Untitled

Pablo Hidalgo García

29/7/2020

1 Introducción

En la compleja sociedad moderna aparecen retos que es necesario abordar y solucionar de la mejor forma posible para que la vida sea fácil y llevadera. Además, existe una conciencia creciente acerca del impacto medioambiental de nuestras actividades por lo que las soluciones a estos retos deben tenerlo en consideración. Uno de esos muchos retos es la gestión de los residuos. La sociedad de consumo moderna implica una alta generación de residuos que es necesario procesar para reducir la huella medioambiental así como evitar la aparición de enfermedades o las incomodidades propias de la convivencia con los residuos. En el año 2017 se recogieron en España más de 22.000 toneladas de residuos (alrededor de 460 kilogramos por habitante) y que da cuenta de la magnitud y la dificultad en la gestión. En áreas rurales o insulares, esta gestión es complicada en el sentido de que los trayectos pueden ser amplios y la frecuencia de recogida no puede ser diaria adaptándose la recogida al comportamiento de la generación de los residuos. Por ello, es fundamental que la recogida de estos residuos se haga de la forma más eficiente posible. Se pueden distinguir dos formas en la que la recogida de los residuos se puede mejorar. La primera de ellas es realizar inversiones que consigan adaptar los recursos e infraestructuras disponibles (camiones de recogida, puntos de recogida, plantas de procesamiento, etcétera) a los patrones de generación de residuos; la segunda es optimizar la recogida contando con los recursos ya existentes.

Es en este segundo enfoque en el que se incide en este trabajo, en particular en optimizar las rutas que recorren los camiones con el objetivo de recoger la máxima cantidad de residuo conforme a las restricciones de recursos. Las restricciones son:

- plantas de procesamiento,
- puntos de recogida,
- camiones,
- horas de trabajo.

De estas restricciones la más fuerte es el número máximo de horas de trabajo que se consideran de 6.5 horas dejando un margen para cualquier eventualidad. Esta restricción es obvia para que los trabajadores descansen adecuadamente y, además, por la naturaleza de la recogida de residuos, se fomenta la recogida durante los periodos nocturnos para molestar lo menos posible a la población.

Este trabajo se desarrolla en este contexto tomando como escenario de estudio la isla de la Palma (Islas Canarias) aunque su aplicación se puede extender y adaptar a cualquier otra área geográfica. La información disponible son: patrones de generación de residuos, emplazamiento de los contenedores y sus características, detalles de las rutas, tiempo y coste y las restricciones de los recursos disponibles. Existen tres puntos de origen y destino de las rutas situadas en tres municipios de la Palma: Breña Alta, Mazo y Los Llanos. Cada vehículo debe llevar a cabo una ruta diaria y, cada ruta se define como un secuencia de putnos de recogida que deben ser visitados por cada vehículo. Se considera que cada vez que un camión visita un punto de recogida, éste recoge todo el residuo acumulado.

Las principales contribuciones de este trabajo son:

1. Propuesta de un algoritmo metaheurístico híbrido entre una búsqudda tabú y una búsqueda por vecindarios variables.
2. Comparativa del algoritmo propuesto con la ruta real y con EXPÓSITO.
3. Estudio del algoritmo bajo distintos escenarios.

Este trabajo está organizado de la siguiente forma. La sección 2 hace un repaso de la literatura existente

En el problema se han considerado al recogida independiente de dos tipos de residuos: residuos de papel y cartón y residuos plásticos identificados, habitualmente, con los colores marrones y amarillos, respectivamente. Para cada tipo de residuo se considera un camión con un punto de origen y un punto de destino (el almacén y la planta de tratamiento de residuos) y una serie de puntos de recogida candidatos. Cada punto de recogida tiene su propia tasa de llenado que determina el nivel de residuo hasta el punto en el que esté saturado. La restricción principal que hay que abordar es la del tiempo máximo de trabajo que los operarios considerando un tiempo de 6.5 horas.

