Genetic Algorithms

Dr. Mahmoud Nabil Mahmoud
mnmahmoud@ncat.edu

North Carolina A & T State University

September 6, 2021

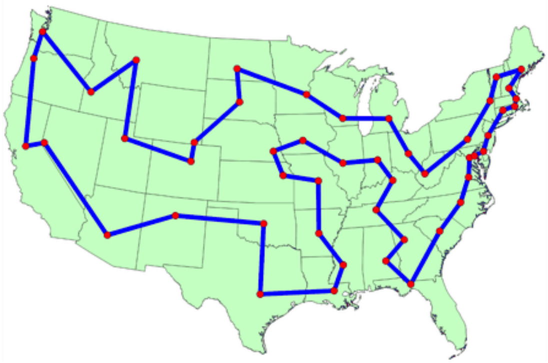
Components of GA

- Parameter set: How we encode each individual in the search space? What is the search space?
- Population: How to initialize the population?
- Fitness function: How to evaluate each individual?
- Operators: Selection, Crossover, Mutation.

Outline

Traveling Sales Man Problem

Given a list of cities and the coordinates of each city, what is the shortest possible route that visits each city exactly once and returns to the origin city?



How many possible solutions to travel through 48 states?

TSP Search space

TSP	Depth of Search Space
4	24
10	3628800
20	2.4×10^{18}
50	3.04×10^{64}
100	9.3×10^{157}
200	$> 1 \times 10^{500}$

Parameter Set

- Each individual can encode the cities traveled
- Length of each chromosome equal to the number of cities
- Integer representation chromosome.

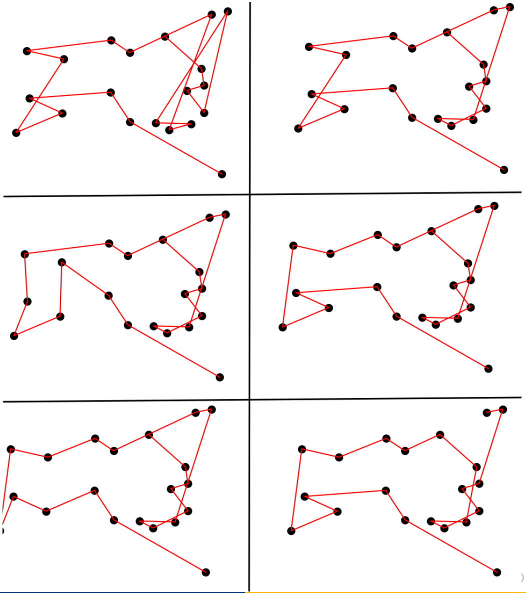
Ex.

- For seven cities an individual 3 5 1 7 8 2 6 4

Population example

Population Example 20 cities:

03 15 01 07 . . . 08 12 06 04
08 11 03 04 . . . 05 17 02 06
04 06 20 07 . . . 08 05 01 03
01 15 16 03 . . . 02 17 08 04
02 04 07 01 . . . 08 13 06 05



Fitness Function

We can set our fitness function as the inverse of the number of dead queens.

$$fitness(x) = \frac{1}{1+\text{total\_distance(path)}}$$

Another way for this problem is to scale the fitness exponentially if all the distances are very close such as

$$fitness(x) = e^{\frac{\text{constant}}{1+\text{total\_distance(path)}}}$$

Cross Over

1. Generate a binary template at random that has the same length a the parents.

Cross Over

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".

Cross Over

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Fill the remaining elements that complete the list of the cities from Parent 2.

Cross Over

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Fill the remaining elements that complete the list of the cities from Parent 2.
4. You can do the same with Child 2.

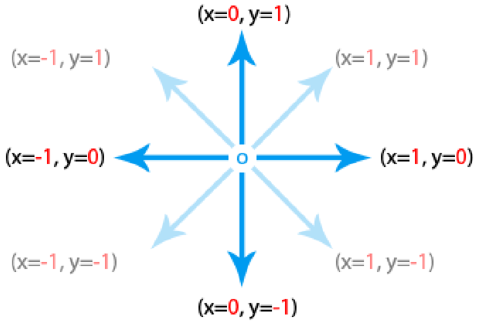
Mutation

- Swap-based: Two cities are selected at random, and their positions are interchanged.
- Scramble: Two positions within a string are selected at random, all of the cities between the two positions are randomly ordered.

