Paths

Table of contents

1. Build Paths on your graphs

The goal of this document is to explain how to use path and dipaths in Mascopt. It is not so quite intuitive to build path. The prinicpal reason is that the paths and dipaths are derivated of the class AbstractGraph such as they can be considered as graphs. This is really usefull because a lot of method have not to be rewritten... But the consequence is that the use of paths and dipaths is a bit more complicated.

The paths and dipaths are derivated from the class AbstractPath as graph and digraphs are derivated from AbstractGraph. In the next sections we use the world path to talk about paths and dipaths.

2. Construct a dipath

A path is made of consecutive . When constructing a path, you must specify the Edge Set which contains all the edges the path can use. In general, you give the edge set of the graph on which you want to build your path. For understanding how works Edge sets, have a look at Mascopt Sets. In fact, the edge set of the path you build is a subset of the edge set of the graph. The direct consequence is that if you remove an edge of the graph and if your path was using this edge, the graph does not remain valid.

```
DiGraph g = ...;
DiPath p = new DiPath(g.getEdgeSet());
```

Then you want to build your empty paths with the edges you have in your graph. You can use the method concat as this:

```
Arc e1 = ...;
Arc e2 = ...;
Arc e3 = ...;
p.concat(e1);
p.concat(e2);
p.concat(e3);
```

In this example, we suppose that the order of edges is e1,e2,e3. So you can't concat e3 before e2 ! (In fact the method concat is returning a boolean indicating if the call has succeeded.

3. How to cover a path ?

Several function may help: you can have the strat vertex and then have the next arc until reaching the end.

```
Vertex current = p.getStart();
```

```
while (current != p.getEnd())
{
  System.out.println("Current vertex: " + current);
  Arc e = p.nextArc(current);
  System.out.println("Current edge: " + e);
  current = p.nextVertex(current);
}
```

4. And I can't do multi-path, isn't it ?

That's not true ! You can do multi path but it's also a quit difficult. You have to merge the two part of a multi path and when covering it, you have to choose one way... First build the two single path, using the instruction below. Then merge the two path which must have the same start vertices and end vertices.

```
DiPath p1 = ...;
DiPath p2 = ...;
boolean ok = p2.merge(p1);
```

To cover a multi path, you have to get the next vertices of one vertices. For example if at vertex n1, two arcs leads to vertex n2 and n3 and then from n2 and n3 you reacg n4, you have to choose the arc n1->n2 or n1->n3. You have the possibility to ask all the vertices next to n1, in our case, the set {n2,n3}. You guessed, we use Edge sets for that !

```
DiPath multi = ...;
Vertex n1 = multi.getStart();
ArcSet as = multi.nextArcSet(n1);
System.out.println("All arcs leaving vertex n1:" + as);
```

5. How to know that my path is a multi path ?

You have a function which says if a path is a multi path:

```
if (p2.isMulti())
{
  System.out.println("This path is a multi path !");
}
else
{
  System.out.println("This path is a mono path !");
}
```

6. Sample

You can find a sample in Mascopt Dev package in mascoptDev/samples/basics. The sample file is UsingPaths.java. It builds some paths and cover it.

```
java UsingPaths
```

```
I've suceeded in merging the mono paths !
A mono path: N0->N2->N3->N4
An other mono path: N0->N2->N4
This path called 'monoPath' is a mono path !
The path called 'multiPath' is a multi path !
Current vertex: N0
Current edge: [N0->N2]
Current vertex: N2
Current edge: [N2->N3]
Current vertex: N3
Current edge: [N3->N4]
Current vertex: N0
Edge possibilities: { [N0->N2] }
Choice of the arc: [N0->N2]
Current vertex: N2
Edge possibilities: { [N2->N3] }
Choice of the arc: [N2->N3]
Current vertex: N3
Edge possibilities: { [N3->N4] }
Choice of the arc: [N3->N4]
```