

# 0. Introducción

IA 3.2 - Programación III

2º C - 2025

Dr. Mauro Lucci

# Motivación



# Transporte de última milla (last mile)

— — —

- Última etapa en la logística del **transporte**.
- Desde un centro de distribución hasta el cliente final.
- **Objetivo.** Construir la ruta del camión (o los camiones).
  1. Comienza en el centro de distribución.
  2. Visita una vez a cada cliente, en algún orden.
  3. Regresa al centro de distribución.

1 h 9

3 h 57

1 h 11

Facultad de Ciencias Exactas, Ingeniería y Ciencias

0

Facultad de Ciencias Económicas y Estadística

1

La Siberia - Centro Universitario Rosario

2

Facultad de Ciencias Médicas - Universidad Nacional de Rosario

3

Universidad Nacional de Rosario, Maipú

4

Facultad de Ciencias Exactas, Ingeniería y Ciencias

0

Agregar destino

Opciones

Enviar al teléfono instrucciones sobre cómo llegar

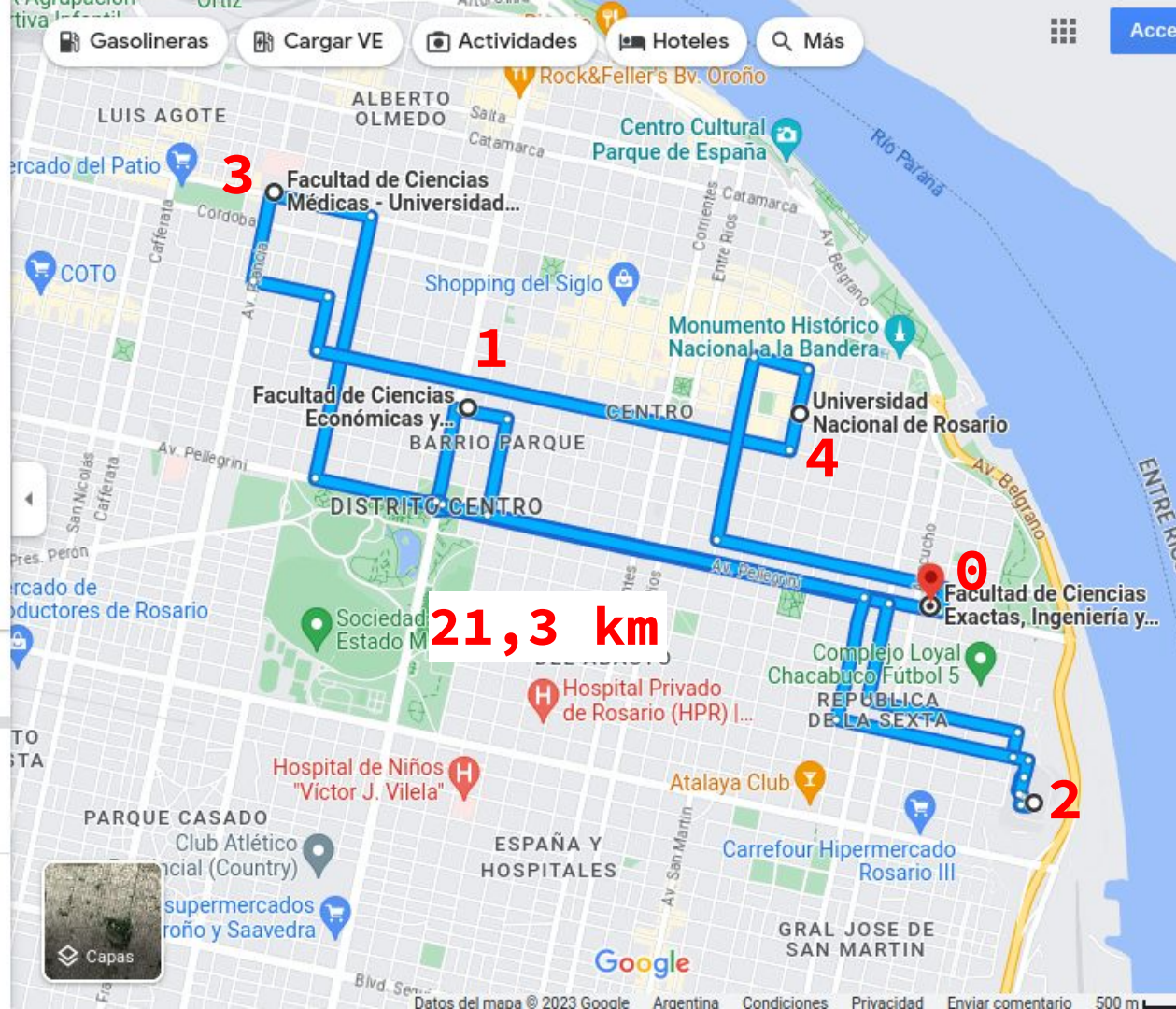
por Av. Pellegrini

1 h 9 min

1 h 9 min sin tráfico

21.3 km

Detalles



48 min

2 h 50

52 min

Facultad de Ciencias Exactas, Ingeniería

0

La Siberia - Centro Universitario Rosario

2

Universidad Nacional de Rosario, Maipú

4

Facultad de Ciencias Económicas y Estadística

1

Facultad de Ciencias Médicas - Universidad Nacional de Rosario

3

Facultad de Ciencias Exactas, Ingeniería

0

+

Agregar destino

Opciones

Enviar al teléfono instrucciones sobre cómo llegar

por Colón y Cerrito

48 min

48 min sin tráfico

14.9 km

Detalles

Map showing a route in Rosario, Argentina, with stops numbered 0, 1, 2, 3, and 4. The route is highlighted in blue. The total distance is 14.9 km.

Key locations and landmarks visible on the map include:

- Facultad de Ciencias Exactas, Ingeniería (Stop 0)
