espol Facultad de Ciencias Naturales y Matemáticas

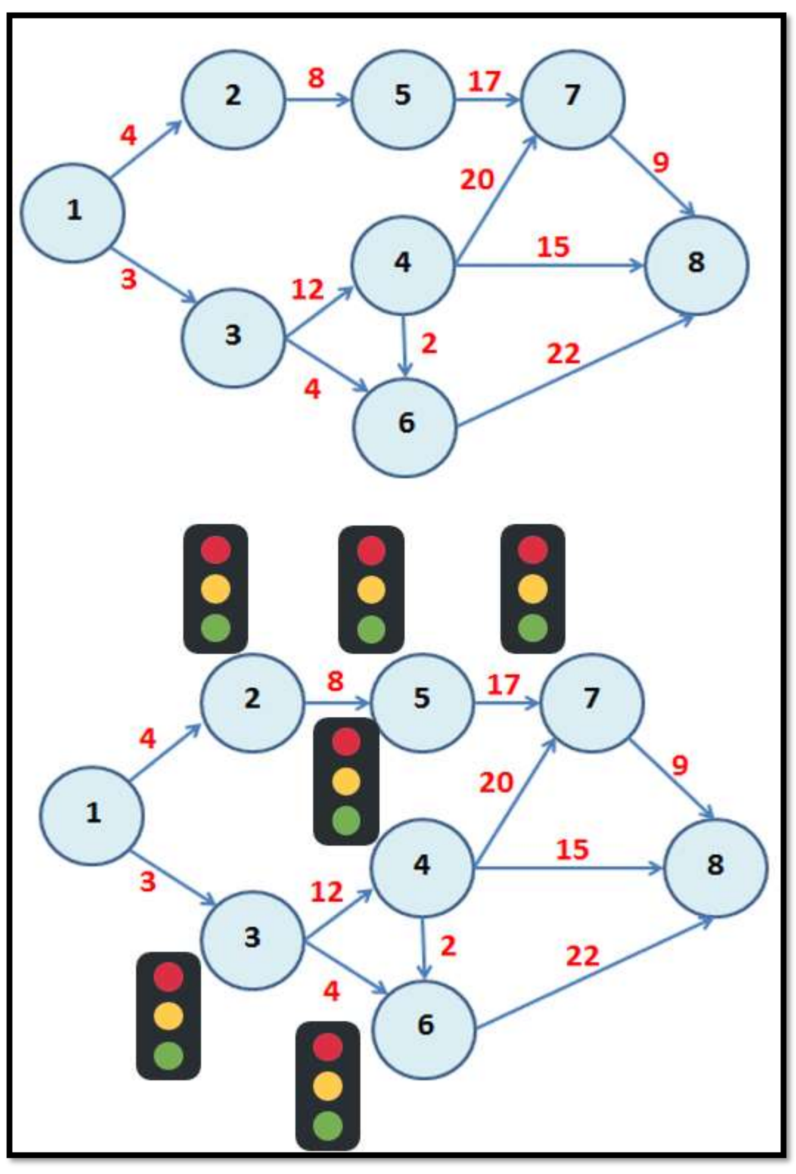
Joddel Varas
jbvaras@espol.edu.ec

Hector Alvarado
Correo espol

Análisis experimental del problema del camino más corto con restricciones en los nodos de un grafo.

