NEO: Elegant Optimization

Objektorienterad Design

MatLi
Linda Rydström (linry786)
Li Sandsveden (lisa459)
Mari Ahlqvist (marah803)
Jonas Granholm (jongr157)
Emil Karlsson (emika583)
David Larsson (davla210)

15 november 2012

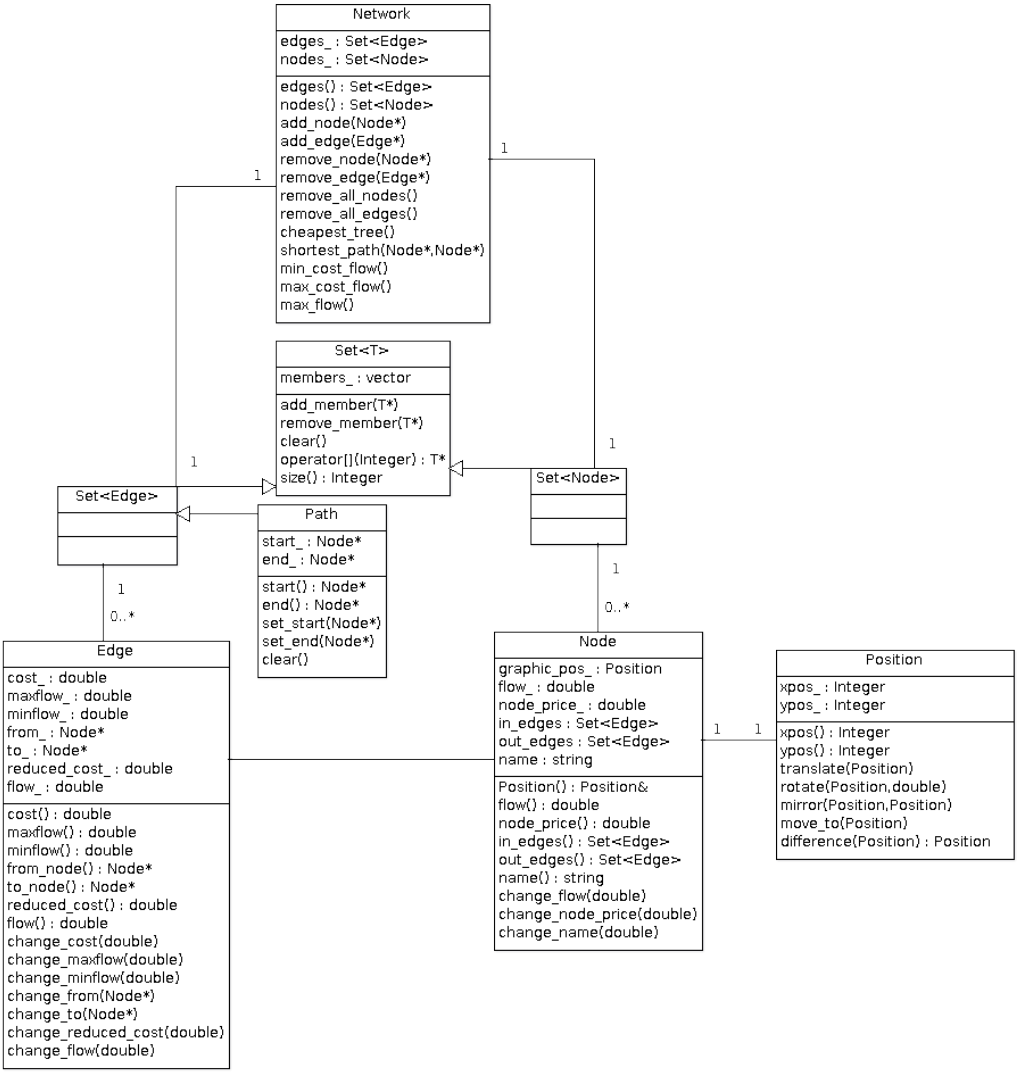
Innehåll

1 Sammanfattning

I det här dokumentet finns genomgående information om ingående klasser i projektet. Projektet, NEO: Elegeant Optimization, är en del av kursen TDDC76, Programmering och datastrukturer, som ges på Institutionen för Datavetenskap vid Linköpings Universitet.

