Metaheuristic Binary Decision Diagram

Finding best BDD ordering on efficient way by metaheuristic solution

Hyobin Park (2021227327)

Binary Decision Diagram

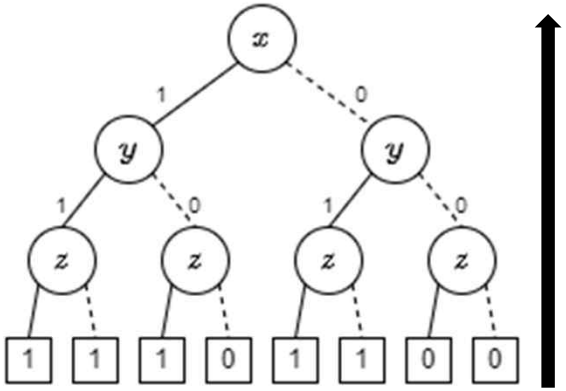


Fig 1. binary decision tree

(reduce)

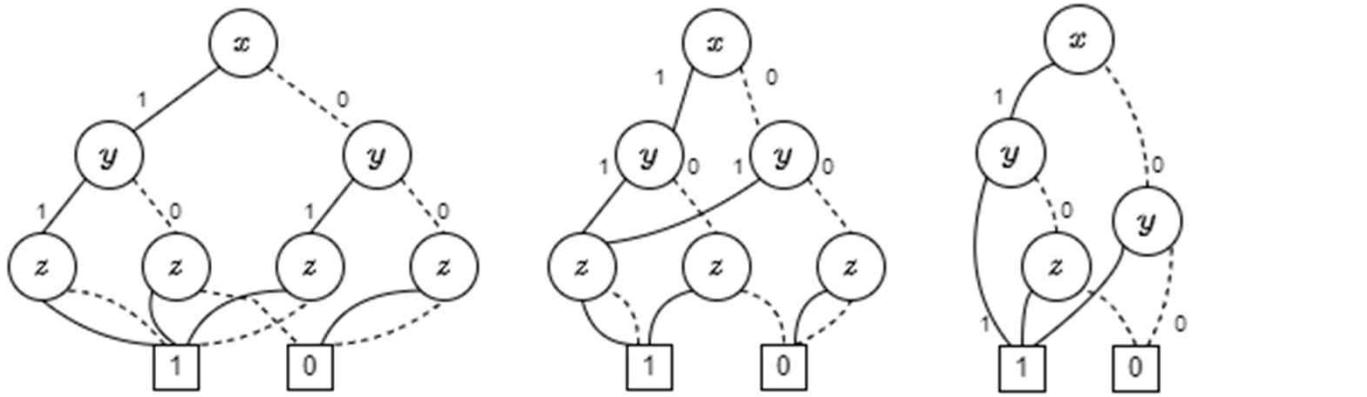


Fig 2. reduced ordered binary decision diagram

non-terminal nodes + terminal nodes

boolean variables 0 or 1

dashed line - - - - - 0

solid line ——— 1

Variable ordering problem (NP-hard)

$$f(a,b,c,d,e,f) = (a\&b)|\ (c\&d)|(e\&f)$$

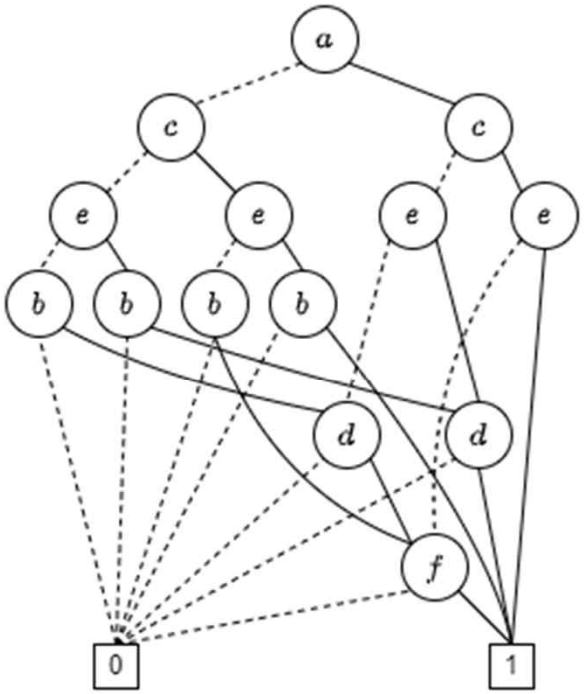


Fig 3. ROBDD ordering [a, c, e, b, d, f]

