



INSTITUTO DE COMPUTACIÓN
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE LA REPÚBLICA



Proyecto de Grado de Ingeniería en Computación

DISEÑO ÓPTIMO DE REDES DE CICLOVÍAS

Autores:

GONZALO MERCADANTE ROMERO
MARÍA JOSÉ RABAZA CHAVES
MARÍA FERNANDA TOLEDO LEDESMA

Supervisor: ANTONIO MAUTTONE

Montevideo, Uruguay

2016

Resumen

En este trabajo se estudia el problema del diseño de redes de ciclovías utilizando técnicas de optimización. Diseñar la red implica seleccionar en cuáles calles se construirán ciclovías, considerando los costos asociados y teniendo en cuenta que se tiene un presupuesto límite para la construcción de las mismas. Además se debe tener en cuenta que hay cierta demanda de viajes que debe satisfacerse, optimizando los objetivos de los usuarios, es decir los caminos asociados a dichos viajes.

Se propone un modelo de programación lineal entera mixta para el problema del diseño óptimo de redes de ciclovías, basado en una formulación general de diseño de redes de transporte. El mismo considera las discontinuidades, las cuales son aquellos puntos en los que una ciclovía o bicisenda llega a su fin y el ciclista debe continuar su camino fuera de ella. Además, se propone un algoritmo para obtener una solución aproximada, basado en la metaheurística GRASP.

Se prueba la propuesta con redes pequeñas y casos representativos de ciudades reales para evaluar la efectividad del algoritmo y su comportamiento ante la variación de presupuesto y costos. También se valida con un caso de estudio relativo a la ciudad de Montevideo. La construcción del modelo computacional para el caso de estudio implicó obtener información acerca de la red vial y de la demanda de viajes.

Se logró aplicar una técnica de optimización en redes al diseño óptimo de ciclovía y se validó la propuesta con varias instancias del problema, incluida una de características realistas.

Palabras clave. Ciclovía, optimización de redes de ciclovías, diseño de redes de ciclovías, GRASP.

Índice general

Introducción	6
1.1 Objetivos y alcance del trabajo	8
1.2 Estructura del informe	9
Revisión bibliográfica	10
2.1. Discontinuidades	11
2.2. Pendientes	12
2.3. Ciclovías con sentido opuesto	13
2.4 Aplicación de las características al modelo propuesto	13
Marco metodológico	16
Modelo de optimización para el problema del diseño de redes de ciclovías	22
4.1 Correspondencia de elementos de la realidad a los elementos del modelo	22
4.2 Formulación de programación matemática	24
Metaheurística GRASP para el diseño óptimo de redes de ciclovías	28
5.1 Notación	28
5.2 Construcción de soluciones	29
5.3 Evaluación de soluciones	33
5.4 Mejora de soluciones	36
Implementación	38
6.1 Procesamiento de datos de entrada	38
6.2 Método de optimización	39
6.3 Procesamiento de datos de salida y visualización	40
Caso de Estudio: Montevideo	42
7.1 Situación actual	42
7. 2 Construcción del caso	44
7.2.1 Preparación de los datos	45
7.2.1.1 Zonificación de la región de estudio	45
7.2.2.2 Construcción de la red	47
7.3 Visualización de los datos	53
Experimentos computacionales	56
8.1 Efectividad del algoritmo	57
8.2 Análisis de sensibilidad respecto al parámetro de presupuesto.	66
8.3 Análisis de sensibilidad respecto a costos y penalizaciones.	67
8.4 Aplicación del algoritmo al caso de estudio Montevideo	68
8.4.1 Análisis de resultados.	70
Conclusiones y trabajos a futuro	76
Anexo A: Archivo .mod AMPL	80
Anexo B: Manual de instalación y ejecución	82

Capítulo 1

Introducción

El uso de la bicicleta como medio de transporte urbano para los viajes habituales (aquellos que se realizan a diario por motivo de trabajo o estudio), se ha impulsado en los últimos tiempos como una medida para paliar los problemas de movilidad existentes en muchas ciudades del mundo. En el caso más general, el éxito en lograr una adopción masiva de este modo de transporte radica en que sus potenciales usuarios tengan condiciones adecuadas para poder utilizarlo. Más allá de las condiciones climáticas y geográficas, la existencia de una infraestructura adecuada es un factor significativo en la adopción de la bicicleta como medio de transporte [1, 2]. Uno de los componentes fundamentales de dicha infraestructura son las vías destinadas de forma exclusiva o compartida para la circulación de bicicletas, conocidas como ciclovías o biciesendas. Es importante diferenciar el término ciclovía de biciesenda [3]. La ciclovía es un espacio dentro de la calle delimitada virtualmente con líneas pintadas en el pavimento o físicamente con cordones de cemento o similar (Figura 1.1). La biciesenda, a diferencia de la ciclovía, no se encuentra sobre la calle, sino que suele ubicarse sobre las veredas o senderos dentro de plazas y parques (Figura 1.2).



Figura 1.1: Ciclovía sobre la calle Guayabos en la ciudad de Montevideo [4]



Figura 1.2: Bicisenda sobre la vereda del Parque Rodó en la ciudad de Montevideo [5]

El diseño de una *red de ciclovías* es crucial para un futuro donde la bicicleta se masifique como modo de transporte. Este diseño se debe realizar teniendo en cuenta no solo a los ciclistas sino también a las necesidades del resto de la sociedad. Una ciclovía implica una facilidad al ciclista pero reduce el espacio para otros vehículos y peatones, tiende a imponer límites de velocidad más bajos y menos espacios para estacionar. Además los gobiernos tienen limitaciones presupuestales para diferentes sectores, por lo tanto se deben optimizar los recursos.

Las decisiones referentes a dónde construir ciclovías deben tener en cuenta diversos elementos, como ser la infraestructura subyacente (por ejemplo calles, parques o plazas), junto con sus características geográficas (distancias, pendientes). Y por otro lado se debe tener conocimiento acerca de los viajes que son realizados en bicicleta y de aquellos que se desean inducir, en particular su origen geográfico, destino y cantidad. También deben contemplarse los objetivos de los usuarios, los del resto de los ciudadanos y las restricciones de infraestructura y presupuesto.

La tarea de diseñar una red de ciclovías urbana con criterios racionales y teniendo en cuenta todos los elementos del problema descritos anteriormente puede ser abordada mediante técnicas de optimización. Este Proyecto de Grado es una primera aproximación en el grupo de investigación en el que se enmarca, al estudio de técnicas de optimización aplicadas al diseño de redes de ciclovías urbanas.

Comprende el desarrollo de modelos y algoritmos de optimización, así como procesamiento de datos e información geográfica necesarios para la aplicación de los mismos.

1.1 Objetivos y alcance del trabajo

El presente trabajo plantea el estudio del problema de diseño de redes de ciclovías utilizando técnicas de optimización. Los objetivos son:

- Conocer las diferentes características a tener en cuenta a la hora de diseñar redes de ciclovías, así como también aquellos estudios existentes sobre problemas de optimización similares.
- Realizar una exploración de posibles soluciones al diseño de la redes de ciclovías, lo que incluye investigar trabajos académicos relacionados.
- Proponer una solución al problema de diseño de redes de ciclovías en base a técnicas de optimización.
- Probar con un caso de estudio real, lo cual comprende obtener los datos y procesarlos.

Considerando los objetivos previamente mencionados, el alcance el proyecto comprende:

- Plantear un modelo de optimización para el problema, contemplando las diferentes características relevadas a la hora de diseñar redes de ciclovías.
- Resolver el modelo planteado, aplicando un método de optimización exacto, estudiando sus limitaciones, y en caso de ser necesario, desarrollar un método aproximado y estudiar la calidad de su aproximación.
- Aplicación a un caso de estudio real. Con el objetivo de validar la propuesta para la solución del problema de diseñar redes de ciclovías, se requiere un caso real que permita interpretar las soluciones obtenidas en términos de una realidad concreta, en este caso la ciudad de Montevideo. Para este objetivo en particular es necesario obtener los datos geográficos de la ciudad, particularmente su red de calles, así como también la matriz origen-destino de viajes que son realizados en bicicleta.

1.2 Estructura del informe

En el capítulo 2 se hace una revisión bibliográfica sobre las características deseables y las problemáticas que surgen en el diseño de redes de ciclovías. En el capítulo 3 se presentan conceptos básicos que serán utilizados a lo largo del trabajo. En el capítulo 4 se propone un modelo de programación lineal entera mixta para el problema del diseño óptimo de redes de ciclovías, basado en una formulación general de diseño de redes de transporte. En el capítulo 5 se propone un algoritmo basado en la metaheurística GRASP para la resolución aproximada del modelo de optimización propuesto. En el capítulo 6 se presentan algunos aspectos referentes a la implementación del sistema. En el capítulo 7 se presentan los procesamientos realizados con los datos necesarios para la construcción del caso de estudio de la ciudad de Montevideo. En el capítulo 8 se presentan las pruebas y resultados numéricos del método de optimización propuesto para diferentes instancias del problema, y en particular para el caso de estudio de Montevideo. En el capítulo 9 se presentan las conclusiones y posibles extensiones de este trabajo.

En el anexo A se presenta el archivo .mod correspondiente a la formulación del modelo exacto en lenguaje AMPL, para ejecutar el método exacto en el solver CPLEX. En el anexo B se brinda un manual de instalación para un correcto funcionamiento de la aplicación de software desarrollada y un manual de ejecución de la misma.

Capítulo 2

Revisión bibliográfica

En este capítulo se realiza una recopilación de los principales aspectos a tener en cuenta en el diseño de redes de ciclovías y los desafíos que el mismo presenta. Esta información fue extraída de un conjunto de artículos académicos y de literatura gris (manuales de agencias de gobierno, reportes de consultoría).

El éxito de una ciclovía depende de si los ciclistas prefieren hacer uso de ella o prefieren seguir su camino habitual previo a la existencia de la misma. Es importante que las ciclovías presenten una solución en vez de un problema. A la hora de diseñar una ciclovía, es elemental tomar en cuenta las necesidades de los ciclistas.

En la bibliografía investigada [6, 7, 8, 9, 10] se mencionan características deseables a tener en cuenta, las cuales se pueden agrupar de la siguiente manera:

- **Seguridad:** Una buena infraestructura debería hacer sentir seguro al ciclista, se debería disminuir el riesgo de accidentes, particularmente cuando se enfrentan a circular cruces de calles.
- **Dirección:** Las ciclovías deben ser lógicas y continuas, evitando obstáculos innecesarios, retrasos y distracciones. Las mismas deben ser planeadas como parte de una red.
- **Comodidad:** Las superficies para el ciclismo deben ser suaves, planas y bien mantenidas.
- **Coherencia:** Las indicaciones y señales deben ser legibles, intuitivas, consistentes y entendibles para todos los usuarios.
- **Atractivo:** La infraestructura debe ser lo más “delicada” posible, evitando agregar desorden en las calles y en términos generales debería mejorar el ámbito urbano.
- **Adaptabilidad:** La infraestructura debe ser diseñada para dar espacio a un posible aumento de los usuarios en el tiempo. Donde las calles han sido diseñadas principalmente para su uso por los vehículos motorizados, es difícil hacer la infraestructura para el ciclismo legible y adaptable.

A los efectos de este estudio se destacarán el análisis de las discontinuidades como parte de la dirección, seguridad y comodidad, las pendientes como parte de la comodidad, y las ciclovías con sentido opuesto al de la calle, que también es parte de la dirección.

2.1. Discontinuidades

El concepto de discontinuidad se utiliza para referirse a aquellos puntos en los que una ciclovía o biciesenda llega a su fin y el ciclista debe continuar su camino fuera de ella. Las discontinuidades presentan un reto para los ciclistas debido a que el fin de una ciclovía significa que los ciclistas deben abandonar el camino especialmente diseñado para ellos y fusionarse nuevamente con el tráfico motorizado.

Más allá de que existen estudios que indican que no hay pruebas de que pedalear en conjunto con los vehículos motorizados sea más peligroso que trasladarse por ciclovías separadas del mismo [11], otros indican que parece ser de una percepción general que las ciclovías promuegan una sensación de seguridad y comodidad para el ciclista [12, 13, 14, 15]. Por lo tanto las discontinuidades deberían ser analizadas cuidadosamente y evitarlas dentro de lo posible para aumentar la comodidad y seguridad de la red en general.

El estudio realizado en [16] se concentró justamente en cuantificar el efecto de las discontinuidades sobre el nivel de comodidad y seguridad percibido por los ciclistas en las ciclovías de Minneapolis, Minnesota, Estados Unidos. Esta ciudad es de gran interés ya que cuenta con el tercer porcentaje más alto de ciclistas en ciudades de dicho país. Fueron identificadas 30 discontinuidades a lo largo de las rutas de Minneapolis (ver Figura 2.1). El estudio consistió en pedirle a voluntarios que realizaran un recorrido en el cual se enfrentaran con al menos diez de las discontinuidades identificadas y votaran según su nivel de comodidad. En total, 28 participantes cumplieron con la encuesta y las respuestas indicaron, en general, menor conformidad en caminos discontinuos.

De las conclusiones del estudio cabe destacar dos ideas importantes. La primera es que las ciclovías sobre la calle juegan un gran papel en una red de ciclovías urbana, ya que ayuda a los ciclistas inexpertos a aumentar su sentimiento de seguridad y por lo tanto aumentar las posibilidades de que dichas personas se trasladen más en bicicleta. Y por otro lado que la terminación de las ciclovías presentan un riesgo considerable, siendo estas discontinuidades debilidades de la

red que terminan afectando la decisión de muchas personas de adoptar dicho modo de transporte. Por estas razones es que se debe prestar especial atención en tratar de minimizar dichas instancias.

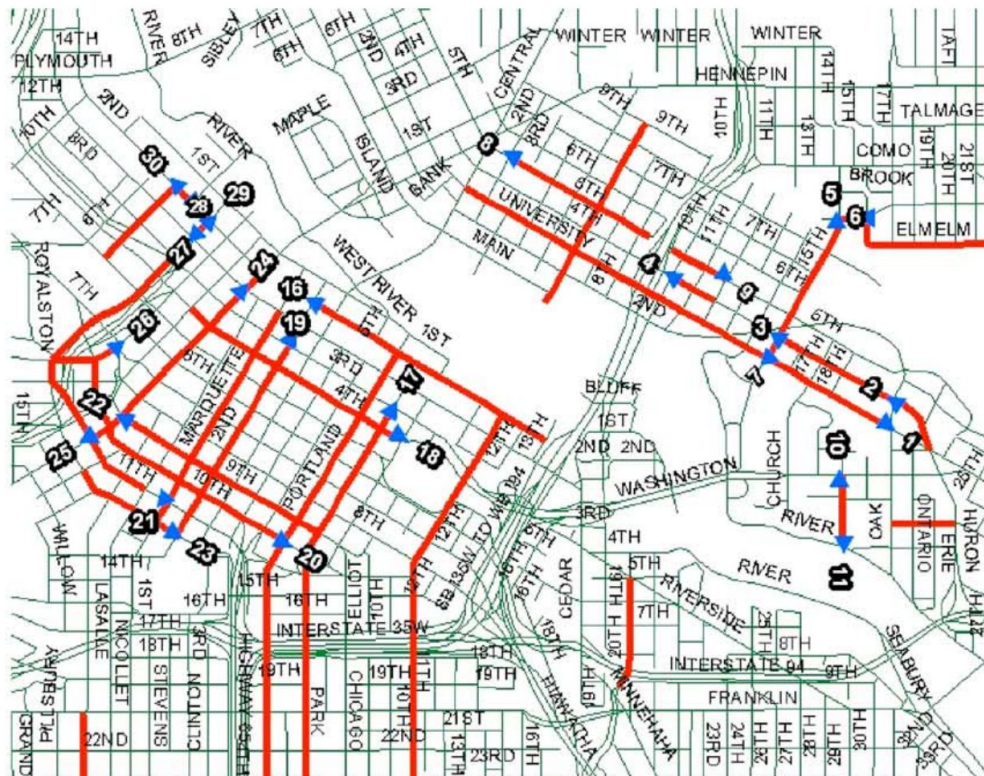


Figura 2.1. Algunas de las 30 discontinuidades identificadas en Minneapolis [16]

2.2. Pendientes

Las pendientes pronunciadas presentan un obstáculo y peligro potencial para los ciclistas. A medida que se desciende una pendiente la velocidad aumenta y puede causar problemas al ciclista para mantener el control o detenerse, lo cual puede provocar accidentes. No solo representan un peligro sino que además los ciclistas optarán por tomar otros caminos menos pronunciados. Cuando se utilizan bicicletas simples, sin velocidades, las pendientes positivas (repechos) presentan mayor dificultad e implican un gran esfuerzo físico, por lo que muchas personas preferirán desviarse y optar por un camino más largo con el objetivo de evitarlo.

También se debe tener en cuenta que al subir una pendiente el movimiento de la bicicleta suele ocupar más espacio ya que no se va en línea recta sino más

bien zigzagueando, lo que deriva en la necesidad de construir una ciclovía más ancha y por lo tanto más costosa.

Es por estas razones que a la hora de diseñar una ciclovía hay que considerar que los usuarios buscan minimizar tanto las distancias como las pendientes. En la literatura consultada hay una propuesta que para lograr esto considera el costo de cada tramo como el menor acumulativo asociado con la pendiente y distancia [17]. Si bien no se encontraron artículos que incluyeran el estudio de pendientes como un factor para el diseño de ciclovía se cree necesario tenerlas en cuenta para aportar a la comodidad de los ciclistas al utilizar esta infraestructura.

2.3. Ciclovías con sentido opuesto

Muchos estudios han sido realizados sobre los impactos que causan los diferentes estilos de diseño de ciclovías [18, 19, 20, 21], pero ninguno ha analizado en particular las ciclovías que permiten la circulación en ambas direcciones sobre calles flechadas para el tránsito de vehículos.

Lo que sí se puede decir es que las ciclovías de un único sentido introducen cierta barrera para los ciclistas, ya que pueden generar grandes desviaciones en el camino seguido por un ciclista. En consecuencia, muchos ciclistas que no querrán desviarse pueden terminar circulando de forma ilegal en el sentido contrario tanto por la calle o acera, poniendo en riesgo su seguridad y la de los demás. Esto causa que la red termine siendo muy incómoda para los ciclistas.

La solución a esto son sendas exclusivas que permiten circular a los ciclistas en ambos sentidos en calles flechadas. Estas ciclovías pueden existir siempre y cuando el ancho de la calle lo permita. En muchos casos el espacio para este tipo de ciclovías puede ganarse, por ejemplo, eliminando el estacionamiento en un lado de la calle, o en ambos lados [22].

2.4 Aplicación de las características al modelo propuesto

Las características expuestas en las secciones 2.1 y 2.3, a tener en cuenta a la hora de diseñar una red de ciclovías, se intentarán mapear en este trabajo a un modelo de optimización sobre redes, teniendo en cuenta los costos de los usuarios,

caminos, demandas y flujos sobre la red. También se definirán modelos y algoritmos de optimización sobre esta red.

El modelo propuesto se basa en un modelo general para problemas sobre redes de transporte con múltiples mercancías expuesto en [23], el cual fue adaptado para la realidad planteada.

Capítulo 3

Marco metodológico

En este capítulo se introducen algunos conceptos y notaciones que serán usados en los siguientes capítulos. En particular se presentan algunos conceptos de grafos [24, 25] y se describe brevemente la metodología GRASP (Greedy Randomized Adaptive Search Procedures) [26, 27].

Grafo. Un grafo G es una dupla $G = (V, A)$, donde V es un conjunto finito y no vacío de elementos llamados vértices y A es el conjunto cuyos elementos se componen de subconjuntos de V de cardinalidad dos, llamados aristas.

Grafo dirigido. Un grafo G^* es dirigido cuando sus aristas tienen asignadas direcciones, o sea cuando existe una relación de precedencia entre los elementos. Sus vértices se llaman nodos, y sus aristas arcos. En estos casos A es una familia de pares ordenados resultantes del producto cartesiano de V .

Cadena (concepto no orientado). Es una secuencia de aristas de G , tal que cada arista de la secuencia tiene un extremo común con la arista precedente y otra con la siguiente

Camino (concepto orientado). Es una cadena $\mu = \{u_1, u_2, \dots, u_q\}$ en la que para todo u_i (con $i < q$) el extremo terminal de u_i coincide con el extremo inicial de u_{i+1} .

Grafo conexo. Es aquel en el que para cada par de vértices de G , existe una cadena que los une.

En grafos dirigidos se definen 2 conceptos:

- **Débilmente conexo:** si existe una cadena (sin tener en cuenta la orientación) que une cada par de nodos distintos.

- **Fuertemente conexo:** si para cada par ordenado de nodos x e y , existe un camino que va de x a y .

Arco ponderado. Arco que tiene asociado un número específico llamado costo.

Camino más corto. Dado un grafo dirigido $G = (V, E)$ con arcos ponderados, se define el camino más corto entre un par de vértices como aquel cuya suma de los costos de las aristas que lo constituyen es mínima.

Algoritmo de Dijkstra. Dado un grafo dirigido $G = (V, E)$, con arcos ponderados, este algoritmo calcula los caminos de mínimo costo de un vértice a todos los demás [28].

Algoritmo de Yen. Algoritmo que calcula los k caminos más cortos para un grafo con costos no negativos. El algoritmo fue propuesto por Yen [29] y utiliza cualquier algoritmo que calcule el camino de menor costo, y luego encuentra las $k-1$ desviaciones del mejor camino. El costo del camino i es menor o igual que el costo del camino $i+1$ para $i < k$.

Optimización. Una instancia de un problema de optimización es un par (F, c) , donde F es un conjunto dado (dominio de puntos factibles) y c es la función de costo, un mapeo $c : F \rightarrow \mathbb{R}^1$. El problema es encontrar un $f \in F$ tal que $c(f) \leq c(y)$ para todo $y \in F$. Tal punto f es llamado solución óptima global a la instancia dada.

