

ANÁLISIS Y OPTIMIZACIÓN DE RECORRIDO DE CAMIONES RECOLECTORES DE RESIDUOS URBANOS

Oveja Smith, Matías; Colombo Rojas, Imanol; Xodo, Daniel, Bazan, David¹

*INTIA, Fac. de Ciencias Exactas, UNICEN
Campus Universitario, Paraje Arroyo Seco, Tandil
matiasovejasmith@gmail.com*

*(1)GESTADII, Facultad Regional Trenque Lauquen, Universidad Tecnológica Nacional,
Racedo 298, Trenque Lauquen*

RESUMEN

El objetivo del trabajo es el análisis y optimización de los recorridos de camiones recolectores de residuos urbanos y suburbanos en el Municipio de Rivadavia

El problema puede ser estudiado a partir de distintos enfoques y algoritmos de resolución. Como problema de Arbol de Expansión Mínimo (MST) utilizando los algoritmos de Kruskal o de Prim.

Son considerados en el análisis todo tipo de residuos y los sectores con distinta frecuencia de recolección en virtud de su ubicación geográfica o densidad poblacional.

A través del análisis de densidad poblacional se circunscriben las áreas diferenciadas en n sectores localizados a partir de su centro geográfico y se determina la frecuencia apropiada en función de la densidad poblacional considerando el promedio de residuos por habitante para cada uno de los sectores especificados.

Tomando como variables las horas de camión y personal ocupado mensual para la tarea es comparada la mejor alternativa con la operación actual.

El proyecto implica la participación activa de l funcionarios comunales responsables del servicio a fin de eliminar errores por desconocimiento de cuestiones operativas y es punto de partida de diversos trabajos de optimización operativa en la tarea.

Palabras Claves: **Recolección.** Optimización de recorridos. Kruskal. Prim.

ABSTRACT

The objective of this work is the analysis and optimization of the routes of collection trucks urban and suburban waste in the municipality of Rivadavia

The problem can be studied from different approaches and solution algorithms. As the problem of minimum spanning tree (MST) algorithms using Kruskal or Prim.

They are considered in the analysis all kinds of waste and sectors with different frequency of collection under their geographic location or population density.

Through analysis of the differentiated population density areas are limited in n sectors located from its geographical center and the appropriate frequency based on the population density is determined by considering the average waste per capita for each of the sectors specified.

Taking as variables the truck hours and monthly staff busy for the task is the best alternative compared to the current operation.

The project involves the active engagement of community service officials to eliminate errors due to lack of operational issues and is the starting point of various operational optimization work on the task.

1. INTRODUCCIÓN

Actualmente el municipio de Rivadavia dispone de dos camiones no compactadores destinados a la recolección de residuos. Estos realizan recorridos preestablecidos: uno recorre calles verticales mientras el otro recorre las horizontales.

Con frecuencia los operarios de los camiones se adelantan a los propios camiones para juntar la basura de una cuadra en una esquina (o en la mitad de una cuadra). A partir de la densidad de habitantes por cuadra y una estimación de la basura generada por persona (500 gr. de basura por día por persona) se puede determinar la cantidad total de residuos resultante en cada esquina. Con esto es posible aplicar algoritmos de generación de árboles de recubrimiento de costo mínimo a fin de diseñar recorridos para cada uno de los camiones que insuman el mínimo número de cuadras a recorrer permitiendo pasar por todas las esquinas con basura.

5. DESARROLLO

5.1. Situación Actual

Actualmente en el municipio de Rivadavia, en la provincia de Buenos Aires la recolección de residuos se efectúa utilizando dos camiones propios de la municipalidad sin sistema de compactación y una capacidad máxima de carga de 4 toneladas. Por lo general se suele requerir de un solo viaje de los camiones ya que su capacidad excede la producción de basura de la población. Los días domingo son especiales ya que como el Sábado no pasan los camiones la gente suele sacar más basura.

Los camiones son operados por 3 personas. Muchas veces una de ellas se adelanta al recorrido y va acumulando basura en las esquinas o a la mitad de la cuadra para que cuando pase el camión ya recoja todo de un mismo lugar.

La densidad poblacional es de 200 habitantes por manzana (50 por cuadra asumiendo una distribución homogénea).