Outline

Path Planning

- Find a path from a source to a destination point.
- Does not have to be the shortest path
- May be there are some obstacles
- Assume 2d movement:



Parameter Set

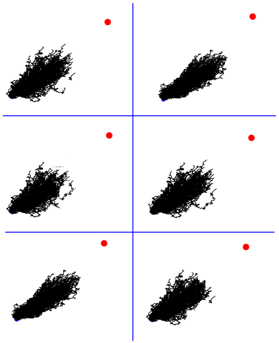
- Assume Maximum Number of moves to reach the goal is N

Ex.
N = 8
-1 0 0 1 0 0 1 1 | 0 1 1 0 1 1 0 1

1. (-1,0)
2. (0,1)
3. (0,1)
4. (1,0)
5. (0,1)
6. (0,1)
7. (1,0)
8. (1,1)

Population example

```
-1 0 0 1 0 0 1 1 | 0 1 1 0 1 1 0 1
1 -1 0 1 1 1 0 1 | 1 -1 1 0 1 0 0 1
1 -1 1 0 1 0 0 1 | 0 0 1 0 1 -1 0 1
-1 0 0 1 0 0 1 1 | 0 -1 1 0 1 1 0 0
-1 1 1 1 0 0 1 1 | 0 1 1 -1 1 1 0 0
```



Fitness Function

We can set our fitness function as the inverse of the number of dead queens.

$$fitness(x) = \frac{1}{1+\text{distance(goal,pathendpoint)}}$$

- Small distance means more fit.
- Can be updated to include obstacles effect.

Cross Over

1. Generate a binary template at random that has the same length a the parents.

Cross Over

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".

Cross Over

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Fill the remaining elements that complete the list of the cities from Parent 2.

Cross Over

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Fill the remaining elements that complete the list of the cities from Parent 2.
4. You can do the same with Child 2.

Mutation

- Any movement can be flipped in the x and/or y positions to $\{0,1,2\}$
- Also know as random resetting.

Outline

The N queen Problem

This is the problem of placing N queens on a regular N $\times$ N chessboard so that no two of them can check each other.

Ex. Let N = 4

	1	**2**	**3**	**4**
1				
2				
3				
4				

The N queen Problem

This is the problem of placing N queens on a regular N $\times$ N chessboard so that no two of them can check each other.

Ex. Let N $=$ 4

	1	**2**	**3**	**4**
1				
2				
3				
4				

Search space is ${}^{N^2}C_N$

N-Queen Search space

N-Queens	**Depth of Search Space**
4	1.8×10^3
10	1.7×10^{13}
20	2.8×10^{33}
50	1.6×10^{33}
100	6.5×10^{241}
200	$> 1 \times 10^{500}$

Parameter set

	1	2	3	4	5	6	7	8
1			x					
2					x			
3	x							
4							x	
5								x
6		x						
7						x		
8				x				

Parameter set

	1	2	3	4	5	6	7	8
1			x					
2					x			
3	x							
4							x	
5								x
6		x						
7						x		
8				x				

String individual: 3 5 1 7 8 2 6 4
Sequence of 8 unique numbers from 1 to 8.
Search space reduced to N!

Population Example

Population Example:

3 5 1 7 8 2 6 4

8 1 3 4 5 7 2 6

4 6 2 7 8 5 1 3

1 5 6 3 2 7 8 4

2 4 7 1 8 3 6 5

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1			x					
2					x			
3	x							
4							x	
5								x
6		x						
7						x		
8				x				

Fitness Function

We can set our fitness function as the inverse of the number of dead queens.

$$fitness(x) = \frac{1}{1+\text{dead-queens}(x)}$$

This way, our fitness function reaches a maximum of 1 whenever the number of dead queens is 0. **Ex.**

What is the fitness of 3 5 1 7 8 2 6 4?

