

Algoritmos Genéticos

El problema de la mochila

Grupo 17

Gomez, Lucas

Volcovinsky, Bruno

Sartorio, Alan



Problema a resolver

Problema a resolver

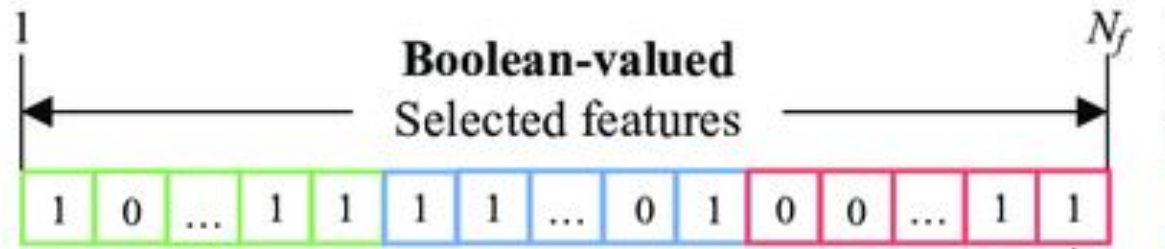
Se posee un conjunto de N elementos, cada uno con un beneficio y un peso asociado.

Se debe decidir que elementos poner en la mochila para maximizar el beneficio sin sobrepasar el peso máximo.



Modelo

Modelado de los cromosomas



Cromosoma válido

$$x_i = \begin{cases} 1 & \text{si se selecciona el elemento } i \\ 0 & \text{sino} \end{cases}$$

