A new framework of operation research and learning path recommendation for next-generation of e-learning services

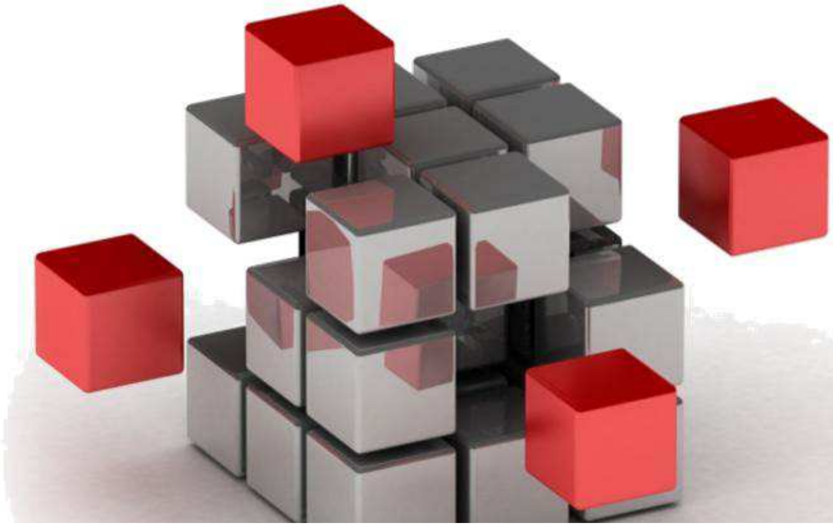
Belacel, Nabil; Durand, Guillaume

A new framework of operation research and learning path recommendation for next-generation of e-learning services

Nabil Belacel, Guillaume Durand
National Research Council Canada
July 15th, 2015
EURO2015

