

El manual del Rocs

Tomaz Canabrava
Andreas Cord-Landwehr
Traductor: Antoni Bella



El manual del Rocs

Índex

1	Introducció	6
1.1	Objectius, destinataris i flux de treball	6
1.2	El Rocs en poques paraules	7
1.2.1	Documents de graf	7
1.2.2	Tipus d'arestes	7
1.2.3	Tipus de nodes	7
1.2.4	Propietats	8
1.3	Guia d'aprenentatge	8
1.3.1	Generar el graf	8
1.3.2	Crear els tipus d'elements	8
1.3.3	L'algorisme	9
1.3.4	Executar l'algorisme	9
2	La interfície d'usuari del Rocs	10
2.1	Elements principals de la interfície d'usuari	10
3	Crear scripts	12
3.1	Executar algorismes en el Rocs	12
3.1.1	Controlar l'execució de l'script	12
3.1.2	Sortida de l'script	12
3.1.3	L'API de creació de scripts	13
4	Importar i exportar	14
4.1	Intercanviar projectes del Rocs	14
4.1.1	Importar i exportar documents de graf	14
4.1.1.1	Format de grafs trivial	14
4.1.1.1.1	Especificació del format	14
4.1.1.1.2	Exemple	15
4.1.1.2	Llenguatge DOT / Format de fitxer de grafs Graphviz	15
4.1.1.2.1	Característiques no implementades	15
4.1.1.2.2	Exemple	15

5 Disposició del graf	16
5.1 Traçar gràfics automàticament en el Rocs	16
5.1.1 Disposició basada en forces	16
5.1.1.1 Disposició d'arbre radial	17
6 Crèdits i llicència	18

Resum

El Rocs és una eina de teoria de grafs.

Capítol 1

Introducció

En aquest capítol s'ofereix una visió general de les principals característiques i el flux de treball típic. Les parts més importants són Secció 1.2 i capítol 3, que en el seu conjunt han de permetre que cada nou usuari comenci directament a emprar el Rocs.

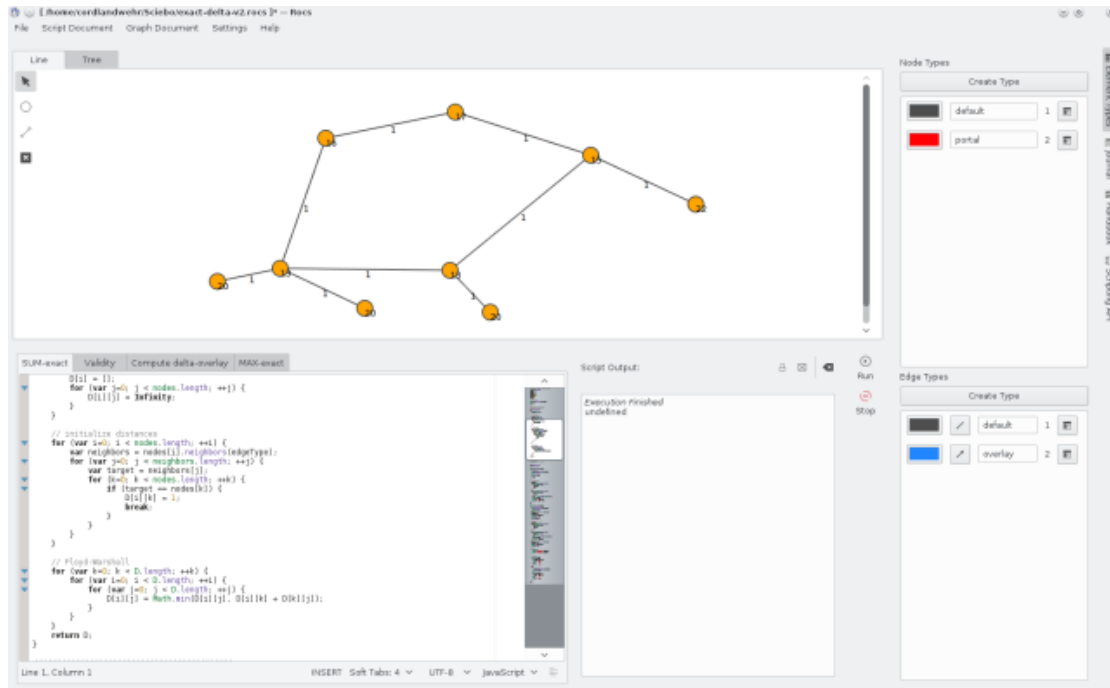
1.1 Objectius, destinataris i flux de treball