PROBLEMA

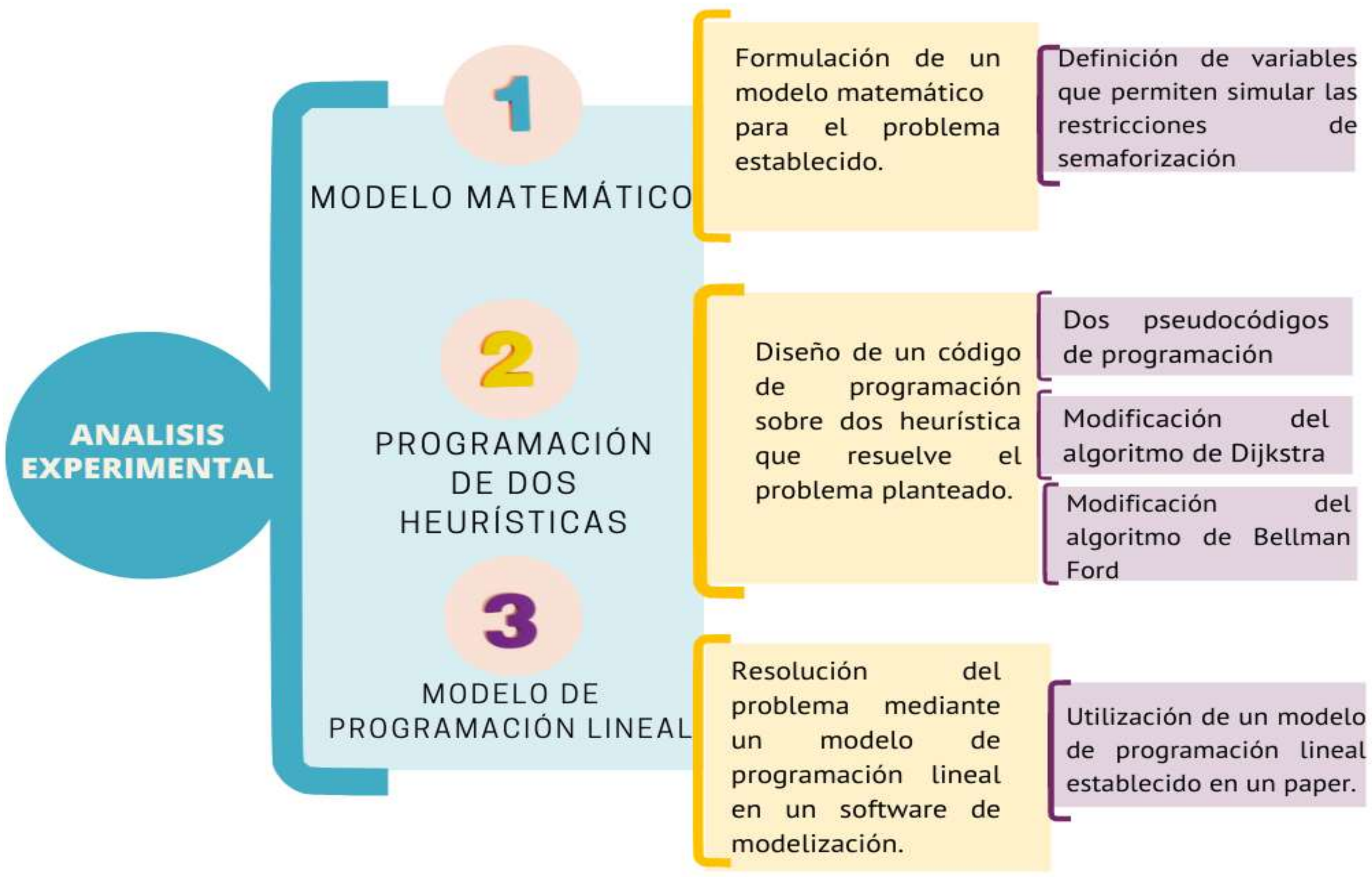
El consiguiente trabajo se basa en la variante de un problema ya conocido, cómo lo es el del camino más corto (SPP). Esta variante se basa en establecer restricciones de semaforización apoyado en la relación con las ventanas de tiempo, siendo una generalización del problema del camino mínimo con restricciones de tiempo (TCSPP).



OBJETIVO GENERAL

Analizar experimentalmente el problema de la ruta más corta con restricciones en los nodos mediante un algoritmo de ruta más corta, utilizando diferentes métodos de optimización para determinar un camino óptimo.