Crossover operator

1. Generate a binary template at random that has the same length a the parents.

Crossover operator

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".

Crossover operator

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Create a list of the elements that correspond to a "0" in the binary: template from Parent 1.

Crossover operator

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Create a list of the elements that correspond to a "0" in the binary: template from Parent 1.
4. Permute these elements so that they are in the same order as the: are in Parent 2.

Crossover operator

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Create a list of the elements that correspond to a "0" in the binary: template from Parent 1.
4. Permute these elements so that they are in the same order as the: are in Parent 2.
5. Supply the gaps in Child 1 with the permuted elements in the orde generated in step 4.

Crossover operator

1. Generate a binary template at random that has the same length a the parents.
2. Copy the positions of Parent 1 to Child 1 wherever the binary, template contains a "1".
3. Create a list of the elements that correspond to a "0" in the binary: template from Parent 1.
4. Permute these elements so that they are in the same order as the: are in Parent 2.
5. Supply the gaps in Child 1 with the permuted elements in the orde generated in step 4.
6. Create Child 2 using a similar process.

Crossover operator

Select two parents

Parent1: 1 2 3 4 5 6 7 8
Parent2: 8 6 4 2 7 5 3 1

Form a template

Template: 0 1 1 0 1 1 0 0

Apply the template to the parents

Child1: - 2 3 - 5 6 - -
Child2: 8 - - 2 - - 3 1

Fill in the gaps to create final children

Child1: 8 2 3 4 5 6 7 1
Child2: 8 4 5 2 6 7 3 1

Mutation operator

Homaifar et al [92].

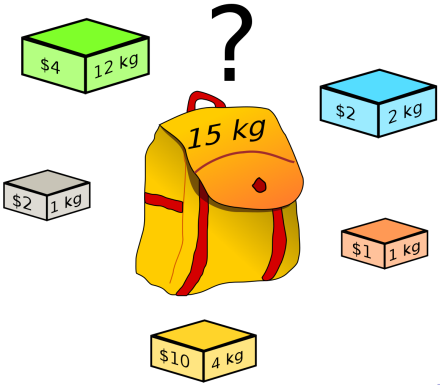
- Order-based: Two queens are selected at random, and their positions are interchanged.
- Scramble: Two positions within a string are selected at random, all of the queens between the two positions are randomly ordered.

Outline

The Knapsack Problem

We are given a set of n items, each of which has attached to it some value v_i, and some cost c_i

The task is to select a subset of those items that maximizes the sum of the values, while keeping the summed cost within some capacity C_{max}



Components of GA

- Parameter set:

Components of GA

- Parameter set:
    - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space
- Population

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space
- Population
  - Set of binary string each of length n.

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space
- Population
  - Set of binary string each of length n.
- Objective function:

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space
- Population
  - Set of binary string each of length n.
- Objective function:
  - Given allele g_i

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space
- Population
  - Set of binary string each of length n.
- Objective function:
  - Given allele g_i
  - $$f(x) = \begin{cases} -1 & \text{if} \quad \sum_{i=1}^{n} c_i \times g_i \leq C_{max} \\ \sum_{i=1}^{n} v_i \times g_i & \text{otherwise} \end{cases}$$

Components of GA

- Parameter set:
  - A Binary string of length n, where a 1 in a given position indicates that an item is included and a 0 that it is omitted.
  - 2^n search space
- Population
  - Set of binary string each of length n.
- Objective function:
  - Given allele g_i
  - $$f(x) = \begin{cases} -1 & \text{if} \quad \sum_{i=1}^{n} c_i \times g_i \leq C_{max} \\ \sum_{i=1}^{n} v_i \times g_i & \text{otherwise} \end{cases}$$
- Operators: Same as before

References

- Goldenberg, D.E., 1989. Genetic algorithms in search, optimization and machine learning.
- Michalewicz, Z., 2013. Genetic algorithms + data structures= evolution programs. Springer Science & Business Media

Questions