Public Transportation Modelling: Planning Bus Lines Routes

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL,BRAZIL

AUTOMATED PLANNING

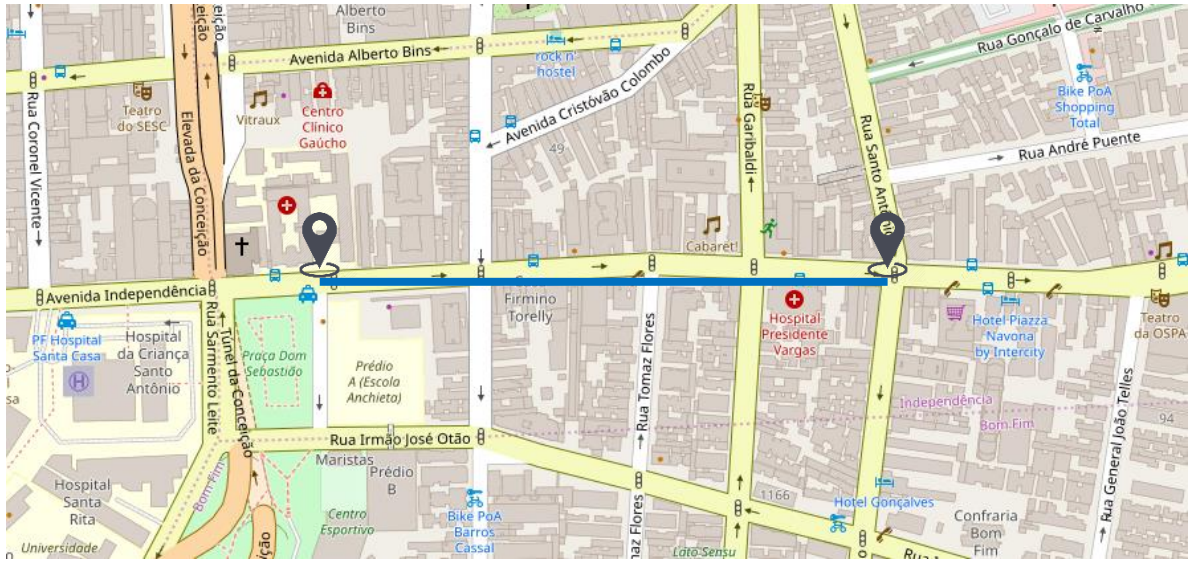
PROFESSOR: FELIPE MENEGUZZI

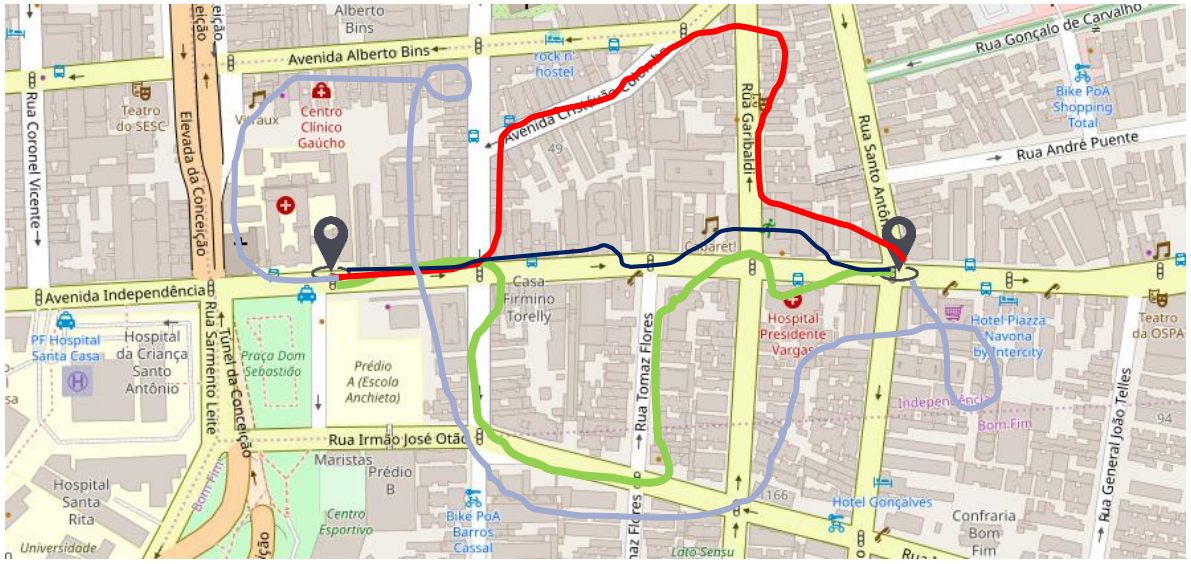
STUDENT: GABRIEL ROSSI FIGLARZ

PORTO ALEGRE, NOVEMBER 2019

Diverse Planning – Top *k*

While cost-optimal planning aims at finding one best quality plan, top-k planning deals with finding a set of solutions. (Katz, Sohrabi and Udrea, 2018)