PROPUESTA



Softwares

RESULTADOS

RESULTADOS DE LA MODELIZACIÓN

En los resultados de la implementación del modelo de programación lineal en GAMS, se tiene la de las variables de decisión los cuales representan, el camino más corto entre ir del nodo origen i al nodo destino d.

Variable $y(v_i, ws_{v_{i,k}})$ *si la ventana k es tomada en cuenta para el nodo* v_i

Variable	Resultado
$y(v_1, ws_{v_{1,1}})$	1
otros	0

Variable $x(v_i, v_j)$ *si el nodo* v_i *es visitado por el nodo* v_j

Variable	Resultado
$x(i, v_1), x(v_1, d)$	1
otros	0

Variable $t(v_i)$ *tiempo óptimo de salida del nodo* v_i

Variable	Resultado
$t(v_1)$	0,5444
$t(d)$	14,328
otros	0

RESULTADOS DE LA PROGRAMACIÓN DE LAS HEURÍSTICAS

NODOS	DISTANCIA MAS CORTA (DIJKSTRA)	PREDECESOR	DISTANCIA MAS CORTA (BELLMAN)	PREDECESOR
i	0	-	0	-
v_1	0,544	i	0,544	i
v_2	5,6	v_1	5,6	v_1
v_3	5,2	i	5,2	i
v_4	8,972	v_3	8,972	v_3
v_5	6,5277	v_2	6,5277	v_2
v_6	7,95555	v_2	7,95555	v_2
v_7	11,1047	v_5	11,1047	v_5
d	14,3277	v_1	14,3277	v_1

Se puede observar la variación de las distancias mínimas de ir del punto de origen i a cada uno de los nodos por parte de los algoritmos de Dijkstra.

NODOS	DISTANCIA MAS CORTA (DIJKSTRA MODIFICADO)	PREDECESOR	DISTANCIA MAS CORTA (BELLMAN MODIFICADO)	PREDECESOR
i	0	-	0	-
v_1	0,544	i	0,544	i
v_2	5,605499	v_1	5,605499	v_1
v_3	5,2	i	5,2	i
v_4	9,182	v_3	9,182	v_3
v_5	6,71222	v_2	6,71222	v_2
v_6	8,14	v_2	8,14	v_2
v_7	11,289222	v_5	11,289222	v_5
d	14,3277	v_1	14,3277	v_1

Ruta más corta, Pascuales-Samanes

Pseudocódigo de Dijkstra modificado

```
Entrada:
   Dígrafo G,          Grafo a estudiar.
   Pesos w:E(G)→ℝ⁺;   E(G)  es un conjunto de pares ordenados que relaciona
dos vértices.
   Vértice i ∈V(G) ;  donde i es el vértice de inicio

   Vértice d ∈V(G) ;  donde d es el vértice de destino objetivo

Salida:
   l(v) ∀v∈V−{i} ,  donde l(v) es el tiempo mínimo de llegar a v desde i.
   p(v) ∀v∈V−{i} ,  donde p(v) es el predecesor de v.

1 establecer
  l(i)=0
  M = Número muy grande
  R= Ø
  tripleta(h,v,g,k), tripleta de ventanas de tiempo k para el nodo v donde
el                   predecesor es h y el destino es g

  a(v,k)= límite inferior de la ventana de tiempo k para el nodo v
  b(v,k)=límite inferior de la ventana de tiempo k para el nodo v

2 Para todo v ∈ V-{i] hacer
  l(v)=M


3 "Encontrar un vértice v donde v ∈ V(G) y v∉ ℝ; tal que l(v)= min l(a)
donde   a ∈ V(G) y a∉R.

4 definir R=R U {V}

5 Para todo a donde ∈ V(G) Y a∉R; tal que (v,a)∈ E(G) hacer;
6   Si v≠i y v≠d entonces
7     Si a(v,k)≤ l(v)≤ b(v,k) entonces
8       Si l(a) >l(v)+w(v,a) entonces
          definir
          l(a)=l(v)+w(v,a)
          p(a)=v
7     Caso contrario;
        l(v)=a(v,k) donde k es la primera ventana de tiempo que se encuentra
la tripleta (h,v,a)
8       Si l(a)> l(v)+w(v,a) entonces
          definir
          l(a)=l(v) +w(v,a)
          p(a)=v


6   Caso contrario
7      si l(a)>l(v) * w(v,a) entonces
         definir
         l(a)= l(v) + w(v,a)
         p(a)=v
```