Un problema de optimización es un conjunto I de instancias de un problema de optimización.

Restricción. Sea $f : X \rightarrow Y$ una función del conjunto X al conjunto Y , tal que el dominio de f esté incluido en X ($\text{dom } f \subseteq X$). Si un conjunto A es un subconjunto de X , entonces la restricción de f al conjunto A es la función $f|_A : A \rightarrow Y$. En particular, $f|_A$ es la función en A tal que $(f|_A)(a) = f(a)$ para a en A . Una función g es la restricción de una función f a un cierto subconjunto del dominio de f si y sólo si el dominio de g es un subconjunto del dominio de f y $g(x) = f(x)$ para todo $x \in D_g$; es decir $g \subseteq f$ [30]. En otras palabras, una restricción de una función f es una nueva función $f|_A$ obtenida eligiendo un dominio A incluido en el original de f .

Espacio factible. El espacio factible de un problema de optimización contiene el conjunto de todas las soluciones que satisface las restricciones del problema [31].

Optimización combinatoria. Un problema de optimización combinatoria $P = (S, f)$ puede ser definido por:

- un conjunto de variables $X = \{x_1, \dots, x_n\}$;
- dominios de las variables D_1, \dots, D_n tal que $x_i \in D_i$;
- restricciones sobre las variables;
- una función objetivo f a ser minimizada, donde $f: D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$;

El conjunto de todas las soluciones factibles es $S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ satisface todas las restricciones}\}$. S es usualmente llamado el espacio de búsqueda, dado que cada elemento del conjunto puede verse como una solución candidata. Para resolver un problema de optimización combinatoria se debe encontrar una solución $s^* \in S$ que minimice el valor de la función objetivo, es decir, $f(s^*) \leq f(s)$ para todo $s \in S$. s^* es llamada solución óptima global de (S, f) y el conjunto $S^* \subseteq S$ es llamado el conjunto de de soluciones óptimas globales [32].

Programación lineal entera mixta. Un problema de programación entera mixta resulta cuando algunas de las variables en el modelo son reales y otras enteras. El modelo es entonces mixto. Cuando la función objetivo y las restricciones son lineales entonces es un problema de programación lineal entera mixta [33]. En estos casos, hay métodos de solución con probada eficiencia implementados en paquetes de software de optimización como ser CPLEX, GUROBI e EXPRESS [34].

Heurística. Son métodos de optimización para problemas específicos que explotan reglas conocidas, trucos o simplificaciones para obtener una solución de buena calidad al problema. Las heurísticas no garantizan encontrar una solución óptima pero son usualmente más rápidas que los métodos exactos. Por lo tanto el objetivo de las heurísticas no es encontrar una solución óptima sino obtener soluciones de calidad aceptable en tiempos de procesamiento admisibles [35].

Vecindad. Dado un punto factible $f \in F$ en un problema particular, una vecindad es el conjunto $N(f)$ de puntos “ceranos” a f . Formalmente, dado un problema de optimización con instancias (F, c) , una vecindad es un mapeo $N: F \rightarrow 2^F$ definido para cada instancia [36].

Óptimo local. Dada una instancia (F, c) de un problema de optimización y una vecindad N , una solución factible $f \in F$ es llamada óptima local con respecto a N cuando se cumple que $c(f) \leq c(g)$ para todo $g \in N(f)$ [36].

Metaheurísticas. Son procesos generales de alto nivel que coordinan heurísticas y reglas simples para encontrar soluciones de buena calidad a problemas de

optimización combinatoria computacionalmente difíciles. Las metaheurísticas están basadas en paradigmas distintos y ofrecen diferentes mecanismos de escape para soluciones óptimas locales. Se encuentran entre las soluciones estratégicas más eficientes para solucionar problemas de optimización combinatoria en la práctica y han sido aplicadas a un amplio conjunto de problemas académicos y del mundo real [37].

GRASP (Procedimientos de Búsqueda Ávidos Aleatorios Adaptativos) [27]. Es una metaheurística diseñada para resolver problemas de optimización combinatoria. Se basa en la ejecución repetida de un procedimiento de construcción de soluciones, en combinación con una exploración local que intenta mejorar la calidad de las mismas. En la Figura 3.1 se puede ver un pseudocódigo.

```
Procedure GRASP (in ListSize , in MaxIter , in Seed );  
1 for k = 1 to MaxIter do  
2   InitialSolution  $\leftarrow$  Construct_Greedy_Randomized_Solution(ListSize, Seed) ;  
3   LocalSearchSolution  $\leftarrow$  LocalSearch(InitialSolution) ;  
4   if cost(LocalSearchSolution) < cost(BestSolutionFound) then  
5     UpdateSolution(BestSolutionFound, LocalSearchSolution) ;  
6   end if;  
7 end for;  
8 return BestSolutionFound ;  
9 end GRASP.
```

Figura 3.1: GRASP, estructura general

GRASP recibe como entradas el número máximo de iteraciones GRASP MaxIter, el tamaño de la lista de candidatos usada en la Fase de Construcción ListSize y la semilla inicial Seed para el generador de números pseudo-aleatorios utilizada en la Construcción Inicial para seleccionar elementos al azar de la lista RCL (sigla en inglés de lista restringida de candidatos) mencionada en detalle a continuación.

En la fase de construcción se construye una solución factible (Figura 3.2) de forma ávida, ya que se parte de una solución vacía y se van agregando elementos hasta llegar a una solución factible. En cada paso, el elemento que es agregado a la solución en construcción (parcial) se selecciona de la lista RCL. Esta contiene los ListSize mejores elementos elegidos según una función que mide el beneficio de

incluirlos en la solución parcial. El siguiente elemento que se incluye en la solución parcial es elegido al azar de la lista RCL.

Una vez finalizada la fase de construcción, se devuelve la solución factible construida.

```
Procedure Construct_Greedy_Randomized_Solution (in ListSize ,in Seed);  
1 Solution  $\leftarrow \emptyset$  ;  
2 Incremental costs evaluation for the candidate elements;  
3 while not_feasible(Solution) do  
4   RCL  $\leftarrow$  the restricted candidate list;  
5   s  $\leftarrow$  select randomly an element from the RCL ;  
6   Solution  $\leftarrow$  Solution  $\cup$  {s} ;  
7   Incremental costs reevaluation;  
8 end while;  
9 return Solution;
```

Figura 3.2: Construcción inicial para GRASP

Las soluciones generadas por algoritmos ávidos no son necesariamente óptimas, incluso con respecto a vecindades simples. Una técnica de búsqueda local intenta mejorar las soluciones de forma iterativa, sustituyendo sucesivamente la solución actual por una mejor solución en un entorno de la solución actual. Se termina cuando no se encuentra una solución en la vecindad que mejore la mejor solución actual. En la Figura 3.3 se presenta un pseudocódigo de una búsqueda local simple.

```
Procedure LocalSearch (in Solution);  
1 while not_locally_optimal(Solution) do  
2   Find Neighbor_Solution  $\in N$  (Solution)  
   satisfying f (Neighbor_Solution) < f (Solution) ;  
3   Solution  $\leftarrow$  Neighbor_Solution ;  
4 end while;  
5 return Solution;
```

Figura 3.3: Búsqueda Local para GRASP

La aplicación de una construcción ávida pura generalmente produce una solución sin realizar una exploración del espacio de soluciones factibles. En el contexto de GRASP, la construcción de soluciones implica la toma de decisiones que dependen

de una variable aleatoria. Esto permite que a partir de distintas muestras de dicha variable, se obtengan distintas trayectorias de exploración del espacio de soluciones factibles. Esta diversificación en la búsqueda aumenta las chances de obtener una mejor solución. La metodología GRASP (Figura 3.1) consiste en repetir la ejecución de la construcción ávida aleatoria una cantidad fija preestablecida de pasos, especificada por el parámetro *MaxIter*. En cada una de las ejecuciones de la construcción se aplica posteriormente una búsqueda local. Esta búsqueda local es una intensificación sobre la solución encontrada. Al aumentar la cantidad de iteraciones (*MaxIter*) en GRASP, la calidad de la solución final obtenida tiene más chances de mejorar, dado que hay una mayor diversificación en la búsqueda.

Capítulo 4

Modelo de optimización para el problema del diseño de redes de ciclovías

En este capítulo se presenta el modelo de optimización propuesto para el problema del diseño de redes de ciclovías.

Diseñar la red implica seleccionar en cuáles calles se construirán ciclovías, considerando los costos asociados, y teniendo en cuenta que se tiene un presupuesto límite para la construcción de las mismas. Además se debe tener en cuenta que hay cierta demanda de viajes que debe satisfacerse. Estos viajes o pares origen-destino sobre la red tienen una cantidad a ser transportada desde su origen al destino.

La formulación del modelo se basa en un modelo general para problemas sobre redes de transporte con múltiples mercancías [23]. En este proyecto dicho modelo es adaptado mediante el agregado de parámetros y restricciones para llevarlo al formato del problema en cuestión, particularmente penalizando la circulación de ciclistas por tramos sin ciclovía, de forma de contemplar la existencia de discontinuidades.

4.1 Correspondencia de elementos de la realidad a los elementos del modelo

La red de calles es representada con un grafo G , en donde cada arco (i,j) del mismo es un tramo de calles, y está ponderado. Por un lado, se tiene una ponderación con respecto al usuario, que puede ser por ejemplo el largo del tramo o el tiempo de circular por el mismo. Por otro lado, se tiene una ponderación que representa el costo de construcción, por ejemplo utilizando el largo de los tramos.

Sobre esta red se define la matriz origen-destino, que son los pares origen-destino k sobre la red, esto es, los diferentes viajes que los ciclistas desean realizar sobre la red (indicando origen y destino), y que se desean satisfacer a la hora de construir la red de ciclovías. En la figura 4.1 se puede ver un ejemplo de correspondencia de una red de calles a un grafo, donde se aprecia cómo las calles son mapeadas a arcos del grafo y los cruces a nodos.

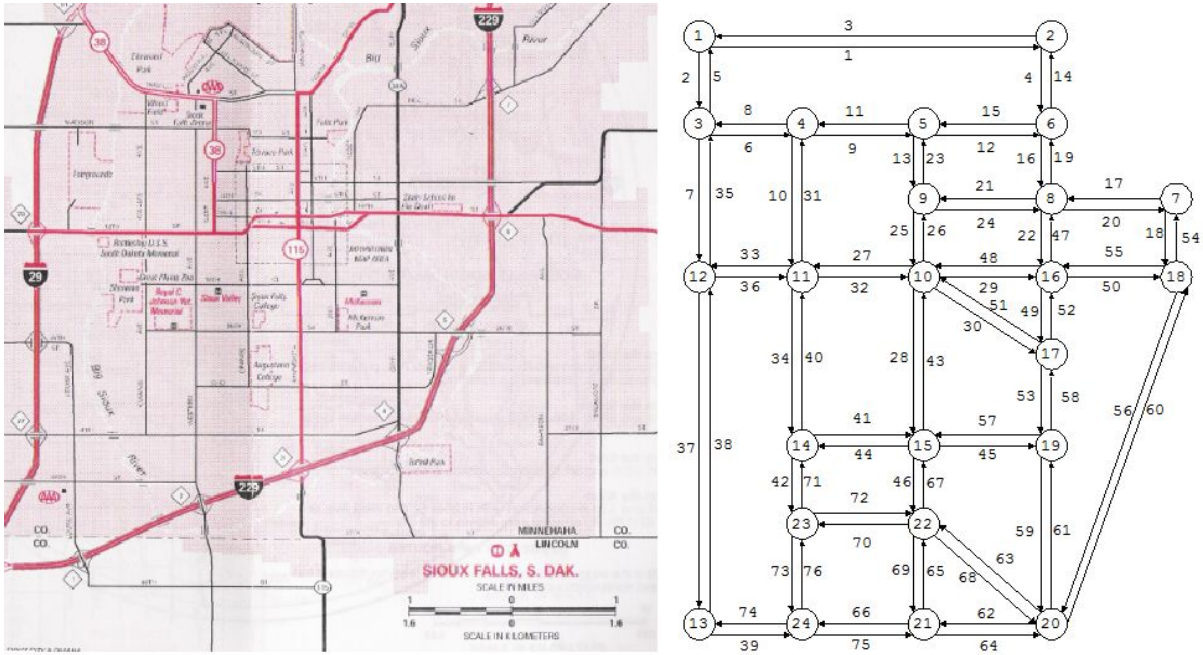


Figura 4.1 - Ciudad Sioux Falls. El grafo puede no ser totalmente representativo de la ciudad real.

Fuente: [38]

A continuación se define notación utilizada en la formulación del modelo referente a los datos del problema:

- $G = (N, A)$ es la red sobre la cual se definen los pares origen-destino. N es el conjunto de nodos ($|N| = n$) y A es el conjunto de arcos.
- l_{ij} es el largo del arco (i, j) .
- c_{ij} es el costo para los usuarios de circular por el arco (i, j) . Este costo podría ser por ejemplo tiempo de viaje en bicicleta.
- c'_{ij} es el costo de viajar por el arco (i, j) sin que haya ciclovía construida. Generalmente se cumple que $c'_{ij} \geq c_{ij}$.
- K es el conjunto de pares origen-destino a realizarse sobre la red (se definen en la matriz origen-destino).
- R_k es la cantidad de flujo (viajes en bicicleta por unidad de tiempo) requerida a transportar desde el origen denotado $O(k) \in N$, hacia el destino denotado $D(k) \in N$, del par origen-destino k perteneciente a K en un determinado período de tiempo.

4.2 Formulación de programación matemática

La formulación incluye variables que representan las decisiones del planificador (y_{ij}) que determinan los costos y permiten verificar que se cumpla con el presupuesto dado, y también variables que representan las decisiones de los usuarios ($f_{ij}^k, f_{ij}'^k$), que es para quienes se construye la infraestructura. Por esto, es de interés que el modelo pueda representar cuánto le cuesta a los usuarios viajar por donde ellos quieren viajar, una vez que el planificador decide por dónde construir ciclovía.

A continuación se define notación utilizada en la formulación del modelo referente a las variables de decisión del problema:

- y_{ij} es una variable binaria que indica si el arco (i,j) es elegido ($y_{ij} = 1$) o no ($y_{ij} = 0$) como parte de la red solución, es decir, si se construye ciclovía sobre el mismo.
- x_{ij} es una variable binaria que indica si por el arco (i,j) hay flujo ($x_{ij} = 1$) a pesar de no haber ciclovía. Con esta variable se logran representar aquellos tramos por los cuales los ciclistas circulan a pesar de no haber ciclovía construida. El costo por circular por estos tramos estará penalizado (c_{ij}'), de este modo se intentará minimizar la cantidad de estos tramos sobre los caminos seguidos por los ciclistas.
- f_{ij}^k es el flujo de ciclistas correspondientes al elemento k de la matriz de demandas origen-destino, sobre el arco (i,j) donde hay ciclovía construida.
- $f_{ij}'^k$ es el flujo del par origen-destino k sobre el arco (i,j) donde no hay ciclovía construida ($y_{ij} = 0 ; x_{ij} = 1$).

Las constantes del problema son:

- $l_{m\acute{a}x}$ es el largo máximo a construir de ciclovía (presupuesto).

La formulación de programación matemática para el problema es la siguiente:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} (c_{ij} f_{ij}^k + c_{ij}' f_{ij}'^k) \quad (2.1)$$

sa:

$$y_{ij} + x_{ij} \leq 1 \quad \forall (i,j) \in A \quad (2.2)$$

$$\sum_{k \in K} f_{ij}^k \leq y_{ij} \sum_{k \in K} R_k \quad \forall (i,j) \in A \quad (2.3)$$

$$\sum_{k \in K} f_{ij}'^k \leq x_{ij} \sum_{k \in K} R_k \quad \forall (i,j) \in A \quad (2.4)$$

$$\sum_{j \in N} (f_{ij}^k + f_{ij}'^k) - \sum_{l \in N} (f_{li}^k + f_{li}'^k) = \begin{cases} R_k & \text{si } i = O(k) \\ -R(k) & \text{si } i = D(k) \quad \forall k \in K \\ 0 & \text{en otro caso} \end{cases} \quad (2.5)$$

$$\sum_{k \in K} l_{ij} y_{ij} \leq l_{\max} \quad \forall (i,j) \in A \quad (2.6)$$

$$f_{ij}^k \geq 0, f_{ij}'^k \geq 0 \quad \forall (i,j) \in A, k \in K \quad (2.7)$$

$$x_{ij} \in \{0,1\}, y_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (2.8)$$

Función objetivo

En la función objetivo se busca minimizar los costos totales de los pares origen-destino en bicicleta. Aquí se penaliza aquellas soluciones en las que existe flujo por arcos donde no se construye ciclovía mediante el término: $c_{ij}' f_{ij}'^k$.

Notar que con esta función objetivo, en la solución óptima pueden aparecer arcos donde las variables x_{ij} o y_{ij} valgan 1, a pesar de que no haya flujo por los mismos. Esto se debe a que en la función objetivo solo se minimizan los costos de los usuarios.

Restricciones

La restricción 2.2 indica que el arco (i,j) está habilitado para circulación de ciclistas si tiene ciclovía construida ($y_{ij} = 1$; $x_{ij} = 0$) o si no tiene pero los usuarios lo utilizan para sus viajes de todos modos ($x_{ij} = 1$; $y_{ij} = 0$).

Las restricciones 2.3 y 2.4 fuerzan a que el total de flujo para todos los viajes sobre un arco (i,j) no exceda la sumatoria total de los flujos que se desean transportar por la red. De esta forma se asegura que haya flujo solo por tramos donde hay ciclovía construida, o donde los usuarios decidieron circular aunque no la haya. En el modelo los arcos no tienen una capacidad, por ello lo que se define en esta restricción es una cota superior.

La restricción 2.5 presenta las usuales ecuaciones para conservación de flujo en la red. Esto significa que, para todo nodo, el flujo entrante del par origen-destino k en el nodo es igual al flujo saliente de k en el mismo. Excepto el origen y el destino de k , en donde en el primero todo el flujo (R_k) es saliente y en el segundo todo el flujo es entrante.

La restricción 2.6 impone el límite con respecto al presupuesto. El total de ciclovía que se decida construir no puede exceder el largo máximo de ciclovía para el cual hay presupuesto.

El modelo presentado hasta ahora permite que los arcos sean dirigidos, lo que implica que la ciclovía no sea necesariamente bidireccional. Si se agrega la siguiente restricción:

$$y_{ij} = y_{ji} \quad \forall (i,j) \in A ; \quad x_{ij} = x_{ji} \quad \forall (i,j) \in A$$

entonces siempre que se construya ciclovía en un sentido, también se construirá en el sentido opuesto (variable y). Análogo es el razonamiento para los arcos por los cuales, a pesar de no haber ciclovía, los ciclistas circulan de todos modos (variable x). El hecho de forzar la bidireccionalidad implica que el valor objetivo pueda ser mayor ya que el presupuesto que sin esta restricción se podría dedicar a mejores caminos, bajo estas condiciones puede ser insuficiente. Sin embargo, con esto se cumple con los criterios mencionados en la sección 2.3.

En la Figura 4.1 se ilustra un ejemplo de solución obtenida al aplicar el modelo descrito por la formulación (2.1)-(2.8). En la solución, los tramos de calles por los cuales hay flujo de ciclistas pero no hay ciclovía construida se marcan en rojo, mientras que los tramos por los cuales hay flujo de ciclistas que utilizan ciclovía son marcados en azul. En el ejemplo se distinguen cuatro viajes con sus respectivos orígenes (O) y destinos (D). El flujo de ciclistas que componen los viajes marcados en verde y naranja se trasladan desde su origen a su destino utilizando la ciclovía construida; para todos los tramos (i,j) que componen los caminos determinados para satisfacer dichos viajes se cumple que $y_{ij} = 1$. Para el viaje marcado en violeta se puede observar que no es posible satisfacer su flujo utilizando las ciclovías construidas, por lo cual los tramos (i,j) que componen el camino, desde su origen a su destino, cumplen que $x_{ij} = 1$. En el viaje marcado en marrón se puede observar que el flujo de ciclistas sigue un camino discontinuo, en donde hay tramos con y sin ciclovía.

