

INVESTIGACIÓN OPERATIVA SUPERIOR

*nuevo módulo y
introducción a metaheurísticas*

Virtual

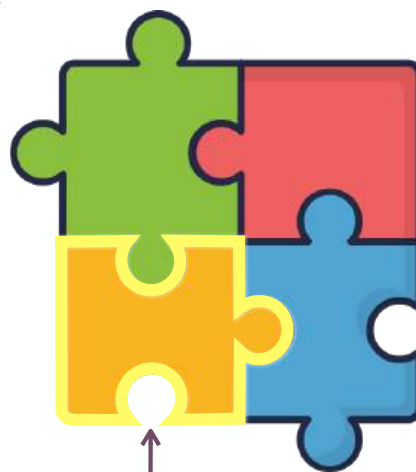
bienvenidos al último módulo

*logística en investigación
operativa*

*integración de mapas
a python*

metaheurísticas

*problema del
viajante*



logística

*optimización de las
actividades logísticas*

*sistema de
información
geográfica (GIS)*

*variaciones del
problema del viajante*

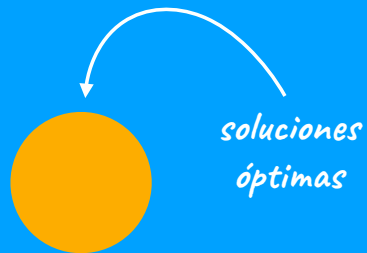
*optimización
combinatoria*



espacio discreto de soluciones



soluciones factibles





logística y ruteo



*transporte
marítimo*



transporte aéreo



*metaheurísticas trata de ...
optimización*





¿para qué otro método más de optimización?

fuerza bruta





el problema de la mochila

*un clásico de
investigación operativa*



v_m = volumen de la mochila

v_i = volumen del artículo i

u_i = utilidad del artículo i

$$\max = \sum b_i \cdot u_i$$

$$\sum b_i \cdot v_i \leq v_m$$



*piece of
cake*



con 4 elementos...



volumen=2
utilidad =7



volumen=20
utilidad =5



volumen=10
utilidad =10



volumen=7
utilidad =15



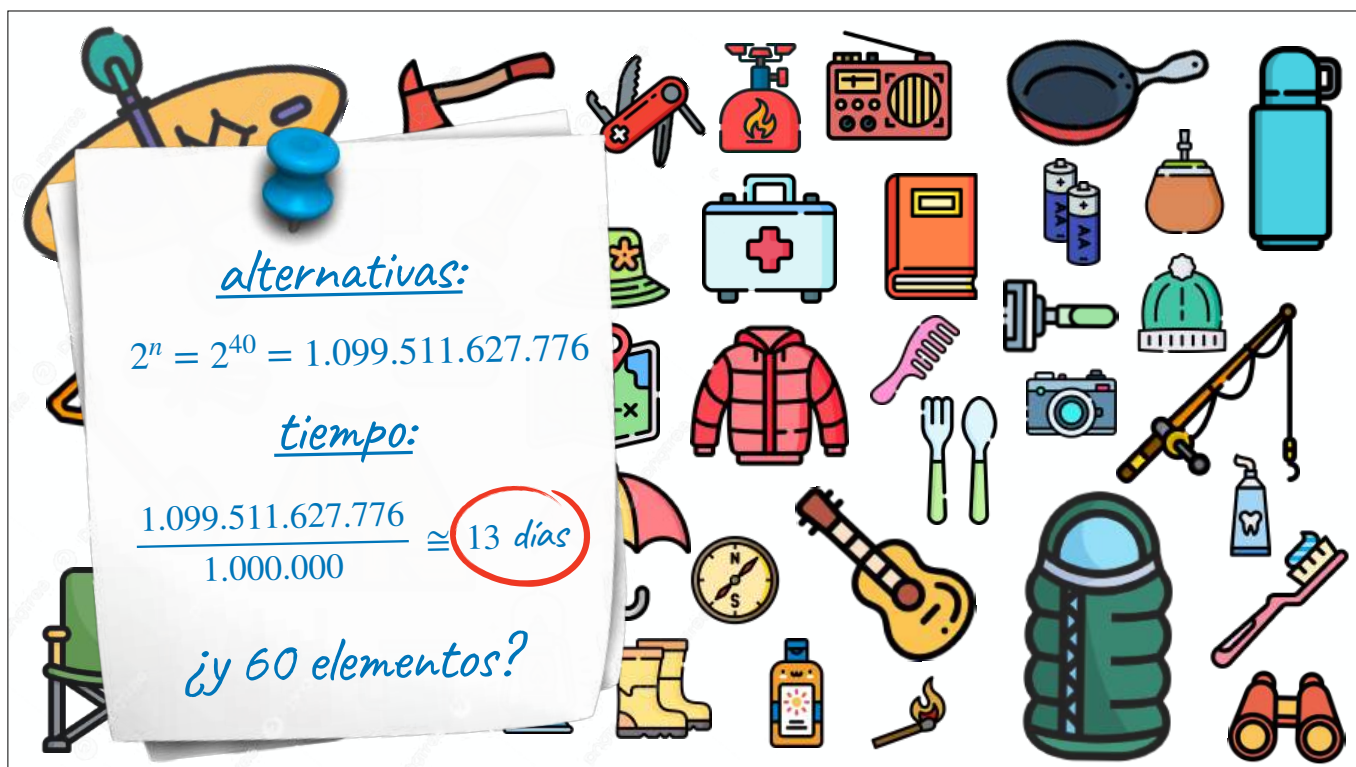
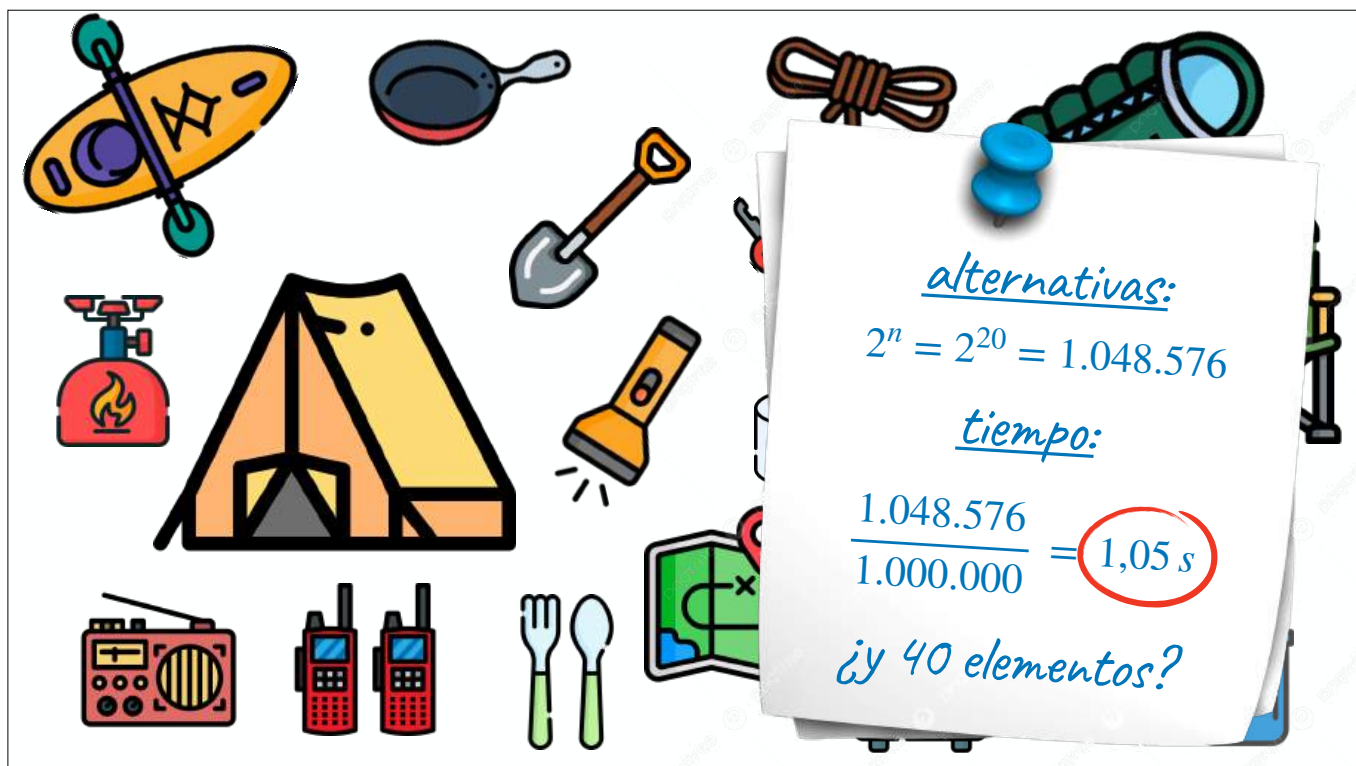
				Volumen total	Utilidad total
0	0	0	0	0	0
1	0	0	0	2	7
0	1	0	0	20	5
0	0	1	0	10	10
0	0	0	1	7	15
1	1	0	0	22	12
0	1	1	0	30	15
0	0	1	1	17	25
1	0	0	1	9	22
1	0	1	0	12	17
0	1	0	1	27	20
1	1	1	0	32	22
0	1	1	1	37	30
1	0	1	1	19	32
1	1	0	1	29	27
1	1	1	1	39	37