Este trabajo toma como punto de partida el ya iniciado en EXPÓSITO. En ese trabajo se aplica modelos metaheurísticos GRASP (greedy randomized adaptative search procedure) para abordar el problema. En este trabajo se se desarrolla un algoritmo híbrido entre la búsqueda tabú y la búsqueda por vecindarios. Además, en EXPÓSITO, los resultados obtenidos se comparan con las rutas existentes en la compañía de recogida de residuos en una semana concreta. En este trabajo, el estudio se extiende para evaluar el comportamiento del algoritmo para distintos escenarios (condiciones iniciales distintas así como distintos horizontes temporales.

El objetivo de este trabajo es el de aplicar algoritmos metaheurísticos en la gestión de las rutas de los residuos como alternativa a lo estudiado en (Expósito).

En este trabajo se ha utilizado un algoritmo híbrido entre la búsqueda tabú y la búsqueda por vecindarios variables (Variable Neighborhood search o VNS).

Durante el año 2017 se recogieron en España más de 22.000 toneladas de residuos. La eficiencia en su recogida hace que

En el conjunto de la sociedad se presentan retos que es necesario abordar. Por ejemplo, la gestión eficiente del tráfico o la gestión de los residuos.

Vivimos en una sociedad cada vez más preocupada por el impacto

Esquema de la introducción

1. Problema de la gestión de residuos en general.
2. Características de la gestión en zonas rurales o remotas.
3. Descripción del problema del trabajo.
4. Pinceladas de la solución.
5. Relación con EXPÓSITO.
6. Novedades de este trabajo.

2 Revisión bibliográfica

3 Descripción del problema

El problema se puede describir formalmente como un grafo completo dirigido $\mathcal{G} = (\Theta, A)$ donde $\Theta = \{\theta_1, \ldots, \theta_n\}$ se corresponde con el conjunto de los n emplazamientos y $A = \{(\theta_i, \theta_j) : \theta_i, \theta_j \in \Theta, i \neq j\}$, las aristas del grafo. Además, el conjunto $\Theta = P \cup E$ de forma que $P \cap E = \emptyset$. Conocemos el tiempo de viaje entre dos emplazamientos θ_i, θ_j al que denominaremos $d_{ij} > 0$ y que, debido a la orografía del terreno y la infraestructura de carreteras, en general, $d_{ij} \neq d_{ji}$.

El conjunto de los puntos de recogida se recogen por una flota de vehículos $\mathcal{V} = \{v_1, \ldots, v_k\}$. El origen del vehículo $v \in \mathcal{V}$ se denomina como $o(v) \in E$ y $t(v) \in E$ al destino final del vehículo.

Se define un horizonte de predicción $\mathcal{H} = \{1, 2, \ldots, h\}$.

$$F_i(d) = \min\{1, f_i(d)\}$$

Aunque la función de residuos acumulados $f_i(d)$ puede tomar cualquier forma, en este trabajo se ha considerado una función lineal $f_i(d) = b_i + q_i \cdot d$

4 Formulación matemática

En esta sección expresaremos el problema de optimización que queremos resolver. Éste se puede formular como un problema de programación entera mixta (Mixed-Integer Programming o MIP). Necesitamos las siguientes variables:

- X^v_{ijh}: variable que toma valor 1 si el vehículo $v \in \mathcal{V}$ va desde el punto θ_i hasta el $theta_j$ en el día $h \in \mathcal{H}$ y 0 en caso contrario, $\forall(\theta_i, \theta_j) \in A$
- Y^v_{ih}: toma valor 1 si se visita el punto $\theta_i \in \Theta$ por el vehículo $v \in \mathcal{V}$ en el día $h \in \mathcal{H}$ y 0 en caso contrario.
- $\mathcal{T}_{ih} \in \mathbb{R}$: tiempo de recogida del punto $\theta_i \in \Theta$. Nótese que esta variable no depende de ningún vehículo ya que se considera que un punto de recogida solo se puede visitar por un único vehículo en un mismo día.

