

Ruteo de vehículos escolares: caso de estudio de la Universidad Autónoma de Coahuila

O. Avalos-Rosales¹, M. Juárez-Peláez², Y. Cardona-Valdés^{1*}

¹Centro de Investigación en Matemáticas Aplicadas

Universidad Autónoma de Coahuila

² Facultad de Sistemas, Universidad Autónoma de Coahuila

* Correo electrónico: y.cardona@uadec.edu.mx

Resumen

En este trabajo se proponen un conjunto de nuevas rutas de transporte para dar servicio a la comunidad estudiantil actual del Campus Arteaga de la Universidad Autónoma de Coahuila atendiendo a la dispersidad de los estudiantes en la zona metropolitana de Saltillo. Se analiza un caso de estudio con la información de una Facultad, considerando 380 estudiantes y 46 paradas correspondientes a las colonias de Saltillo. El problema se formula matemáticamente como un modelo lineal entero mixto, el cual es resuelto a través de un optimizador comercial. Con el modelo sólo es posible resolver a optimalidad instancias de 10 paradas y 50 estudiantes, por lo que se diseñó un algoritmo heurístico constructivo híbrido basado en el algoritmo de Clarke y Wright, en donde la factibilidad de una solución se verifica al resolver una versión simplificada del modelo matemático. Se implementan dos estrategias para reducir el tiempo de cómputo del algoritmo heurístico y se reportan las nuevas rutas propuestas, así como algunas recomendaciones para facilitar la construcción de las instancias del problema.

Palabras clave: Ruteo de vehículos escolares; Caso de estudio; Programación entera mixta; Heurísticos.

Introducción

La Universidad Autónoma de Coahuila en el año 2010 inauguró un nuevo campus en el municipio de Arteaga Coahuila, el cual se encuentra a 15.7 kilómetros de Saltillo, Ciudad capital del Estado de Coahuila. Ya que el transporte público era muy limitado, en el año 2011, la Universidad implementó el servicio de transporte universitario (Lobús) para trasladar a los usuarios del campus Arteaga con el municipio de Saltillo. Dicho servicio constaba de sólo dos autobuses y era suficiente para cubrir la demanda de los usuarios, ya que hasta ese momento solo se había instalado la Facultad de Sistemas. Sin embargo, el servicio

original se componía de una sola ruta con una cobertura muy limitada.

En los últimos años, la población estudiantil y el número de usuarios se han incrementado al establecerse en el campus Arteaga además de la Facultad de Sistemas otras tres Facultades. Actualmente, la cantidad de estudiantes es de 3072 y se estima que 2500 son usuarios del Lobús. Sin embargo, a la fecha la Universidad sólo ha incrementado el número de autobuses (de dos a cuatro) para dar el servicio, pero manteniendo la misma ruta y paradas que en el diseño original.

Cabe mencionar que los estudiantes viven en zonas alejadas de las paradas oficiales por las que pasa el transporte universitario, lo que hace no-

tar que la ruta y las paradas existentes sean insuficientes para la creciente demanda y la dispersidad de la comunidad Universitaria en la zona metropolitana de Saltillo, por lo que urge un plan en el rediseño de las rutas escolares que tome en cuenta ambas problemáticas (ver Figura 1).



Figura 1: Dispersidad de las colonias de los estudiantes de la Facultad de Sistemas de la base de datos del 2014.

La problemática planteada se ha estudiado en la literatura científica como un problema de ruteo de vehículos escolares (SBRP por sus siglas en inglés) el cual involucra el transporte de estudiantes desde las paradas de autobús hacia sus escuelas. El objetivo es diseñar rutas optimizando una función objetivo de interés. El SBRP al ser una variante del bien conocido problema de ruteo de vehículos (VRP) se clasifica como un problema NP-hard [4].

En la literatura científica se han propuesto formulaciones matemáticas y algoritmos heurísticos y metaheurísticos para resolver el problema [5, 6] y [3].

El SBRP se descompone en 5 subproblemas:

1. Preparación de los datos
2. Selección de paradas de autobús
3. Generación de rutas
4. Ajuste a los horarios de la escuela

5. Programación de las rutas

La propuesta de solución que consideramos en este trabajo se enfoca en los tres primeros subproblemas. Donde, además se considera el problema de selección de paradas y generación de rutas de manera simultánea.

Descripción del problema

El SBRP considera un conjunto potencial de paradas y un conjunto de estudiantes, donde cada estudiante puede acceder a una o más paradas, expresado a través de lo que denominaremos matriz de accesibilidad. Se dispone de una flota homogénea de autobuses con capacidad conocida. Todas las rutas comienzan en una estación de autobuses central, recorren un conjunto de paradas y culminan en la escuela. Se debe determinar cuáles paradas serán visitadas por los autobuses escolares, así como en qué parada debe recogerse cada estudiante y el conjunto de rutas que minimicen el costo total de recorrido de todos los autobuses. Dicho costo podría ser la distancia de recorrido, el tiempo de recorrido o algún otro.