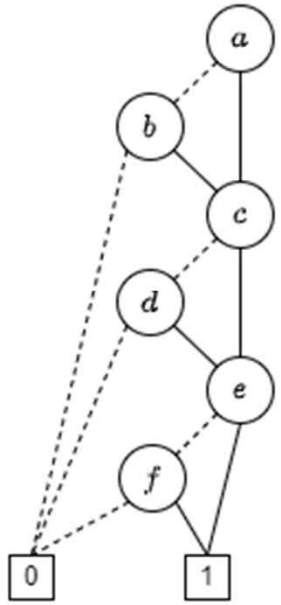
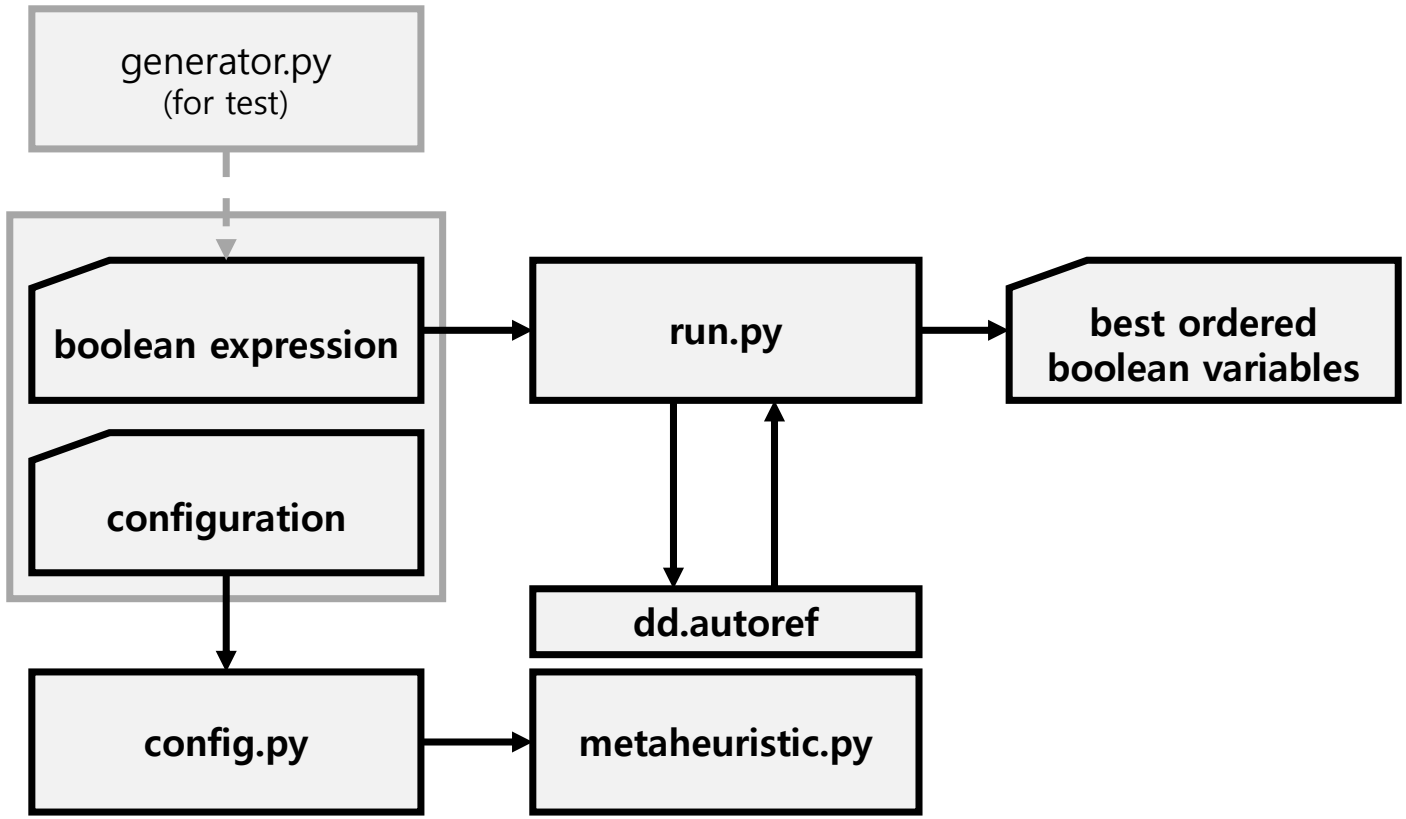


Fig 4. ROBDD ordering [a, b, c, d, e, f]

Overall Structure



Parameters (configuration)

• **POP_SIZE**	**creating population**	size of population
• **POOL_SIZE**	**crossover, mutation**	size of parent pool
• **N_LIMIT**	**stopping criteria**	iteration limitation
• **LIMIT_RATIO**	"	constant ratio for CRITERIA
• **CRITERIA**	"	minimum number of new offsprings which means no changing.