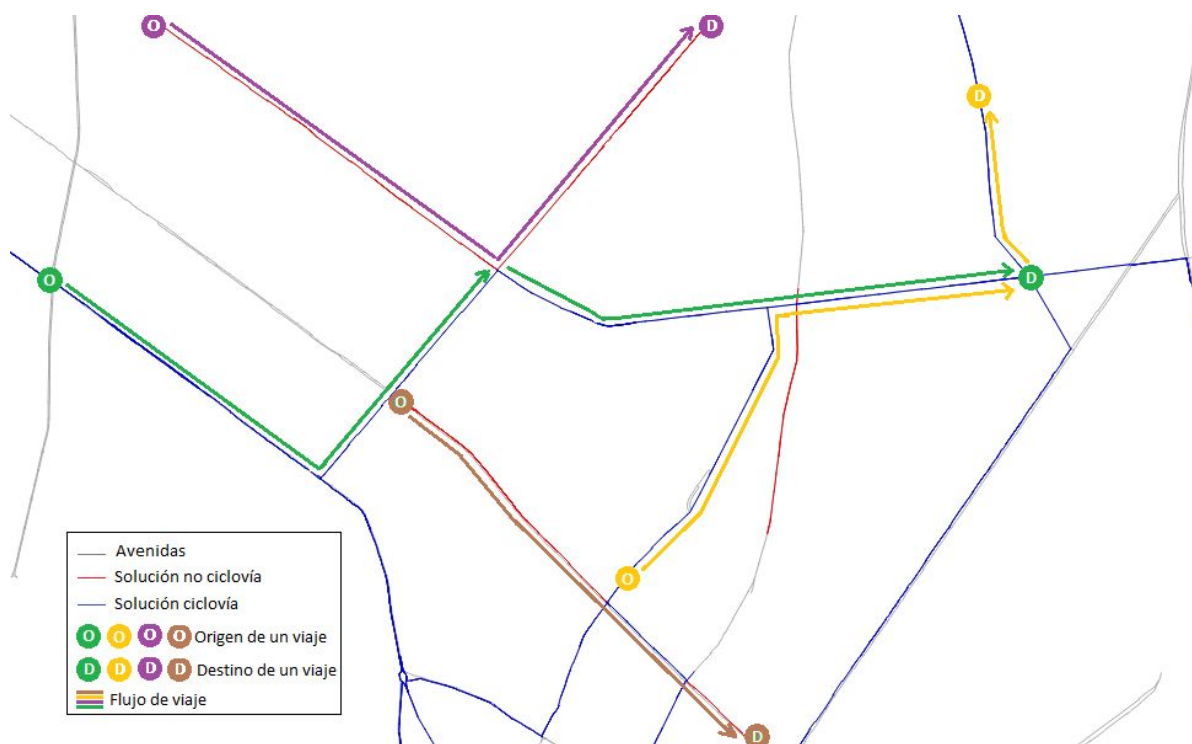


Figura 4.1 Ejemplo de un posible resultado del modelo propuesto

El modelo presentado es un modelo de programación lineal entera mixta, por lo tanto existen métodos de resolución implementados en paquetes de software, que permiten resolver el problema para instancias de tamaño moderado.

Capítulo 5

Metaheurística GRASP para el diseño óptimo de redes de ciclovías

El modelo de optimización presentado en el capítulo 4 resuelve de forma exacta el problema que se propone, sin embargo, luego de realizadas algunas pruebas computacionales se pudo verificar que para instancias del problema de tamaño significativo (en términos de cantidad de nodos, arcos y pares origen-destino), no es posible encontrar la solución óptima en un tiempo razonable y con los recursos computacionales disponibles. Para el caso de estudio que se seleccionó (Montevideo), no se pudo encontrar una solución con los recursos computacionales disponibles, en particular la memoria. Esto es consistente con los resultados conocidos acerca de la complejidad computacional de este tipo de problemas de optimización en redes [23]. Por este motivo se decide proponer e implementar una metaheurística con el objetivo de encontrar una solución aproximada.

El algoritmo propuesto resuelve de forma aproximada el modelo planteado en el capítulo 4. Para el diseño del mismo no solo se tuvo en cuenta el modelo de optimización original, sino que se agregó un elemento adicional que aporta realismo. El mismo consiste en que a la hora de evaluar las soluciones, se considera una mejor solución a aquella en la que la ciclovía propuesta cuenta con menor cantidad de discontinuidades en los caminos. Esto es difícil de formular matemáticamente, por esta razón es incluido solamente en la metaheurística y no en el modelo matemático.

En este capítulo se presenta dicho algoritmo, que se basa en la metaheurística GRASP (Greedy Randomized Adaptive Search Procedures).

5.1 Notación

Gran parte de la notación que se utiliza es igual a la utilizada en el capítulo 4. A continuación se presenta notación adicional.

- P conjunto de todos los p_k , donde p_k es el camino de menor costo para el par origen-destino k , es decir el camino de menor costo que va del nodo $O(k)$ al nodo $D(k)$
- $\{c_{ij}\}$ conjunto de todos los c_{ij} .
- $k_g = \{k_i \in K / R_k^i \geq \overline{R_k}\}$. Es el conjunto de elementos de la matriz origen-destino que cumplen que la cantidad requerida a transportar es mayor o igual al promedio.
- $k_c = K \setminus k_g$. Es el conjunto de elementos de la matriz origen-destino que cumplen que la cantidad requerida a transportar es menor al promedio.
- $\hat{p} = (p_k \in P / \sum_{ij} f'_{ij}{}^k \text{ es el máximo, } \forall (i,j) \in p_k)$. Dada una solución, \hat{p} es aquel camino por donde hay mayor cantidad de flujo de ciclistas por tramos en donde no hay ciclovía.
- $\bar{p} = (p_k / \sum_{ij} f_{ij}{}^k \text{ es mínimo, } \forall (i,j) \in p_k)$. Dada una solución, \bar{p} es aquel camino por donde hay menor cantidad de flujo de ciclistas por tramos con ciclovía.

5.2 Construcción de soluciones

El procedimiento ávido aleatorio de construcción de soluciones propuesto, genera un conjunto de caminos $P = \{p_1, p_2, \dots, p_k\}$ para los diferentes pares origen-destino k sobre la red, de modo de conectar su origen $O(k)$ con su destino $D(k)$. En el camino puede haber tanto tramos (i,j) en donde se construye ciclovía ($y_{ij} = 1$), como tramos en los que a pesar de que no se construya, los ciclistas decidan circular de todos modos ($x_{ij} = 1$), con el objetivo de alcanzar el destino del par origen-destino.

En líneas generales, el algoritmo intenta encontrar el mejor camino para cada par origen-destino, y en caso de que haya presupuesto ($l_{m\acute{a}x}$), decide construir ciclovía por estos caminos. En el caso de que no haya presupuesto, el algoritmo decide que los ciclistas igualmente circulen por allí, pero penalizando este hecho.

Dado que el algoritmo va seleccionando de forma aleatoria un par origen-destino k e intenta construir el camino de menor costo para este, es natural que en determinado momento no haya más presupuesto disponible, en cuyo caso el algoritmo comienza a asignar arcos por los cuales a pesar de no haber ciclovía, los ciclistas decidan circular de todos modos. Por estos tramos el costo está penalizado

(c'_{ij}). Para minimizar la cantidad de este tipo de tramos, e intentar mejorar la solución, el algoritmo realiza las siguientes acciones:

- El conjunto de K se separa en dos conjuntos. Un primer conjunto con todos aquellos pares origen-destino en los cuales la cantidad requerida a transportar es mayor o igual a la cantidad requerida promedio de todos ellos (k_g). Y un segundo conjunto que contiene el resto (k_c). De este modo se le da prioridad a los pares origen-destino pertenecientes al primer conjunto, ya que serán los que presenten mayor flujo, por lo que esto hará que la penalización sea menor. En la sección 5.3 se profundiza en la explicación de cómo se evalúa una solución.
- La matriz de costos $\{c_{ij}\}$ se redefine en cada iteración teniendo en cuenta lo que se lleva construido hasta el momento. Este cambio en los costos se debe a que por la forma en que se realiza la construcción de la ciclovía en la solución, el algoritmo rápidamente se quedaría sin presupuesto (ya que va construyendo los caminos para cada par origen-destino). Al redefinir los costos, se asigna un menor costo (mitad del costo original) a aquellos arcos en los que el algoritmo decidió construir ciclovía. De este modo, a la hora de calcular el camino de menor costo, se tenderá a utilizar la ciclovía ya construida (utilizando menos presupuesto) tratando de tener la menor cantidad de discontinuidades posibles, por lo tanto menor penalización.

En la Figura 5.1 se presenta un pseudocódigo de la construcción ávida aleatoria.

procedure ConstrucccionInicial

```
01  $K \leftarrow$  conjunto de pares origen-destino
02  $l_c \leftarrow 0$  ; // Largo construido hasta el momento para la ciclovía.
03  $k_g \leftarrow$  Conjunto de todos lo pares origen-destino en donde la cantidad requerida a
transportar es mayor o igual a la cantidad promedio requerida para un viaje.
04  $k_c \leftarrow$  Conjunto de pares origen-destino con cantidad requerida menor que la promedio.
05  $\{y_{ij}\} \leftarrow 0$ 
06  $\{x_{ij}\} \leftarrow 0$ 
07  $\{f_{ij}^k\} \leftarrow 0$ 
08
09 for  $k$  in  $k_g$ 
10   calcularCaminoViaje( $k$ )
11 end
12
13 for  $k$  in  $k_c$ 
14   calcularCaminoViaje( $k$ )
15 end
```

end ConstrucccionInicial

Figura 5.1 Construcción ávida aleatoria de soluciones.

procedure calcularCaminoViaje(k)

```
01  $o \leftarrow$  nodo origen de k.
02  $d \leftarrow$  nodo destino de k.
03 for  $c_{ij}$  in  $\{c_{ij}\}$  // Redefinir la matriz de costos teniendo en cuenta lo construido.
04   if  $y_{ij} == 1$  then
05      $\widehat{c}_{ij} = 0$  // Si hay ciclovía, se asigna costo 0 para que se utilice la misma.
06   else
07      $\widehat{c}_{ij} = c_{ij}$  // Si no hay ciclovía, se asigna el costo del arco (i,j).
08   end
09 end
10  $p_k = \text{caminoMenorCostoDijkstra}()$  // Se inserta en  $P$  .
11  $n \leftarrow$  Largo de  $p_k$ 
12 for  $i = 0$  to  $i = n - 1$  do //  $i$  es el i-ésimo nodo de  $p_k$ 
13    $j = i + 1$  //  $j$  es el siguiente nodo luego de  $i$  en  $p_k$ 
14   if  $l_c + l_{ij} \leq l_{max}$  or  $y_{ij} = 1$  then
15      $f_{ij}^k = R_k$  // Se asigna el flujo de ese par origen-destino en el arco (i,j)
16     if  $y_{ij} == 0$  then
17        $l_c += l_{ij}$  // Se construye ciclovía por el arco (i,j)
18        $y_{ij} = 1$ 
19     end
20   else
21      $f'_{ij} = R_k$ 
22      $x_{ij} = 1$ 
23   end
24 end
25 end

end calcularCaminoViaje(k)
```

Figura 5.2 Cálculo del camino a seleccionar para un par origen-destino k.

A continuación se presentan algunas aclaraciones sobre el algoritmo:

- Línea 09 y 13 (Figura 5.1): La selección de los k en los conjuntos se realizan de forma aleatoria con distribución de probabilidad uniforme.
- Línea 10 (Figura 5.2): Para buscar el camino de menor costo se utiliza el algoritmo de Yen. En este caso Yen retorna los m caminos de menor costo, y el algoritmo selecciona uno de forma aleatoria con distribución de probabilidad uniforme. De este modo se pueden buscar diferentes soluciones en las diferentes iteraciones. En un principio se utilizó el algoritmo de Dijkstra, pero se observó que los resultados eran mejores con el algoritmo de Yen. Esto se debe a que el algoritmo de Yen, además de retornar el camino de

menor costo, retorna otras alternativas, lo que aporta diversidad a las soluciones generadas.

- Línea 14 (Figura 5.2): En esta condición se evalúa si hay presupuesto disponible para construir el tramo o si ya hay ciclovía construida, ya que en ambos casos el flujo por ciclovía (f_{ij}^k) aumentará. Luego en la línea 18 se evalúa si estaba construida o no, si no, se construye.
- Línea 21 (Figura 5.2): En el caso que no haya presupuesto y el tramo no esté construido, entonces el flujo sería por un tramo en el cual a pesar de no haber ciclovía los ciclistas circulan de todos modos ($f_{ij}^{'k}$). Aquí es donde se asegura que y_{ij} y x_{ij} no valgan 1 al mismo tiempo.

5.3 Evaluación de soluciones

Dada una solución factible para el problema, la evaluación de la solución se hace siguiendo la función objetivo representada por la expresión (2.1) del capítulo 4. Se tiene en cuenta la ciclovía construida, los flujos de cada par origen-destino, los costos y penalizaciones.

Adicionalmente se implementó una variación a la metaheurística en la cual al momento de evaluar una solución se tiene en cuenta si la ciclovía que se construye cuenta con discontinuidades o no, agregando una penalización a aquellas soluciones en las que hay presentes discontinuidades. De este modo se intenta que la ciclovía sea lo más continua posible.

Para la implementación de esta disminución de discontinuidades se tienen en cuenta los saltos presentes en los caminos de cada par origen-destino (p_k). Se considera que en un camino hay un salto cuando los ciclistas que circulan por dicho camino, pasan desde un tramo en el que hay construida ciclovía a un tramo en el cual no la hay. De este modo se agrega una penalización por cada salto encontrado en un camino. Al momento de evaluar la solución se suma esta penalización, lo que permite dar prioridad a aquellas soluciones en las que hay menor cantidad de saltos

(por lo tanto la ciclovía es más continua para el ciclista). La figura 5.3 muestra un ejemplo en el cual se utiliza el algoritmo con dicho cambio:

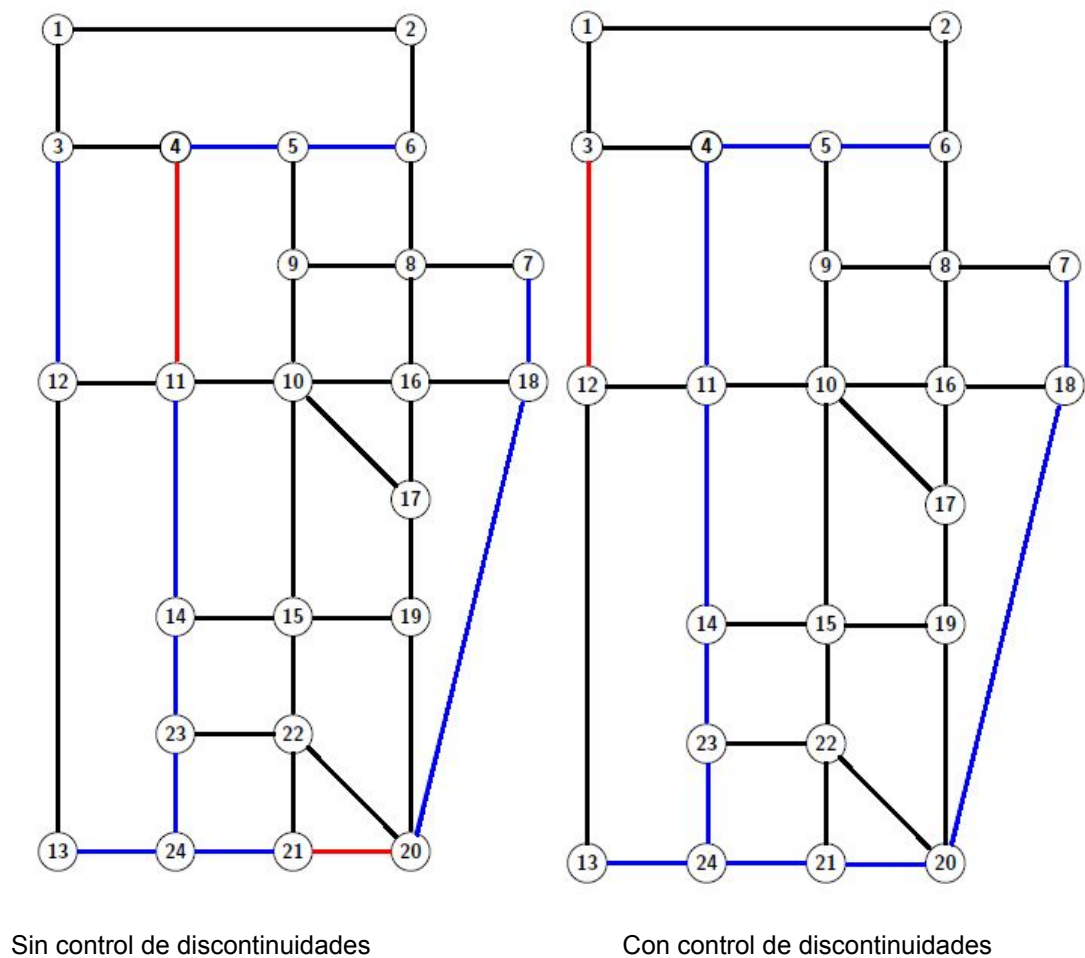


Figura 5.3 - Ejemplo de aplicación de disminución de discontinuidades

En este ejemplo existen 7 pares origen-destino con las siguientes propiedades:

Id par OD	Flujo requerido	Origen	Destino
1	200	13	11
2	200	4	6
3	200	13	21
4	200	20	7
5	199	12	3
6	1	14	5
7	1	24	18

Al construir la solución sin control de discontinuidades los pares origen-destino 6 y 7, que son los que tienen menor flujo, se ven afectados por discontinuidades. El 6 se ve afectado ya que el camino que recorre es 14, 11, 4 y 5 y el arco que va del nodo 4 al 5 no tiene ciclovía construida. El par origen-destino 7 sigue el camino 24, 21, 20 y 18 y el arco que va del nodo 21 al 20 no tiene ciclovía construida por lo cual también se ve afectado. Al aplicarle una gran penalización a las discontinuidades, la solución tiene esos caminos completamente contruidos, pero en este caso el presupuesto no alcanza para construir el arco que va del nodo 12 al 3, afectando negativamente el par origen-destino 5 que tiene un flujo significativamente mayor al de 6 y 7. Como resultado no se presentan discontinuidades en los caminos de un mismo viaje pero el flujo que viaja por ciclovía cae considerablemente debido al par origen-destino 5.

Si bien este cambio tiene en cuenta las discontinuidades, esto no quiere decir que la solución sea mejor respecto a los costos para el ciclista que diferencia entre costos de viaje por ciclovía y por calle sin ciclovía. Construir ciclovía en ciertos tramos para lograr mayor continuidad puede implicar aumentar el flujo por tramos sin ciclovía.

5.4 Mejora de soluciones

En el algoritmo propuesto las mejoras de soluciones se implementan con una búsqueda local.

El procedimiento de búsqueda local propuesto parte de una solución en la cual la ciclovía está construida, y cada par origen-destino tiene su camino p_k definido. En líneas generales, se buscan ciertos tramos que puedan ser quitados de la ciclovía para así tener presupuesto disponible y construir ciclovía en un tramo diferente. De este modo se puede evaluar la nueva solución y decidir si es mejor que la que se tenía.

El algoritmo selecciona dos caminos, uno para ser mejorado y el otro para ser quitado de la solución. El primero es aquel camino que cuenta con mayor flujo total de ciclistas que circulan por tramos aunque no haya ciclovía, esto es p_k donde $\max \sum_{ij} f'_{ij}{}^k$ para todo arco (i,j) en p_k . El segundo es aquel camino que cuenta con menor flujo de ciclistas que circulan por ciclovía, esto es $\min \sum_{ij} f_{ij}{}^k$ para todo arco (i,j) en p_k . Con esto, el algoritmo intenta mejorar el camino en el cual hay gran flujo pero no hay ciclovía construida, y quitar aquel en el cual hay poco flujo de ciclistas en ciclovía construida.

Este procedimiento de búsqueda local se realiza hasta que no se encuentren más mejoras, o sea, hasta que no se logre mejorar la solución actual. En la figura 5.3 se muestra un pseudocódigo de la búsqueda local.

procedure BusquedaLocal

```
01  $\hat{p} \leftarrow$  Camino a mejorar.
02  $\bar{p} \leftarrow$  Camino a quitar.
03  $n_1 \leftarrow$  Cantidad de nodos de  $\hat{p}$ .
04  $n_2 \leftarrow$  Cantidad de nodos de  $\bar{p}$ .
05 for  $i = 0$  to  $i = n_1 - 1$  do //  $i$  es el  $i$ -ésimo nodo de  $\hat{p}$ 
06    $j = i + 1$  //  $j$  es el siguiente nodo luego de  $i$  en  $\hat{p}$ 
07   if  $y_{ij} == 1$  then
08      $l_c -= l_{ij}$ 
09   end
10    $y_{ij} = 0$ ;  $x_{ij} = 0$ 
11 end
12 for  $i = 0$  to  $i = n_2 - 1$  do //  $i$  es el  $i$ -ésimo nodo de  $\bar{p}$ 
13    $j = i + 1$  //  $j$  es el siguiente nodo luego de  $i$  en  $\bar{p}$ 
14   if  $y_{ij} == 1$  then
15      $l_c -= l_{ij}$ 
16   end
17    $y_{ij} = 0$ ;  $x_{ij} = 0$ 
18 end
19 for  $k$  in  $K$ 
20   calcularCaminoViaje( $k$ )
21 end
22
```

end BusquedaLocal

Figura 5.3 - Búsqueda local.

A continuación se presentan algunas aclaraciones sobre el algoritmo:

- Líneas 5 y 12 : Tanto para el camino a quitar como para el camino a mejorar, se quita todo lo que se haya construido allí, ya que son los caminos que se decidieron cambiar. Al quitar estos tramos, quedará presupuesto disponible para construir otros tramos.
- Línea 20 : La selección de los k en el conjunto de todos los pares origen-destino K se hace dando prioridad a aquellos viajes que tenían flujo por algún tramo del camino \hat{p} . Dado que \hat{p} es el camino por el que mayor cantidad de ciclistas circulan por tramos que no hay ciclovía, es de interés tratar de construir ciclovía por esos tramos. Por lo tanto, dándole prioridad a estos viajes, por esos tramos será más probable que se construya ciclovía cuando se ejecute *calcularCaminoViaje*.

Capítulo 6

Implementación

En este capítulo se presentan aspectos importantes de la implementación del sistema, lo que incluye la construcción de la red, el procesamiento de la matriz de viajes y la metaheurística.

6.1 Procesamiento de datos de entrada

Se implementaron dos procesadores de archivos, uno de archivos de texto (.txt) y otro de archivos shapefile (.shp), mediante los cuales se construye la red del caso a estudiar, se almacenan los datos sobre los pares origen-destino, y los principales datos del problema.

Un archivo shapefile es un formato de almacenamiento de datos vectoriales creado por la empresa Esri para almacenar la ubicación, la forma y los atributos de las entidades geográficas [40].

