

openGVE / Mascopt

Bibliothèque Java pour les graphes et les
réseaux

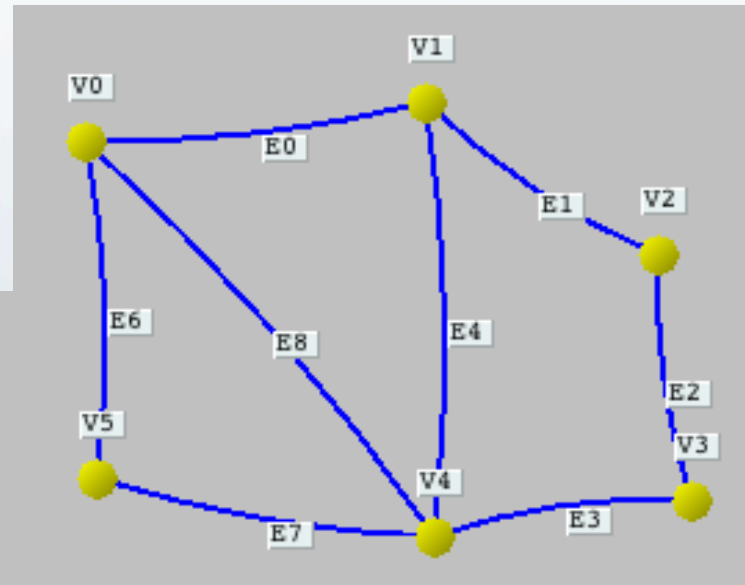
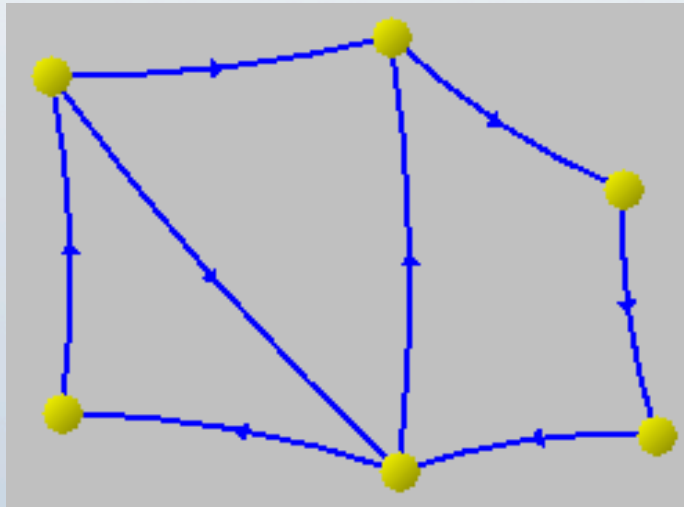


INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



Objectifs : Graphes

- Graphes : $G=(V,E)$
 - Relations sur un ensemble
 - Orienté ou non



Objectifs : Réseaux

- Réseaux de transport
 - Capacités
 - Source / puits





Exemple :