w_i = Peso del elemento i

$$\sum_{i=1}^n w_j * x_j \leq W$$





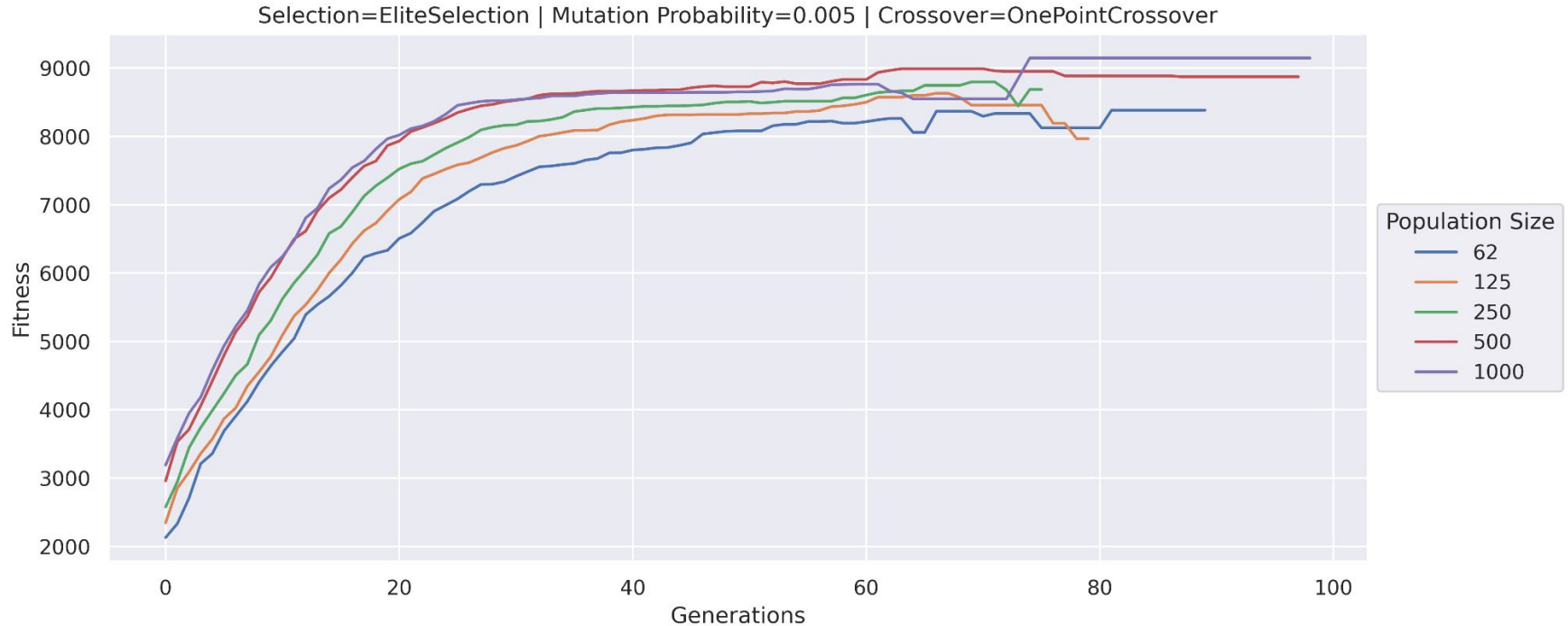
Función de fitness

b_i = Beneficio del elemento i

$$f(x) = \begin{cases} \sum_{i=1}^n b_i * x_i & \text{si el cromosoma es valido} \\ 0 & \text{sino} \end{cases}$$