*volumen máximo
de la mochila 25*

*en este tipo de
problemas hay
2ⁿ soluciones
posibles*

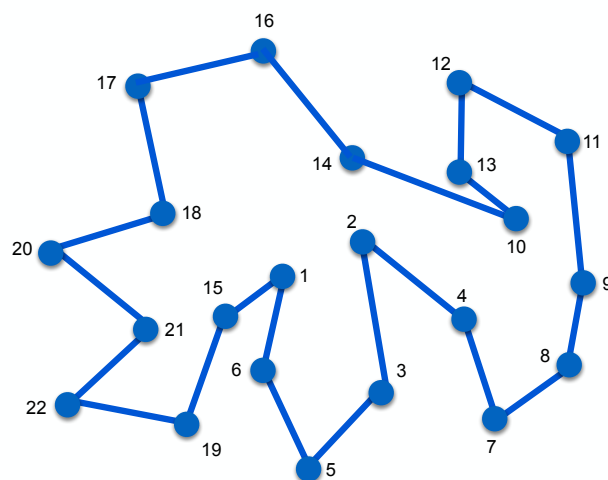
*si podemos generar y evaluar
un millón de posibilidades por
segundo . . .*

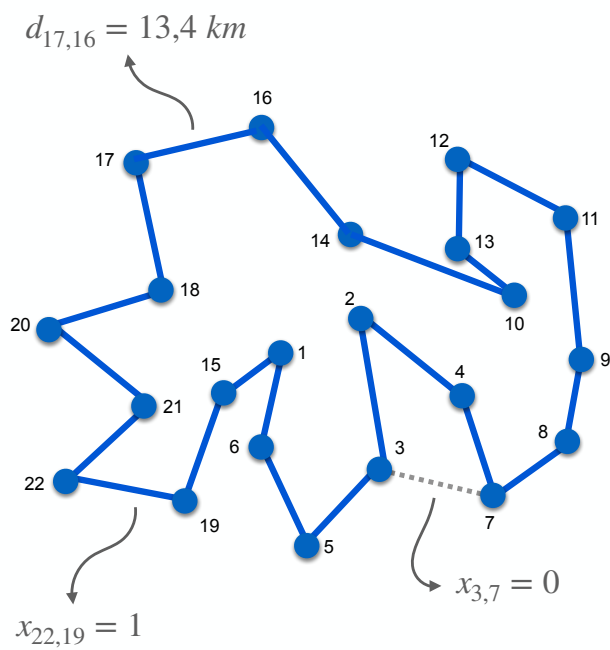
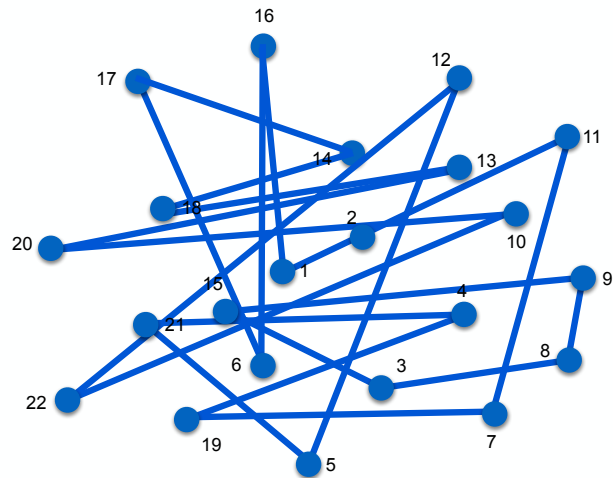


problema del viajante de comercio



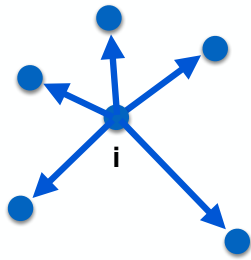
traveling salesman problem - otro clásico de investigación operativa





$$\min \sum_{i,j \in n} \boxed{x_{ij}} \cdot \boxed{d_{ij}} \quad \leftarrow \text{funcional}$$

