caseine.org

CaseInE

A community of teachers for an active pedagogy in OR

Grenoble, France

Nadia Brauner, Hadrien Cambazard, Nicolas Catusse

A training plateform

- Increase engagement and autonomy of students
- Better use of teacher time
- Improve the quality of the contents
  - (sharing = reviewing from others)
- Improve visibility of the contents
  - (communication)

caseine.org

Menu

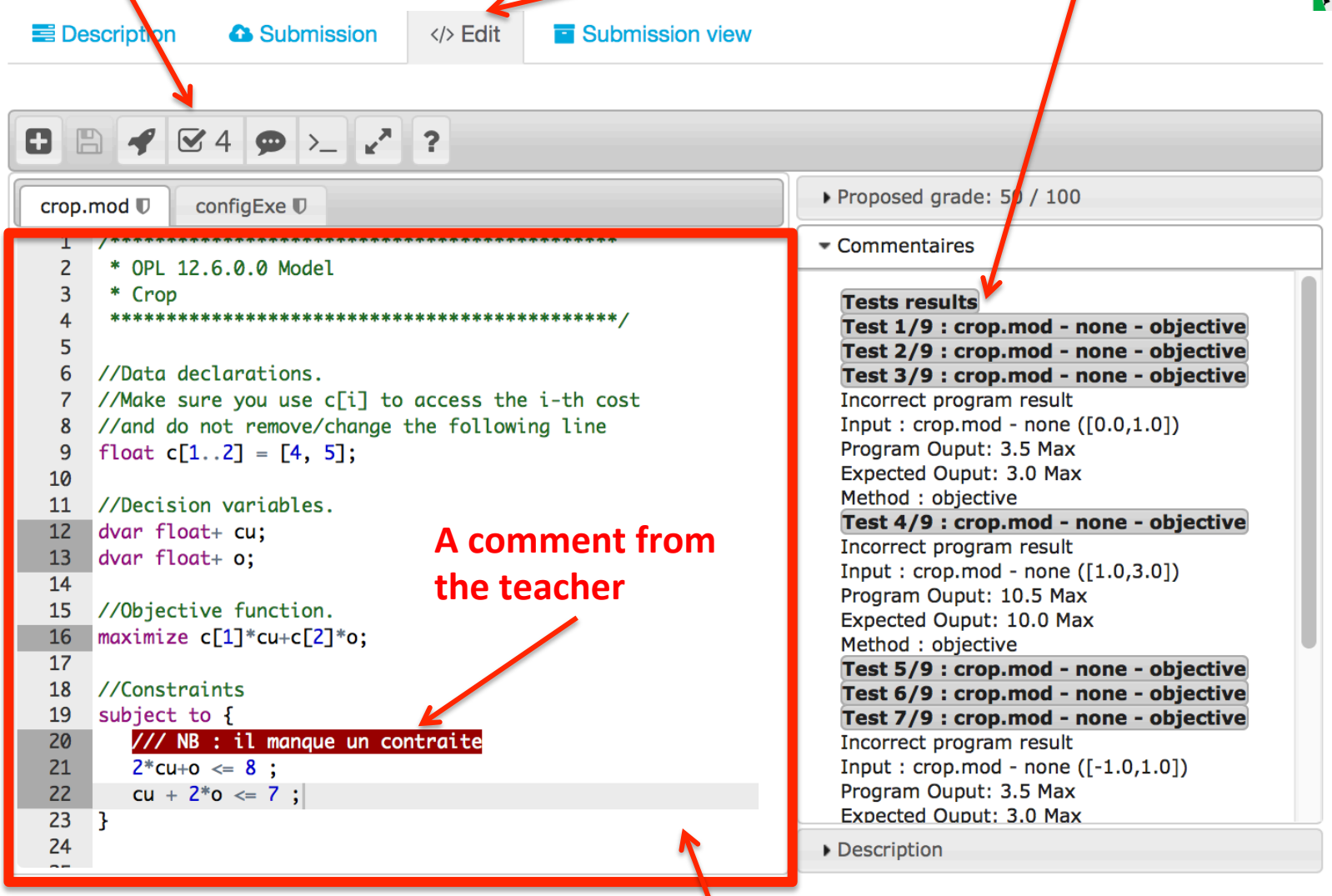
- Automatic evaluation
  - Linear programming
  - Mixed Integer Programming
  - Dynamic programming
  - Graph algorithms
  - Others : Java, Python, C, R…
- An environment for the students
- A community sharing resources
  - The principle
  - How to join?