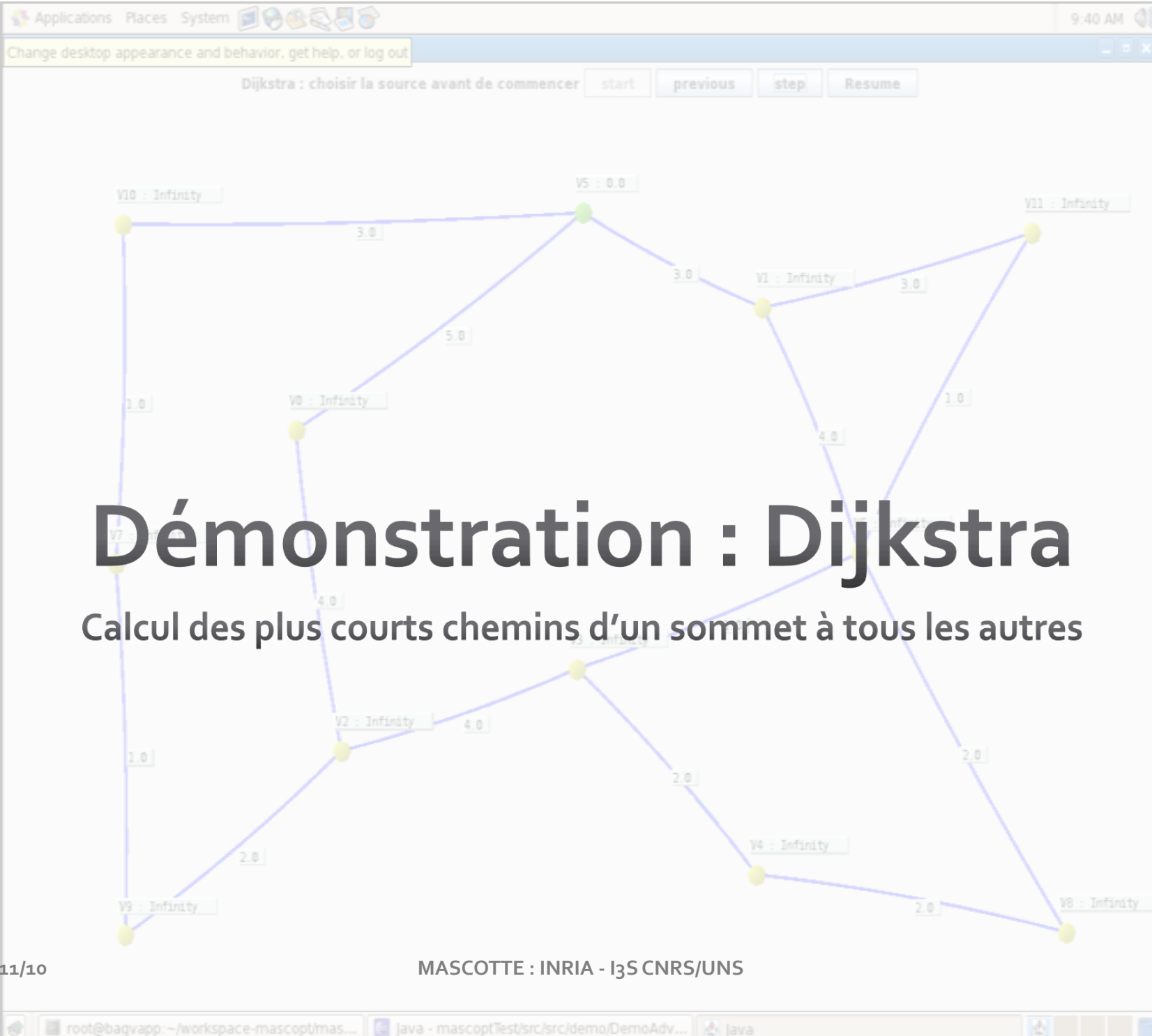
RENATER / DWDM



Objectifs : Algorithmes

- Ecrire des programmes pour valider / tester des algorithmes
- Réutiliser des algorithmes courants : plus courts chemins, arbres couvrants, flots maximum, ...
 - Dijkstra, Bellman Ford
 - Kruskal, Prim
 - Ford Fulkerson, ...

```
▼  A Dijkstra<V, E, G>  
  ● DIJKSTRADISTANCE : String  
  ●  Dijkstra(G, Map)  
  ●  C Dijkstra(G, Map, boolean)  
  ● evaluateFromSource(V) : void  
  ●  △ run() : void  
  ● getDistances() : Hashtable<V, AbstractScalar>  
  ● getDistanceTo(Object) : int  
  ● getShortestPathTo(V) : Path<V, E>  
  ● getStartNode() : V
```



Démonstration : Dijkstra

Calcul des plus courts chemins d'un sommet à tous les autres

mascoptLib

Histoire / contributeurs

- 2001 : suite à PORTO
- Michel Syska
- Bruno Bongiovanni
- Jean-François Lalande
- Yann Verhoeven
- Fabrice Peix

Caractéristiques

- Java
- LGPL
- I/O structurées XML
- Interfaces JNI pour la programmation linéaire
- Spécialisé « backbone »

Pourquoi une nouvelle bibliothèque ?

