CS4811 Artificial Intelligence

Genetic Algorithms & Differential Evolution

Nyew Hui Meen

February 10, 2014

Joint Work with
Dr. Onder Nilufer *(CS Department)* and
Dr. Abdelkhalik Ossama *(MEEM Department)*

What is a Genetic Algorithm?

A genetic algorithm (GA) is an adaptive heuristic search algorithm based on evolutionary ideas of natural selection.

When to use Genetic Algorithms?

- Search space is large, complex and difficult to understand.
- Optimization problems where the solution need not be *globally optimal*.

Elements of Genetic Algorithms

- Given a function $f(x) | \, 0 \leq x \leq 10$, the goal is to find an x value that maximizes $f(x)$.
- In GA's terms, f is called the *fitness function*, a candidate solution x is called a *chromosome* and the $f(x)$ value is called the *chromosome fitness*.
- A collection of chromosomes is called a *population*.

Genetic Algorithm Process

1. Create a population of *n* chromosomes.
2. Repeat the following steps until *n* offspring have been created
    1. **Select** two parent chromosomes.
    2. Produce two offspring from the parent chromosome by **crossover**.
    3. **Mutate** the offspring.
    4. Place the offspring in the new population.
3. Repeat step 2 until termination conditions are met.

Create Population

Example population of size 3

Chromosome label	Chromosome string	Fitness
A	110	1
B	001	2
C	101	3

The Selection Operation

- Random selection
- Fitness-proportionate selection

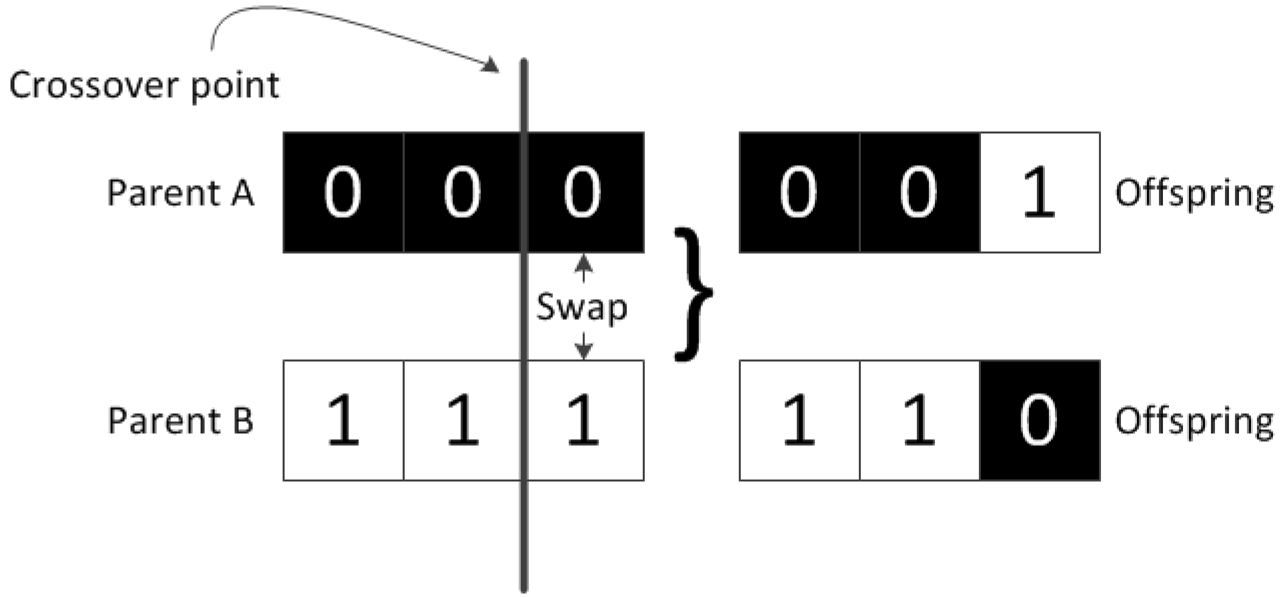
    Probability a chromosome selected is equal to its fitness divided by the total fitness in the population.