ForbidIterative Planner

Generated Plans

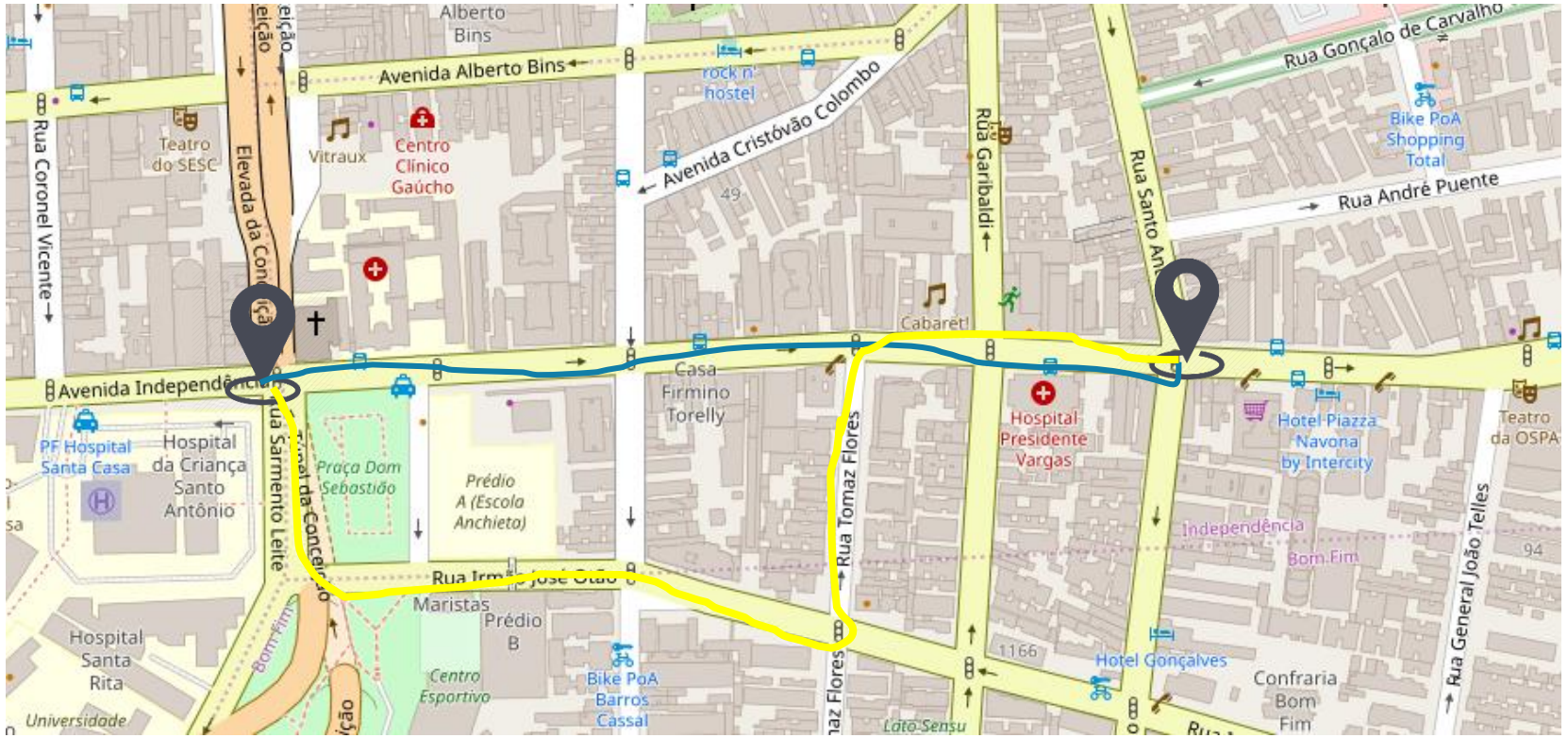
{
[Plan 1:
Independência;Independência;Independência]
}

ForbidIterative Planner

Generated Plans

{
[Plan 1:
Independência;Independência;Independência;Independência],
[Plan 2:
Conceição;IrmãoJosé;TomazFlores,Independência]

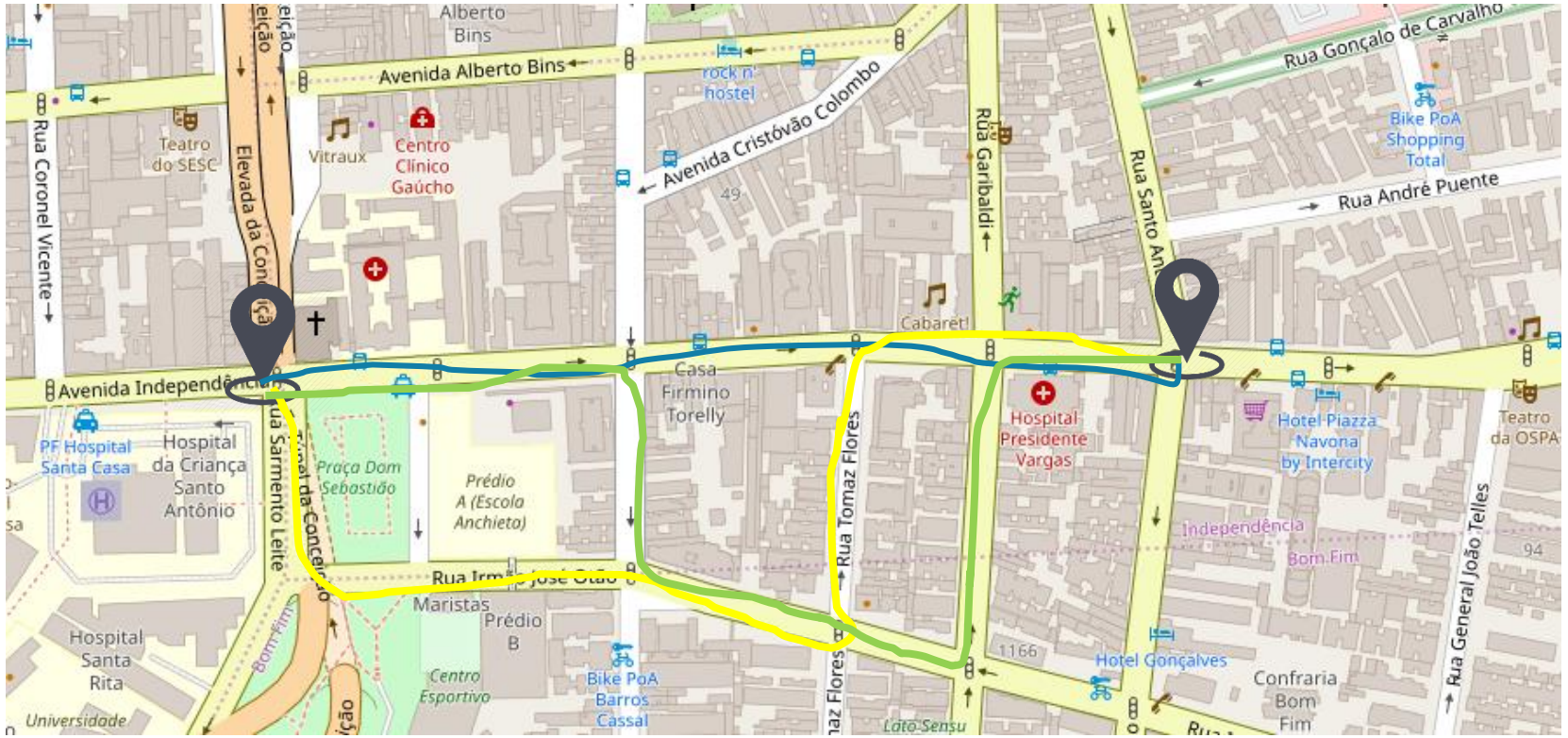
}



ForbidIterative Planner

Generated Plans

{
[Plan 1:
Independência;Independência;Independência],
[Plan 2:
Conceição;IrmãoJosé;TomazFlores;Independência]
[Plan3:
Conceição;Paralela;FelipeCamarão;Independência]
}



The Top-*k* Planning Problem < Π, k>

$$\Pi = (V, O, s0, s^*, cost)$$

k = nunmber of plans

Planning Task

State variables

Actions

Initial State

Goal State

The objective is to find the k plans of lowest cost for a planning taks

The Top-*k* Planning Problem < Π, k>

1 – Find an optimal plan π for a planning task Π

2 - Reformulate Π to a planning task Π’ with the same set of plans but excluding π