Población Inicial

Variación de población inicial

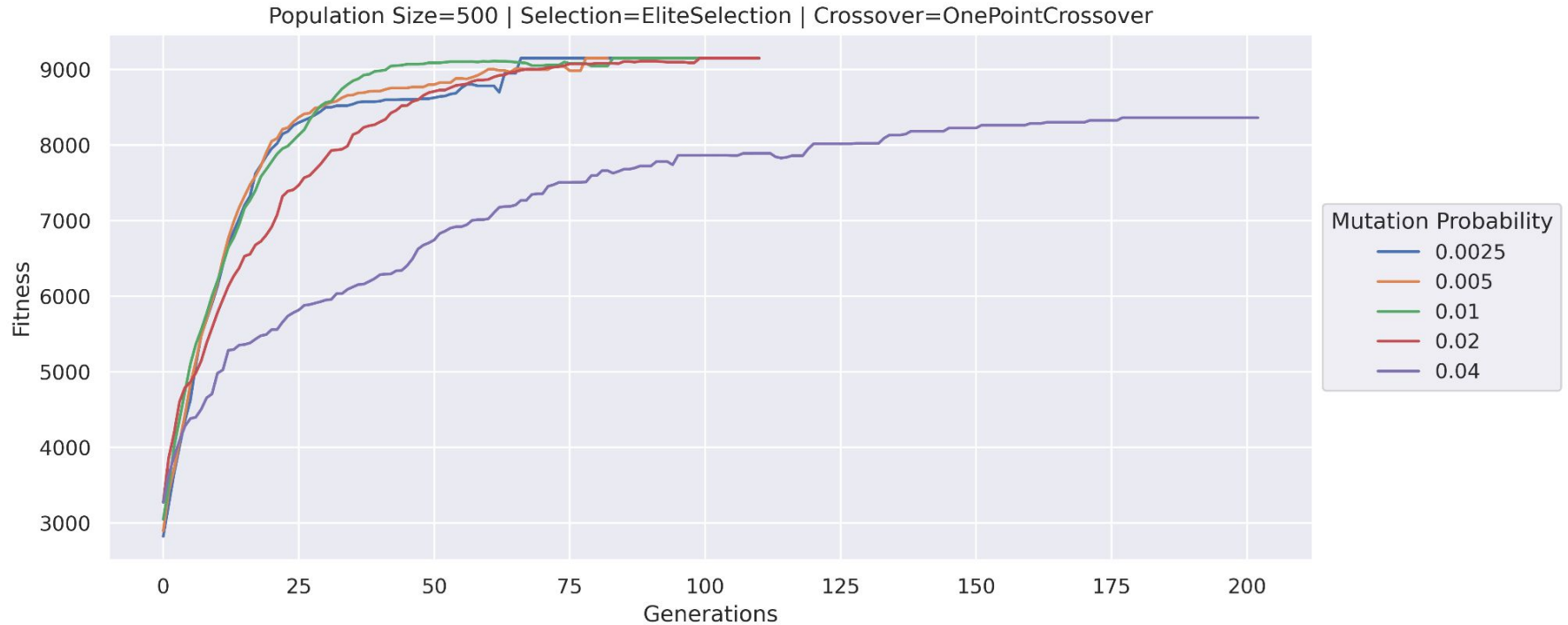




Mutación



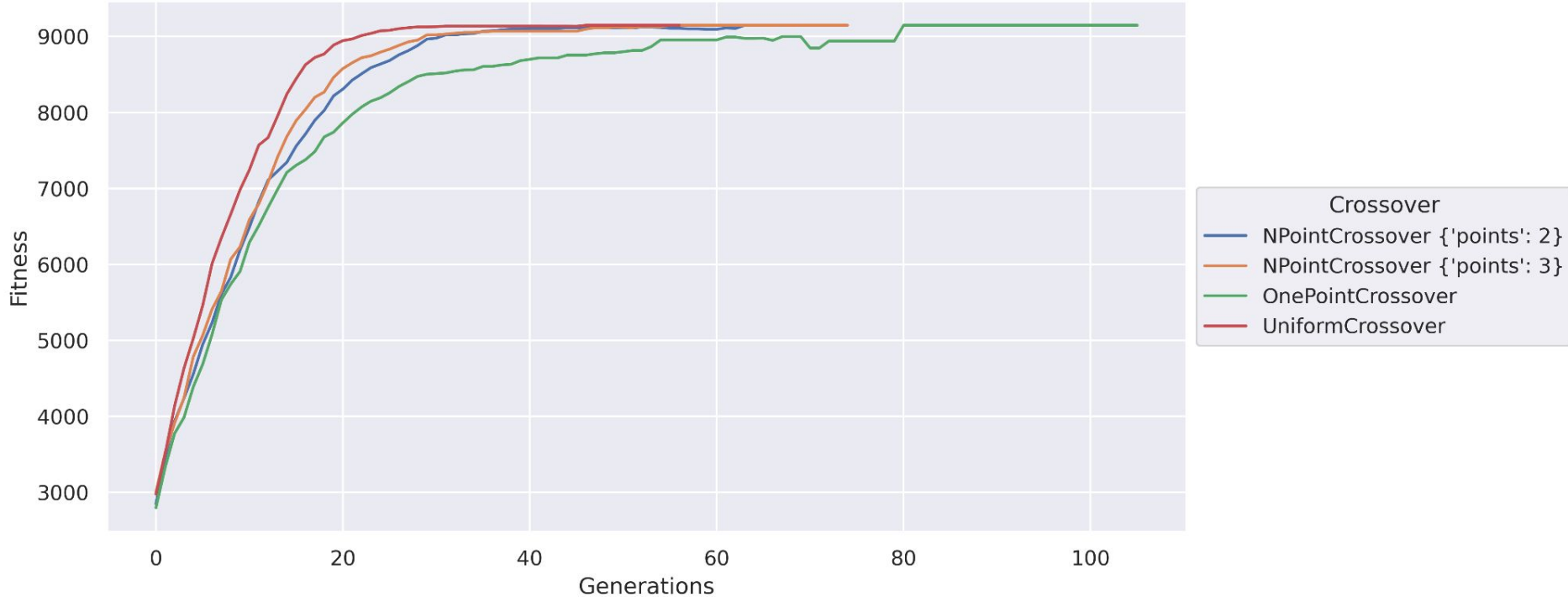
Mutación



