Modelado, optimización y simulación
ISIS-3302

Proyecto Etapa 3 - Implementacion - algoritmo genetico

Integrantes:

Juan David Duarte – 202215070
Juan Nicolas Suarez – 202222678
David Santiago Carrillo – 202225276

Índice

1. Instancias evaluadas

- **Caso Base**: 15 clientes.
- **Instancia M**: 19 clientes.
- **Instancia L**: 24 clientes.

El costo por kilómetro es $c = c_m + c_c = 0{,}11$.

2. Métodos

1. **Modelo exacto**: Pyomo 6.7 + Gurobi 10.
2. **Metaheurístico**: Algoritmo Genético (GA) pob $= 50$, $g = 600$, $P_c = 0{,}85$, $P_m = 0{,}10$, seed $= 0$.

3. Resultados comparativos

Instancia	**Método**	**Clientes**	Costo	Tiempo [s]	Mem. [MB]	GAP [%]
Caso Base	Pyomo	15	17,16	41,78	3,50	0,00
	GA	15	18,09	3,55	0,10	5,40
Instancia M	Pyomo	19	21,74	1,00 270.18	5,20	0,00
	GA	19	23,42	4,47	0,10	7,71
Instancia L	Pyomo	24	25,24	1,00 269.22	7,10	0,00
	GA	24	29,41	3,83	0,10	16,50

Cuadro 1: Desempeño de Pyomo (exacto) y GA en las tres instancias.

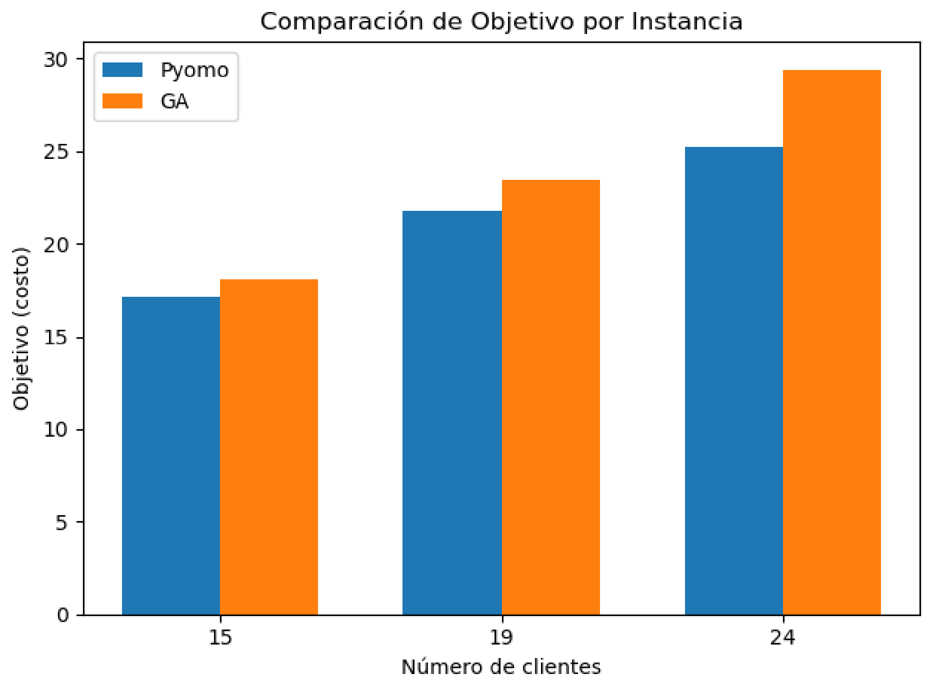

Figura 1: Costo final de Pyomo vs GA para cada tamaño de instancia.

