

De esta situación se tiene que no será levantada la misma cantidad de contenedores por el camión cada día.

En base a la información de cuales contenedores se visitarán cada día se podría aplicar algún mecanismo de optimización o alguna heurística para resolver la ruta diaria.

Si se encontrara un algoritmo exacto de tiempo de ejecución polinomial según la cantidad total de contenedores para este problema, también se lo encontraría para TSP (Travelling Salesman Problem), ya que es un caso particular del mismo. Por ejemplo, si tomamos que el tiempo en que se demora en llenarse cada contenedor es de 24 horas y el camión tiene capacidad infinita, cada día deberían recorrerse todos los contenedores y por tanto se convierte cada recorrido en un problema TSP, y estaríamos teniendo entonces un algoritmo exacto de complejidad polinomial en el tiempo para TSP, cuando en realidad no se conoce ninguno aún, si es que existe. Por lo tanto se tiene un problema perteneciente a la clase NP-hard, ya que TSP es reducible a este problema.

Es por eso que elegimos este problema a ser resuelto con algoritmos evolutivos como metaheurística, ya que se pueden obtener soluciones de buena calidad con un uso eficiente del recurso tiempo.

III. ESTRATEGIA DE RESOLUCIÓN

Se utilizarán Algoritmos Evolutivos, en particular Algoritmos Genéticos.

A. Modelo matemático

El problema refiere a intentar minimizar el costo, que en este caso es el consumo de combustible por parte del camión. Como se tiene la suposición de que es siempre el mismo mientras el camión está en marcha, a modo de simplificación se tiene en cuenta el tiempo en que tarda en hacer todos el recorridos como función objetivo, ya que minimizar esto es equivalente a minimizar el consumo de combustible. También se puede suponer que el tiempo de carga de un contenedor es constante, de parámetro T_c . Entonces los tiempos serán los de traslado entre contenedores, que se tomará en función de la distancia recorrida entre los mismos y la velocidad promedio, y los de cargarlos.

Se tiene que la municipalidad está dispuesta a realizar hasta un recorrido por día [2], en horario nocturno. Se quiere planificar el recorrido para una semana modelo. Basta con determinar cuáles contenedores serán recorridos en cada día, y en qué orden. Hay que recordar que cada contenedor tiene una capacidad máxima de almacenamiento, la cual se satisface según la demanda en su entorno. Cuánto tarda en llenarse depende de la población de sus manzanas adyacentes. Existe entonces la restricción de que no debe dejarse que los contenedores se llenen por completo, ya que esto implicaría que luego la gente no tenga dónde dejar su basura, y hasta inclusive a veces la dejan al costado de los contenedores sobre la vereda o la calle, ensuciando la ciudad. También se tiene la restricción de que los camiones tienen una capacidad máxima de carga, que en un recorrido no pueden sobrepasarla.

Se sabe que los camiones salen de un depósito dado y al finalizar el recorrido se dirigirán a un vertedero determinado. Además, cabe destacar que en una solución candidata, en un día determinado tiene que haber por lo menos un contenedor a recoger, sino ni vale la pena hacer el recorrido ese día.

Se tiene entonces que la función objetivo es la siguiente:

$$\min \sum_{d=1}^7 1_{N_d > 0} \times (N_d T_c + \frac{1}{v} \sum_{i=0}^{N_d} \text{dist}(x_{d,i}, x_{d,i+1}))$$

Siendo $x_{d,i}$ el número de contenedor (están numerados de 1 a K) visitado el día d en el lugar (en el orden) i , N_d la cantidad de contenedores visitados el día d , T_c el tiempo en recoger la basura de un contenedor y v la velocidad promedio de traslado entre contenedores.

Lo que expresa esta función es el tiempo del recorrido de cada día, si es que se hace.

Las restricciones se pueden expresar de la siguiente forma, siendo $y_{d,c}$ la variable que indica si el contenedor c es recogido el día d :

$$\forall c \in \{1, \dots, K\}, d_0 \in \{1, \dots, 7\} \rightarrow \exists d \in \{d_0, \dots, d_{0+|D_c|-1}\} / y_{d,c} = 1$$

Es decir que para todo contenedor y para todo intervalo de días que tarda justo antes de llenarse (según lo que demora en llenarse, D_c), debe haber al menos uno de los días en el intervalo cuyo recorrido pase por el mismo. Se tiene en cuenta que los días son circulares respecto a las semanas, y entonces que el final de la semana se “engancha” con el principio de la misma. Considerar que la semana modelada es una semana representativa, y entonces que el estado de los contenedores la primera vez que se pone en práctica una solución no se tiene en cuenta, sino que se trata al problema como si fuera una semana más del año.