A continuación se explica brevemente el formato que deben seguir los archivos de texto que son utilizados como entrada de la metaheurística implementada:

01	Nodos	n	\rightarrow	Cantidad de nodos de la red.
02	Aristas	a	\rightarrow	Cantidad de aristas de la red.
03			
04	i	c_i	c'_i	l_i o_i d_i
05			
06	Viajes	m	\rightarrow	Cantidad de pares origen-destino sobre la red.
07			
08	k	R_k	o_k	d_k
09			
10	Parámetros			
11	MaxLargo	l_{max}	\rightarrow	Máximo presupuesto disponible.

Donde:

- Línea 4: Representa un arco de la red. Donde i es el identificador y luego se indican costo, penalización, largo, nodo origen y nodo destino de la arista.
- Línea 8: Representa un par origen-destino sobre la red. Donde k es el identificador y luego se indican cantidad requerida de ciclistas a trasladarse, origen y destino.

Este archivo es procesado y utilizado para cargar los datos correspondientes a la red, los pares origen-destino sobre la misma y el presupuesto disponible.

Para realizar algunas de las pruebas que se presentan en el capítulo 8, se utilizaron archivos de datos de algunas ciudades del mundo, para las cuales se dispone de información sobre sus respectivas redes de transporte para propósitos académicos [38]. En particular se utiliza el caso de Anaheim y Sioux Falls, ciudades de EEUU. Los archivos descargados se procesan para cargar tanto la red como la matriz origen-destino del problema.

Como entrada del algoritmo se pueden utilizar también archivos de formato shapefile (.shp). En este caso la información geográfica contenida en el archivo es procesada mediante una base de datos como se explica en el capítulo 7.

6.2 Método de optimización

El método exacto es implementado mediante la codificación de la formulación (2.1)-(2.8) en el lenguaje AMPL y resuelto mediante CPLEX [39]. El mismo requiere dos archivos, un archivo *.mod* que contiene el modelo matemático y un archivo *.dat* que contiene los datos de prueba.

El método aproximado está implementado en el lenguaje Java en su versión 1.8. Para la implementación del algoritmo de Yen se utilizó una implementación de código abierto [41], la cual fue modificada para adaptarla a las otras estructuras del algoritmo.

6.3 Procesamiento de datos de salida y visualización

Los procesadores de archivos implementados permiten exportar la información del modelo, que contiene los datos de la red, los pares origen-destino sobre la misma y el presupuesto, a un archivo .dat. Este es utilizado conjuntamente con el archivo .mod como entrada del método exacto para comparar los resultados de este y el método aproximado. Si los datos procesados como entrada del algoritmo fueron tomados de un archivo shapefile, también es posible exportar el resultado a un archivo del mismo tipo para visualizar la solución de forma geográfica sobre el mapa.

En la figura 6.1 se presenta un diagrama de alto nivel de la implementación, que ilustra la importación y exportación de archivos en el algoritmo, y la utilización de estos como entrada tanto de la metaheurística como del solver CPLEX, y sus correspondientes exportaciones.

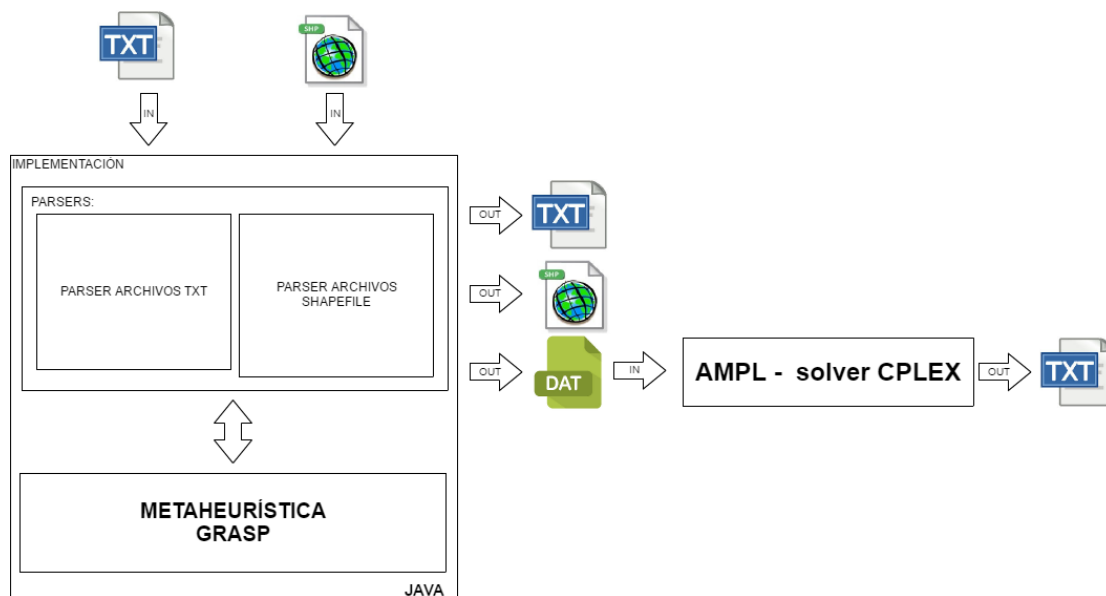


Figura 6.1: Diagrama del flujo de datos

Capítulo 7

Caso de Estudio: Montevideo

En este capítulo se presenta cómo se construye el caso de estudio. Para este se elige la ciudad de Montevideo, capital de Uruguay, con el objetivo de validar la propuesta para la solución del problema en un caso real.

En el capítulo 8 se detallan las pruebas realizadas para este caso de estudio y se analizan los resultados obtenidos tras aplicar el método de optimización implementado.

7.1 Situación actual

Según cifras brindadas por la Dirección de Movilidad Urbana de la Intendencia de Montevideo [42], hasta el mes de agosto del año 2014 se había construido 3800 metros de ciclovías que componen el Bicicircuito de Ciudad Vieja y 650 metros para el Bicicircuito estudiantil que conecta las zonas de las facultades de Ciencias Sociales, Derecho, Economía, Arquitectura e Ingeniería de la Universidad de la República. Además se construyó 11 km de bisisendas entre la Rambla de Punta Carretas, Bulevar Artigas, Av. Belloni y el Parque lineal del arroyo Miguelete. En setiembre del mismo año la Intendencia informaba la futura construcción de seis nuevas ciclovías en distintas zonas de la ciudad de Montevideo. También se señalaron muchos kilómetros de calles como "calles a 30" (tramos donde la velocidad máxima autorizada para vehículos motorizados son 30 km/h) para brindar mayor seguridad al ciclista cuando debe tomar dichas calles tras el fin de una ciclovía.

En un artículo publicado por el sitio web Cien8chenta [42] Javier Taks, del programa Unibici (Bicicletas y ciclistas en la Universidad de la República), señaló que "hay una voluntad de tener una transformación" en la ciudad. La Intendencia de Montevideo sigue apostando a la promoción del uso de la bicicleta para mejorar la salud y el bienestar de los ciudadanos además de promover con esto el cuidado del medio ambiente. También en el año 2014 la Intendencia inauguró un sistema de bicicletas públicas en Ciudad Vieja, el cual propone el préstamo de bicicletas tanto

para ciudadanos montevideanos como a turistas con el objetivo de disfrutar de este medio de transporte por ser ágil, económico, saludable y ecológico [43].

Si bien se ha evolucionado en la idea de promover el uso de la bicicleta como medio de transporte en la ciudad, todavía hay algunos aspectos importantes a mejorar en este ámbito. Uno de los más importantes es el respeto al ciclista por parte de los conductores de otros medios de transporte (autos, ómnibus, motos) e incluso por parte de los peatones. En un artículo publicado por el diario El País [44] se indica que si bien la Intendencia ha colocado carteles que indican la existencia de ciclovías y bisisendas, todavía no hay suficiente conocimiento por parte de los ciudadanos no ciclistas sobre qué significan dichas señalizaciones. Los peatones caminan por las ciclovías y los vehículos estacionan sobre ellas. También se ha señalizado con carteles la preferencia que se le debe dar al ciclista en ciertos cruces. Esto tampoco es respetado en la mayoría de los casos y no hay, por el momento, inspectores designados a sancionar y educar a los infractores. Otro problema expuesto en este artículo son las pendientes, varios tramos de ciclovía son construidos sobre pendientes, lo cual demanda un esfuerzo físico considerable al ser transitados. Además se presenta el problema de las discontinuidades en las ciclovías. La Intendencia ha construido, para el caso de bisisendas, cortes en los cordones para que los usuarios puedan bajar la bicicleta sin problemas si desean cruzar una calle, pero esto no soluciona el problema en general.

En un estudio publicado en 2012 [45] se señala que la estructura de la movilidad en Montevideo sigue los patrones regionales, con fuerte presencia del transporte público colectivo y de los modos activos (viajes a pie y en bicicleta). Del total de los viajes, el 36,8% corresponde a los modos activos. Si se discrimina por estrato de ingresos se observa un patrón relativamente similar al de otros países de la región. En el primer tercil el 18% corresponde a modos activos, en el segundo 11% y en el tercero 5%.

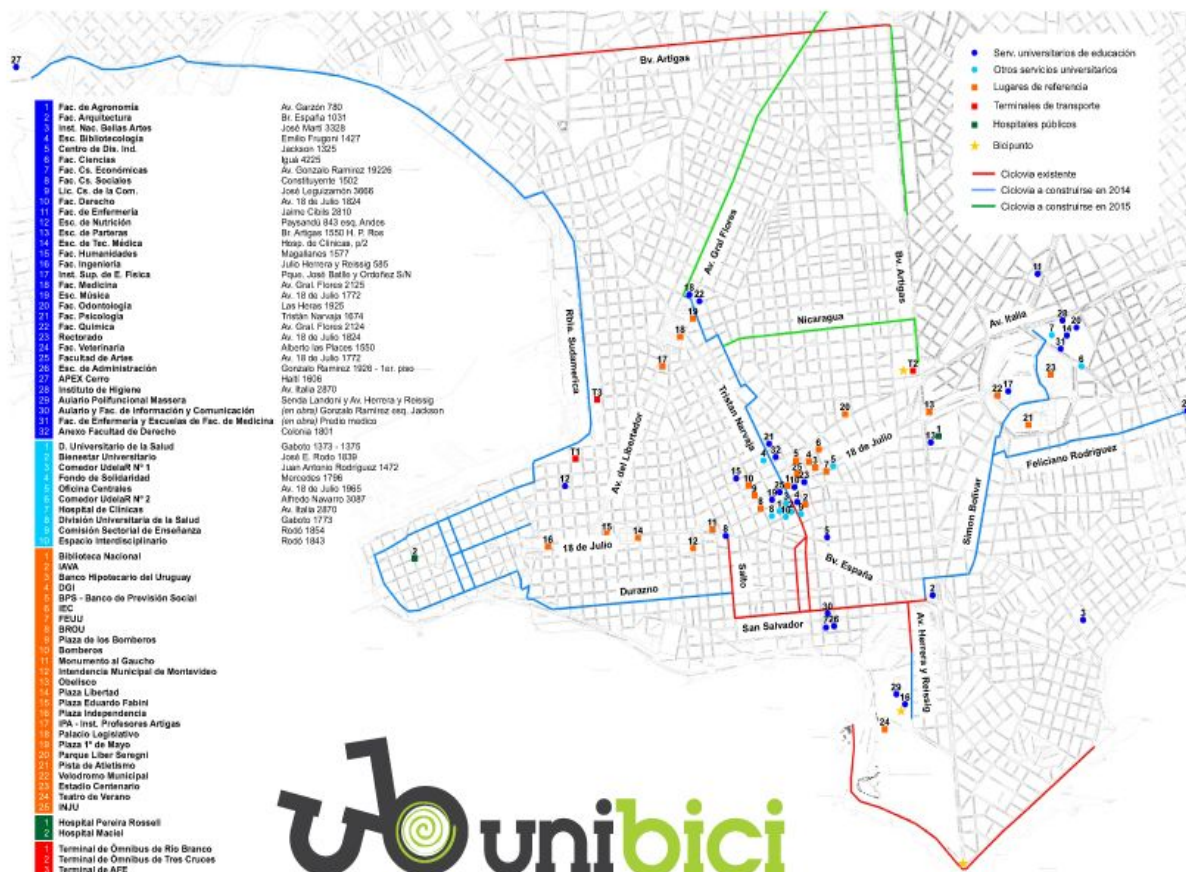


Figura 7.1: “La ciudad va cambiando de color” [46]

7. 2 Construcción del caso

Para la construcción del caso se utilizaron los datos abiertos brindados por la Intendencia de Montevideo (IM), más precisamente los referentes a las calles, avenidas y límites del departamento. Además, por medio del Arquitecto Juan Francisco Vespa y el Licenciado en Estadística Eduardo Coutiño, ambos miembros de la Unidad Ejecutiva del Plan de Movilidad Urbana de la IM, se obtuvieron los datos correspondientes a la Encuesta Domiciliaria de Movilidad realizada en 2009 la cual incluye el estudio de los viajes en bicicleta realizados por los ciudadanos. El objetivo de esta encuesta fue analizar la composición, motorización y disponibilidad del espacio de estacionamiento de los hogares, patrones de movilidad de las personas y los hogares, y los hábitos, opiniones y preferencias tanto de los usuarios como de los no usuarios del sistema de transporte colectivo. El área de estudio es el Área Metropolitana de Montevideo (AMMON) que incluye completamente al departamento de Montevideo y todas aquellas localidades que poseen al menos una

zona censal (aproximadamente una manzana) en un radio de 60 km del kilómetro cero de Montevideo en la plaza Cagancha. Junto con la encuesta se obtuvo la matriz origen-destino generada con los datos obtenidos en la encuesta.

Se obtuvo conjuntamente con la encuesta el shapefile correspondiente a las zonas de Transporte entre las cuales se realizaron dichos viajes. La estructura de zonas es realizada por la IM, respetando, en la medida de lo posible, microzonas que fueron definidas con anterioridad. Luego se ajustan dichas zonas agregando zonas censales del Instituto Nacional de Estadística (INE)[47], de manera de combinar información socioeconómica relevada en la Encuesta Continua de Hogares con información de movilidad.

7.2.1 Preparación de los datos

Herramienta utilizada

Se utilizó la herramienta PostGIS [48], módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL [49], convirtiéndola en una base de datos espacial para su utilización en un Sistema de Información Geográfica. Procesar los datos mediante una base de datos relacional permitió un manejo más sencillo de la información.

7.2.1.1 Zonificación de la región de estudio

La información de zonas abarca la región metropolitana incluyendo zonas de los departamentos de Canelones y San José.

Se procesó el shapefile para analizar solo las zonas de interés, es decir, las pertenecientes al departamento de Montevideo (Figura 7.2).

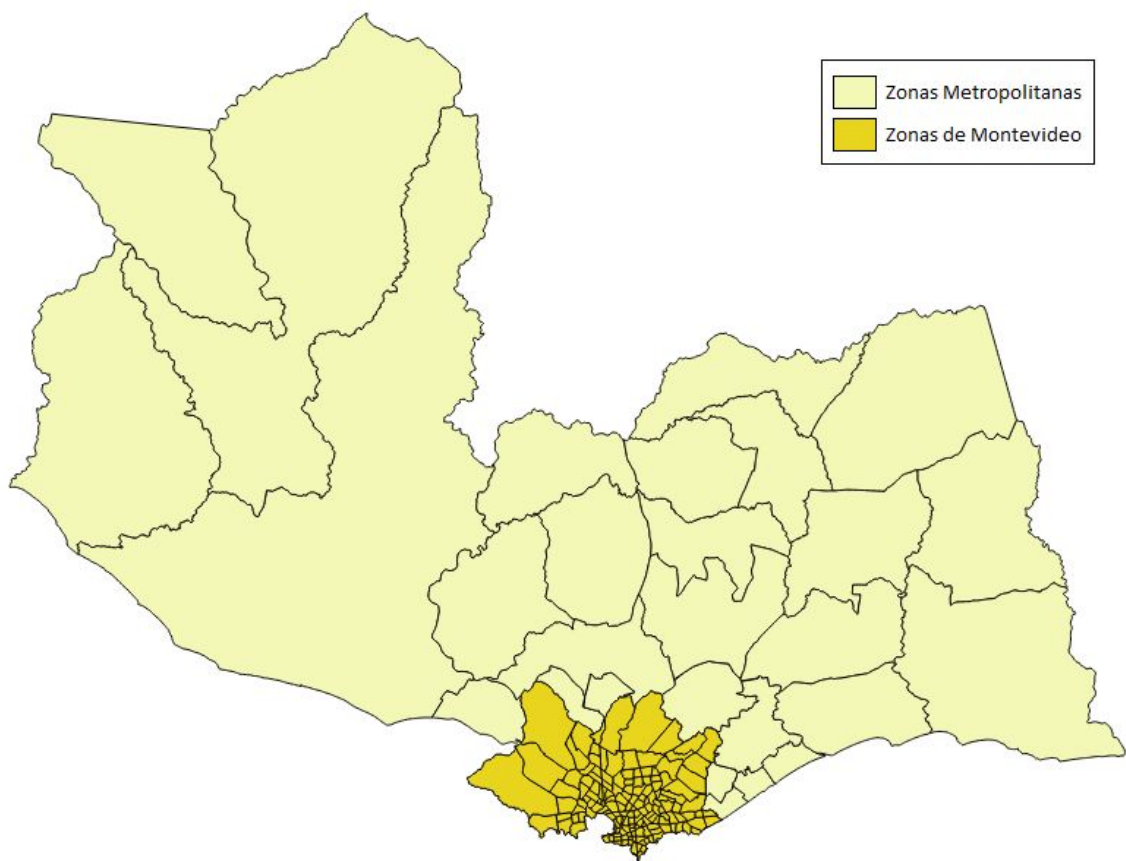


Figura 7.2: Zonificación Metropolitana

Los datos de las zonas de transporte se procesaron, utilizando las herramientas ya mencionadas, para filtrar la zonas correspondientes a Montevideo (Figura 7.3).

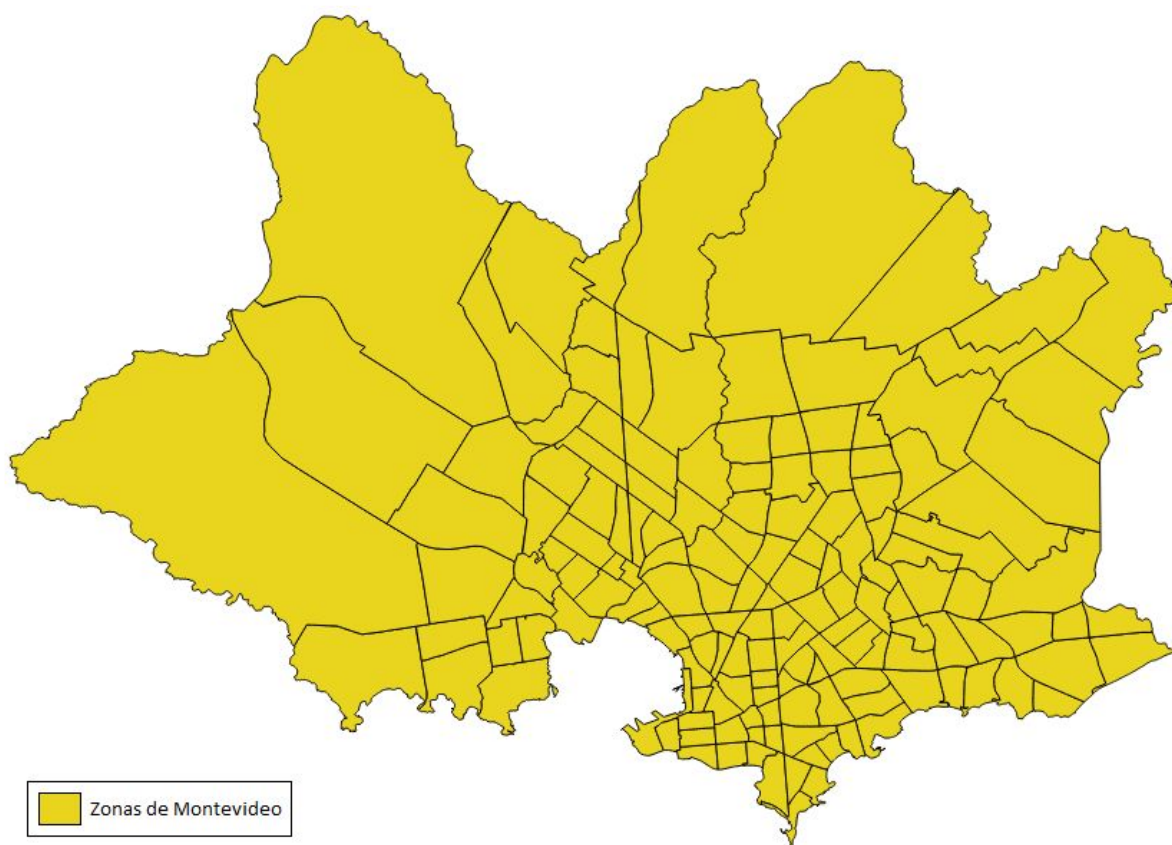


Figura 7.3: Zonificación de Montevideo

7.2.2.2 Construcción de la red

Para la construcción de la red se utilizaron los datos generados al procesar el shapefile de avenidas de Montevideo (Figura 7.4). La decisión de utilizar las avenidas en lugar de la red de calles en su totalidad es debido que no todas las calles son aptas para construir ciclovía, y el estudio de cuáles lo son excede el alcance de este proyecto. Una alternativa rápida de realizar la selección es tomar en cuenta sólo las avenidas ya que el tamaño de la red es significativamente mayor si se deciden utilizar todas las calles, provocando que los tiempos de ejecución de las pruebas aumenten considerablemente.