- Facultad de Ciencias Médicas - Universidad Nacional de Rosario (Stop 3)
- Shopping del Siglo (Stop 1)
- Monumento Histórico Nacional a la Bandera (Stop 4)
- Parque Casado (Stop 2)
- Río Paraná
- Centro Cultural Parque de España
- Shopping del Siglo
- Monumento Histórico Nacional a la Bandera
- Parque Urquiza
- Complejo Loyal Chacabuco Fútbol Club
- Hospital Privado de Rosario (HPR)
- Hospital de Niños "Víctor J. Vilela"
- Atalaya Club
- Carrefour Hipermercado Rosario III
- GRAL JOSE DE SAN MARTIN
- ESPAÑA Y HOSPITALES
- PARQUE CASADO
- Club Atlético (Country)
- supermercados Oroño y Saavedra
- Capas







# Algoritmo de fuerza bruta

— — —

- Iterar sobre todas las rutas posibles e ir manteniendo en memoria la más corta.
- Para 4 clientes, las posibles combinaciones son:

- |     |             |     |             |
|-----|-------------|-----|-------------|
| 1.  | 1, 2, 3, 4. | 13. | 3, 1, 2, 4. |
| 2.  | 1, 2, 4, 3. | 14. | 3, 1, 4, 2. |
| 3.  | 1, 3, 2, 4. | 15. | 3, 2, 1, 4. |
| 4.  | 1, 3, 4, 2. | 16. | 3, 2, 4, 1. |
| 5.  | 1, 4, 2, 3. | 17. | 3, 4, 1, 2. |
| 6.  | 1, 4, 3, 2. | 18. | 3, 4, 2, 1. |
| 7.  | 2, 1, 3, 4. | 19. | 4, 1, 2, 3. |
| 8.  | 2, 1, 4, 3. | 20. | 4, 1, 3, 2. |
| 9.  | 2, 3, 1, 4. | 21. | 4, 2, 1, 3. |
| 10. | 2, 3, 4, 1. | 22. | 4, 2, 3, 1. |
| 11. | 2, 4, 1, 3. | 23. | 4, 3, 1, 2. |
| 12. | 2, 4, 3, 1. | 24. | 4, 3, 2, 1. |

# Factorial

---

Para  $n$  clientes, el número de posibles rutas es:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$$

Este número es conocido como **factorial** de  $n$ .



**Ejemplo.**

$$4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$



# Performance del algoritmo de fuerza bruta

---

- **Consumo de tiempo:**  $n!$  iteraciones
- **Consumo de memoria:** 2 rutas (la actual y la mejor).