CONCLUSIONES

- Se pudo comprobar que los pseudocódigos y los códigos de programación tanto del algoritmo modificado de Dijkstra y el del Bellman Ford permiten obtener los resultados óptimos del problema de restricciones de semaforización.
- Se aportó además una modelización en GAMS de un modelo de programación Lineal que nos permita resolver el problema determinado.
- Fomenta la investigación a nuevas consideraciones del problema, por ejemplo, una de las modificaciones podría ser la necesidad de que ciertos nodos del grafo formen parte del camino más corto, es decir, que obligatoriamente sean visitados o considerar que algunos nodos no contengan restricciones de semaforización para así tomar en cuenta más todas las intersecciones de una red.
- Así mismo, en el modelo de programación lineal se puede modificar la función objetivo, en donde nuestra función objetivo sea obtener la ruta con menos tiempos de espera en las intersecciones.

Facultad de Ciencias Naturales y Matemáticas

Joddel Varas
Correo espol

Hector Alvarado
hfalvara@espol.edu.ec

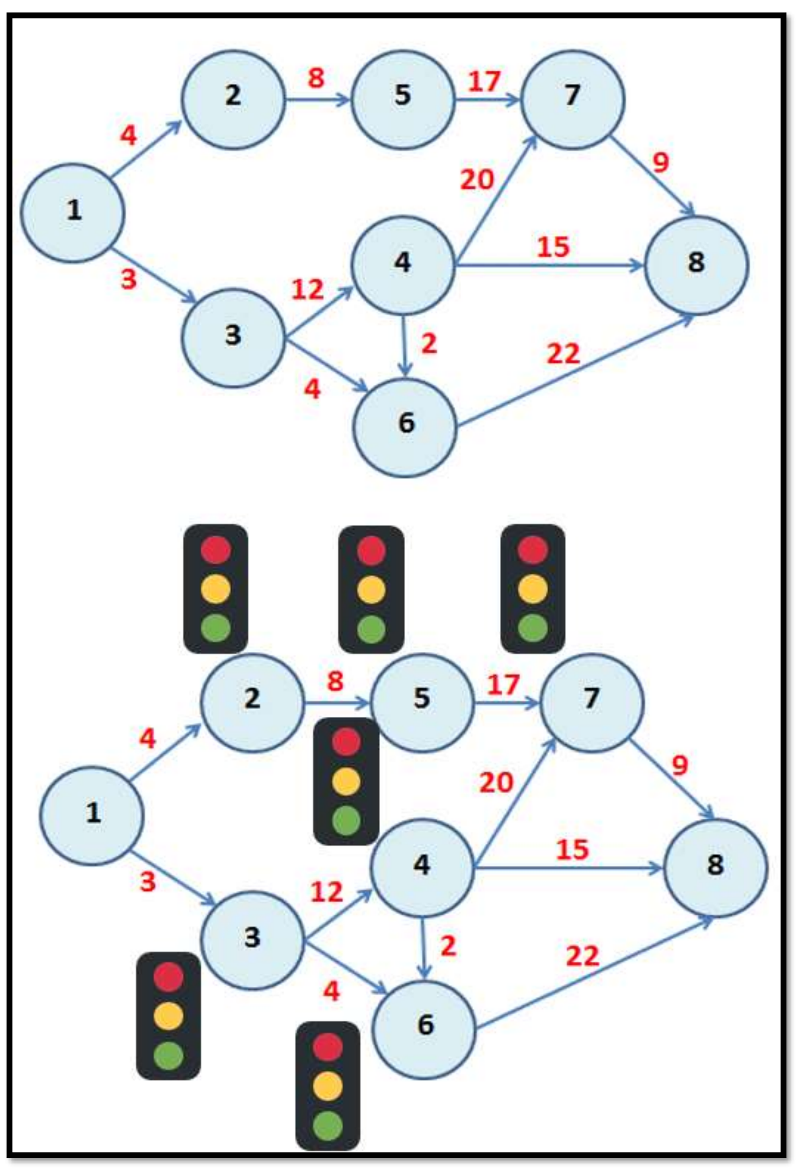
Experimental analysis of the shortest path problem with constraints on the nodes of a graph.