En algunos barrios de difícil acceso para los camiones por ser los pasajes muy angostos, se han ubicado cestos generales con separaciones por vivienda.

En lo que respecta a recorridos, uno de los camiones se utiliza para recorrer cuadras horizontales y el otro para recorrer cuadras verticales.

5.2. Abstracción del Problema y su Representación

Primero que nada, para simplificar la representación y posterior manipulación de los datos, asumiremos que toda la basura es llevada a las esquinas de la forma en la que lo indica la Figura 1.



Figura 1

Según estimaciones, cada persona produce un promedio de 500 gramos de basura por día. Sabiendo que por manzana hay 200 habitantes y que por lo tanto por cuadra hay 50 habitantes cada uno de los cuales producen aproximadamente 500 gramos de basura. Con esto es posible completar una matriz que guarde en cada una de sus casillas una cantidad estimada de basura. Por supuesto si a una esquina no inciden flechas procedentes de cuadras pobladas en la casilla habrá un 0 indicando que no llega basura a la esquina en cuestión. Dicho esto, los movimientos válidos de cada camión serían moverse hacia la esquina de arriba, la de abajo, la de la izquierda o la de la derecha sin salirse de los límites de la matriz que cubriría la planta urbana.

La matriz quedaría como en la Figura 2. Puesto que la zona urbana del pueblo está inscripto en la matriz pero no tiene forma rectangular para adaptarse exactamente a la matriz, quedarán celdas de la matriz fuera de la zona habitada y que por supuesto no se corresponden con ningún cruce de calles. A

dichas celdas se les asignan un 0 en cantidad de basura para que el algoritmo a utilizar las ignore al diseñar el recorrido de los camiones.

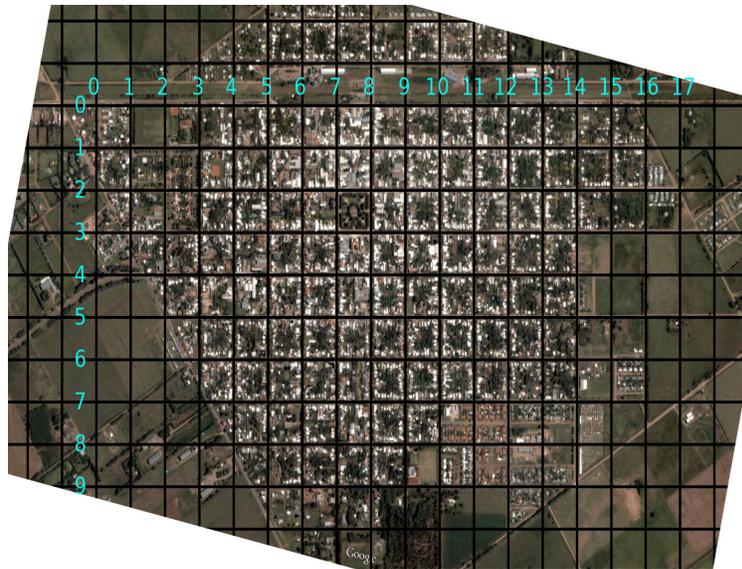


Figura 2

5.3 Algoritmo a Utilizar: Prim

El algoritmo de Prim es utilizado para obtener árboles que permiten alcanzar todos y cada uno de los nodos de un grafo con el costo mínimo (Véase 5.3.1). Puesto que para este trabajo no se utilizará un grafo sino una matriz se hace necesario adaptarlo al caso. Esto último se analizará en la subsección 5.3.3.

5.3.1 Árboles de Recubrimiento de Costo Mínimo

Antes de abordar el tema de árboles de recubrimiento es necesario comprender el concepto de árbol libre. Un árbol libre es un grafo no dirigido, acíclico y conexo de n vértices y $n-1$ aristas. Si a un árbol libre se le agrega un arco se genera un ciclo; es decir, un árbol libre tiene la mínima cantidad de arcos para que pueda ser considerado un grafo acíclico. Puesto que el grafo es conexo, existe un camino que vincule cualquier par de vértices del árbol libre; no obstante, si se retira un arco este dejaría de ser conexo. Por último, un grafo de esta índole, como todo árbol, posee un vértice distinguido que se denomina raíz.