Individual ordering sequence of boolean variables. **all numbers are unique!**
These individuals make population with POP_SIZE.

for $[x_0, x_1, x_2, x_3, x_4, x_5]$ ...
[0, 1, 2, 3, 4, 5], [0, 2, 4, 1, 3, 5], [3, 4, 0, 5, 2, 1] **...**

Stopping Criteria

```
if len(newPop) < CRITERIA:
    nIter += 1
    if nIter == N_LIMIT:
        break
```

If there is not enough change, increase iteration limitation.
When that chance reaches to the limitation, the process is over.

Fitness building reordered BDD from individual, return number of new ordered BDD nodes.

For boolean function $f(x_1, \dots, x_n)$

$$fitness = N_{BDD\_nodes} \quad 2n + 2 \leq N_{BDD\_nodes} \leq 2^{n+1}$$

Selection FPS (Fitness Probability Selection)

```
def fps(pop):
    total_fitness = sum(fitness)
    weight = [f/total_fitness for f in fitness]
    return weight
```

Crossover keeping pivot and its backward, shuffle others.

[0, 1, 2, 3, 4, …, 7, 8, 9, 10] → [3, 7, 4, 1, 2, …, 6, 8, 9, 10]

Mutation just swap 2 variables to avoid local optima.

Design elitism.