PROBLEM

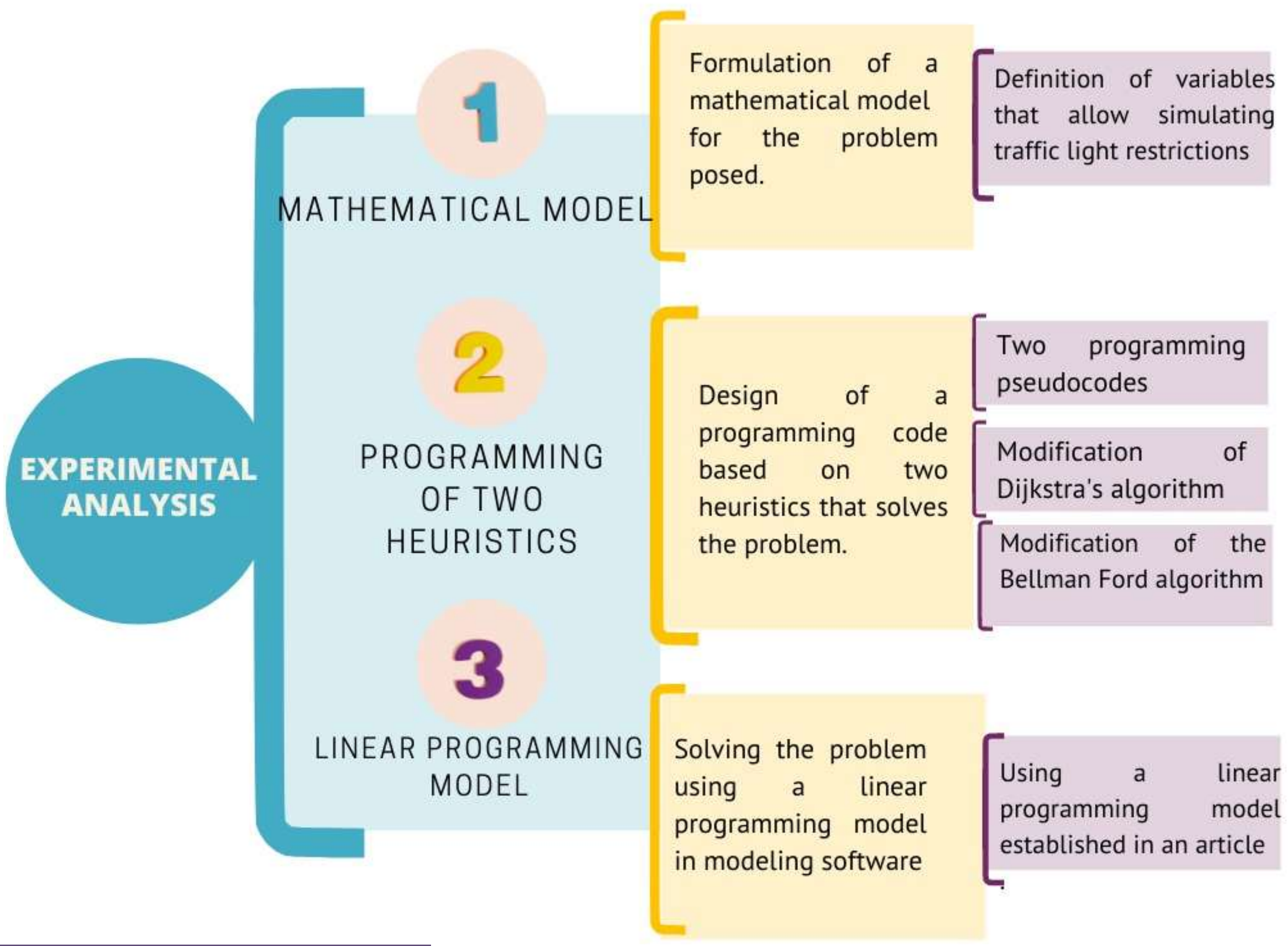
The work is based on a variant of a well- known shortest path problema (SPP). This variant is based on establishing traffic light restrictions based on the relationship with the time Windows, being a generalization of the time-constrained mínimum path problema (TCSPP).