A programming activity: Student's point of view

To run the evaluation

To edit the program

Result of the evaluation



A comment from the teacher

VPL

The program of the student

Automatic evaluation

The teacher: describes the exercice

The student:

- enters the code/model
- lauches the evaluation
- gets the results

The teacher:

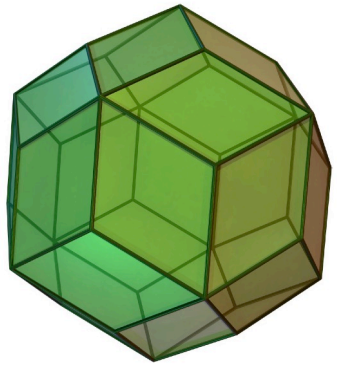
- can access the code
- can comment the code

Automatic evaluation

- Based on VPL tool
  - A Moodle plugin
  - vpl.dis.ulpgc.es
- Used on Caseine for
  - LP, MIP models
  - CP models
  - Dynamic programming
  - Graph algorithms and data structures
  - Basic and advanced programming
    - Java, Python, C…
  - R

Evaluate an LP model

- Check the vertices of the polyhedron defined by the constraints
- Give information to the student
- Semi-automatic generation of tests



Evaluate an LP model

crop.mod | configExe

```
1  /*********************************************
2   * OPL 12.6.0.0 Model
3   * Crop
4   *********************************************/
5
6  //Data declarations.
7  //Make sure you use c[i] to access the i-th cost
8  //and do not remove/change the following line
9  float c[1..2] = [4, 5];
10
11 //Decision variables.
12 dvar float+ cu;
13 dvar float+ o;
14
15 //Objective function.
16 maximize c[1]*cu+c[2]*o;
17
18 //Constraints
19 subject to {
20    /// NB : il manque un contraite
21    2*cu+o <= 8 ;
22    cu + 2*o <= 7 ;
23 }
24
```

Proposed grade: 50 / 100

Commentaires

Tests results
Test 1/9 : crop.mod - none - objective
Test 2/9 : crop.mod - none - objective
Test 3/9 : crop.mod - none - objective
Incorrect program result
Input : crop.mod - none ([0.0,1.0])
Program Ouput: 3.5 Max
Expected Ouput: 3.0 Max
Method : objective
Test 4/9 : crop.mod - none - objective
Incorrect program result
Input : crop.mod - none ([1.0,3.0])
Program Ouput: 10.5 Max
Expected Ouput: 10.0 Max
Method : objective
Test 5/9 : crop.mod - none - objective
Test 6/9 : crop.mod - none - objective
Test 7/9 : crop.mod - none - objective
Incorrect program result
Input : crop.mod - none ([-1.0,1.0])
Program Ouput: 3.5 Max
Expected Ouput: 3.0 Max

Description

Evaluate a MIP model

matching.mod | configExe | InstanceA.dat

```
1  /*********************************************
2   * OPL 12.6.0.0 Model
3   * The maximum matching problem
4   * authors: Olivier briant and Hadrien cambazard
5   *********************************************/
6
7  //Data
8  int   n = ...;           // number of vertices
9  range vertices = 1..n;
10 int   w[vertices][vertices] = ...;   // the weight matrix
11 int   adj[vertices][vertices] = ...; // the adjacency
12
13 //Variables
14 dvar boolean x[vertices][vertices] ;
15
16 //Objective
17 maximize sum(i in vertices) sum(j in vertices) x[i][j]*w[i][j];
18
19 //Constraints
20 subject to {
21     forall(i in vertices) {sum(j in vertices) (x[i][j]*adj[i][j]+x[j][i]*adj[j][i]) <= 1;}
22     forall(i in vertices) forall(j in vertices) x[i][j] <= adj[i][j] ;
23 }
24
25 /* Show the solution */
26 execute {
27   writeln("Post-traitement: ");
28   writeln("La valeur de l'objectif est de "+cplex.getObjValue());
29   //TODO: make sure you print the solution in the console to check it is a tour
30 }
```

External Data
Forall, sum…
Execution control

Evaluate dynamic programs

- In Java
- Tests in Junit
- Complexity check
  - Time control: Distinguish pseudo-polynomial from complete enumeration
  - More precise ?
- Backtrack check