Figura 7.4: Avenidas de Montevideo

Se procesó cada tramo de las avenidas como un arco. El término tramo refiere a una cuadra de calle que une dos cruces (Figura 7.5). Cada tramo se procesa como un arco no dirigido de la red y cada extremo del tramo como un nodo de la misma. A cada tramo se le asocia un costo igual a su largo.

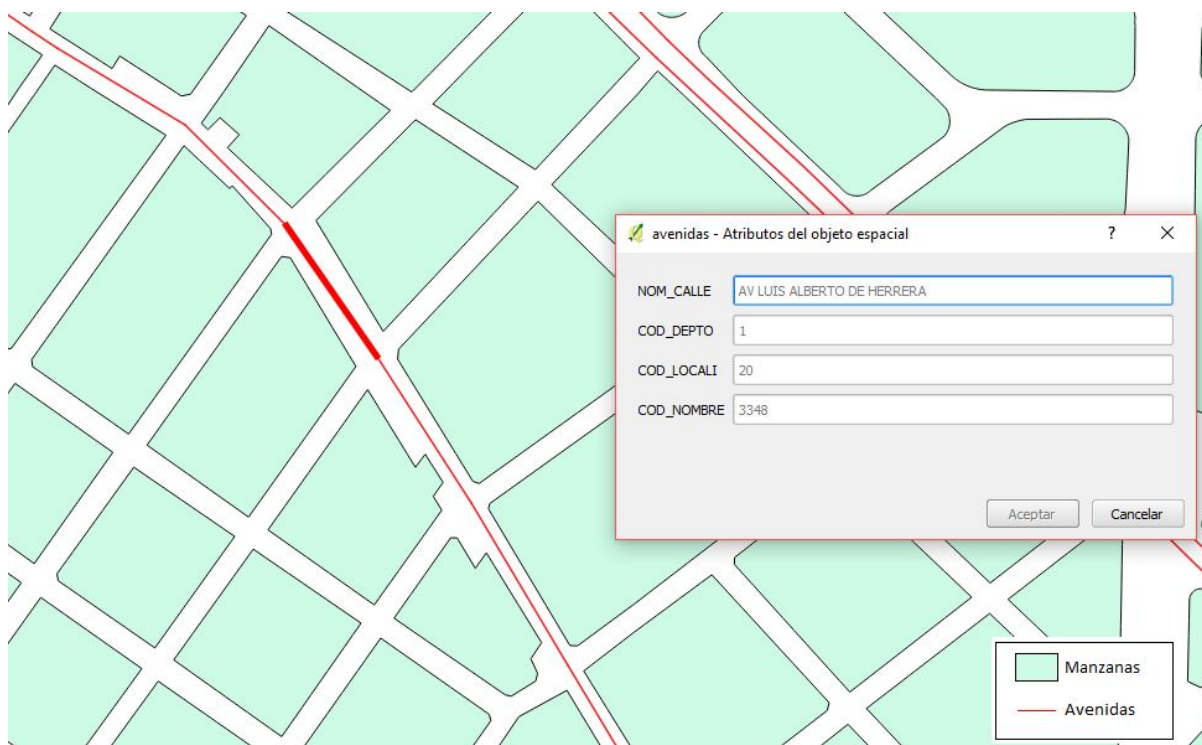


Figura 7.5: Tramos y manzanas

El procedimiento descrito resultó en una red de 12759 nodos y 26165 arcos (Figura 7.6).

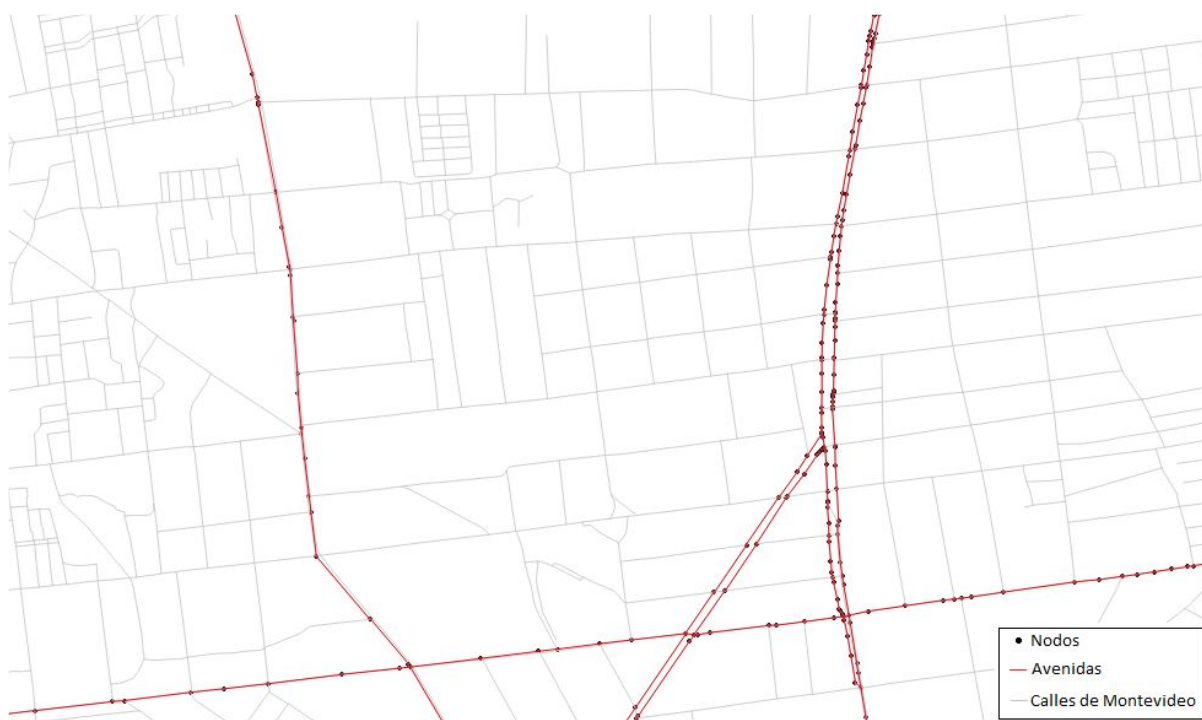


Figura 7.6: Parte de la red de Montevideo

Una característica importante y a tener en cuenta del modelo de red utilizado, es que los arcos de la red construida no son dirigidos. Estos arcos se construyen a partir de vías (calles) que en algunos casos tienen un único sentido en la realidad. La decisión de hacer arcos no dirigidos fue tomada debido a la naturaleza del modelo y del algoritmo propuesto, es decir, siempre se construye ciclovía doble vía aunque la calle sea flechada.

7.2.2.3 Construcción de la matriz origen-destino

Se procesó la matriz origen-destino proporcionada por la IM mencionada en la sección 7.2. La misma cuenta con los viajes en un día hábil.

Se consideraron los pares origen-destino entre zonas diferentes y no los que se encuentran dentro de una misma zona. Se observó que no todas las zonas registraban viajes entrantes o salientes de las mismas (Figura 7.7).

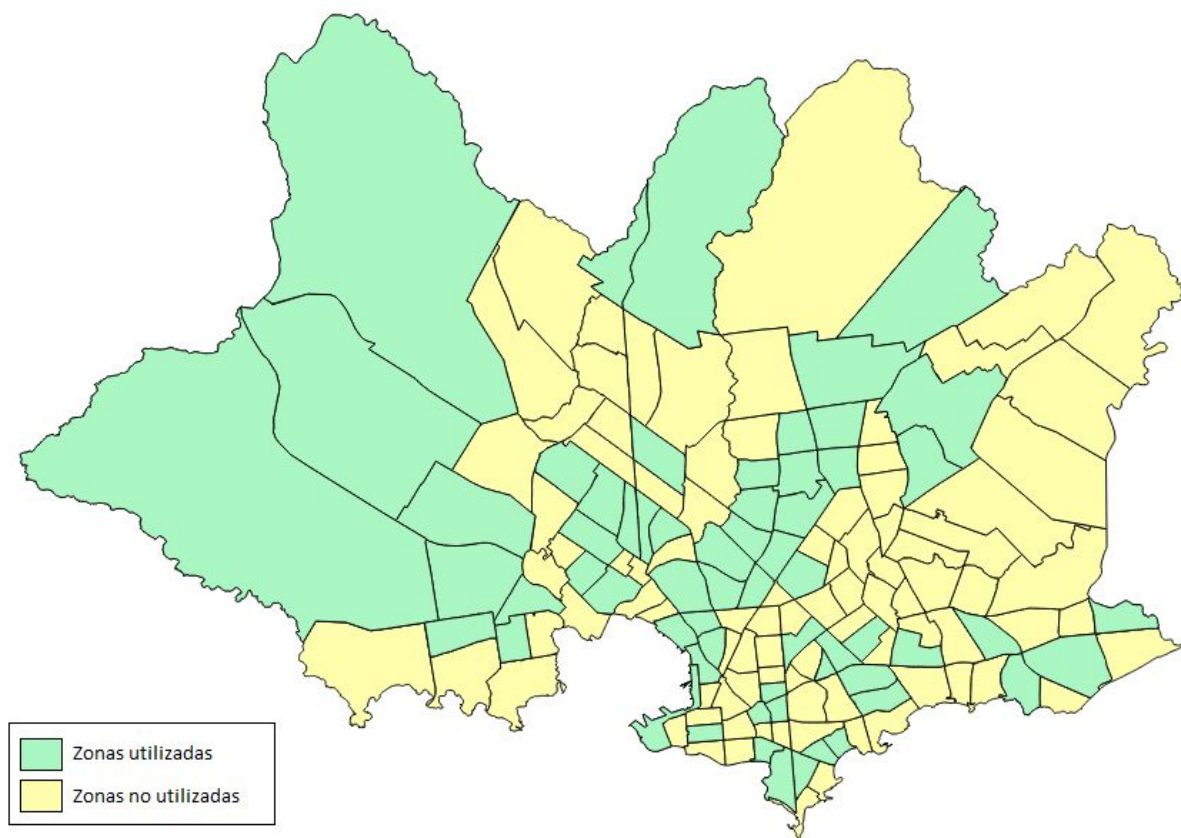


Figura 7.7: Zonas utilizadas (zonas las cuales son origen o destino de algún par origen-destino)

Se procesaron los pares origen-destino para así definir los viajes sobre la red.

Se verificó si todas las zonas eran accesibles por medio de las avenidas y se constató que sólo una zona no lo era (Figura 7.8).

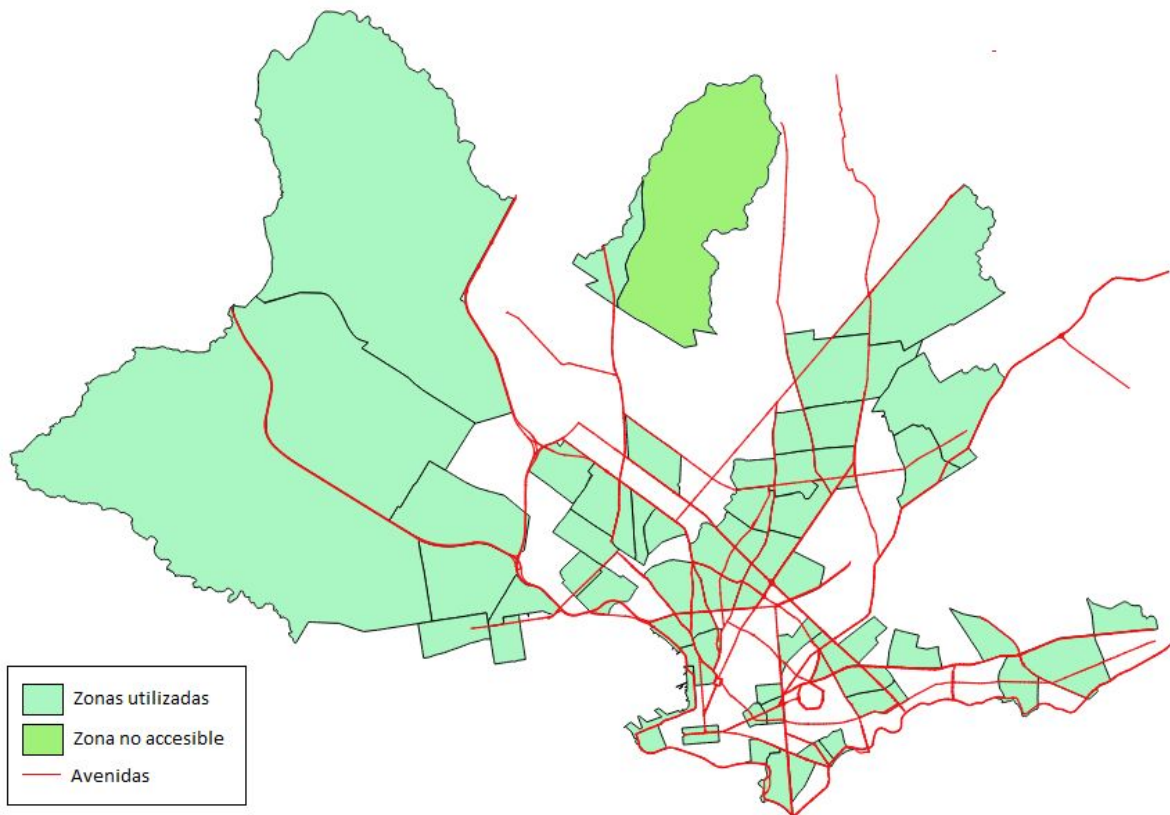


Figura 7.8: Zonas utilizadas accesibles.

En los problemas de transporte se suelen utilizar centroides para determinar los puntos de acceso a las zonas, es decir, los puntos donde se genera la demanda especificada en la matriz OD. Estos son elementos ficticios, sin correspondiente geográfico real, generalmente consistente en el baricentro de una zona [50]. En este proyecto se utiliza otro enfoque. Se intersectaron, respectivamente, la zonas origen y destino con la red de avenidas, y se eligió un vértice perteneciente a la intersección de la zona y dicha red (Figura 7.9).

Para el caso de una zona no accesible se eligió un extremo perteneciente al tramo de avenidas más próximo a la zona.

El flujo que se desea transportar entre zonas se obtuvo de la matriz origen-destino, entregada en una hoja de cálculo en formato .xls.

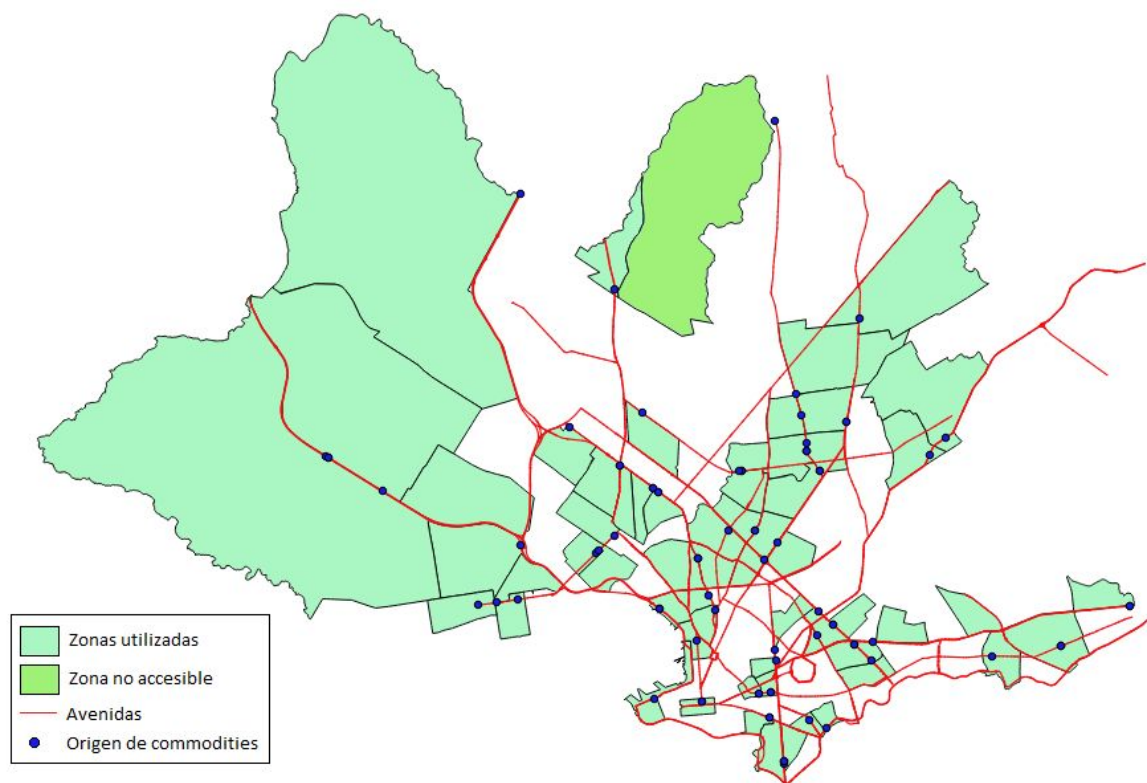


Figura 7.9: Punto de acceso de los viajes según las zonas utilizadas accesibles.

En la Figura 7.10 se muestran las zonas con mayor cantidad de flujo saliente o entrante.

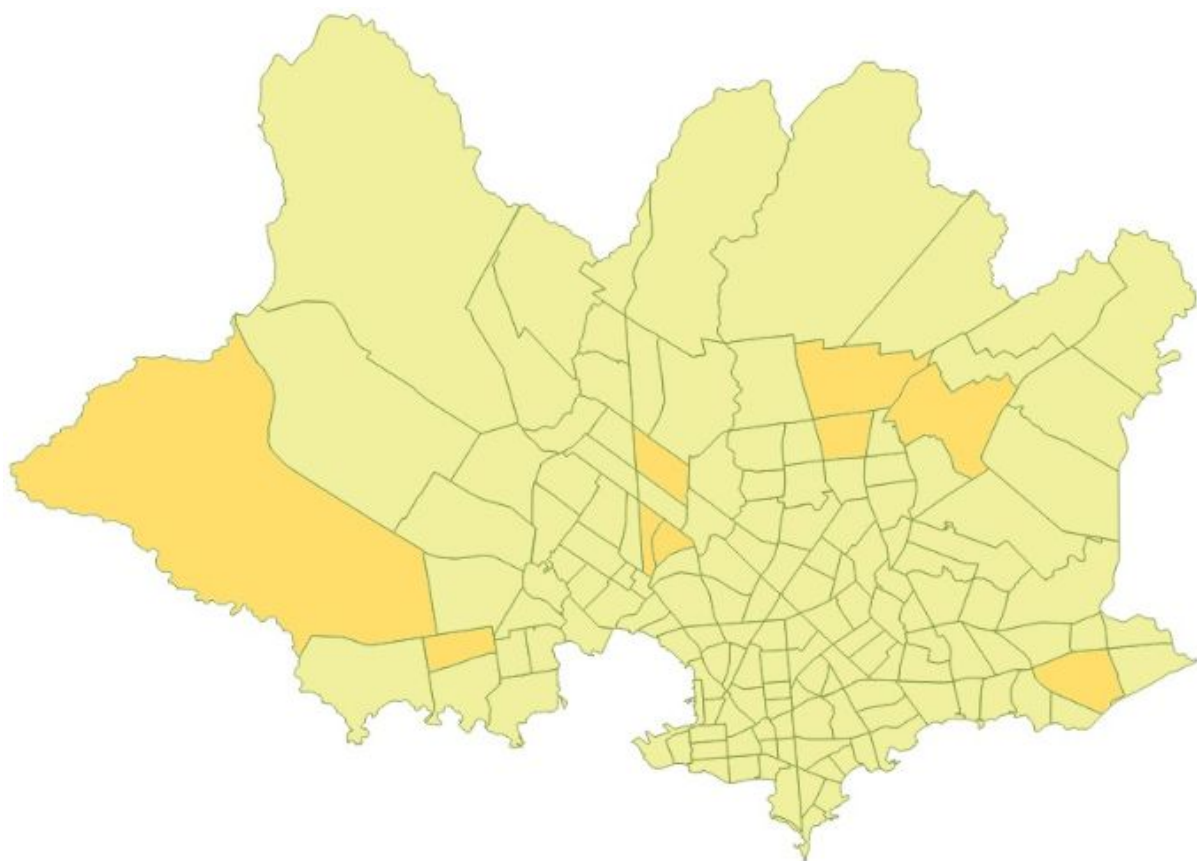


Figura 7.10: Zonas con mayor cantidad de flujo saliente o entrante.

7.3 Visualización de los datos

Una vez aplicado el modelo de optimización al caso de estudio se obtiene una solución con los recorridos optimizados para los pares origen-destino, indicando en qué partes es más conveniente construir ciclovía y en cuáles no.

Se implementó la exportación a formato shapefile de dichas soluciones, las cuales incluyen los tramos de avenidas involucrados (Figura 7.11).

Para poder visualizar los shapefiles se utilizó la herramienta QGIS Desktop [51].