El Rocs és una eina per a la teoria de grafs per a tothom interessat en el disseny i anàlisi d'algorismes de graf. En particular, aquests són

- professors, que volen demostrar algorismes als seus estudiants,
- estudiants i investigadors, que volen entendre i veure com actua el seu algorisme,
- i tothom que estigui interessat en les estructures de dades i algorismes.

Per a tots ells, el Rocs proporciona un fàcil editor gràfic per a la creació de grafs, un poderós motor per a la creació de scripts per a executar algorismes, i diverses eines d'ajuda per a les simulacions, experiments i exportacions dels grafs. La forma típica d'usar el Rocs és crear un graf, ja sigui a mà (és a dir, arrossegant els nodes i arestes a la pissarra) o emprant un dels generadors de grafs. Els algorismes de grafs després es poden implementar i executar en el graf creat, i tots els canvis que realitzi l'algorisme, seran visibles immediatament en l'editor de grafs.

El manual del Rocs



1.2 El Rocs en poques paraules

Cada sessió del Rocs és un projecte: en obrir el Rocs es crea un projecte buit, en carregar algun projecte es convertirà en el projecte actual. En si, un projecte consisteix en *documents de graf*, *scripts/algorismes* i *scripts*.

1.2.1 Documents de graf

Un document de graf representa el contingut d'una pissarra en l'editor de grafs. Conté informació sobre els tipus de nodes i arestes definits per l'usuari, les seves propietats, i sobre els nodes i arestes ja creats. És a dir, el Rocs entén el conjunt de tots els nodes i arestes d'un document de graf per a formar un (no necessàriament connectat) graf. Tot el que pertany a un document de graf és accessible des del motor de scripts mitjançant l'objecte global **Document**.

1.2.2 Tipus d'arestes


En alguns escenaris, els grafs consisteixen en diferents tipus d'arestes (p. ex., un graf no dirigit, a més de les tres arestes calculades per un algoritme de cerca en amplada) que s'han de manejar i mostrar de forma diferent. Per això, a més d'un tipus d'aresta predeterminat, podeu definir altres tipus d'arestes arbitràries. Cada tipus d'aresta té la seva representació visual individual, propietats dinàmiques, i es pot canviar de no dirigit o dirigit. La interfície per a la creació de scripts proporciona mètodes convenients per a només accedir específicament als tipus específics d'arestes.

1.2.3 Tipus de nodes

De la mateixa manera que per als tipus d'arestes, podeu definir diferents tipus de nodes d'un graf (p. ex., per a donar un paper especial a alguns nodes). Cada tipus de node posseeix la seva pròpia representació visual i propietats dinàmiques.

1.2.4 Propietats

Cada element (node o aresta) pot tenir propietats. Aquestes propietats s'han de configurar en el corresponent tipus de node o aresta. Les propietats s'identifiquen i accedeix mitjançant els seus noms i poden contenir qualsevol valor. Per a crear una propietat nova o per a modificar una

d'existente, empreu la barra lateral **Tipus d'elements** i empreu el botó  **Propietats** per a obrir el diàleg.

També podeu emprar el motor de crear scripts per a accedir a les propietats registrades i canviar els seus valors. En el següent exemple se suposa que la propietat 'weight' està registrada per al tipus d'aresta predeterminat.

```
var nodes = Document.nodes()
for (var i = 0; i < nodes.length; ++i){
    nodes[i].weight = i;
}
for (var i = 0; i < nodes.length; ++i){
    Console.log("weight of node " + i + ": " + nodes[i].weight);
}
```

1.3 Guia d'aprenentatge

En aquesta secció volem crear un projecte d'exemple per a explorar algunes de les característiques més importants del Rocs. L'objectiu és crear un graf i un script que il·lustri un algorisme d'aproximació a 2 simple per al problema *cobertura de vèrtexs mínima*. En aquest problema s'han de trobar un subconjunt de nodes del graf C de mida mínima, de manera que cada aresta del graf estarà connectada a almenys un node de C . Se sap que aquest problema és NP-complex i volem il·lustrar com trobar una aproximació de factor 2 cercant una coincidència en el graf donat.