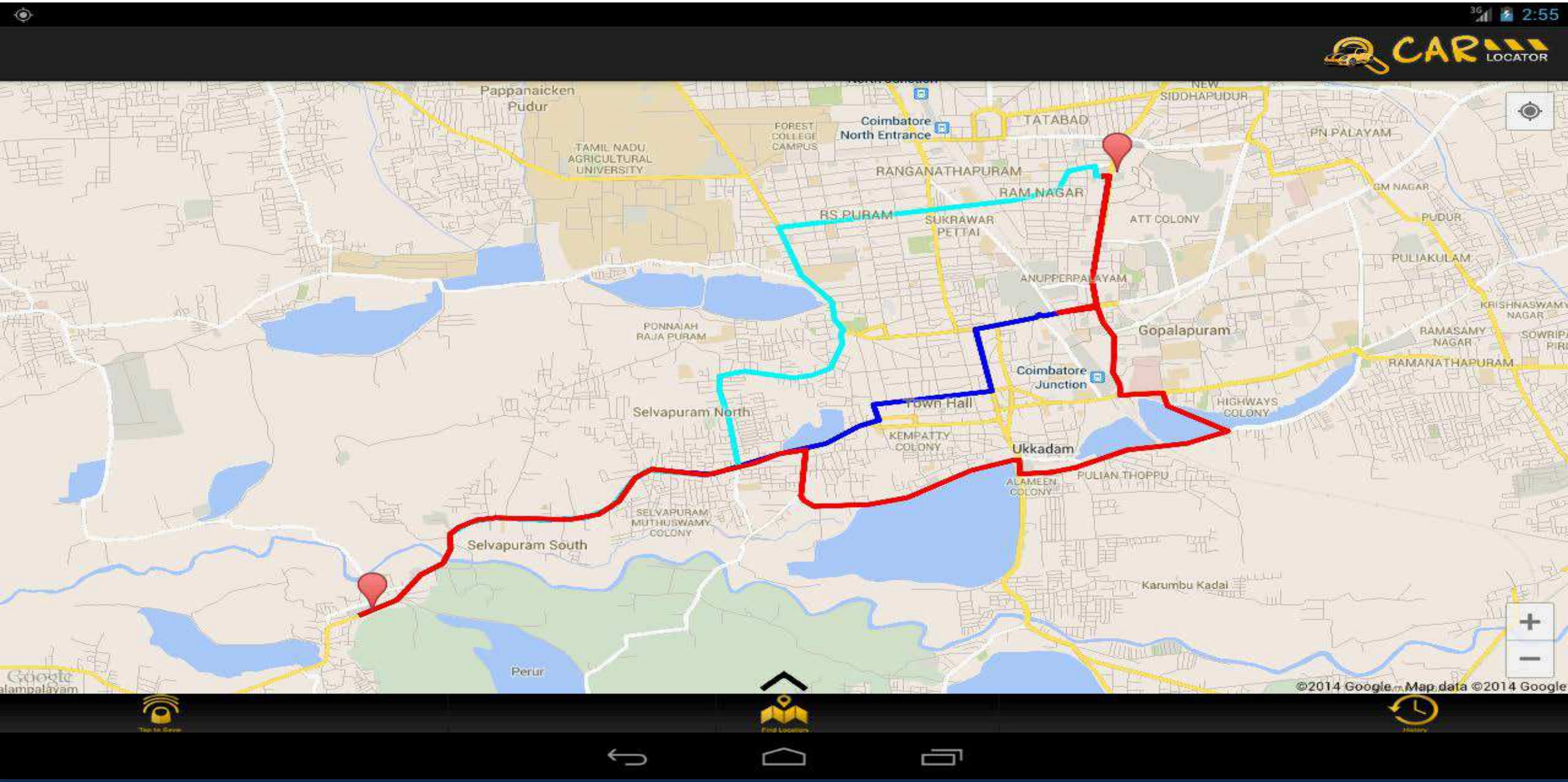
Outline

- Introduction
  - Learning Design concept, and challenges
- Proposed graph model and initial solutions
  - Graph Model
  - Induced sub-graph
- BIP Solver
- Example
- Discussion



Introduction

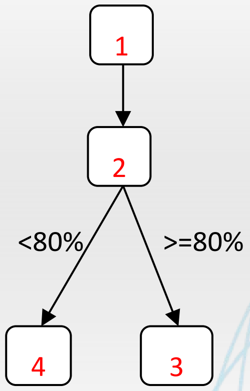
Car navigation system ←→ Learning path



A learning design (LD) is a learning path, a **sequence** of ordered learning objects.

Example:

1. Read article A
2. Take a quiz
3. Do the lab
4. Read supplementary material S



'A teacher preparing a course is a learning designer, and learning design could be as simple as the activity of preparing a course.'

Introduction

Definitions:

- A competency is “an observable or measurable ability of an actor to perform a necessary action(s) in given context(s) to achieve a specific outcome(s)” (ISO 24763)

- A learning object (LO) is any digital resource that can be reused to provide a competency gain.

Model and initial solutions

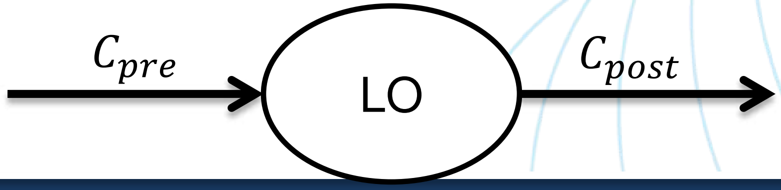
Graph Model

Personalized Learning path:

- Let G = (V, E) be a directed graph
- V (vertex/node): learning object set,
- E (arc): competency dependencies
- Arc(u, v): the LO *v* is accessible from *u* (Two nodes are connected if there exist a dependency relation, such that one node is a prerequisite to the other.).

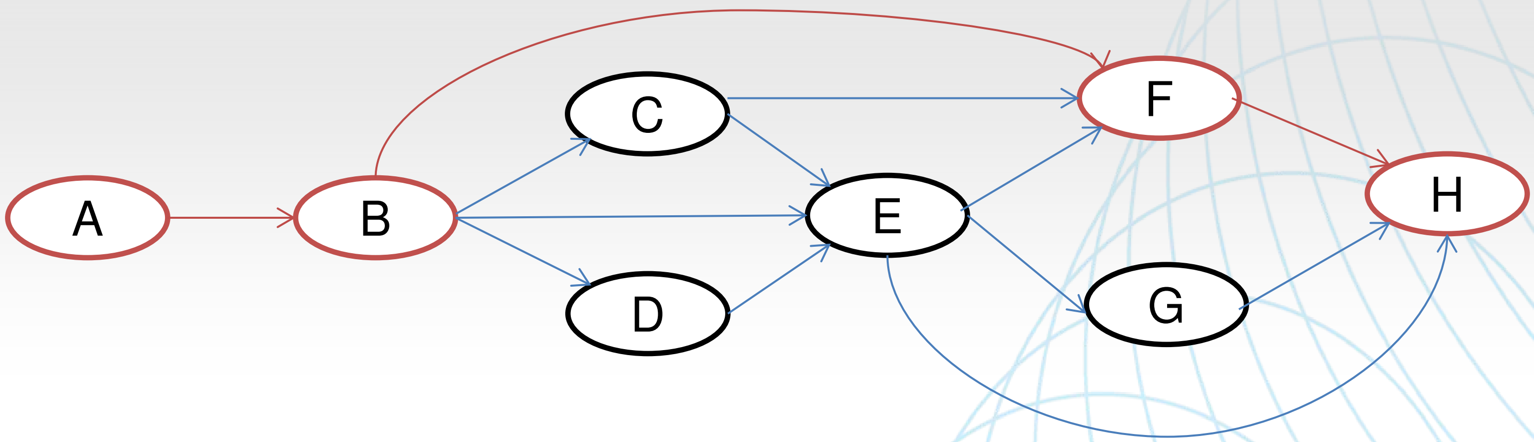
For each vertex, we have:

- C_{pre} is a set of the competencies required by vertex *v*
- C_{post} is a set of competencies offered by vertex *v*



Personalized Learning path:

- A learning path is a path that starts from the initial knowledge of the learner and ends at the target knowledge.