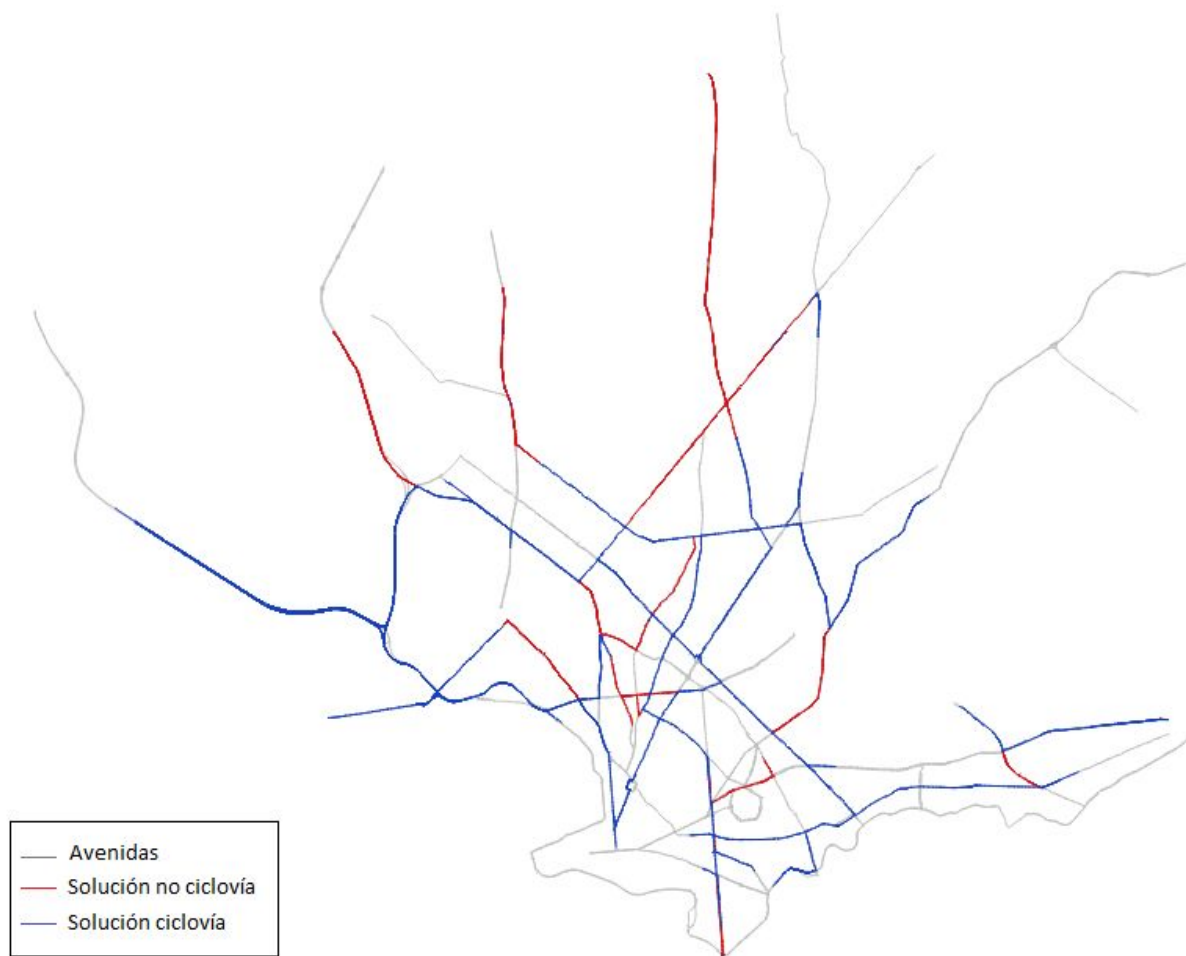


Figura 7.11: Exportación de la solución del caso de prueba N°2 del caso de estudio de Montevideo sin el control de discontinuidades. (ver sección 8.5).

Capítulo 8

Experimentos computacionales

Este capítulo presenta los resultados numéricos que se obtuvieron al realizar pruebas del modelo de optimización, que incluye resultados tanto del método exacto como de la heurística planteada para el problema. Las pruebas fueron realizadas ejecutando el algoritmo GRASP propuesto, y comparando en algunos casos, los resultados que se obtienen de ejecutar el método exacto en las mismas condiciones.

Para todos los casos en el modelo se agregó la restricción mencionada en la sección 3.4, que fuerza a que la ciclovía se construya en ambos sentidos, por lo tanto siempre que se decida construir ciclovía en un arco (i,j) , también se construirá en el arco (j,i) . Para esto, por cada arco se agregó un arco con la misma dirección y sentido opuesto. Esto fue agregado dado que en algunos casos de prueba, por ejemplo el caso de estudio Montevideo, el grafo generado para la red no era fuertemente conexo y no se lograba satisfacer todos los pares origen-destino de la red. En este caso el grafo sí era débilmente conexo por lo cual no se consideró el sentido de las calles, haciendo que éstas fueran en ambos sentidos.

Las pruebas se clasifican en las siguientes categorías: Efectividad del algoritmo; Análisis de sensibilidad respecto al parámetro de presupuesto; Análisis de sensibilidad respecto a costos y penalizaciones; Aplicación del caso de estudio. En las últimas tres se analiza la variación, ante diferentes cambios, del porcentaje de flujo de ciclistas por ciclovía con respecto al total del flujo. Este flujo es calculado como la sumatoria del flujo generado por cada par origen-destino en cada uno de los tramos donde hay ciclovía construida ($\sum_{k \in K} \sum_{(i,j) \in A} f_{ij}^k$).

8.1 Efectividad del algoritmo

El primer objetivo de esta sección es probar que efectivamente existen casos que no pueden ser resueltos si se utiliza el método exacto, y que por lo tanto se necesita la metaheurística para encontrar un resultado aproximado. En particular se verificó que el caso de Montevideo no es posible resolverlo con el método exacto.

El segundo objetivo de esta sección es realizar pruebas que permitan analizar la efectividad del algoritmo, y obtener un margen de error aproximado en base a la comparación de resultados entre el método exacto y la metaheurística.

Para la selección de los casos de prueba se analizaron cuáles eran las diferentes variaciones que se podían tener en el universo de instancias del problema, y estas se dividen en dos grupos: variaciones en el tamaño y variaciones en la estructura. Las variaciones en el tamaño pueden ser con respecto a la cantidad de nodos, la cantidad de aristas o la cantidad de pares origen-destino. Las variaciones en la estructura pueden ser sobre la estructura de la red o sobre la estructura de la matriz origen-destino. Debido a que la estructura de la red para todos los casos es similar, y es una estructura que se asemeja a la de una ciudad, esto hace que algunos de los otros parámetros de variación no sea necesario tenerlos en cuenta a la hora de seleccionar los casos de prueba. Dada una cierta cantidad de nodos n , la cantidad de aristas va a ser aproximadamente proporcional teniendo en cuenta que la estructura de la red es plana (se asemeja a una ciudad). Además para todos los casos se decidió que la cantidad de pares origen-destino sea el 5% de n^2 . Este porcentaje es consistente con la matriz origen-destino de viajes en Transporte Público de la ciudad de Rivera, caso de estudio utilizado en [50]. Por lo tanto, esto limita a que la variación elegida para los casos de prueba sea solamente la cantidad de nodos.

Los casos 1 al 4 corresponden a grafos de diferente cantidad de nodos y aristas. El nombre de la prueba indica las cantidades utilizadas, por ejemplo: *prueba_10n_32a* corresponde a una prueba con 10 nodos y 32 aristas. Para los casos 5 y 6 se utilizan ciudades de EE UU [38].

El margen de error a analizar se calcula utilizando el valor de la función objetivo del método exacto (f_{exacto}) y el método aproximado ($f_{aproximado}$), como:

$\frac{f_{aproximado} - f_{exacto}}{f_{exacto}}$. De esta forma se logra saber cuánto se aleja el resultado del método aproximado con el método exacto.

Los casos de prueba 1, 2, 3 y 4 han sido realizadas con una configuración particular de ciertos parámetros, ya que se identificó que para el objetivo de esta sección no era relevante estudiar el cambio en los mismos. En las siguientes secciones se estudiará la variación de algunos de dichos parámetros.

La configuración mencionada es:

- $c_{ij} = 1 \quad \forall (i,j)$
- $c'_{ij} = 1,5 \quad \forall (i,j)$
- $l_{ij} = 1 \quad \forall (i,j)$
- Se configura el algoritmo de Yen para que retorne $m = 10$ caminos.
- Iteraciones GRASP = 800.

Dada esta configuración, donde los arcos de la red tienen el mismo costo y penalización, puede existir más de una solución óptima.

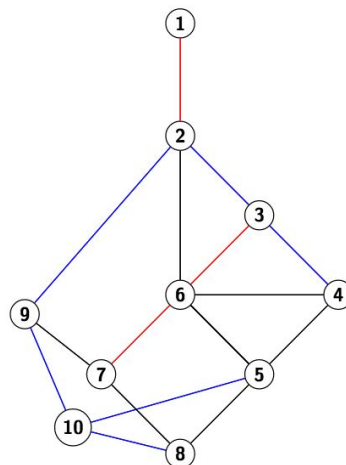
Las pruebas fueron ejecutadas en una máquina Core i7 con 16 GB de RAM. La versión de CPLEX utilizada fue 12.6.3.0.

A continuación se presentan los casos de prueba antes mencionados, mostrando tanto las características del caso como los resultados obtenidos luego de aplicar el modelo de optimización:

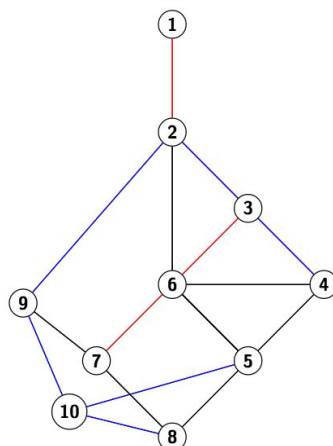
Caso 1

Nombre de caso de prueba	prueba_10n_32a
Cantidad de pares origen-destino	11
Presupuesto	12
Resultado método exacto	126,5
Tiempo ejecución método exacto	0.130216 s
Resultado método aproximado	126,5
Tiempo de ejecución método aproximado	5 s
Margen de error	0%

Grafo solución de GRASP



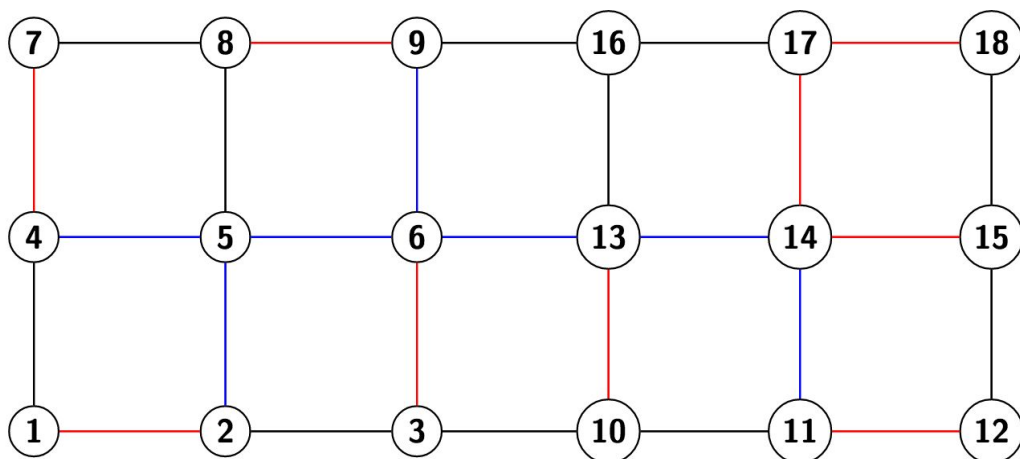
Grafo solución de CPLEX



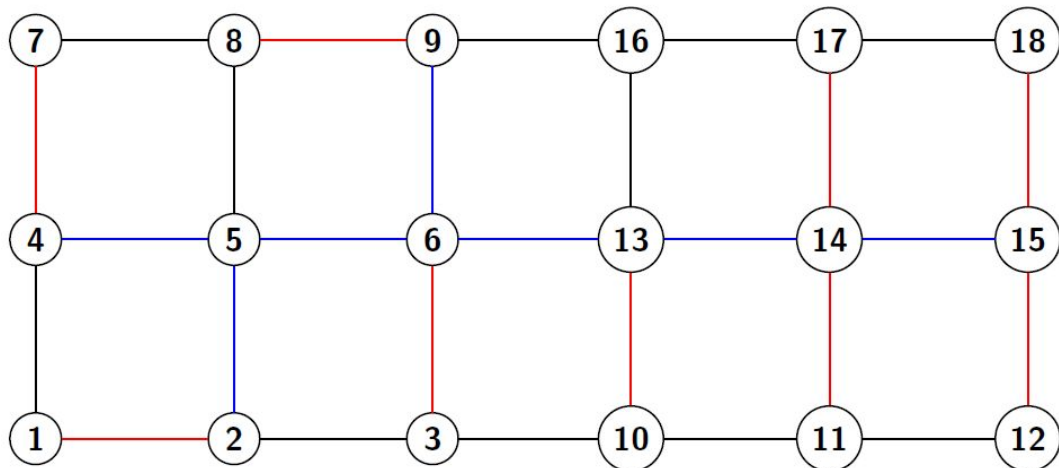
Caso 2

Nombre de caso de prueba	prueba_18n_54a
Cantidad de pares origen-destino	14
Presupuesto	15
Resultado método exacto	59,5
Tiempo ejecución método exacto	0.219726 s
Resultado método aproximado	59,5
Tiempo de ejecución método aproximado	83 s
Margen de error	0%

Grafo solución de GRASP



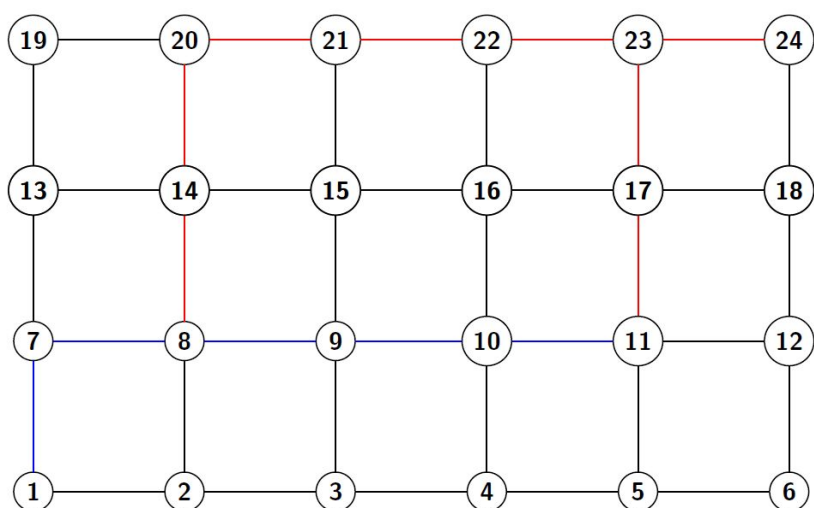
Grafo solución de CPLEX



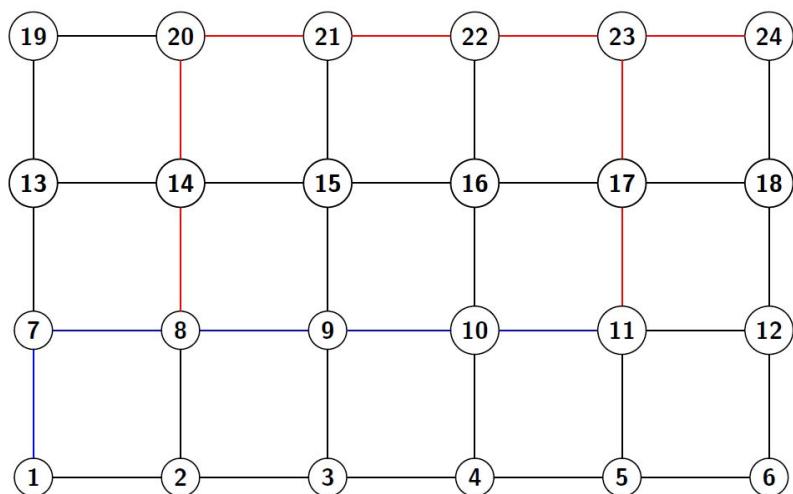
Caso 3

Nombre de caso de prueba	prueba_24n_76a
Cantidad de pares origen-destino	7
Presupuesto	10
Resultado método exacto	220,5
Tiempo ejecución método exacto	0,227212 s
Resultado método aproximado	220,5
Tiempo de ejecución método aproximado	30 s
Margen de error	0%

Grafo solución de GRASP



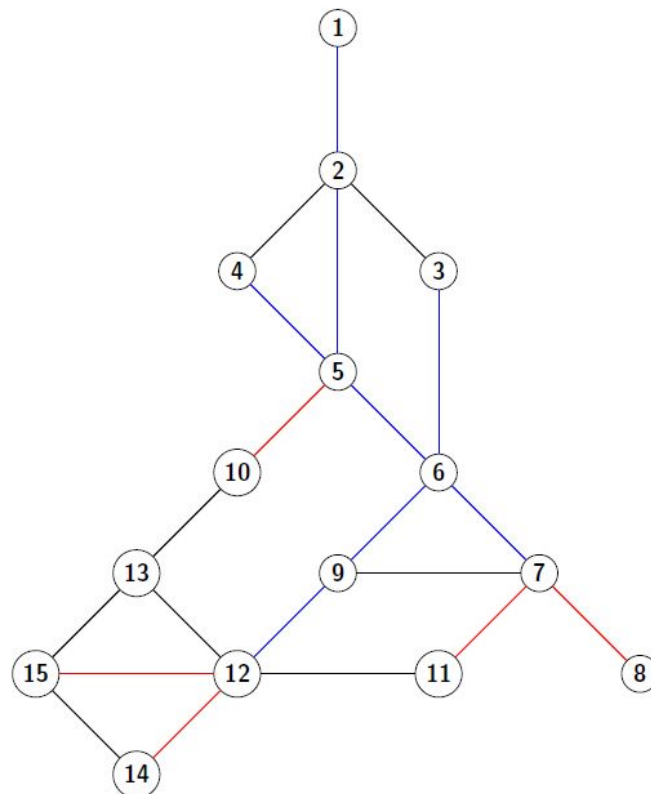
Grafo solución de CPLEX



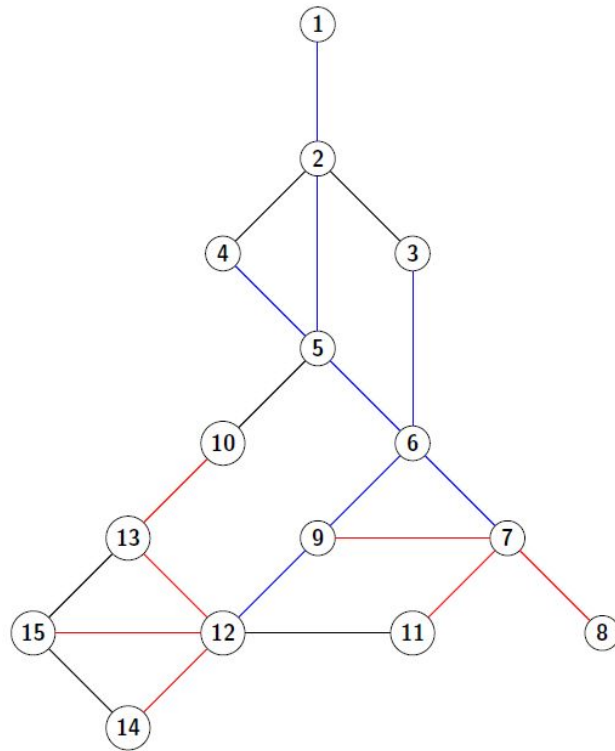
Caso 4

Nombre de caso de prueba	prueba_15n_42a
Cantidad de pares origen-destino	12
Presupuesto	16
Resultado método exacto	77
Tiempo ejecución método exacto	0,218031 s
Resultado método aproximado	81
Tiempo de ejecución método aproximado	28 s
Margen de error	5,2 %

Grafo solución de CPLEX



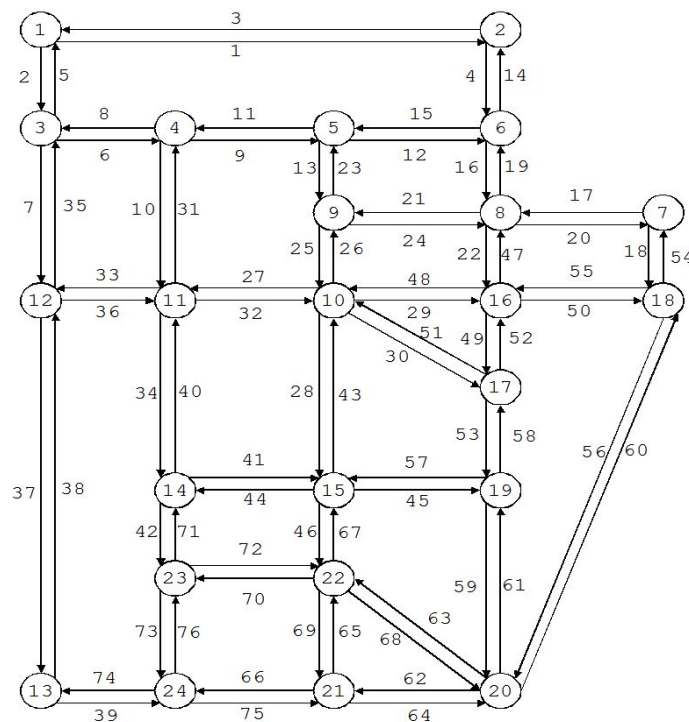
Grafo solución de GRASP



Caso 5

Nombre de caso de prueba	SiouxFalls
Cantidad de aristas	76
Cantidad de nodos	24
Cantidad de pares origen-destino	528
Presupuesto	116818 (15% del largo de todas las aristas)
Costo	Largo
Penalización	Largo x 2
Largo	Extraído del archivo de la ciudad.
Resultado método exacto	9037067999
Tiempo ejecución método exacto	13,297 s
Resultado método aproximado	9555139689
Tiempo de ejecución método aproximado	245 s
Margen de error	5,73%