2 Klassdiagram

I figur 1 nedan följer ett klassdiagram som ska ge en övergripande blick över systemets klasser och hur de är relaterade till varandra.



Figur 1: Klassdiagram

3 Klassbeskrivningar

Nedan beskrivs klasserna med en kort kommentar om vad klassen innebär och sedan listas datamedlemmar och medlemsfunktioner för klassen.

Både för datamedlemmarna och funktionerna gäller att den vänstra kolumnen innehåller namnet och den högra en förklaring.

3.1 Edge

Representerar en kant (båge) i ett nätverk.

Datamedlemmar	
double cost_	Bågkostnad
double maxflow_	Maximalt flöde
double minflow_	Minimalt flöde
Node* from_	Kantens startnod
Node* to_	Kantens slutnod
double reduced_cost_	Reducerad kostnad - ”Bara i lösta problem”
double flow_	Flöde - ”Bara i lösta problem”

Medlemsfunktioner	
double cost() const	Returnerar bågkostnaden
double maxflow()	Returnerar det maximala flödet
double minflow()	Returnerar det minimala flödet
Node* from_node() const	Returnerar startnoden
Node* to_node() const	Returnerar slutnoden
double reduced_cost() const	Returnerar den reducerade kostnaden
double flow() const	Returnerar flödet
void change_cost(double new_cost)	Sätter bågkostnaden till new_cost
void change_maxflow(double new_maxflow)	Sätter det maximala flödet till new_maxflow
void change_minflow(double new_minflow)	Sätter det minimala flödet till new_minflow
void change_from(node* new_from)	Sätter startnoden till new_from
void change_to(node* new_to)	Sätter slutnoden till new_to
void change_reduced_cost(double)	Sätter reducerad kostnad
void change_flow(double,double)	Sätter flödet

3.2 Node

Representerar noder i ett nätverk.

Datamedlemmar	
Position graphic_pos_	Nodens position
double flow_	Flödet
double node_price_	Nodpriset
Set<Edge> in_edges_	Ingående bågar
Set<Edge> out_edges_	Utgående bågar
string name	Nodens namn

Medlemsfunktioner	
Position& Position()	Returnerar nodens position
double flow()	Returnerar flödet
double node_price()	Returnerar nodpriset
void change_flow(double)	Ändrar flödet
void change_node_price(double)	Ändrar nodpriset
void change_name(string)	Ändrar namnet
Set<Edge> in_edges()	Returnerar ingående bågar
Set<Edge> out_edges()	Returnerar utgående bågar

3.3 Position

Representerar en position i den grafiska representationen av nätverket.

Datamedlemmar	
int xpos_	Den horisontella koordinaten för positionen
int ypos_	Den vertikala koordinaten för positionen

Medlemsfunktioner	
int xpos()	Returnerar den horisontella koordinaten för positionen
int ypos()	Returnerar den vertikala koordinaten för positionen
void translate(Position vect)	Förflyttar positionen med vektorn vect
void rotate(Position center, double angle)	Roterar positionen med vinkeln angle (i grader) moturs runt mittpunkten center
void mirror(Position point, Position vect)	Speglar positionen i linjen genom point som är parallell mot vektorn vect
void move_to(Position new_position)	Flyttar positionen till new_position
Position difference(Position pos)	Returnerar vektorn från positionen till pos

3.4 Set - template

En template för en mängd av en typ T.

Datamedlemmar	
std::vector<T*> members_	Medlemmar

Medlemsfunktioner	
void add_member(T*)	Lägger till ett objekt T
void remove_member(T*)	Tar bort ett objekt T
virtual void clear()	Tömmer mängden
T* operator[](unsigned int) const	Åtkomst av element på position
unsigned int size() const	Mängdens kardinalitet

3.5 Path

Representerar en väg mellan två noder i ett nätverk. Ärver från Set<Edge>.