¿Habr  un algoritmo  
mejor que el de  
fuerza bruta? 🤔

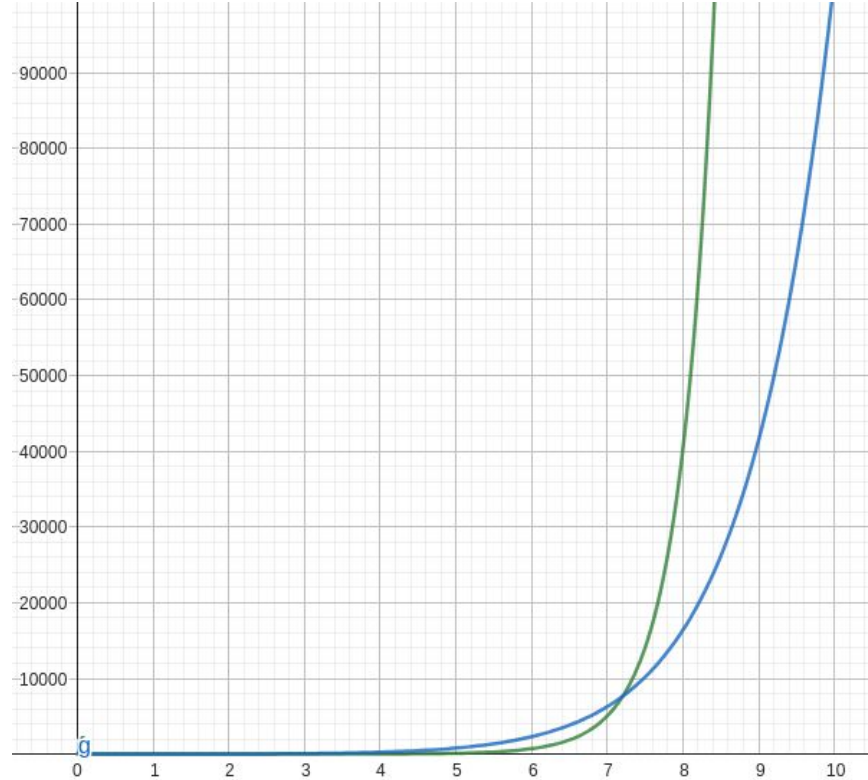
# Algoritmo de Held–Karp

— — —

- Desarrollado por Held y Karp en 1962.
- Está basado en **programación dinámica**.
- **Consumo de tiempo:**  $n^2 \cdot 2^n$
- **Consumo de memoria:**  $n \cdot 2^n$

# Crecimiento exponencial

— — —



$$f(n) = n!$$

$$g(n) = n^2 \cdot 2^n$$



# Explosión combinatoria

— — —

Disponiendo de una computadora capaz de evaluar 1M de rutas por segundo:

<i>n</i>	Fuerza bruta	Held-Karp
10	3,6 segundos	0,1 segundos
15	363 horas	7,4 segundos
20	77146 años	7 minutos
25	$4,9 \times 10^{11}$ años	5,8 horas
30	$8,4 \times 10^{18}$ años	11 días

Edad del universo:  
 $13,7 \times 10^9$  años.

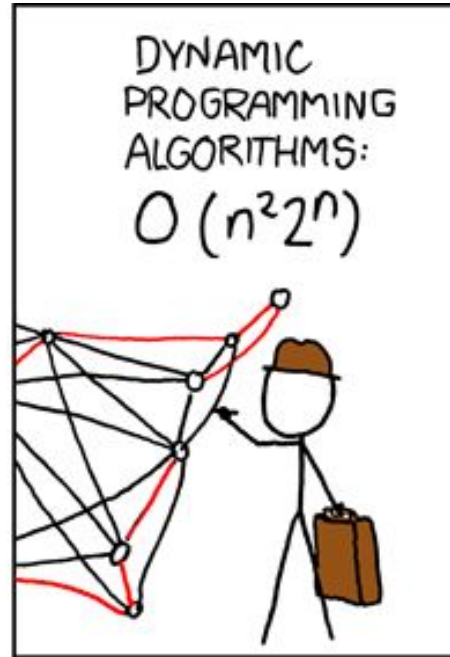
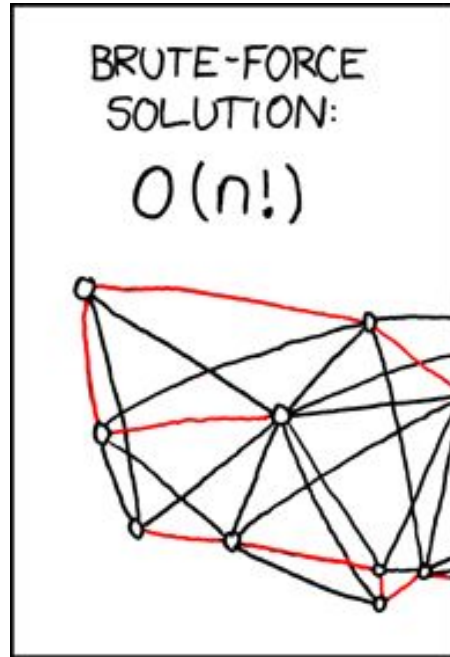


# Soluciones en tiempo real

— — —

- En Amazon, cada camión visita típicamente entre 32 y 237 clientes por día, con un promedio de 148 clientes.
  - ✓ <https://www.math.uwaterloo.ca/tsp/amz/index.html>
- En estos casos, encontrar la ruta más corta con el algoritmo de fuerza bruta o el de Held-Karp tomaría un tiempo impráctico.
- Además, la ruta más corta no siempre es la preferida.
  - ✓ Facilidad de navegación.
  - ✓ Estacionamiento.
  - ✓ Congestión de tráfico.
  - ✓ Zonas peligrosas.
- El desafío es encontrar en pocos minutos rutas relativamente cortas que se desempeñen bien según ciertas métricas.





¿Cómo hacen las empresas de entrega de paquetería para resolver este tipo de problemas?