3 - Repeat (1) with Π=Π’ and π = π’ unless either k solutions have been found or the Π’ is provably unsolvable

Algorithm 1: IterativeTopK($\langle \Pi, k \rangle$)

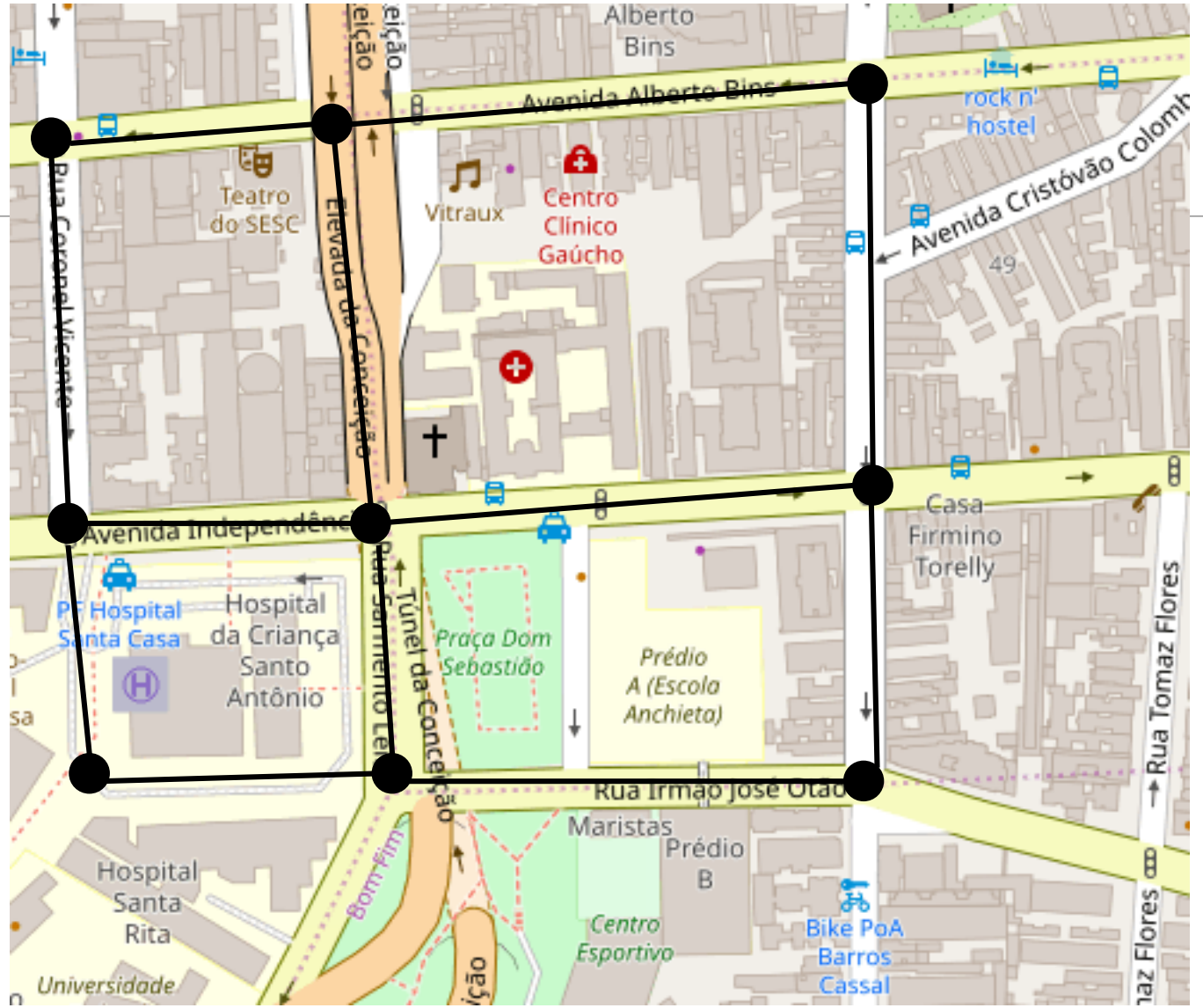
$P \leftarrow \emptyset, \quad \pi \leftarrow \emptyset, \quad r \leftarrow id$
while $|P| < k$ *and no failure occurs* **do**
  $\Pi, r' \leftarrow$ PLANFORBIDREFORMULATION(Π, π)
  $r \leftarrow r \circ r'$
  $\pi \leftarrow$ GETOPTIMALPLAN(Π)
  if GETOPTIMALPLAN(Π) *reports* UNSOLVABLE **then**
    return P
  end
  $P \leftarrow P \cup \{r(\pi)\}$
end
return P