Comparison of Selection Method

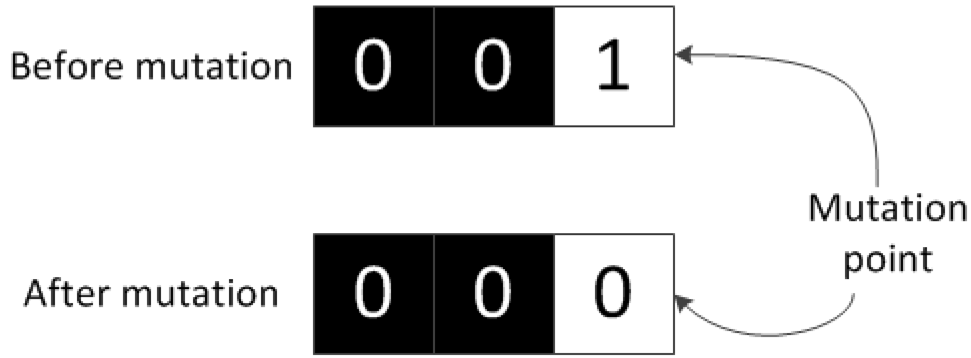
Chromosome label	Chromosome string	Fitness	Random	Fitness-proportionate
A	110	1	**33%**	**17%**
B	001	2	**33%**	**33%**
C	101	3	**33%**	**50%**

The Crossover Operation

Single-point Crossover



The Mutation Operation



Differential Evolution Process

1. Create a population of *n* chromosomes.
2. Repeat the following steps ~~until *n* offspring have been created~~ **for every chromosome *x* in the population**.
    1. Select ~~two~~ three parent chromosomes**, a, b and c**.
    2. Produce ~~two~~ **one** offspring from parent chromosomes by ~~crossover~~ **transformation operation**.
    3. ~~Mutate the offspring.~~
    4. Place the offspring in the ~~new~~ population**, if the offspring's fitness is better than *x's* fitness**.
3. Repeat step 2 until termination conditions are met.

Transformation

1. For every parameter x_i in chromosome x:
   1. Pick a random number r.
   2. If $r <$ transformation probability p:

      Compute new offspring y parameter y_i as follows:
      $y_i = a_i + F(b_i - c_i)$
   3. If $y \geq$ transformation probability p:

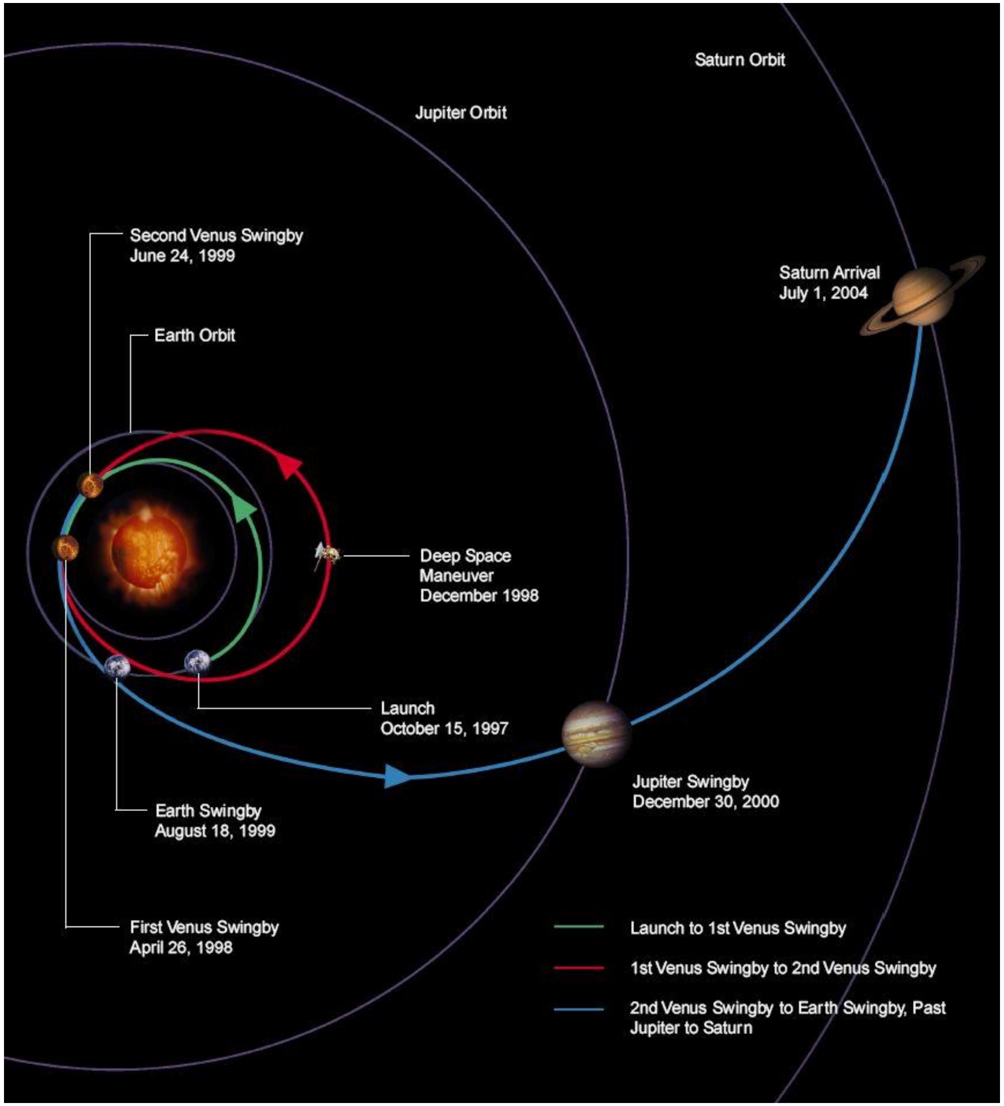
      Compute new offspring y parameter y_i as follows:
      $y_i = x_i$

