DynBCN
A Constructive Geometric Constraint-based Dynamic Geometry System

Marta Hidalgo, Robert Joan-Arinyo
Grup d'Informàtica a l'Enginyeria
Universitat Politècnica de Catalunya

Foreword

One of the goals of the *Grup d'Informàtica a l'Enginyeria* of the *Universitat Politècnica de Catalunya* at Barcelona is to conduct research on geometric constraint solving and its applications. Here we offer to the community `dynBCN`, a software prototype of a dynamic geometry system based on a constructive geometric constraint solving technology. So far we are offering a `jar` Java file. In the near future we plan to distribute `dynBCN` under a free software license.

`dynBCN` is one of the outputs developed by graduated students pursuing a PhD degree while collaborating with the *Grup d'Informàtica a l'Enginyeria*. Any ideas and suggestions will be welcome.

1 Resources, downloading and installing

Here we describe the resources required, how to download, install and run `dynBCN`.

1.1 Resources

You must have a Java Virtual Machine installed on your computer. We recommend JVM 1.5 or later versions.

Full functionality of `dynBCN` is achieved under Linux. We have been able to run under Windows 7 examples previously solved the files of which are stored in the `testcases` folder.

1.2 Downloading dynBCN

To download `dynBCN` visit the web page `http://floss.cs.upc.edu` and select

```
Software:  dynBCN, a constraint-based dynamic geometry system.
```

Then you will download the `dynBCN.tar.gz` file selecting

```
Download dynBCN
```

You can also download `dynBCN` visiting the web page `http://www.cs.upc.edu` and selecting

```
Download the dynBCN dynamic geometry system
```

1.3 Installing dynBCN

Move the tar file to the folder where you want to install `dynBCN`, for example

```
dynGeom
```

Then extract files in the `dynBCN.tar` file

```
tar xvf dynBCN
```

Two files will be created, `dynBCN.jar` and `testcases`. `dynBCN.jar` is the `dynBCN` Java jar file, and `testcases` is the folder where files of dynamic geometry problems are stored. Each dynamic geometry problem, say `example`, includes six files. Five of them, `example.dgp`, `example.dis`, `example.dom`, `example.ini` and `example.end` are created by `dynBCN`. One, `example.cnx`, is hand-edited by the user.

1.4 Running dynBCN

Once you are in the folder where `dynBCN` has been installed, just run the Java jar file

```
java -jar dynBCN.jar
```

A window will pop up in your screen. Click on OK and a new window with the graphic user interface will show up.

2 `dynBCN` architecture

The `dynBCN` prototype considers geometric problems in a two dimensional space and includes four main components: the geometric problem editor, the constructive geometric constraint solver, the dynamic simulator and the reachability problem solver. In what follows we describe each component.

2.1 The geometric problem editor

After running the jar file, and closing the pop-up window or clicking on the `OK` button, the program displays the main screen of `dynBCN`. The screen has four different panels: the menu bar, the tool bar, the problem panel and the ruled canvas. In what follows we briefly describe each of them.

2.1.1 The menu bar

Tabs in the menu bar lead to different functionalities featured by `dynBCN`.

- `File`: Includes typical operations on files like `Open, New, Save, Save as, ....`
- `Edit`: Offers the option `Undo, Redo` and the check box `Show element form`. The last one displays on the left side a panel with information corresponding to elements in the geometric problem.
- `Tools`: Preferences can be set from here.
- `Solution`: Selecting the field `Show solution` triggers the geometric constraint solver `solBCN` to solve the geometric problem. If the solver succeeds, a solution instance is shown in a small window framed in green. Otherwise, the window is framed in red.
- `Dynamics`: Offers the operations to interact with `dynBCN`, the dynamic geometry system.
- `Help`: Lists the field `About`.

2.1.2 The tool bar

The tool bar displays the set of tools provided by the constructive geometric constraint solver `solBCN` to define and edit constraint-based geometric problems.

2.1.3 The ruled canvas

Geometric problems are defined and displayed in the ruler canvas which is the largest display area placed in the central-right side of the screen.

A geometric problem is created by selecting specific constructions and constraints from the solver tool bar and placing the geometric elements by clicking on the canvas.

Once the geometric constraint problem is defined it must be saved in a file. It is suggested to save files of a geometric problem within a specific folder in the `testcases` folder.

2.1.4 The problem panel

The problem panel displays information concerning geometric components and constraints included in the geometric problem. Additional information on suggested constraints for under-constrained problems as well as sets of potential conflictive constraints is offered.

As the mouse cursor moves over the list of valid constraints, the geometric elements and constraints involved are displayed in yellow in the ruled panel.