El nostre objectiu és visualitzar la relació de les coincidències i la cobertura de vèrtexs mínima. Per això, volem especificar dos tipus d'arestes, un per a mostrar les arestes coincidents i un per a mostrar les arestes 'comunes', així com dos tipus de nodes que utilitzarem per a distingir els nodes continguts en C i els que no estan continguts en C .

1.3.1 Generar el graf

Per a crear el graf, emprem un generador de grafs predeterminat proporcionat pel Rocs. El trobareu al menú principal a **Document de graf** → **Eines** → **Genera el graf**. Després, seleccionarem un **Graf aleatori** amb 30 nodes, 90 arestes i amb 1 llavor (la llavor és la llavor inicial per al generador de grafs aleatoris; si s'utilitza la mateixa llavor diverses vegades es tornaran a reproduir els mateixos grafs).

1.3.2 Crear els tipus d'elements

Emprem els **Tipus d'elements** per a crear un segon tipus de node i un segon tipus d'aresta.

Per a aquests dos nous tipus, obrirem el diàleg de propietats amb els respectius botons de  **Propietats** i establim els seus ID a 2. A més, canviarem els colors dels elements d'aquests dos tipus nous (per a distingir els tipus predeterminat). Finalment, establim que tots els tipus d'arestes són bidireccionals i establim els ID del tipus predeterminat a 1.

1.3.3 L'algorisme

Per a acabar, hem d'implementar l'algorisme d'aproximació. Per a això utilitzarem la següent implementació:

```

for (var i=0; i < Document.nodes.length; i++) {
    Document.nodes[i].type = 1;
}
for (var i=0; i < Document.edges.length; i++) {
    Document.edges[i].type = 1;
}

var E = Document.edges(); // conjunt d'arestes sense processar
var C = new Array();      // arestes coincidents
while (E.length
> 0) {
    var e = E[0];          // prenem la primera aresta e={u,v}
    var u = e.from();
    var v = e.to();
    e.type = 2;           // establim l'aresta com a coincident
    E.shift();            // eliminem «e» (és a dir, E[0]) des de la llista ←
                           d'arestes
    C.push(u);             // afegim «u» a «C»
    C.push(v);             // afegim «v» a «C»

    // marquem «u,v» com a nodes en «C»
    u.type = 2;
    v.type = 2;

    // eliminem de «E» totes les arestes adjacents a «u» o «v»
    var adjacent = u.edges();
    for (var i=0; i < adjacent.length; i++) {
        var index = E.indexOf(adjacent[i]); // troba l'índex
        if (index != -1) {
            E.splice(index, 1); // l'elimina si ja s'ha trobat
        }
    }
    var adjacent = v.edges();
    for (var i=0; i < adjacent.length; i++) {
        var index = E.indexOf(adjacent[i]); // troba l'índex
        if (index != -1) {
            E.splice(index, 1); // l'elimina si ja s'ha trobat
        }
    }
}
Console.log("Vertex Cover contains " + C.length + " nodes.");