❖ F is differential weight

Transformation Example

- Chromosomes
  - $x = [1,4]$
  - $a = [2,3]$
  - $b = [4,6]$
  - $c = [9,4]$
- Transformation variables:
  - $F = 1$
  - $p = 0.5$
- Let first $r = 0.4$. $y_1 = a_1 + F(b_1 - c_1) = 2 + (4 - 9) = -3$
- Let second $r = 0.7$. $y_2 = x_2 = 4$
- Thus offspring $y = [-3,4]$

Cassini Mission

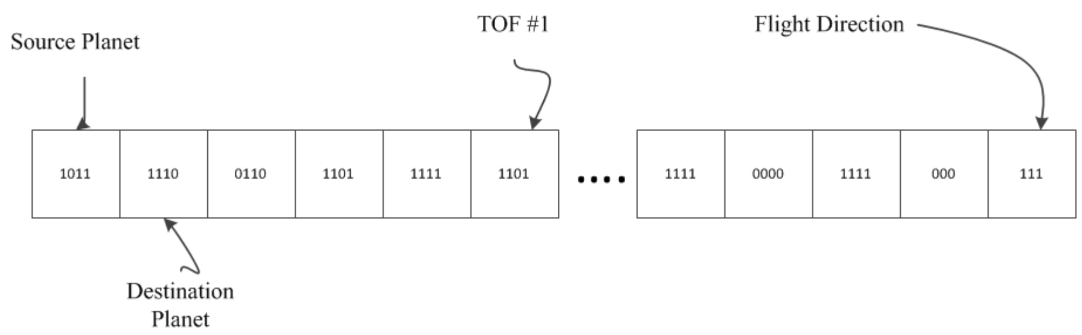