Así, las restricciones del problema son las siguientes:

$$\sum_{j \in P} X^v_{o(v)jh} = 1$$

$$\sum_{i \in P} X^v_{it(v)h} = 1$$

$$\sum_{j \in \Theta} X^v_{jkh} = \sum_{j \in \Theta} X^v_{kjh}, k \in P,$$

$$\sum_{j \in \Theta} X^v_{ijh} = Y^v_{ih}$$

$$\sum_{v \in \mathcal{V}} Y^v_{ih} \leq 1$$

$$T_{jh} \geq T_{ih} + s_i + t_{ij} - M \cdot \left(1 - \sum_{v \in \mathcal{V}} X^v_{ijh}\right)$$

$$T_{ih} + s_i + t_{it(v)} \leq W_{vh}$$

$$X^v_{ijh} \in \{0, 1\}$$

$$Y^v_{ih} \in \{0, 1\}$$

$$T_{ih} \geq 0$$

El objetivo es

$$\max \sum_{v\in V}\sum_{i\in P}\sum_{h\in H} F_i(\eta_{ih}) \cdot Y_{ih}^v$$

5 Solution approach

En esta sección se describe la solución propuesta. La formulación matemática anterior es intratable computacionalmente cuando la dimensionalidad de los escenarios es alta (Expósito-Márquez et al. 2019), como es el caso del escenario de recogida de residuos en la isla de la Palma que utilizamos aquí.

La solución propuesta busca diseñar las rutas que deben seguir cada vehículo cada uno de los días contenidos en el horizonte temporal. Para ello se aplica en la marco general de la búsqueda en vecindarios variables o VNS (Mladenović and Hansen 1997) es un algoritmo metaheurístico para resolver problemas de optimización combinatorios y globales cuya principal idea es el cambio sistemático del vecindario de búsqueda tanto en un fase de búsqueda local como de una fase de perturbación que permita escapar de óptimos locales. Las distintas extensiones de este algoritmo se pueden consultar en (Hansen et al. 2010). El algoritmo VNS está basado en tres hechos fundamentales: (1) un mínimo local con respecto a una estructura de vecindario no lo es necesariamente respecto de otra, (2) un mínimo global es un mínimo local con respecto a todas las estructuras de vecindario posibles y (3) para muchos problemas, el mínimo local con respecto a uno o varios vecindarios están relativamente cercanos unos de otros.

Aunque hay diferentes variantes del algoritmo VNS, una de las aproximaciones que más éxito han tenido es es la denominada General VNS (Hansen et al. 2010). Para definir este algoritmo comenzamos definiendo la función `NeighborhoodChange` que permite comparar los valores de dos soluciones de forma que, en caso de que la nueva solución sea mejor que la anterior, se actualiza la solución y se vuelve a usar el vecindario original $\mathcal{N}_1$. En caso contrario, se utiliza el siguiente vecindario.

Inicialización. Seleccionar el conjunto de las estructuras de vecindarios $\mathcal{N}_k$ para $k = 1, \ldots, k_{max}$ que se usará en la búsqueda; encontrar una solución inicial x y su valor de la función objetivo $f(x)$; asignar $x_{opt} \leftarrow x, f_{opt} \leftarrow f(x)$; elegir un critero de parada.

Repetir lo siguiente hasta que se satisfaga el criterio de parada:

(1) Asignar $k \leftarrow 1$;
(2) Repetir hasta que $k = k_{max}$:
   (a) **Agitación**. Generar un punto x' aleatoriamente para el vecindario $\mathcal{N}_k(x)$;
   (b) $l \leftarrow 1$;
   (c) Repetir hasta que $l = l_{max}$:
      (i) **Búsqueda local.** Encontrar el mejor vecino en $\mathcal{N}_l(x')$ y denotarlo como x'';
      (ii) **Cambio de vecindario.** Si $f(x'') > f_{opt}$ hacer $f_{opt} \leftarrow f(x'')$, $x_{opt} \leftarrow x''$ y $l \leftarrow l_{max}$; en otro caso $l \leftarrow l + 1$;
   (d) **Mover o no.** Si $f(x'') < f_{opt}$, asignar $k \leftarrow k + 1$; en caso contrario, $k \leftarrow 1$.