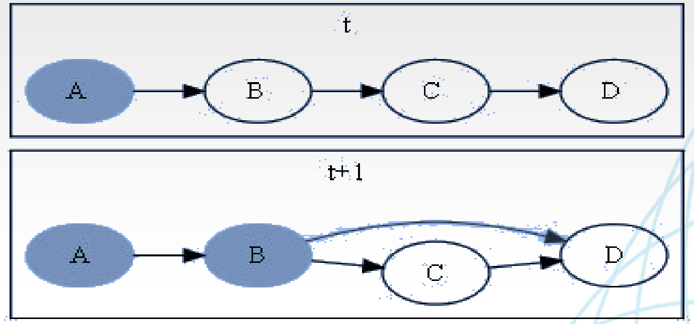


- (A→B→F→H is the optimised personalized learning path.

Model and initial solutions

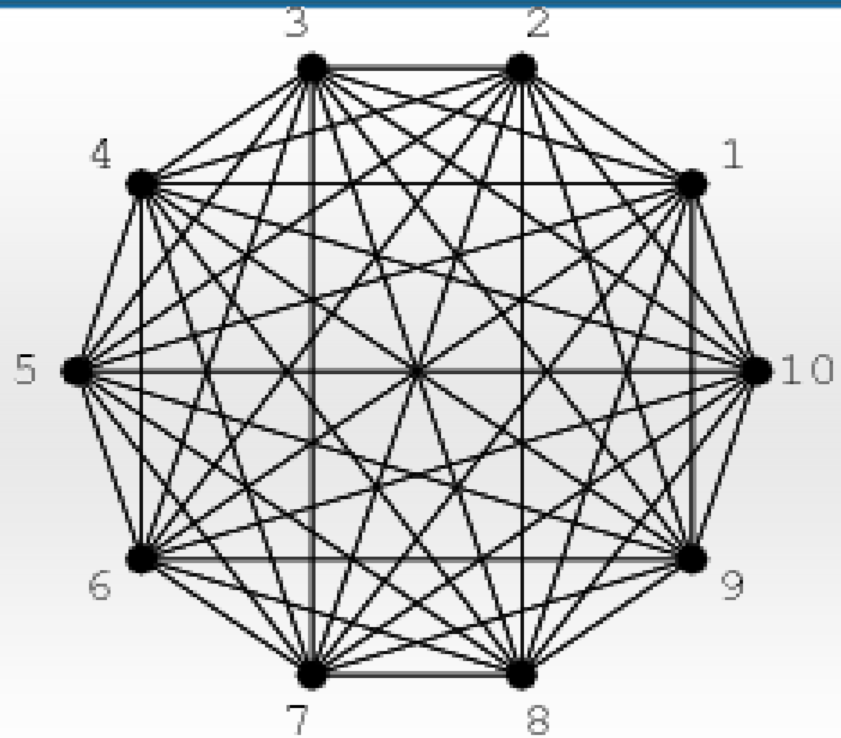
Graph Model

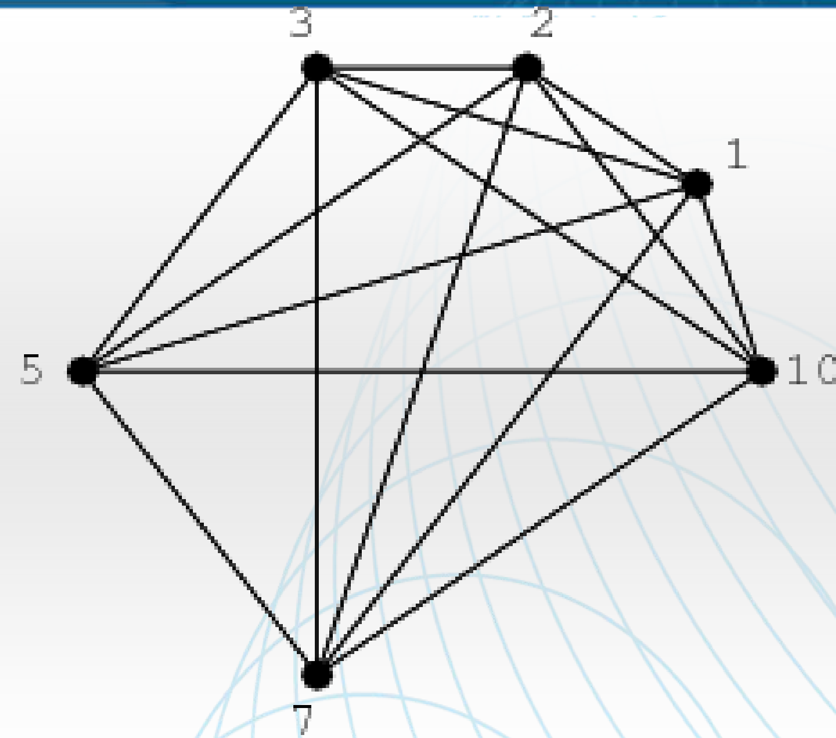
- C_{pre} is a set of the competencies required by vertex *v*
- C_{post} is a set of competencies offered by vertex *v*
- $C_{pre}(v) \subseteq C_{post}(u) \Rightarrow Arc\{u, v\}$
- $Arc\{u, v\} \Leftrightarrow C_{pre}(v) \subseteq C_{post}(u) \cup C_{learner}(t)$