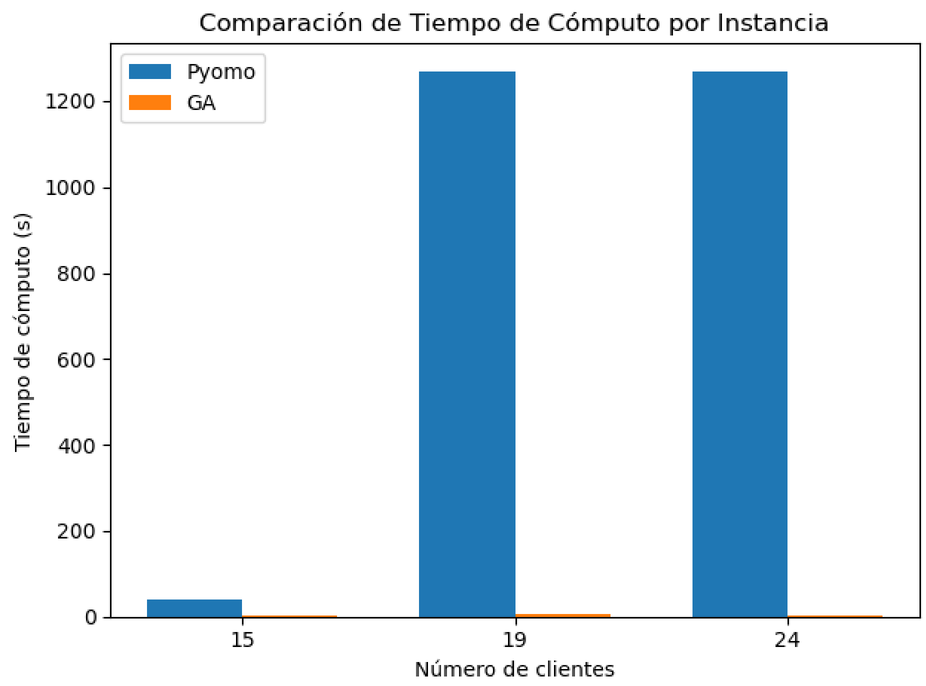

Figura 2: Tiempo de cómputo de Pyomo vs GA (escala lineal) por instancia.

4. Análisis

- **Escalabilidad temporal**. Pyomo pasa de 42,00 s (15 clientes) a >20,00 min (19–24 clientes). El GA resuelve las tres instancias en menos de 5,00 s.
- **Calidad**. La brecha GA–óptimo crece moderadamente: 5,4 % $\rightarrow$ 7,7 % $\rightarrow$ 16,5 % al aumentar el tamaño, manteniéndose útil para decisiones rápidas.
- **Memoria**. El GA se mantiene en $\approx$ 0,10 MB gracias a la codificación compacta, mientras que el modelo exacto crece a 7,10 MB.

5. Detalle de demanda atendida

15 clientes (Pyomo)

```
Veh 1: 0    Veh 2: 56   Veh 3: 57  Veh 4: 0
Veh 5: 58   Veh 6: 60   Veh 7: 0   Veh 8: 15
```

19 clientes (Pyomo)

```
Veh 1: 43  Veh 2: 0   Veh 3: 59   Veh 4: 54
Veh 5: 0   Veh 6: 40  Veh 7: 62   Veh 8: 61
```

24 clientes (Pyomo)

```
Veh 1: 57  Veh 2: 62  Veh 3: 59  Veh 4: 27
Veh 5: 60  Veh 6: 0   Veh 7: 60  Veh 8: 52
```

6. Implementacion en pyomo: CVRP Estándar sin Ventanas de Tiempo

6.1. Descripción

Este primer caso corresponde a un problema de ruteo de vehículos capacitado (CVRP) clásico. El objetivo es planificar las rutas de una flota homogénea de vehículos terrestres, partiendo desde un centro de distribución (CD), para satisfacer la demanda de un conjunto de clientes geográficamente distribuidos, minimizando el costo total de operación.

En este escenario no se consideran ventanas de tiempo para las entregas ni la posibilidad de reabastecimiento. Cada cliente debe ser atendido una sola vez y la demanda total asignada a un vehículo no debe superar su capacidad. Se deben cumplir las restricciones de continuidad de ruta y eliminación de subtours.

Los datos de entrada incluyen:

- Coordenadas geográficas del CD y de los clientes.
- Demandas individuales por cliente.
- Vehículos con identificadores, capacidad y autonomía (rango máximo).

Todos los contenidos del caso base (Caso 1), se encuentran en la carpeta llamada `caso1`.

6.2. Formulación Matemática

Sea:

- V: Conjunto de nodos (CD y clientes).
- $v \subset V$: Conjunto de clientes (excluyendo CD).
- K: Conjunto de vehículos.
- d_{ij}: Distancia entre nodo i y nodo j.
- q_i: Demanda del cliente i.
- Q_k: Capacidad del vehículo k.
- R_k: Rango máximo del vehículo k.
- $x_{ijk} \in \{0, 1\}$: Variable binaria que indica si el vehículo k va del nodo i al j.
- g_{ik}: Carga acumulada transportada por el vehículo k hacia el cliente i.