Este algoritmo GVNS tiene varias extensiones, entre ellas la hibridación con otros algoritmos. Uno de los algoritmos metaheurísticos más utilizado es la búsqueda tabú. Esta algoritmo mantiene una *memoria* acerca de de la búsqueda ya realizada para prohibir (tabú) algunas soluciones recientes. Aunque se puede hibridar con GVNS de distintas formas, en este trabajo utilizaremos una hibridación de forma que el conjunto de soluciones vecinas esté limitado por la lista tabú. Así, el algoritmo anterior quedaría como

Inicialización. Seleccionar el conjunto de las estructuras de vecindarios $\mathcal{N}_k$ para $k = 1, \ldots, k_{max}$ que se usará en la búsqueda; encontrar una solución inicial x y su valor de la función objetivo $f(x)$; asignar $x_{opt} \leftarrow x, f_{opt} \leftarrow f(x)$; elegir un critero de parada; definir un conjunto de soluciones tabú $\mathcal{T}$.

Repetir lo siguiente hasta que se satisfaga el criterio de parada:

(1) Asignar $k \leftarrow 1$;

(2) Repetir hasta que $k = k_{max}$:

   (a) **Agitación**. Generar un punto x' aleatoriamente para el vecindario $\mathcal{N}_k(x, \mathcal{T})$;

   (b) $l \leftarrow 1$;

   (c) Repetir hasta que $l = l_{max}$:

      (i) **Búsqueda local.** Encontrar el mejor vecino en $\mathcal{N}_l(x')$ y denotarlo como x'';

      (ii) **Cambio de vecindario.** Si $f(x'') > f_{opt}$ hacer $f_{opt} \leftarrow f(x'')$, $x_{opt} \leftarrow x''$, $l \leftarrow l_{max}$ y actualizar el conjunto tabú $\mathcal{T}$; en otro caso $l \leftarrow l + 1$;

   (d) **Mover o no.** Si $f(x'') < f_{opt}$, asignar $k \leftarrow k + 1$; en caso contrario, $k \leftarrow 1$.

6 Características del problema

EL algoritmo definido en la sección anterior se puede aplicar a cualquier problema combinatorio. Para la aplicación del problema de optimización de recogida de residuos enunciado es necesario definir la estructura de los vecindarios y cómo se va a ir modificando la lista tabú.

Se consideran tres estructuras de vecindarios.

- Vecindario $\mathcal{N}_1$: dada una ruta, se añade un nuevo punto de recogida teniendo en cuenta que un punto de recogida no debe aparecer en dos rutas del mismo día.
- Vecindario $\mathcal{N}_2$: se intercambian dos puntos de recogida entre dos días distintos.
- Vecindario $\mathcal{N}_3$: se intercambia un punto de recogida que aparece en una de las rutas por uno que no haya sido recogido teniendo en cuenta, de nuevo, que un punto de recogida no puede ser recogido dos veces en el mismo día.

Cabe señalar que una de las principales restricciones del problema es el máximo tiempo que puede durar una ruta. Por tanto, en la búsqueda local en el vecindario de una solución, en el caso de que se encuentre algún vecino que, aunque tenga un valor de la función objetivo igual que la mejor solución, si ésta supone una disminución en el tiempo total de las rutas, se seleccionará esta solución.

Cada vez que se añade un punto de recogida a la solución, éste se introduce en la lista tabú $\mathcal{T}$. Además, es necesario definir un parámetro ρ que especifique durante cuántas iteraciones un punto de recogida no se considerará desde que fue introducido en la lista $\mathcal{T}$.

En realidad, el problema se puede dividir en dos partes: (1) asignar un punto de recogida a una ruta y (2) elegir el orden adecuado en el que se deben visitar los puntos de recogida en esa ruta. Para esta segunda parte, se realiza una estrategia de mejora de la ruta mediante el heurístico Lin-Kernighan (Helsgaun 2000) el cual se considera uno de las mejores heurísticas para aplicar al problema TSP. Este heurístico se implementa dentro de las soluciones obtenidas en las distintas estructuras de vecindario.

