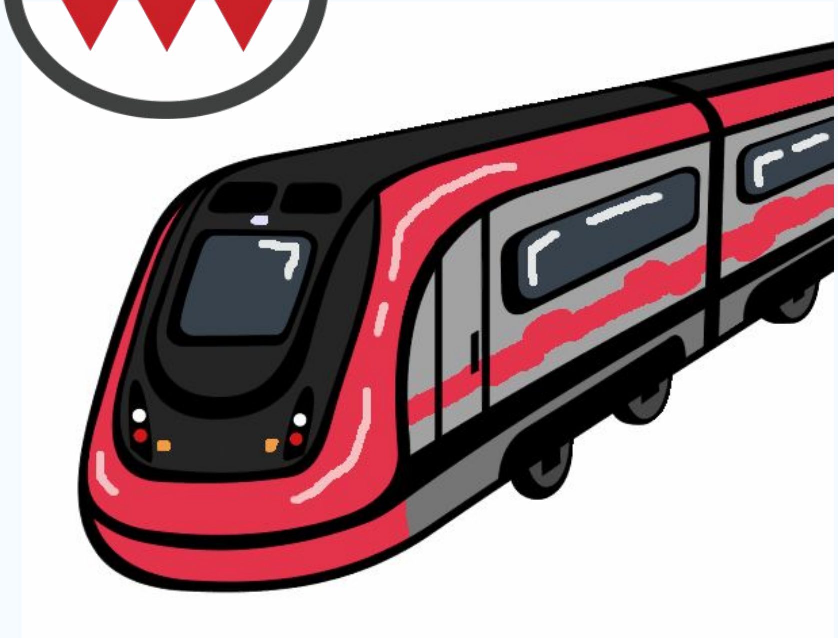


MA4402-1 -Simulación Estocástica:
Teoría y Laboratorio

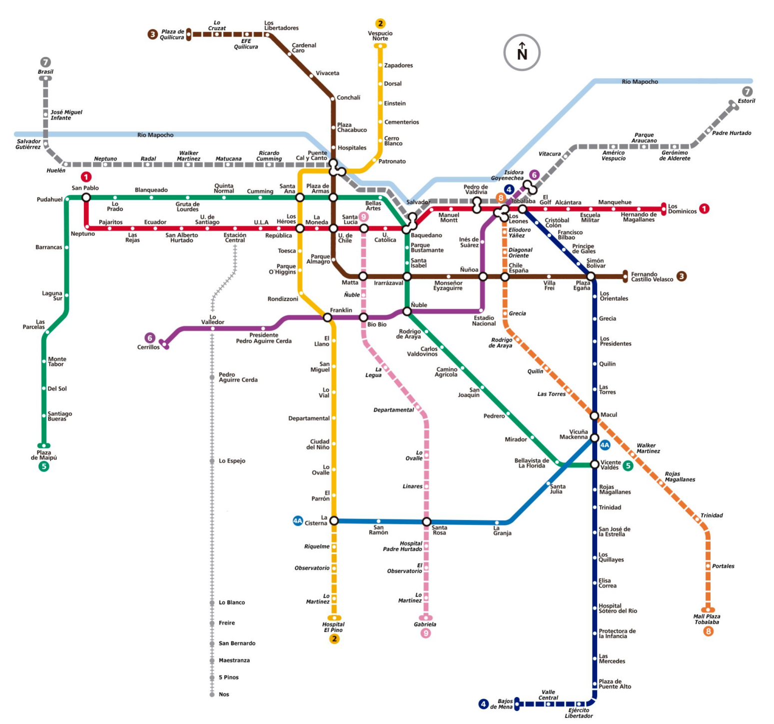
Presentación Proyecto Final

Simulated Annealing Aplicado a Sistema de Metro de Santiago

Allen Arroyo
Isidora Miranda

En qué consiste nuestro proyecto

Nuestro objetivo es reducir el tiempo de viaje de los pasajeros en el sistema de metro de Santiago.