# Organización

# Cátedra

— — —

- **Mauro Lucci**

[mlucci@fceia.unr.edu.ar](mailto:mlucci@fceia.unr.edu.ar)

- **Gonzalo Longo**

[gonzalolongo.fceia@gmail.com](mailto:gonzalolongo.fceia@gmail.com)

- **Salvador Sanchez**

[salvadorsanchez1832005@gmail.com](mailto:salvadorsanchez1832005@gmail.com)

# Calendario académico

	AGOSTO				SEPTIEMBRE					OCTUBRE				NOVIEMBRE				DICIEMBRE			
	4	11	18	25	1	8	15	22	29	6	13	20	28	3	10	17	24	1	8	15	22
Nº de semana clases			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
Clases																					
Cierre de promoción (*)																					
Consultas																					
Exámenes																					
Flotantes																					

- [https://web.fceia.unr.edu.ar/images/PDF/Calendario\\_Acadmico/2025/res\\_910\\_24\\_calendario\\_academico.pdf](https://web.fceia.unr.edu.ar/images/PDF/Calendario_Acadmico/2025/res_910_24_calendario_academico.pdf)
- Clases. 25/8 al 5/12 (16 semanas)
- Cierre. 8/12 al 19/12 (2 semanas)

# Horarios

— — —

- **Miércoles de 18 a 20.**

Teoría virtual: [meet.google.com/msv-ubac-wth](https://meet.google.com/msv-ubac-wth)

- **Viernes de 18 a 20.**

Lab. de básica en sede Pellegrini (a mitad de pasillo en el 1<sup>er</sup> piso).

- **Horarios semanales de consultas.**

**Confirmar asistencia por correo hasta el día anterior.**



# Comunicación

— — —

- **Campus virtual de FCEIA.**

<https://campusv.fceia.unr.edu.ar/course/view.php?id=473>

# Actividades con evaluación

— — —

Contribución a  
la **nota final**

- **Actividades de laboratorio.**

- 1 TP grupal con entrega y defensa.

}

**33.3 %**

- **Actividades prácticas/teóricas.**

- 2 parciales.

}

**66.6 %**

- **Recuperatorios.**

- Se puede recuperar **el TP y hasta 1 parcial.**

# Condiciones

— — —

- Alcanzarán la condición de **promoción directa** quienes hayan aprobado todas las instancias evaluativas (de forma directa o por medio de recuperatorios).
- De lo contrario, la condición será de **libre**.

# ¿De qué trata el curso?

---

- Herramientas para resolver problemas que no tienen una *solución analítica directa*, sino que su resolución involucra **buscar** una *solución aceptable* entre **muchas** o incluso **infinitas** alternativas.
- Aprender a **formular** un problema de búsqueda.
- Conocer diferentes **algoritmos de búsqueda**, que toman como entrada un problema y retornan una solución.
- Compararlos según ciertos **criterios de evaluación**.
- Implementarlos en **Python**.

# Contenidos sintéticos

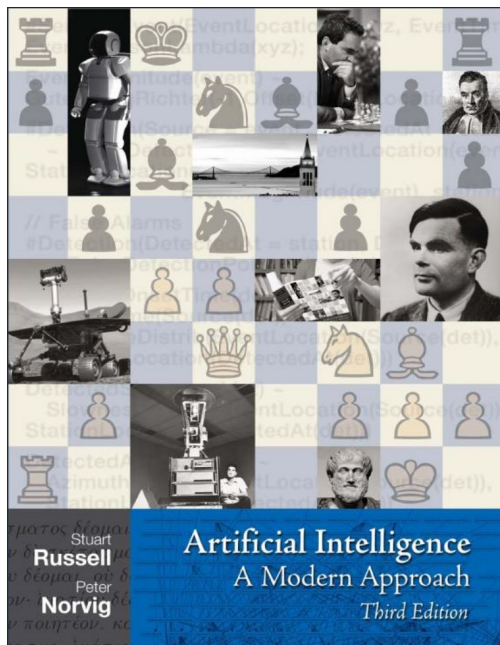
— — —

- **Unidad 1.** Resolución de problemas mediante búsqueda.
- **Unidad 2.** Estrategias de búsqueda no informadas.
- **Unidad 3.** Estrategias de búsqueda informadas.
- **Unidad 4.** Búsqueda entre adversarios.
- **Unidad 5.** Búsqueda local.
- **Unidad 6.** Problemas de satisfacción de restricciones.
- **Unidad 7.** Búsqueda inspirada en la naturaleza.
- **Unidad 8.** Búsqueda en la actualidad.

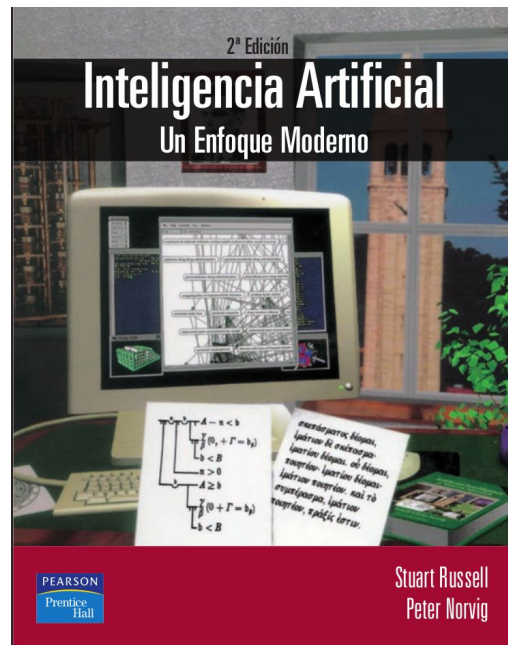


# Bibliografía

— — —



(2009)



(2003)

# Aplicaciones


# Problemas de factibilidad y optimización

— — —

- **Problema de factibilidad.** El objetivo es hallar cualquier **solución factible**, es decir, una que cumpla con todos los requerimientos del problema.
- **Problema de optimización.** Las soluciones tienen costos (o ganancias) y el objetivo es hallar una **solución óptima**, es decir, una que cumpla con todos los requerimientos y cuyo costo sea mínimo (o su ganancia sea máxima).

