Advanced Optimization Methods -
Single-state methods

- **Objective function:**

$$min.\ f(x1,\ x2)\ =\ (3x_1+14x_2-17)^2+(6x_1+7x_2-19)^2,\ \ -6\le x_1\le 6\ and\ -6\le x_2\le 6.$$

- **Optimization methods: Single-state methods**
  - 1. Exhaustive Enumeration
  - 2. Random methods
    - A. Random jump
    - B. Random walk
  - 3. Gradient-based methods
    - A. Steepest Gradient Descent (SGD)
    - B. Newton's method
    - C. Marquardt's method
  - 4. Stochastic Local Search methods
    - A. Iterated Local Search (ILS) method
    - B. Simulated Annealing (SA) method
    - C. Tabu Search (TS) method

❖ Input (Initial settings) & Output

1. Exhaustive Enumeration

Input:

Grid interval: 0.01

Output:

Optimal point: (2.3269, 0.7156)

Optimum: 0.000853

Number of iterations: 7215

2. Random methods

A. Random jump

Input:

Expected optimal point: (2.33, 0.72) # Exhaustive Enumeration

Tolerate error: 0.05

Random seed: 1

Random interval: 20

Grid size scale: ⅓

Jump times per grid: 20

Output:

Optimal point: (2.3433, 0.7207)

Optimum: 0.025215

Number of iterations: 320

Number of used random seeds: 640

❖ Input (Initial settings) & Output (Cont.)

B. Random walk

B.1 Random walk 1 - decaying lambda

Input:

Initial point: (0, 0)

Number of iterations per random seed: 100

Number of random seeds: 5

Random seed: 0

Random interval: 1

Lambda: 1

Epsilon: 0.05

Output:

Optimal point: (0.3318, 1.4628)

Optimum: 65.849032

Number of iterations: 1000

Number of used random seeds: 1000

B.2 Random walk 2 - optimal lambda

Input:

Initial point: (0, 0)

Number of iterations per random seed: 100

Number of random seeds: 5

Random seed: 0

Random interval: 1

Lambda: 1

Epsilon: 0.05

Number of iterations: 1000

Output:

Optimal point: (0.3246, 1.431)

Optimum: 65.558849

Number of iterations: 1000

Number of used random seeds: 1000

❖ Input (Initial settings) & Output (Cont.)

3. **Gradient-based methods**

   A. **Steepest Gradient Descent (SGD)**

      Input:

         Initial point: (0, 0)

         Epsilon: 0.01

      Output:

         Optimal point: (2.3333, 0.7143)

         Optimum: 0.0

         Number of iterations: 10

   B. **Newton's Method**

      Input:

         Initial point: (0, 0)

         Epsilon: 0.01

      Output:

         Optimal point: (2.3333, 0.7143)

         Optimum: 0.0

         Number of iterations: 2

   C. **Marquardt's Method**

      Input:

         Initial point: (0, 0)

         Epsilon: 0.01

         alpha = 10000

         c1 = 0.25

         c2 = 2

      Output:

         Optimal point: (2.3333, 0.7143)

         Optimum: 0.0

         Number of iterations: 9

❖ Input (Initial settings) & Output (Cont.)

4. **Stochastic Local Search Methods**

   A. **Iterated Local Search (ILS) method**

   Input:

   Initial point: (0, 0)

   Random seed: 1

   Random interval: 20

   Alpha: 1

   Scale: 0.999

   Ratio: 0.8

   Number of iterations: 500

   Number of search: 20

   Output:

   Optimal point: (2.3422, 0.7095)

   Optimum: 0.002006

   Number of iterations: 500

   B. **Simulated Annealing (SA) method**

   Input:

   Initial point: (0, 0)

   Random seed: 1

   Random interval: 20

   Alpha: 1

   Scale: 0.999

   Ratio: 0.8

   Number of iterations: 500

   Number of initial seeds: 5

   Temperature reduction factor: 0.5

   Output:

   Optimal point: (2.3374, 0.7068)