Función objetivo:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} x_{ijk} \cdot d_{ij} \cdot (C_{\text{mant}} + C_{\text{cons}})$$

Sujeto a:

$$
\begin{aligned}
&\sum_{j\in V}\sum_{k\in K} x_{ijk} = 1 && \forall i \in v \quad \text{(cada cliente es visitado una vez)}\\
&\sum_{j\in V} x_{jik} - \sum_{j\in V} x_{ijk} = 0 && \forall i \in v, \forall k \in K \quad \text{(flujo conservado)}\\
&g_{\text{CD},k} = 0 && \forall k \in K \quad \text{(carga inicial en CD)}\\
&g_{ik} \geq g_{jk} + y_{jk} - M(1 - x_{ijk}) && \forall i, j \in v, i \neq j, \forall k \in K\\
&\sum_{i\in v} g_{ik} \leq Q_k && \forall k \in K\\
&\sum_{k\in K} g_{ik} = q_i && \forall i \in v\\
&\sum_{i\in V}\sum_{j\in V} d_{ij} \cdot x_{ijk} \leq R_k && \forall k \in K\\
&x_{iik} = 0 && \forall i \in V, \forall k \in K \quad \text{(prohibición de lazos)}\\
&u_i - u_j + (n-1)\cdot \sum_{k\in K} x_{ijk} \leq n - 2 && \forall i, j \in v, i \neq j\\
&\sum_{j\in V} x_{\text{CD},jk} = 1 && \forall k \in K \quad \text{(sale del CD)}\\
&\sum_{i\in V} x_{i,\text{CD},k} = 1 && \forall k \in K \quad \text{(regresa al CD)}
\end{aligned}
$$

