Opti-maze

Playing with QAOA

Nikita Astrakhantsev
Vasiliy Bokov
Alberto Colombo
Reinis Irmejs
Tom Westerhout

Goals

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;
  → focus on **combinatorial optimization** and **QAOA**;

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;
  → focus on **combinatorial optimization** and **QAOA**;

  → create a game that uses it;

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;
  → focus on **combinatorial optimization** and **QAOA**;

  → create a game that uses it;

- Apply quantum computing to real-world problems;

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;
  → focus on **combinatorial optimization** and **QAOA**;

  → create a game that uses it;
- Apply quantum computing to real-world problems;
  → focus on **climate change**

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;
  → focus on **combinatorial optimization** and **QAOA**;

  → create a game that uses it;
- Apply quantum computing to real-world problems;
  → focus on **climate change**

  → efficient resource allocation

Goals

- Popularize quantum computing by teaching about its capabilities and limitations;
  → focus on **combinatorial optimization** and **QAOA**;

  → create a game that uses it;
- Apply quantum computing to real-world problems;
  → focus on **climate change**

  → efficient resource allocation
- Develop a framework that could aid with **complicated optimization problems** in the future

Premise of the game:

Premise of the game:

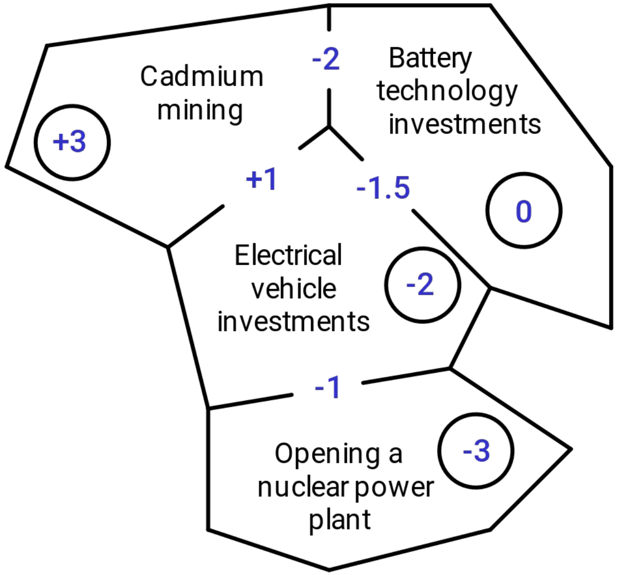
- In many popular games like Monopoly and Sid Mayer's Civilization, players have to find optimal solution to resource allocation problems.

Premise of the game:

- In many popular games like Monopoly and Sid Mayer's Civilization, players have to find optimal solution to resource allocation problems.

- In Opti-maze the goal of the player is to **perform optimization better** than the *Quantum Artificial Player* who is using the QAOA.

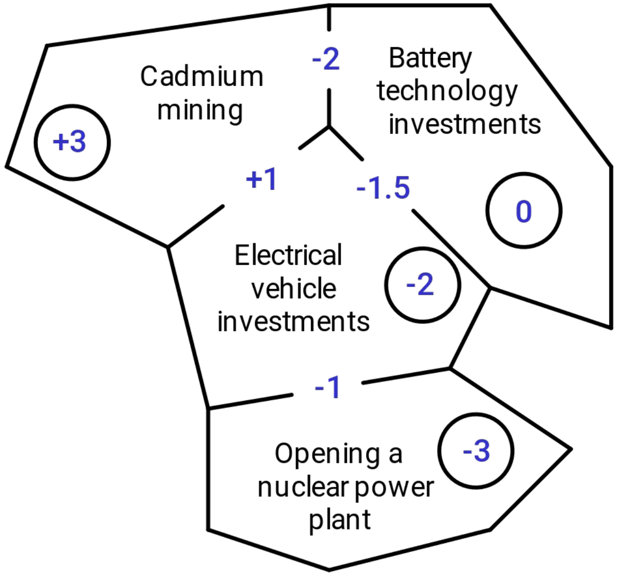
Example optimization problem


Cadmium mining
+3
-2
Battery technology investments
0
+1
-1.5
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

Example optimization problem

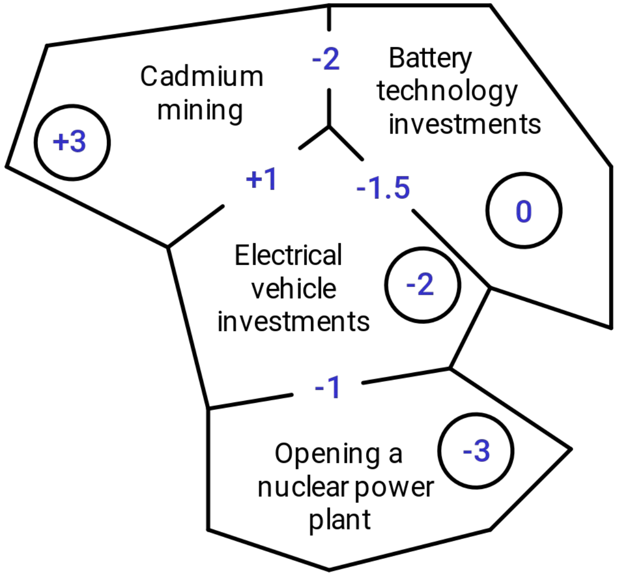
- Each investment has a cost (either positive or negative);


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

Example optimization problem

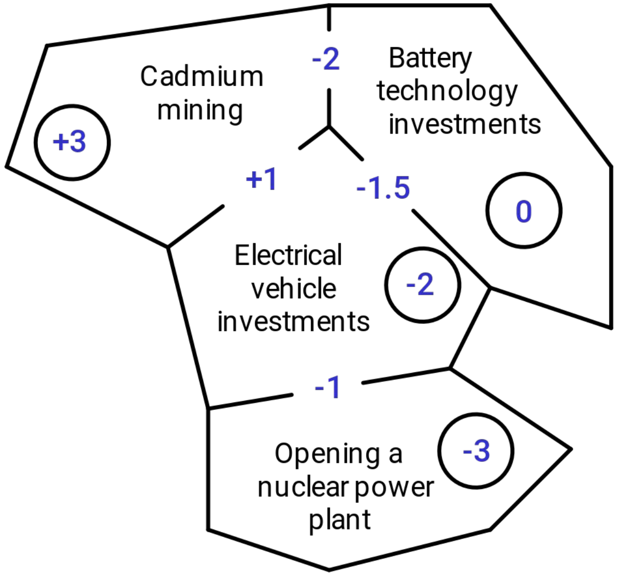
- Each investment has a cost (either positive or negative);
- Some combinations of investments may lead to synergy and lower the overall energy;


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

Example optimization problem

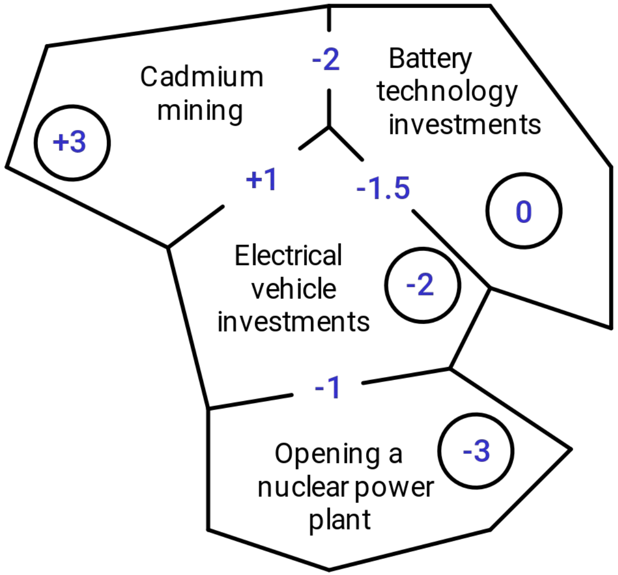
- Each investment has a cost (either positive or negative);
- Some combinations of investments may lead to synergy and lower the overall energy;
- The problem is mapped to a classical Ising-like model (no higher-order terms to reduce the circuit depth);


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

Example optimization problem

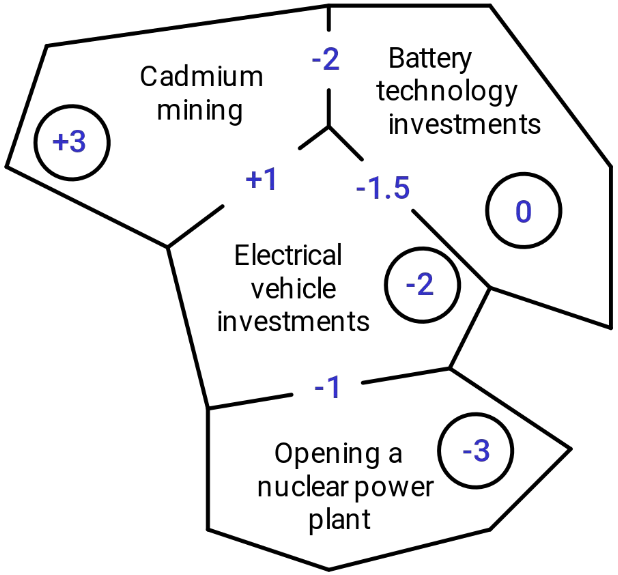
- Each investment has a cost (either positive or negative);
- Some combinations of investments may lead to synergy and lower the overall energy;
- The problem is mapped to a classical Ising-like model (no higher-order terms to reduce the circuit depth);
- The problem is NP-hard in general;


Cadmium mining
-2
Battery technology investments
+3
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

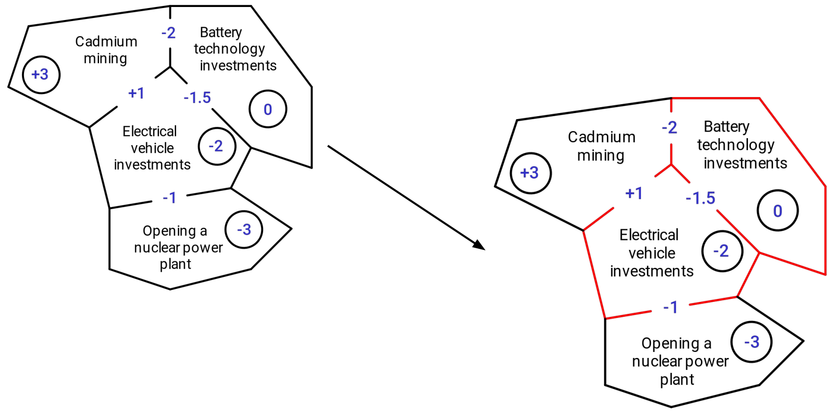
Example optimization problem

- Each investment has a cost (either positive or negative);
- Some combinations of investments may lead to synergy and lower the overall energy;
- The problem is mapped to a classical Ising-like model (no higher-order terms to reduce the circuit depth);
- The problem is NP-hard in general;
- Any two regions can interact – maps well to IonQ's topology;


