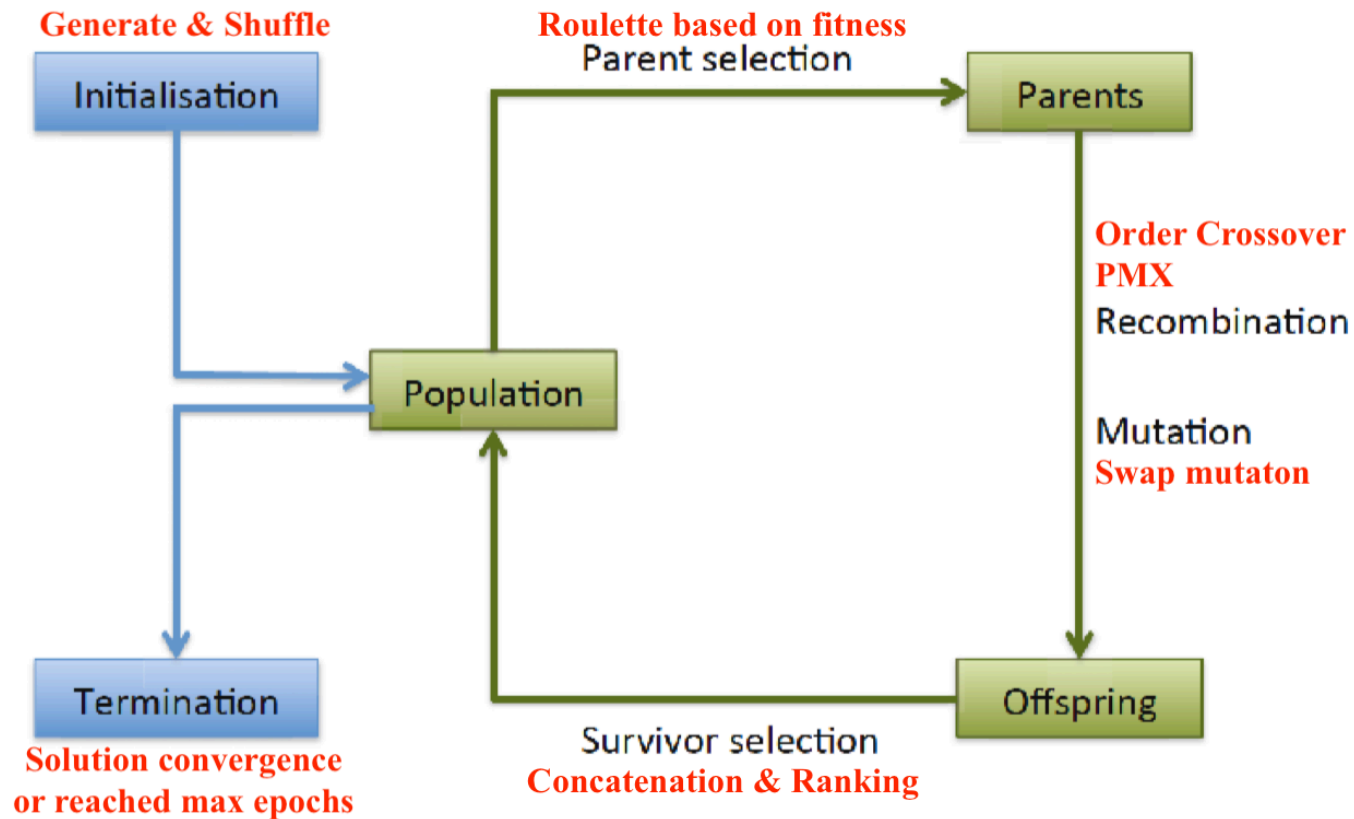


# Program Description

## 1. Program pipeline



## 2. Implementation method

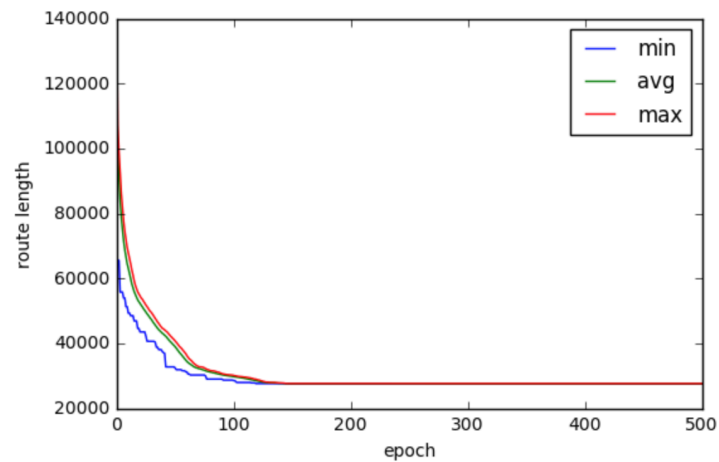
Stage	Method	Implementation Method
<b>Initialization</b>	Generate & Shuffle	1) Create a 100-elements [1, 2, 3, ..., 98, 99, 100] list. 2) Shuffle the list pop_size times to create n random instances as the initial population. 3) Each instance represents a possible solution (city visit order).
<b>Parent selection</b>	Roulette based on probability of fitness	1) Compute the travel length for each solution in the population. 2) Since longer travel length means less fit individual, we define fitness as <b><math>\exp(\text{avg\_length}/l * 5)</math></b> where avg_length is the average travel length of the current population. 3) The longer travel length an individual has, the smaller its fitness value is. 4) Compute the selection probability by <b><math>\text{prob\_indiv} = \text{fit\_indiv}/\text{sum}(\text{fit})</math></b> . 5) The shorter travel length is, the higher probability the individual will be chosen. 6) Roulette choose pop_size individuals as the mating pool.
<b>Crossover</b>	Order Crossover	For each pair of individuals in the mating pool, randomly generate a number between 0 and 1, if the number is less or equal than crossover rate, do order crossover. If not, just copy the parents.
	Partially mapped crossover (PMX)**	For each pair of individuals in the mating pool, randomly generate a number between 0 and 1, if the number is less or equal than crossover rate, do PMX. If not, just copy the parents.  <b>**note:</b> both crossover methods were implemented (see ga.py source code), however order crossover method is used in the final program since it has better performance in practice.
<b>Mutation</b>	Swap mutation	1) Randomly select two points and swap their values. 3) The probability for each point to be chosen is muta_rate.
<b>Survivor selection</b>	Concatenation & Ranking	Concatenate the population of current generation and the generated offsprings, do ranking based on their fitness and choose the top pop_size individuals as the population of next generation.
<b>Termination</b>	Solution convergence or reached max epochs	Terminate the iteration if the fitness of all individuals converges to one same value or number of iteration reaches the max set value.
<b>Set of parameters</b>	f_name	name of the city coordinate file
	pop_size	population size

Stage	Method	Implementation Method
Set of parameters	muta_rate	mutation rate
	cross_rate	crossover rate
	gen	max evolution epochs allowed for the program to run
Runtime for 29 cities	CPU times: user 1min 2s	
Runtime for 734 cities	CPU times: user 30min 46s	

### 3. Results

#### 1) Result for A2\_TSP\_WesternSahara\_29.txt

500th Gen:  
min length: 27601.1737745  
avg length: 27601.1737745  
max length: 27601.1737745



## 2) Result for A2\_TSP\_Uruguay\_734.txt

500th Gen:  
min length: 760905.485158  
avg length: 765390.631584  
max length: 766417.759468