También se tiene que tener en cuenta la capacidad del camión:

$$\forall d \in \{1, \dots, 7\}, m \times k \times \sum_{c=1}^K y_{d,c} \times \frac{d - \text{ant}(d, c)}{D_c} \leq M$$

Esto es que la suma de la capacidad fija m de cada contenedor, multiplicado por la capacidad de compactación de los camiones (entre 0 y 1), multiplicado por qué tan lleno estaba al recogerlo, sea menor o igual a la capacidad del camión M . Para saber qué tan lleno estaba, se mira la diferencia de días con el día de la última recogida del contenedor y se lo divide entre lo que se tarda en llenar. Éste se define así:

$$\text{ant}(d, c) = \max_a \{a < d / y_{a,c} = 1\}$$

Cabe aclarar que lo que se demora en desplazarse hasta un contenedor una vez que se empieza el recorrido no es tenido en cuenta en el tiempo de la demanda del mismo, es decir que si por ejemplo un contenedor se llena en 24 horas y ese tiempo termina al comenzar el recorrido, se estaría cumpliendo con la demanda igualmente. Este tiempo es despreciable teniendo en cuenta lo que se espera que demoren los recorridos. Si se mira el peor caso, que sería ir un día primero a un contenedor particular que demora 24 horas en llenarse, y al siguiente día visitarlo en lugar último, la diferencia sería de unas pocas horas, además agregando que a las horas que se recorren los contenedores hay muy poca actividad de arrojar residuos en ellos.

Una solución candidata es entonces un conjunto de recorridos, uno por cada día de la semana.

B. Representación

La representación es de largo variable. Es un vector de 7 elementos, uno por cada día de la semana, en donde cada uno adentro tiene una lista de contenedores, que indicará el orden que deben ser recorridos. Dentro de la lista de contenedores a recorrer de un día particular no debe haber repetidos. Un individuo a su vez debe cumplir las restricciones anteriormente planteadas.

La decodificación es parecida a la usada en la representación con permutaciones del problema TSP, se calcula de antemano los caminos y sus distancias entre todo par de puntos a visitar, y entonces se puede armar fácilmente el camino del recorrido una vez que se sabe qué contenedores son visitados y en qué orden.

C. Inicialización de la población

Para inicializar a la población se hará de la siguiente forma. Mientras haya contenedores cuya demanda no es satisfecha (se llenan antes de que pase un camión), se toma uno de ellos y se elige un día al azar para pasar a recogerlo. Luego de esta etapa, se procede a elegir un orden aleatorio de recorrida, para cada día.

D. Función de fitness

La función de fitness será entonces la siguiente, pasando a un problema de maximización:

$$f(x) = - \sum_{d=1}^7 1_{N_d > 0} \times (N_d T_c + v \sum_{i=0}^{N_d} \text{dist}(x_{d,i}, x_{d,i+1}))$$

En donde se tienen en cuenta entonces las distancias de los caminos entre un contenedor y el siguiente en el fenotipo.

E. Selección

Para el problema consideramos que en la selección no queremos que se sesgue tanto a los individuos con más fitness, así que decidimos usar Selección estocástica universal (Stochastic universal sampling, SUS).

F. Operador de cruzamiento

Se procederá de forma parecida a SPX (Single Point Crossover). Se tomará al azar uno de los dos padres a cruzar, y para el elegido se elegirá el punto de corte, teniendo en cuenta al individuo como si todas sus listas de recorridos estuvieran adyacentes, y entonces pudiendo caer el punto de corte en donde termina un día y comienza otra.

Luego de aplicar esto vamos a tener que hacer correcciones. Si llegan a quedar, dentro del día de punto de corte, contenedores a visitar repetidos, se sacarán los repetidos, eligiendo al azar uno de cada par de repetidos. Si se deja de satisfacer la demanda de llenado de un contenedor, se tomará un día de los intervalos para los cuales no se cumple y se lo agregará al recorrido, en algún lugar al azar. Luego, si se está superando la capacidad de carga del camión en alguno de los días, se le quitarán contenedores al azar de su recorrido hasta que se cumpla la restricción, pero eligiendo aquellos contenedores que no comprometan la demanda de llenado. De no ser posible no quedará otra opción más que descartar al individuo.

Creemos que buenas características de los individuos pueden ser mantenidas de esta manera.

G. Operador de mutación

Primero, se tomará al individuo como una lista sola, de forma parecida al operador de cruzamiento. Se recorrerá cada posición y con cierta probabilidad se podrá o bien agregar un contenedor a un recorrido (que no esté presente ya), o quitar uno del mismo, o intercambiarlo con otro al azar (posiblemente en otro recorrido). Si esto hace que no se verifiquen restricciones, va a tener que descartarse.

En [3] se abarca el estudio de un operador especial de cruzamiento, HPRM (Hybridizing PSM and RSM Operator) que es un híbrido de PSM (Partial Shuffle Mutation) y RSM (Reverse Sequence Mutation), el cual da mejores resultados para problemas de tipo TSP. Nuestro problema tiene mucho parecido con TSP, lo cual implica que vamos a seguir la sugerencia. Se va a hacer otra pasada, pero para cada día particular, el cual se aplicará este algoritmo para mutar.

Por motivos de completitud, se le asignará la mitad de la probabilidad de mutación a la primera parte mencionada, y la otra mitad a la segunda parte.

H. Reemplazo

Los hijos competirán con los padres y quedarán aquellos de mejor fitness.

