- Geographical constraint
- Compactness constraint
- Student preferences
- ...

Definition

A **feasible solution** satisfies all hard constraints.

Definition

An **optimal solution** is the feasible solution that satisfies all soft constraints, or minimize the objective function as small as possible.

Graph

Definition

Graph $G = (V, E)$ contains a set of elements V and their relations E.

- Vertices $V = \{v_1, \ldots, v_n\}$.
- Edges $E = \{(x, y) \mid x, y \in V\}$

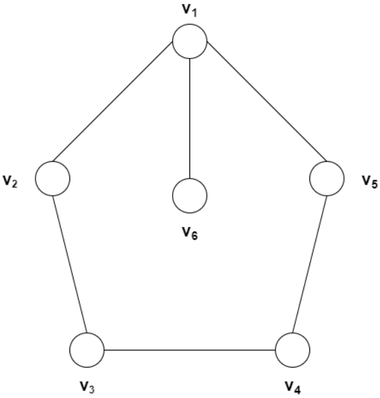

Figure: Simple graph

Graph Coloring

Graph Coloring

Label every vertex v a color such that no pair of adjacent vertices have the same color.

- $C = (C_1, \ldots, C_k)$: partition of V into a number k of subsets.
- $E = (E_1, \ldots, E_k)$: set of edges having both endpoints in C.

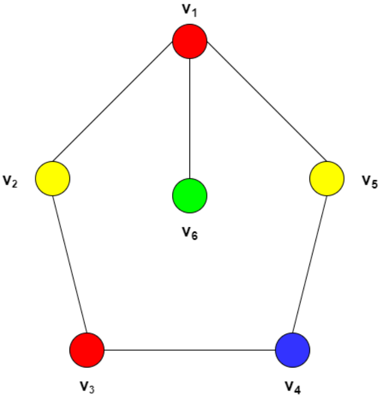

Figure: Colored simple graph

Table of Contents

1. Background
2. Problem Description
   - Elements
   - Constraints
   - Graph Coloring
3. Methodology
   - Tabu Search
   - TABCOL Algorithm
4. Experiences
   - Assumptions
   - Examples
   - Data Generation
   - Numerical Results
5. Conclusion

Tabu Search: Illustration

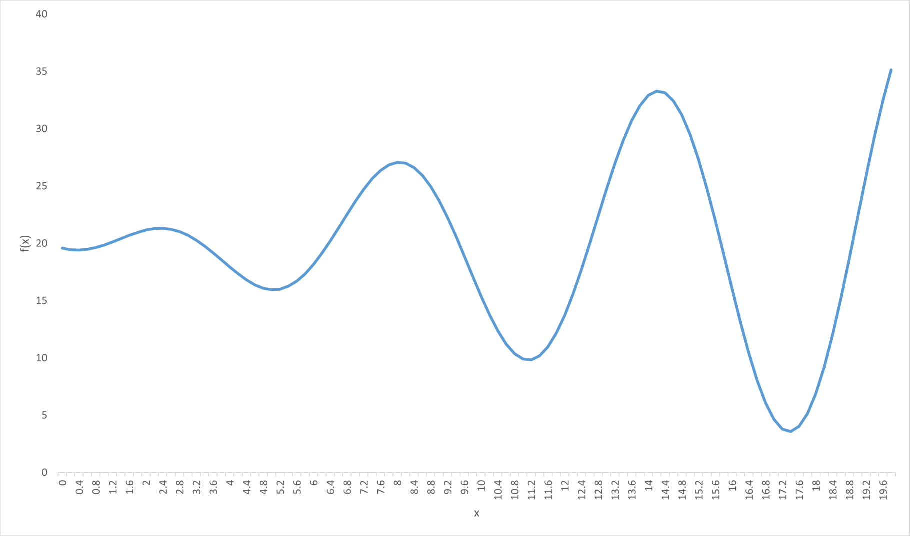

Figure: Find an optimal solution for $min\{f(x)\}$, $f : \mathbf{R} \rightarrow \mathbf{R}$

Tabu Search and its Modified Version

Tabu Search [1] [3]

Tabu Search is a hyper-heuristic algorithm designed for finding a **global optimal solution**.

TABCOL [2]

TABCOL is designed for minimizing the number of conflicted edges $f(s = (C_1, \ldots, C_k)) = \sum_{i=1}^{k} |E_i|$ to get a k-color graph.