Clicking on the `Edit` option of the top menu bar, the `Show element form` check box is displayed. When selected, a new panel listing information corresponding to an element in the geometric problem is listed. To select an element in the geometric problem place the mouse cursor on it. A list with the associated information will be displayed. Clicking on it will make the information to be displayed at the problem panel bottom.

2.2 The constructive geometric constraint solver

`dynBCN` is built on top of `solBCN`, a ruler-and-compass constructive geometric constraint solver. User interacts with `solBCN` through the menu bar options plus three specific panels: the tool bar, the ruled canvas and the problem information panel. All of them have been already described.

2.3 The dynamic simulator

The dynamic simulator shows how a geometric constraint-based construction changes as the variant parameter value describes a predefined path in the variant parameter domain. We shall describe it later on.

2.4 The reachability problem solver

The reachability simulator first solves the reachability problem stated as follows.

> Let I_s and I_e be two instances of a well defined geometric construction where I_s is called the starting instance and I_e the ending instance. Are there continuous transformations that, preserving the incidence relationships established in the geometric construction, brings I_s to I_e?

If the answer is positive, among the possibly different variant parameters paths that solve the problem, the system shows the dynamic behavior corresponding to a path with minimum length. We shall describe later on the reachability simulator.

3 Dynamic simulator

The dynamic simulator shows how a geometric constraint-based construction changes as the variant parameter value describes the path defined in the `example.trn` file.

To simulate the dynamic behavior of a dynamic problem, select in the `Dynamics` pop up menu the option `Dynamics Simulator`. Then select the specific problem in the dialog box and click on the `Open` button. There are two different situations depending on whether the path file `example.trn` has been defined or not.

3.1 The path file has been defined

Whenever a `example.trn` file has been created in the dynamic problem folder, the dynamic simulator displays a window with three different panels. Listed from top to bottom and from left to right, they are:

1. The simulation panel, where the geometry is displayed.
2. The domain panel, where the variant parameter domain and the current variant parameter value are displayed. The domain is shown as a set of domain intervals, each drawn as a rectangle labeled with the set of signs chosen (the index) for the construction steps with more than one solution. The interval where the variant parameter is currently taking values is filled in green. The current variant parameter value is shown as a vertical red line.
3. The buttons panel. This panel includes three buttons and a slider. Clicking on button `Go` triggers the simulation. Button `Set` resets the starting geometric instance. Button `Hold` holds the simulation. Clicking on button `Go` after holding the simulation, resumes it. Once the simulation is over, the user can rerun it by clicking on the `Set` button and then selecting `Go`. Moving the slider fixes the simulation velocity.

3.2 The path file has not been defined

When no `example.trn` file is present, `dynBCN` shows a window warning the user. After clicking on the `Accept` button, the simulator window described in Section 3.1is displayed. Selecting now the `Go` button, results in a new warning window which reminds the user the need of editing an `example.trn` file.

The file `example.trn` that stores the variant parameter path must be hand-edited by the user. Each line in the `example.trn` file includes two space separated text items, say X Y. The first item X takes values in the set {L, R} and describes whether the variant parameter path takes decreasing or increasing values respectively.

The second item Y takes values in the set of integers {1..n} where a given value identifies a specific interval in the variant parameter domain. For example L 5 describes a path along the domain interval number 5 according to decreasing values while R 3 describes a path along the domain interval number 3 according to increasing parameter values.

Notice that it is the user's responsibility to define paths such that at each point in the path, the construction of the solution instance is feasible. Otherwise, `dynBCN` fails.

When no `example.trn` is present, `dynBCN` displays the simulator window depicting the starting instance geometry, the variant parameter domain and the corresponding initial variant parameter value. Notice that the starting domain interval, filled in green, is fixed by `dynBCN`. Therefore, the path in the file must start in this domain interval.

Now, using his favorite file editor, the user can edit the `example.trn` file defining path steps by looking at the variant parameter domain. Once the path file has been defined, apply the procedure described in Section 3.1.

4 Reachability solver

The reachability solver decides on the reachability problem stated in Section 2.4 represented as a dynamic problem. If the answer is positive, among the possibly different variant parameter paths that solve the problem, the system shows the dynamic behavior corresponding to a path with minimum length.

4.1 Creating and editing a dynamic problem

A dynamic problem is always build on an already defined geometric constraint-based problem. The user can associate to a geometric constraint problem as many different dynamic problems as different geometric parameters can be selected as the variant parameter. In what follows, we assume that in the folder `testcases` there is a folder, say `example`, with a specific geometric problem defined by constraints, `example.dis`.

4.1.1 Creating a new dynamic problem

To create a new dynamic problem click on the `Dynamics` button in the top menu bar and select `New Dynamic Problem`. A dialog window offers the user to navigate and select the geometric constraint problem of interest, say `testcases/example/example.dis`. After clicking on the `Open` button, the geometric problem is displayed in the graphic canvas in the user interface and a display box is shown.