I. Paralelismo

Vamos a utilizar el modelo de subpoblaciones distribuidas, en la cual se ejecutan varias subpoblaciones por separado, y a través de migraciones permite la introducción de diversidad entre las mismas.

J. Biblioteca a utilizar

Utilizaremos Malva [4] por su flexibilidad y cantidad de características que soporta, incluyendo paralelismo.

K. Datos necesarios

Se utilizará la información de [2] para saber la ubicación de los contenedores. Los datos de población están disponibles en [5].

IV. EVALUACIÓN EXPERIMENTAL

Para cada una de las siguientes partes, para cada instancia del problema, haremos al menos 30 ejecuciones, independientes entre sí, de forma de garantizar mínimamente que los datos estadísticos sean robustos.

A. Comparación con otro algoritmo

Vamos a comparar nuestro algoritmo contra uno Greedy que elige cuándo recorrer los contenedores la menor cantidad de veces en la semana, satisfaciendo la demanda. Luego, al saber cuáles recorrer cada día, se procederá con una heurística, utilizada en TSP, que se mueve desde un punto al más cercano del actual, hasta recorrer todos.

B. Ajuste de parámetros

Vamos a probar con valores de población de 50, 100 y 150. Las probabilidades de cruzamiento a determinar serán de 60%, 70% y 80%. Las probabilidades de mutación a ajustar serán 10%, 1% y 0,1%.

Probaremos todas las combinaciones de parámetros mencionadas para cada instancia de configuración que serán las siguientes: una con un juego de datos de zonas con mucha densidad de población, otra sobre una zona con poca densidad de población y otra sobre una zona de densidad media que en lo posible mezcle distintas densidades en la misma zona.

C. Evaluación

De las instancias del problema se tomarán datos como el mejor fitness obtenido, el promedio y desviación estándar de los mejores fitness obtenido en cada ejecución.

Las instancias a evaluar serán del mismo tipo que las de la parte de ajuste de parámetros en cuanto a densidades de población, pero distintos juego de datos (distintas zonas).

Se llevará a cabo un análisis estadístico, en el cual primero se determinará para cada instancia si el comportamiento de los mejores fitness es el de una distribución normal o no. Luego, a partir de ello se llevarán a cabo tests de hipótesis para comparar los resultados de las instancias que usan el mismo juego de datos.

Para evaluar la eficiencia computacional se procederá de forma similar, pero midiendo el tiempo total de cada ejecución y el tiempo para llegar a determinada calidad de resultados.

Se realizará todo el mismo trabajo con el algoritmo Greedy (en una sola ejecución, ya que es determinista) y se compararán las instancias con nuestro algoritmo genético.

D. Comparación con otros trabajos

Intentaremos comparar nuestro desempeño con otros trabajos, por ejemplo con [6], o con [7], aunque tienen especificaciones del problema distintas.

V. REFERENCIAS

- [1] IMM, «Sistema de recolección de residuos domiciliarios con contenedores,» 11 Setiembre 2013. [En línea]. Available: [http://www.montevideo.gub.uy/ciudadania/desarrollo-](http://www.montevideo.gub.uy/ciudadania/desarrollo-ambiental/limpieza/disposicion-de-residuos/sistema-de-recoleccion-de-residuos-)

[ambiental/limpieza/disposicion-de-residuos/sistema-de-recoleccion-de-residuos-](http://www.montevideo.gub.uy/ciudadania/desarrollo-ambiental/limpieza/disposicion-de-residuos/sistema-de-recoleccion-de-residuos-).

- [2] IMM, «Contenedores de residuos domiciliarios y sus horarios de recolección,» [En línea]. Available: <http://www.montevideo.gub.uy/ciudadania/desarrollo-ambiental/limpieza/disposicion-de-residuos/contenedores-de-residuos-domiciliar>. [Último acceso: Mayo 2014].
- [3] O. Abdoun, C. Tajani y J. Abouchabka, «Hybridizing PSM and RSM Operator for Solving NP-Complete Problems: Application to Travelling Salesman Problem,» *IJCSI International Journal of Computer Science Issues*, vol. 9, n° 1, pp. 374-378, 2012.
- [4] T. M. Project, «The Malva Project,» [En línea]. Available: <https://themalvaproject.github.io/>. [Último acceso: Mayo 2014].
- [5] AGESIC, «Población por zona censal en Montevideo,» 2004. [En línea]. Available: <https://catalogodatos.gub.uy/dataset/poblacion-por-zona-censal-en-montevideo>. [Último acceso: Mayo 2014].
- [6] T. Srivarapongse y R. Pitakaso, «Improvement of Garbage Collecting Route : Case Study of Municipal Muang Srikai, Warinchamrab District, Ubonratchathani Province.,» de *Asia Pacific Industrial Engineering & Management Systems Conference*, Patong Beach, Phuket, Thailand, 2012.
- [7] S. Sahoo, S. Kim, B.-I. Kim, B. Kraas y A. Popov Jr, «Routing Optimization for Waste Management,» *Interfaces*, vol. 35, n° 1, pp. 24-36, Enero-Febrero 2005.