```

1.3.4 Executar l'algorisme

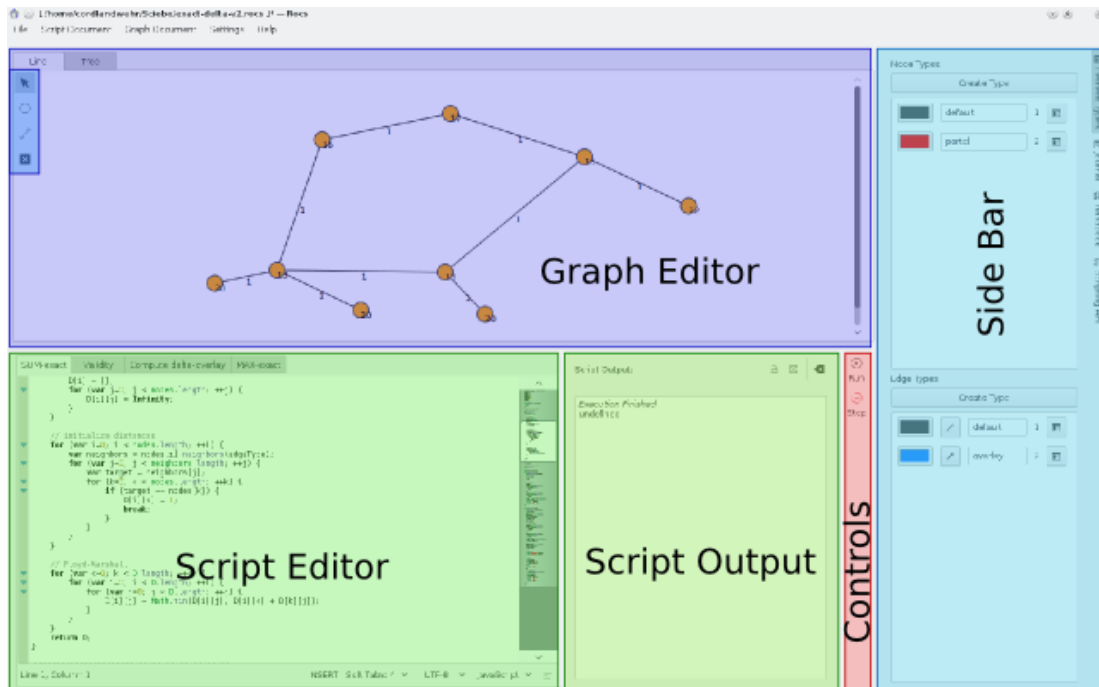
L'algorisme es pot executar amb el botó  Executa al plafó de control de l'script.

Capítol 2

La interfície d'usuari del Rocs

2.1 Elements principals de la interfície d'usuari

La interfície d'usuari es divideix en diverses parts lògiques que es presenten en la següent captura de pantalla.







Editor de grafs

L'editor proporciona una pissarra en blanc on es poden situar els nodes i les arestes. En fer doble clic a qualsevol dels seus elements obrirà un menú adequat. Podeu utilitzar les eines des de les *pestanyes en la Barra lateral* per a crear i modificar els grafs.

Eines disponibles:

- A dalt a l'esquerra del quadre, trobareu les icones d'acció següents. Fer clic sobre una acció significa que el punter del ratolí aplicarà aquesta acció a la pissarra de l'editor de grafs:

-  **Selecciona i mou:** Per a seleccionar elements, o bé feu clic sobre l'espai sense utilitzar a la pissarra, manteniu premut el ratolí i dibuixeu un rectangle que contingui alguns elements de dades i/o arestes per a seleccionar aquests elements o en cas contrari feu clic directament sobre un element sense seleccionar per a seleccionar-lo. Si feu clic sobre un element o conjunt d'elements seleccionats, respectivament, els podreu moure mantenint premut el ratolí i movent-lo. També és possible moure els elements seleccionats amb les tecles del cursor.
-  **Crea un node:** Feu clic sobre una posició arbitrària a la pissarra de l'editor de grafs per a crear un element de dades nou que pertanyi a l'estructura de dades actualment seleccionada. Si manteniu premut el punter del ratolí sobre el botó, es mostrarà un menú des d'on es podran seleccionar (només si hi ha diferents tipus de dades) els tipus de dades dels elements de dades creats.
-  **Crea una aresta:** Feu clic sobre un element de dades, manteniu premut el ratolí i dibuixeu una línia cap a un altre element de dades al qual apunti l'aresta. Aquesta acció només té èxit si el graf actual permet afegir aquesta aresta (p. ex., en un graf no dirigit no podreu afegir múltiples arestes entre dos elements de dades). Si manteniu premut el punter del ratolí sobre el botó, es mostrarà un menú des d'on es podran seleccionar (només si hi ha diferents tipus d'arestes) els tipus d'arestes dels nous elements de dades creats.
-  **Elimina l'element:** Feu clic sobre un element per a eliminar-lo. Si elimineu un node, totes les arestes adjacents també s'eliminaran.