Datamedlemmar	
Node* start_	Startnoden
Node* end_	Slutnoden

Medlemsfunktioner	
Node* start_node()	Returnerar startnoden
Node* end_node()	Returnerar slutnoden
void set_start_node(Node*)	Sätter startnoden
void set_end_node(Node*)	Sätter slutnoden
virtual void clear() override final	Tömmer mängden

3.6 Network

Representerar ett helt nätverk av noder och bågar. Lösningarna till de olika problemen kommer vara ett Network.

Datamedlemmar	
Set<Edge>	edges_
Set<Node>	nodes_

Medlemsfunktioner	
Set<Edge> edge_set()	Returnerar alla bågar
Set<Node> node_set()	Returnerar alla noder
void add_node(Node*)	Lägger till en nod
void add_edge(Edge*)	Lägger till en båge
void remove_node(Node*)	Tar bort en nod
void remove_edge(Edge*)	Tar bort en båge
void remove_all_nodes	Tar bort alla noder
void remove_all_edges	Tar bort alla bågar
void cheapest_tree()	Genererar en billigaste-träd-lösning
void shortest_path(Node*, Node*)	Genererar en kortaste-väg-lösning
void min_cost_flow()	Genererar minkostnadsflöde
void max_cost_flow()	Genererar maxkostnadsflöde
void max_flow()	Genererar maxflöde

4 Datalagring

Ett nätverk, det vill säga ett network-objekt, ska kunna sparas i en svg-fil med den grafiska representationen av nätverket. Filen ska kunna visas som en svg-bild av nätverket som det ser ut i programmet i en vanlig svg-läsare, och datan sparas i kommentarer i filen. I svg-filen, som i sig är en xml-standard, lagras extra xml-formaterad information för att kunna återskapa nätverket. Användaren kan välja att inte ha med denna information genom att exportera.

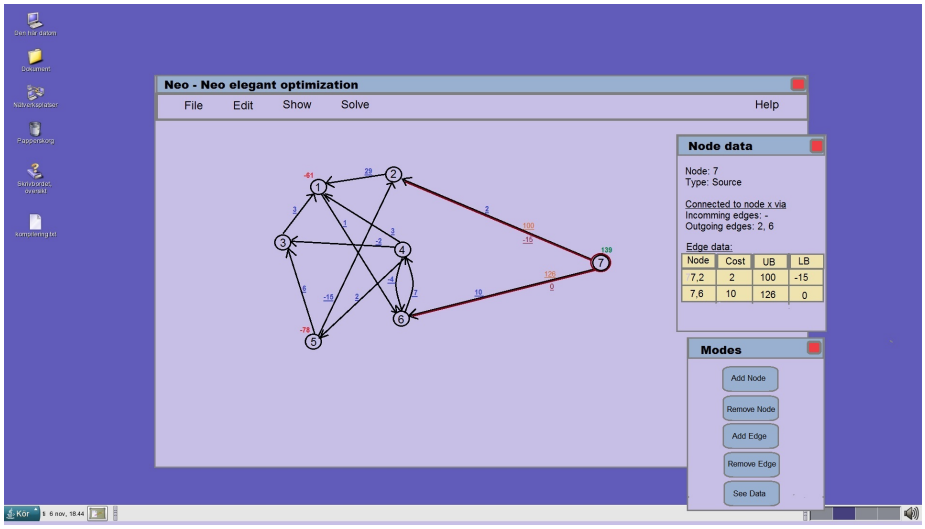
För att konstruera och läsa xml kommer vi använda oss av Xerces.

5 Beskrivning av det grafiska gränssnittet

Vi vill inte att användaren ska behöva läsa på en massa för att använda programmet utan att det ska ske intuitivt (förutsatt att man har grundkunskaper i optimering).

Vi vill också att användaren själv ska kunna förändra hur gränssnittet ser ut. Till exempel så ska man kunna ändra vad som visas i menyerna ”Modes” och ”Node Data”. Dessa kommer, när man öppnar programmet, att sitta fast som paneler till höger som man, om man vill, kan ”dra loss” så att de blir lösa menyer. Alla visuella inställningar finns under ”Show” i menyraden högst upp. Man bör kunna spara sina visuella inställningar.

I figur 2 nedan följer ett utkast till hur gränsnittet ska se ut.



Figur 2: Visuellt utkast