- Pas de solution disponible à l'époque
- LEDA C++ non libre
- Nombreuses bibliothèques spécialisées Graph Layout / Graph Drawing
- Besoin d'interface avec l'API CONCERT CPLEX
- Aujourd'hui : voir JUNG

Exemple de création de graphe

```
// création des noeuds
Vertex n0 = new Vertex();
Vertex n1 = new Vertex();
Vertex n2 = new Vertex();

// création des arcs entre les noeuds
Arc a0 = new Arc(n0, n2);
Arc a1 = new Arc(n1, n2);

// création d'un ensemble de noeuds V
VertexSet V = new VertexSet();

// on ajoute les noeuds dans l'ensemble de noeuds
V.add(n0); V.add(n1); V.add(n2);

// création d'un ensemble d'arc E
ArcSet E = new ArcSet(V);
E.add(a0); E.add(a1);

// création d'un Digraph graph=(V,E)
DiGraph graph = new DiGraph(V, E);
```


Format MGL

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ... mgl_v2.0.dtd>
<OBJECTS xmlns:xsi= ...
  <GRAPHS>
    <GRAPH id="G0">
      <NAME>G1</NAME>
      <EDGE_SET_REF idref="ES0"/>
    </GRAPH>
  </GRAPHS>
  <SETS>
    <EDGE_SET id="ES0">
      <EDGE_REF idref="E8"/>
      <EDGE_REF idref="E1"/>
      ...
    <VERTEX_SET id="VS0">
      <VERTEX_REF idref="V0"/>
      <VERTEX_REF idref="V1"/>
    </VERTEX_SET>
  <LINKS>
    <EDGE color="-1" id="E8">
      <VERTEX_REF idref="V4"/>
      <VERTEX_REF idref="V0"/>
    </EDGE>
```

Difficultés et Avantages

- Programmation **très** orientée objet
- Utilisation poussée de l'abstraction, des interfaces
- Un choix possible dans l'implémentation des graphes
 - des graphes lourds et complets
 - des graphes minimaux économes et rapides

Architecture logicielle verticale

Mascot est découpée **verticalement** pour faciliter l'intégration de différentes implémentations et maximiser la réutilisabilité:

- Interface "Bridge" de définition des graphes (PACA+ Brésil + Région Centre)
- Classes abstraites "Mascot" de comportements communs de graphes Niçois
- Classes concrètes de graphes de Sophia Antipolis

```

public interface Graph<V, E extends Link<V>> {
    public boolean addEdge(V o1, V o2);
    public HierarchicalSet<V> neighborhood(V v);
    public Graph<V, E> complement(); ... }

public abstract class MascoptAbstractGraph <E extends
MascoptAbstractLink> implements Graph<MascoptVertex, E> {
    public boolean addEdge(MascoptVertex o1, MascoptVertex o2);
    public MascoptVertexSet vertexSet();
    public Iterator<MascoptVertex>
        depthFirstIterator(MascoptVertex root);
    ... }