LM – Cut Heuristics

A* search

Study Case

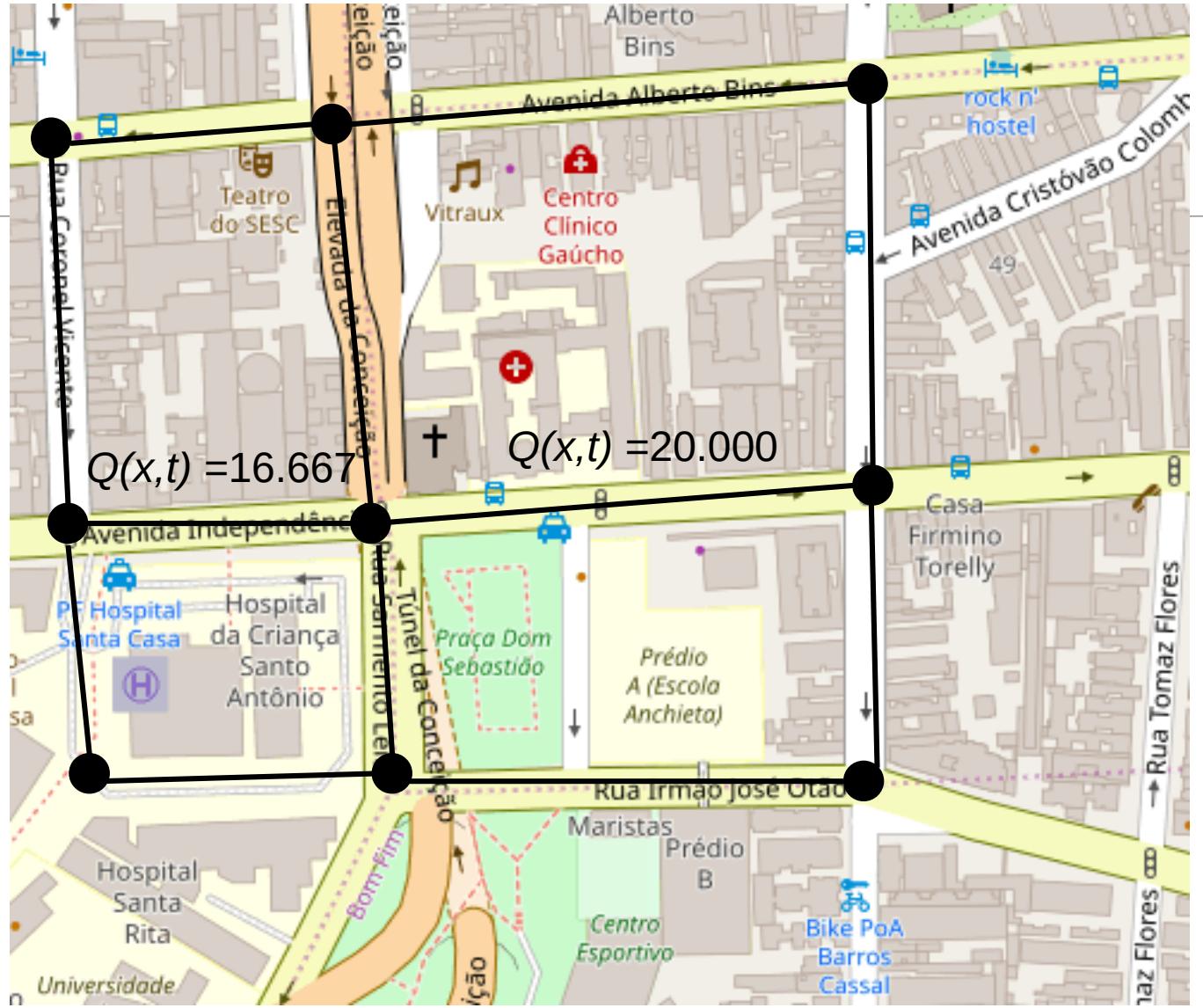

Alberto Bins
Avenida Alberto Bins
rock n' hostel
Avenida Cristóvão Colombo
Rua Gonçalo de Carvalho
Bike PoA Shopping Total
Rua André Puente
Rua Coronel Vicente
Teatro do SESC
Elevada da Conceição
Vitraux
Centro Clínico Gaúcho
Rua Garibaldi
Rua Santo Antônio
49
Cabaret
Avenida Independência
Casa Firmino Torelly
Hospital Presidente Vargas
Hotel Piazza Navona by Intercity
Teatro da OSPA
PF Hospital Santa Casa
Hospital da Criança Santo Antônio
Rua Sarmento Leite
Túnel da Conceição
Praça Dom Sebastião
Prédio A (Escola Anchieta)
Rua Tomaz Flores
Independência
Bom Fim
94
Rua Irmão José Otão
Maristas
Prédio B
Hospital Santa Rita
Bom Fim
Centro Esportivo
Bike PoA Barros Cassal
1166
Hotel Gonçalves
Confraria Bom Fim
Rua General João Telles
Universidade
Lato Sensu


Alberto
Bins
Avenida Alberto Bins
rock n'
hostel
Avenida Cristóvão Colomb
Rua Coronel Vicente
Teatro
do SESC
Elevada da Conceição
Vitraux
Centro
Clínico
Gaúcho
49
Avenida Independência
Casa
Firmino
Torelly
Rua Tomaz Flores
PF Hospital
Santa Casa
Hospital
da Criança
Santo
Antônio
Rua Sarmento Le
Túnel da Conceição
Praça Dom
Sebastião
Prédio
A (Escola
Anchieta)
Rua Irmão José Otão
Maristas
Prédio
B
Hospital
Santa
Rita
Bom Fim
Centro
Esportivo
Bike PoA
Barros
Cassal
Universidade

EPTC Data:

$Q(x,t)$ =16.667 vehicles/hour

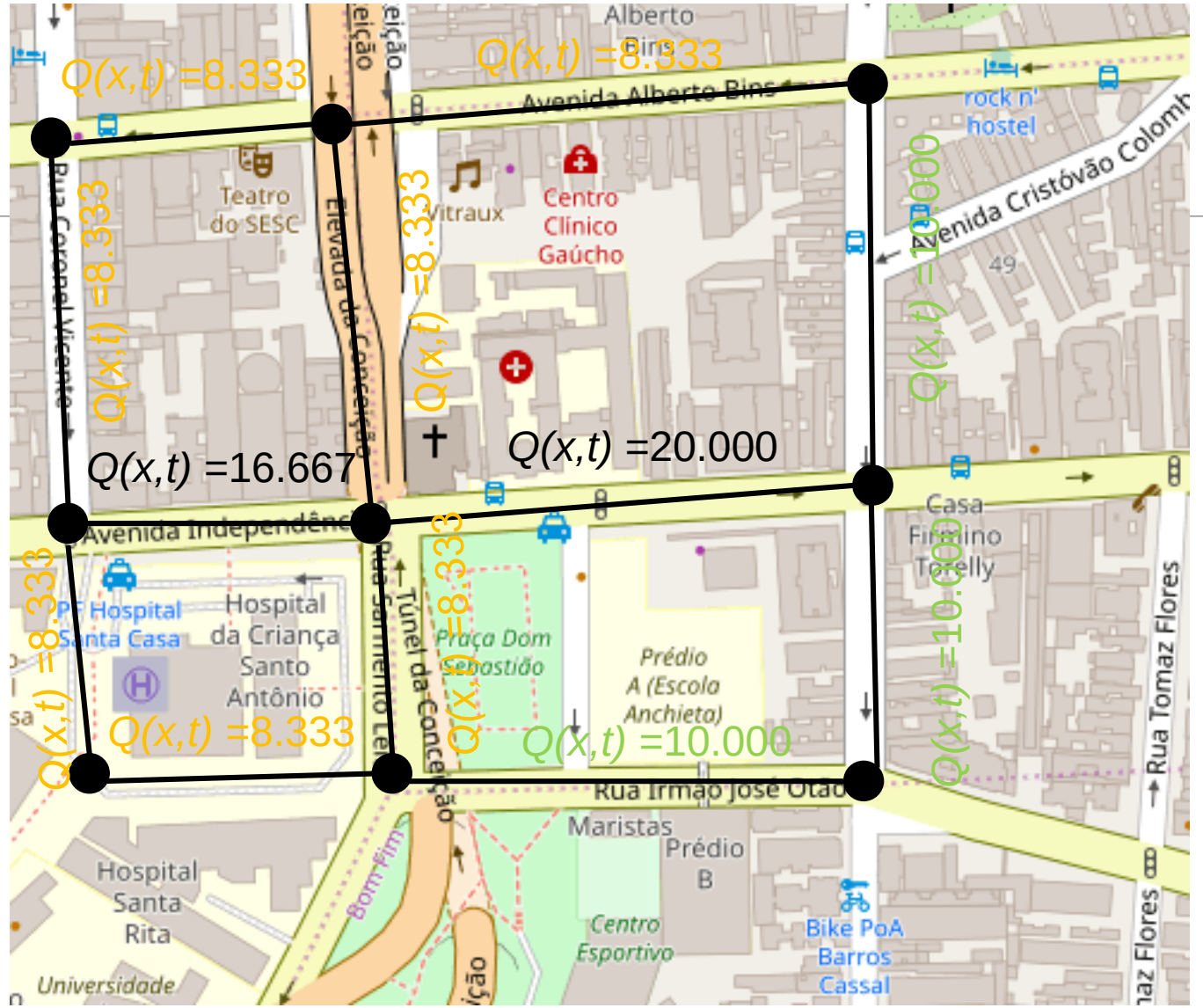
$Q(x,t)$ =20.000 vehicles/hour



EPTC Data:

$Q(x,t)$ =16.667 vehicles/hour

$Q(x,t)$ =20.000 vehicles/hour

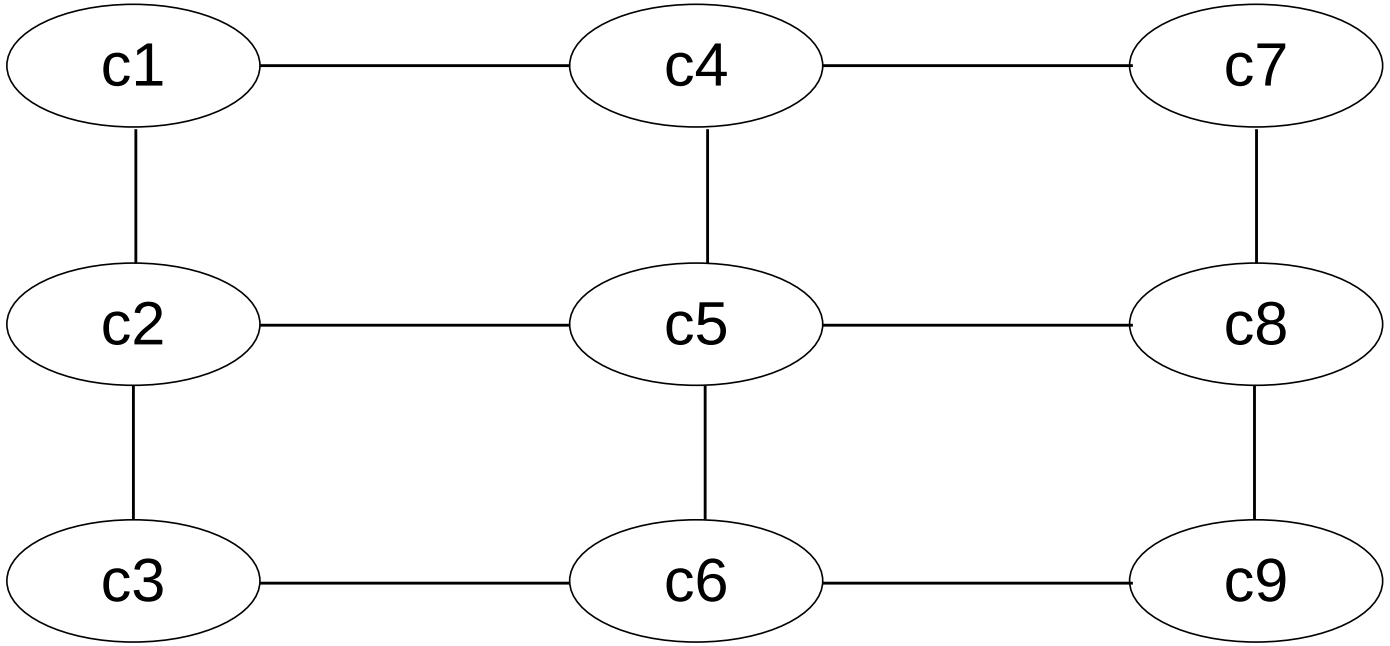


Estimated Data:

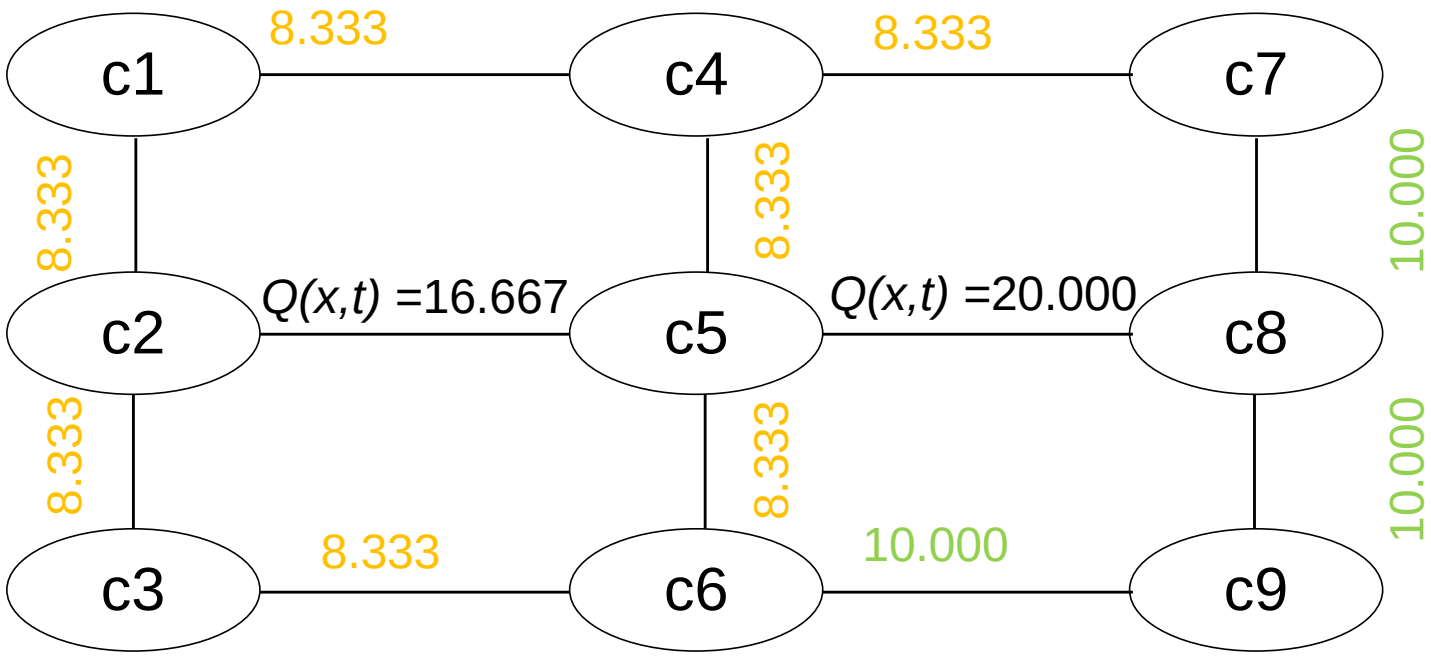
$Q(x,t)$ =8.333 vehicles/hour

$Q(x,t)$ =10.000 vehicles/hour

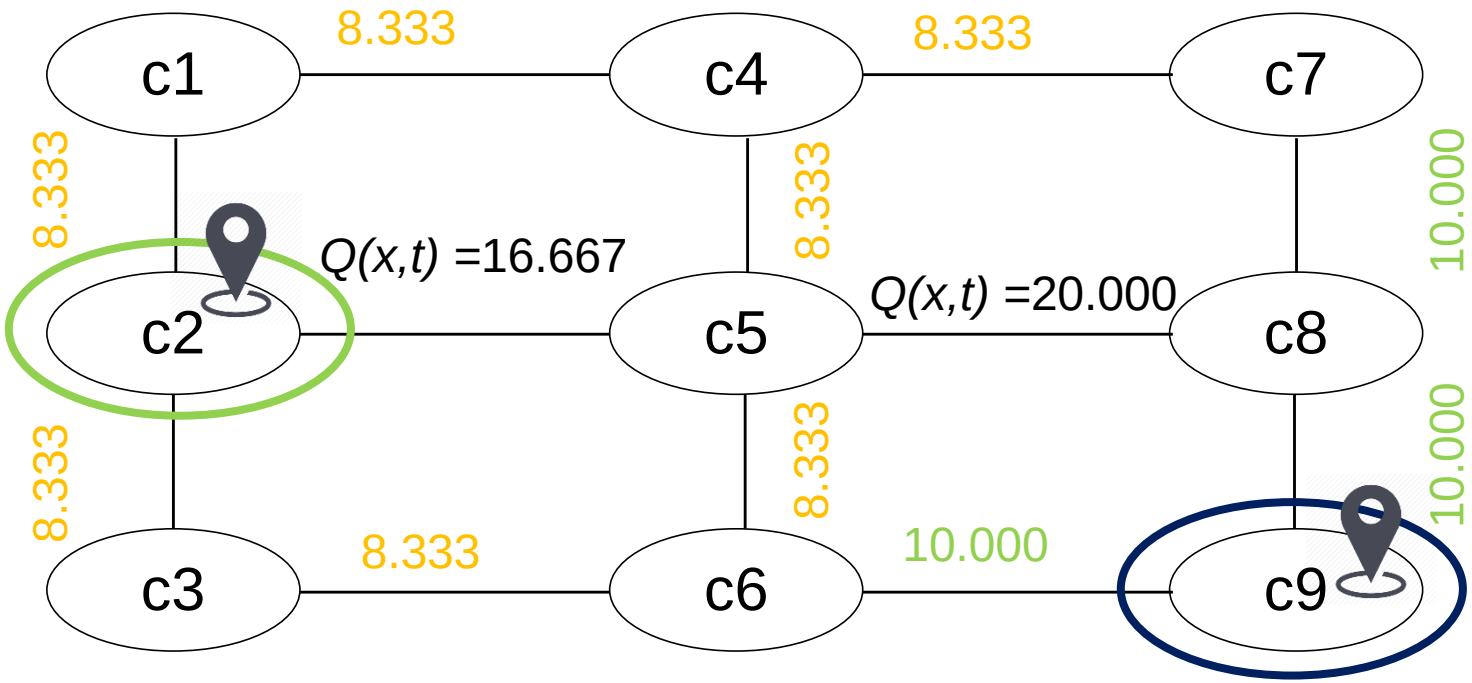
Simplified Problem



Simplified Problem

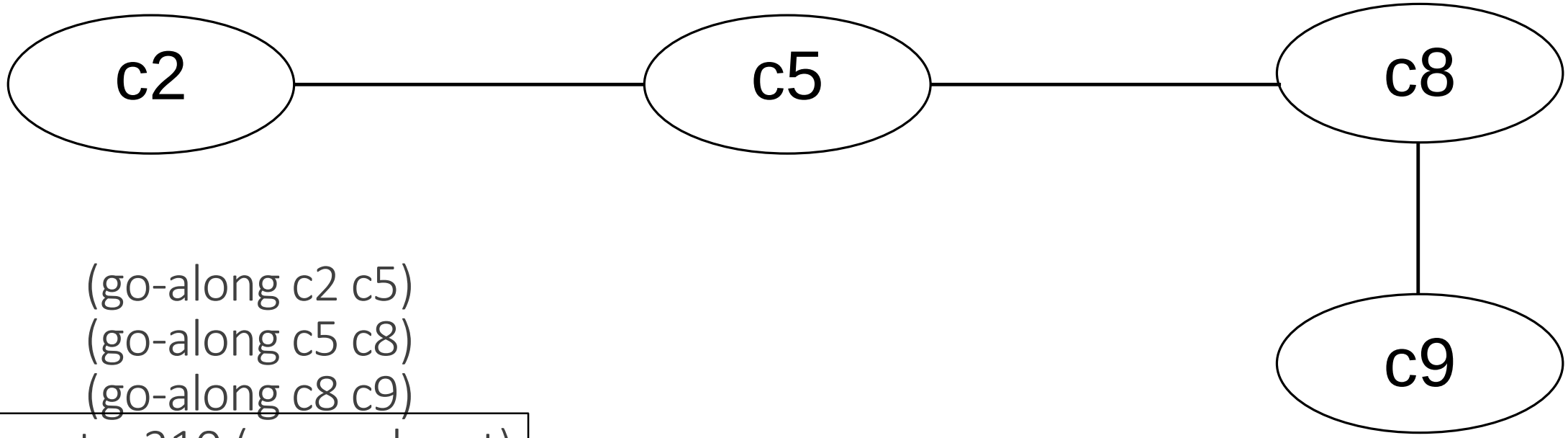


Simplified Problem



Generated Plans

Plan 1

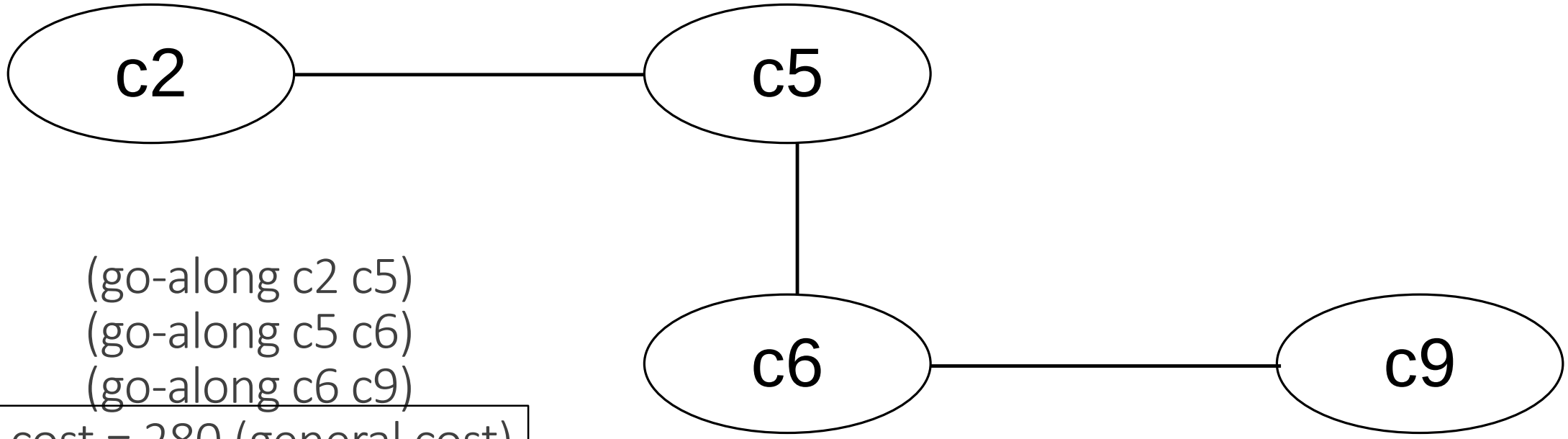


(go-along c2 c5)
(go-along c5 c8)