GENERAL OBJETIVE

Experimentally analyze the shortest path problema with node constraints by means of a shortest path algorithm, using different optimization methods to determine an optimal path



PROPOSAL



SOFTWARE

Microsoft Excel

Phyton

Gams

RESULTS

MODELING RESULTS

In the results obtained from the impementation of the linear programming model in GAMS, we have the decisión variables which represent the shortedt path between going from the origin node ***i*** to the destination node ***d.***

Variable $y(v_i, ws_{v_{i,k}})$ *if the window k is taken into account for node* v_i	
Variable	Result
$y(v_1, ws_{v_{1,1}})$	1
others	0

Variable $x(v_i, v_j)$ *if node* v_i *is visited by node* v_j	
Variable	Result
$x(i, v_1), x(v_1, d)$	1
others	0

Variable $t(v_i)$ *optimal node exit time* v_i	
Variable	Result
$t(v_1)$	0,5444
$t(d)$	14,328
others	0

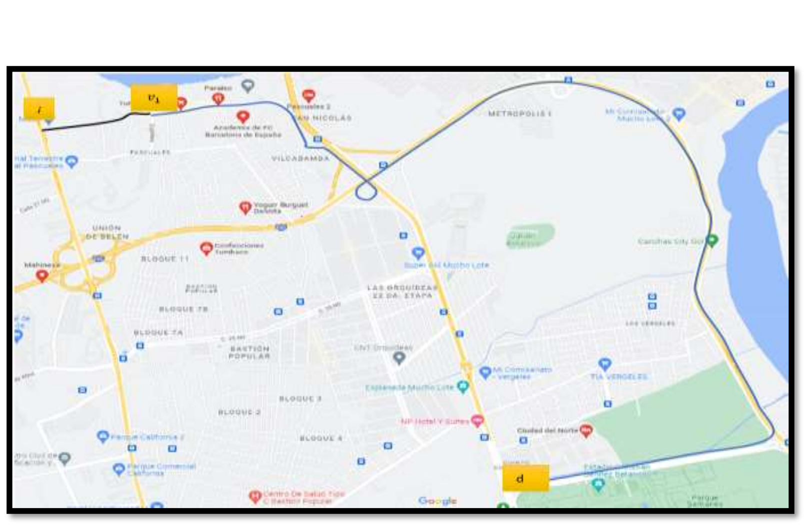
RESULTS OF HEURISTICS PROGRAMMING

NODES	SHORTEST DISTANCE (DIJKSTRA)	PREDECESSOR	SHORTEST DISTANCE (BELLMAN)	PREDECESSOR
i	0	-	0	-
v_1	0,544	i	0,544	i
v_2	5,6	v_1	5,6	v_1
v_3	5,2	i	5,2	i
v_4	8,972	v_3	8,972	v_3
v_5	6,5277	v_2	6,5277	v_2
v_6	7,95555	v_2	7,95555	v_2
v_7	11,1047	v_5	11,1047	v_5
d	14,3277	v_1	14,3277	v_1