- LO can bring competencies that could be among the prerequisites of future learning objects

Induced Subgraph
Reducing the solution space





- An induced sub-graph H of graph G is a graph whose vertex set is a subset of G's vertex set, and whose edges between vertices are kept from G.

- An induced sub-graph that is a complete graph is called a clique.

- Any induced subgraph of a complete graph forms a clique.

NRC·CNRC

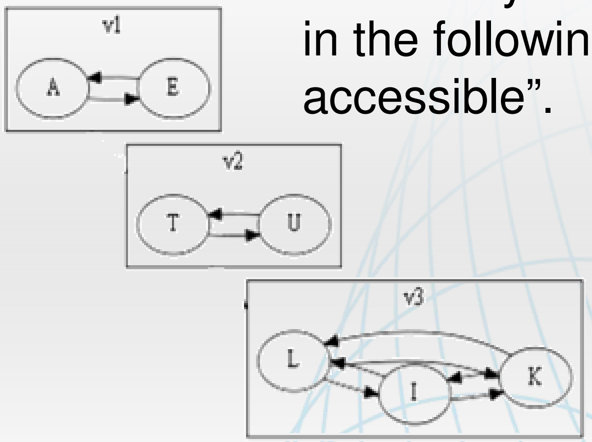
Model and initial solutions

Cliques as a graph reducer

	β_6	
v_1	A^{6}_{5} $E^{6}_{3,5}$	$\uparrow$ 6
v_2	$T^{3,2,4}_{7}$ U^{5}_{0}	$\uparrow$ 3,5
v_3	$L^{0,7}_{8,9}$ I^{7}_{9} K^{0}_{8}	$\uparrow$ 0, 7
	$\alpha^{8,9}$	$\uparrow$ 8, 9

α: Fictitious LO with initial learner competency state
β: Fictitious LO with targeted learner competency state
$LO^{\text{list of gained competencies}}$ $LO_{\text{list of prerequisite competencies}}$

“if every learning object in the clique is completed, then every learning object in the following clique is accessible”.



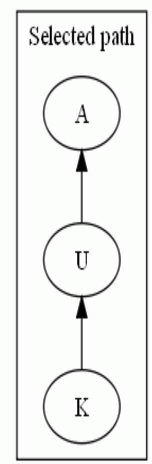
targetClique = new clique with only the target learning object
clique = *targetClique*
while *clique*’s prerequisites are not a subset of the learner’s competencies
    preClique = a new clique with all learning objects leading to any of *clique*’s prerequisites
    if *preClique*’s prerequisites contain all of *clique*’s prerequisites AND are not a subset of the learner’s competencies
        break, as an infinite loop would ensue
    clique = *preClique*

NRC·CNRC

Notation:

- Let $Q_{n,m}$, $G_{n,m}$ *matrices (prerequisite and Gained competences of n items and* $C_{n,v}$ is the clique distribution

$$Q_{n=7,m=9} = \begin{array}{cc} & \begin{array}{ccccccccc} 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} A \\ E \\ T \\ U \\ L \\ I \\ K \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$



	β_6	
v_1	$\boldsymbol{A^6{}_5}$ $E^6{}_{3,5}$	$\uparrow$ **6**
v_2	$T^{3,2,4}{}_7$ $\boldsymbol{U^5{}_0}$	$\uparrow$ **3,5**
v_3	$L^{0,7}{}_{8,9}$ $I^7{}_9$ $\boldsymbol{K^0{}_8}$	$\uparrow$ **0,7**
	$A^{8,9}$	$\uparrow$ **8, 9**

$$G_{n=7,m=9} = \begin{array}{cc} & \begin{array}{ccccccccc} 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} A \\ E \\ T \\ U \\ L \\ I \\ K \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$C_{n=7,v=3} = \begin{array}{cc} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} A \\ E \\ T \\ U \\ L \\ I \\ K \end{array} & \begin{pmatrix} \mathbf{1} & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & \mathbf{1} \end{pmatrix} \end{array}$$

Model and initial solutions

Theoretical optimal solution

Strategy: minimize the cognitive load to the learner (function degree).