Barra lateral

A la dreta, trobareu la barra lateral que proporciona diverses eines per al vostre flux de treball:

- **Tipus d'element:** Aquest giny permet accedir directament a les arestes i tipus de node disponibles.
- **Diari:** Cada projecte disposa del seu propi diari que es pot utilitzar per a, p. ex., anotar tasques, resultats o observacions.
- **API de creació de scripts:** Per a obtenir accés directe a la documentació dels scripts, podeu obrir aquest giny.



Editor de scripts

En aquest editor de text podeu escriure algorismes com s'explica en detall en capítol 3. Podeu treballar sobre diversos documents de script simultàniament utilitzant diverses pestanyes.

Sortida de l'script:

Aquesta àrea de text bé mostra informació de depuració o la sortida de l'script del vostre algorisme, depenent de l'opció indicada a la part superior d'aquest giny. Si l'script retorna un error, automàticament es presentarà la sortida de depuració.

Controls

Aquí trobareu els controls per a l'execució de scripts. Podeu executar l'script que està obert a l'editor de scripts prement  **Executa**. Mentre s'executa l'script, és possible aturar l'execució prement el botó  **Atura**.

Capítol 3



Crear scripts

3.1 Executar algorismes en el Rocs

El Rocs empra internament el motor JavaScript de QtScript. Això vol dir, que tots els algorismes que implementeu empraran JavaScript. A continuació, us expliquem com accedir i canviar els elements d'un document de graf des del motor per a la creació de scripts. És important tenir en compte que els canvis realitzats pel motor per a la creació de scripts es reflecteixen directament en les propietats dels elements a l'editor de grafs.

3.1.1 Controlar l'execució de l'script

Hi ha diferents modes d'execució per als vostres algorismes:

-  **Executa:** Executa l'script fins a finalitzar.
-  **Atura:** Atura l'execució de l'script (només disponible quan s'executa un script).

3.1.2 Sortida de l'script

Durant l'execució d'un algorisme, la sortida de depuració i del programa es mostren a la *depuració i sortida de l'script*. Si el motor de creació de scripts detecta un error de sintaxi, l'error també es mostrarà com a missatge de depuració. Cal tenir en compte que tots els missatges del programa també es mostraran a la sortida de depuració (com a text en negreta).

Podeu controlar el text que es mostrarà a la sortida de l'script mitjançant les següents funcions:

```
Console.log(string message);           // mostra el missatge com a sortida ↔  
de l'script  
Console.debug(string message);        // mostra el missatge com a sortida ↔  
de depuració  
Console.error(string message);         // mostra el missatge com a ↔  
sortida d'error
```

3.1.3 L'API de creació de scripts

Les diferents parts del Rocs proporcionen un element estàtic al qual es pot accedir des del motor per a la creació de scripts. Aquests són:

- **Document** per al document de grafs
- **Consola** per a la sortida de registre de la consola

Per a l'ús explícit de l'API i per a un mètode de referència, si us plau, vegeu l'ajuda en línia a la barra lateral del Rocs.

Capítol 4

Importar i exportar

4.1 Intercanviar projectes del Rocs

Els projectes del Rocs es poden importar i exportar com fitxers d'arxiu `.tar.gz`. Aquests arxius es poden utilitzar per a intercanviar projectes. Podeu importar i exportar amb els elements de menú **Document de graf** → **Importa un graf...** i **Document de graf** → **Exporta el graf com a...**, respectivament.

4.1.1 Importar i exportar documents de graf

El Rocs actualment admet la importació i exportació dels següents formats de fitxer:

- Fitxers DOT, també coneguts com a fitxers Graphviz
- Fitxers GML
- Fitxers en el format de grafs trivial
- Format del llenguatge de marques Keyhole

4.1.1.1 Format de grafs trivial

El *format de grafs trivial* (TGF) és un format de fitxer basat en text simple per a la descripció dels grafs. Un fitxer TGF consta d'una llista de definicions de node, que assignen els identificadors dels nodes a les etiquetes, seguits per una llista de les arestes. En aquest format, és possible tenir més d'una etiqueta per node i un valor per a cada aresta. El Rocs interpreta grafs importats com a grafs no dirigits. Els grafs exportats contindran dues arestes per connexió si les connexions són bidireccionals.