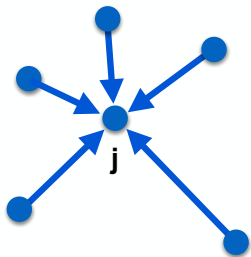
$$\left. \begin{array}{l} \sum_{j, i \neq j}^n x_{ij} = 1 \quad \forall i \in n \\ \sum_{i, i \neq j}^n x_{ij} = 1 \quad \forall j \in n \end{array} \right\} \text{restricciones}$$



$$\sum_{j, i \neq j}^n x_{ij} = 1$$

$$\forall i \in n$$

se debe salir una
única vez de cada
ciudad i

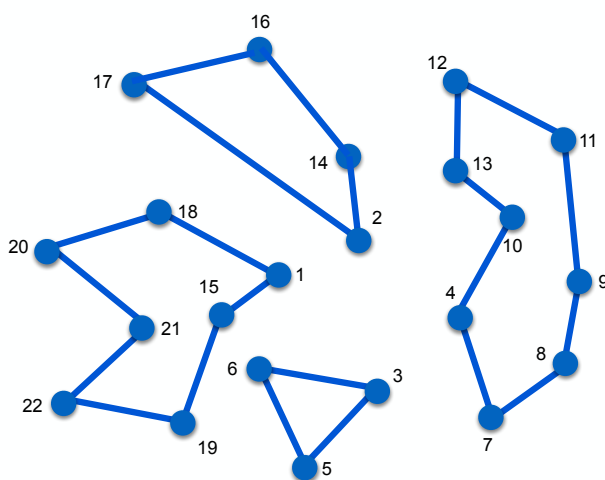


$$\sum_{i, i \neq j}^n x_{ij} = 1$$

$$\forall j \in n$$

se debe llegar una
única vez a cada
ciudad j

pero ...



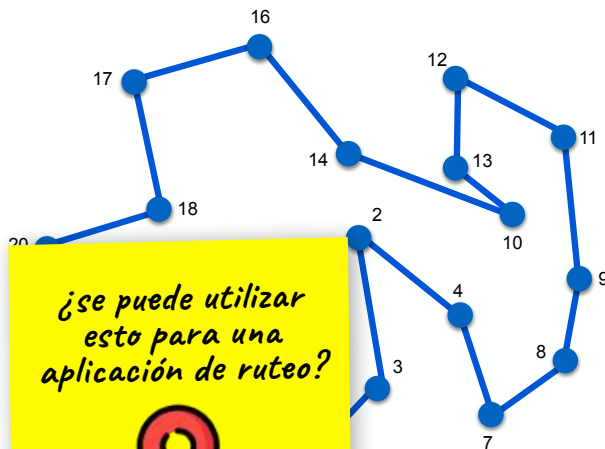
$$\min \sum_{i,j \in n} x_{ij} \cdot d_{ij}$$

$$\sum_{j, i \neq j}^n x_{ij} = 1 \quad \forall i \in n$$

$$\sum_{i, i \neq j}^n x_{ij} = 1 \quad \forall j \in n$$

$$u_i - u_j + n \cdot x_{ij} \leq n - 1 \quad \forall j > 1, j \neq i$$

ecuación anti-subtours



¿se puede utilizar
esto para una
aplicación de ruteo?



$$\min \sum_{i,j \in n} x_{ij} \cdot d_{ij}$$

$$\sum_{j, i \neq j}^n x_{ij} = 1 \quad \forall i \in n$$

$$\sum_{i, i \neq j}^n x_{ij} = 1 \quad \forall j \in n$$

$$u_i - u_j + n \cdot x_{ij} \leq n - 1 \quad \forall j > 1, j \neq i$$

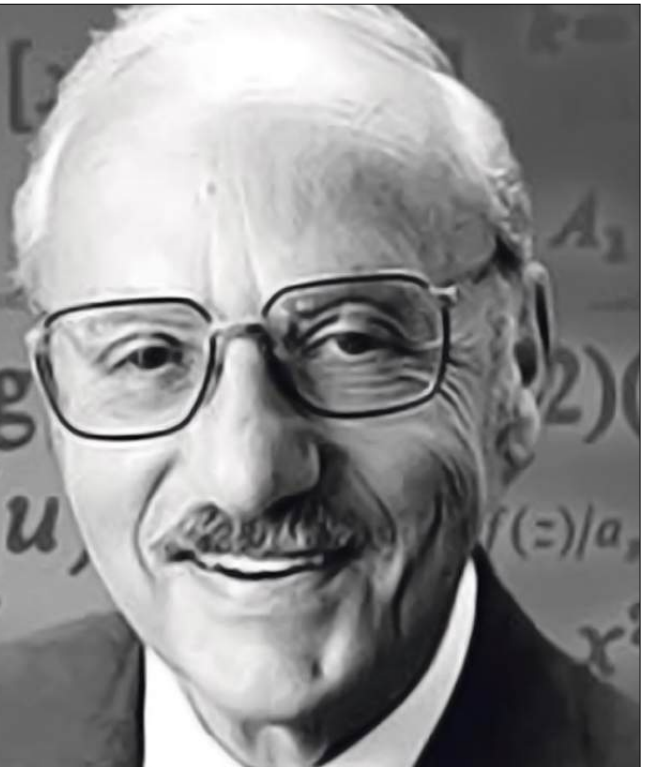
funciona muy bien pero ... **¡es super ineficiente!**
(lo que se traduce en pérdida de tiempo)

lo podemos resolver
con ...