Variable-length Chromosome

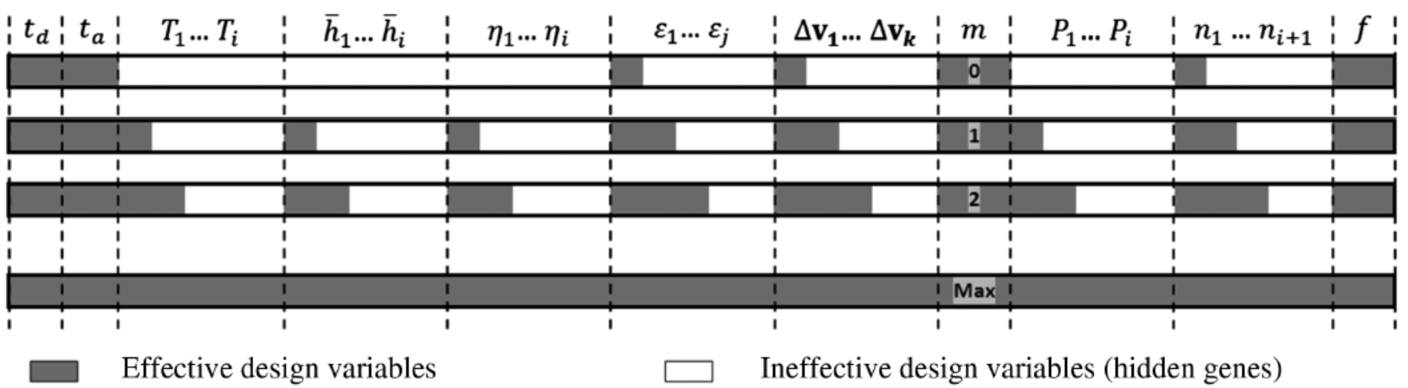
- Cassini Mission’s fitness function is complex and its chromosomes have variable length.
- This creates difficulty for single-point crossover operation.

Cassini Mission's Chromosome

Fixed length Chromosome



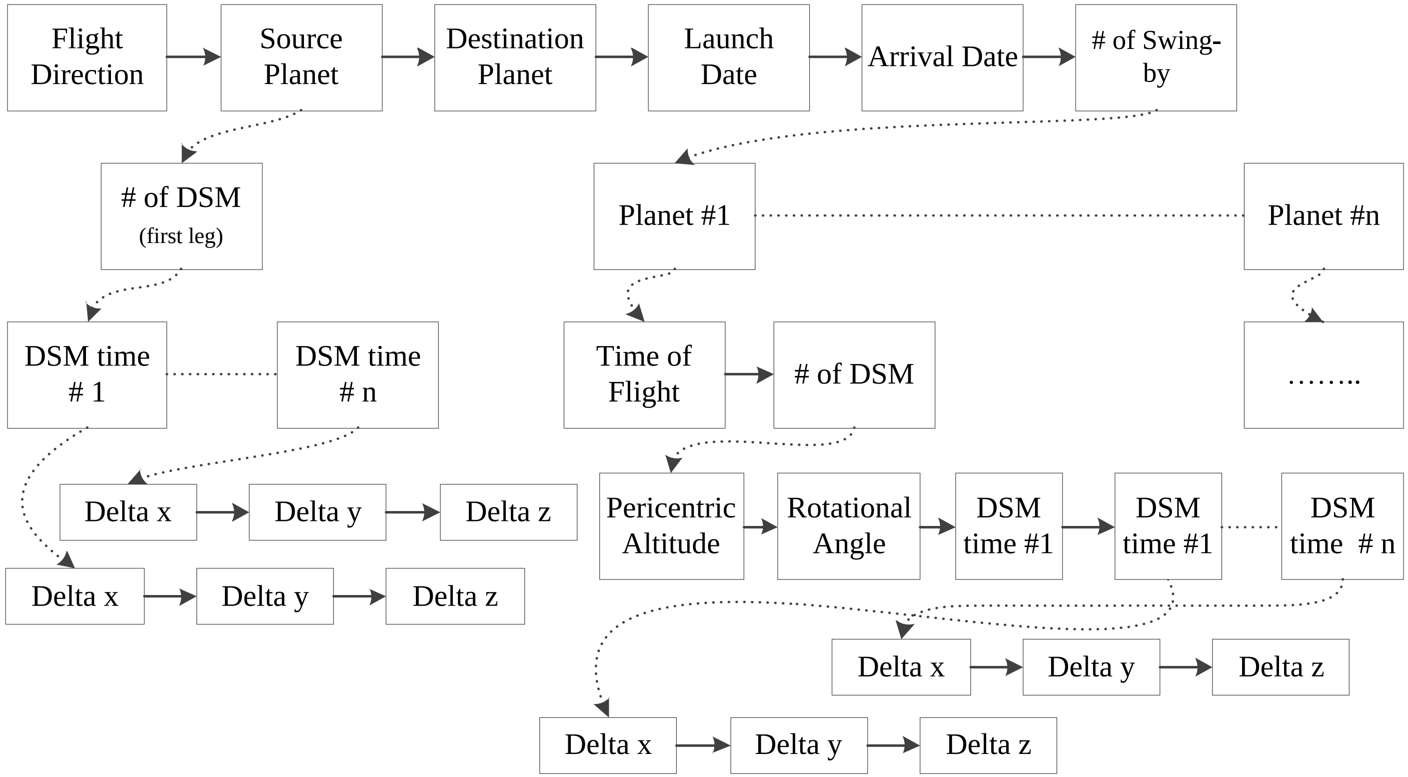
Fixed length Chromosome with Hidden Genes