The variation of the mínimum distances of going from the origin point *i* to each of the nodes by Dijkstra's algorithms can be observed

NODES	SHORTEST DISTANCE (DIJKSTRA)	PREDECESSOR	SHORTEST DSITANCE (BELLMAN)	PREDECESSOR
i	0	-	0	-
v_1	0,544	i	0,544	i
v_2	5,605499	v_1	5,605499	v_1
v_3	5,2	i	5,2	i
v_4	9,182	v_3	9,182	v_3
v_5	6,71222	v_2	6,71222	v_2
v_6	8,14	v_2	8,14	v_2
v_7	11,289222	v_5	11,289222	v_5
d	14,3277	v_1	14,3277	v_1

Shortest route, Pascuales-Samanes



Modified Dijkstra's pseudocode

```
Entrada:
   Dígrafo G,          Grafo a estudiar.
   Pesos w:E(G)→ℝ⁺;   E(G)  es un conjunto de pares ordenados que relaciona
dos vértices.
   Vértice i ∈V(G) ;  donde i es el vértice de inicio

   Vértice d ∈V(G) ;  donde d es el vértice de destino objetivo

Salida:
   l(v) ∀v∈V−{i} ,   donde l(v) es el tiempo mínimo de llegar a v desde i.
   p(v) ∀v∈V−{i} ,   donde p(v) es el predecesor de v.

1 establecer
  l(i)=0
  M = Número muy grande
  R= Ø
  tripleta(h,v,g,k), tripleta de ventanas de tiempo k para el nodo v donde
el                    predecesor es h y el destino es g

  a(v,k)= límite inferior de la ventana de tiempo k para el nodo v
  b(v,k)=límite inferior de la ventana de tiempo k para el nodo v

2 Para todo v ∈ V-{i] hacer
  l(v)=M


3 "Encontrar un vértice v donde v ∈ V(G) y v∉ ℝ; tal que l(v)= min l(a)
donde    a ∈ V(G) y a∉R.

4 definir R=R U {V}

5 Para todo a donde ∈ V(G) Y a∉R; tal que (v,a)∈ E(G) hacer;
6   Si v≠i y v≠d entonces
7     Si a(v,k)≤ l(v)≤ b(v,k) entonces
8       Si l(a) >l(v)+w(v,a) entonces
          definir
          l(a)=l(v)+w(v,a)
          p(a)=v
7     Caso contrario;
        l(v)=a(v,k) donde k es la primera ventana de tiempo que se encuentra
la tripleta (h,v,a)
8       Si l(a)> l(v)+w(v,a) entonces
          definir
          l(a)=l(v) +w(v,a)
          p(a)=v


6   Caso contrario
7      si l(a)>l(v) * w(v,a) entonces
         definir
         l(a)= l(v) + w(v,a)
         p(a)=v
```

CONCLUSIONS

- It was possible to prove that the pseudocodes and the programing codes of both the modified Dijkstra's algorithm and the Bellman Ford's allow obtaining the optimal results of the semaphoresis constraint problema.
- It was also provided a modeling in GAMS of a Linear programming model that allows us to solve the given problema.
- Encourage research into new considerations of the problema, for example one of the modifications could be the need for certain nodes of the network to be part of the shortest path, that they must be complsorily visited or consider that some nodes do not contain traffic light restriccions in order to take into account more all the intersections of a network.
- Likewise, in the linear programming model we can modify the objetive function, where our objetive function into obtain the route with the shortest waiting time at intersections.