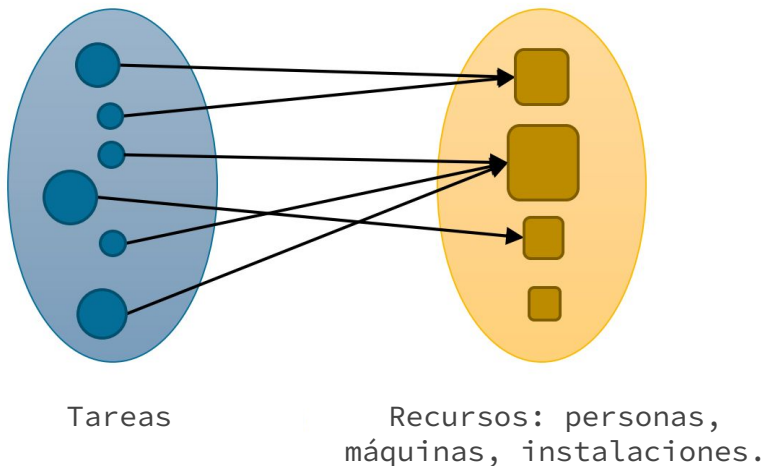
# Variaciones

— — —

- En el curso trabajaremos con problemas de factibilidad y optimización muy variados, provenientes de la **industria**, **academia**, **juegos**, etc.
- En la práctica, suelen aparecer nuevos requerimientos, que derivan en **variaciones** de los problemas originales.
- Las variaciones pueden hacer más **fácil/difícil** un problema.
- A continuación, se mencionan algunos ejemplos .

# 1. Problemas de asignación

---



Es una amplia familia de problemas que involucran **asignar** los **recursos** disponibles de la mejor forma posible para cumplir ciertas **tareas**.



# 1.1 Problema de asignación de aulas

## Classroom assignment problem

— — —



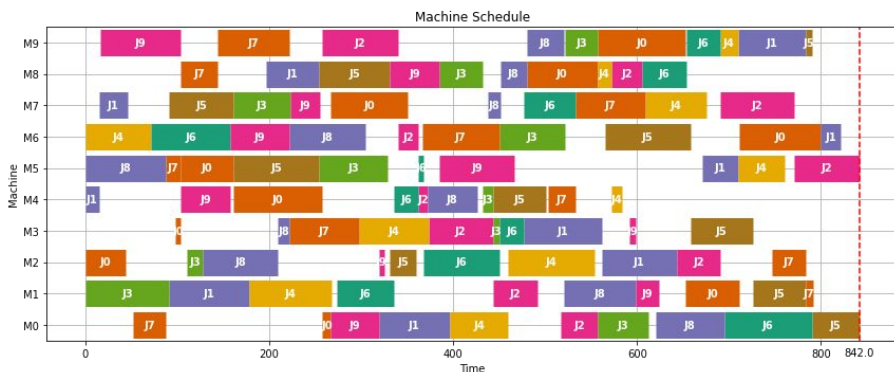
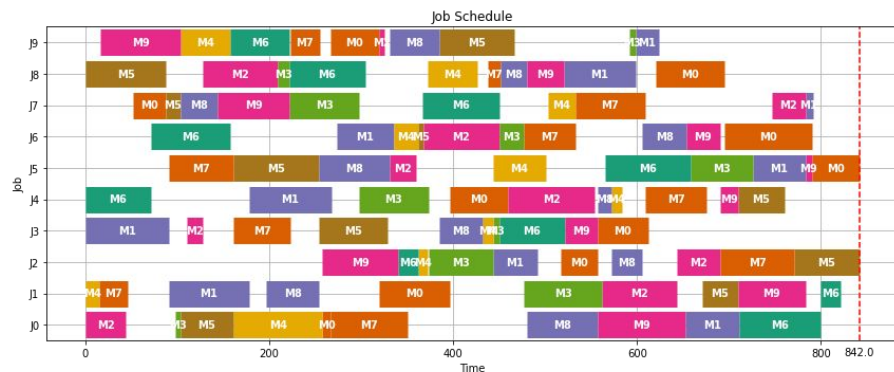
**Entrada.** Clases con horarios y cantidades de inscriptos, y aulas con capacidades.