Métodos de Cruza

Métodos de cruza

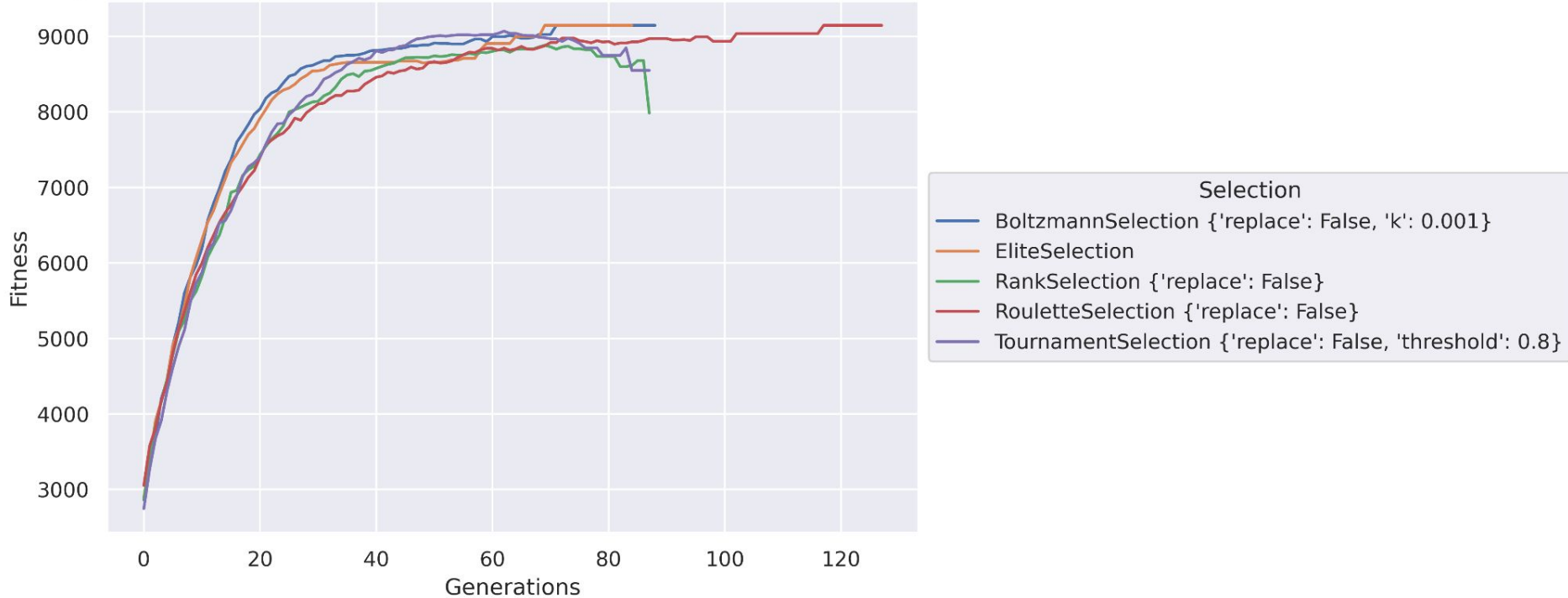
Population Size=500 | Selection=EliteSelection | Mutation Probability=0.005