Sea un grafo $G=(V,E)$ donde V es el conjunto de vértices del mismo y E el conjunto de arcos, y un árbol libre $T=(V',E')$; si $V'=V$ y $E'=E$ se dice que T es un árbol de recubrimiento mínimo si cumple que:

$$W(T)=\sum_{(u,v) \in E'} w(u,v) \quad (1)$$

es mínimo siendo $W(T)$ la función peso de un grafo. En la figura 3 se muestra un ejemplo de un grafo árboles de recubrimiento con diferentes costos que se pueden obtener del mismo.

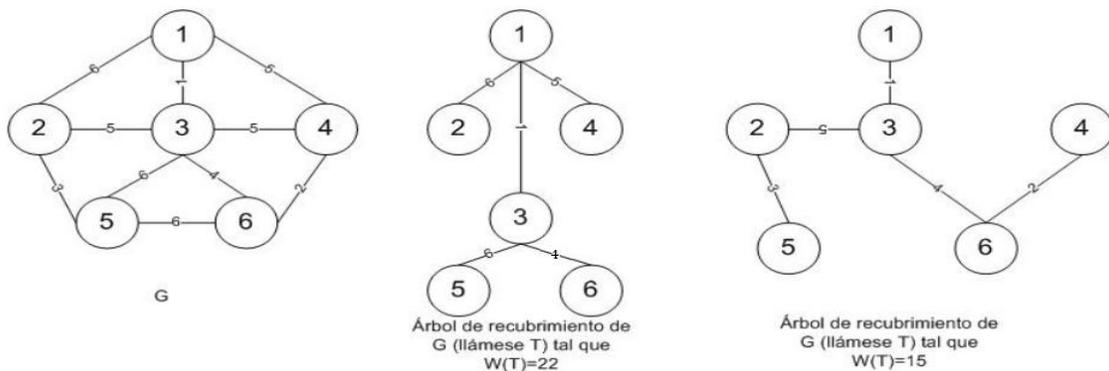


Figura 3

El algoritmo de Prim es un algoritmo que busca obtener el árbol de recubrimiento de costo mínimo para un grafo dado. Es un algoritmo Greedy y por tanto demanda un tiempo polinomial para entregar el resultado final.

Un algoritmo Greedy es aquel que, para resolver un determinado problema, sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. El algoritmo escoge en cada paso al mejor elemento $x \in C$ posible, conocido como el elemento más prometedor. Se elimina ese elemento del conjunto de candidatos

$$(C \leftarrow C \setminus \{x\})$$

y, acto seguido, comprueba si la inclusión de este elemento en el conjunto de elementos seleccionados $(S \cup \{x\})$ produce una solución factible.

En caso de que así sea, se incluye ese elemento en S . Si la inclusión no fuera factible, se descarta el elemento. Se itera el bucle, comprobando si el conjunto de seleccionados es una solución y, si no es así, se pasa al siguiente elemento del conjunto de candidatos.[1]

5.3.2 El algoritmo de Prim

El algoritmo de Prim requiere de dos componentes importantes:

- Un conjunto A de aristas que van pasando a formar parte de la solución.
- Un conjunto S de vértices de V que se va a ir completando conforme se vaya armando el árbol libre.

La estrategia consiste en ir añadiendo aristas (u,v) a A sin crear ciclos. Además, cada una de estas aristas debe producir el mínimo costo. Se puede empezar desde cualquier vértice (en el caso de estudio el árbol tendrá raíz en el punto en el que los camiones entren a la planta urbana del municipio). Los vértices que van siendo seleccionados se van agregando a S.

La iteración termina cuando la operación entre conjuntos V-S arroja como resultado el conjunto vacío. El árbol resultante, $T=(S,A)$, es el árbol de recubrimiento de mínimo costo.

5.3.3 Adaptación al Caso de Estudio

Se escogió la estructura de una matriz en lugar de la de un grafo para desarrollar la aplicación puesto que la disposición de las manzanas es bastante regular. No obstante se hace necesario identificar cuáles son los nodos y cuales son los vértices. Un vértice es una esquina de la ciudad. El identificador de los vértices no es un solo número como es el caso normal de un grafo sino un par de números que constituyen las coordenadas que uno utiliza normalmente para moverse sobre la propia matriz.