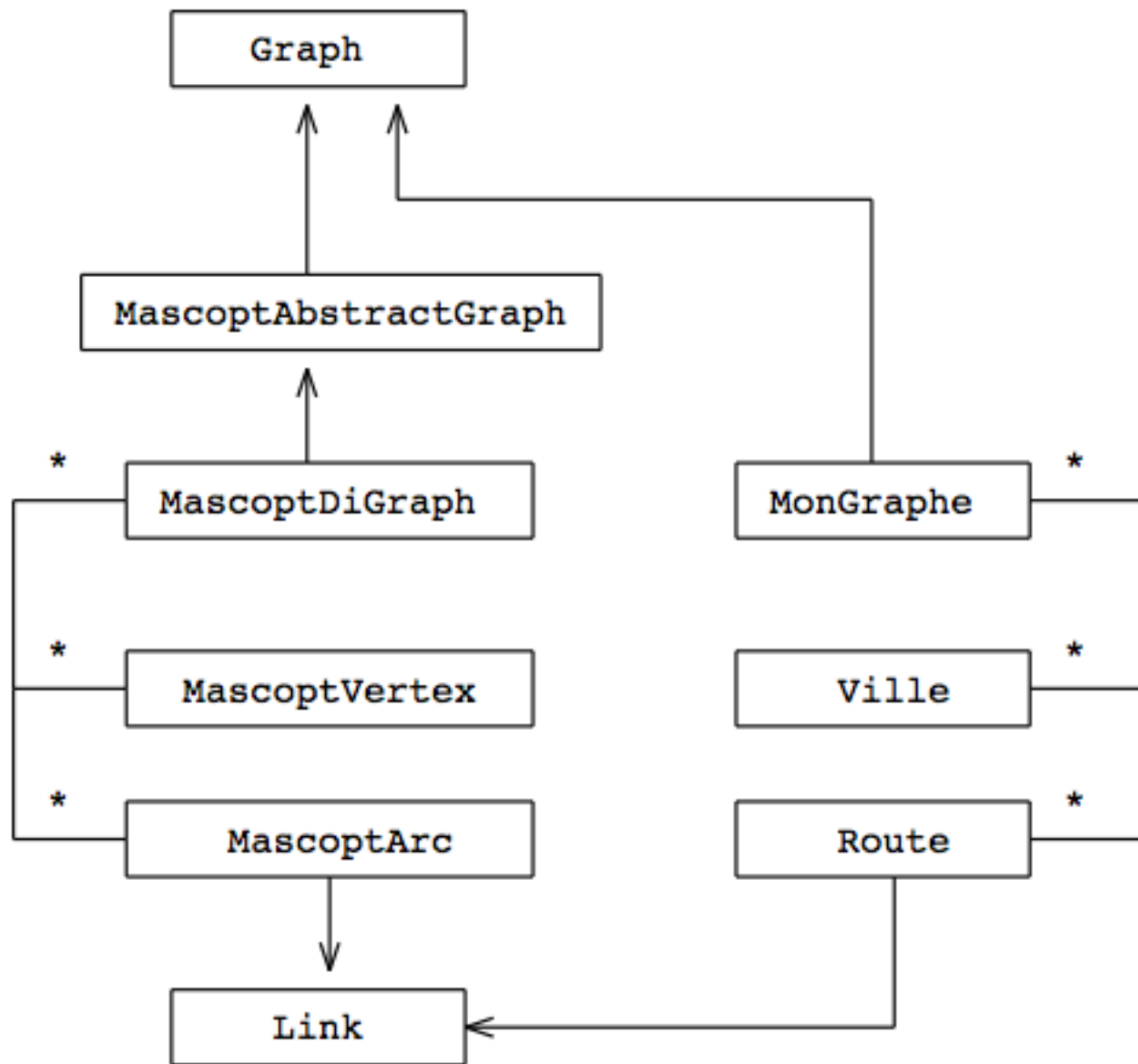
public class MascoptDiGraph
    extends MascoptAbstractGraph<MascoptArc>
    implements Graph<MascoptVertex, MascoptArc>

    public MascoptDiGraph(MascoptArcSet arcSet) {
        this(arcSet, new MascoptDirectedMetaFactory());
    }
    ... }

```

Exemple

```
MascotDiGraph grille = new MascotDiGraph(  
    new MascotArcSet(new MascotVertexSet()));  
for (...  
    grille.addVertex(vertices[i][j]);  
  
for (...  
    newArc =  
        new MascotArc(vertices[i][j], vertices[i+1][j]);  
    grille.addEdge(newArc);  
  