Métodos de selección

Metodos de seleccion

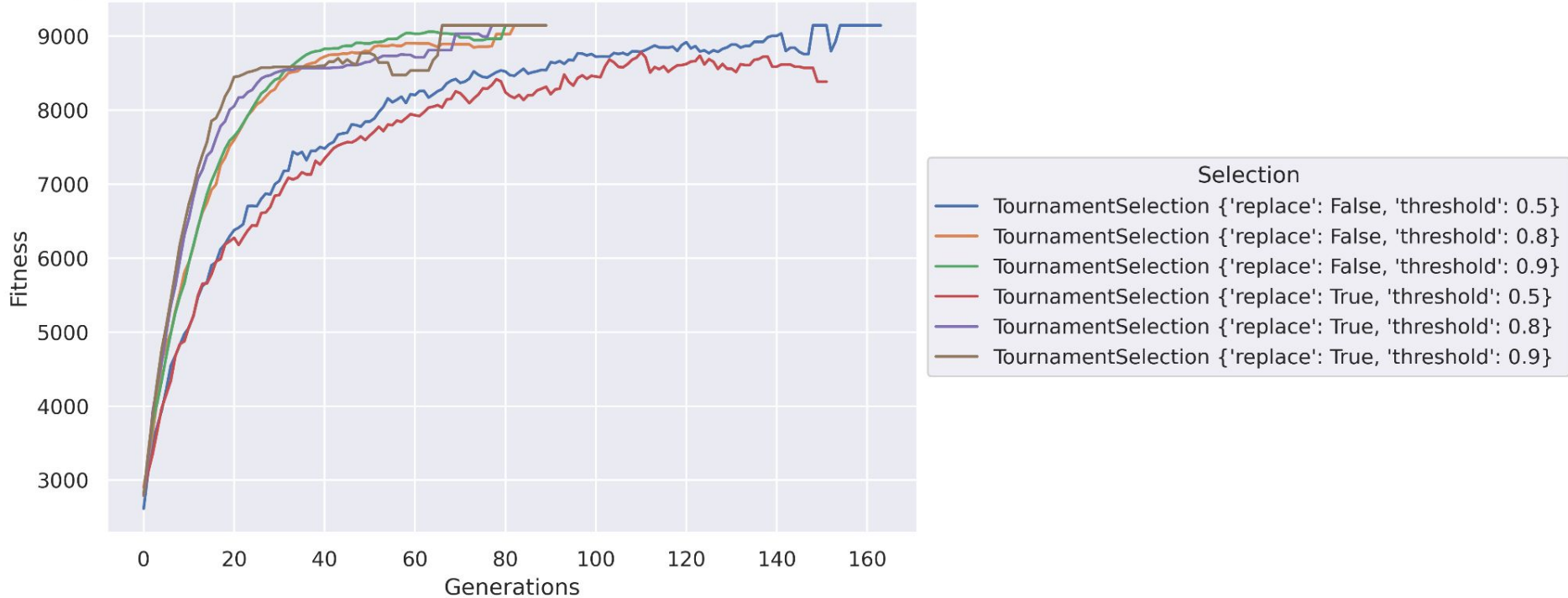
Population Size=500 | Mutation Probability=0.005 | Crossover=OnePointCrossover