**Objetivo.** Asignar un aula a cada clase, de forma tal que ningún aula tenga más de una clase en simultáneo y sin exceder las capacidades de las aulas.

# 1.2 Problema de programación de trabajos

## Job scheduling problem

— — —



**Entrada.** Trabajos con distinta duración y máquinas.

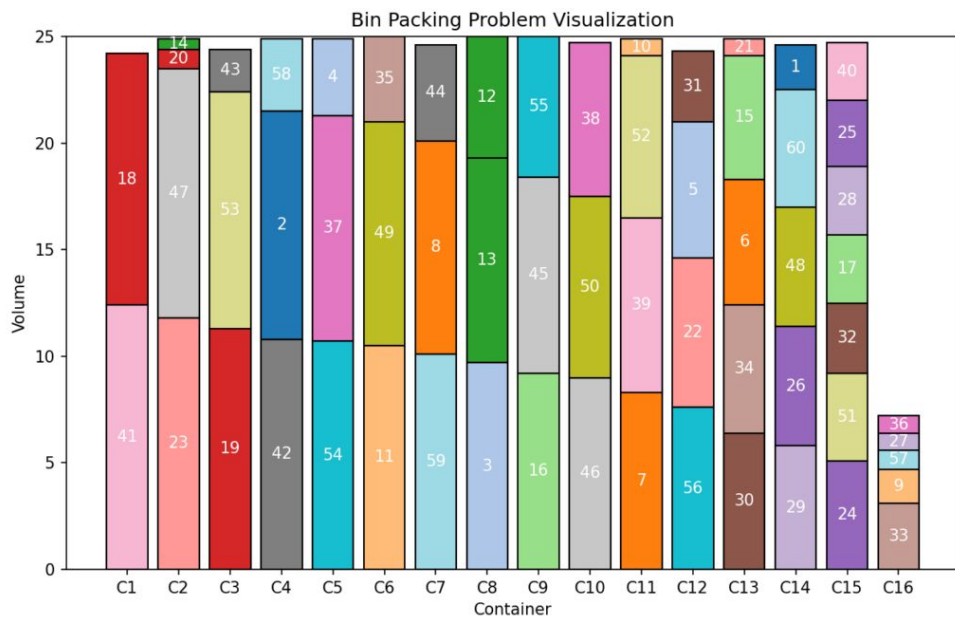
**Objetivo.** Asignar a cada trabajo una hora de inicio y una máquina, de forma tal que cada máquina realice un único trabajo a la vez.

**Variaciones.** Cada trabajo puede constar de una o múltiples fases que pueden requerir ser ejecutadas en paralelo o en secuencia.

# 1.3 Problema de empaquetado en contenedores

## Bin packing problem

— — —



**Entrada.** Conjunto de objetos con diferentes dimensiones.

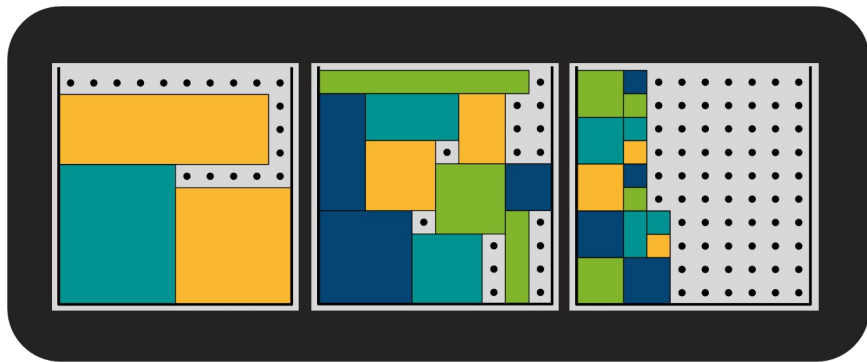
**Objetivo.** Asignar cada objeto a un contenedor, sin exceder la capacidad del contenedor, y minimizando el número de contenedores utilizados.

# 1.3 Problema de empaquetado en contenedores

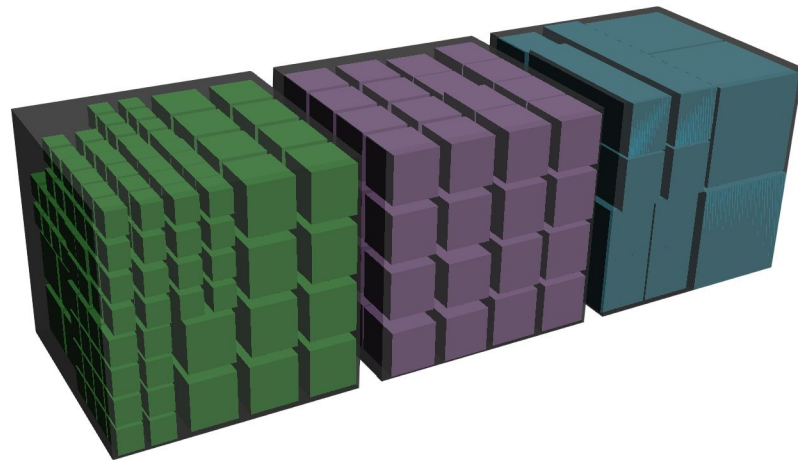
Bin packing problem

— — —

Variaciones.