Let $S = \{s_0, s_1, \dots, s_v, s_{v+1}\}$ a solution set (s_i contains at least one learning object).

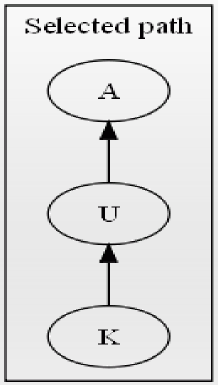
$$\forall s_{i=1..v} \in S, \qquad s_0 = \alpha,\ s_{v+1} = \beta, \qquad Q_{s_i} \subseteq G_{s_{i-1}} \qquad (i)$$
$$\forall\, j = 1 \dots v \neq i = 1 \dots v,\ C_{s_i} \cap C_{s_j} = \emptyset \quad (ii)$$

$$\deg(S = \{s_0, s_1, \dots, s_v, s_{v+1}\}) = \sum_{i=0}^{v+1} \sum_{j=1}^{m} (Q_{s_{i,j}} + G_{s_{i,j}}) \quad (iii)$$

$$\forall s_{i=1..v+1} \in S;\ \exists\, s^*_{i=1..v+1} \in S^*$$
$$\deg(S^* = \{s^*_0, s^*_1, \dots, s^*_v, s^*_{v+1}\}) \leq \deg(S = \{s_0, s_1, \dots, s_v, s_{v+1}\}) \quad (iv)$$

Model and initial solutions

Heuristic solver



	β_6	
v_1	$\mathbf{A^6_5}$ $E^6_{3,5}$	↑ 6
v_2	$T^{3,2,4}_7$ $\mathbf{U^5_0}$	↑ 3,5
v_3	$L^{0,7}_{8,9}$ I^7_9 $\mathbf{K^0_8}$	↑ 0,7
	$\alpha^{8,9}$	↑ 8, 9

The local optimum is considered obtained when the minimum subset of vertices with a minimum “degree”, being the sum of the number of prerequisite competencies and output competencies of the vertex are found.

Starting from targeted competencies.

```
for each prerequisite to satisfy, prerequisite
  selectedObject =  a blank object whose degree = ∞
    for each learning object in the clique, object
      if object is already in localOptimum continue to next prerequisite
      else if object produces prerequisite AND object's degree < selectedObject's degree
        selectedObject = object
  localOptimum.add(selectedObject)
return localOptimum
```

Model and initial solutions

Heuristic solver

	β_6	
v_1	M^6_5 $N^{6,7}_4$	↑ 6
v_2	$O^5_{3,9}$ P^4_8	↑ 4,5
v_3	T^8_7 Y^9_7, Z^3_7	↑ 3, 9, 8
	α^7	↑ 7

Heuristic solver result:
$\alpha, Y, Z, O, M, \beta$

$$\deg(\alpha, Y, Z, O, M, \beta) = 1 + 2 + 2 + 3 + 2 + 1 = 11$$
$$\deg(\alpha, T, P, N, \beta) = 1 + 2 + 2 + 3 + 1 = 9$$

BIP Solver

Binary integer programming (BIP) as follows:

Minimize:

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{m}(Q_{i,j}+G_{i,j})\,x_i\right) = \deg(X) \qquad (1)$$

Subject to:

$$Q_{i,j}x_i - \left(\sum_{k=1}^{i-1} G_{k,j}x_k\right) \times Q_{i,j} \le 0 \qquad (2)$$

$$for \quad i = 2,\dots,n-1; \quad for \quad j = 1,\dots,m; \quad x_i \in \{0,1\};$$

X = {x_i, *i=1,…,n*}, are the decision variables such that:

$$x_i = \begin{cases} 1 & if\ the\ item\ i\ is\ selected; \\ 0 & otherwise, \end{cases} \qquad (3)$$

We suppose that x_1 = 1 and x_n = 1, knowing that:

$$x_1 = 1\ presenting\ the\ initial\ item\ \alpha$$
$$and\ x_n = 1\ presenting\ the\ resulting\ item\ \beta$$

The function (1) represents the total number of prerequisite and gained competencies to be minimized.