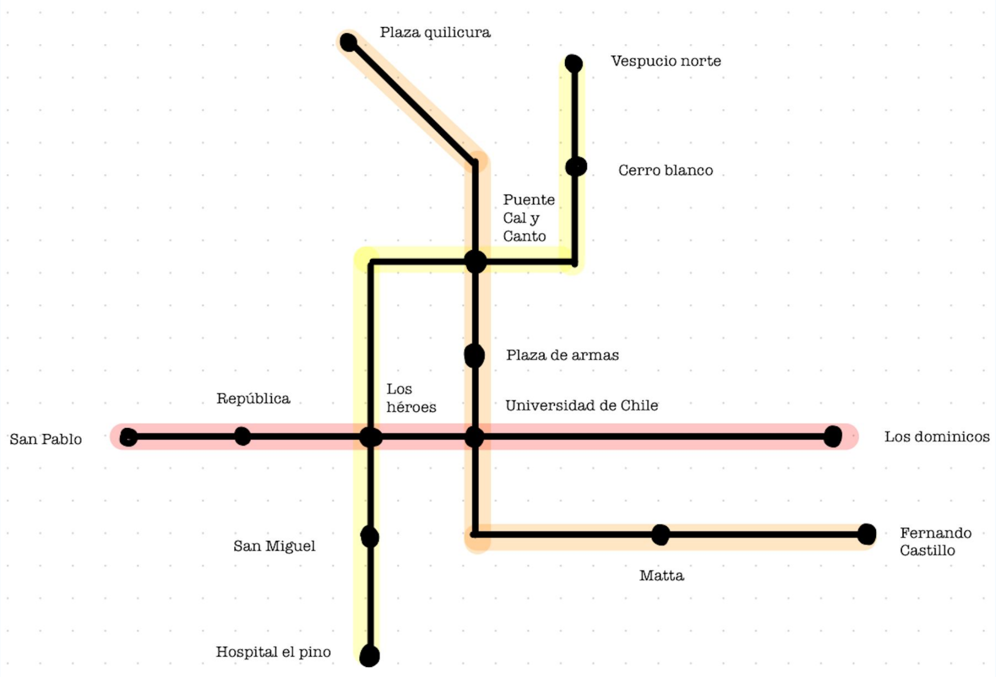


¿Pero cómo haremos esto?

Veremos si es posible coordinar las salidas de los primeros trenes desde las estaciones terminales, los tiempos de viaje y los tiempos de parada con el objetivo que el tiempo total de espera para el transbordo de pasajeros entre líneas de metro sea el mínimo.

Cómo lo realizaremos

Para esto realizaremos un modelo de juguete del metro que consiste en :



Donde trabajaremos con las líneas 1,2 y 3, las cuales cuentan con menos estaciones que el sistema real, trabajaremos con las estaciones que se ven en la imagen.

¿Qué variables consideraremos para el modelo?

- Tiempos entre estaciones
- Tiempos de parada en estaciones
- Tiempo de transferencia entre una línea y otra en las combinaciones
- Tiempos de salida entre trenes en las estaciones terminales
- Tiempo de la primera salida de las estaciones terminales

Inicialización del problema

Tenemos que nuestro conjunto de líneas es $L = \{l_{1up}, l_{1down}, l_{2up}, l_{2down}, l_{3up}, l_{3down}\}$
Ya que debemos considerar las direcciones de las líneas, es decir, estaremos trabajando realmente con 6 líneas.

```
l1up = ["san pablo", "republica","los heroes", "universidad de chile", "los dominicos"]
l2up= ["hospital el pino", "san miguel", "los heroes", "puente cal y canto", "cerro blanco", "vespucio norte"]
l3up = ["plaza quilicura","puente cal y canto", "plaza de armas","universidad de chile","matta","fernando castillo"]
```

```
l1down = ["los dominicos", "universidad de chile", "los heroes", "republica", "san pablo"]
l2down = ["vespucio norte", "cerro blanco", "puente cal y canto", "los heroes", "san miguel", "hospital el pino"]
l3down = ["fernando castillo", "matta", "universidad de chile", "plaza de armas", "puente cal y canto", "plaza quilicura"]
```

Donde tenemos tres estaciones de combinación que corresponden a:

- **Los Héroes:** Combinación Línea 1 y Línea 2
- **Puente Cal y Canto:** Combinación Línea 2 y Línea 3
- **Universidad de Chile:** Combinación Línea 1 y Línea 2

Qué restricciones tiene este problema

Para encontrar una solución factible del problema, nuestras variables deben cumplir restricciones. Las cuales explicaremos a continuación, pero primero definiremos las variables que establecimos anteriormente pero de una forma más específica con respecto a las estaciones y las líneas:

- t_l^0 = Tiempo que sale el primer tren de la estación terminal de la línea l, corresponde a los t0 definidos antes.
- $t_{lk(k-1)}^R$ = Tiempo de traslado de la estación k-1 a k en la línea l, corresponden a los Headways.
- t_{lk}^{Dw} = Tiempo de parada en la estación k en la línea l, corresponden a los Dwells.
- t_{sl}^A = Tiempo de llegada del primer tren a la estación s en la línea l.
- t_{sl}^D = Tiempo de salida del primer tren en la estación s en la línea l.