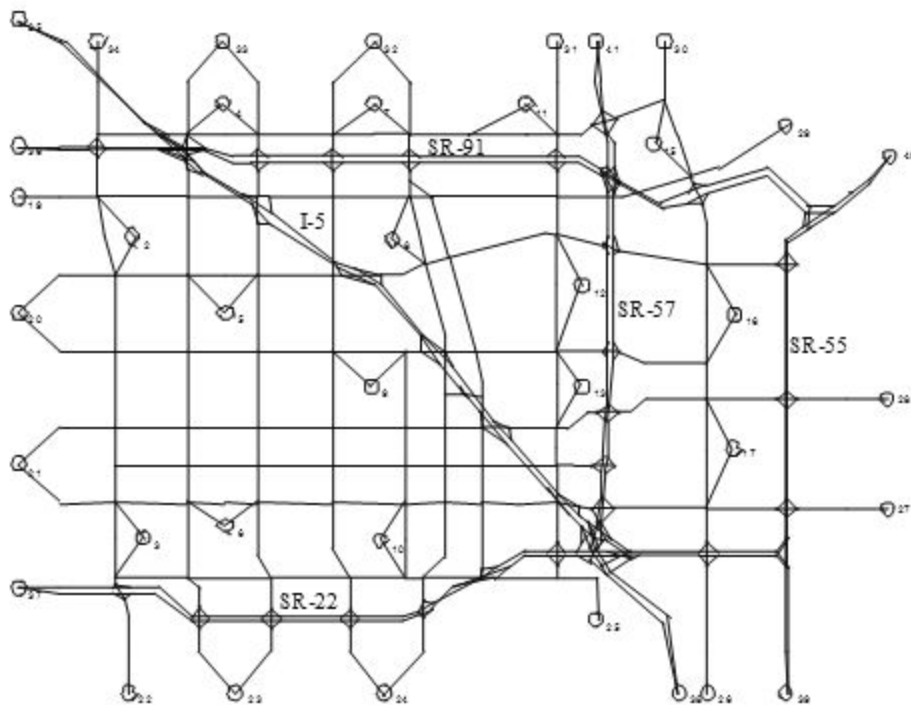
Mapa



Caso 6

Nombre de caso de prueba	Anaheim
Cantidad de aristas	432
Cantidad de nodos	200
Cantidad de pares origen-destino	812
Presupuesto	477900 (aprox. 15% del largo de todas las aristas)
Costo	Largo
Penalización	Largo x 2
Largo	Extraído del archivo de la ciudad.
Resultado método exacto	10448116380
Tiempo ejecución método exacto	18 h.
Resultado método aproximado	10537569180
Tiempo ejecución método aproximado	23 m
Margen de error	0.86 %

Mapa



En base a los resultados obtenidos se pueden realizar las siguientes observaciones:

- El margen de error o gap entre el método exacto y el algoritmo propuesto es menor al 6 % para los casos probados, lográndose en algunos casos el 0%.
- Se puede observar que, generalmente, para grafos con menor cantidad de nodos y pares origen-destino sobre la red el margen de error es menor.
- Para el caso 6 se modificó el archivo original de la ciudad ya que con la cantidad de nodos inicial no se pudo encontrar una solución con el método exacto. Mediante la disminución en la cantidad de nodos, y aristas relacionadas, se logró encontrar solución exacta de modo de poder realizar la comparación el método aproximado.
- Con respecto a los tiempos de ejecución, se puede ver que los tiempos ejecución del método exacto son menores que los del método aproximado para los casos chicos, pero para casos grandes como el 6, se observa que el tiempo de ejecución del método aproximado es significativamente menor.

8.2 Análisis de sensibilidad respecto al parámetro de presupuesto.

Un parámetro fundamental del modelo de optimización es el parámetro de presupuesto, ya que éste limita la cantidad de ciclovía a construir, por lo tanto mediante variaciones en el mismo los resultados podrían ser considerablemente diferentes. El objetivo de esta sección de las pruebas es estudiar el comportamiento del modelo ante la variación de este parámetro, analizando la sensibilidad ante diferentes cambios.

El caso de estudio seleccionado para esta sección es el de SiouxFalls. Las pruebas fueron realizadas con 800 iteraciones GRASP en todos los casos. Se configura el algoritmo de Yen para que retorne $m = 10$ caminos.

A continuación se presentan los resultados obtenidos.

Presupuesto	Método exacto	Flujo por ciclovía	Tiempo de ejecución (segundos)
7788 (1%)	24715576920	0%	0,4178
38939 (5%)	21053546830	27,3%	1005,0900
116818 (15%)	9037067999	65,3%	13,5007
233636 (30%)	7422846567	94,6%	9,4802
389394 (50%)	7422846567	94,6%	9,4774
778788 (100%)	6178894231	100%	1,0077

Cuadro 8.2.1: Pruebas realizadas sobre el el caso SiouxFalls.

Al analizar los resultados de las pruebas se comprueba que al aumentar el parámetro de presupuesto aumenta el flujo de ciclistas que circulan por tramos con ciclovía construida. Notar que con un 30% del presupuesto ya se construye más del 90%.

8.3 Análisis de sensibilidad respecto a costos y penalizaciones.

El objetivo de esta sección de las pruebas es estudiar el comportamiento del modelo ante la variación de la penalización con respecto al costo, es decir la variación del costo de circular por un tramo sin ciclovía en relación al costo de circular por el mismo con ciclovía construida. Una vez fijados los costos, se variará la penalización entre: penalización menor al costo, penalización igual al costo y penalización mayor al costo. La pruebas se corren con el método exacto. El presupuesto utilizado para todos los casos es 37%.

La Tabla 8.3 presenta los resultados obtenidos.

Costo	Penalización	Flujo por ciclovía
1	0,5	0%
1	1	14,2%
1	1,5	89,17%
1	2	91,9%

Tabla 8.3: Pruebas realizadas sobre el el caso prueba_10n_32a

En las pruebas se observa que el porcentaje de flujo por tramos con ciclovía aumenta al incrementar la penalización. En el caso en que la penalización sea menor al costo, no se construye ciclovía, esto es porque al ciclista le es menos costoso circular por tramos sin ciclovía. Una vez que la penalización es mayor que el costo, el flujo por ciclovía aumenta considerablemente. Este incremento se debe a que cuando la penalización es mayor que el costo, el modelo intentará maximizar el flujo por ciclovía, es decir, intenta construir más ciclovías, dado que la circulación por fuera de las mismas es más costoso para los ciclistas.

8.4 Aplicación del algoritmo al caso de estudio Montevideo

Los objetivos de esta sección de pruebas son:

- Encontrar una solución aproximada al caso de estudio de Montevideo ya que como se mencionó en el capítulo 5 para este caso no se encontró una solución exacta.
- Analizar las soluciones encontradas al variar el presupuesto disponible y el control de discontinuidades en la construcción de ciclovías del caso de estudio elegido.

Las pruebas fueron realizadas en el sistema operativo Windows 10 Home con un procesador Intel(R) Core(™) i7-3630QM CPU 2.40GHz, y memoria RAM de 8GB.

Las pruebas se dividen en dos bloques, el primero donde no se tiene en consideración el control de discontinuidades, y el segundo donde sí se tiene en cuenta.

Se decidió utilizar para cada bloque de pruebas tres presupuestos diferentes. Para determinar los valores de dichos presupuestos se obtuvo y analizó los kilómetros de ciclovías construidos en la actualidad sobre las calles de Montevideo (se procesó el shapefile de ciclovías obtenido de los datos abiertos de la IM). Luego se obtuvo los kilómetros de calles del departamento. Con estos dos datos se calculó el porcentaje de calles con ciclovías construidas obteniéndose como resultado un 3%. Este valor será utilizado como un presupuesto mínimo para analizar qué solución brinda nuestro algoritmo. Como un presupuesto máximo se decidió elegir los kilómetros de avenidas construidas en el departamento y como un valor intermedio la mitad de dicho valor.

Para las pruebas realizadas con el caso de Montevideo se decidió igualar el costo de cada tramo (i,j) a su largo ($c_{ij} = l_{ij}$) y su penalización igual a $\frac{3}{2}$ de su largo ($c'_{ij} = \frac{3}{2}l_{ij}$). Se configura el algoritmo de Yen para que retorne $m = 10$ caminos. Se plantean dos pruebas: una con 100 iteraciones GRASP y otra con 800. Se analizará para ambos bloques de pruebas, el flujo por ciclovía y fuera de ella y el tiempo de ejecución de los métodos de optimización.

La Tabla 8.4 presenta los resultados obtenidos para 100 iteraciones GRASP.

Nº de prueba	Presupuesto (km)	Flujo por ciclovía	Flujo no ciclovía	Tiempo de ejecución hh:mm:ss
SIN CONTROL DE DISCONTINUIDADES				
1	11,451	6%	94%	03:39:48
2	233,405	91%	9%	02:45:54
3	466,809	100%	0%	00:57:59
CON CONTROL DE DISCONTINUIDADES				
4	11,451	9%	91%	03:46:35
5	233,405	92%	8%	02:52:48
6	466,809	100%	0%	01:04:01

Tabla 8.4: Pruebas realizadas sobre el Caso de estudio de Montevideo con 100 iteraciones GRASP.

La Tabla 8.5 presenta los resultados obtenidos para 800 iteraciones GRASP.

Nº de prueba	Presupuesto (km)	Flujo por ciclovía	Flujo no ciclovía	Tiempo de ejecución hh:mm:ss
SIN CONTROL DE DISCONTINUIDADES				
1	11,451	11%	89%	29:29:57
2	233,405	92%	8%	25:00:33
3	466,809	100%	0%	08:54:03

CON CONTROL DE DISCONTINUIDADES				
4	11,451	11%	89%	25:41:45
5	233,405	93%	7%	21:48:58
6	466,809	100%	0%	08:57:12

Cuadro 8.5: Pruebas realizadas sobre el Caso de estudio de Montevideo con 800 iteraciones GRASP.

8.4.1 Análisis de resultados.

El caso de prueba Montevideo encontró una solución aproximada en todas las pruebas realizadas. La solución es coherente con la matriz origen-destino de entrada, dado que los tramos construidos se concentran en aquellas zonas con mayor flujo de ciclistas entrante o saliente. Esto puede comprobarse en la Figura 8.1, la cual corresponde a la prueba con presupuesto mínimo.

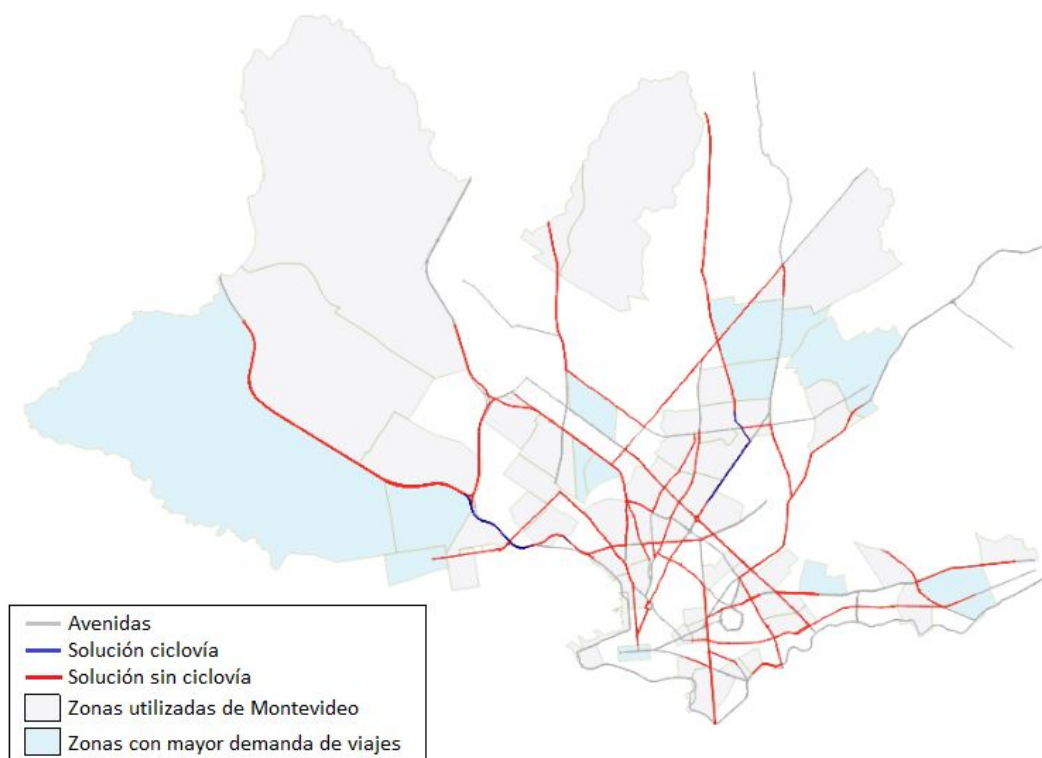


Figura 8.1: Solución con presupuesto mínimo y zonas con mayor flujo de ciclistas destacadas

Presupuesto

En todas las pruebas se observó que conforme aumenta el presupuesto los tiempos de ejecución disminuyen. Esto se debe a que al haber más presupuesto, se construye más ciclovía, por lo que mayor cantidad de ciclistas circulan por tramos con ciclovía. Esto lleva a que el algoritmo tenga que probar menos cambios para intentar mejorar la solución, dado que los cambios que realiza el mismo son tomar tramos sin ciclovía e intentar construir en ellos, al haber una menor cantidad de estos tramos, la cantidad de cambios posibles es menor.

Discontinuidades

Dado que las discontinuidades presentan un reto para los ciclistas, uno de los objetivos de estas pruebas fue agregar el control de discontinuidades para ver el impacto de dicho control sobre la red de ciclovías que se obtiene como resultado. En las pruebas con presupuesto medio, se observó que el control de discontinuidades del algoritmo provoca efectivamente una disminución de las mismas a lo largo de los caminos construidos.

Al analizar las soluciones en el mapa se puede observar que las discontinuidades en la solución disminuyeron (Figura 8.2 y 8.3). Previo al control de discontinuidades habían 38 discontinuidades, luego las mismas disminuyeron a 24. El tiempo de ejecución entre ambos bloques de pruebas aumentó levemente cuando se agregó este control.

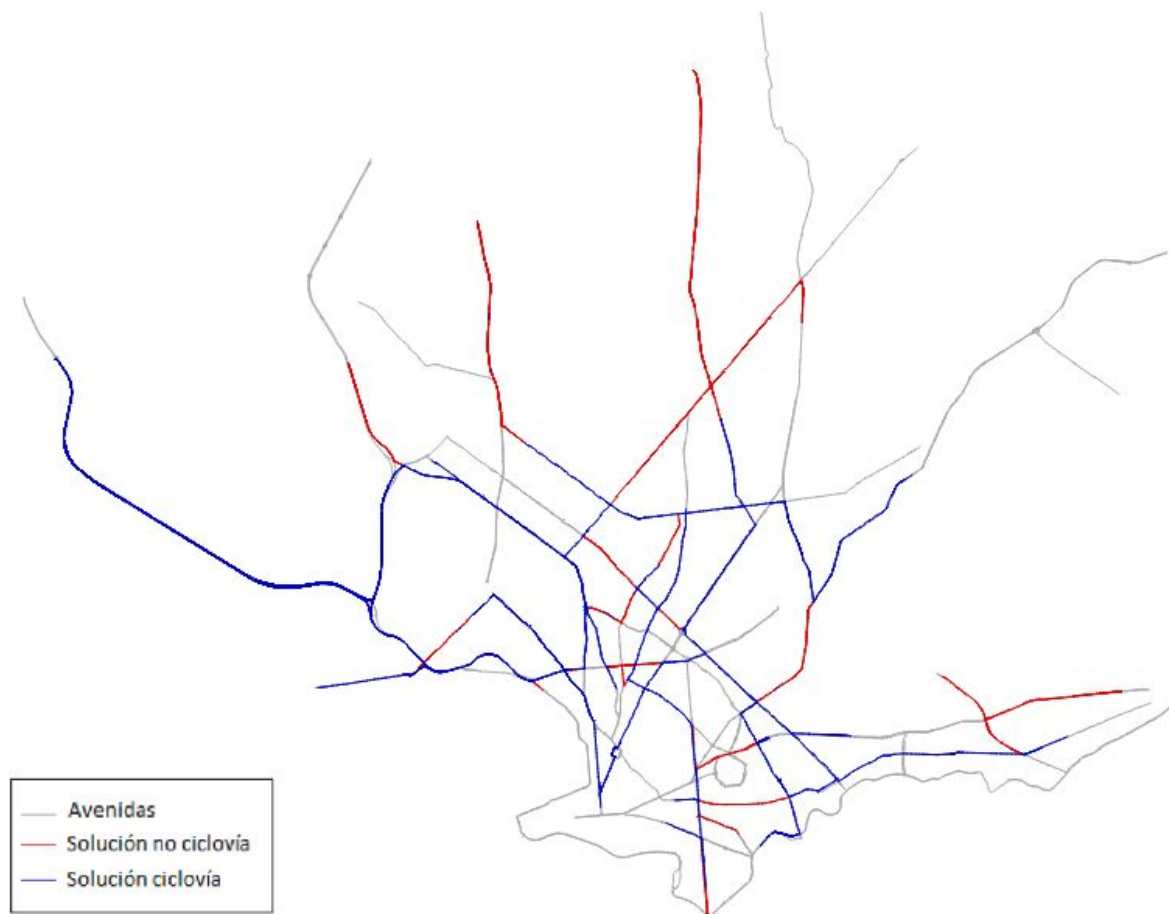


Figura 8.2: Exportación de la solución del caso de prueba N°2 del caso de estudio de Montevideo sin el control de discontinuidades.

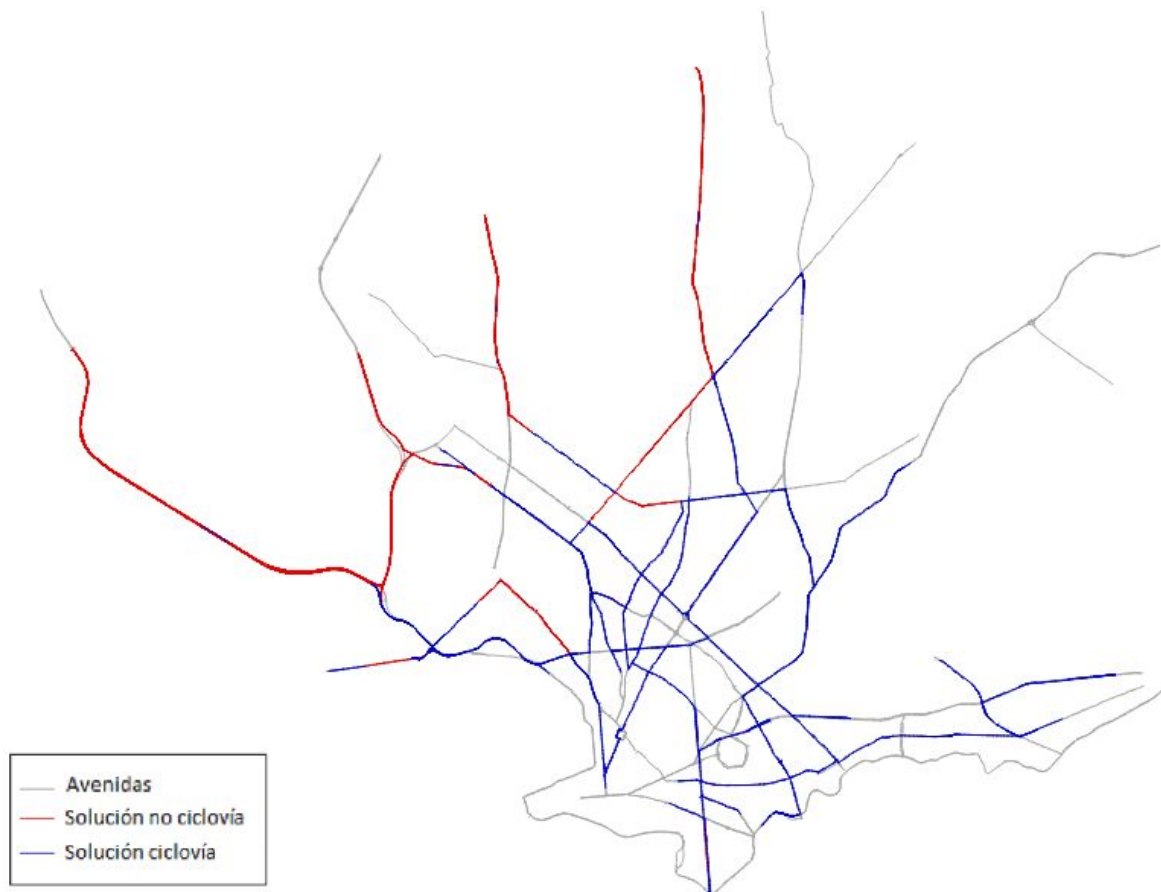


Figura 8.3: Exportación de la solución del caso de prueba N°5 del caso de estudio de Montevideo con el control de discontinuidades.

Durante el proceso de pruebas se observó un comportamiento particular en algunas de las soluciones encontradas (Figura 8.4). En algunos casos se detectó que en avenidas con doble vía (por ejemplo en la calle Rambla Edison) las soluciones muestran que si bien hay ciclovía construida por una de sus vías, el algoritmo da como solución no circular por ella para cumplir alguno de los viajes. Esto se debe a que el shapefile de avenidas obtenido tiene datos incompletos, presentándose en algunos casos discontinuidades en las avenidas que no existen en la realidad. Las avenidas doble vía procesadas, en algunos casos, no se conectan entre sí por lo cual ambas vías son consideradas como calles diferentes. Para solucionar este problema se debería hacer un procesamiento adicional del shapefile, de forma de

agregar las conexiones faltantes, que no fue realizado dado que no formaba parte del alcance del proyecto.