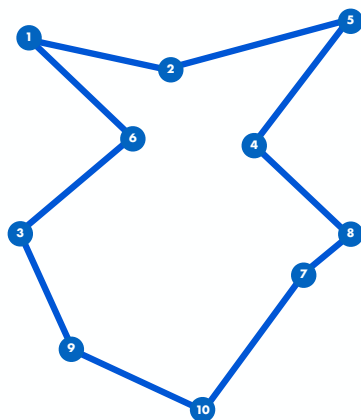
SIMPLEX

"True optimization is the revolutionary contribution of modern research to decision processes."

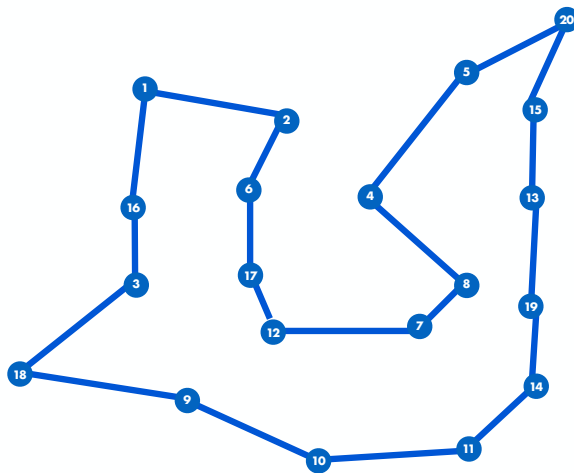
— George Dantzig



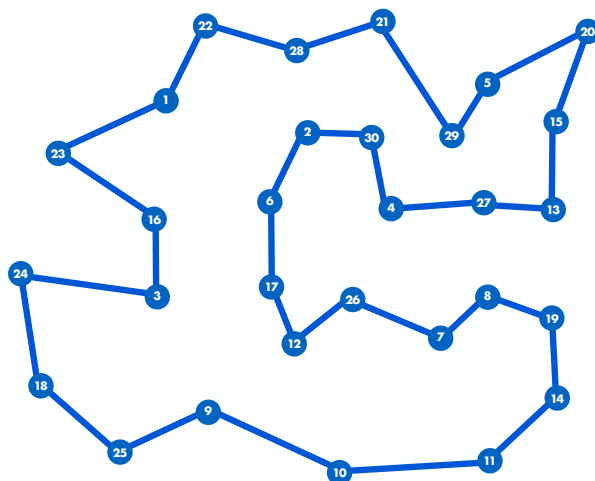
10 ciudades

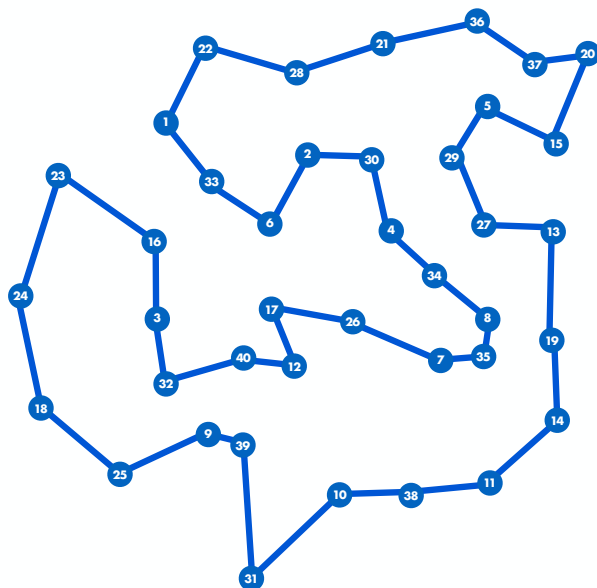


20 ciudades

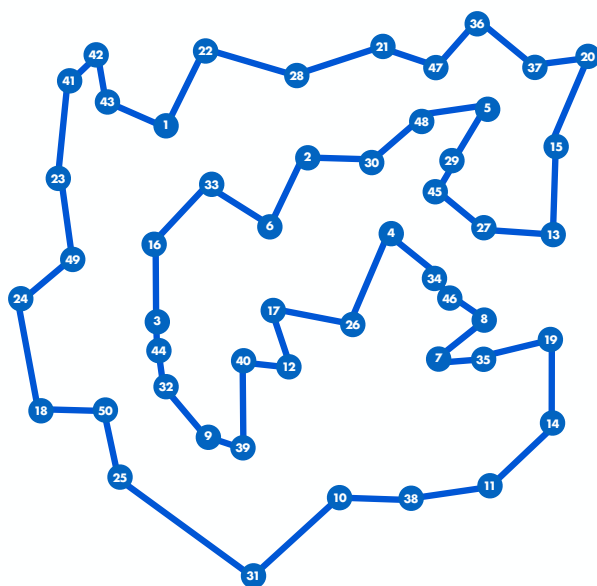


30 ciudades



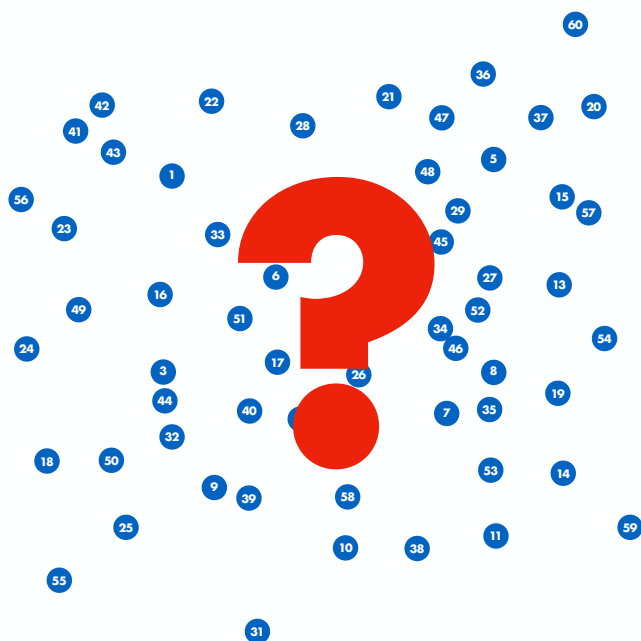


40 ciudades

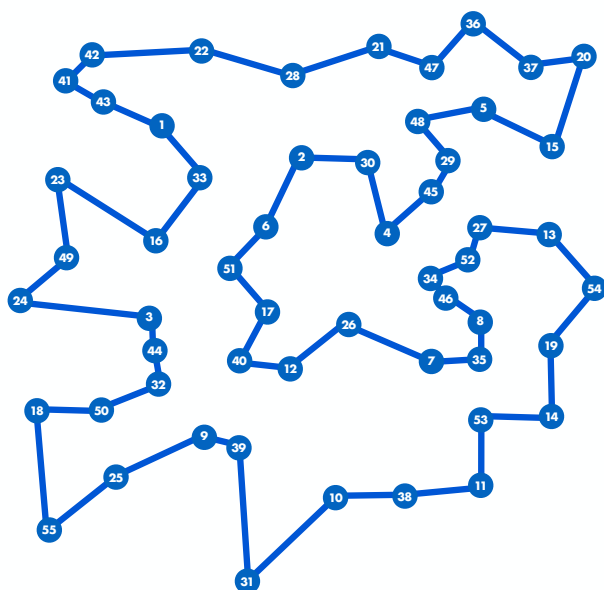


50 ciudades





60 ciudades



55 ciudades



nuevamente . . . el tiempo



DEMO 2

*... si unicamente
tenemos un martillo*



heurísticas

no se adaptan
fácilmente a
otros problemas

sentido
común

reglas y experiencia

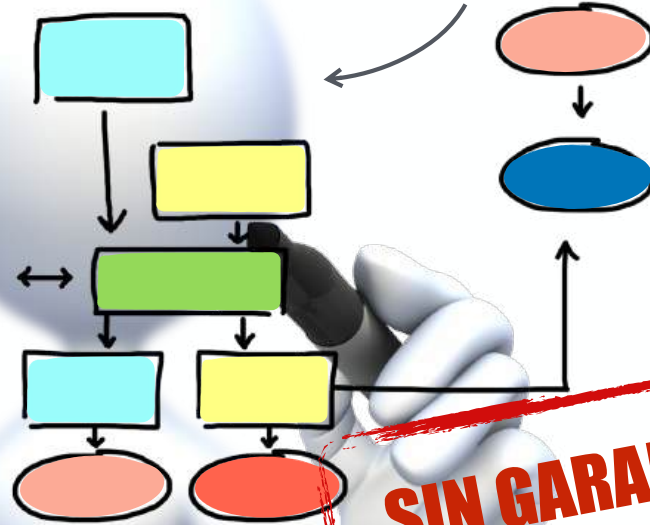


metaheurística

muy adaptable a distintos
problemas (flexibilidad)