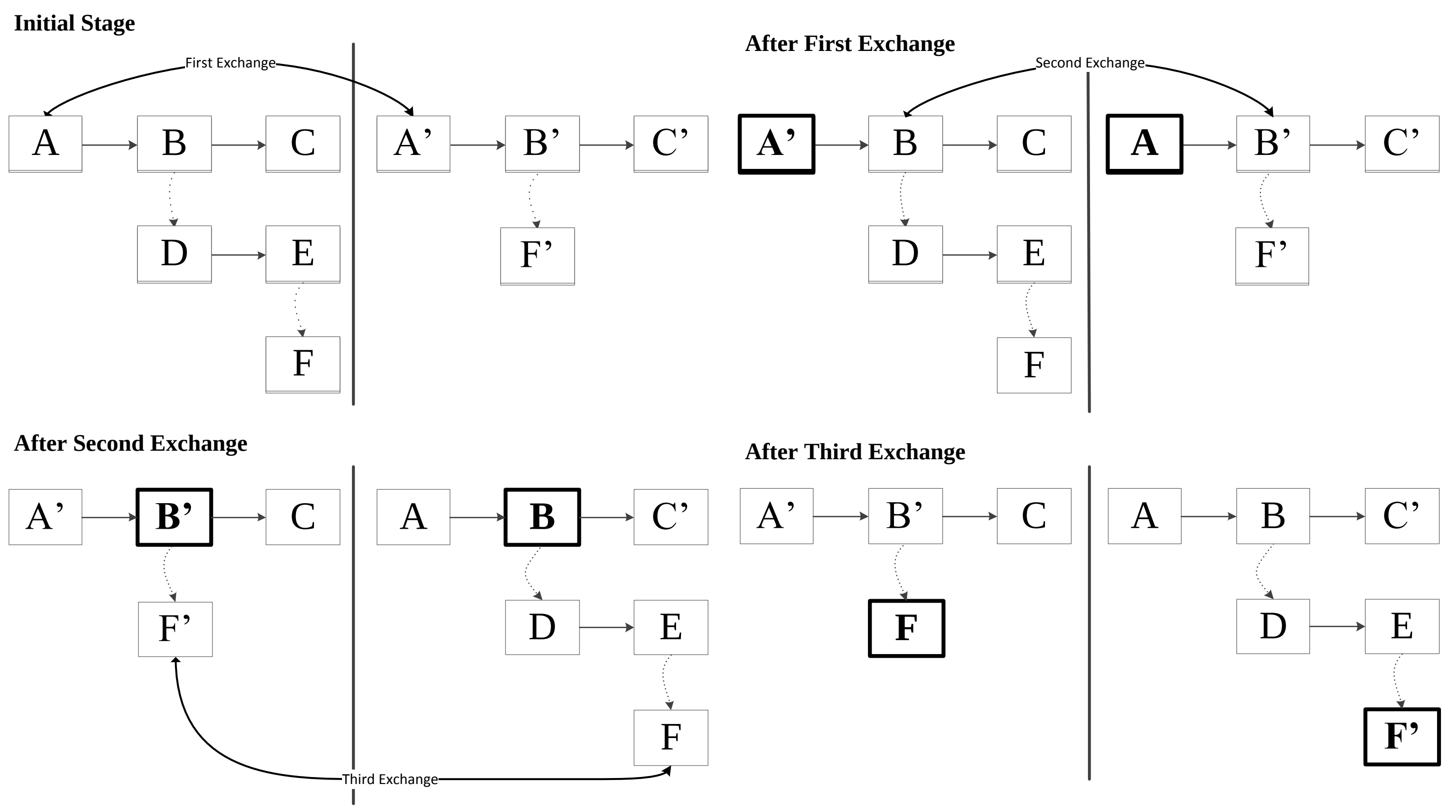
Structured Chromosome

- Describe parameters with *parameter existential dependency relationship*.
- Chromosomes grow and shrink during crossover and mutation operations.