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

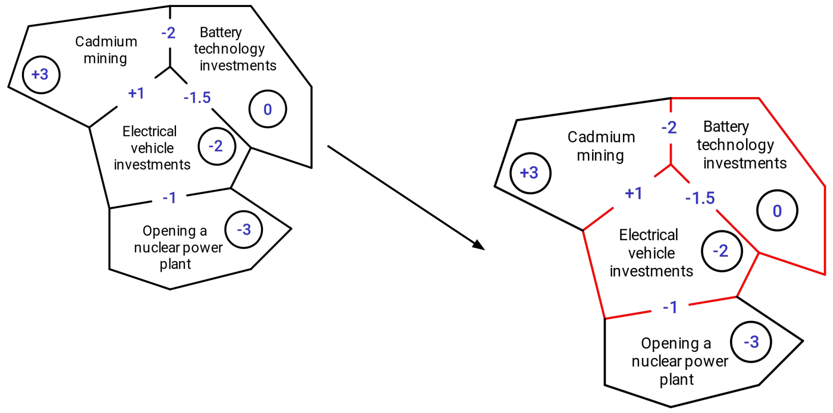
Approach: Opti-maze, a game



Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

Approach: Opti-maze, a game

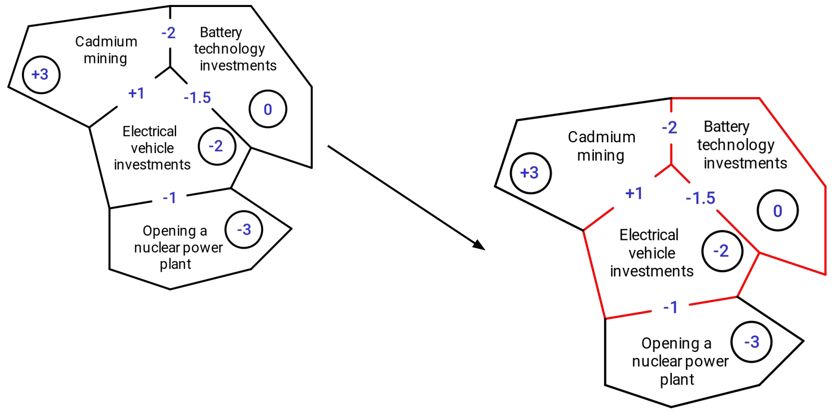
General idea: try to minimize the carbon emissions by selecting areas in which to invest


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

Approach: Opti-maze, a game

General idea: try to minimize the carbon emissions by selecting areas in which to invest

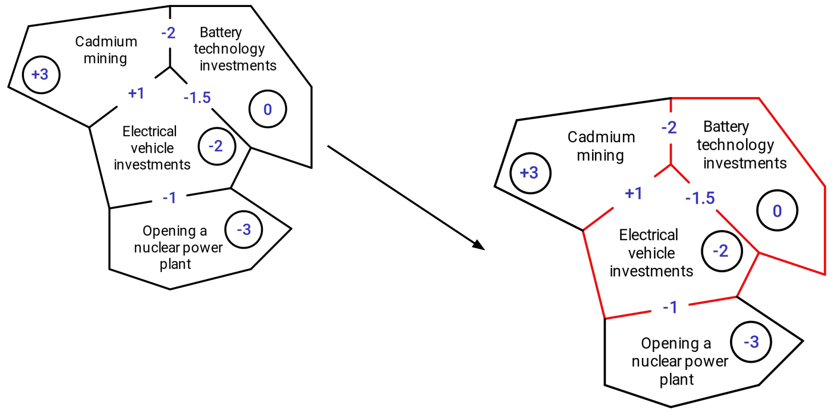
- Two modes:


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

Approach: Opti-maze, a game

General idea: try to minimize the carbon emissions by selecting areas in which to invest

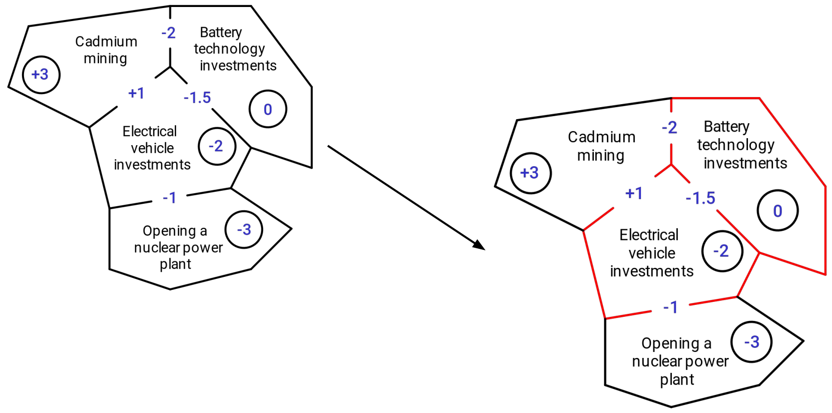
- Two modes:
    - Standard: compete against QAOA


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

Approach: Opti-maze, a game

General idea: try to minimize the carbon emissions by selecting areas in which to invest

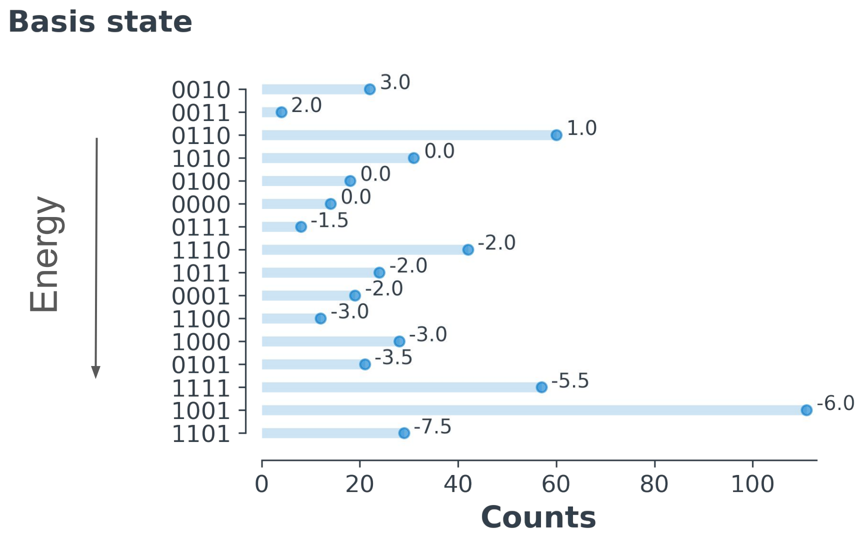
- Two modes:
  - Standard: compete against QAOA
  - Cooperative: use QAOA as a tool

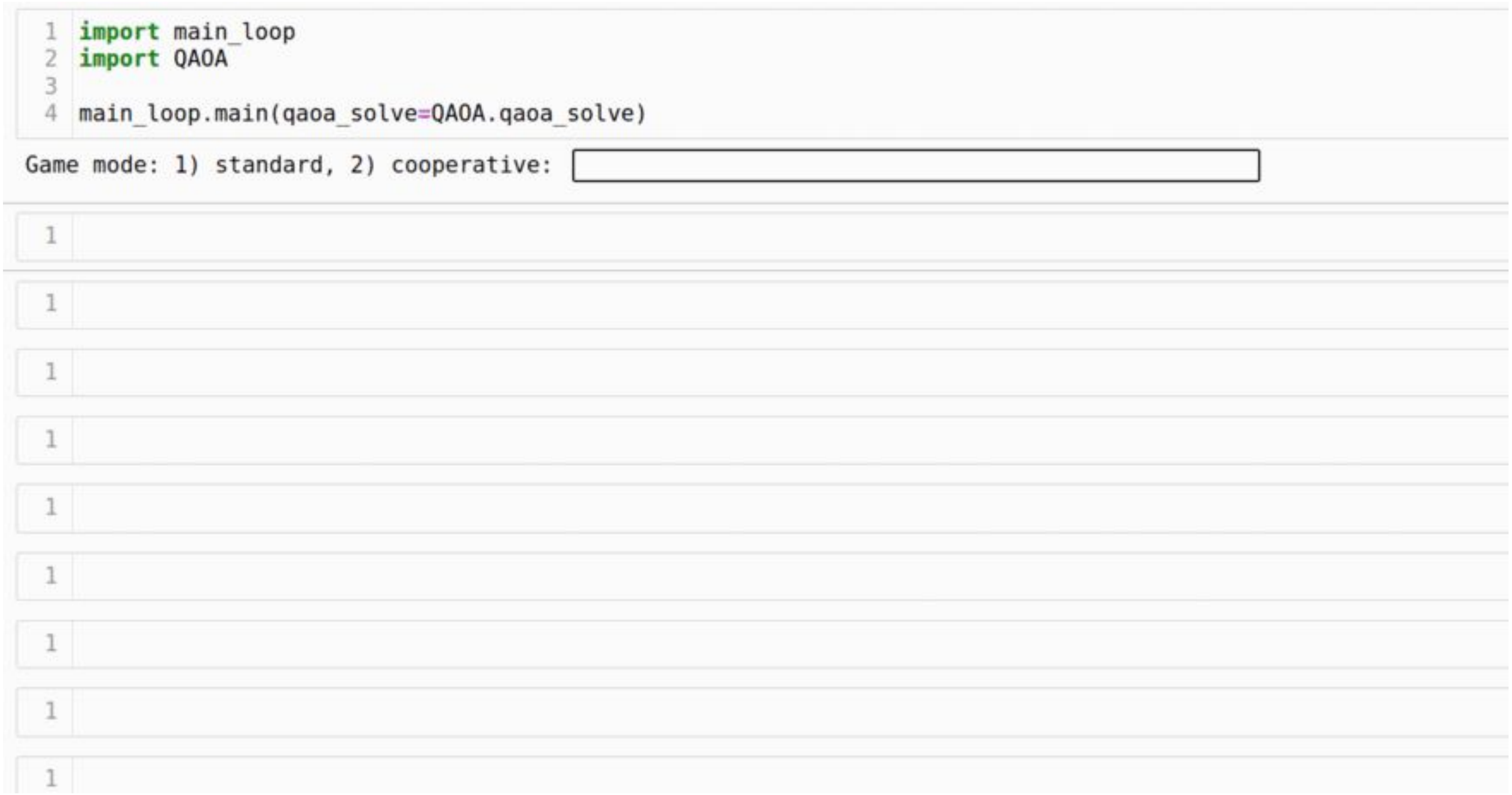


Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3

Standard mode: can you beat IonQ?