   Optimum: 0.009437

   Number of iterations: 500

- **Input (Initial settings) & Output (Cont.)**

C. Tabu Search (TS) method

Input:

Initial point: (0, 0)

Random seed: 1

Random interval: 20

Alpha: 1

Scale: 0.999

Ratio: 0.8

Number of iterations: 500

Number of initial seeds: 5

Temperature reduction factor: 0.5

Maximal size of tabu list: 7

Neighborhood size: 20

Output:

Optimal point: (2.3335, 0.7141)

Optimum: 0.000002

Number of iterations: 500

❖ Comparison of results I. Effectiveness & Efficiency

Method	Optimal point	Optimum	Number of iterations
1. Basic method			
Exhaustive Enumeration	(2.3269, 0.7156)	0.000853	7215
2. Random methods			
Random Jump	(2.3433, 0.7207)	0.025215	320 (num of random seeds = 640)
Random Walk - decaying lambda	(0.3318, 1.4628)*	65.849032*	1000
Random Walk - optimal lambda	(0.3246, 1.4310)*	65.558849*	1000
3. Gradient-based methods			
Steepest Gradient Descent (SGD)	(2.3333, 0.7143)	0.0	10
Newton's method	(2.3333, 0.7143)	0.0	2
Marquardt's method	(2.3333, 0.7143)	0.0	9
4. Stochastic Local Search methods			
Iterated Local Search (ILS) method	(2.3422, 0.7095)	0.002006	500
Simulated Annealing (SA) method	(2.3374, 0.7068)	0.009437	500
Tabu Search (TS) method	(2.3335, 0.7141)	0.000002	500

*Trapped into the local minimum.

*ILS based on the perturbation scheme can be regarded as the improved version of Random Walk.

❖ Comparison of results II. Pros & Cons

Method	**Pros & Cons**
1. Basic methods	
Exhaustive Enumeration	**Pros** - Simple & intrinsic **Cons** - Grid search - *High complexity* - O(#grids^2)
2. Random methods	
Random Jump	**Pros** - Lower complexity by random sampling **Cons** - Result is determined by *chosen seeds* - Expected optimum should be first known for stopping criterion
Random Walk - decaying lambda	**Pros** - Lower complexity by random sampling **Cons** - Result is determined by the *initial point* - Each step is based on the previous step (*local minimum* - more significant than the gradient-based methods due to the randomness)
Random Walk - optimal lambda	**Pros** - Lower complexity by random sampling **Cons** - Result is determined by the *initial point* - Each step is based on the previous step (*local minimum* - more significant than the gradient-based methods due to the randomness)
3. Gradient-based methods	
Steepest Gradient Descent (SGD)	**Pros** - Fast & effective **Cons** - The gradients or Hessian matrix (Jacobian) should be known

<table>
<tr><td>Newton’s method</td><td rowspan="2">- Each step is based on the previous step (<u>local minimum</u> - not that significant as Random Walk due to the criterion - the steepest direction)
- <u>Greedy search</u></td></tr>
<tr><td>Marquardt’s method</td></tr>
<tr><td colspan="2">4. Stochastic Local Search methods</td></tr>
<tr><td>Iterated Local Search (ILS) method</td><td><u>Pros</u>
- Alleviation of the local minimum problem (vs. Random Walk & Gradient-based methods)
<u>Cons</u>
- Perturbation scheme (local search) design</td></tr>
<tr><td>Simulated Annealing (SA) method</td><td><u>Pros</u>
- Alleviation of the local minimum problem (vs. Random Walk & Gradient-based methods)
<u>Cons</u>
- Perturbation scheme (local search) design
- Temperature reduction mechanism design</td></tr>
<tr><td>Tabu Search (TS) method</td><td><u>Pros</u>
- Alleviation of the local minimum problem (vs. Random Walk & Gradient-based methods)
<u>Cons</u>
- Perturbation scheme (local search) design
- Temperature reduction mechanism design
- Tabu list design</td></tr>
</table>