- $X = \mathcal{N}_k(x)$
- Repetir hasta encontrar una solución que verifique las restricciones:
  - Asignar a x' la solución de X con un mayor valor de la función objetivo; en caso de que haya varias soluciones posibles x', se escoge aquella que tenga un menor tiempo total en las rutas.
  - Aplicar la heurística Lin-Kernighan a x'.
  - Si x' no cumple las restricciones, $\mathcal{N}_k(x) \leftarrow \mathcal{N}_k(x) \setminus \{x'\}$

Algoritmo shake

Comenzamos definiendo los vecindarios.

Una ruta $r_h^v = (o(v), t(v))$

7 Estructuras de vecindarios

Se definen tres estructuras de vecindarios:

- $\mathcal{N}_1$: añadir nuevos puntos de recogida.

Los vecinos de una solución se obtienen añadiendo un nuevo punto de recogida a las rutas.

1. while improvement do:
    1. Improvement <- False
    2. For $\theta \in candidates$ do:
        1. For $v \in V$ do:
        2. For $h \in \mathcal{H}$ do:

- $\mathcal{N}_2$: intercambiar puntos de recogida.

Cada vecino se define como el intercambio entre rutas de puntos de recogida existentes. Es decir, dados dos camiones $v, v' \in \mathcal{V}$ (que puede darse $v = v'$) y dos días $d, d' \in \mathcal{H}$ de forma que $d \neq d'$, se escogen dos puntos de recogida $\theta \in R_d^v$ y $\theta \in R_{d'}^{v'}$ y se intercambian entre ellos.

- $\mathcal{N}_2$: cambiar puntos de recogida actuales por otros no considerados.

Los vecinos se obtienen al intercambiar un punto de recogida visitado en la solución actual en un día por otro que no haya sido visitado en ese día.

1. Inicialización:
    - $P_h =$ conjunto de puntos
    - $\hat{\Theta}_h$: conjunto de puntos de recogida que no se visitan en el día h.
2. Repetir hasta mejorar:

7.1 Vecindario 1: añadir punto de recogida

7.2 Vecindario 2: intercambiar puntos de recogida entre días

7.3 Vecindario 3: intercambiar punto de recogida con uno no visitado

8 Vecindarios

8.1 Vecindario 1: añadir

Sea $h \in \mathcal{H}$ un día del horizonte y r_h la ruta de ese día tal que $r_i = (o(\nu), p_1, p_2, \ldots, p_n, t(\nu))$. Representamos todas las rutas como $R = \{r_1, r_2, \ldots, r_{|H|}\}$.

Este vecindario busca añadir un nuevo punto $p \in P(\nu)$ tal que el incremento en la función objetivo sea máximo.

8.2 Vecindario 2: intercambiar entre días

8.3 Vecindario 3: cambiar puntos de recogida

9 Resultados computacionales

En esta sección se describen los resultados computacionales obtenidos en el escenario de análisis. Los objetivos de este análisis son:

1. Evaluar el comportamiento de la solución propuesta.
2. Comparar los resultados de la solución propuesta con la situación real actual y la solución propuesta en (Expósito-Márquez et al. 2019).
3. Evaluar el comportamiento de la solución en bajo diversos escenarios.

Los resultados computacionales que aparecen en esta sección se han realizado en un ordenador equipado con un procesador Intel Core i7-8700 3.20GHz y 16 GB de memoria RAM. La implementación de la solución se ha realizado utilizando el lenguaje de programación Python 3.8 recogido en el repositorio https://github.com/papabloblo/tfm.

En la sección 6.1 se describen los datos que componen el escenario bajo el que se realizan las pruebas computacionales.