4.1.1.1.1 Especificació del format

- El fitxer comença amb una llista de nodes (un node per línia), seguit per una línia amb el caràcter únic '#', seguit per una llista d'arestes (una aresta per línia).
- Un node consisteix d'un nombre enter (identificador), seguit d'un espai, seguit per una cadena arbitrària.
- Un aresta es compon de dos nombres enters (identificadors) separats per un espai, seguit per un espai, seguit per una cadena arbitrària. Se suposa que l'aresta dirigida apunta des del primer identificador fins al segon.

4.1.1.1.2 Exemple

```
1 node inicial
2 transmissor
3 pila
#
1 2 blau
2 1 vermell
2 3 verd
```

4.1.1.2 Llenguatge DOT / Format de fitxer de grafs Graphviz

El llenguatge DOT és un llenguatge de descripció de grafs de text net, que permet, tant una representació de grafs llegible, així com un processament eficient per als programes de disseny de grafs. DOT és el format de fitxer predeterminat per a la suite de visualització de grafs Graphviz, però també és àmpliament utilitzat per altres eines de graf. Les extensions de fitxer habituals per a DOT són `.gv` i `.dot`.

4.1.1.2.1 Característiques no implementades

El Rocs pot analitzar cada fitxer de grafs que contingui un graf especificat d'acord amb l'especificació del llenguatge DOT¹. La implementació de les característiques del llenguatge és completa, tot i les excepcions següents:

- subgraf: A causa de la manca d'un concepte de subgraf en el Rocs, els subgrafs només s'importen com un conjunt d'elements de dades i connexions. En especial, no s'importaran les connexions cap o des dels subgrafs.
- Els atributs HTML i XML: Els atributs (com les etiquetes) que contenen la sintaxi HTML o XML es llegeixen sense canvis. En especial, no es llegeix cap atribut com el tipus de lletra i estil.

4.1.1.2.2 Exemple

```
digraph myGraph {
  a -> b -> c;
  b -> d;
}
```

¹<https://graphviz.org/doc/info/lang.html>

Capítol 5

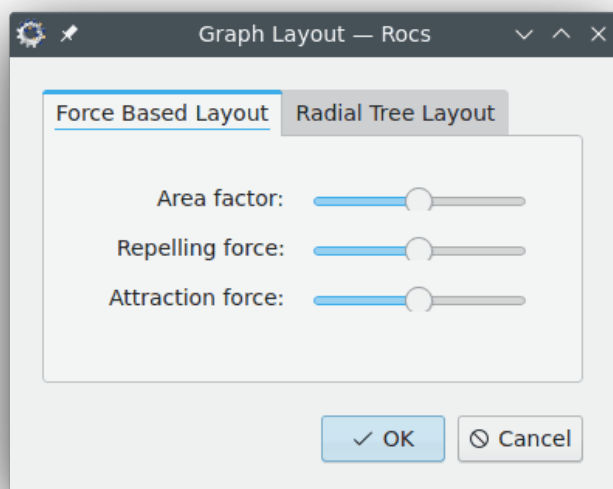
Disposició del graf

5.1 Traçar gràfics automàticament en el Rocs

El Rocs pot traçar gràfics automàticament. L'eina de disposició de grafs del Rocs es troba en el menú principal a **Document de graf** → **Eines** → **Disposició del graf**. Hi ha dos algoritmes de disposició diferents que es poden aplicar: disposició Basada en la força i disposició d'Arbre radial. Per a aplicar-ne una, seleccioneu la pestanya corresponent de l'eina de disposició de grafs, trieu els paràmetres desitjats i executeu l'algoritme prement el botó **D'acord**. Els detalls que són específics de cadascun dels algoritmes de disposició es proporcionen en les seccions següents.