Cassini Mission Chromosome



Structured Chromosome Crossover Operation



Experiment Setup for Cassini Mission

- Crossover probability = 0.1
- Mutation probability = 0.1

Parameter	Upper bound	Lower bound
Source Planet	3 (Earth)	3 (Earth)
Destination Planet	6 (Saturn)	6(Saturn)
Number of Swing-by	4	3
Planet	2	5
Launch year	1997	1997
Launch month	11	11
Launch day	31	1
Arrival year	2007	2007
Arrival month	6	1
Arrival day	30	1
Time of flight	1000	40
Number of DSM	0	0
Flight Direction	1	0

Experiment Process

- We run SCGA and SCDE 200 times and store each run’s best solution.
- Then the solutions are optimized by running it through the MATLAB local optimization toolbox.
- For each set of solutions, we compute their success rate of producing the target cost and the success rate of producing the target planet sequence.

Success rate calculations

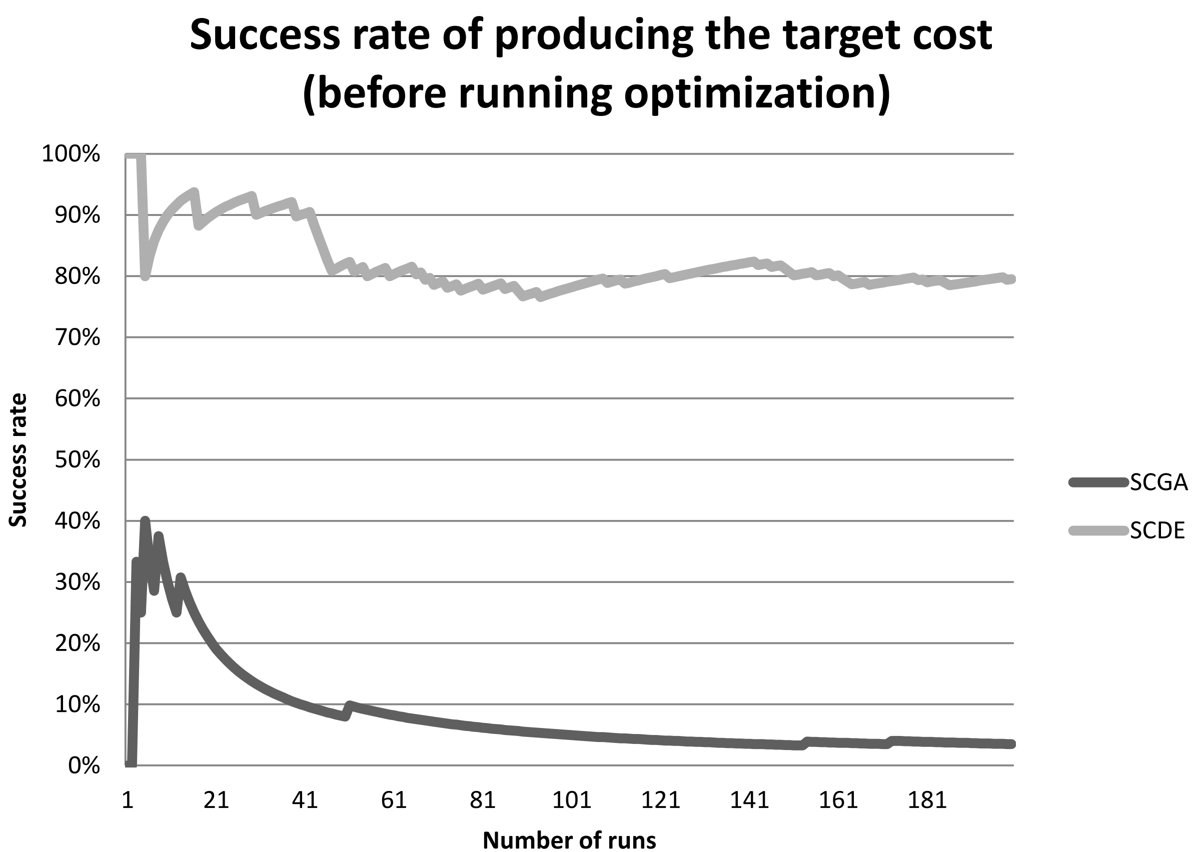
```
tolerance ← 0.1
best_cost ← 12
success_count ← 0
i ← 1
while i ≤ 200 do
 if cost(i) ≤ best_cost then do
     success_count ← success_count + 1
     best_cost ← cost(i) + tolerance

 success_rate(i) ← success_count/i
 i ← i + 1
```