The constraints (2) states that if the item *i* has competency *j* as one of its prerequisite competencies; the competency *j* should be gained from the items on the learning path (1,…, *i*-1)

NRC·CNRC

Example

BIP solver

Minimize :

$$\deg(X) = 2x_2+2x_3+2x_4+3x_5+2x_6+2x_7+3x_8$$

Subject to:

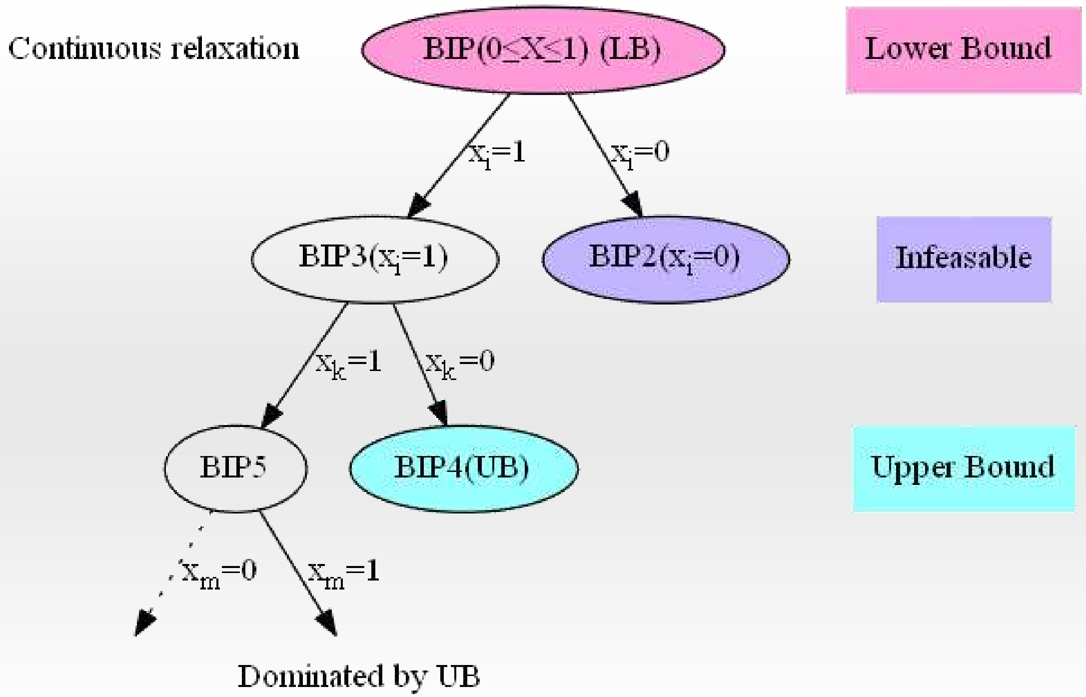
$$x_5 - x_3 \leq 0$$
$$x_5 - x_4 \leq 0$$
$$x_6 - x_2 \leq 0$$
$$x_7 - x_5 \leq 0$$
$$x_8 - x_6 \leq 0$$
$$-x_7 - x_8 \leq -1$$
$$x_i \in \{0,1\}, i = 2, \dots, 8$$

	β_6	
v_1	M^6_5 $N^{6,7}_4$	↑ 6
v_2	$O^5_{3,9}$ P^4_8	↑ 4,5
v_3	T^8_7 Y^9_7, Z^3_7	↑ 3, 9, 8
	α^7	↑ 7

Decision Variables	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
LO	α	T	Y	Z	O	P	M	N	β