Un arco es una transición entre dos vértices, por ende se hace necesario especificar el vértice de origen y el de destino. Tanto vértice de origen como vértice de destino son identificados mediante dos números que son sus respectivas coordenadas sobre la matriz como ya se ha dicho. Por lo tanto, para identificar un arco se necesitarán de cuatro números.

En lo referente a los movimientos posibles desde cada estado es algo más simple pues de antemano siempre sabremos que son a lo sumo cuatro desde una posición dada sobre la matriz que es donde estaría el camión en un momento dado. Estas son: sumarle uno a las fila, restarle uno a la fila, sumarle uno a la columna o restarle uno a la columna. Por supuesto no se pueden hacer dos a la vez pues no se puede avanzar en diagonal sobre la matriz.[2]

Los arcos no se rotulan. Se asume que cada arco tiene un costo de 1. Dado dos puntos de la matriz, para calcular la distancia entre los mismos en término de arcos, o sea cuadras se utiliza el cálculo de la Distancia Manhattan como si se tratase de un par de puntos en el plano de ejes coordenados. esta se calcula de la siguiente forma:

$$D=|(Xd-Xo)+(Yd-Yo)| \quad (2)$$

5.4. Formato de los Resultados y su Análisis

Lo que hará el algoritmo de Prim, como ya se explicó, es devolver una secuencia de vértices (coordenadas en la matriz) que constituyen puntos que al unirlos se generará el árbol de recubrimiento de costo mínimo. Como se trata de un árbol y estos son estructuras fuertemente recursivas, recorrerlo supone utilizar una función recursiva que permita ir recorriendo la matriz teniendo presente la lista de puntos entregados por Prim. Una vez que dicha lista se ha vaciado se ha finalizado el recorrido y entonces se tendrá un posible recorrido. Un solo árbol de recubrimiento no supone un único recorrido, pues el orden en el que se visitan los nodos(esquinas del pueblo) puede variar; un árbol puede ser recorrido de varias formas. Lo que si no va a cambiar es el costo de recorrerlo pues al final los arcos involucrados siempre serán los mismos y en igual cantidad.

Una vez que se visita una esquina, la casilla en la matriz correspondiente a dicha esquina debe ser seteada a cero. Al final del módulo del recorrido la matriz debería terminar con todos ceros lo cual indicaría que toda la basura del pueblo ha sido levantada correctamente.

Puesto que la idea es utilizar dos camiones, y para simplificar la resolución del problema, la matriz se divide en dos partes una para cada camión. El primer camión recibirá una matriz con las mismas dimensiones que la primera pero con ceros en las partes que no le corresponde recorrer para que al algoritmo no le resulten atractivos los puntos de esa zona.

Asimismo, para emular el comportamiento en días feriados y en los Domingos, fechas en las que la producción de basura se incrementa, se multiplica las respectivas matrices con un factor que puede ser seteado fácilmente. [3]

Paralelamente al desarrollo del algoritmo de Prim, se creó un recorrido haciendo que los camiones vayan de lado a lado, dentro de sus respectivas matrices. A este algoritmo se lo conoce como Barrido, para los diferentes camiones de recolección. No es exactamente el recorrido actual de los camiones del municipio pero se aproxima en la adaptación. Por otro lado, en la implementación este algoritmo no considera las esquinas en las que se deposita basura, llegando a recorrer las zonas fuera de la planta urbana. Para esto, realizamos una corrección matemática, de 0.81, reduciendo así el trayecto que no debería realizar. Para más detalle véase la Figura 4.



Figura 4

Sean :

- P una coordenada perteneciente al conjunto de puntos con basura a recorrer
- S(P) una función determinística para la siguiente coordenada desde P
- B(P) la cantidad de basura en un punto P
- M el conjunto de puntos de la matriz

Entonces

$$P_0 \in M, S(P_0) = P_1$$

tal que

$$P_1 \in \{(P_0x, P_0y+1), (P_0x, P_0y-1), (P_0x+1, P_0y), (P_0x-1, P_0y)\}$$

y además

$$B(P_1) > 0$$

El algoritmo recibe como parámetros un mínimo y un máximo, los cuales se utilizan como cotas para el recorrido del camión. En caso del camión llegar a la capacidad máxima, se incrementa la distancia recorrida en base al costo de regresar al depósito y regresar al punto desde donde se llenó.

```

Para cada cuadra en vertical
  Recorrer desde min a máx
  Recolectar basura
  if camión lleno
    volver a descargar
  Fin if
Fin recorrer
Fin para
  