Selección competitiva

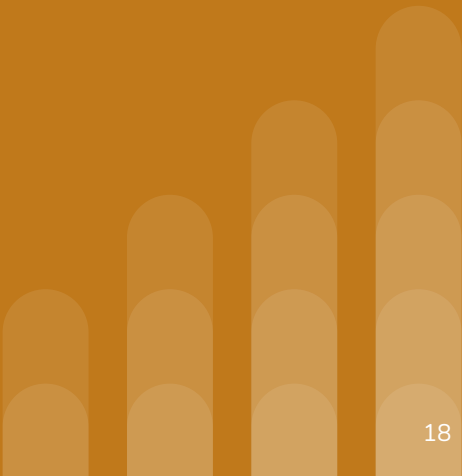
Selección competitiva

Population Size=500 | Mutation Probability=0.005 | Crossover=OnePointCrossover

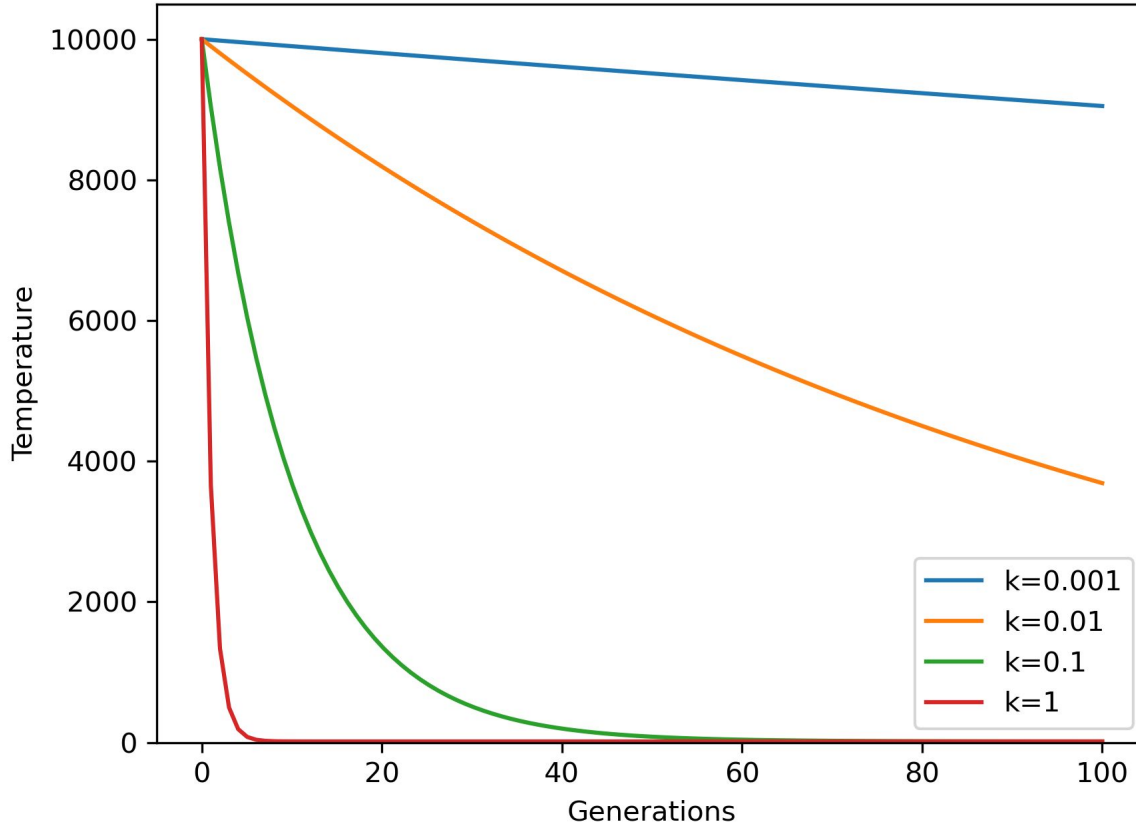




Variación de temperatura



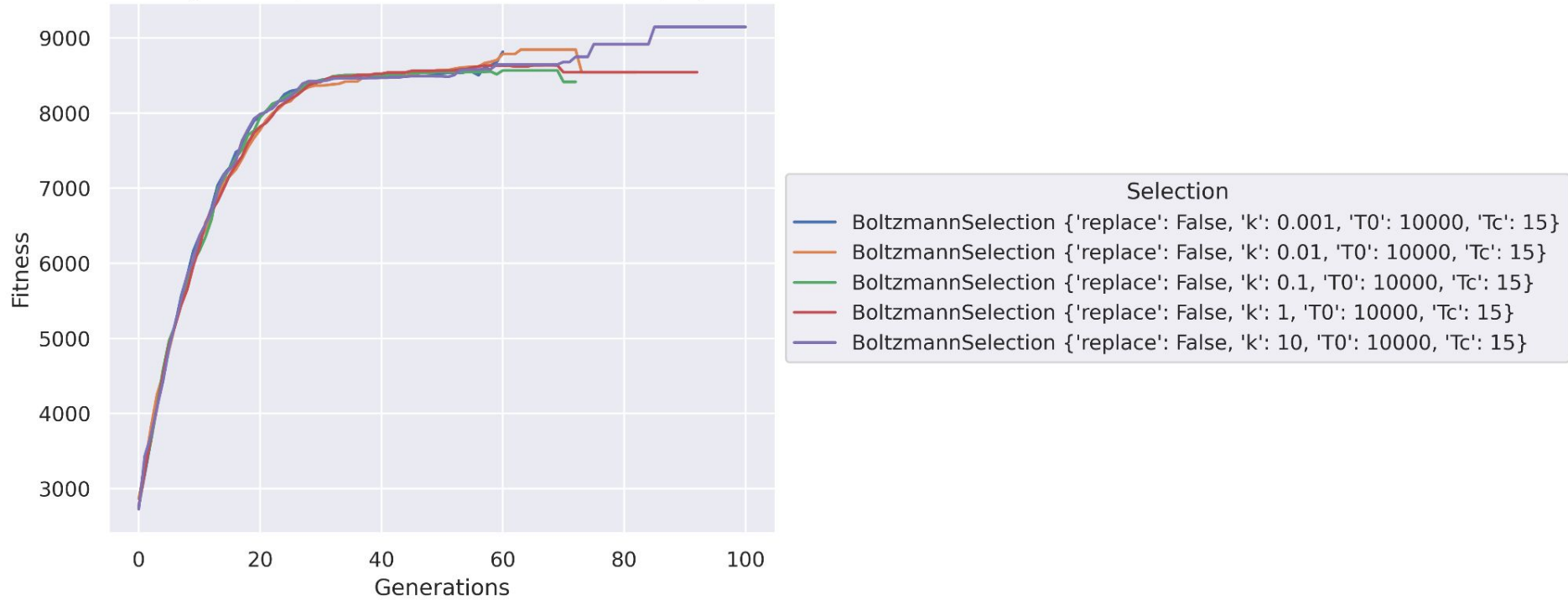
Variación de temperatura



$$T(t) = T_c + (T_0 - T_c) e^{-kt}$$

Variación de temperatura

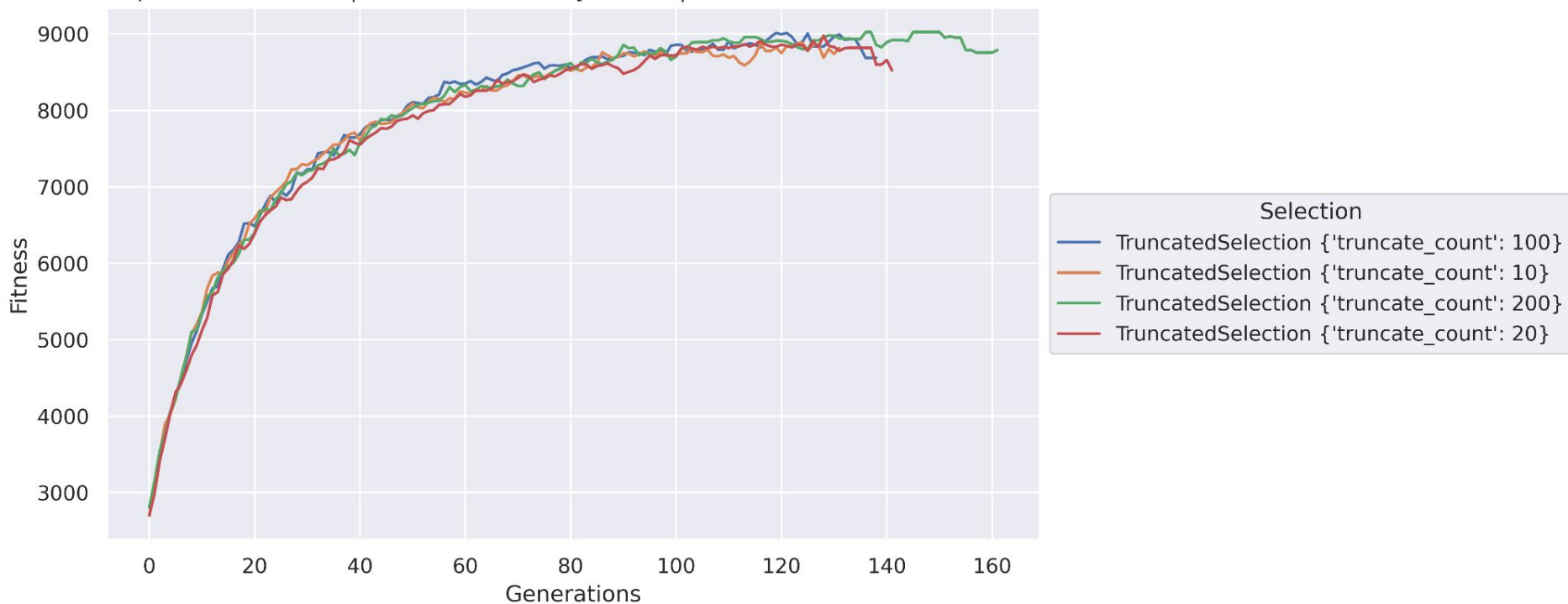
Mutation Probability=0.005 | Crossover=OnePointCrossover | Population Size=500



Método truncado

Método truncado

Population Size=500 | Mutation Probability=0.005 | Crossover=OnePointCrossover



Solución al problema



Solución al problema

- **Fitness: 9147**
- **Población inicial: 500**
- **Crossover: UniformCrossover**
- **Mutación: 0.005**
- **Método de selección: EliteSelection**

Conclusiones



Conclusiones

- El método de cruza simple requiere de más generaciones para encontrar el mejor fitness
- El método de cruza uniforme es el que más rápido converge
- El método de rankSelection fue el que peor fitness obtuvo mientras que el de RouletteSelection fue el que más generaciones necesito para hallar un buen fitness
- A mayor tamaño de población inicial se obtendrá un mejor fitness en menos generaciones.
- En TournamentSelection, para valores de threshold cercanos a 0.5 la evolución es más lenta.
- Para valores de k muy altos se llegara a la temperatura crítica en menos generaciones



¡Gracias!