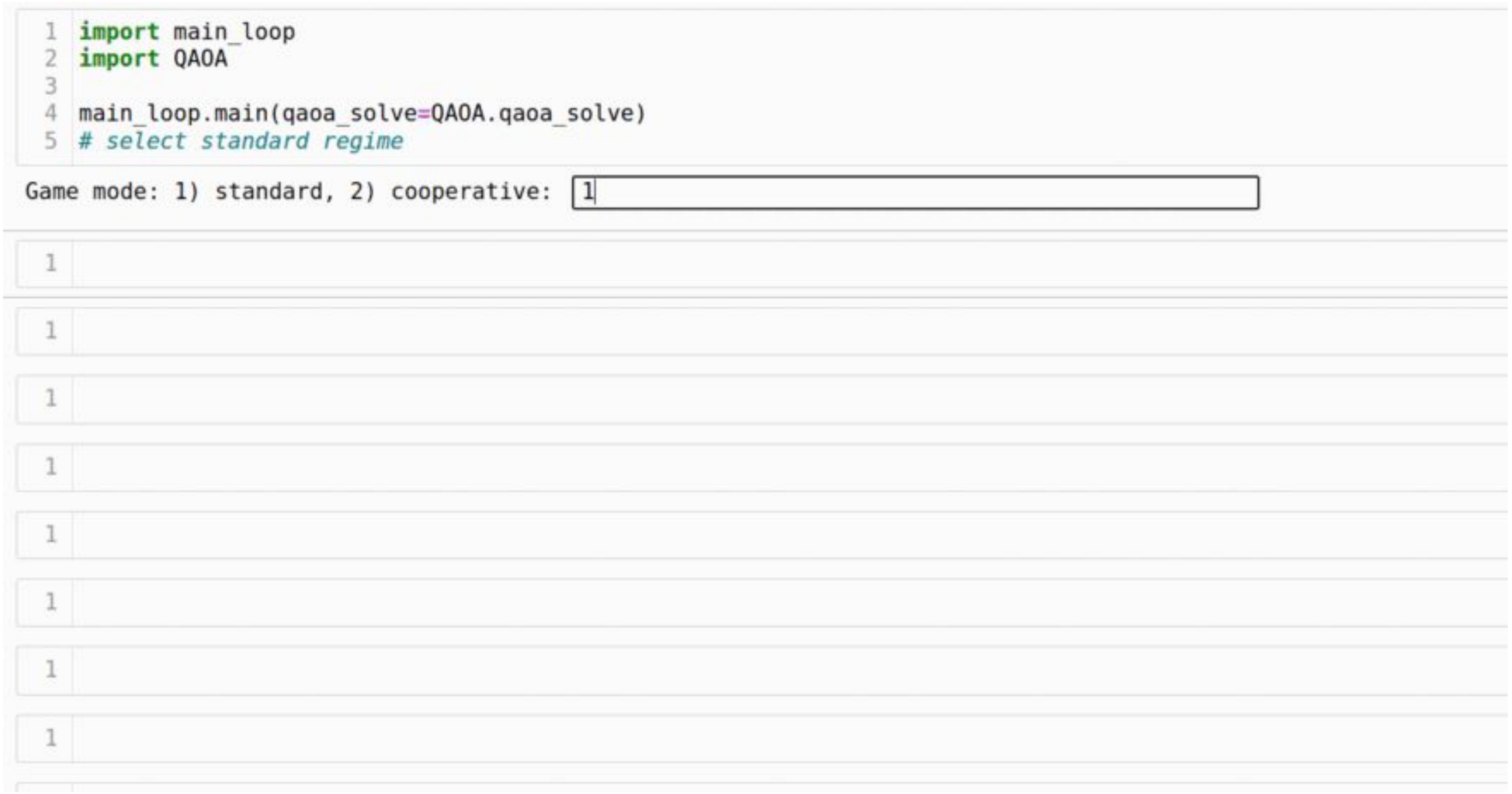
- Player tries to obtain a lower energy than QAOA;
- Number of shots in QAOA defines the difficulty level;





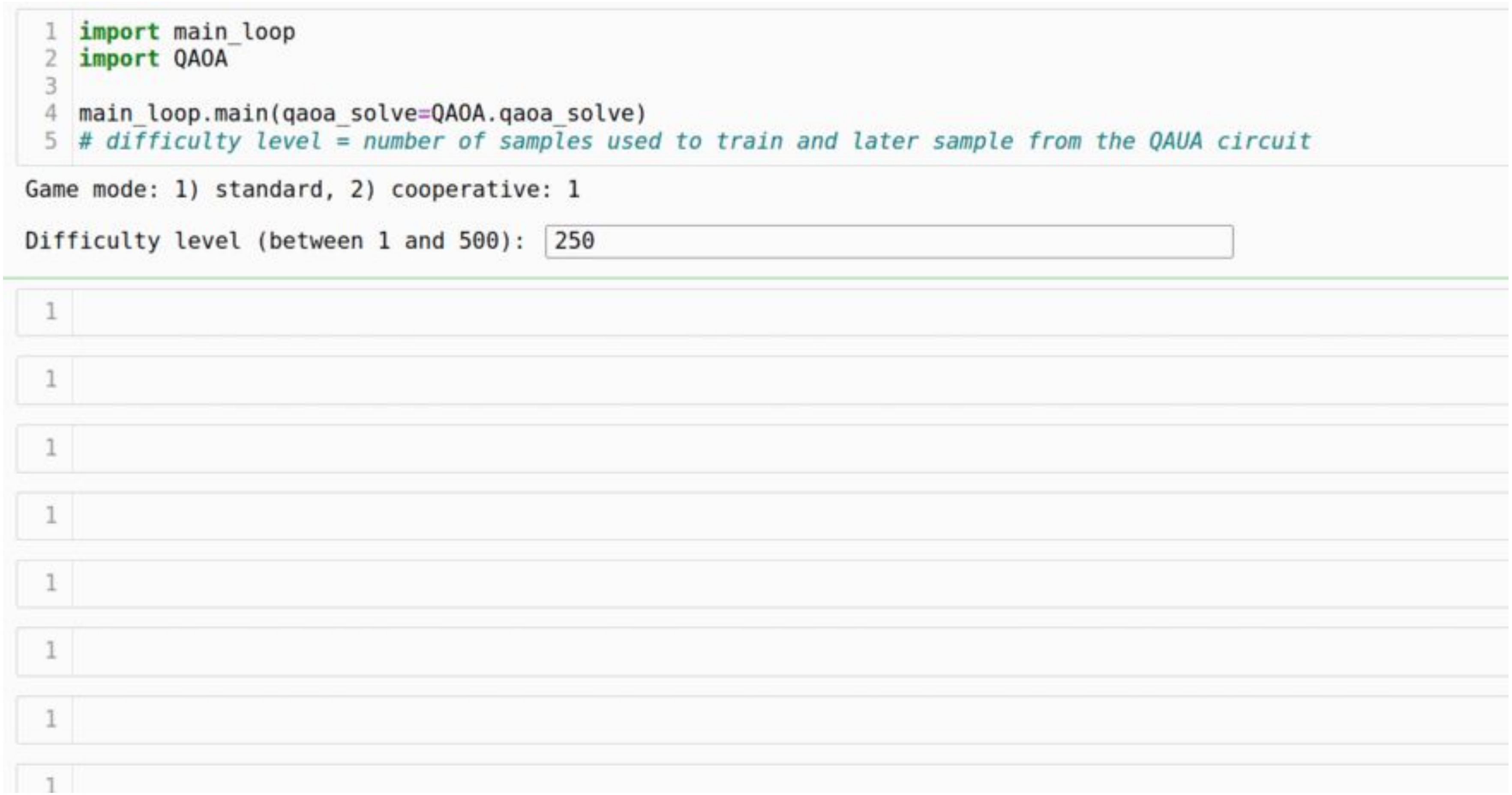
```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
```

```
Game mode: 1) standard, 2) cooperative:
```



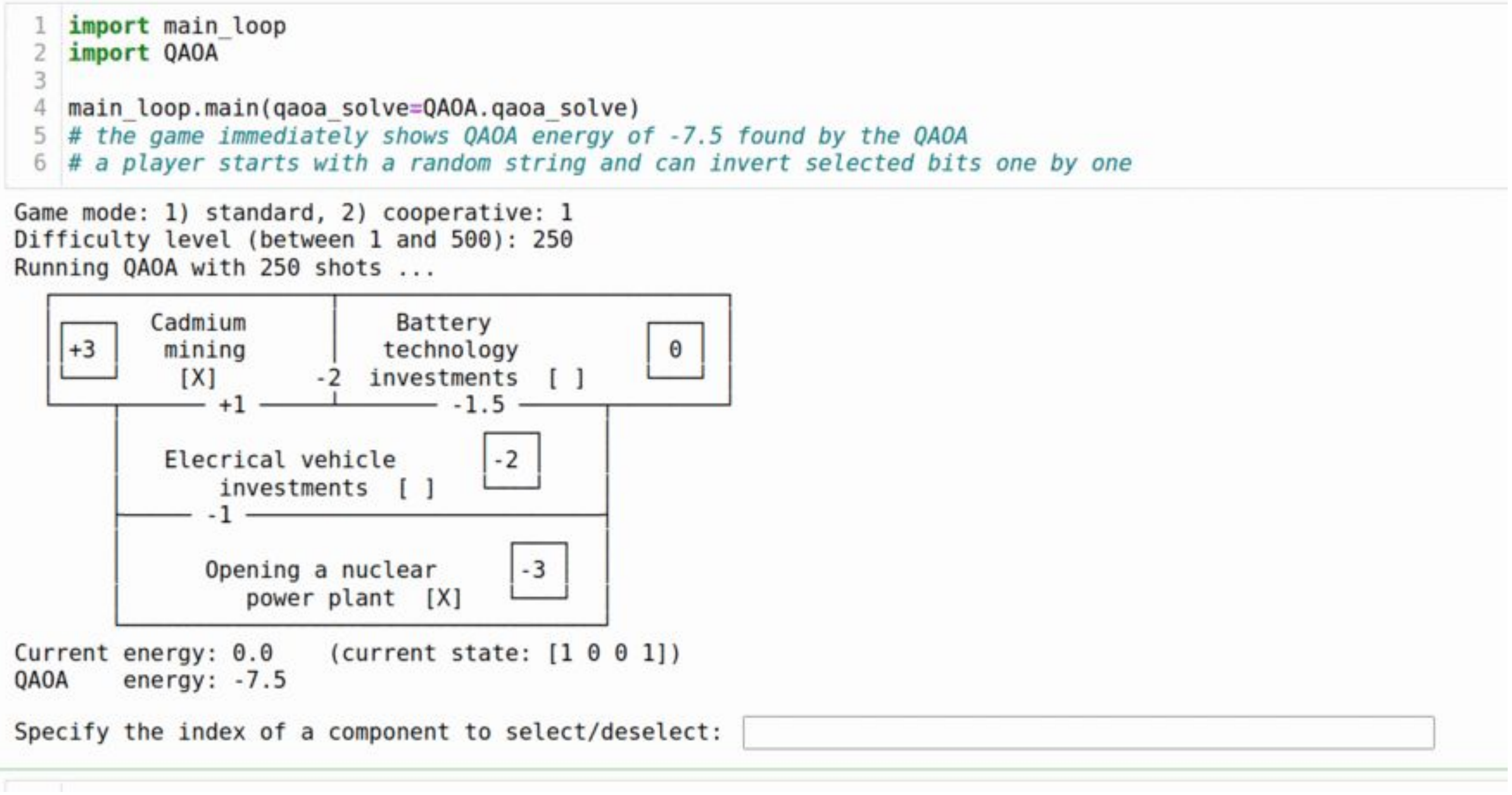
```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# select standard regime
```

```
Game mode: 1) standard, 2) cooperative: 1
```



```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# difficulty level = number of samples used to train and later sample from the QAUA circuit
```

```
Game mode: 1) standard, 2) cooperative: 1

Difficulty level (between 1 and 500): 250
```