Example

Branch and Bound solver

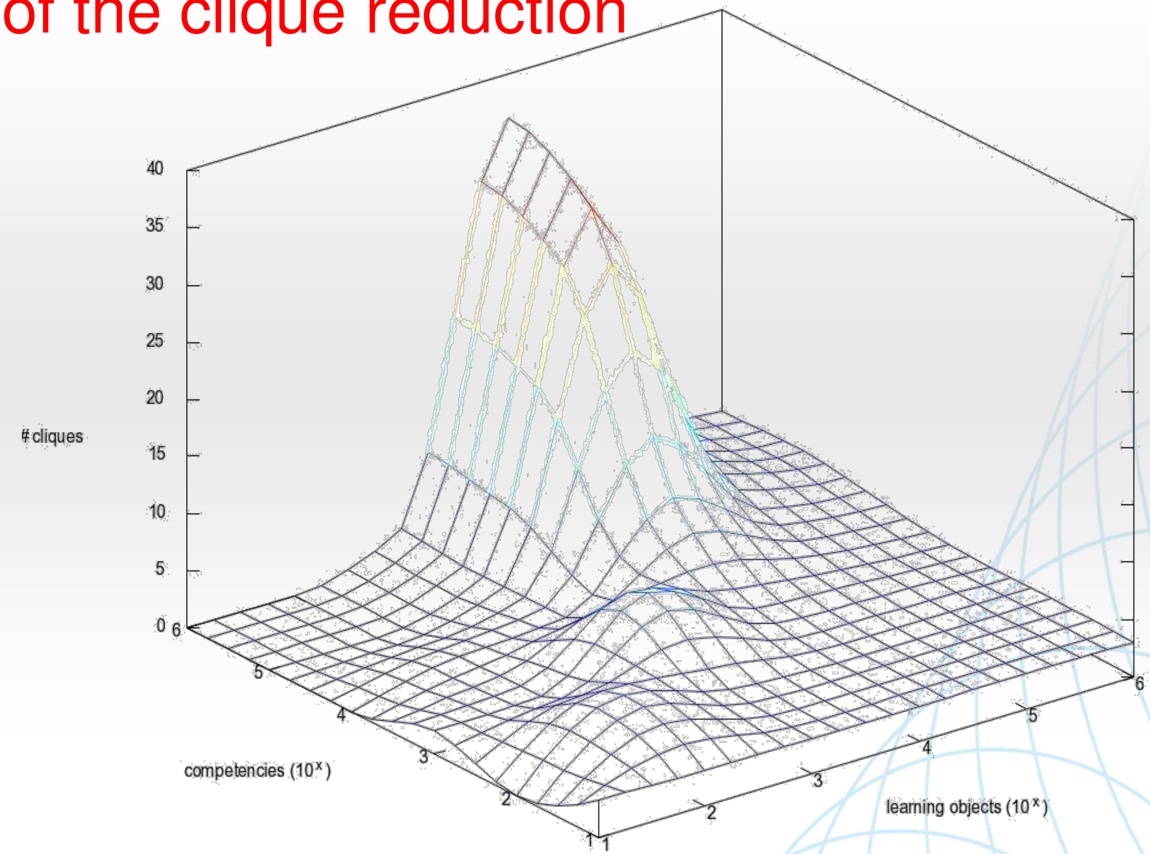


Simplex method to LP-relaxation of the example gave an integral lower bound solution (fathomed)

Decision Variables	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
LO	α	T	Y	Z	O	P	M	N	β
X*	1	1	0	0	0	1	0	1	1