```

5.5 Resultados

5.5.1 Día de Semana

La ejecución del algoritmo de Prim y del de Barrido para días de semana arrojó como resultado lo que se muestra en la Figura 5.

```
Terminal - mati@portal: /var/www/html/CM12
File Edit View Terminal Tabs Help
---|PRIM|---
DIA DE SEMANA. . .
Calculando recorrido camión #1. . .
Saliendo camión #1 . . .
r:67 dist :83
Calculando recorrido camión #2. . .
Saliendo camión #2 . . .
r:63 dist :73
---|BARRIDO|---
DIA DE SEMANA. . .
Saliendo camión #1 . . .
recorrió:93
Saliendo camión #2 . . .
recorrió:79
FINALIZO EL DIA DE SEMANA.
```

Figura 5

En el resultado r es lo que recorre cada camión y la distancia real sumándole las cuadras que debe recorrer para volver al principio. Esto es necesario ya que al recorrer el árbol entregado por Prim no se termina necesariamente en el mismo punto en el que se partió. Barrido considera el recorrido completo desde un punto inicial hasta uno final.

El camino que va recorriendo Prim para el camión 1 es el siguiente:

{8,8},{7,8},{6,8},{5,8},{4,8},{3,8},{2,8},{1,8},{0,8},{0,7},{1,7},{2,7},{3,7},{4,7},{5,7},{6,7},
{7,7},{8,7},{9,7},{9,6},{8,6},{7,6},{6,6},{5,6},{4,6},{3,6},{2,6},{1,6},{0,6},{0,5},{1,5},{2,5},
{3,5},{4,5},{4,4},{3,4},{2,4},{2,3},{1,3},{1,4},{0,4},{0,3},{0,2},{0,1},{1,1},{1,0},{2,0},{2,1},
{2,2},{3,2},{3,3},{4,3},{4,2},{5,2},{5,3},{5,4},{5,5},{6,5},{6,4},{7,4},{7,5},{8,5},{8,4}



Figura 6

Visto en la matriz, quedaría como se muestra en la ilustración 6.

Para el camión 2 el recorrido a seguir sería:

{9,12},{9,11},{9,10},{9,9},{8,9},{7,9},{6,9},{5,9},{4,9},{3,9},{2,9},{1,9},{0,9},{0,10},{1,10},
 {2,10},{3,10},{4,10},{5,10},{6,10},{7,10},{8,10},{8,11},{7,11},{6,11},{5,11},{4,11},{3,11},
 {2,11},{1,11},{0,11},{0,12},{1,12},{2,12},{3,12},{4,12},{5,12},{6,12},{6,13},{5,13},{4,13},
 {3,13},{2,13},{1,13},{0,13},{0,14},{1,14},{1,15},{0,15},{0,16},{1,16},{2,16},{2,15},{2,14},
 {3,14},{4,14},{5,14},{6,14},{7,14},{7,13},{7,12},{8,12},{8,13}

el cuál se puede apreciar con mayor facilidad en la Figura 7 Ninguno de los dos recorridos tiene forma de un árbol como uno se lo imagina normalmente, pero no está mal. Una lista es una especialización de un árbol, así como un árbol es una especialización de un grafo.



Figura7

5.5.2 Día de Fin de Semana (o Feriado)

El comportamiento de los camiones en días de fin de semana no difiere mucho salvo por el hecho de que como los camiones se terminan llenando se hace necesario realizar más de una vuelta. Los camiones tienen una capacidad de 4 toneladas, o sea 4000000 de gramos. Si la basura que se va llevando excede la capacidad máxima capacidad de basura del camión, el camión debe volver a la base y eso supone incrementar en número de cuadras a recorrer. Más a allá de esto el resultado será el mismo que antes solo que más acentuado; si con Prim se recorren menos cuadras que con Barrido un día de semana, un día de fin de semana se recorrerán muchas menos cuadras con Prim respecto a Barrido.