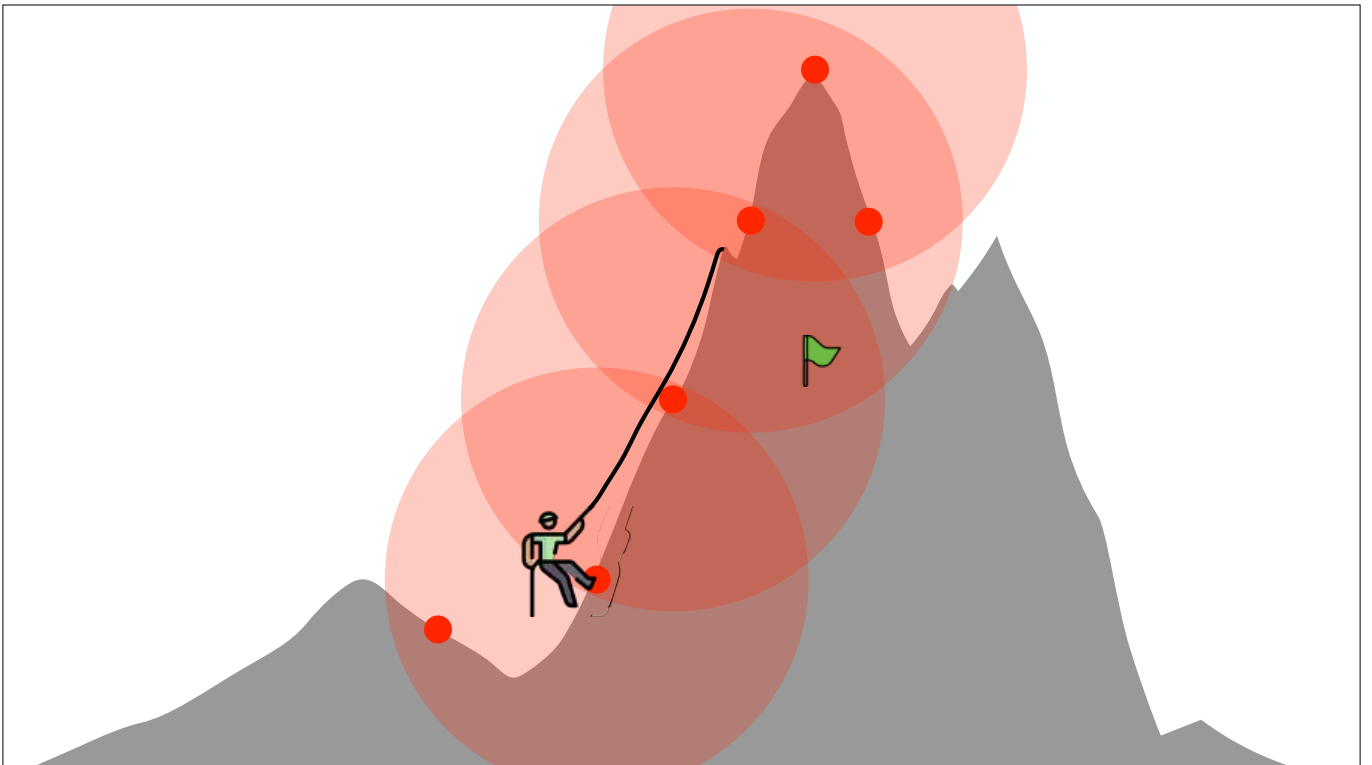
búsquedas
más robustas

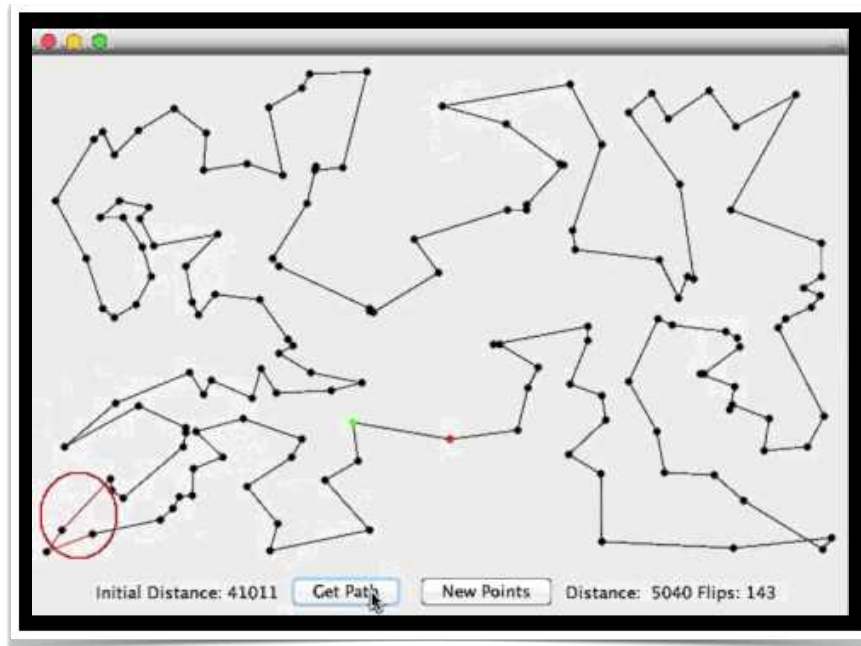
estrategias a fin de
mejorar la solución



SIN GARANTÍA

búsqueda local





DEMO 3

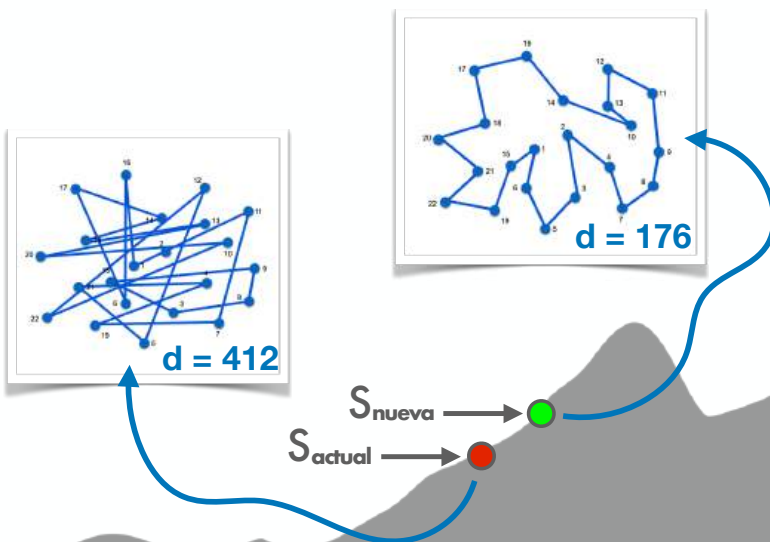
recocido simulado





*¿cómo salir de los
óptimos locales con
recocido simulado?*

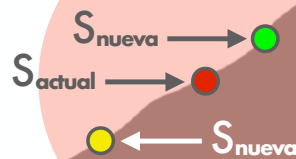
¿cuál es la lógica?



¿cuál es la lógica?

*acepto todas las