Metodología de solución

Para abordar el problema se implementan dos metodologías tomando como base las propuestas por Schittekat et al. (2013) [5], una es un modelo matemático de programación lineal entera mixta (MILP por sus siglas en inglés) y la otra un algoritmo heurístico. Para ambas metodologías se consideran las siguientes suposiciones: 1) se dispone de suficientes autobuses para cubrir la demanda, 2) una parada sólo será visitada por un autobús, esto significa que el número de estudiantes por parada no puede exceder la capacidad del autobús, 3) todos los autobuses tienen la misma capacidad, 4) un autobús sólo puede realizar una ruta, y 5) cada estudiante cuenta como una unidad de demanda que nos sirve para validar que no se exceda la capacidad del autobús.

Dada la complejidad del problema, utilizando el modelo matemático sólo fue posible resolver a optimalidad instancias de 10 paradas con 50 estudiantes. Para más detalles sobre el modelo y su desempeño ver [2]. Para instancias de mayor tamaño se implementó un algoritmo heurístico del tipo constructivo basado en el método de los ahorros de Clarke-Wright, que se describe en la siguiente sección.

Algoritmo constructivo

La idea central del algoritmo consiste construir una solución donde cada parada se visita en una ruta independiente, luego ir uniendo pares de rutas al conectar la última parada de una ruta con la primer parada de la otra ruta, verificar si para dicha unión es posible asignar los estudiantes de tal forma que no se exceda la capacidad de ningún autobús y seguir uniendo rutas hasta que ya no sea posible unir ningún par de ellas. Para ello se crean matrices auxiliares que permitirán seleccionar el par de paradas a unir. El algoritmo se esquematiza diagrama de flujo de la Figura 2.

De manera inicial, el algoritmo construye una **solución tipo estrella** en donde se establece una ruta independiente por cada parada (ver Figura 3). En caso de que la solución no sea factible el algoritmo termina su ejecución.

Si la solución tipo estrella resulta factible, se procede al proceso de unión de rutas. Para ello se calcula por única ocasión la **matriz de ahorros** aplicando la fórmula $\sigma_{ij} = c_{ij} - c_{i(np+1)} - c_{0j}$, la cual es una modificación de la fórmula propuesta por Clarke y Wright [1] (ver Figura 4). Donde np representa el número de paradas, c_{ij} representa el costo de ir de la parada i a la parada j y, σ_{ij} representa el ahorro que resultaría al unir dos rutas, tal que i es la última parada de la primer ruta y j es la primer parada de la segunda ruta.

Luego calculamos la **matriz de conectables** que define cuáles son los pares de paradas posibles que se podrían conectar. Ésta es una matriz binaria donde un elemento toma el valor de 1