Restricción (1)

$$t_{sl}^A = t_0^l + \sum_{k=2}^{s} t_{lk(k-1)}^R + \sum_{k=2}^{s-1} t_{lk}^{Dw}$$

Restricción (2)

$$t_{sl}^D = t_0^l + \sum_{k=2}^{s} t_{lk(k-1)}^R + \sum_{k=2}^{s} t_{lk}^{Dw}$$

Qué restricciones tiene este problema

Restricción (3)

$$t^{w}_{sll'} = h \cdot T^{H}_{l'} + t^{D}_{sl'} - (t^{A}_{sl} + T^{Tra}_{sll'})$$

Modela cada cuanto tiempo llegan trenes a s en la línea l'

Modela en qué momento llegará el pasajero a la estación s en la línea l

- $t^{w}_{sll'}$=Tiempo de espera para realizar la transferencia de la línea l a la línea l′.
- h= Número de trenes que han pasado cuando los pasajeros del tren de alimentación llegan a la plataforma de conexión.
- $T^{H}_{l'}$=Es el intervalo de tiempo entre trenes consecutivos en la línea l′.
- $t^{D}_{sl'}$= Tiempo de salida del primer tren en la estación s en la línea l′.
- t^{A}_{sl}= Tiempo de llegada del primer tren a la estación s en la línea l.
- $T^{Tra}_{sll'}$= Tiempo de transferencia caminando entre las líneas l y l′ en la estación s (s es estación de combinación).

Cómo llevamos a cabo nuestro proyecto

Objetivo

Disminuir el tiempo de viaje de los pasajeros

Cómo lo haremos

Coordinando las salidas de los trenes desde las estaciones terminales, los tiempos de parada y los tiempos de viaje

con el fin

De que el tiempo de espera entre una línea u otra sea el mínimo

es decir

Minimizar el total de los tiempos de espera de transferencia

Lo cual se hace así

$$min f = \sum_{s \in S(l) \cap S(l')} \sum_{l \in L} \sum_{l' \in L} t^{w}_{sll'}$$

Qué datos usaremos para el proyecto

Para nuestras variables $t_l^0, t_{lk(k-1)}^R, t_{lk}^{Dw}, T_{sll'}^{Tra}, T_{l'}^H$

usaremos datos generados aleatoriamente, pero que tengan carácter adecuado. Donde nuestros tiempos corresponden a segundos, además, cabe destacar que esto se generó para cada línea y sus respectivas estaciones.

Luego usando estas variables fue posible calcular $t_{sl}^A, t_{sl'}^D, t_{sll'}^w$ que cumplieran

con las restricciones (1) , (2) y (3). Entonces ahora con todos los datos bien definidos es posible comenzar a calcular.

$$minf = \sum_{s \in S(l) \cap S(l')} \sum_{l \in L} \sum_{l' \in L} t_{sll'}^w$$

¿Pero cómo implementaremos esto?

Para implementar nuestro problema, es decir, $$min f = \sum_{s \in S(l) \cap S(l')} \sum_{l \in L} \sum_{l' \in L} t^w_{sll'}$$

usaremos **Simulated Annealing.**

Donde necesitamos soluciones factibles para cada iteración, para hacer esto existen diversos métodos, pero en este caso usaremos el método **Vector Modifying Algorithm.** de nuestro paper de referencia , que consiste en lo siguiente pseudocódigo:

```
input: Valores iniciales de tR,tDw,t0,M
  Sea t = tR o t = tDw o t = t0
  Anteriormente se definen nuestras cotas tmax y tmin
  Genera un indice pi entre [1,len(t)]
  Genera u = random.uniform(-M,M)
  t_pi = t[pi-1]

  definimos modified_value = t_pi*u
  if  modified_value > tmax :
        modified_value = tmax
  if  modified_value < tmin :
        modified_value = tmin
  t[pi-1] = modified_value
  return t
```

Función Objetivo y SA(Simulated Annealing)

- Pseudo-Algoritmo:

Algorithm 1: SA para dos iteraciones cualesquiera

Result: f, t_f^0, t_f^R, t_f^{Dw}

initialization;