Tabu Search and its Modified Version

Tabu Search [1] [3]

Tabu Search is a hyper-heuristic algorithm designed for finding a **global optimal solution**.

TABCOL [2]

TABCOL is designed for minimizing the number of conflicted edges $f\left(s=(C_1,\ldots,C_k)\right)=\sum_{i=1}^{k}|E_i|$ to get a k-color graph.

Algorithm	P	Neighbor	A. F.	O. F.
Tabu Search	$min\{f(x)\}$ $f:\mathbf{R}\to\mathbf{R}$	$\bar{B}(x_0,r)$	f	f
	$min\{f(x)\}$ $f:\mathbf{R^n}\to\mathbf{R}$	$\bar{B}(x_0,r)$	$\{f_i\}$	f
TABCOL	$min\{f(s)\}$	s'	f	f

Stopping Criterion

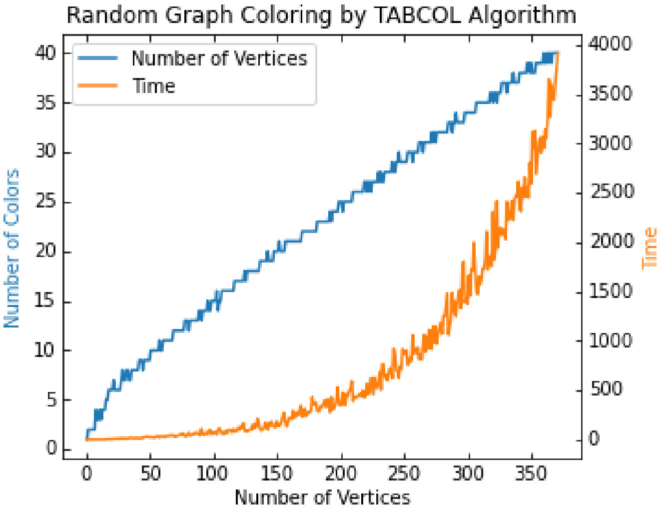

Figure: Time required to color a graph with different number of vertices.

Table of Contents

1. Background
2. Problem Description
    - Elements
    - Constraints
    - Graph Coloring
3. Methodology
    - Tabu Search
    - TABCOL Algorithm
4. Experiences
    - Assumptions
    - Examples
    - Data Generation
    - Numerical Results
5. Conclusion

Course Timetable Problem in XJTLU

Assumptions

- No sections
- Same type of courses
- Repeatability
- No teacher overlap
- No compactness constraints

Examples

Example I: Basic Model

- Student overlap constraint
- Uniform distriubution

Example II: with Soft Constraints

- Pre-assignment constraint
- Geographical constraint
- Distribution constraint

Examples

Example I: Basic Model

- Student overlap constraint
- Uniform distriubution

Example II: with Soft Constraints

- Pre-assignment constraint
- Geographical constraint
- Distribution constraint

Pre-assignment Constraint

Label vertices with fixed colors in the initial solution and remain unchanged.

Example II: with Soft Constraints

Distribution Constraint

- $h(a)$: number of courses labelled by color a with distriubution $H(a)$.
- $f_1 = \mathrm{Var}(H)$

Geographical Constraint

- $A = \{a_i \mid i = 1, \ldots, 9\}$: sorted color set.
- v_m, v_n: two courses labelled by two color a_m, a_n respectively.
- $f_2(v_m, v_n) = \begin{cases} q & \text{if } |i-j| = 1 \\ 0 & \text{otherwise} \end{cases}$, q: number of common students in two courses.

Objective Function

- $f(s) = w_1 f_1 + \sum_{\substack{v_m \neq v_n \\ v_m, v_n \in V}} w_2 f_2(v_m, v_n),\ w_1,\ w_2 \in \mathbf{R}$

Data Generation

Major	Compulsory	Optional	Number
AMY3S2	MTH301	MTH302, MTH308,PHY302,MTH310,MTH318	150
FMY3S2	MTH301, MTHH302, ECO310	FIN302, MTH316	200
ASY3S2	MTH301, MTH302, MTH306	ECO310, ECO304	80
AMY2S2	MTH208, MTH210	MTH209, MTH224, MTH203	300
FMY2S2	MTH203, FIN202, CPT206	FIN206, MTH208, MTH222	400
ASY2S2	MTH202, MTH214, MTH223	ECO216, FIN206	100
AMY1S2	MTH106, MTH108, MTH118, MTH122		300
FMY1S2	MTH116, MTH106, ECO120, FIN104		400
ASY1S2	MTH120, MTH116, ECO120, FIN104		100

Figure: Course selection for AM, FM, AS in year 1, 2 and 3.