Para el caso del camión 1 el recorrido es :

{9,8},{9,7},{9,6},{9,5},{9,4},{8,4},{7,4},{6,4},{5,4},{5,3},{5,2},{4,2},{3,2},{2,2},{2,1},{2,0},
 {1,0},{0,0},{0,1},{1,1},{1,2},{0,2},{0,3},{1,3},{2,3},{3,3},{4,3},{4,4},{3,4},{2,4},{1,4},{0,4},
 {0,5},{1,5},{2,5},{3,5},{4,5},{4,6},{4,7},{5,7},{5,6},{5,5},{6,5},{7,5},{8,5},{8,6},{8,7},{8,8},
 {7,8},{7,7},{7,6},{6,6},{6,7},{6,8},{5,8},{4,8},{3,8},{3,7},{3,6},{2,6},{2,7},{2,8},{1,8},{1,7}, {1,6},{0,6},{0,7}

y para el camión 2 es:

{9,12},{9,11},{9,10},{9,9},{8,9},{7,9},{6,9},{5,9},{4,9},{3,9},{2,9},{1,9},{0,9},{0,10},{1,10},
 {2,10},{3,10},{4,10},{5,10},{6,10},{7,10},{8,10},{8,11},{7,11},{6,11},{5,11},{4,11},{3,11},
 {2,11},{1,11},{0,11},{0,12},{1,12},{2,12},{2,13},{1,13},{0,13},{0,14},{1,14},{1,15},{0,15},
 {0,16},{1,16},{2,16},{2,15},{2,14},{3,14},{3,13},{3,12},{4,12},{5,12},{5,13},{4,13},{4,14},
 {5,14},{6,14},{7,14},{7,13},{6,13},{6,12},{7,12},{8,12},{8,13}

Los recorridos de los camiones 1 y 2 los días feriados se muestran más claramente en las ilustraciones 8 y 9 respectivamente. Difieren en el orden tal y como se esperaba, pero los puntos visitados son los mismos para cada caso. En el caso se probó con un factor de 12 (recuérdese que para los fines de semana se trabaja con la misma matriz de los días de semana pero multiplicadas por un factor representativo que marca el incremento en la producción de basura), ya que según fuentes del municipio se hacen necesario hasta 12 viajes los días Domingos ya que los días Sábados no se recogen residuos.



Figura 8



Figura 9

El recorrido que realizan los camiones siguiendo el algoritmo para control de barrido puede apreciarse en las Figuras 10 y 11.



Figura 10



Figura 11

5.5.3 Análisis comparativo de los resultados obtenidos.

Para el día de semana, se puede apreciar que se termina recorriendo más con barrido que con Prim para los días de semana. El ahorro en distancia anda en torno a las 15 cuadras lo cuál no es poca cosa.

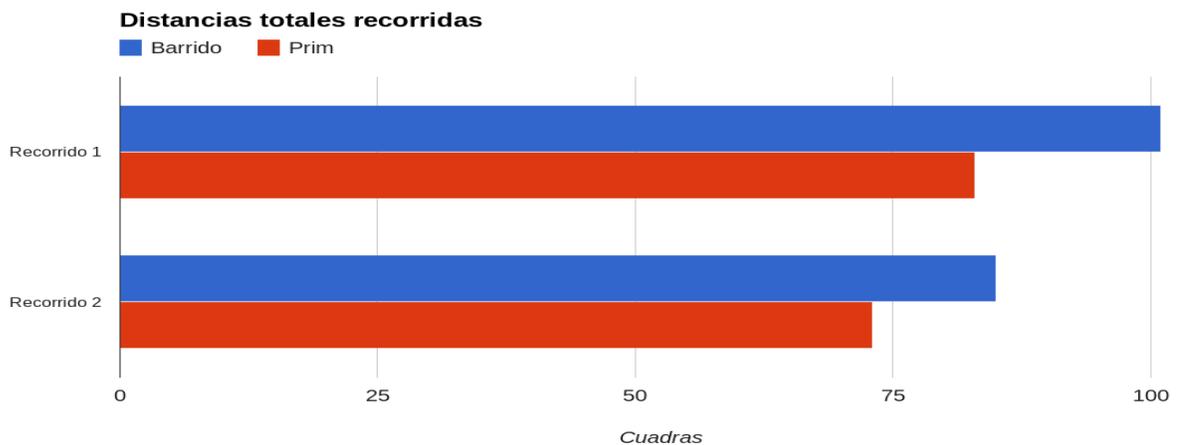


Figura 12

En la caso de fin de semana, las diferencias son aún más marcadas. Recuérdese que para simular el comportamiento de los fines de semana se multiplicaba la cantidad de basura que exhibía la matriz normal por 12. El incremento puede deberse, probablemente, a que ya no basta con una vuelta sino que se hace necesario realizarás de una para juntar la totalidad de los residuos.

