**1. Explicación del Problema del Viajante**

El **Problema del Viajante (TSP)** es un problema clásico en informática y matemáticas. Consiste en lo siguiente:

* Un viajante debe visitar un conjunto de ciudades, comenzando y terminando en la misma ciudad.
* El objetivo es encontrar la ruta más corta posible que visite cada ciudad exactamente una vez.
* Es un problema de optimización combinatoria, ya que implica encontrar la mejor combinación de rutas entre todas las posibles.

El TSP es un problema **NP-duro**, lo que significa que no se conoce un algoritmo eficiente para resolverlo en todos los casos, especialmente cuando el número de ciudades es grande.

**2. Cálculo de las combinaciones posibles para 15 ciudades**

En el TSP, el número de rutas posibles es el número de **permutaciones** de las ciudades. Para n*n* ciudades, el número de rutas posibles es (n−1)!/2(*n*−1)!/2, donde:

* (n−1)!(*n*−1)! representa las permutaciones de las ciudades (se divide por 2 porque la ruta puede recorrerse en dos direcciones).

Para **15 ciudades**:

Nuˊmero de rutas=(15−1)!2=14!2Nuˊmero de rutas=2(15−1)!​=214!​

Calculamos 14!14!:

14!=87,178,291,20014!=87,178,291,200

Dividimos por 2:

87,178,291,2002=43,589,145,600287,178,291,200​=43,589,145,600

Por lo tanto, hay **43,589,145,600 rutas posibles** para 15 ciudades.

**3. Métodos heurísticos para resolver el TSP**

En la página de Wikipedia mencionada, se describen varios métodos heurísticos para resolver el TSP. Aquí tienes un resumen de los principales:

**a) Heurísticas constructivas**

Estos métodos construyen una solución paso a paso:

* **Vecino más cercano (Nearest Neighbor):**
  + Comienza en una ciudad y en cada paso se mueve a la ciudad más cercana no visitada.
  + Es rápido pero no siempre produce la mejor solución.
* **Inserción más barata (Cheapest Insertion):**
  + Comienza con un subtour pequeño y en cada paso inserta la ciudad que aumenta menos la longitud total del recorrido.
  + Es más lento que el vecino más cercano pero suele dar mejores resultados.

**b) Heurísticas de mejora**

Estos métodos parten de una solución inicial y tratan de mejorarla:

* **2-opt:**
  + Elimina dos aristas del recorrido y reconecta los segmentos de manera diferente para reducir la longitud total.
  + Se repite hasta que no se puedan hacer más mejoras.
* **3-opt:**
  + Similar al 2-opt, pero elimina tres aristas y reconecta los segmentos.
  + Es más lento pero puede producir mejores soluciones.

**c) Metaheurísticas**

Estos métodos son más avanzados y exploran el espacio de soluciones de manera más inteligente:

* **Algoritmos genéticos:**
  + Usan conceptos de evolución biológica (selección, cruce, mutación) para generar soluciones cada vez mejores.
* **Recocido simulado (Simulated Annealing):**
  + Explora soluciones aleatorias, permitiendo ocasionalmente empeorar la solución para evitar quedarse atrapado en óptimos locales.
* **Colonia de hormigas (Ant Colony Optimization):**
  + Inspirado en el comportamiento de las hormigas, utiliza "feromonas" para guiar la búsqueda de soluciones.

**d) Heurísticas basadas en divide y vencerás**

* **Algoritmo de Christofides:**
  + Es un algoritmo aproximado que garantiza una solución dentro del 50% del óptimo para el TSP métrico (donde se cumple la desigualdad triangular).
  + Combina árboles de expansión mínima y emparejamientos perfectos.