Discussion

- Benefit of the clique reduction

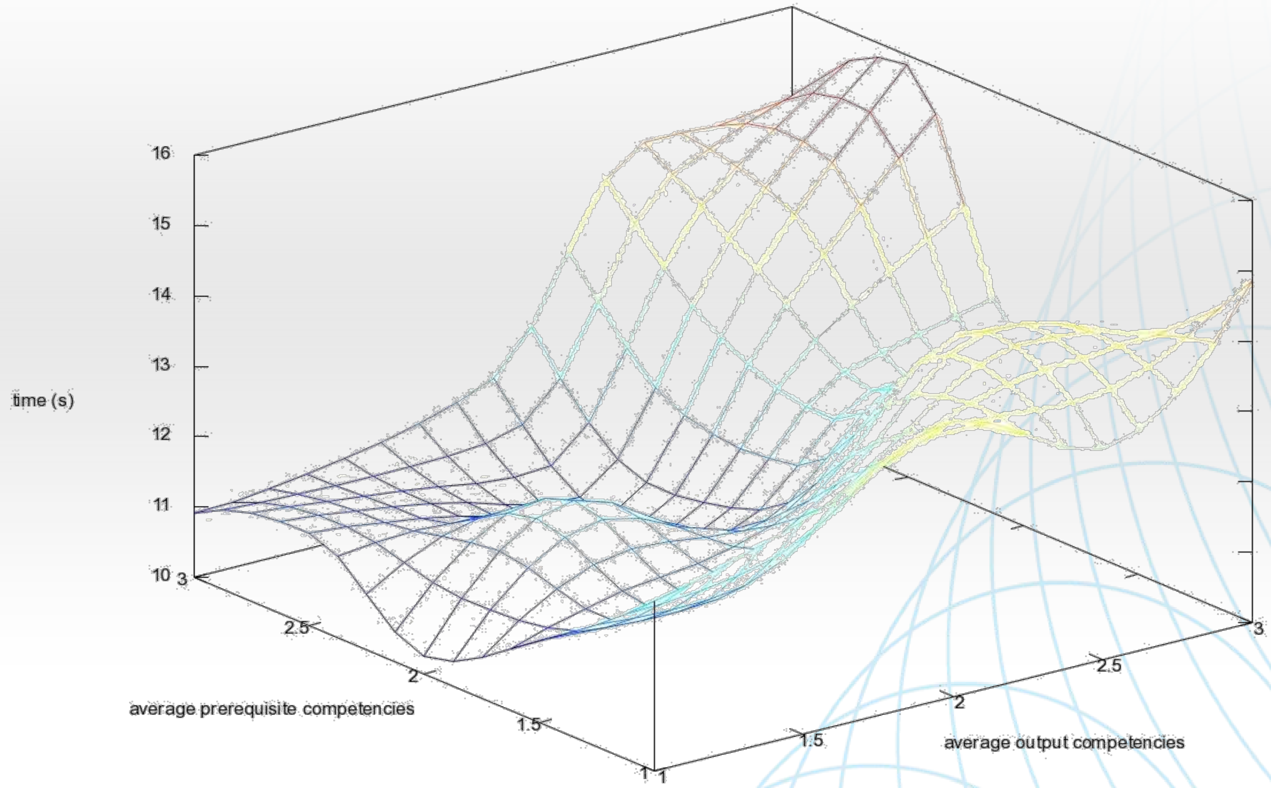


Average Number of Cliques on Calculated Learning Path Given 1 to 2 Output Competencies and 1 to 6 Prerequisites Competencies per Learning Object

NRC·CNRC

Discussion

- Local vs global optimal performance



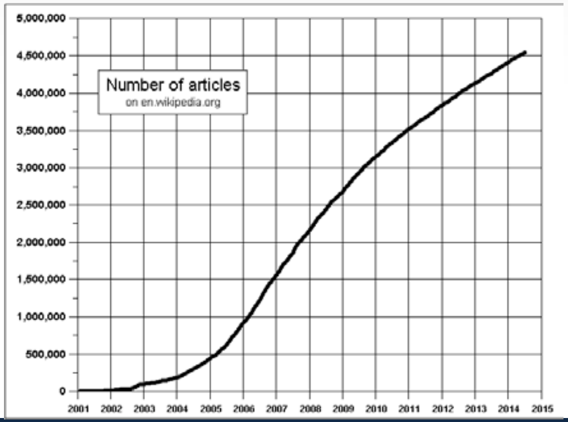
Average Calculation Time of Learning Paths Given 10^5 Learning Objects and 10^4 Competencies

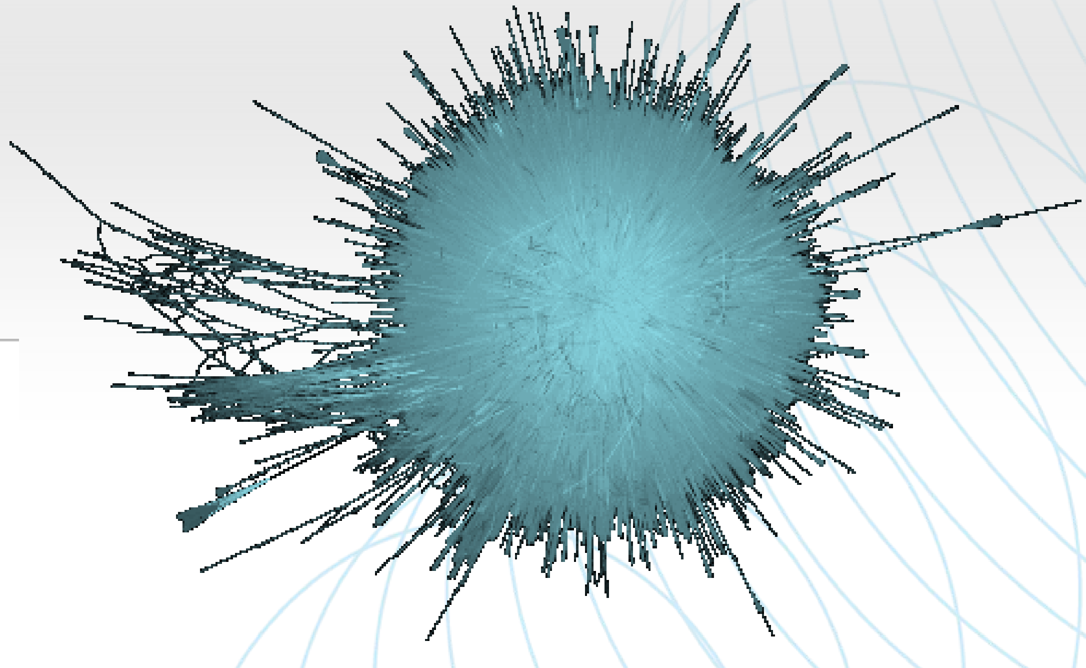
NRC·CNRC

Conclusion

Challenges

- Require a teacher/expert:
- Human capacity of processing information…





Gleich@wikipedia-20051105. 2872475 nodes, 19716499 edges.

References

- Guillaume Durand, Nabil Belacel, François LaPlante (2013) Graph theory based model for learning path recommendation, *Information Sciences*, Volume 251, 1:10-21.

- Belacel, N., Durand, G., Laplante, F. A binary integer programming model for global optimization of learning path discovery (2014) CEUR Workshop Proceedings, 1183, pp. 6-13.