5.1.1 Disposició basada en forces

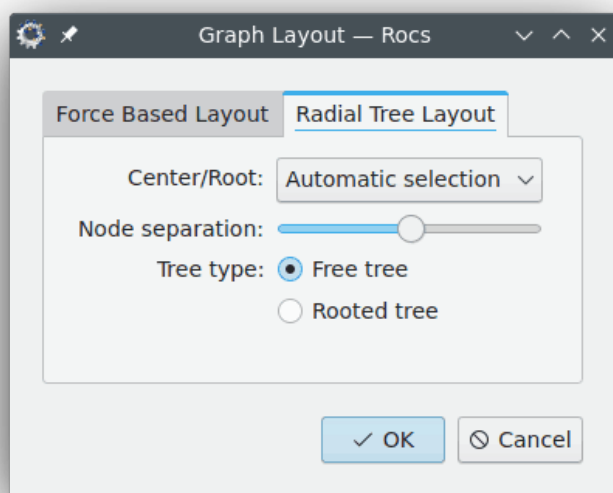
La Disposició basada en la força es pot aplicar a qualsevol graf. De manera intuïtiva, aquest algoritme simula les forces que actuen en cada node. Hi ha forces repel·lents entre parells de nodes i forces d'atracció entre parells de nodes veïns. La magnitud d'aquestes forces es pot especificar movent els controls lliscants corresponents a la interfície d'usuari.



Un altre paràmetre que es pot controlar és el Factor d'àrea. Aquest paràmetre controla com es distribueixen els nodes. Les disposicions generades amb valors alts de Factor d'àrea tendeixen a tenir grans distàncies entre els nodes.

5.1.1.1 Disposició d'arbre radial

La Disposició d'arbre radial només es pot aplicar als arbres. Qualsevol intent d'aplicar aquest algoritme de disposició a altres tipus de grafs produirà un missatge d'error. Els paràmetres per a la Disposició d'arbre radial es poden seleccionar utilitzant la interfície d'usuari proporcionada.



El paràmetre de tipus d'arbre selecciona entre una disposició d'arbre lliure i una disposició d'arbre amb arrel. En una disposició d'arbre lliure, els nodes es col·loquen lliurement sense cap jerarquia aparent entre ells. En una disposició d'arbre amb arrel, el node arrel es col·loca a la part superior i els subarbres es col·loquen a sota, el qual dona una idea d'una jerarquia entre els nodes.

El paràmetre centre/arrel defineix quin node s'utilitzarà com a arrel per a la disposició de l'arbre amb arrel o com a centre per a la disposició de l'arbre lliure. El centre d'una disposició d'arbre lliure és el primer node que col·loca l'algoritme. Tots els altres nodes es col·locaran en cercles centrats en el node central. L'algoritme de la disposició pot seleccionar automàticament un centre/arrel.

El paràmetre de separació dels nodes controla la distància entre els nodes. En fer augmentar el valor d'aquest paràmetre, augmentarà la distància entre els nodes. De manera similar, fer disminuir el valor d'aquest paràmetre farà que la distància entre els nodes disminueixi.

Capítol 6

Crèdits i llicència

Rocs

Copyright del programa:

- Copyright 2008 Ugo Sangiori (ugorox AT gmail.com)
- Copyright 2008-2012 Tomaz Canabrava (tcanabrava AT kde.org)
- Copyright 2008-2012 Wagner Reck (wagner.reck AT gmail.com)
- Copyright 2011-2015 Andreas Cord-Landwehr (cordlandwehr AT kde.org)

Copyright de la documentació:

- Copyright de la documentació 2009 Anne-Marie Mahfouf annma@kde.org
- Copyright de la documentació 2009 Tomaz Canabrava (tcanabrava AT kde.org)
- Copyright de la documentació 2011-2015 Andreas Cord-Landwehr (cordlandwehr AT kde.org)

Traductor de la documentació: Antoni Bella antonibella5@yahoo.com

Aquesta documentació està llicenciada d'acord amb les clàusules de la [Llicència de Documentació Lliure de GNU](#).

Aquest programa està llicenciat d'acord amb les clàusules de la [Llicència Pública General de GNU](#).