AllToAllLp lp = new AllToAllLp(grille);  
lp.compute();
```

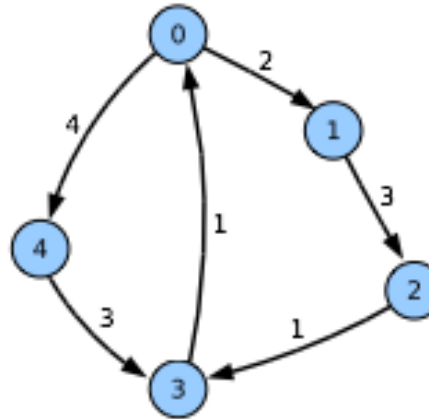


Architecture logicielle horizontale

Découpage **horizontal** pour faciliter l'ajout d'outils complémentaires :

- Bibliothèques de graphes
- Bibliothèques de programmation linéaire
- Bibliothèques graphiques
- Bibliothèques d'algorithmes
 - de génération de graphes
 - de flots
 - de parcours et transformations
- Bibliothèques d'entrées/sorties

Programmation Linéaire



Minimize $2x_{01} + 4x_{04} + 3x_{12} + x_{23} + x_{30} + 3x_{43}$

s.t. :

$$\begin{aligned}x_{30} - x_{01} - x_{04} &= -1 \\x_{01} - x_{12} &= 0 \\x_{12} - x_{23} &= 0 \\x_{23} + x_{43} - x_{30} &= 1 \\x_{04} - x_{43} &= 0 \\x_{01}, \dots, x_{43} &\geq 0\end{aligned}$$

Solveurs de PL

- CPLEX IBM ILOG
- XPRESS
- CPL/CBC <http://www.coin-or.org/>
- GLPK <http://www.gnu.org/software/glpk>
- LP_SOLVE <http://sourceforge.net/projects/lpsolve/>

API, AMPL, ...

Exemple de lpsolve

```
-bash-4.0$ cat pcc_unitaire_0_3.lp
```

```
min: x_0_1 + x_0_4 + x_1_2 + x_2_3 + x_3_0 + x_4_3;
```

```
x_3_0 - x_0_1 - x_0_4 = -1;
```

```
x_0_1 - x_1_2 = 0;
```

```
x_1_2 - x_2_3 = 0;
```

```
x_2_3 + x_4_3 - x_3_0 = 1;
```

```
x_0_4 - x_4_3 = 0;
```

```
-bash-4.0$ ../lp_solve_4.0/lp_solve pcc_unitaire_0_3.lp
```

```
Value of objective function: 2
```

```
Actual values of the variables:
```

```
x_0_1          0
```

```
x_0_4          1
```

```
x_1_2          0
```

```
x_2_3          0
```

```
x_3_0          0
```

```
x_4_3          1
```

Exemple dans mascopt avec CPLEX

```
private void createCplexConstraints(IloCplex cplex) throws IloException {
    Iterator edgesIt = g.edgeSet().iterator();
    while (edgesIt.hasNext()) {
        MascoptAbstractLink currentEdge = (MascoptAbstractLink) edgesIt.next();
        MascoptVertex v1 = (MascoptVertex) currentEdge.getVertices()[0];
        MascoptVertex v2 = (MascoptVertex) currentEdge.getVertices()[1];

        if (currentEdge.leavesFrom(v1)) {
            IloLinearNumExpr expr = cplex.linearNumExpr();
            expr.addTerm(1.0, (IloNumVar) vertexToCplexVar.get(v2));
            expr.addTerm(-1.0, (IloNumVar) vertexToCplexVar.get(v1));
            cplex.addLe(expr, map.getValue(currentEdge, lengthName, lengthContext)
                .doubleValue());
        }

        if (currentEdge.leavesFrom(v2)) {
            IloLinearNumExpr expr = cplex.linearNumExpr();
            expr.addTerm(1.0, (IloNumVar) vertexToCplexVar.get(v1));
            expr.addTerm(-1.0, (IloNumVar) vertexToCplexVar.get(v2));
            cplex.addLe(expr, map.getValue(currentEdge, lengthName, lengthContext)
                .doubleValue());
        }
    }
    cplex.addEq((IloNumVar) vertexToCplexVar.get(source), 0);
}
```

Quoi d'autre?

- Editeur de graphe (voir démo)
- Step-algo pour réaliser les démonstrations comme celle vue pour Dijkstra

To Do

- Diffusion publique
- Tutoriels
- Deadline : janvier 2011
- Puis : d'autres implémentations « légères » pour les grands graphes
- En attendant :

[mailto : Michel.Syska at inria.fr](mailto:Michel.Syska@inria.fr)