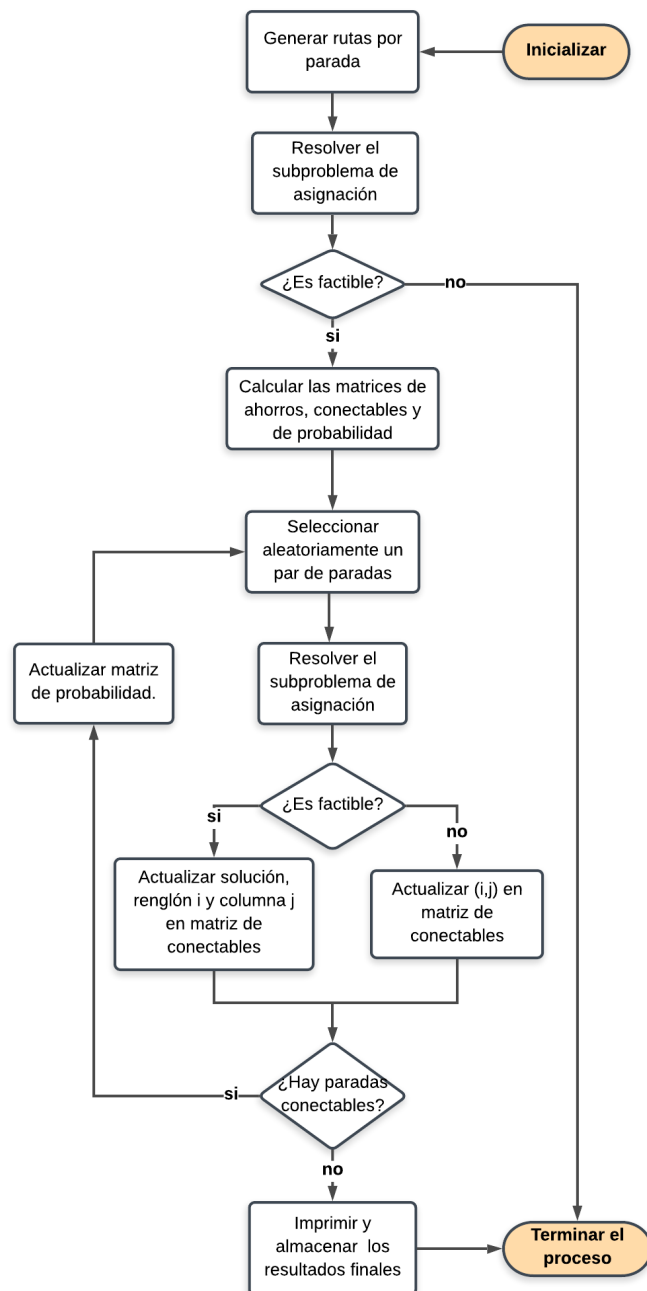


Figura 2: Diagrama de flujo del algoritmo heurístico

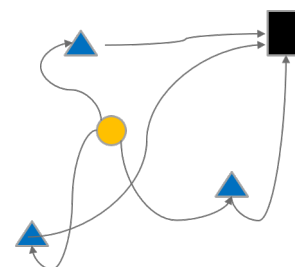


Figura 3: Solución tipo estrella. Depósito (amarillo), paradas (azul), escuela (negro)

si un par de paradas son conectables y 0 en caso contrario. En la solución inicial tipo estrella todas las paradas son conectables con todas las

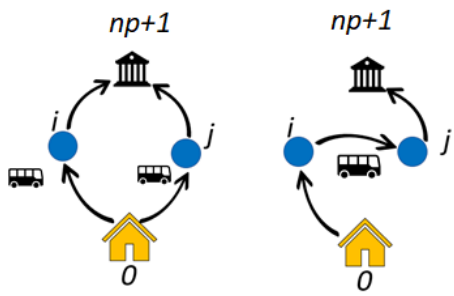


Figura 4: Modificación algoritmo Clarke y Wright. Depósito en color amarillo (nodo 0), paradas en color azul (i y j), escuela en color negro ($np + 1$).

demás paradas.

Enseguida calculamos la **matriz de probabilidad** que nos permite establecer prioridades en cuanto a la selección de paradas a conectar, donde la probabilidad de conectar cada par de paradas i y j es proporcional a su ahorro. Esta probabilidad se calcula como el cociente del costo de conectar dichas paradas, sobre la suma de los costos de todas las paradas que aún son conectables (de acuerdo a la matriz de conectables).

Entonces llevamos a cabo el proceso de **selección de ruleta**, el cual consiste en generar un número aleatorio entre 0 y 1, e ir sumando las probabilidades de cada elemento en la matriz de probabilidades, hasta que la suma iguale o exceda al número aleatorio generado, identificando así las paradas seleccionadas para conectarse y, en consecuencia, las rutas a que estas paradas pertenecen.

Obtenemos la **solución candidata** al modificar las dos rutas seleccionadas: pasamos todas las paradas de la segunda ruta al final de la primera ruta, y dejamos vacía la segunda ruta. Entonces se valida la factibilidad de dicha solución candidata al resolver el **subproblema de asignación** de estudiantes a autobuses (siguiente sección).

Si la unión de rutas genera una solución factible, se actualizan la solución, la asignación de estudiantes a autobuses, la función objetivo y la matriz de conectables. En la matriz de conectables se ponen como cero todos los elementos del

renglón i y de la columna j . Si la unión de rutas no da solución factible sólo se pone como cero el elemento (i, j) de la matriz de conectables y se desecha la solución candidata.

Si aún hay pares de paradas por conectar se recalcula la matriz de probabilidades y se regresa al proceso de selección de ruleta. En caso contrario, se guarda la información relevante y el algoritmo termina.

Subproblema de asignación de estudiantes a autobuses

Dado que cada estudiante puede acceder a una o más paradas, la demanda de una parada no se determina de forma directa y es necesario resolver un subproblema de asignación de estudiantes a autobuses para verificar si para un conjunto de rutas dado, existe una asignación factible al problema, es decir, que respete la capacidad de los autobuses.

Para ello, se calcula la matriz de **costo de asignación** de estudiantes a autobuses y se resuelve el subproblema que minimice el costo asignar de estudiantes a autobuses.

