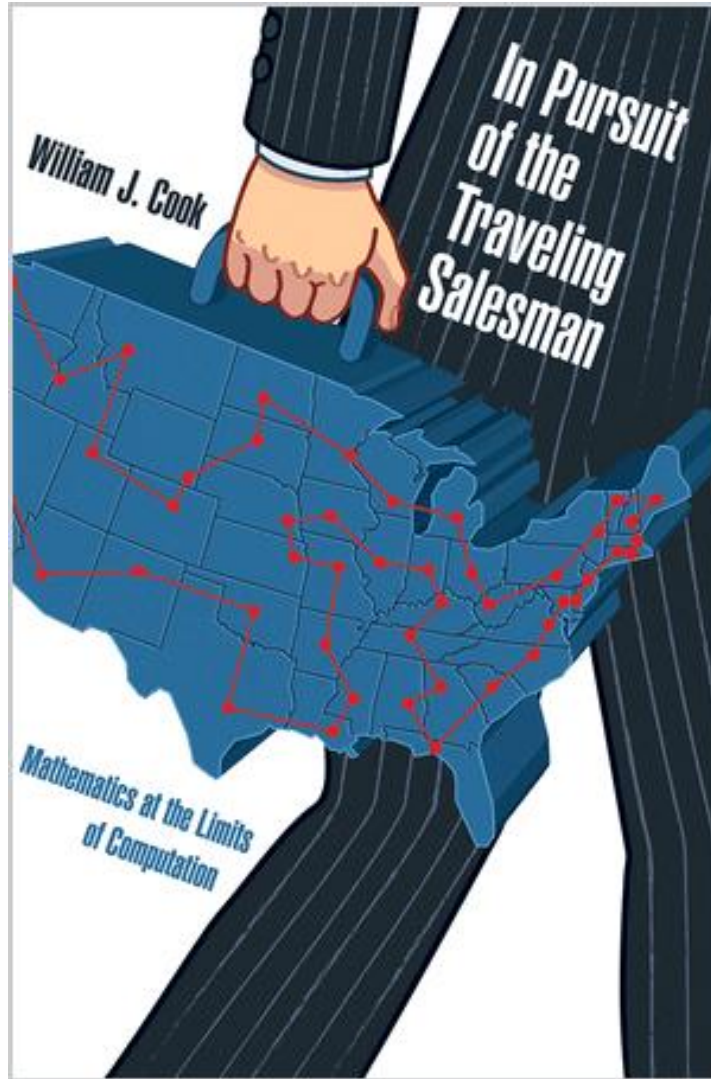


Problema TSP

Problema del agente viajero



También conocido por TSP por su nombre en inglés *Travel Salesman Problem* consiste en que, dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible para visitar cada ciudad exactamente una vez y regresar a la ciudad origen?

¿Cuántas soluciones tiene este problema? $N!$

¿Y diferentes? $\frac{(N - 1)!}{2}$

El -1 es porque dada una ruta, da igual el punto de partida. Esto reduce el número de rutas a examinar en un factor N . Por ejemplo, si para 3 ciudades la ruta óptima es 1-2-3, también lo es 2-3-1 y 3-1-2

El $1/2$ es porque no importa la dirección en que se mueva el agente. Por ejemplo si para 3 ciudades la ruta óptima es 1-2-3, también lo es 3-2-1

¿Qué significa esto? Pues que por ejemplo para 5 ciudades hay 12 rutas diferentes, para 10 ciudades 181.440, para 20 ciudades ≈ 60 mil billones, etc.

Solución mediante búsqueda exhaustiva

```
min = INF
for i = 1 to N:
    ci = i
do:
    if true:
        process(c)
while nextPermutation(c)

process(c):
    sum = 0
    for i = 1 to n-1:
        sum += distance(ci, ci+1)
    if sum < min:
        min = sum
        best = c
```

¿Cuál es el orden de complejidad de este algoritmo? $(N + 1)!$

Problema del agente viajero

Dato de interés: una de las mejores soluciones conocidas es el algoritmo Held–Karp, de programación dinámica, que resuelve el problema en $O(2^N N^2)$, requiriendo un espacio de memoria $O(N 2^N)$

Interrogante

¿Será posible solucionar un Sudoku mediante búsqueda exhaustiva?

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

		1	5		
		2	6		
	6			3	
	1			4	
		3	4		
	4			5	

2			3
			1
1			
3			2

Si, para cualquier mini-sudoku, o para uno normal siempre y cuando la cantidad de pistas iniciales sea “generosa” (5 en promedio por submatriz)