soluciones que
mejoran la actual*

*existe una
probabilidad de
aceptar soluciones
que empeoran el
funcional*



¿qué probabilidad?

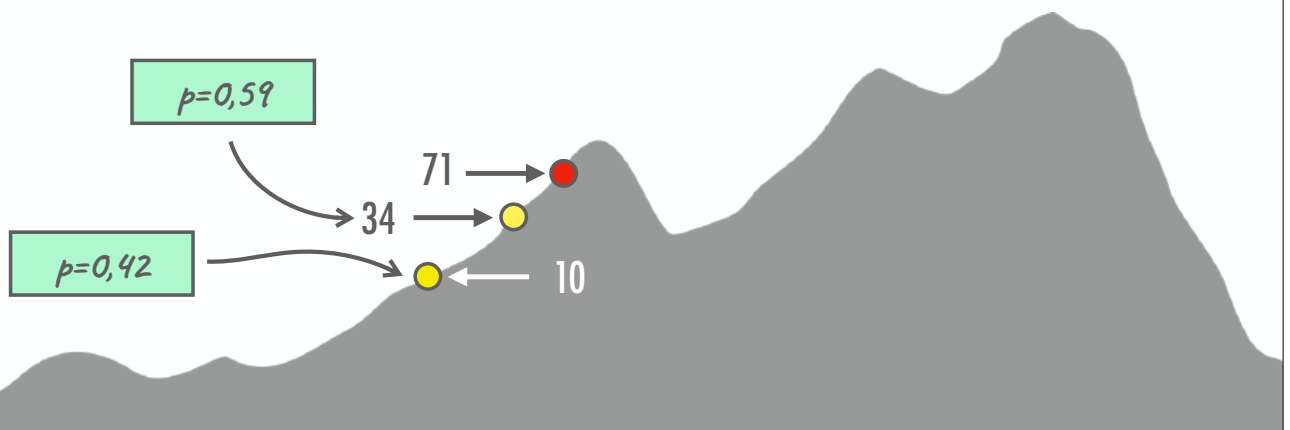
diferencia
entre ambas
soluciones

$$e^{\frac{\delta}{T}}$$

si estoy
minimizando ...
al exponente lo
multiplico por -1

si fijamos la temperatura en ...

$$T = 70^{\circ}\text{C} \quad p = e^{\frac{S_n - S_a}{T}} = e^{\frac{s_n - 71}{70}}$$