El modelo del subproblema de asignación se establece como sigue:

$$\text{minimizar } \sum_{l \in S} \sum_{k \in K} t_{kl} z'_{kl} \quad (1)$$

sujeto a:

$$\sum_{k \in K} z'_{kl} = 1; \quad \forall i \in S \quad (2)$$

$$\sum_{l \in S} z'_{kl} \leq Q; \quad \forall k \in K \quad (3)$$

$$z'_{kl} \in \{0, 1\} \quad \forall k \in K, l \in S \quad (4)$$

Las restricciones (2) aseguran que cada estudiante l es asignado a exactamente una ruta k . Las restricciones (3) garantizan que la capacidad de los autobuses no se exceda y, por último, las restricciones (4) establecen la naturaleza binaria de las variables, sin embargo éstas se pueden relajar ya que al resolver la asignación de estudiantes

a las rutas, se está resolviendo un problema de transporte y es sabido que en este tipo de problemas si el suministro y la demanda son enteros, la solución será entera.

El modelo anterior también fue planteado por Schittekat et al. (2013) [5], sin embargo ellos optaron por usar el bien conocido *primal-dual labeling method*. En nuestro heurístico sí se resuelve de forma exacta con el modelo matemático, al llamar internamente al optimizador CPLEX.

Asignación final de estudiantes a paradas

De la heurística anterior se obtienen las rutas de los autobuses, es decir, la asignación y orden en que cada autobús debe visitar las paradas que le fueron asignadas. Sin embargo, no se sabe con exactitud cuál es la asignación de estudiantes a las paradas; y si es necesario visitarlas todas.

La asignación de estudiantes a paradas la hacemos a través de una matriz binaria P , el valor 1 indica que un estudiante l es asignado a una parada i y cero en otro caso. La matriz se inicializa en ceros y asignaremos un estudiante a la vez, primero el estudiante 1, 2, 3, etc.

- Para cada estudiante l buscaremos cual fue la ruta k^* a la que fue asignado.
- Usando la ruta k^* y la matriz de accesibilidad (de los parámetros del problema) determinamos cuál es la primer parada i^* en la ruta k^* a la que puede llegar caminando y lo asignamos a ella.

Con esto hemos determinado las rutas de los autobuses, es decir, la asignación de paradas a autobuses y el orden en que se visitan. También se ha determinado la asignación de estudiantes a paradas.

Validación de la metodología

Adicional a la instancia generada para el caso de estudio, para determinar el alcance del modelo matemático y del heurístico presentado en las

No.	Instancia	MILP		Heurístico	
		FO	Tiempo CPU	Desv	Tiem.
1	5p25e25q5w	332.63	0.06	20.9	0.92
2	5p25e50q5w	332.63	0.05	20.9	1.86
3	5p25e25q10w	374.91	0.06	12.1	2.35
4	5p25e50q10w	374.91	0.06	9.8	4.91
5	5p25e25q20w	362.36	0.09	26.1	4.71
6	5p25e50q20w	362.36	0.09	30.9	5.84
7	5p25e25q40w	361.17	0.09	17.1	5.99
8	5p25e50q40w	361.17	0.11	30.7	6.64
9	5p50e25q5w	638.29	0.07	17.3	5.35
10	5p50e50q5w	326.6	0.09	3.0	1.51
11	5p50e25q10w	611.3	0.06	7.3	5.44
12	5p50e50q10w	402.88	0.08	17.2	4.31
13	5p50e25q20w	622.68	0.10	0.0	3.99
14	5p50e50q20w	293.56	0.28	32.1	2.04
15	5p50e25q40w	433.00	0.25	8.9	5.09
16	5p50e50q40w	274.39	0.15	22.8	1.24
17	5p100e25q5w	1306.41	0.03	0.0	8
18	5p100e50q5w	831.03	0.07	8.6	3.6
19	5p100e25q10w	1245.28	0.02	0.0	8.26
20	5p100e50q10w	616.69	0.06	0.0	2.92
21	5p100e25q20w	998.82	0.09	0.0	6.79
22	5p100e50q20w	710.10	0.21	13.9	4.17
23	5p100e25q40w	711.73	0.86	6.7	7.42
24	5p100e50q40w	457.26	0.73	12.6	3.06
25	10p50e25q5w	733.41	20.33	19.0	9.52
26	10p50e50q5w	555.15	11.93	30.9	2.58
27	10p50e25q10w	747.57	70.30	34.1	10.12
28	10p50e50q10w	595.00	1.75	39.0	7.15

Tabla 1: Resultados comparativos entre el MILP y el algoritmo heurístico para las primeras 20 instancias

secciones anteriores también se construyeron un conjunto de instancias pseudoaleatorias.

Se generaron 14 grupos de instancias con diferente número de paradas (p) y estudiantes (e), cada grupo se conforma de 8 instancias en las que la capacidad del autobús (q) varía en 25 ó 50, y la distancia que el estudiante puede caminar a una parada (w) varía entre 5 y 40 unidades. Para más detalles sobre la construcción de las instancias ver [2].