$f_{old}, f_{new}, T_0, u, \omega, \tau, k, ML$

while $\tau < T$ **do**

  while $k < ML$ **do**

    $\Delta f = f_{new} - f_{now}$;

    if $\Delta f \leq 0$ **then**

      $f_{old} = f_{new}$;

    else

      $\rho = min\{1, exp(-\frac{\Delta f}{T})\}$;

      if $u_k < \rho$ **do** ;

       $f_{old} = f_{new}$;

    end

    k = k +1

  end

  $T = T \cdot \omega$

end

- Función a minimizar:

$$min f = \sum_{s \in S(l) \cap S(l')} \sum_{l \in L} \sum_{l' \in L} t^w_{sll'}$$

- Dado el paper en cuestión utilizamos las constantes :

$$T_0 = 100 \qquad \omega = 0.98$$

$$\tau = 0.1 \qquad Ml = 50$$

- **Vector Modifying Algorithm** debe ser aplicado en cada iteración del segundo while generando nuevos tnew's con los cuales se puede calcular un nuevo fnew.

¿Funcionó nuestro proyecto?

Nuestro proyecto si funcionó en nuestro modelo simplificado , mostraremos algunos ejemplos:

Ejemplo (1) con tDw y t0 iguales y con tR, tDw, t0 no aleatorios

```
t1r = [[40,50,100,120],[121,45,55,60,50],[56,70,50,89,102]
t2dw = [[11,11,11,11,11],[11,11,11,11,11,11], [11,11,11,11,11,11]]
t30 = [[0,0],[0,0],[0,0]

f(t1r,t2dw,t30)

632

s = SA(T0,tau,omega,ML,t1r,t2dw,t30,M1,M2)

(608.76798054252,
[[41.15785581497245, 40.34806561612547, 45.01838850457689, 43.64754851493289], [41.24070914761239,
42.90713700548127, 43.10096914930738, 47.61985143521244,41.49437239735259],
[40.6690459674818, 42.374476665520405,42.77921437268487,41.4881096846738, 51.43375629404919]],
[[10.266237878502846,10.035738717978225,10.022840650756564,10.049410022505258,13.43545102586806],
[11.783764025111376,10.408829528759439,10.02260383176619,10.045277279101871,11.29408403358851 7,
10.128308468104294],
[11.074463622643526,10.044465672518534,11.96304637785153,10.007397678981592,10.157134444606386,
 11.38839550187812]],
 [[0, 0], [0, 0], [0, 0]])
```

¿Funcionó nuestro proyecto?