6.3. Diagrama de gantt caso 1

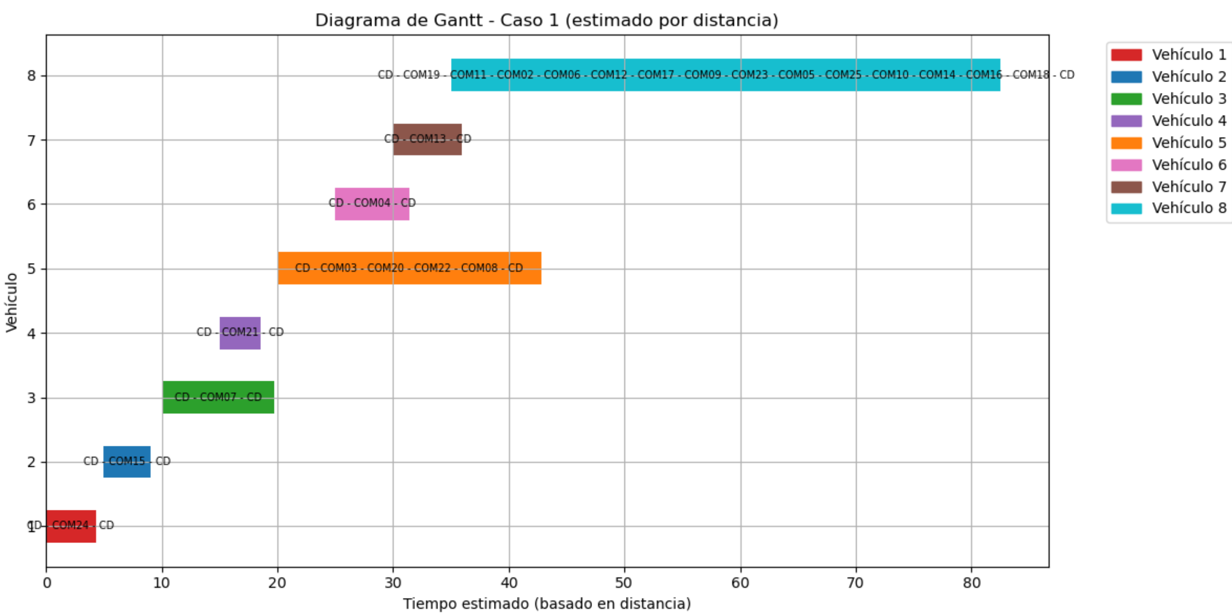

Figura 3: Enter Caption

6.4. Mapa del Caso 1

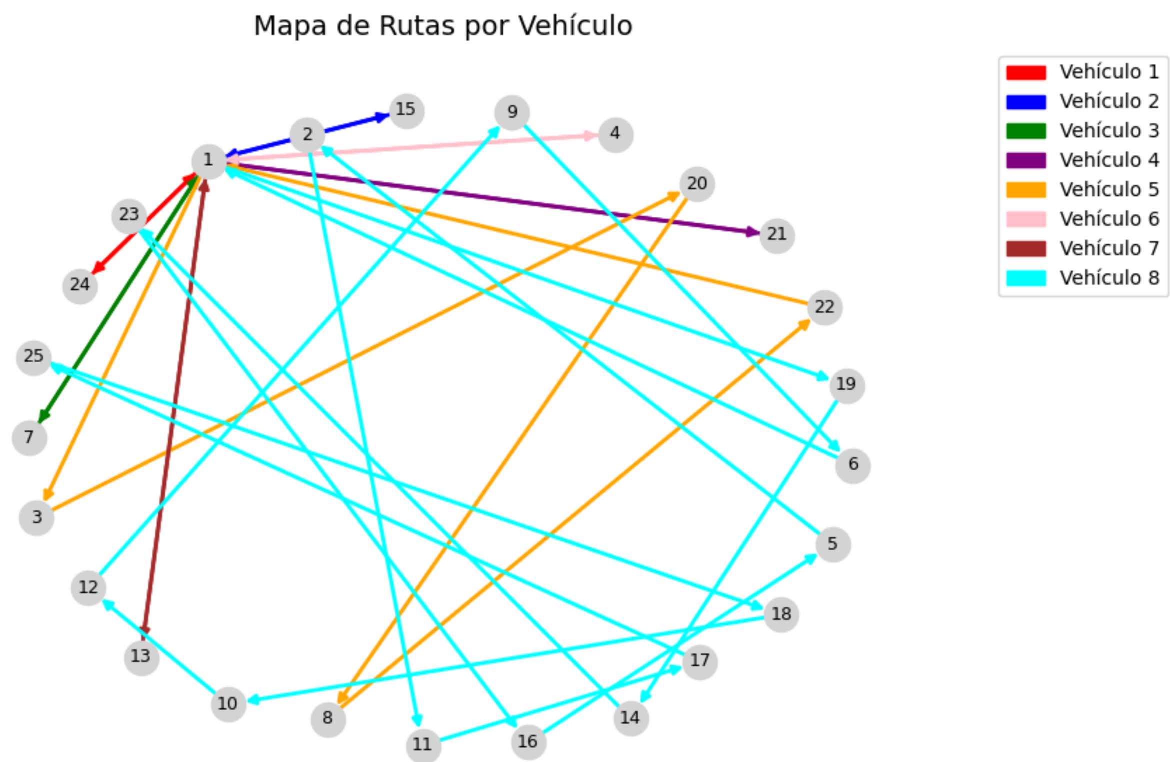

Figura 4: Enter Caption

6.5. Archivo de Verificación caso 1

El archivo `verificacion_caso1.csv` contiene información sobre las rutas realizadas, los clientes atendidos, la carga transportada, distancia recorrida y costo operativo total por vehículo. Este archivo se genera automáticamente desde el código de resolución tras ejecutar el método `solve()` y puede usarse para auditar el cumplimiento del modelo.

7. Conclusiones

1. El algoritmo genetico ofrece soluciones factibles en muy pocos segundos, incluso cuando Pyomo tarda varios minutos para alcanzar la optimalidad.

2. La implementacion de pyomo se realizo con *Gurobi* que es mucho mas rapido que otros solvers como *GLPK* o *Higgs*. Incluso sabiendo esto, el tiempo en el que se ejecuta la implementacion del algoritmo genetico es infimo comparado al tiempo que toman los solvers.

3. El modelo exacto sigue siendo la única fuente de optimalidad, pero su tiempo de cómputo se vuelve prohibitivo más allá de 15 clientes.

4. El GA ofrece soluciones con GAP $< 10\,\%$ hasta 19 clientes y aceptable $< 17\,\%$ a 24 clientes en un tiempo $>300\times$ menor.