La nomenclatura utilizada se ejemplifica a continuación con la instancia 5p25e25q5w, con 5 paradas potenciales, 25 estudiantes, la capacidad del autobús en 25 unidades, y finalmente, la distancia que el estudiante puede caminar hacia una parada es de 5 unidades.

La experimentación se llevó a cabo en una estación de trabajo WorkStation HP Z440, Intel Xeon, RAM 8GB, 1 TB. El modelo matemático

se implementó en Visual C++ usando CPLEX 12.6 con Concert Technology y estableciendo un tiempo límite de ejecución de 7200 segundos.

En la Tabla 1 se muestran los resultados para las primeras 28 instancias, donde el MILP fue capaz de obtener soluciones óptimas. En la columna 2 se muestran los resultados correspondientes al modelo, indicando el valor de la función objetivo (FO) y el tiempo (Tiempo) en segundos. En la columna 3 se muestran los resultados del algoritmo heurístico, indicando el valor de la desviación con respecto al óptimo (Desv) y el tiempo (Tiempo CPU) en segundos.

Cabe destacar que el heurístico es capaz de obtener soluciones que coinciden con el óptimo del modelo, sin embargo en otros casos obtienen soluciones que se desvían hasta un 34.1 % del costo de las soluciones óptimas.

Reducción de tiempos

En la experimentación computacional realizada con el heurístico se notó que el tiempo computacional resultó ser muy elevado, por lo que se diseñaron dos estrategias para reducirlo. Ambas estrategias modifican el criterio de paro, el cual actualmente consiste detener el proceso de unión de rutas si no hay más pares de paradas por conectar. Las estrategias son las siguientes:

1. Detener el proceso de unión de rutas si la cantidad de veces que de forma consecutiva ésta ha resultado infactible, es menor que una predeterminada cantidad de veces (umbral).
2. Detener el proceso de unión de rutas si la cantidad de rutas actuales es igual a la cantidad mínima de autobuses requeridos.

La Figura 5 muestra los tiempos de ejecución CPU en segundos, para las instancias de la 49 a la 112, para distintos valores umbrales (10, 20, 30 y 40) comparados con los tiempos sin Umbral.

Se encontró que con todos los umbrales se pierde muy poco en la calidad de las soluciones

obtenidas respecto a la versión sin umbral (menos de 2.66 %) y además se obtiene una reducción de tiempos importante. Se eligió un umbral de 30 por ser el que ofrece los mejores valores (0.6 % de pérdida en calidad, y tiempo promedio de ejecución de 27.77seg). La segunda estrategia incrementa ligeramente el tiempo computacional respecto a la primera, por lo que al final no fue considerada en la versión final del algoritmo.

Caso de estudio

Para el caso de estudio de la UAdeC, los datos de entrada a considerar son: el número de estudiantes, el número de paradas, el número de autobuses, la capacidad de los autobuses, las coordenadas de ubicación de los estudiantes, de las paradas, del depósito de autobuses, y de la escuela, así como la máxima distancia que puede caminar un estudiante a una parada. A partir de dichos datos se determinan los siguientes parámetros para construir una instancia del SBRP en estudio:

- Matriz de accesibilidad de estudiantes a paradas
- Matriz de costos entre paradas

Para el caso de estudio se consideró la base de datos (BD) del año 2014, proporcionada por la Facultad de Sistemas, dicha base consta de 885 alumnos de los cuales 736 indicaron una dirección de la ciudad de Saltillo y 149 se registraron como foráneos.

Para determinar las paradas potenciales se tomó en cuenta a todas aquellas colonias que tienen registrado en la BD al menos 4 estudiantes, ya que se busca cubrir el mayor número de demanda de los estudiantes que se encuentran dispersos geográficamente por la ciudad de Saltillo y sean trasladados al Campus Arteaga (ver Tabla 2).

Con todas estas consideraciones la instancia a trabajar considera: 46 paradas (una por cada