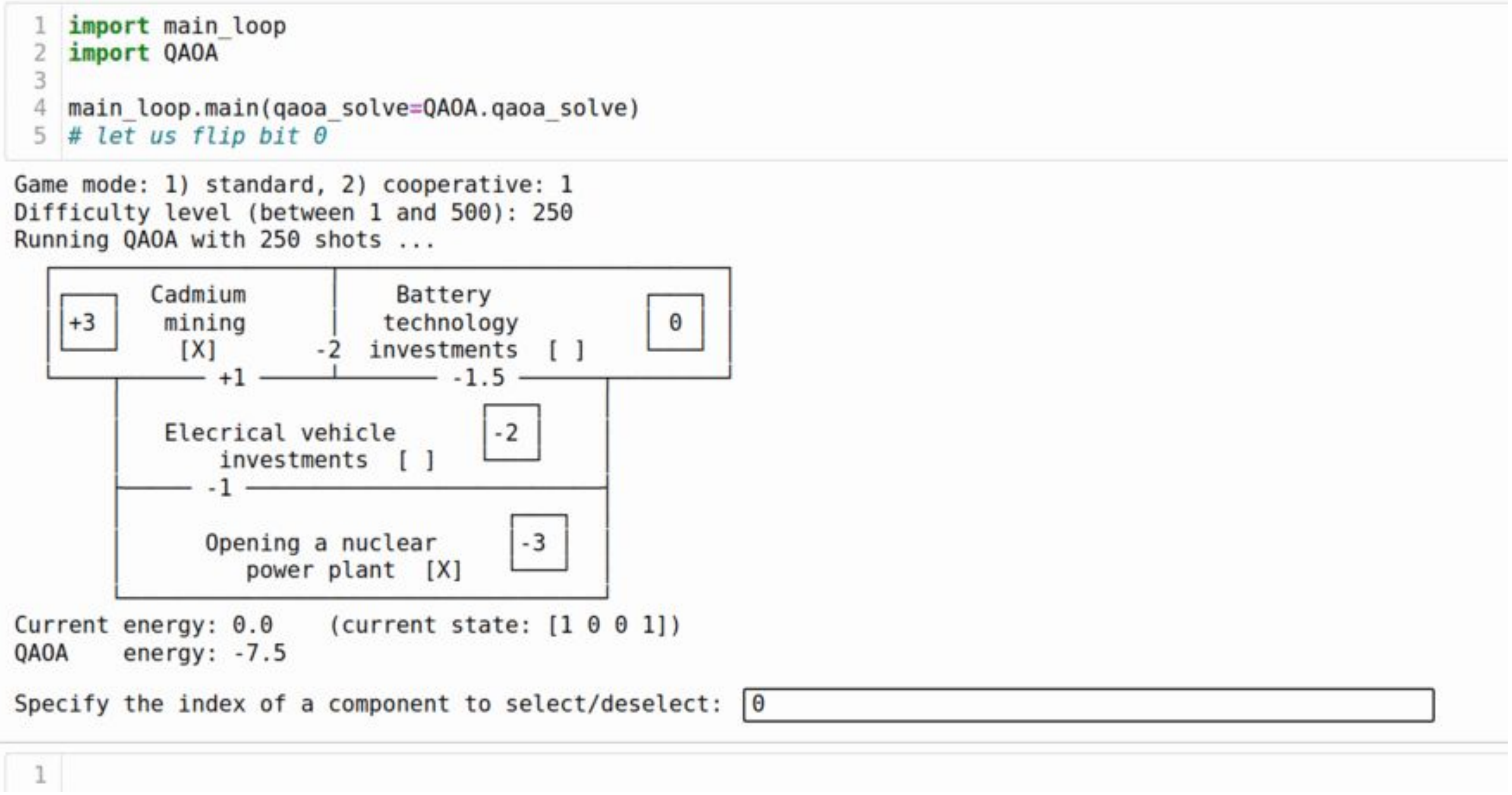


```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# the game immediately shows QAOA energy of -7.5 found by the QAOA
# a player starts with a random string and can invert selected bits one by one
```

```
Game mode: 1) standard, 2) cooperative: 1
Difficulty level (between 1 and 500): 250
Running QAOA with 250 shots ...
```

+3
Cadmium mining [X]
-2
Battery technology investments [ ]
0
+1
-1.5
Elecrical vehicle investments [ ]
-2
-1
Opening a nuclear power plant [X]
-3

```
Current energy: 0.0    (current state: [1 0 0 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect:
```

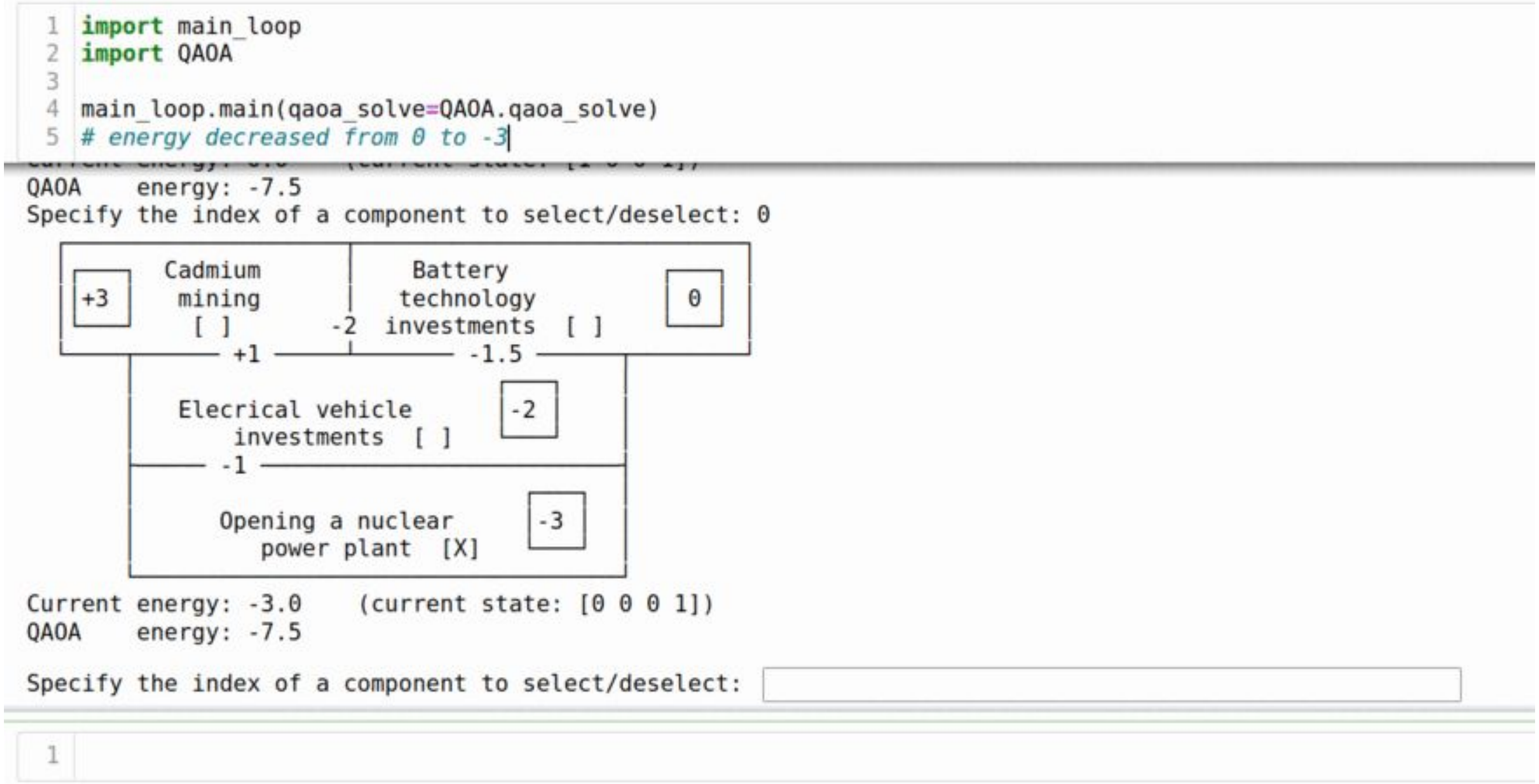


```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# let us flip bit 0
```

```
Game mode: 1) standard, 2) cooperative: 1
Difficulty level (between 1 and 500): 250
Running QAOA with 250 shots ...
```