Ejemplo (2) con tR, tDw y t0 generados aleatoriamente dada una v.a uniforme

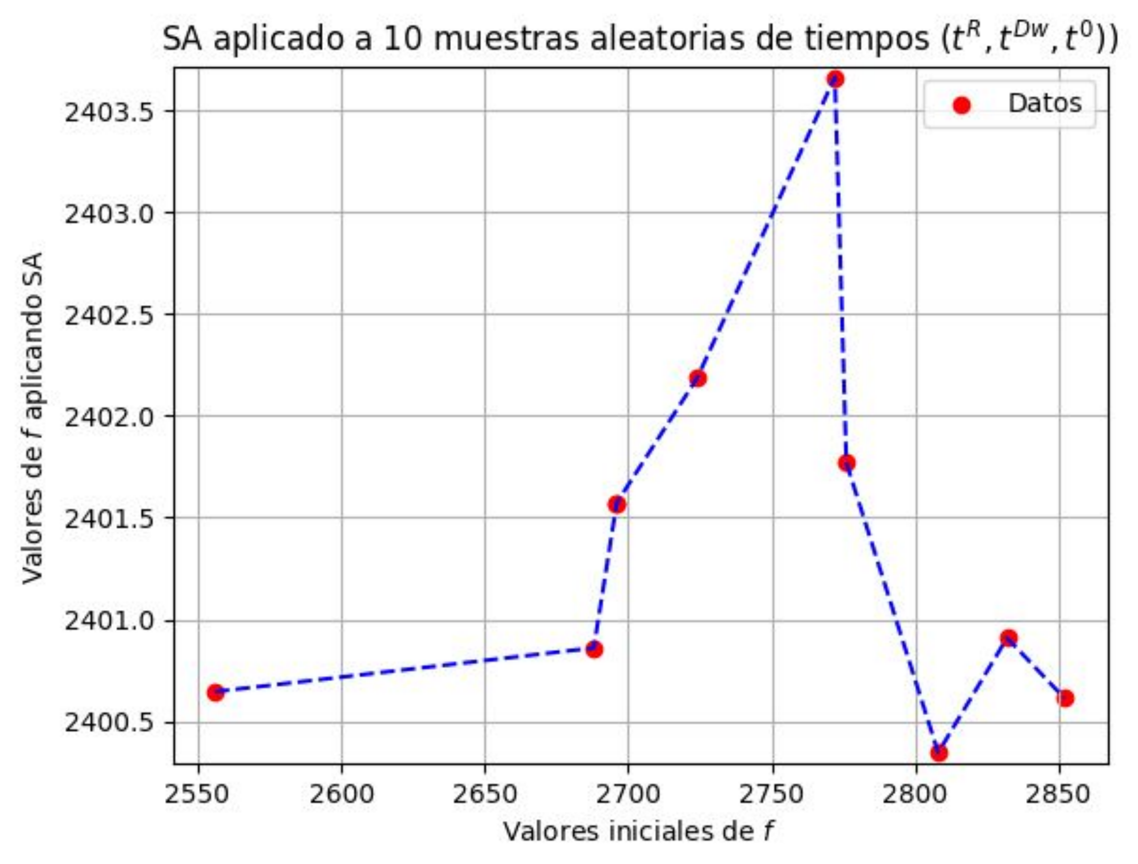
```
f(tR,tDw,t0), tR, tDw,t0

(980, [[157, 69, 130, 79], [143, 100, 93, 121, 69], [71, 158, 93, 124, 79]],
      [[0, 10, 13, 34, 0], [0, 33, 30, 33, 26, 0], [0, 23, 12, 20, 31, 0]],
      [[262, 289], [498, 700], [113, 50]])

solution = SA(T0,tau,omega,ML,tR,tDw,t0,M1,M2)
solution

(609.4099036412999,
[[45.6010393451097, 42.64315968935932, 47.91274472073371, 43.58556482306018],
[43.57717031568484, 43.16947002939291, 46.988383738793154, 45.02948713356016,41.06316188600738],
[44.19024363681156,42.875462511329296,45.00552035127857,40.15379131591442,41.0475788792748941],
[[0, 10.131199797069096, 10.025518751374172, 10.030213465691373, 0],
[0, 10.135666357858346,10.131248292091978,10.137917035329627,11.066633502578389,0],
[0,10.02231083944628, 11.064386425051833, 10.005267526391561,10.447217827569093,0]],
[[262, 289], [498, 700], [113, 50]])
```

¿Funcionó nuestro proyecto?

Se **generaron** 10 3-tuplas de muestras de tiempos tR,tDw,t0 aleatoriamente y posteriormente fue **aplicado el algoritmo SA** para **minimizar** los **tiempos totales de espera para el transbordo de pasajeros** entre las 3 líneas de la Red de metro estudiada.

¿Funcionó nuestro proyecto?

Para otras 10 muestras Independientes a las anteriores :



Información extra

Rangos de nuestras variables generadas aleatoriamente:

- $T_{l'}^{H}$ → Para la línea 1 es $[6 * 60, 8 * 60]$, para la línea 2 es $[3 * 60, 5 * 60]$ y para la línea 3 es $[5 * 60, 8 * 60]$.
- $T_{sll'}^{Tra}$ → Para la combinación de Los Héroes es $[60, 8 * 60]$, para la Universidad de Chile $[2 * 60, 6 * 60]$ y para Puente Cal y Canto $[60, 5 * 60]$.
- $\mathbf{t_l^0}$ → Entre $[0, 800]$
- $\mathbf{t_{lk}^{Dw}}$ → Entre $[10, 40]$
- $t_{lk(k-1)}^{R}$ → Entre $[40, 3 * 60]$

Referencia:

[1] Kang, L., & Zhu, X. (2015, 7 de diciembre). A simulated annealing algorithm for first train transfer problem in urban railway networks. Applied Mathematical Modelling. https://doi.org/10.1016/j.apm.2015.05.008

Tiempos de ejecución:

Para 3 líneas el costo de aplicar SA en Colab es :

- 1 muestra → 10[s]
- 2 muestra → 30[s]
- 10 muestras → 2[min]

Las Cotas M1 y M2 se deben elegir según algún criterio heurístico.

FIN
¿Preguntas?