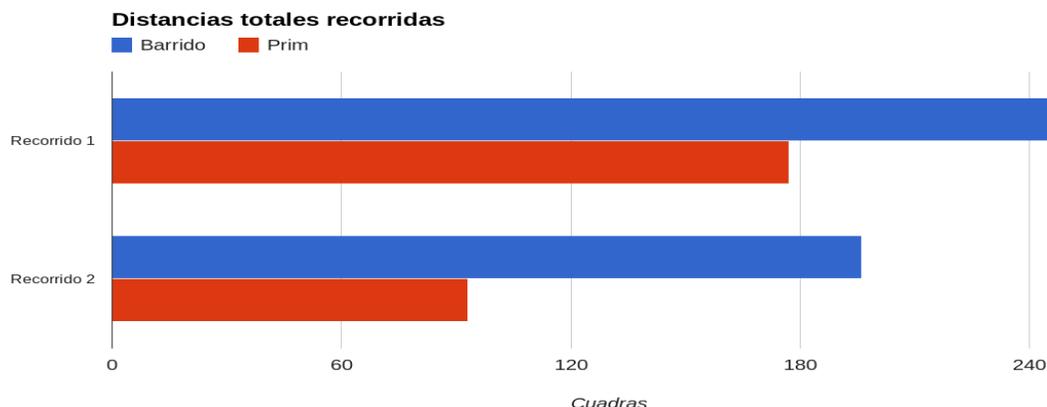


Figura 13

6. CONCLUSIONES

Las distancias que los camiones deben recorrer caminos elaborados por Prim son menores que las que deben recorrer con Barrido.

Existen otros algoritmos que permiten obtener árboles de recubrimiento de mínimo costo como lo es Kruskal. Sin embargo, está probado que los resultados entregados han de ser similares. Más allá del método utilizado, si el algoritmo en cuestión es óptimo, el resultado siempre será el mismo: el óptimo; árbol de recubrimiento de costo mínimo hay uno solo.

El problema se pudo haber visto como un grafo con arcos rotulados, siendo el rótulo la cantidad de basura de la cuadra. Cada nodo al igual que en la estrategia presentada en este trabajo representaría una esquina del pueblo; sin embargo, ahora en las esquinas ya no estaría la basura sino que esta estaría en los arcos como ya se ha dicho. En este contexto ya no tendría sentido utilizar la distancia Manhattan pues no se buscaría minimizar la distancia que se que se recorre sino maximizar la basura que se recoge.

Del análisis de varias posibles formas de resolver un mismo problema suelen salir las mejores formas de resolverlo.

No obstante, el enfoque matricial del problema tenía sus ventajas. En primera instancia, como ya se ha dicho, la disposición de las cuadras es algo bastante regular, como ocurre en muchas ciudades hispanoamericanas y se asemejan mucho más a la forma que tiene una matriz. La estrategia del grafo supone juntar la basura del lugar en el que se encuentra mientras que, la de la matriz supone recogerla de las esquinas. Ninguna de las dos estrategias se condice exactamente con lo que ocurre en la realidad. Este tema de la ubicación también podría ser visto como un pro del enfoque de la matriz pues juntar la basura en las esquinas, si es por poco tiempo y de forma organizada.

Queda abierta la posibilidad de continuar indagando alternativas de optimización.

7.REFERENCIAS

- [1] CHARTRAND, G y ZHANG.(2005).*Introduction to Graph Theory*. McGraw-Hill. 1. New York.
- [2] THULASIRAMAN,K. y SWAMY, M. N. S.(1992). *Graphs: Theory and Algorithms*. JOHN WILEY & SONS, INC. Concordia University. Montreal, Canadá.
- [3] CHARTRAND, Gary LESNIAK, Linda y ZHANG, Ping.(2011). *Graphs & Digraphs, Fifth Edition*. Chapman & Hall/CRC. United States of America.