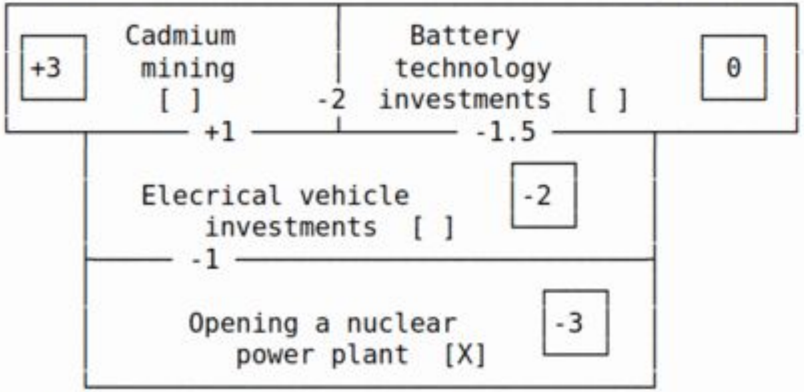
+3
Cadmium mining [X]
-2
Battery technology investments [ ]
0
+1
-1.5
Elecrical vehicle investments [ ]
-2
-1
Opening a nuclear power plant [X]
-3

```
Current energy: 0.0    (current state: [1 0 0 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect: 0
```



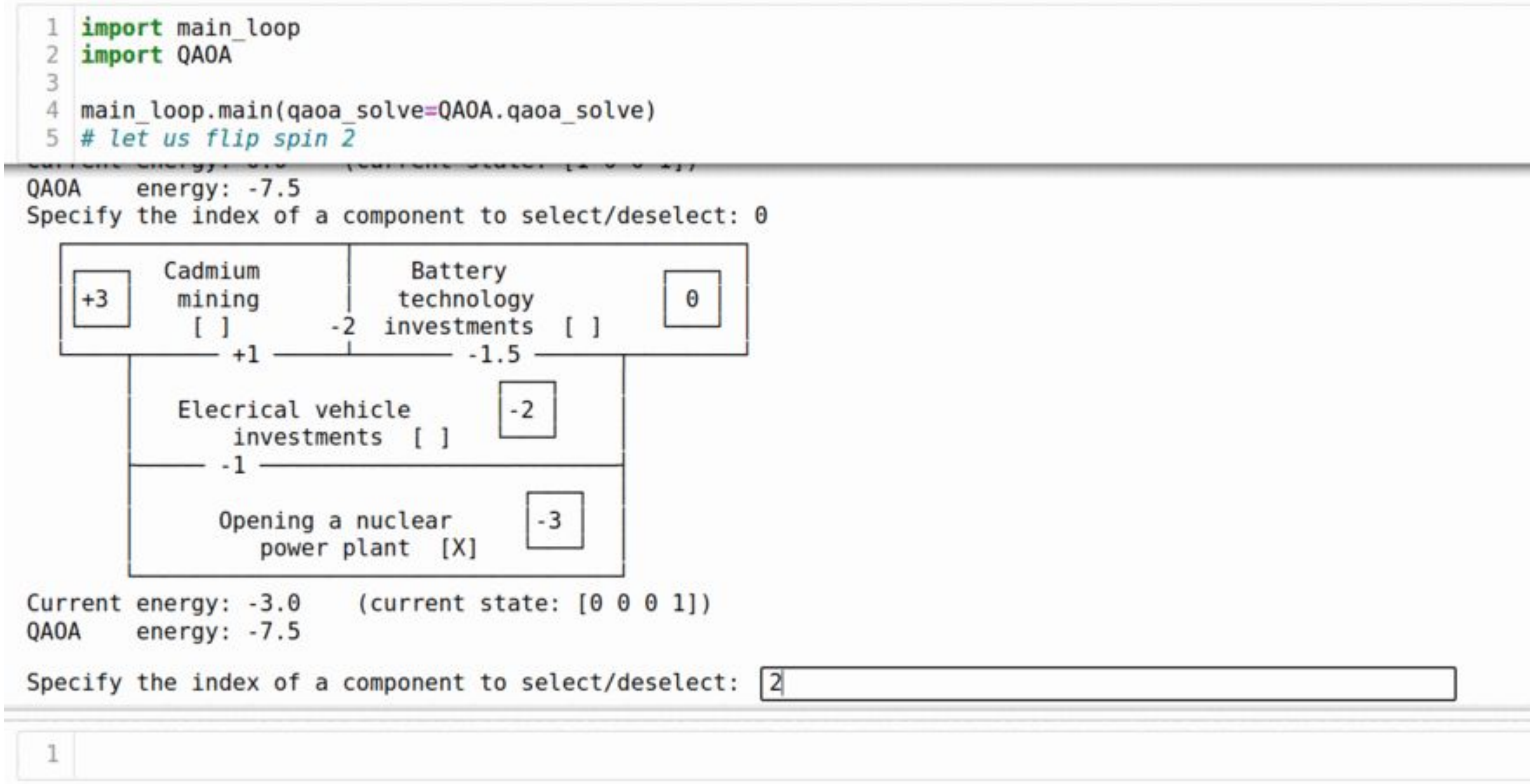
```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# energy decreased from 0 to -3
```

```
QAOA    energy: -7.5
Specify the index of a component to select/deselect: 0
```



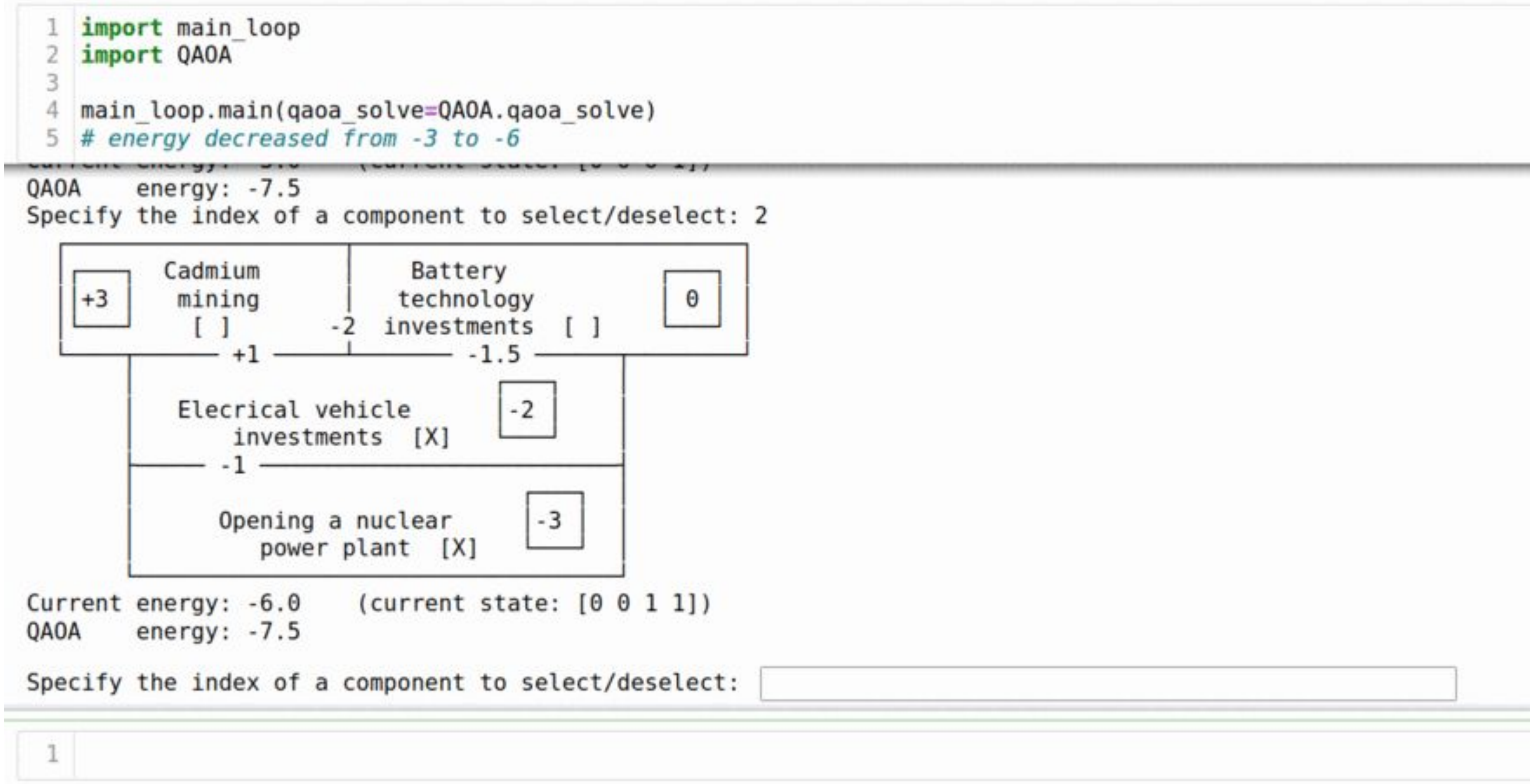
+3
Cadmium mining [ ]
-2
Battery technology investments [ ]
0
+1
-1.5
Elecrical vehicle investments [ ]
-2
-1
Opening a nuclear power plant [X]
-3

```
Current energy: -3.0    (current state: [0 0 0 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect:
```