9.1 Resumen de los datos

El escenario sobre el que se aplica la solución propuesta se realiza sobre la isla de La Palma (Islas Canarias, España). Esta isla tiene una extensión de 708.32 km^2 con una población de 82,671 habitantes. [fuente: https://www.ine.es/jaxiT3/Datos.htm?t=2910#!tabs-tabla] La isla tiene como punto más elevado el Roque de los Muchachos (2426 metros). Está compuesta por 14 municipios. Su particular orografía y las conexiones entre los municipios, hace que las distancias y tiempos de viaje no sea simétricos. Además, se trata de una isla con un fuerte componente turístico son una media de 140,000 turistas anuales [¿CITA?].

Los datos utilizados para el estudio provienen de un caso real de estudio, descrito en detalle en (Expósito-Márquez et al. 2019). Se contemplan dos escenarios de recogida de residuos y que constituyen problemas completamente independientes: residuos de papel y cartón (asociado al color azul) y residuos plásticos (asociado al color amarillo). Ambos escenarios comparten 338 puntos de recogida de los que se puede ver su situación en la figura X.

Los tiempos de viaje entre puntos de recogida se han obtenido a través de Google Distance Matrix API como una matriz T de tamaño $N \times N$ siendo N el número de puntos de recogida. Esta matriz tiene como características: (1) $t_{ii} = 0$, (2) $t_{ij} \geq 0, \forall i \neq j$ y (3), en general, $t_{ij} \neq t_{ji}$. En la matriz de distancias, la media de tiempo entre un punto de recogida y el punto de recogida más cercano es de 109.61 con una desviación típica de 111.47.

Un dato importante en el escenario es la forma en la que los puntos de recogida se van llenando de residuos. El modelo no impone de qué forma se debe producir este llenado. Se hace una simplificación de forma que se supone que el llenado se hace de forma lineal conforme a una tasa propia de cada punto de recogida. La tasa media de de llenado es de 0.134 con una desviación típica de 0.089 para la recogida de plásticos y 0.188 con una desviación típica de 0.108, para la recogida de cartones.

En ambos escenarios se considera que está disponible, diariamente, dos vehículos de recogida (uno para cada tipo de residuo) sobre el que, a efectos prácticos, no se considera un límite de capacidad de residuos recogidos.

Para el primer escenario y poder realizar la comparativa con (Expósito-Márquez et al. 2019), se considera datos obtenidos en la semana del 2 al 6 de octubre para obtener el nivel de llenado en el día de partida de la optimización. El tiempo máximo en el que puede durar un ruta se correpsonde con 6.5 horas.

Los algoritmos descritos anteriormente se aplican en un contexto de recogida de residuos de la isla de la Palma. Se consideran un total de XX puntos de recogida entre los que se tiene un tiempo medio de viaje de XX. Además, tal y como aparece en (Expósito-Márquez et al. 2019), se ha considerado añadir un tiempo de recogida de cada punto de 120 segundos. Tiempo máximo de ruta: 6.5 horas.

La orografía hace que los tiempos entre puntos de recogida no sea simétirca.

- Mapa de los puntos de recogida.

Conclusiones y trabajo futuro

Expósito-Márquez, Airam, Christopher Expósito-Izquierdo, Julio Brito-Santana, and J. Andrés Moreno-Pérez. 2019. "Greedy Randomized Adaptive Search Procedure to Design Waste Collection Routes in La Palma." *Computers & Industrial Engineering* 137 (November): 106047. https://doi.org/10.1016/j.cie.2019.106047.

Hansen, Pierre, Nenad Mladenović, Jack Brimberg, and José A. Moreno Pérez. 2010. "Variable Neighborhood Search." In *Handbook of Metaheuristics*, edited by Michel Gendreau and Jean-Yves Potvin, 146:61–86. Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-1665-5_3.

Helsgaun, Keld. 2000. "An Effective Implementation of the Lin–Kernighan Traveling Salesman Heuristic." *European Journal of Operational Research* 126 (1): 106–30. https://doi.org/10.1016/S0377-2217(99)00284-2.

Mladenović, N., and P. Hansen. 1997. "Variable Neighborhood Search." *Computers & Operations Research* 24 (11): 1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2.