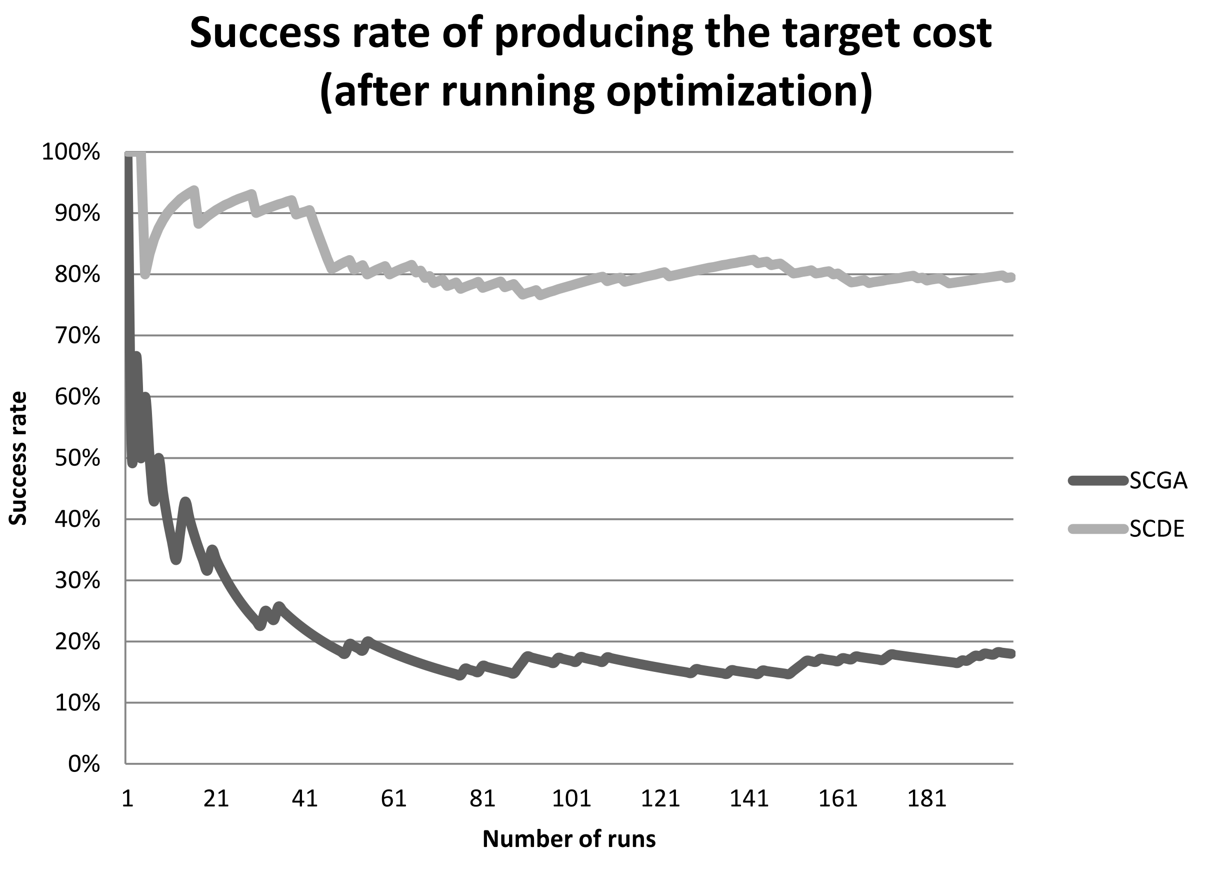
```
best_sequence ← 235
success_count ← 0
i ← 1
while i ≤ 200 do
 if cost(i) ≤ best_sequence then do
     success_count ← success_count +
1

 success_rate(i) ← success_count/i
 i ← i + 1
```