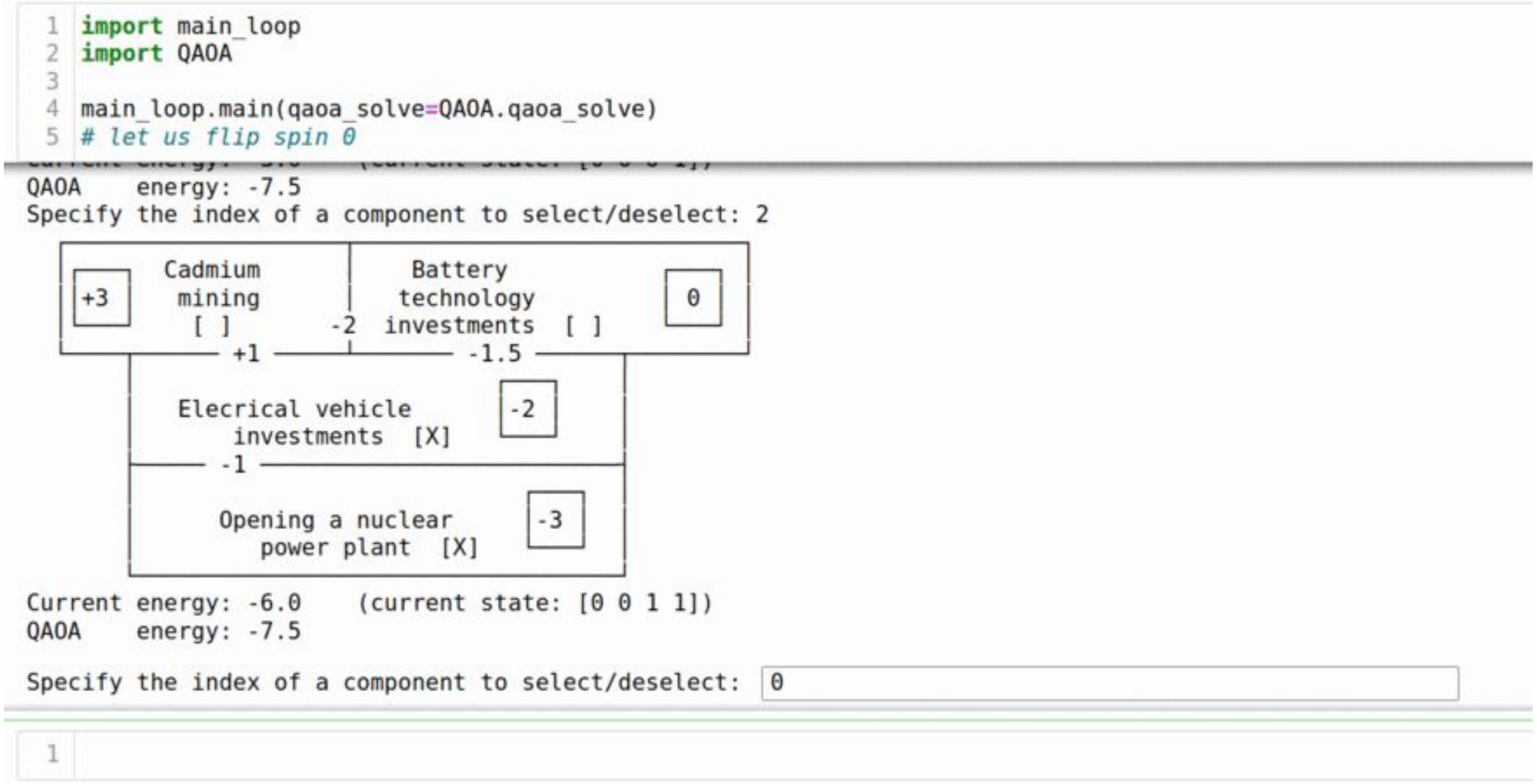
```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# let us flip spin 2
```

```
QAOA    energy: -7.5
Specify the index of a component to select/deselect: 0

 +3    Cadmium              Battery            0
        mining             technology
         [ ]          -2  investments  [ ]
               +1               -1.5

        Elecrical vehicle               -2
            investments  [ ]
            -1

           Opening a nuclear              -3
              power plant  [X]

Current energy: -3.0    (current state: [0 0 0 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect: 2
```



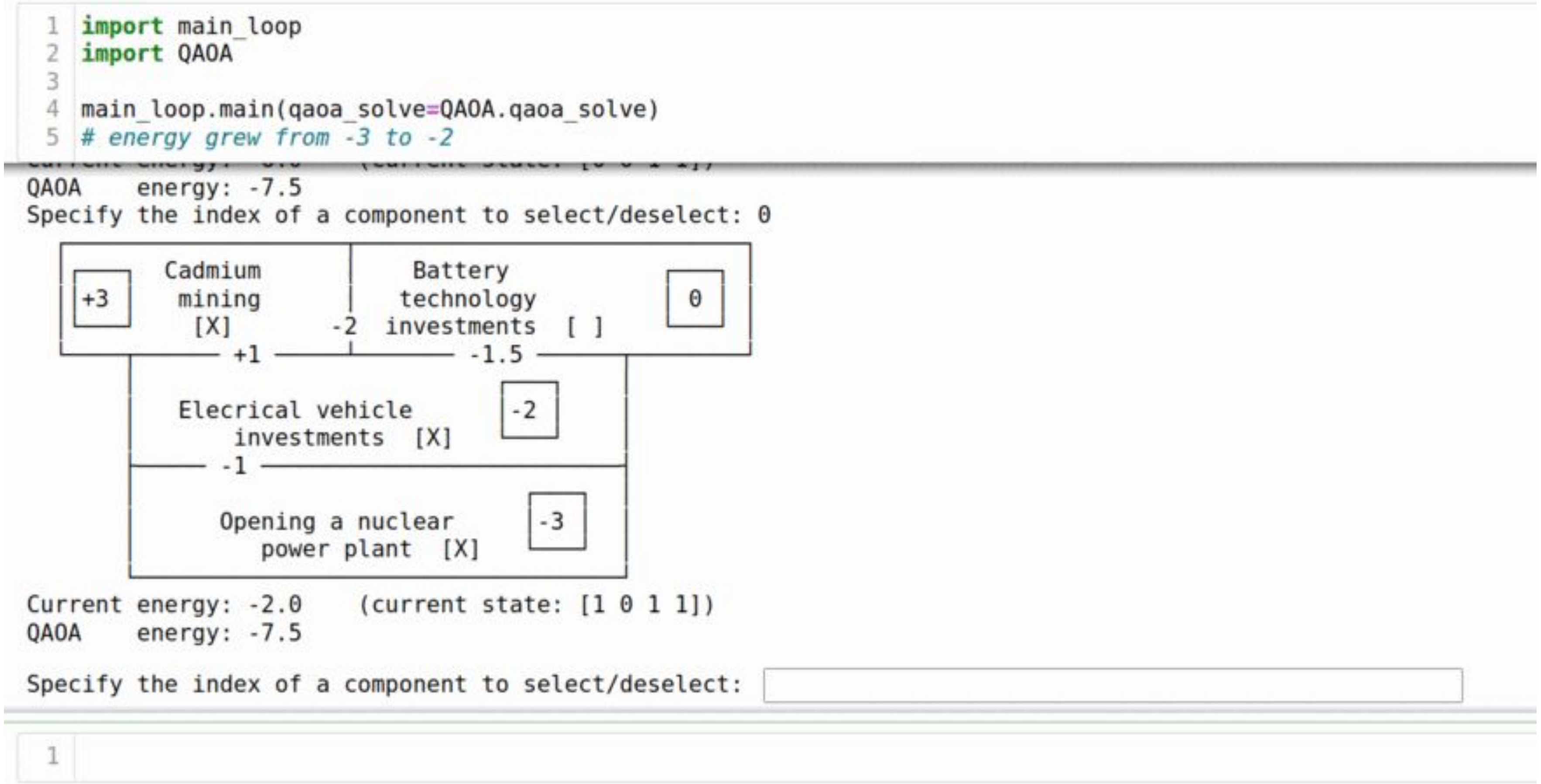
```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# energy decreased from -3 to -6
```

```
QAOA    energy: -7.5
Specify the index of a component to select/deselect: 2

  +----------------------+----------------------------+
  | +3 |   Cadmium       |     Battery           | 0 ||
  |        mining        |    technology              |
  |         [ ]       -2 |  investments  [ ]          |
  +-------- +1 ----------+--------- -1.5 ---+---------+
      |                                     |
      |  Elecrical vehicle          | -2 |  |
      |      investments  [X]               |
      +-------- -1 -------------------------+
      |                                     |
      |    Opening a nuclear        | -3 |  |
      |       power plant  [X]              |
      +-------------------------------------+

Current energy: -6.0    (current state: [0 0 1 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect:
```



```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# let us flip spin 0
```

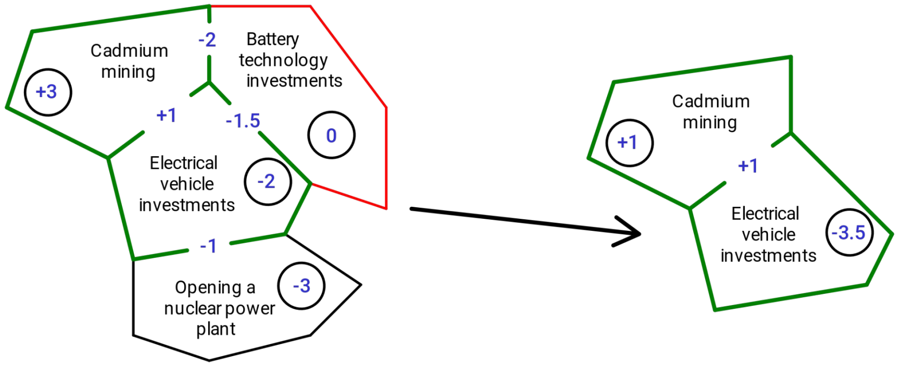
```
QAOA    energy: -7.5
Specify the index of a component to select/deselect: 2

 +----------------------+-------------------------------------+
 | +3 |   Cadmium       |      Battery                  | 0 | |
 |        mining        |     technology                      |
 |         [ ]       -2 |   investments   [ ]                 |
 +---------- +1 --------+------------ -1.5 ------------------+
      |                                      |
      |   Elecrical vehicle            | -2 ||
      |        investments   [X]             |
      +------ -1 ----------------------------+
      |                                      |
      |     Opening a nuclear          | -3 ||
      |        power plant   [X]             |
      +--------------------------------------+

Current energy: -6.0    (current state: [0 0 1 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect:  0
```



```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# energy grew from -3 to -2
```

```
QAOA    energy: -7.5
Specify the index of a component to select/deselect: 0

 ┌─────┬───────────────────┬──────────────────────────────────┬─────┐
 │ +3  │   Cadmium         │   Battery                        │  0  │
 │     │   mining          │   technology                     │     │
 │     │   [X]          -2 │  investments   [ ]               │     │
 └──┬────────── +1 ───────┴──────────── -1.5 ──────┬──────────────────┘
    │                                              │
    │  Elecrical vehicle                  ┌────┐   │
    │      investments   [X]              │ -2 │   │
    │                                     └────┘   │
    ├──────── -1 ──────────────────────────────────┤
    │                                              │
    │      Opening a nuclear                ┌────┐ │
    │         power plant   [X]             │ -3 │ │
    │                                       └────┘ │
    └──────────────────────────────────────────────┘

Current energy: -2.0    (current state: [1 0 1 1])
QAOA    energy: -7.5

Specify the index of a component to select/deselect:
```

Cooperation mode: QAOA to the rescue!