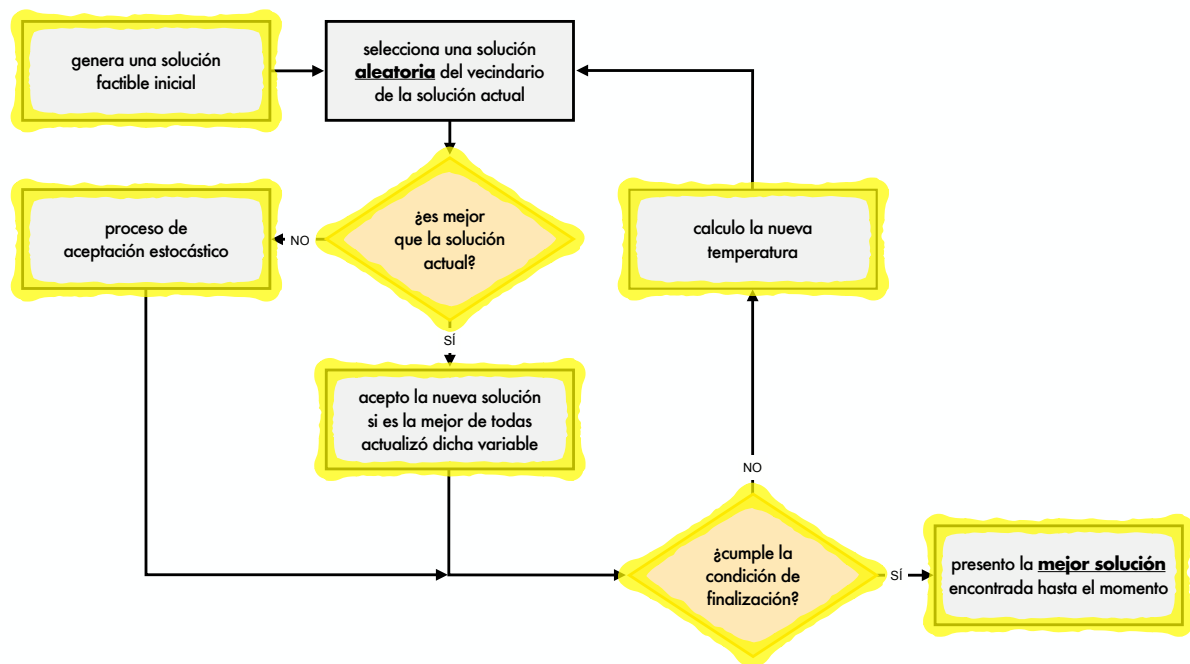
70°C	p = 0,59
60°C	p = 0,54
50°C	p = 0,48
40°C	p = 0,40
30°C	p = 0,29
20°C	p = 0,16
10°C	p = 0,02
5°C	p = 0,00

... si fijamos el delta

$$p = e^{\frac{S_n - S_a}{T}} = e^{\frac{-37}{T}}$$

71 → ●
34 → ●

resumiendo . . .



¿hasta cuando?



se enfría el sistema



máximo número de iteraciones



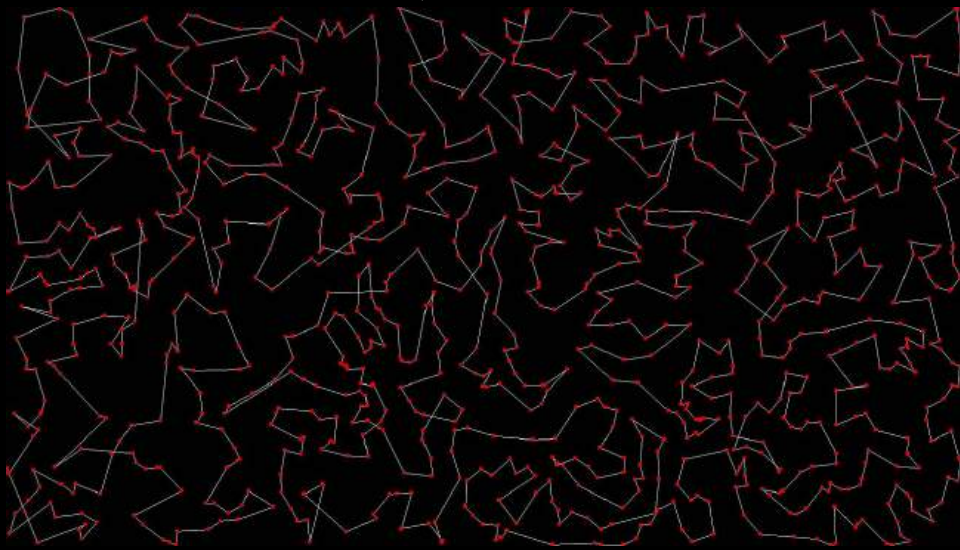
no se producen mejoras



se alcanza un tiempo máximo

... ¿y qué tan bueno es todo esto?

re-búsqueda local



temperatura →

TSP (1000 ciudades) con recocido simulado + búsqueda local

DEMO 4



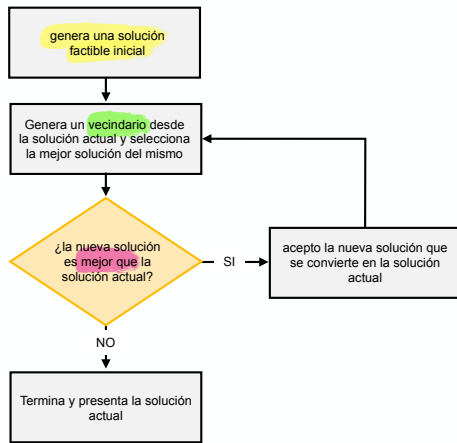
diversificación



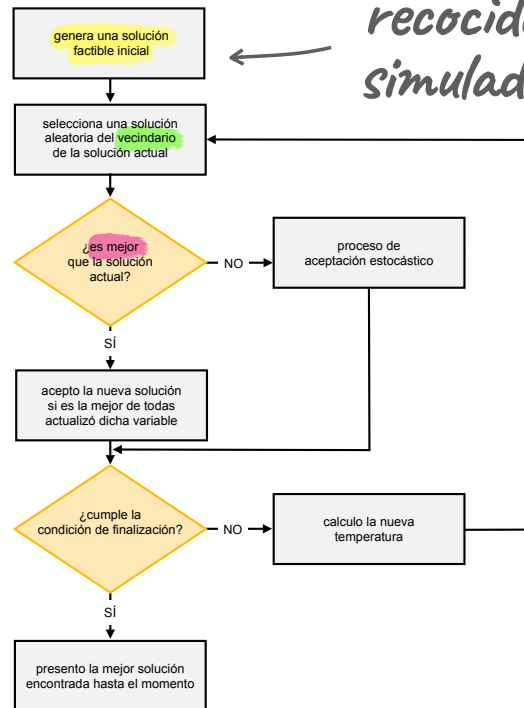
¿preguntas?