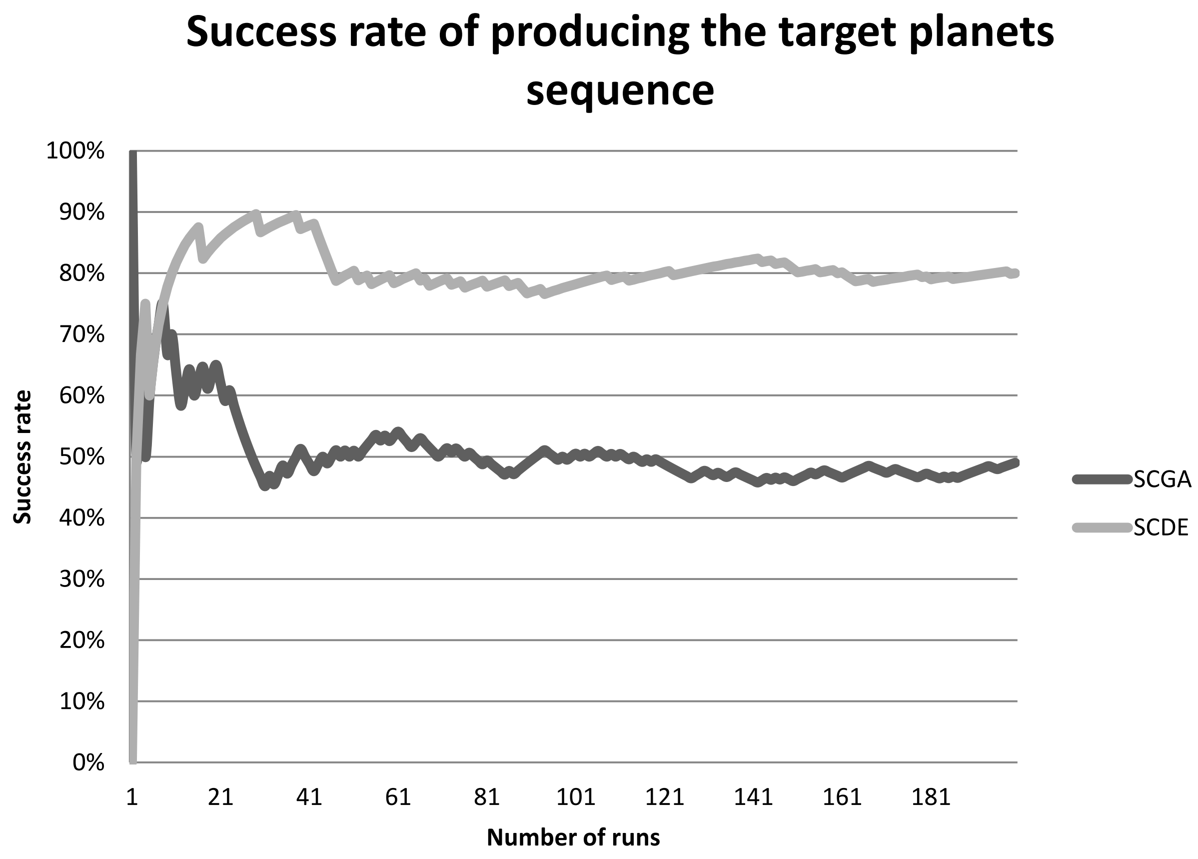
Experimental Results (1)



Experimental Results (2)



Experimental Results (3)



References

- *A. E. Eiben & J. E. Smith*, Introduction to Evolutionary Computing, *Springer, 2008*
- *O. Abdelkhalik and A. Gad*, Dynamic-Size Multi-Population Genetic Optimization for Multi-Gravity-Assist Trajectories, *AIAA Journal of guidance, control, and dynamics. Accepted, July 2011, doi: 10.2514/1.54330*
- *A. Gad, O. Abdelkhalik*, Hidden Genes Genetic Algorithm for Multi-Gravity-Assist Trajectories Optimization, *AIAA Journal of Spacecraft and Rockets, AIAA, Vol. 48, No 4, pp 629-641, July-August 2011. doi: 10.2514/1.52642*