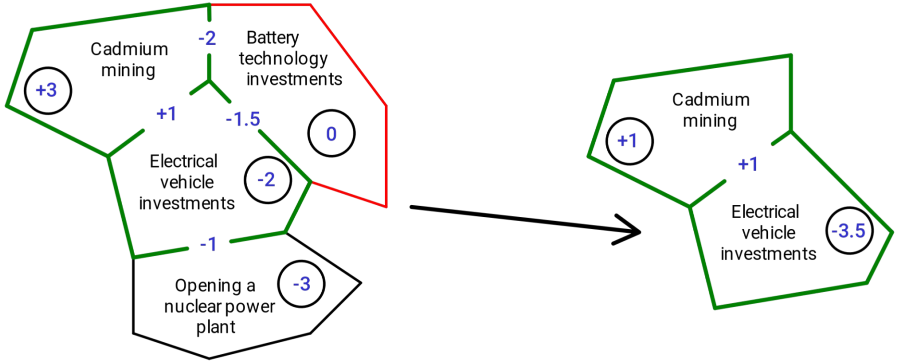

Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+1
+1
Electrical vehicle investments
-3.5

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

$$H' = \sum_{i,j \in \mathcal{G}} J_{ij} S_i S_j + \sum_{i \in \mathcal{G}} \big(h_i + \sum_{j \notin \mathcal{G}} J_{ij} S_j\big) \cdot S_i$$

Cooperation mode: QAOA to the rescue!

- The user can still select/unselect regions of investment;



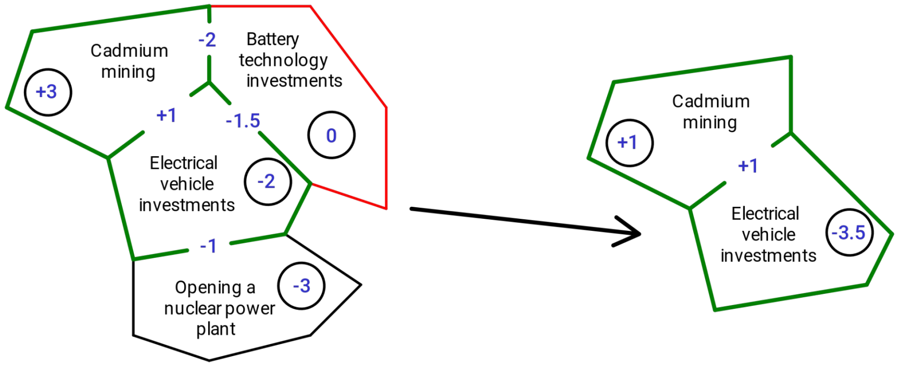
Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+1
+1
Electrical vehicle investments
-3.5

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

$$H' = \sum_{i,j \in \mathcal{G}} J_{ij} S_i S_j + \sum_{i \in \mathcal{G}} \left(h_i + \sum_{j \notin \mathcal{G}} J_{ij} S_j\right) \cdot S_i$$

Cooperation mode: QAOA to the rescue!

- The user can still select/unselect regions of investment;
- No QAOA solution to compete against;

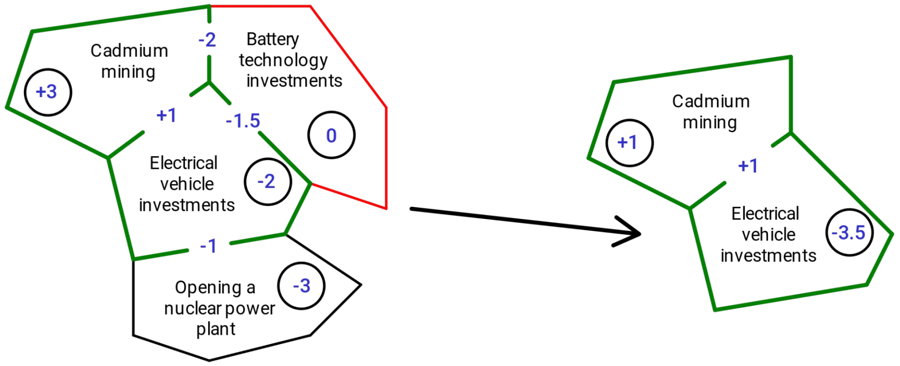

Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+1
+1
Electrical vehicle investments
-3.5

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

$$H' = \sum_{i,j \in \mathcal{G}} J_{ij} S_i S_j + \sum_{i \in \mathcal{G}} \left(h_i + \sum_{j \notin \mathcal{G}} J_{ij} S_j\right) \cdot S_i$$

Cooperation mode: QAOA to the rescue!

- The user can still select/unselect regions of investment;
- No QAOA solution to compete against;
- QAOA can be used to solve subproblems;

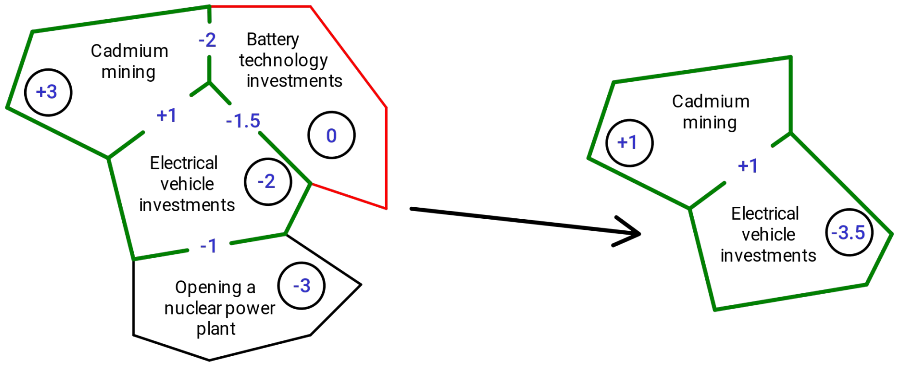

Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+1
+1
Electrical vehicle investments
-3.5

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

$$H' = \sum_{i,j \in \mathcal{G}} J_{ij} S_i S_j + \sum_{i \in \mathcal{G}} \left(h_i + \sum_{j \notin \mathcal{G}} J_{ij} S_j\right) \cdot S_i$$

Cooperation mode: QAOA to the rescue!

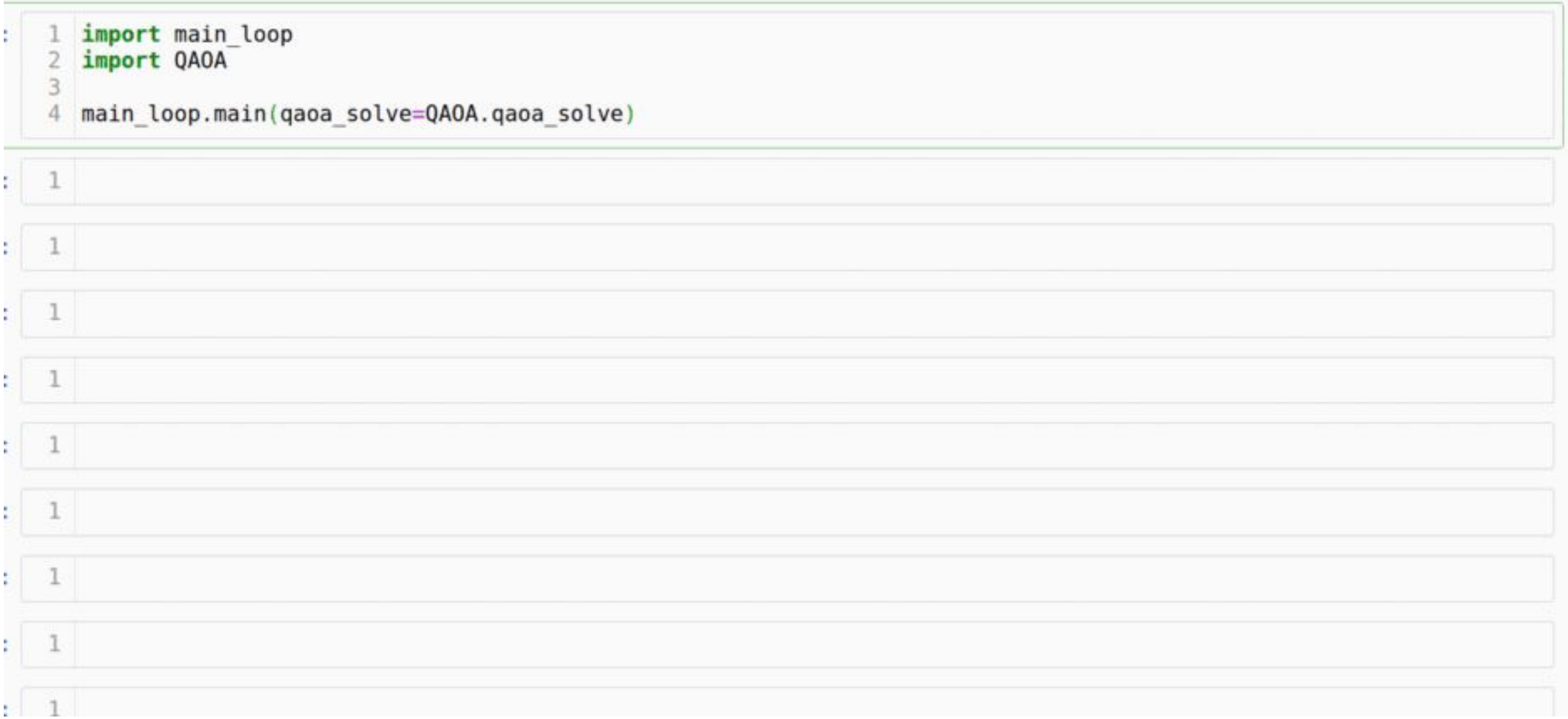
- The user can still select/unselect regions of investment;
- No QAOA solution to compete against;
- QAOA can be used to solve subproblems;
- Applicable to cases when the whole graph does not fit into a quantum computer;