búsqueda local



recocido simulado



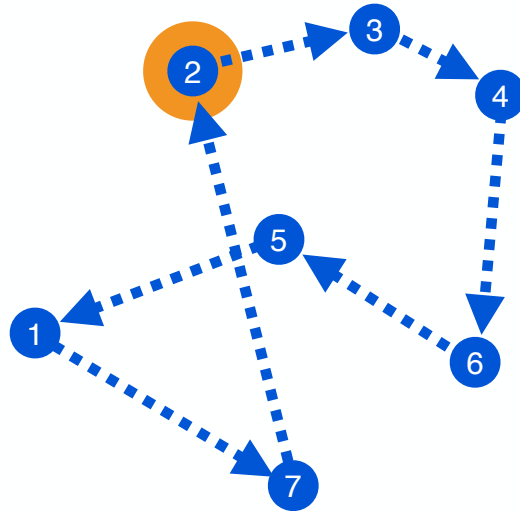
algoritmo greedy

(goloso o avaro)



soluciones
iniciales

algoritmo greedy

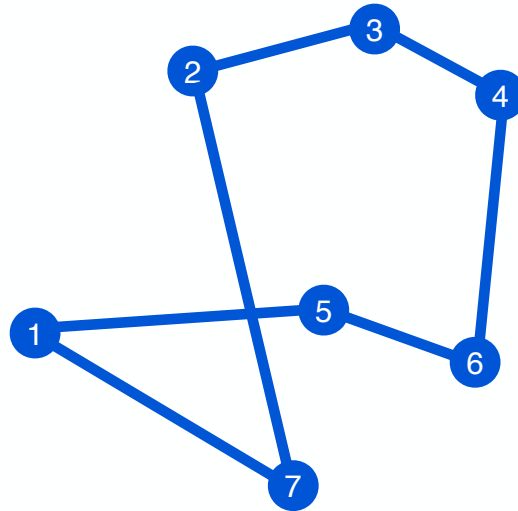


siempre busco
el punto que
mejora el
funcional

generación de vecindarios



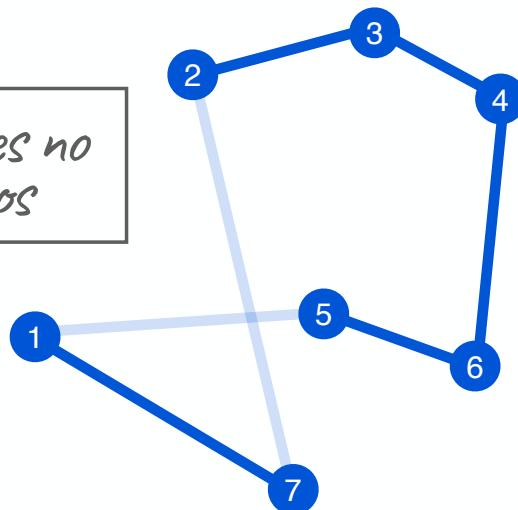
intercambio 2-opt



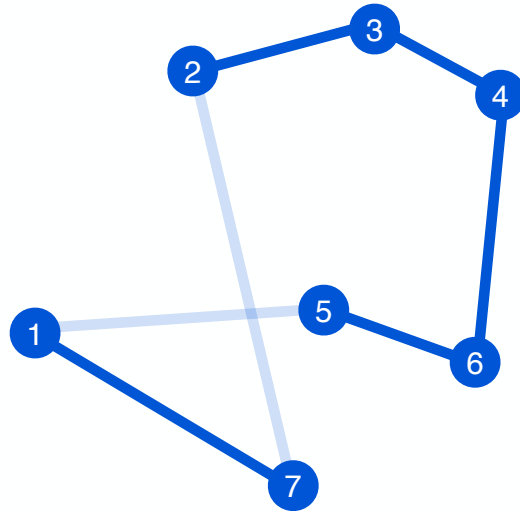
vecindarios
principalmente para
problemas de ruteo
simples

intercambio 2-opt

quito 2 enlaces no
consecutivos

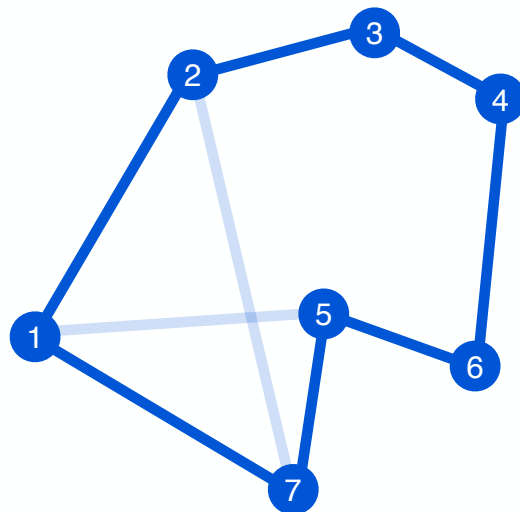


intercambio 2-opt

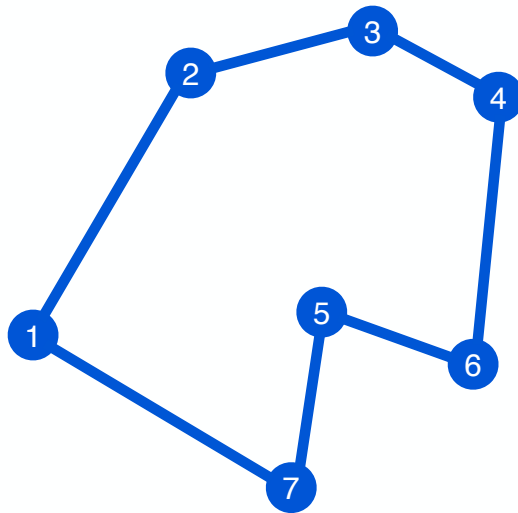


y agrego 2 nuevos
enlaces ubicados de
forma diferente

intercambio 2-opt



intercambio 2-opt



una práctica optativa ...

*pensar en un problema que hayan
resuelto en investigación operativa e
intentar resolverlo con alguna
metaheurística*



bibliografía y otros ...

[Python] Bibliotecas:

<https://docs.python.org/3/library/itertools.html>
<https://pypi.org/project/PuLP/>

[Videos]:

<https://www.youtube.com/watch?v=SC5CX8drAtU>
<https://www.youtube.com/watch?v=3TrnjUKeFg8>



Ant Colony Optimization

Autor: Dorigo - Stützle

Editorial: MIT Press



próxima clase:

clase presencial

*más sobre metaheurísticas, uso de
mapas, gis y presentación del problema
integrador*

INVESTIGACIÓN OPERATIVA SUPERIOR

¡muchas gracias!