Evaluate dynamic programs

t_i		4	10	7	5	3
q \ i	**0**	**1**	**2**	**3**	**4**	**5**
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	3
4	0	4	4	4	4	4
5	0	4	4	4	5	5
6	0	4	4	4	5	5
7	0	4	4	7	7	7
8	0	4	4	7	7	8
9	0	4	4	7	9	9
10	0	4	10	10	10	10
11	0	4	10	11	11	11
12	0	4	10	11	12	12
13	0	4	10	11	12	13
14	0	4	14	14	14	14

t_i		4	10	7	5	3
q \ i	**0**	**1**	**2**	**3**	**4**	**5**
0	-	-	-	-	-	-
1	-	1	1	1	1	1
2	-	2	2	2	2	2
3	-	3	3	3	3	0
4	-	**0**	**4**	4	4	4
5	-	1	5	5	0	5
6	-	2	6	6	1	6
7	-	3	7	0	7	4
8	-	4	8	1	8	5
9	-	5	9	2	4	9
10	-	6	0	10	10	7
11	-	7	1	**4**	**11**	11
12	-	8	2	5	7	9
13	-	9	3	6	8	10
14	-	10	4	14	**14**	**11**

Evaluate dynamic programs

BalancingData.java | PdynSolver.java | Order.java | Executable.java

```
51
52      /**
53       * Solve the maximum load problem:
54       */
55      public void solve() {
56          this.init();
57          int w = data.getC() ;
58          int n = data.getN() ;
59
60          for (int i=0; i<n+1 ; i++)
61              f[0][i] = 0 ;
62
63          for (int j=0; j<w+1 ; j++)
64              f[j][0] = 0 ;
65
66          for (int i = 1 ; i < n+1 ; i++)
67              for (int j=1; j < w+1 ; j++){
68                  int di = (data.getOrder(i)).getDuration() ;
69                  f[j][i] = Math.max(f[j][i-1] , (j>=di)?f[j-di][
70                  if (f[j][i]==f[j][i-1]) p[j][i]=j; else p[j][i]
71              }
```

▸ Proposed grade: 0 / 100

▾ Commentaires

Tests results
Test 1/13 : Vpl_tests.testSPInstanceA
Incorrect program result
Input : hidden
Program Ouput: 0
Expected Ouput: hidden
Method : testSPInstanceA
Test 2/13 : Vpl_tests.testSPInstanceB
Incorrect program result
Input : hidden
Program Ouput: 0
Expected Ouput: hidden
Method : testSPInstanceB
Test 3/13 : Vpl_tests.testSPInstanceC
Incorrect program result
Input : hidden
Program Ouput: 0

▸ Execution

▸ Description

DP/Load balancing

Evaluate graph algorithms

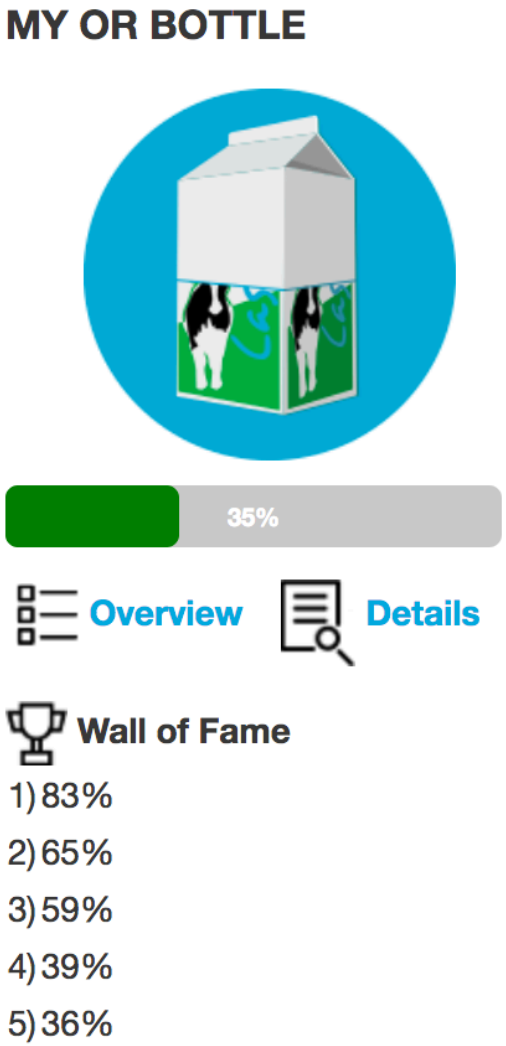
- In Java
  - (and some in Python)
- Tests in Junit
- Classical algorithms
  - Simple problems: max degree, number of connected components…
  - Graph representation
  - BFS, DFS
  - Dijkstra, Kruskal, Ford-Fulkerson…