2D



3D

## 2. Problemas de ruteo de vehículos

— — —



Es una amplia familia de problemas que involucran diseñar **rutas** óptimas para que una flota de **vehículos** pueda cumplir con pedidos de **clientes**.

## 2.1 Problema de primera/última milla

### First/Last mile problem

— — —



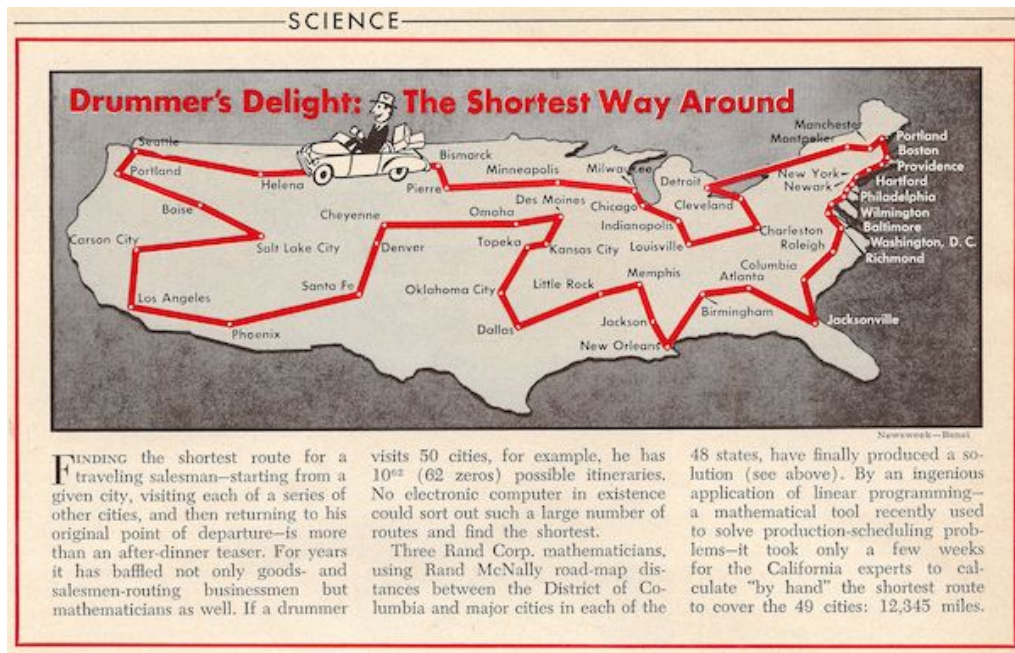
- **Primera milla:** desde los productores al centro de distribución.
- **Última milla:** desde el centro de distribución a los clientes.



## 2.2 Problema del viajante

### Travelling Salesman Problem (TSP)

— — —



**Entrada.** Conjunto de ciudades.

**Objetivo.** Hallar un tour de distancia mínima que recorra exactamente una vez cada ciudad y regrese al origen.

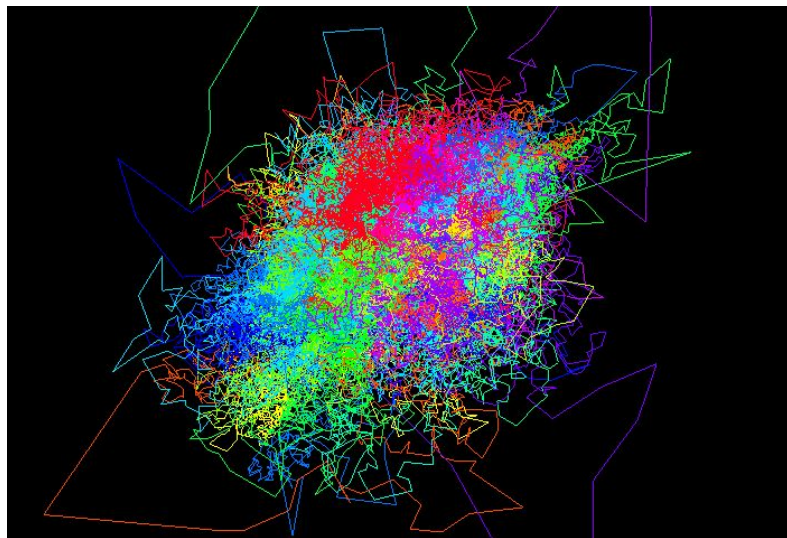
## 2.2 Problema del viajante

### Travelling Salesman Problem (TSP)

— — —

¿Si en vez de ciudades queremos visitar estrellas? 🌠

**TSP en 3D.**



Tour mínimo para 2.079.471  
estrellas.