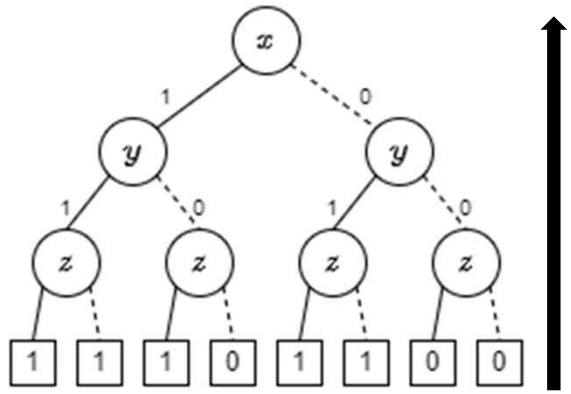
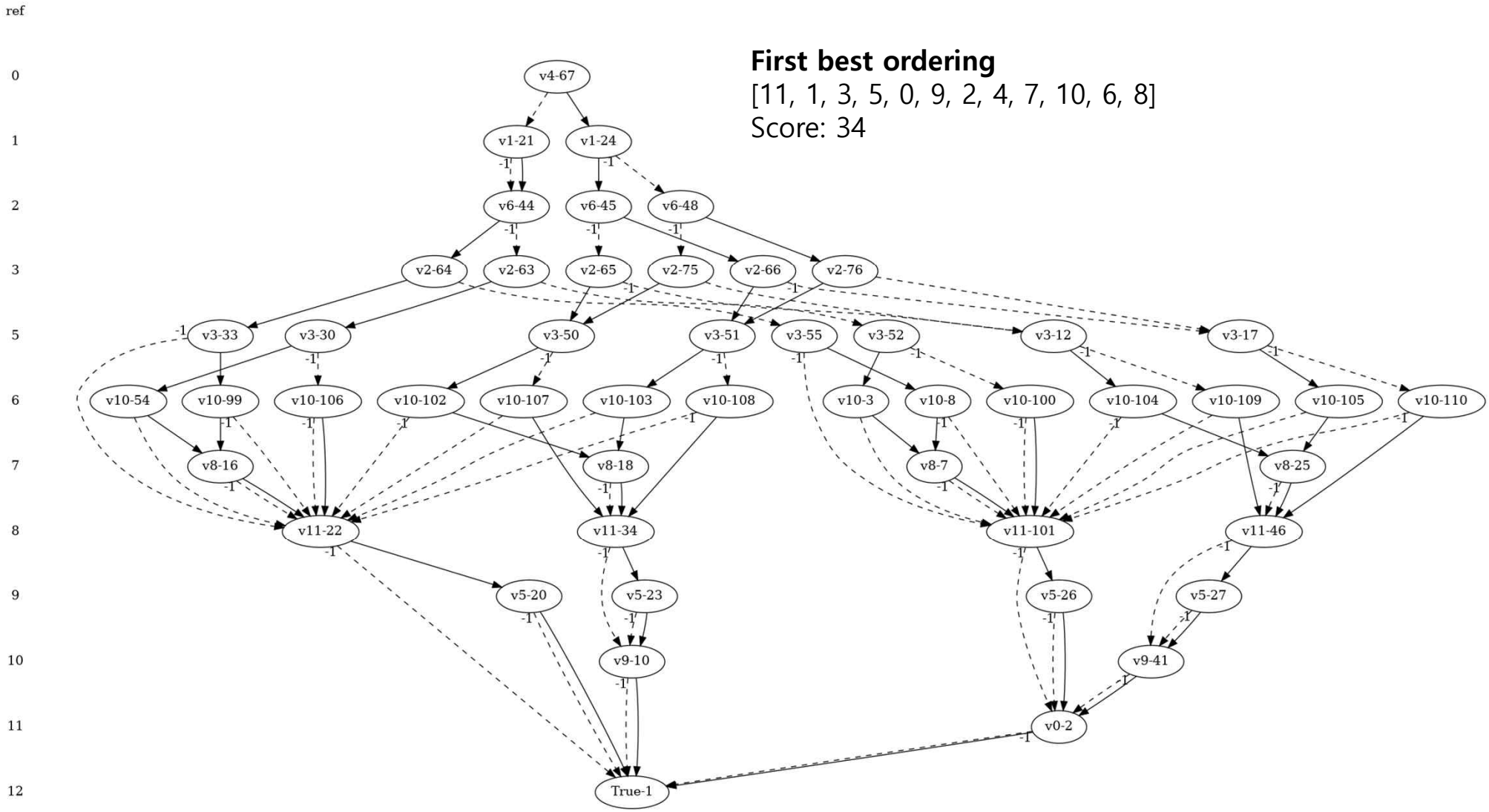
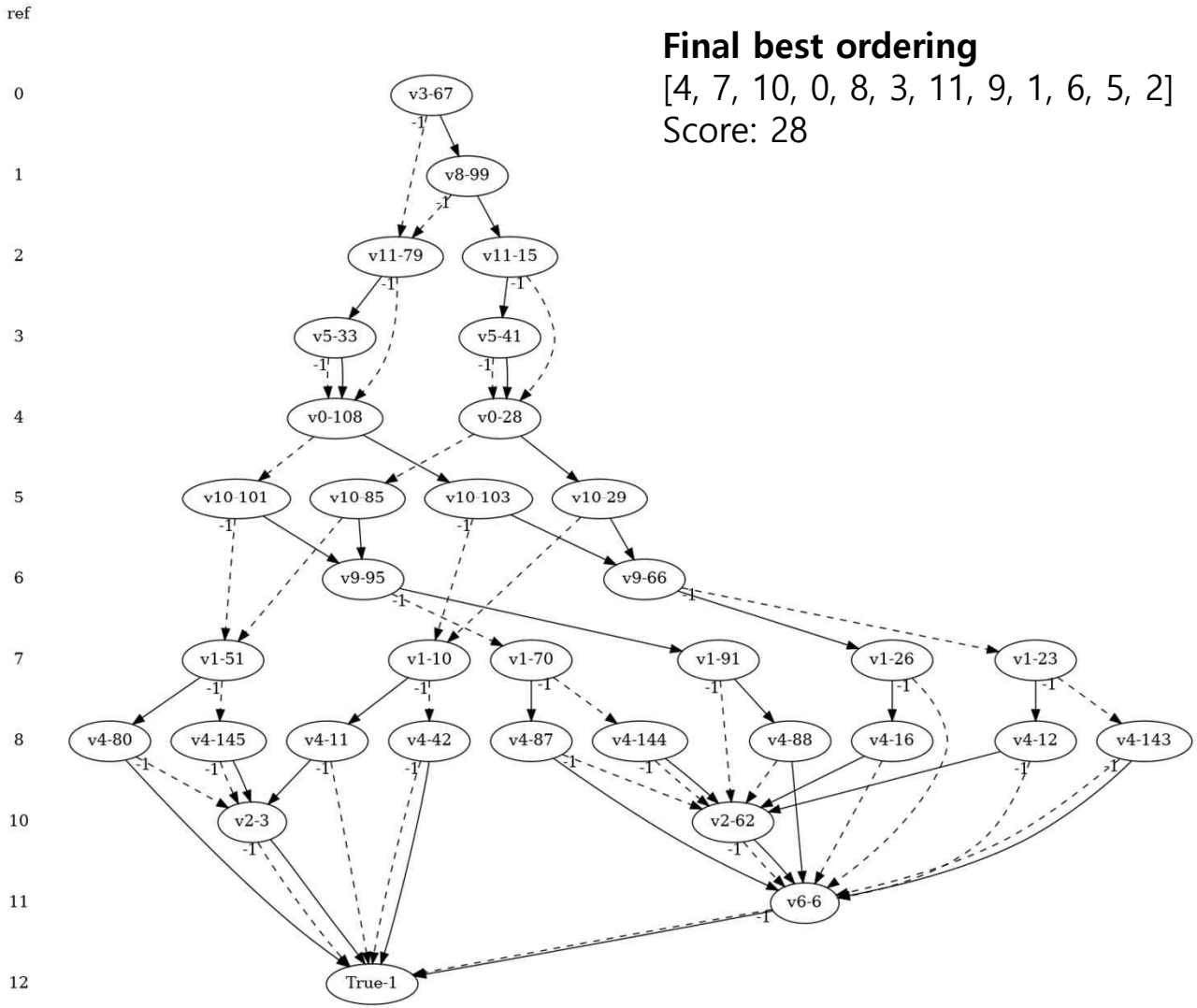