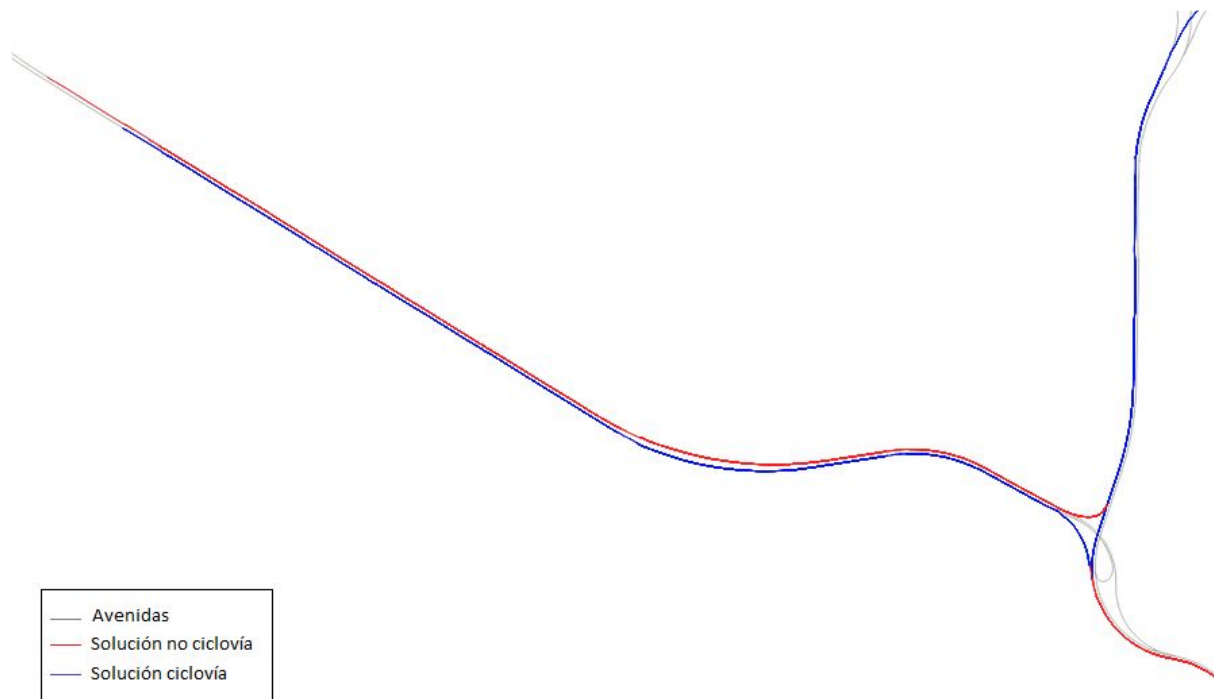


Figura 8.4: Caso particular de las soluciones en el caso de prueba de Montevideo.

Capítulo 9

Conclusiones y trabajos a futuro

Conclusiones

Se lograron identificar las principales características a tener en cuenta a la hora de construir una red de ciclovías y se logró un modelo de programación matemática que incluye las mismas.

Se encontraron empíricamente las limitaciones de eficiencia del método exacto. Se comprobó que para redes con gran cantidad de arcos y nodos, o para realidades con gran cantidad de pares origen-destino, no era posible encontrar la solución óptima utilizando el método exacto. Debido a esto se diseñó e implementó un método aproximado.

Se propuso una solución al problema mediante el uso de una metaheurística GRASP. Se realizaron pruebas para determinar la efectividad de la misma, y se comprobó que el margen de error para los casos estudiados estuvo por debajo del 6%. El promedio obtenido fue de 1,97%, y el mayor margen fue de 5,73%.

Se comprobó que era relevante estudiar el problema de las discontinuidades e incluirlas en el modelo. Se logró implementar una mejora en el algoritmo para disminuir las discontinuidades en la solución obtenida, y mediante las pruebas realizadas se comprobó que dicho control efectivamente reduce las mismas.

Se pudo obtener los datos geográficos necesarios. Los mismos fueron procesados para poder construir la red del caso de estudio de Montevideo. Además se obtuvo una matriz origen-destino real para poder aplicar el algoritmo al caso. Se logró validar el modelo y el algoritmo propuesto con un caso de estudio real, y se pudo obtener una solución para el mismo. La misma fue analizada y se pudo observar que efectivamente la misma es consistente con los datos de entrada, es

decir que se construye ciclovía por las zonas donde existe una mayor concentración de viajes.

Finalmente, al cumplirse todos los objetivos propuestos, se concluye que se logró cumplir con el alcance de este proyecto.

Trabajo a futuro

Con respecto a lo desarrollado en el presente trabajo, se identifican las siguientes líneas de continuación del mismo:

- Profundizar el estudio y desarrollo del modelo de programación matemática con el objetivo de encontrar la solución al problema sin la necesidad de utilizar una metaheurística.
- Incluir en el modelo características a tener en cuenta a la hora de diseñar redes de ciclovías que no impliquen grandes cambios en la formulación actual. Por ejemplo, pendientes. El modelo actual es flexible, y permite que los costos puedan ser definidos teniendo en cuenta las pendientes, pero sería deseable hacer un estudio profundo de esta problemática y evaluar los posibles cambios que pudieran ser necesarios en el modelo.
- Agregar al modelo costos en los nodos, para tener en cuenta, por ejemplo, cruces complejos o peligrosos, donde se requieren inversiones considerables de infraestructura.
- Si bien la metaheurística desarrollada retorna una solución aproximada al problema, se sabe que esta podría mejorarse. Para esto se podría realizar una exploración de otras metaheurísticas que se ajusten al problema, o perfeccionar la metaheurística propuesta, por ejemplo cambiando la exploración local de soluciones que se realiza en la misma.
- Los datos utilizados sobre la encuesta de movilidad urbana fueron relevados en el año 2009, por lo cual posiblemente, no son representativos de la

situación actual de la ciudad de Montevideo. Sería de interés contar con datos más recientes.

- Procesar la red completa de calles disponible en los datos abiertos de la IM para seleccionar las calles aptas para la construcción de ciclovía, teniendo en cuenta también el sentido de las mismas. Construir el caso de estudio de Montevideo utilizando dicha red.

Anexo A: Archivo .mod AMPL

A continuación se presenta el archivo .mod AMPL [52] del modelo matemático:

```
set N; #nodos
```

```
set A; #aristas
```

```
set K; #commodities
```

```
set outgoing {N} within A; #aristas salientes del nodo N
```

```
set incoming {N} within A; #aristas entrantes al nodo N
```

```
param origen_arista {A} > 0 integer; #origen de la arista A
```

```
param destino_arista {A} > 0 integer; #destino de la arista A
```

```
param c {A} >= 0; #costos de la arista A
```

```
param c_p {A} >= 0; #costo penalizado de la arista A
```

```
param R {K} > 0; #required amount of flow of commodity K
```

```
param origin {K} > 0 integer; #origen de la commodity K
```

```
param destin {K} > 0 integer; #destino de la commodity K
```

```
#param Kmax {A} > 0 integer; #maximum capacity of arc A
```

```
param largo {A} > 0; #largo de la arista A
```

```
param maxLargo > 0; #largo m?ximo
```

```
param Kmax > 0 integer; #m?ximo de capacidad
```

```
var flujos_ciclovía {A,K} >= 0;
```

```
var flujos_no_ciclovía {A,K} >= 0;
```

```
var y {A} binary; #igual a 1 si hay ciclovía y se decide ir por ahí.
```

var $x\{A\}$ binary; #no hay ciclo? a construida en A pero igual elijo ir por ah?.

minimize z: $\sum\{a \text{ in } A\} (\sum\{k \text{ in } K\} (c[a] * \text{flujos_ciclovía}[a,k] + c_p[a] * \text{flujos_no_ciclovía}[a,k]))$;

s.t. $\text{una_u_otra}\{a \text{ in } A\}: x[a] + y[a] \leq 1$;

s.t. $\text{capacidad_flujo}\{a \text{ in } A\}: \sum\{k \text{ in } K\} (\text{flujos_ciclovía}[a,k]) \leq K_{\max} * y[a]$;

s.t. $\text{capacidad_flujo_p}\{a \text{ in } A\}: \sum\{k \text{ in } K\} (\text{flujos_no_ciclovía}[a,k]) \leq K_{\max} * x[a]$;

s.t. $\text{largos}: \sum\{a \text{ in } A\} (\text{largo}[a] * y[a]) \leq \text{maxLargo}$;

s.t. $\text{control_flujo}\{n \text{ in } N, k \text{ in } K\}: \sum\{a \text{ in } \text{outgoing}[n]\} (\text{flujos_ciclovía}[a,k] + \text{flujos_no_ciclovía}[a,k]) - \sum\{a \text{ in } \text{incoming}[n]\} (\text{flujos_ciclovía}[a,k] + \text{flujos_no_ciclovía}[a,k]) = (\text{if } n=\text{origin}[k] \text{ then } R[k] \text{ else if } n=\text{destin}[k] \text{ then } -R[k] \text{ else } 0)$;

s.t. $\text{bidireccionalidad_y}\{a1 \text{ in } A, a2 \text{ in } A: \text{origen_arista}[a1] = \text{destino_arista}[a2] \text{ and } \text{origen_arista}[a2] = \text{destino_arista}[a1]\}: y[a1]=y[a2]$;

s.t. $\text{bidireccionalidad_x}\{a1 \text{ in } A, a2 \text{ in } A: \text{origen_arista}[a1] = \text{destino_arista}[a2] \text{ and } \text{origen_arista}[a2] = \text{destino_arista}[a1]\}: x[a1] = x[a2]$;

Anexo B: Manual de instalación y ejecución

Manual de instalación

En este anexo se detalla el software necesario y los pasos que se deben seguir para poder ejecutar el caso de estudio de Montevideo utilizando el algoritmo aproximado implementado en un sistema operativo Windows.

Se debe descargar e instalar lo siguiente:

1. PostgreSQL versión 9.5.1-1. [53]
2. Java 8. [54]
3. QGIS Desktop. [55]

Los pasos a seguir para la instalación y correcta ejecución de la aplicación se detallan a continuación.

1. Instalar PostgreSQL. Esta instalación también instala el programa Stack Builder.
2. Buscar el programa Stack Builder para poder instalar mediante él el módulo PostGIS. Durante esta instalación se deberá:
 1. Seleccionar la versión de PostgreSQL instalada y se presiona siguiente.
 2. En Spatial Extensions se selecciona la versión 2.0 de PostGIS.
 3. Seleccionar la opción create spatial database.
 4. Ingresar los datos para la conexión a la bd: user: postgres, password: postgres.

5. Nombre de la base de datos proyecto_postgis.
2. Agregar la variable de entorno “PG_HOME” con la ubicación del comando pgsql2shp (en windows: C:\Program Files\PostgreSQL\9.5\bin).
3. Se debe cargar la base de datos geográfica con la información de las avenidas de Montevideo y las Zonas de Transporte (estos shapefiles son entregados en la carpeta Shapefile entregada junto al código fuente).
 1. Entrar al programa PostGis.
 2. Ir a view connection details y conectarse a la base de datos. usuario y contraseña:postgres y en database: proyecto_postgis.
 3. Después seleccionar *Add file* y cargar el archivo avenidas.shp, zats.shp.
 4. Presionar *Import*.
4. Correr el script eliminar_commodities.sql. Este script es necesario correrlo ya que, dado que el shapefile de avenidas está incompleto, al procesar la matriz origen-destino se observó que no habían caminos entre el origen y destino de algunos viajes.
5. Para visualizar los shapefiles es necesario descargar e instalar QGIS Desktop.

Manual de ejecución

En esta sección se detalla un manual de ejecución de la aplicación.

Para comenzar se debe localizar el archivo proyecto.jar ubicado en la carpeta del código fuente del proyecto /proyecto/proyecto.jar. Se debe ejecutar el archivo .jar utilizando el comando de la siguiente forma: `java -jar proyecto.jar`.

La aplicación comienza preguntando qué tipo de importación de datos se realizará. Se despliegan dos opciones, una para importar la información procesando un archivo de texto (opción 1) y otra para hacerlo a través de un archivo shapefile (previamente cargado en la base de datos):

1- Texto (txt).

2- Shapefile (shp).

Se deberá ingresar el número de opción deseada. Si se selecciona el tipo de archivo de texto (opción 1), se solicita el nombre del archivo que se desea procesar (ejemplo prueba1.txt).

A continuación se debe ingresar la cantidad de iteraciones GRASP (valor mayor que cero). Luego se pregunta si se desea realizar el control de discontinuidades. Se despliegan las opciones:

1- Si.

2- No.

Se debe ingresar la opción deseada, 1 si se desea realizar el control y 2 en caso contrario.

Dados los datos ingresados anteriormente se comienza la ejecución del algoritmo. Al finalizar éste desplegará el porcentaje del flujo de ciclistas por ciclovía, el porcentaje del flujo de ciclistas que no utiliza ciclovía y el total de largo de ciclovías construido. Para visualizar los resultados puede utilizarse la herramienta QGIS Desktop si se tratan de soluciones en formato shapefile.

Bibliografía

[1] Basu, S. & Vasudevan, V. Effect of Bicycle Friendly Roadway Infrastructure on Bicycling Activities in Urban India. Procedia - Social and Behavioral Sciences 104, 1139-1148. 2013

[2] Khuperkar, A., Viechnicki, P., Fishman, T. & Eggers, W. Bike commuting: Unleashing its economic, health, and safety benefits. Deloitte University Press. 2015.
<<https://dupress.deloitte.com/dup-us-en/industry/public-sector/smart-mobility-trends-bike-commuting.html>> [Consulta: 1 de Noviembre 2016]

[3] Intendencia de Montevideo. Normativa Departamental, Volumen V Tránsito y Transporte, Artículo R.424.1 .Espacios Físicos. 1981.
<<http://normativa.montevideo.gub.uy/armado/65077>> [Consulta: 4 de noviembre 2016]

[4] La República. Montevideo sumará dos ciclovías. 2014.
<<http://www.republica.com.uy/montevideo-sumara-dos-ciclovias/464574/>>
[Consultado: 25 de Setiembre 2016]

[5] El País. Una vida en bicicleta. 2014.
<<http://www.elpais.com.uy/que-pasa/vida-bicicleta.html>> [Consultado: 25 de Setiembre 2016]

[6] Christchurch City Council. Design Guidelines Part B: Revision B Design Principles Best Practice Guide. 2016.

[7] Dutch Cycling Embassy. Sustainable safety. Bicycle inclusive planning and design.
<<http://www.dutchcycling.nl/library/file/Bicycle%20inclusive%20planning%20and%20design%20ThinkBike%20Helsinki.pdf>> - [Consulta: 13 de Mayo 2016].

- [8] Hull, A. & O'Holleran, C. Bicycle infrastructure: can good design encourage cycling, *Urban, Planning and Transport Research* 2(1), 369-406. 2014.
- [9] Brussel, M. Bicycle planning and network design. 2015
- [10] Transport for London. London Cycling Design Standards, Chapter 2: Tools and techniques. 2014 -
<<http://content.tfl.gov.uk/lcds-chapter2-toolsandtechniques.pdf>>[Consultado: 2 de Julio]
- [11] Forester, J. The bikeway controversy. *Transportation Quarterly* 55, 7–17. 2001.
- [12] Lott, D.F., Tardiff, T. & Lott, D.Y. Evaluation by experienced riders of a new bicycle lane in an established bikeway system. *Transportation Research Record* 683, 40–46. 1978.
- [13] Antonakos, C.L. Environmental and travel preferences of cyclists. *Transportation Research Record* 1438, 25–33. 1994.
- [14] Harkey, D.L. & Stewart, J.R. Evaluation of shared-use facilities for bicycles and motor vehicles. *Transportation Research Record* 1578, 111–118. 1997.
- [15] Moritz, W.E. Survey of North American bicycle commuters: design and aggregate results. *Transportation Research Record* 1578, 91–101. 1997.
- [16] Krizek, K. & Roland, R. What is at the end of the road? Understanding discontinuities of on-street bicycle lanes in urban settings, *Transportation Research Part D* 10(1), 55-68. 2005.
- [17] Zafar, D. Determination of an Optimal Bike Path Based on Slope & Distance Data. 2011.
- [18] Buehler, R. & Dill, J. Bikeway Networks: A review of Effects on Cycling. *Transport Reviews*, 36, 9-27. 2015.

- [19] Buehler, T. & Handy, S. Fifty years of bicycle policy in Davis, California. *Transportation Research Record*, 2074, 52–57. 2008.
- [20] Furth, P. Bicycling infrastructure for mass cycling. In J. Pucher & R. Buehler (Eds.), *City cycling*, 105–140. Cambridge, MA: MIT Press. 2012.
- [21] Pucher, J. & Buehler, R. Making cycling irresistible: Lessons from the Netherlands, Denmark, and Germany. *Transport Reviews*, 28(1), 495–528. 2008.
- [22] Bicycle Network. Contraflow Lanes. -
<<https://www.bicyclenetwork.com.au/general/for-government-and-business/2842/>>
[Consulta: 18 de Junio 2016].
- [23] Magnati, T.L. & Wong, R. T. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*, 18, 1-55. 1984.
- [24] Facultad de Ingeniería - UDELAR. Notas curso Introducción a la Investigación de Operaciones. <<https://www.fing.edu.uy/inco/cursos/io/>> [Consultado: 10 de Julio 2016]
- [25] Cormen, T., Leiserson, C., Rivest, R., & Stein, C. Section 24.3: Dijkstra's algorithm. *Introduction to Algorithms* (Second ed.), 595–601. 2001.
- [26] Feo, T.A. & Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6, 109-133. 1995.
- [27] Festa, P. & Resende, M.G.C. An annotated bibliography of GRASP – Part I: Algorithms. *International Transactions in Operational Research*, 16, 1–24. 2009.
- [28] Dijkstra, E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. 1959.
- [29] Yen, J.Y. Finding the k shortest loopless paths in a network. *Management Science*, 17, 712–716. 1971.

- [30] Stoll, R. Sets, Logic and Axiomatic Theories. *Journal of Symbolic Logic* 25(3), 278-279. 1960.
- [31] Osman, I. H. & Kelly, J. P. Meta-Heuristics: An Overview. 1996.
- [32] Blum, C. & Roli, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys* 35(3), 268–308. 2003.
- [33] Chinneck, J. Practical Optimization: A Gentle Introduction. 2015.
- [34] Meindl, B & Templ, M. Analysis of commercial and free and open source solvers for linear optimization problems. *Journal Transactions on Data Privacy* 6(2), 147-159. 2013.
- [35] Rothlauf, F. Design of Modern Heuristics: Principles and Application. Springer Publishing Company. 2011.
- [36] Papadimitriou, C. H. Combinatorial optimization: algorithms and complexity. Dover Publications. 1998.
- [37] Resende, M.G.C. & Ribeiro, C.C. Greedy Randomized Adaptative Search Procedures. *Handbook of Metaheuristics* 57, 219–249, Kluwer Academic Publishers. 2003.
- [38] Bar-Gera, H. Transportation Network Test Problems. 2016.
<<http://www.bgu.ac.il/~bargera/tntp/>> [Consulta: 5 de Setiembre 2016]
- [39] IBM. IBM. CPLEX Optimizer.
<<https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>>
[Consulta: 1 de Noviembre 2016].
- [40] ArcGis. Ayuda Online - Shapefiles.
<<https://doc.arcgis.com/es/arcgis-online/reference/shapefiles.htm>> [Consultado: 06 Octubre 2016]

- [41] Arizona State University. An implementation of K-Shortest Path Algorithm (Java Version). 2015. <<https://github.com/yan-qi/k-shortest-paths-java-version>> [Consulta: 15 de Enero 2016]
- [42] 180. Montevideo tiene casi 4,5 km de ciclovías. 2014. <<http://www.180.com.uy/articulo/41058>> [Consulta: 6 de Junio 2016]
- [43] Intendencia de Montevideo. Movete: Llegan las bici a Ciudad Vieja. 2014. <<http://www.montevideo.gub.uy/institucional/noticias/movete-llegan-las-bici-a-ciudad-vieja>> [Consulta: 6 de Junio 2016]
- [44] El País. Nadie respeta las ciclovías de la IM. 2014. <<http://www.elpais.com.uy/informacion/nadie-respeta-ciclovias-imm.html>> [Consulta: 6 de Junio 2016]
- [45] Hernández, D. Políticas de tiempo, movilidad y transporte público: rasgos básicos, equidad social y de género. Intendencia de Montevideo. PNUD. 2012.
- [46] Unibici. La ciudad va cambiando de color. 2014. <<http://www.unibici.edu.uy/La-ciudad-va-cambiando-de-color>> [Consulta: 19 de Junio 2016]
- [47] Instituto Nacional de Estadística <<http://www.ine.gub.uy/>> [Consulta: 3 de Noviembre 2016]
- [48] PostGIS <<http://postgis.net/>> [Consulta: 13 de Octubre 2016]
- [49] Postgres <<https://www.postgresql.org/>> [Consulta: 13 de Octubre 2016]
- [50] Mauttone, A. Optimización de Recorridos y Frecuencias en Sistemas de Transporte Público Urbano Colectivo. Tesis de Maestría en informática. PEDECIBA. Universidad de la República. 2005.
- [51] QGIS <<http://www.qgis.org/es/site/>> [Consulta: 12 de Octubre 2016]

[52] Fourer, R., Gay D. & Kernighan, B. A Modeling Language for Mathematical Programming. Management Science 36, 519-554. 1990

[53] PostgreSQL. Sitio de descarga: <https://www.postgresql.org/download/windows/>

[54] Java 8. Sitio de descarga:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

[55] QGis Desktop. Sitio de descarga:

<https://www.qgis.org/en/site/forusers/download.html>