OR automatic evaluation

- Ideas to test smartly the students program
- Time consuming development -> share
- What's next
  - Other ideas
  - More fluent use for teachers
  - Enhance the collection of exercices
  - Share with broader community

A pedagological environment

- Based on Moodle
- Plugin development
  - VPL questions
  - Completion levels
  - Likes
  - Ski run color



A pedagological environment

- Open to academic community:
  edugain connexion
  - Belnet federation, DFN-AAI…

The teacher community of Caseine

- A shared space *i.e.* a feature to easily:
  - Tag/mark your activities with relevant information for sharing
  - Share your activities
  - Search among the shared activities

- How to join
  - Create an account (your existing academic login might work)
  - Have a look at the opened courses (e.g. OR course)
  - Have a look at the tutorial (key to enter the tutorial: *cincle*)
  - Express your will (send an email) for starting a course and have access to the shared space.

- Support and training for a start…

The teacher community of Caseine

- Access to open courses: **Free** for initial university courses and individual training

- Creating a course
  - Free for initial university courses
  - Contribution to costs for lifelong university training (formation continue)
  - Paying service for companies which sell formations

- Terms
  - Everyone is author of its creations
  - Everyone can choose to share or not
  - Moodle developments (shared plugins)
  - Caseine specific developments

Community animation tools

- Tutorial
  - Discussion Forum
  - FAQ
  - Instructions for Caseine specific features
  - Newsletter link in Tutorial

- **Association** model to financially ensure the continued existence of the platform

  (handled by Grenoble university for the moment)
  - Fees for the hosting, Administration, Development, Maintenance, Support

Variety of usages

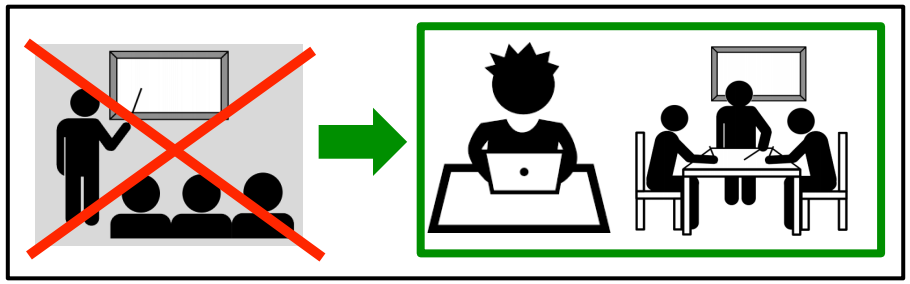
Support for « active classrooms » but as many practices as there are teachers:

Autonomy
Personal work

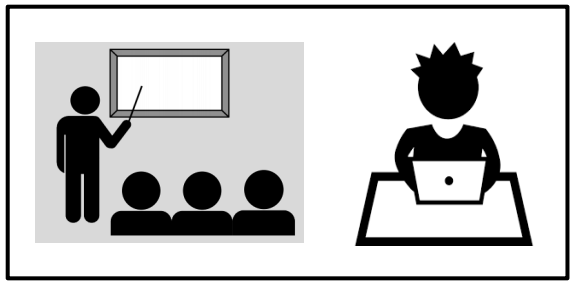
Autonomy
Team work

Autonomy
Personal work at home

Individual
Evaluation



Support to conduct a *flipped* classroom



Traditional classroom
with validation in autonomy

Only one agreement: focus on student's active role

Core team

Hadrien Cambazard
(manager, Java + RO)

Nicolas Catusse
(manager, Java + RO)

Nadia Brauner
(manager, RO)

Fabrice Ménard
(pedagogical ingineer)

Aurélie Lagoutte
(Python)

Pierre Lemaire
(R)

Bernard Penz
(RO)

Julie Peyre
(Java)

Anne-Laure Ladier
(RO)

Christophe Saint-Marcel
(Design Pattern)

Denis Bouhineau
(Algo/Prog)

Céline Fouard
(Language for the Web)

Quantities in 17-18

- Academic use
  - 1200 active students
  - 14 bachelor / 11 master training programs
  - 5 universities (in courses)
    - UGA, Grenoble INP, INSA de Lyon, Université Clermont Auvergne, Centrale Lille
  - 36 teachers
- + free connexions from everywhere

A training plateform

- Increase engagement and autonomy of students
- Better use of teacher time
- Improve the quality of the contents (sharing = reviewing from others)
- Improve visibility of the contents (communication)

Contact link on top of the main page…