Fig 1. binary decision tree

```
Target expr: v0 | v2 ^ v10 & ~ v6 ^ ~ v9 | ~ v10 | ~ v4 & v9 ^ ~ v5 | ~ v11 ^ ~ v4 ^ ~ v3 | ~ v8 | ~ v10 ^ ~ v1 | v4 & ~ v2 ^ v1 & ~ v9 | ~ v7 | v7
Best Ind  : [11, 1, 3, 5, 0, 9, 2, 4, 7, 10, 6, 8]
Best Score: 34
Limitation: 0
======================================
Best Ind  : [5, 9, 10, 2, 8, 1, 11, 7, 6, 4, 0, 3]
Best Score: 32
Limitation: 0
======================================
Best Ind  : [6, 7, 0, 8, 5, 4, 1, 3, 10, 11, 9, 2]
Best Score: 31
Limitation: 0
======================================
Best Ind  : [3, 2, 0, 11, 4, 6, 1, 8, 5, 10, 9, 7]
Best Score: 30
Limitation: 0
======================================
Best Ind  : [4, 7, 10, 0, 8, 3, 11, 9, 1, 6, 5, 2]
Best Score: 28
Limitation: 0
======================================
Best Ind  : [4, 7, 10, 0, 8, 3, 11, 9, 1, 6, 5, 2]
Best Score: 28
```

Best Order [4, 7, 10, 0, 8, 3, 11, 9, 1, 6, 5, 2]
v10, v4, v3, v8, v1, v9, v6, v0, v2, v7, v11, v5


ref
0
1
2
3
5
6
7
8
9
10
11
12
v4-67
v1-21
v1-24
v6-44
v6-45
v6-48
v2-64
v2-63
v2-65
v2-75
v2-66
v2-76
v3-33
v3-30
v3-50
v3-51
v3-55
v3-52
v3-12
v3-17
v10-54
v10-99
v10-106
v10-102
v10-107
v10-103
v10-108
v10-3
v10-8
v10-100
v10-104
v10-109
v10-105
v10-110
v8-16
v8-18
v8-7
v8-25
v11-22
v11-34
v11-101
v11-46
v5-20
v5-23
v5-26
v5-27
v9-10
v9-41
v0-2
True-1
First best ordering
[11, 1, 3, 5, 0, 9, 2, 4, 7, 10, 6, 8]
Score: 34



Final best ordering
[4, 7, 10, 0, 8, 3, 11, 9, 1, 6, 5, 2]
Score: 28

Limitation

- **Time Consuming**
  - building BDD takes a lot of time.
  - only work in small size (no more than 30)

- **Mutation**
  - too weak to avoid local optima.

Further work

- **Mutation**
  - do more than swapping only two elements.

- **Applicability**
  - design for applying BDD-based model checker.