(go-along c8 c9)

; cost = 210 (general cost)

Generated Plans

Plan 2

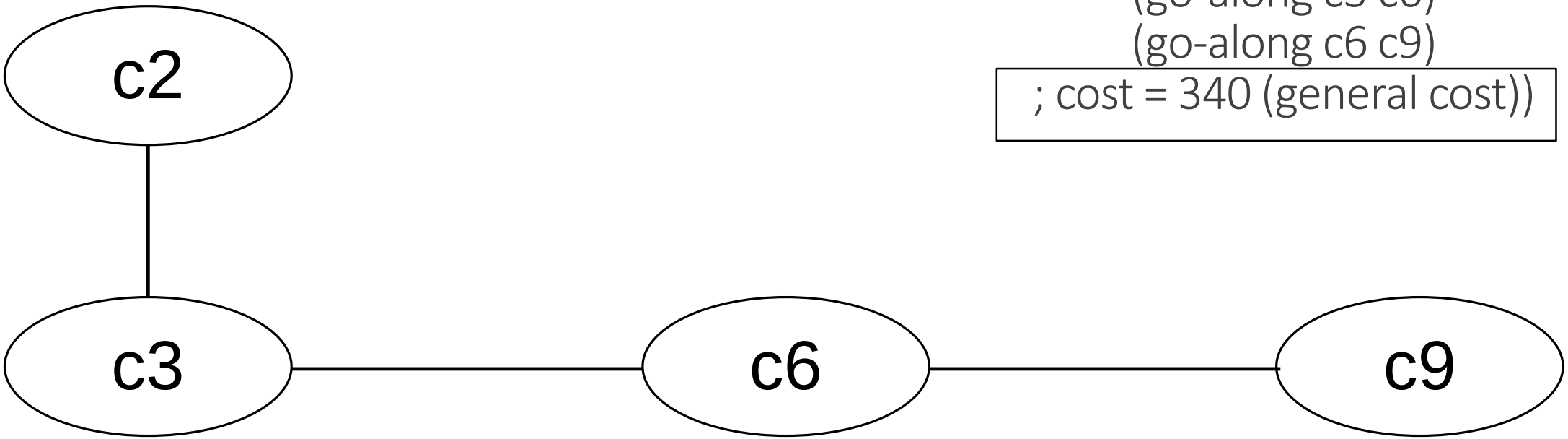


(go-along c2 c5)
(go-along c5 c6)
(go-along c6 c9)

; cost = 280 (general cost)

Generated Plans

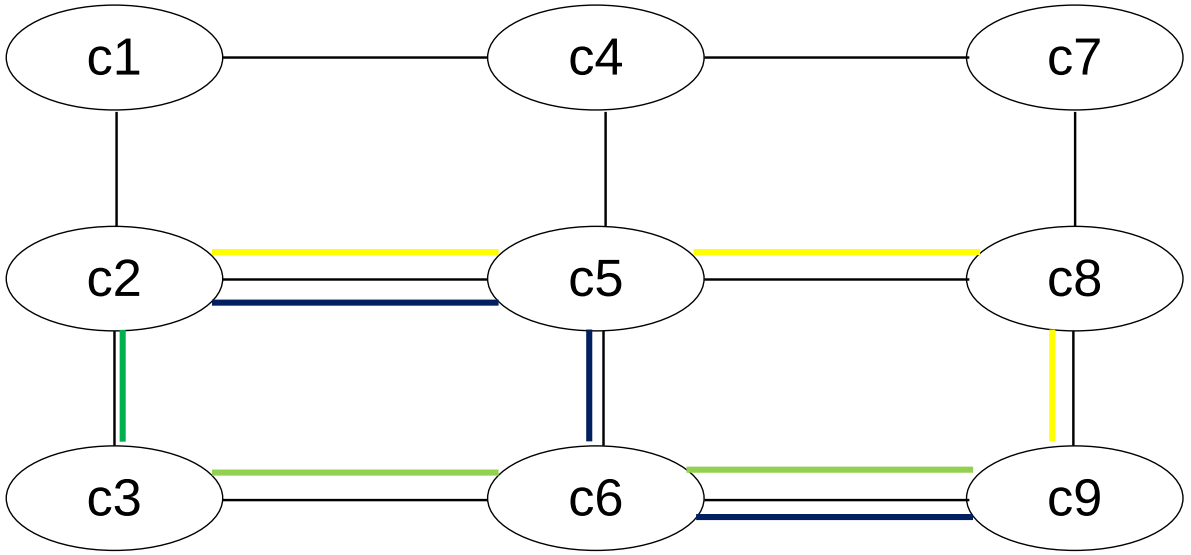
Plan 3



(go-along c2 c3)
(go-along c3 c6)
(go-along c6 c9)
; cost = 340 (general cost))