Numerical Results

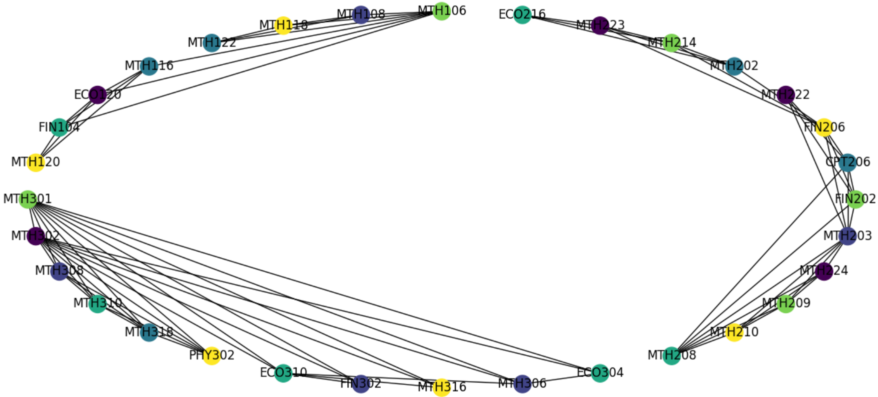

Figure: Graph Coloring Result. The big graph is composed of three isolated small graph.

Numerical Results

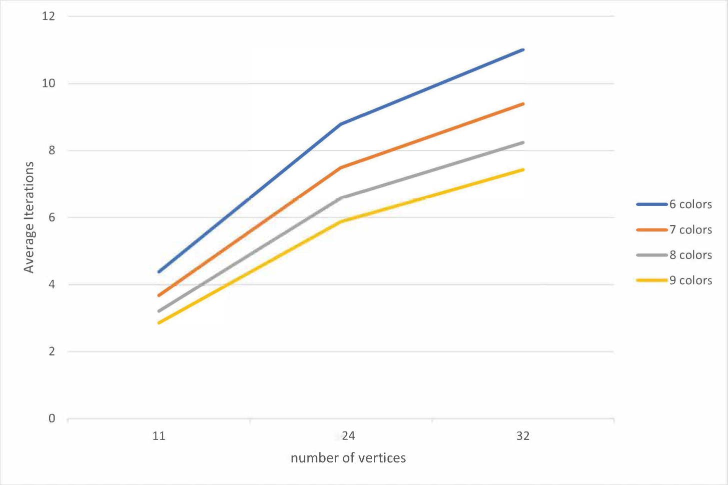

Figure: Average iterations of n colors, per 10000 attempts. $n = 5$ is not listed since its value is nbmax.

Table of Contents

1 Background

2 Problem Description
- Elements
- Constraints
- Graph Coloring

3 Methodology
- Tabu Search
- TABCOL Algorithm

4 Experiences
- Assumptions
- Examples
- Data Generation
- Numerical Results

5 Conclusion

Conclusion and Plan

Conclusion

- Feasible solution is accessible
- Optimal solution depends on function design
- Generalizability

Plan

- Student grouping problem
- Graph decomposition algorithm

References I

[1] F. Glover, *Tabu Search: A Tutorial*, Interfaces, 20 (1990), pp. 74–94.

[2] A. Hertz and D. Werra, *Using tabu search techniques for graph coloring*, Computing, 39 (1987), pp. 345–351.

[3] D. Werra and A. Hertz, *Tabu search techniques: A tutorial and an application to neural networks*, Operations-Research-Spektrum, 11 (1989), pp. 131–141.

[4] A. Wren, *Scheduling, timetabling and rostering — A special relationship?*, in Practice and Theory of Automated Timetabling, G. Goos, J. Hartmanis, J. Leeuwen, E. Burke, and P. Ross, eds., vol. 1153, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996, pp. 46–75.

Thank you!