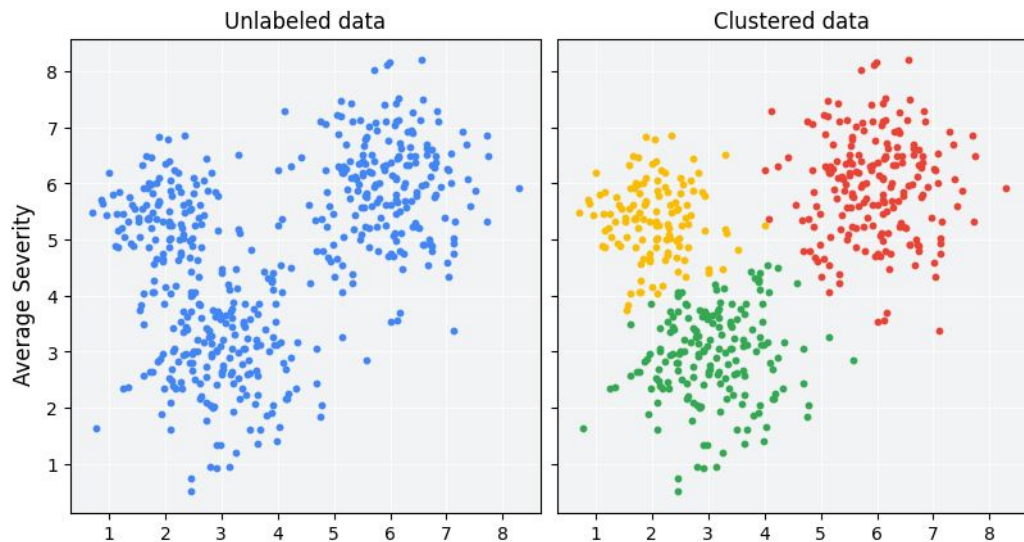
<https://www.math.uwaterloo.ca/tsp/star/gaia1.html>

**Desafíos abiertos.** 10M y 100M de  
estrellas.



# 3. Clustering

— — —



Problemas asociados a **particionar** un conjunto de objetos en **clusters** (grupos) de forma tal que los objetos dentro de un mismo cluster tengan más **similitud** entre ellos que con los de los otros clusters.

## 4. Juegos/acertijos

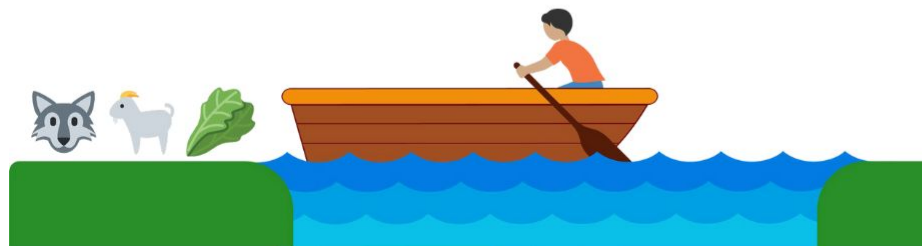
— — —

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

4.1 Sudoku

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

4.2  
Criptoaritmo



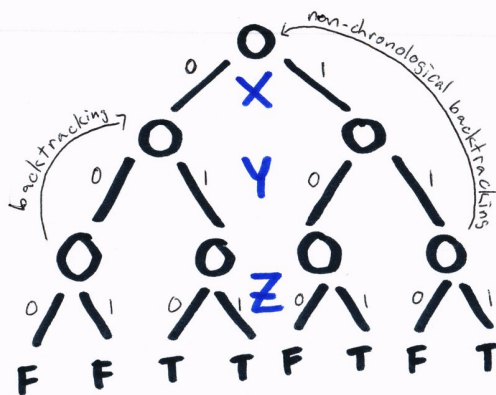
4.3 Cruce de río

# 5. Problemas fundamentales de las ciencias exactas

---

SAT: (X or Y) and (Z or not X)

"conjunctive normal form"  
with literals and clauses



## 5.1 Problema de satisfacibilidad booleana Boolean Satisfiability Problem (SAT)

**Entrada.** Fórmula booleana.

**Objetivo.** Encontrar, si existe, una asignación de valores de verdad a literales, de forma tal que la fórmula evalúe a **TRUE** (se satisfaga).

# Problemas difíciles

— — —

- Estos problemas tienen en común que la cantidad de asignaciones posibles (combinaciones) **crece muy rápidamente** (en general exponencialmente) con pequeños incrementos en el tamaño de la entrada.
- La mayoría pertenece a la clase de problemas **NP-difícil** (el SAT fue el primero en ser clasificado en esta clase).
- Es decir, **no se conocen** algoritmos que resuelvan **cualquier** entrada en tiempo razonable, y hay consenso de que no existen, aunque no fue probado.
- **Conjetura P vs NP.** Demostrar la existencia o no de algoritmos eficientes para resolver este tipo de problemas difíciles. Es uno de los 7 problemas del milenio, cuya resolución vale US\$1M 🏆 (hasta ahora, el único resuelto es la Conjetura de Poincaré).