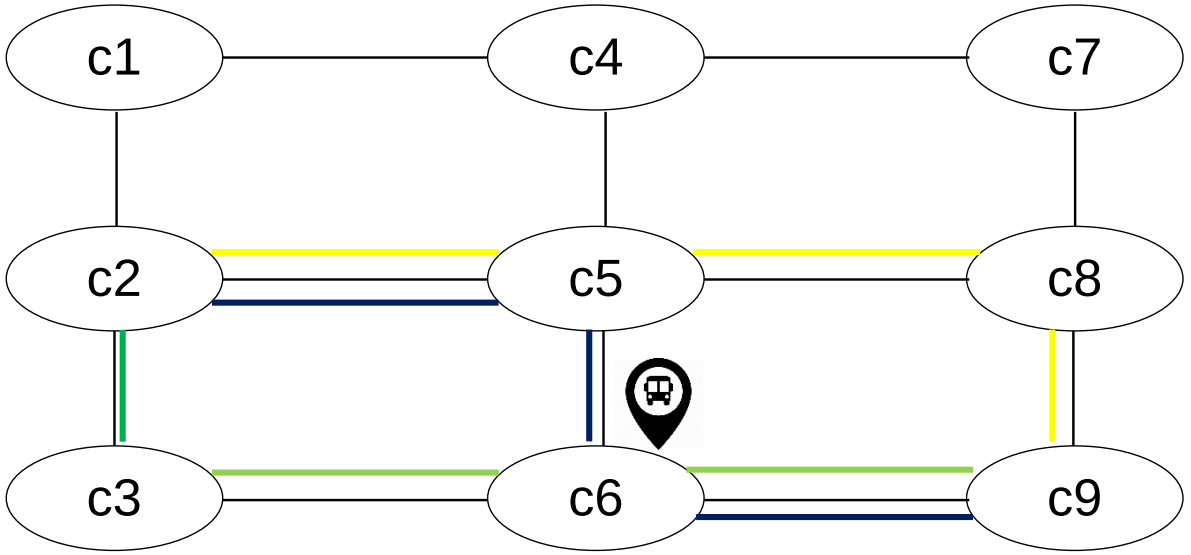
Results

Plans	Costs
Plan 1	210
Plan 2	280
Plan 3	340



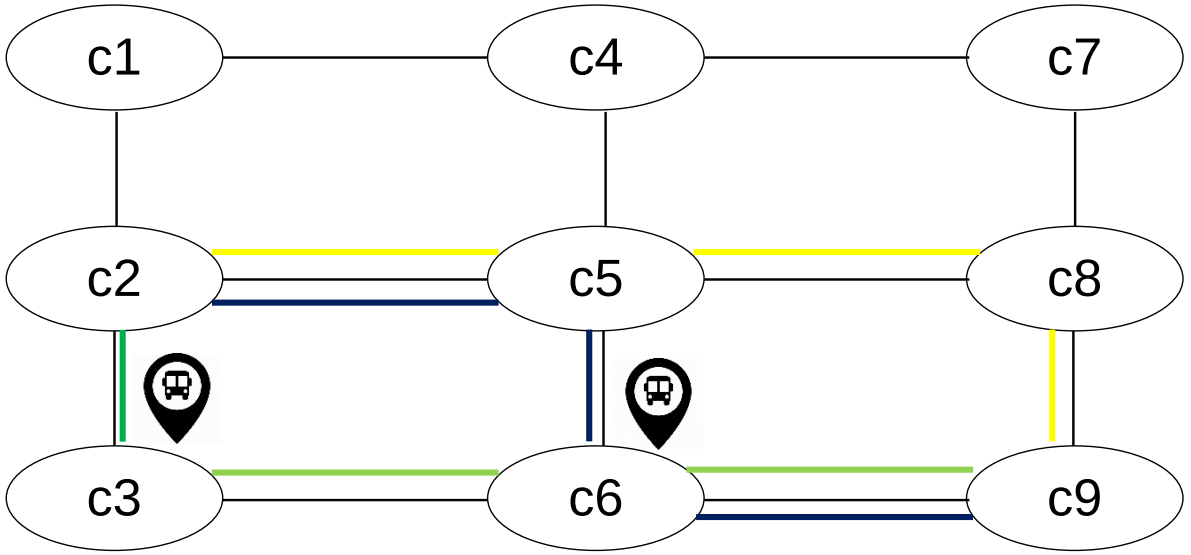
Bus Stop

Plans	Costs
Plan 1	210
Plan 2	280
Plan 3	340



Bus Stop

Plans	Costs
Plan 1	210
Plan 2	280
Plan 3	340



Road Maps

```
(define (domain planning)
  (:requirements :strips :action-costs)
  (:predicates (in ?x) (visited ?x) (not-visited ?x)
      (starting ?x) (complete) (not-complete) (distance ?x ?y)
      (connected ?x ?y))
  (:functions
   (distance ?x ?y)
   (total-cost)
   )
  (:action go-along
    :parameters (?x ?y)
    :precondition (and (in ?x) (not-visited ?y) (not-complete)
      (connected ?x ?y))
    :effect (and (not (in ?x))  (increase (total-cost) (distance ?x ?y)) (in ?y) (visited
?y) (not (not-visited ?y)) ))

 )
```

Typing requirements allows more than one bus and bus stops

Add a “visited” and “complete route” predicates to monitor if the route contemplates all the required bus stops.

Add effect (bus stop visited)

Road Maps

```
(define (problem poa)
  (:domain planning)
  (:objects c1 c2 c3 c4 c5 c6 c7 c8 c9)
  (:init (connected c1 c2) (connected c1 c4) (connected c2 c5) (connected c2 c3) (connected c3 c6)
(connected c4 c5) (connected c5 c6) (connected c5 c8)
         (connected c6 c9) (connected c7 c8) (connected c8 c9)
         (= (distance c1 c2) 120) (= (distance c2 c1) 120) (= (distance c1 c4) 120) (= (distance c4
c1) 120) (= (distance c2 c5) 60) (= (distance c5 c2) 60)
         (= (distance c2 c3) 120) (= (distance c3 c2) 120) (= (distance c3 c6) 120) (= (distance c6
c3) 120) (= (distance c4 c5) 120) (= (distance c5 c4) 120)
         (= (distance c5 c6) 120) (= (distance c6 c5) 120) (= (distance c5 c8) 50) (= (distance c8 c5)
50) (= (distance c6 c9) 100) (= (distance c9 c6) 100)
         (= (distance c7 c8) 100) (= (distance c8 c7) 100) (= (distance c7 c8) 100) (= (distance c8
c7) 100) (= (distance c8 c9) 100) (= (distance c9 c8) 100)
         (visited c1) (not-visited c2) (not-visited c3)
         (not-visited c4) (not-visited c5) (not-visited c6)
         (not-visited c7) (not-visited c8) (not-visited c9)
         (= (total-cost) 0)
         (in c2) (starting c2) (not-complete))
  (:goal (visited c9))
   (:metric minimize (total-cost))
  )
```

Import nodes connections from Open Street Maps

Thank You!

gabriel.figlarz@edu.pucrs.br