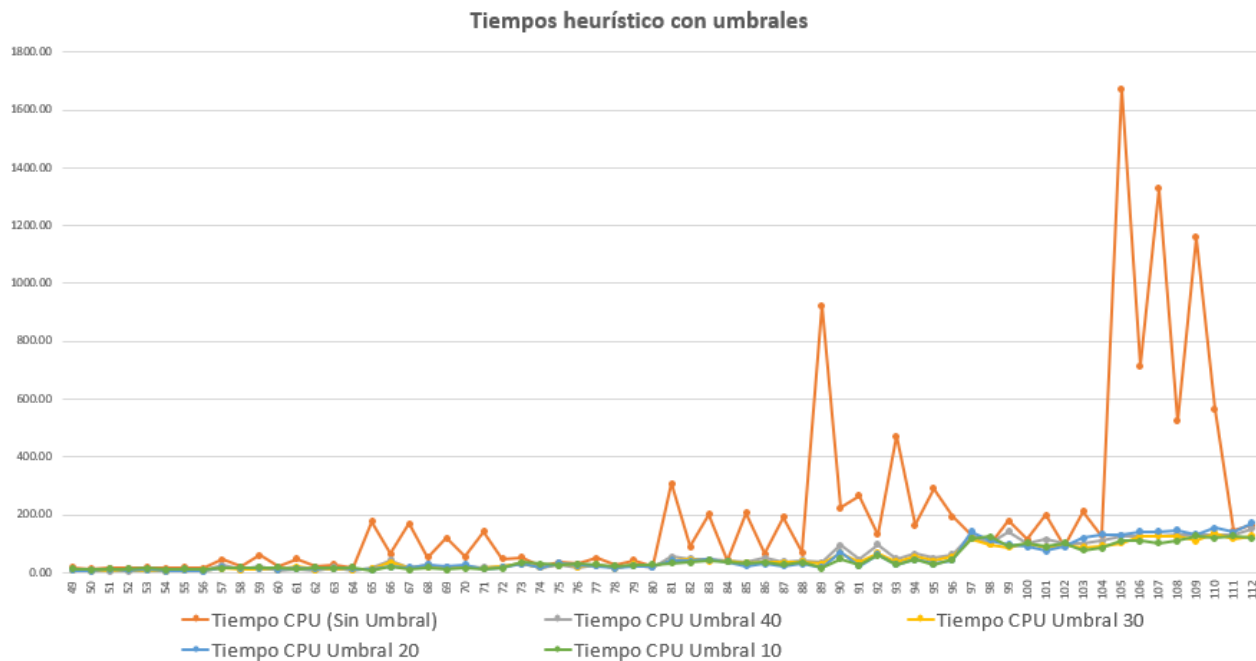


Figura 5: Comparativo del tiempo de ejecución para algunas las instancias 49 a 112 para distintos valores umbrales (Fuente: Elaboración Propia)

colonia), 380 estudiantes y la capacidad del autobús de 50. Para determinar la matriz de accesibilidad que representa la distancia caminando de los estudiantes a las paradas, no se consideraron las distancias reales caminando de los estudiantes a cada parada debido al tiempo que conlleva el cálculo de las mismas; en su lugar, se consideró que un estudiante únicamente puede llegar caminando a la parada que se encuentra en su colonia. Las coordenadas de una parada (colonia) se obtuvieron al introducir el nombre de la colonia en GoogleMaps y registrar la latitud y longitud que éste reporta por defecto.

A partir de dichos datos, se construyeron tres instancias variando la matriz de costos entre paradas. La primera instancia considera la matriz de costos como la matriz de distancia euclidiana, la segunda considera la matriz de costos como la matriz de distancias más corta entre calles entre cada par de paradas, y la tercera considera la matriz de costos como la matriz de menor tiempo de traslado entre cada par de paradas. Las dos últimas matrices fue posible obtenerlas mediante el uso del Software NEVA.¹ Al variar en cada

instancia la matriz de costos se varía la función objetivo, para las primeras dos se minimiza la distancia y para la tercera el tiempo de recorrido.

Rutas propuestas

La Figura 6 muestra para cada una de las tres instancias consideradas para el caso de estudio, el valor de la función objetivo (FO), el tiempo de ejecución del algoritmo (CPU en segundos) y las rutas generadas. Las primeras dos instancias minimizan la distancia en kilómetros (km) y la tercera minimiza el tiempo en minutos (min). Las rutas generadas se componen por un conjunto de paradas, el cero que denota la salida del depósito de autobuses y el 47 denota el destino, es decir, el Campus Arteaga.

Las Figuras 7 y 8 muestran las nueve rutas generadas para la instancia cuya matriz de costos es la matriz de tiempos obtenida a través de NEVA.

¹emplea ficheros shape para cargar cartografía digitalizada.