The displayed box provides the tools for the user to set the information required to define the specific dynamic problem. The box has two tabs, `Parameters` and `Extra Elements`.

The fields in the `Parameters` tab that must be filled are

- `File name`: the name of the dynamic problem, omitting the name extension, for example, `example`. However, the name of the dynamic problem do not need to be the same of the corresponding constraint problem. As a matter of fact, different dynamic problems derived from a given geometric constraint problem, must have different names.

- `Variant parameter name`: Name of the variant parameter, taken from the geometric constraint problem. For example, distance `d3`.

- `Variant parameter type:` Parameter type of the variant parameter: distance or angle. For the distance variant parameter `d3`, check the box `distance`.

- `First reference element`: The identifier of one geometric element used to define a frame of reference for displaying the dynamic geometry.

- `Second reference element`: The identifier of one geometric element. Along with the first reference element defines the frame of reference to display the dynamic geometry.

- `Simulation step`: Step applied to change the variant parameter value.

The `Extra Elements` tab defines additional geometric elements which are not part of the geometric constraint problem but that helps in visualizing the dynamic simulation. Three different types of additional geometric elements can be defined: lines, segments and circles. Additional lines and segments are defined by typing the name of two points in the geometric problem in the boxes labeled `Related to`.

Additional circles are defined by the center and radius. The center is fixed by typing in the first text field under the label `Related to` the name of a point in the geometric problem. The radius is fixed by typing in the second text field the identifier of a distance in the constraint geometric problem. Finally click on the button `Add`.

To remove an already defined additional element, the user must select the element type into the combo box, type in the identifiers of points as described above and select `Remove`.

The new dynamic geometry problem is stored as a set of files including `example.ini,` `example.dom` and `example.cnx`.

4.1.2 Editing a dynamic problem

When editing an already existing dynamic problem, the dialog box shows the current settings for both parameter and additional information. Edition is performed just by changing the current settings.

4.2 Running the reachability solver

The reachability solver is launched by clicking on the `Dynamics` button on the top bar menu and then selecting the `Dynamics Simulator` option.

The dynamic problem you are interested in is selected by navigating in the dialog box and clicking on the `Open` button.

Then a window with six panels is displayed. Listed from top to bottom and from left to right, they are:

1. The reachability simulation panel, where the geometric simulation is displayed.
2. The domain panel, where the variant parameter domain and the current variant parameter value are displayed. The domain is shown as a set of domain intervals, each drawn as a rectangle labeled with the set of signs chosen (the index) for the construction with more than one solution. The interval where the variant parameter is currently taking values is filled in green. The current variant parameter value is shown as a vertical red line.
3. The information panel, where information about the files involved is shown. The bottom line of text, usually shown in colored fonts, informs about the current status of the reachability problem.
4. The index panel, where the different index assignments to construction steps with more than one solution are listed.
5. The file names panel, where the different file names are listed.
6. The reachability parameters panel. Here text fields, buttons and a slider are provided to set the conditions under which the solution to the reachability problem must be shown.

    - Setting the starting and ending geometric problem instances is performed by assigning values to the corresponding variant parameter values and domain intervals in the starting/ending instance text fields.
    - `Set`: Sets as the current geometric problem instance the one defined in the starting/ending instance text fields. The geometric problem instance is displayed in the simulation panel and the domain panel is updated consequently.

      The system checks whether the selected final instance is reachable from the initial one. If the answer is positive, the message `Ready to run!` in green font is displayed. Otherwise an error message in red font is displayed.
    - `Go`: The simulation is triggered by clicking on the `Go` button. Whenever the geometric instance shown in the geometric problem panel does not correspond to the geometric instance defined as the starting one in the starting/ending

instance panel, first the geometric instance in the problem panel is set to the starting one then the simulation is started.

Once the simulation has reached the ending problem instance, clicking again on the `Go` button will show the solution to the same reachability problem.

- `Hold`: The simulation can be hold at any time by clicking on the `Hold` button. Clicking then on the `Go` button will resume the simulation at the geometric instance on hold.
- The slider allows the user to set the path step to trace the variant parameter path.

5 Some known problems

Here we describe some common problems arising when launching a simulation and their possible solutions.

- Since ruler-and-compass solvers are not complete, whenever the solver fails solving the geometric constraint problem, `dynBCN` consequently fails.
- The Simulator fails. Edit the `example.trn` file to avoid points in the variant parameter path where the solution instance is unfeasible.
- The Reachability Simulator shows no instance at the simulation panel: Edit the associated dynamic problem and check whether the reference elements of the problem are properly selected.
- The Reachability Simulator suddenly ends the simulation when the end of one domain interval is reached: Edit the associated dynamic problem and check whether the variant parameter type is properly set in the `Variant parameter type` check box.