¹NEVA es un Sistema de Información Geográfica desarrollado por el Dr. Joaquín A. Pecheo (Universidad de Burgos, España), que

No.	Colonia	#Est.	No.	Colonia	#Est.	No.	Colonia	#Est.
1	Zona Centro	42	17	Brisas Poniente	7	33	Privadas la Torre	5
2	Mirasierra	21	18	Nuevo Mirasierra	7	34	San Ramón	5
3	Morelos	17	19	Valle de las Torres	7	35	Satélite Norte	5
4	Zaragoza	17	20	Bonanza	6	36	Topochico	5
5	Amp. Morelos	15	21	Cerro del Pueblo	6	37	Buitres	4
6	Bella vista	13	22	Chamizal	6	38	Europa	4
7	Satélite Sur	13	23	La Fragua	6	39	Girasol	4
8	Ignacio Zaragoza	12	24	Landín	6	40	Jardines Coloniales	4
9	Vista Hermosa	12	25	Lomas de Lourdes	6	41	La Aurora	4
10	Fundadores	11	26	Oceanía Boulevares	6	42	La Minita	4
11	Vicente Guerrero	11	27	Chapultepec	5	43	Pueblo Insurgentes	4
12	Saltillo 2000	10	28	Colinas Del Sur	5	44	República Oriente	4
13	Amp. 23 de Noviembre	9	29	Guayulera	5	45	Santa Cristina	4
14	Las Teresitas	9	30	Miguel Hidalgo	5	46	Zapaliname	4
15	Amp. 26 de Marzo	8	31	Oceanía	5			
16	Amistad	7	32	Provivienda	5			

Tabla 2: Colonias consideradas en la instancia del caso de estudio

Figura 6: Análisis de la instancia real $46p380e50q40w$ (Fuente: Elaboración Propia)

Instancia	FO	Tiempo CPU (s)	Ruta
$46p380e50q40w$ Matriz euclidiana Google Maps	511.469 km	68.199	R1: 0 1 47 R6: 0 7 18 43 30 26 47 R7: 0 8 27 9 35 47 R10: 0 11 36 6 25 5 47 R13: 0 14 2 39 38 47 R14: 0 15 29 10 45 16 28 19 47 R30: 0 31 46 21 33 37 3 22 47 R39: 0 40 12 32 23 4 44 47 R40: 0 41 13 17 20 24 42 34 47
$46p380e50q40w$ Matriz de distancias NEVA	425.610 km	361.806	R2: 0 3 27 40 4 44 47 R13: 0 14 8 47 R16: 0 17 5 30 33 34 19 37 47 R19: 0 20 36 10 39 18 47 R21: 0 22 1 47 R30: 0 31 38 26 12 11 28 35 47 R31: 0 32 15 13 7 21 23 47 R42: 0 43 6 41 2 47 R45: 0 46 16 24 25 9 42 29 45 47
$46p380e50q40w$ Matriz de tiempos NEVA	697.47 min	69.501	R0: 0 1 15 47 R2: 0 3 44 12 40 18 47 R19: 0 20 43 2 47 R23: 0 24 27 7 16 31 23 30 47 R24: 0 25 19 17 5 45 47 R27: 0 28 14 6 29 38 26 47 R34: 0 35 36 32 10 11 34 22 47 R36: 0 37 8 33 4 41 47 R45: 0 42 39 9 21 13 47

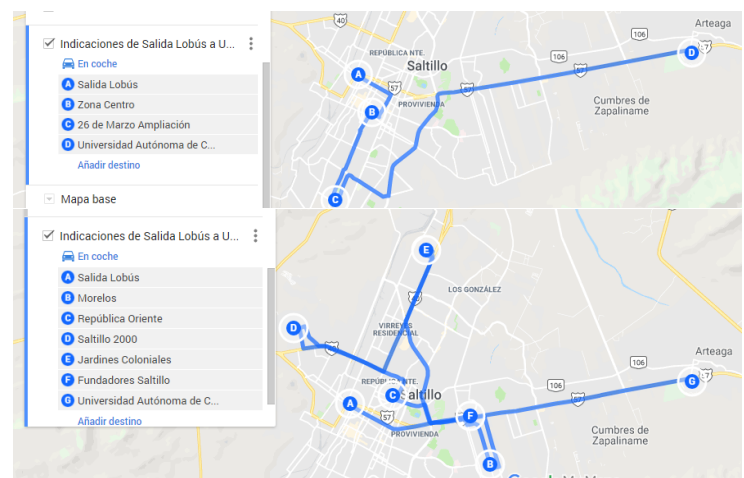


Figura 7: Rutas R0 y R2 generadas con la matriz de tiempos del software Neva

generan en GoogleMaps se observó que éstas tienen una distancia de recorrido muy larga, y el orden de visita de las paradas no resulta muy práctico, incluso si se reordenan a posteriori la distancia del recorrido sigue siendo larga.

Se concluye que el heurístico debe adaptarse para proporcionar soluciones de mejor calidad. Actualmente se emplea un proceso de selección de ruleta en el que a mayor ahorro, mayor probabilidad de ser elegido; sin embargo, aunque la probabilidad sea baja, por la aleatoriedad del proceso también es elegible una unión con baja probabilidad.

Como trabajo a futuro, se proponen cambiar la forma de seleccionar las paradas, restringir la longitud de las rutas, plantear alternativas para

Conclusiones

A diferencia de la ruta actual seguida por el Lobús, al establecer que cada estudiante sólo puede acceder a la parada ubicada en su colonia, obliga a que las rutas visiten dichas paradas, con ello se atiende la dispersidad de los estudiantes en el área metropolitana de Saltillo.

Al visualizar la secuencia de las rutas que se

generar la matriz de accesibilidad y desarrollar un sistema que permita la actualización y manipulación de los datos que conforman la instancia de una manera más eficiente.

Referencias

- [1] Clarke, G. y Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.
- [2] Juárez-Peláez M. (2019). Implementación de un sistema heurístico para la generación de rutas para el servicio de transporte de Lobús de la UAdeC. Tesis de maestría. Universidad Autónoma de Coahuila, México.
- [3] Pacheco, J., Caballero, R., Laguna, M., y Molina, J. (2013). Bi-objective bus routing: An application to school buses in rural areas. *JORS*, 47:397–411.
- [4] Park, J. y Kim, B.-I. (2010). The school bus routing problem: A review. *European Journal of Operational Research*, 202(2):311–319.
- [5] Schittekat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F., y Springael, J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, 229(2):518 – 528.
- [6] Soares de Siqueira, V., Jorge Ebrahim Lima e Silva, F., Nascimento da Silva, E., Vieira Santos da Silva, R., y Lisboa, M. (2016). Implementation of the metaheuristic GRASP applied to the school bus routing problem. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 6:137–145.

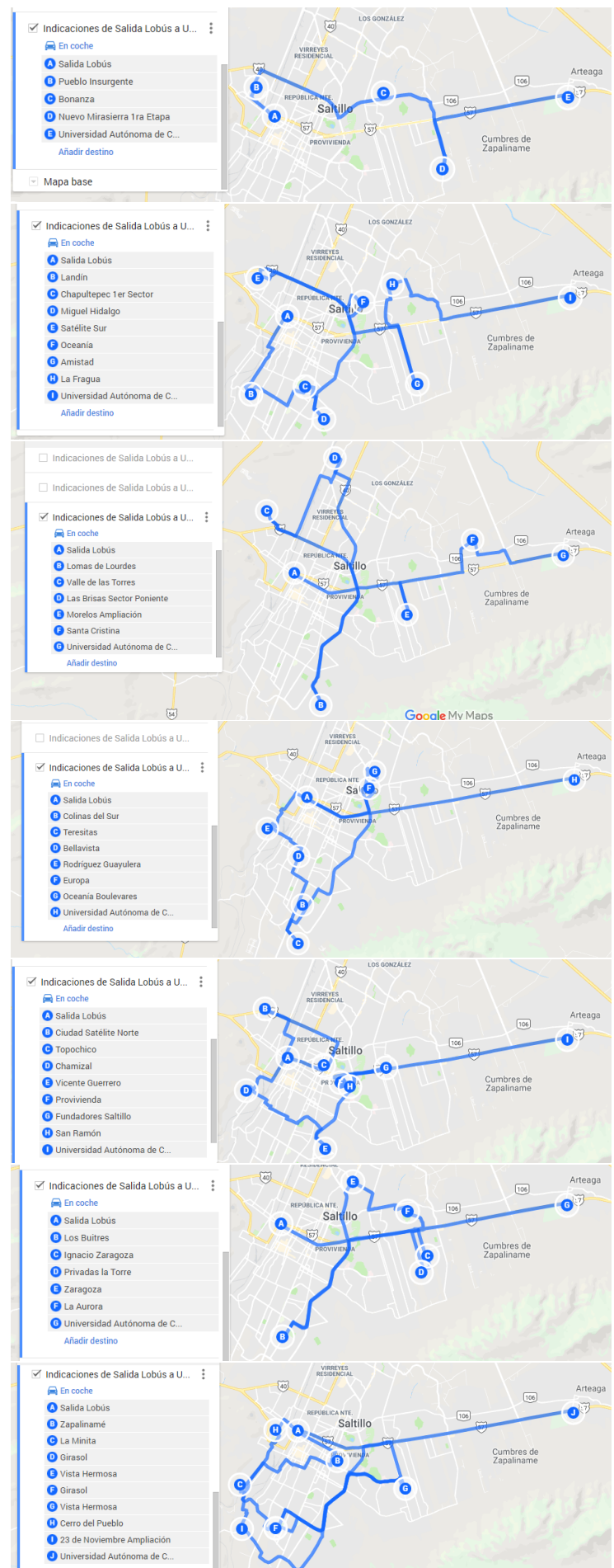


Figura 8: Rutas R19 a R45 generadas con la matriz de tiempos del software Neva