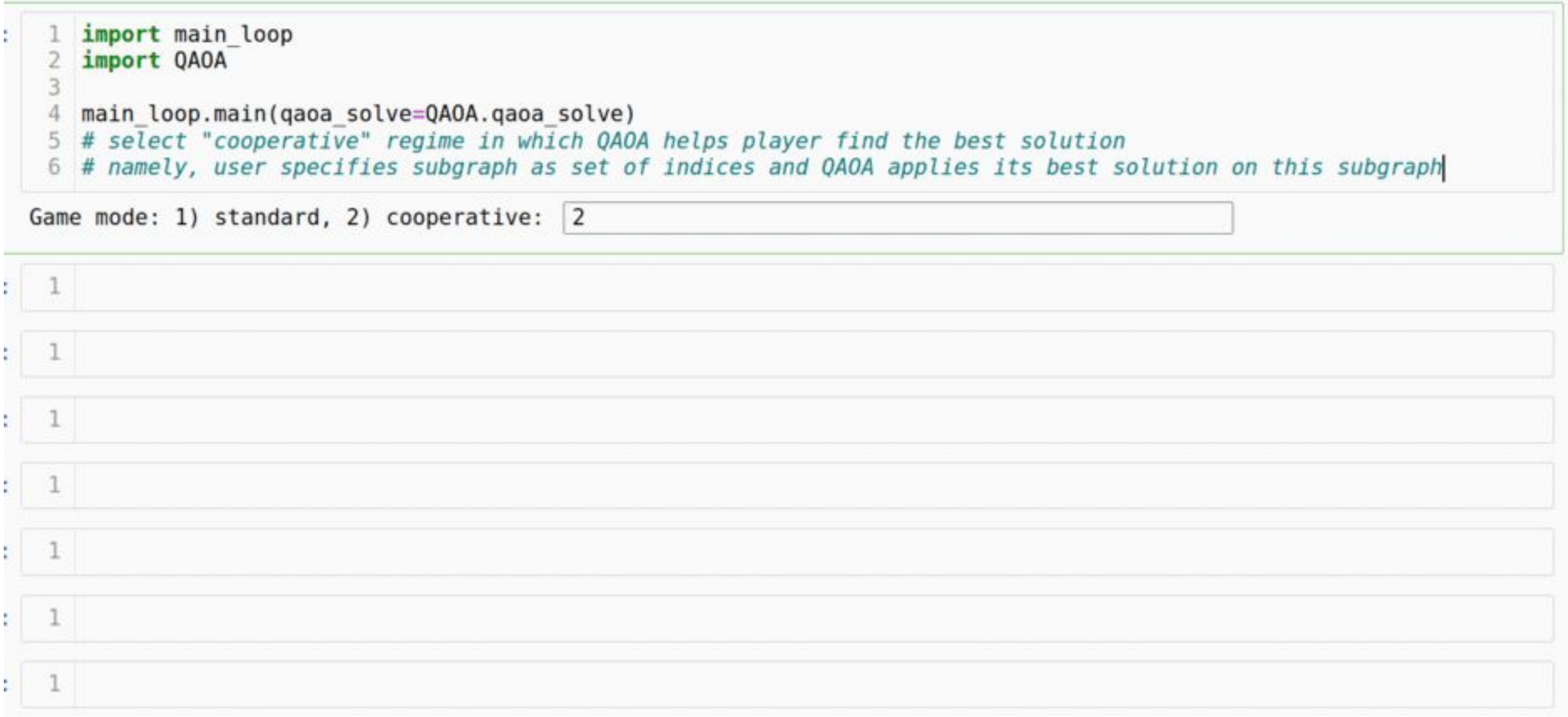
Cadmium mining
+3
-2
Battery technology investments
+1
-1.5
0
Electrical vehicle investments
-2
-1
Opening a nuclear power plant
-3
Cadmium mining
+1
+1
Electrical vehicle investments
-3.5

$$H = \sum_{i,j} J_{ij} S_i S_j + \sum_i h_i S_i$$

$$H' = \sum_{i,j \in \mathcal{G}} J_{ij} S_i S_j + \sum_{i \in \mathcal{G}} \left(h_i + \sum_{j \notin \mathcal{G}} J_{ij} S_j\right) \cdot S_i$$

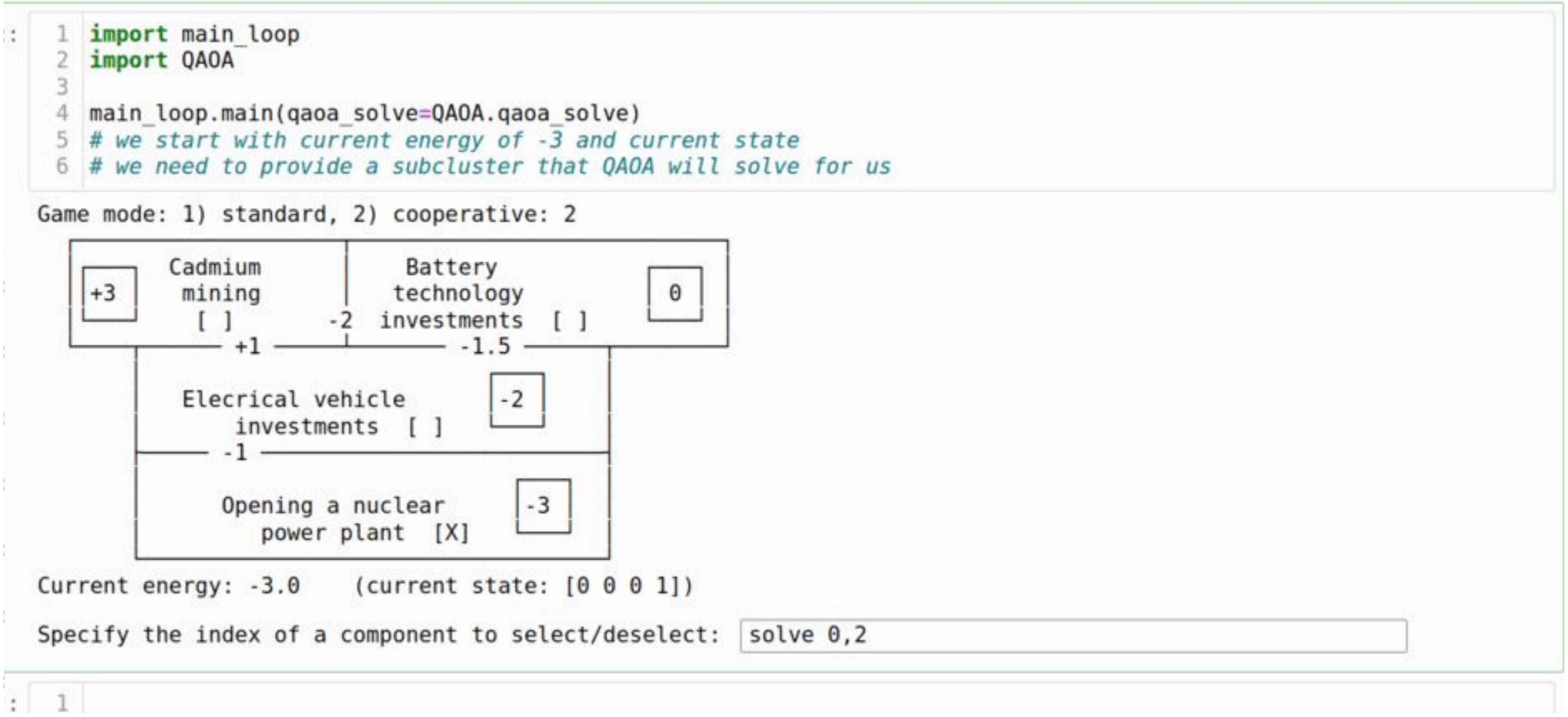


```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
```



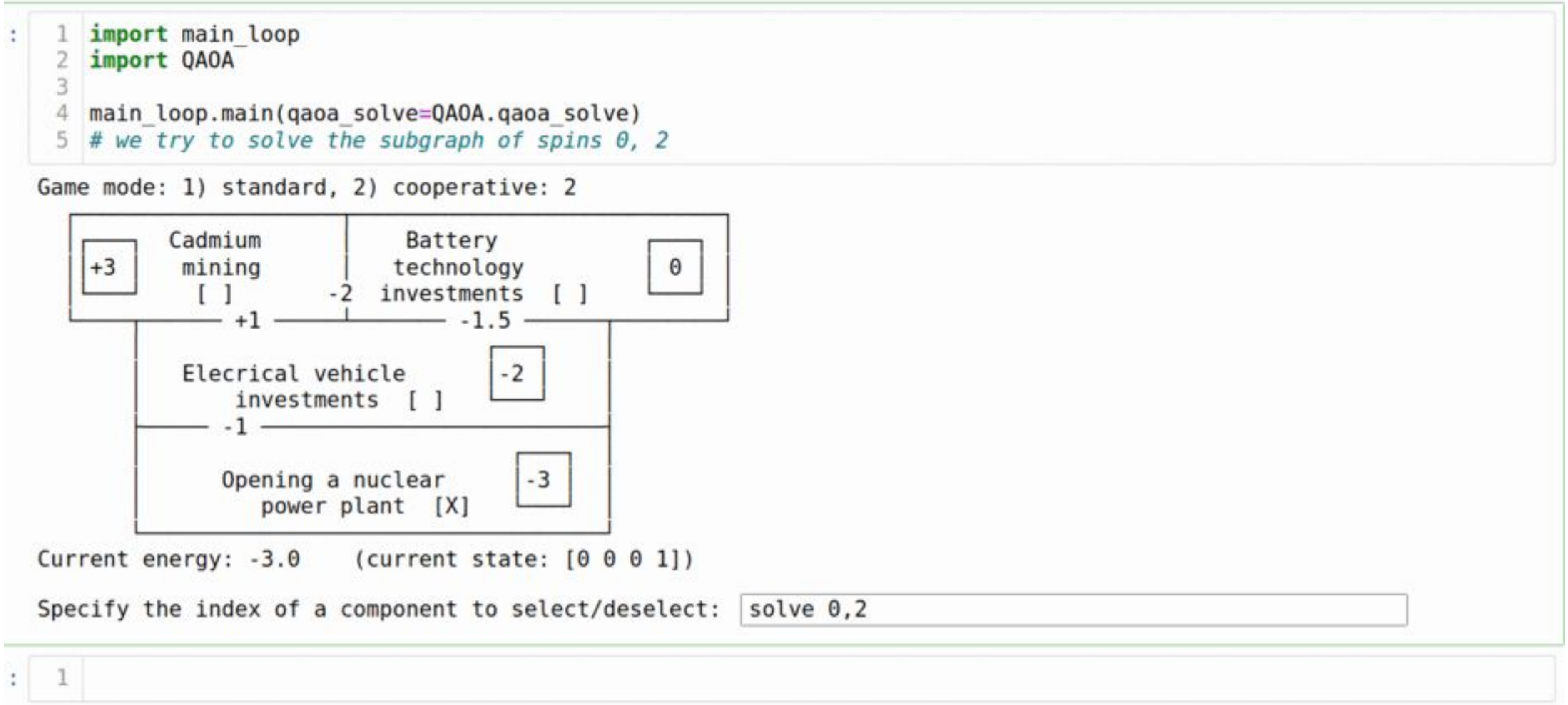
```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# select "cooperative" regime in which QAOA helps player find the best solution
# namely, user specifies subgraph as set of indices and QAOA applies its best solution on this subgraph
```

```
Game mode: 1) standard, 2) cooperative: 2
```



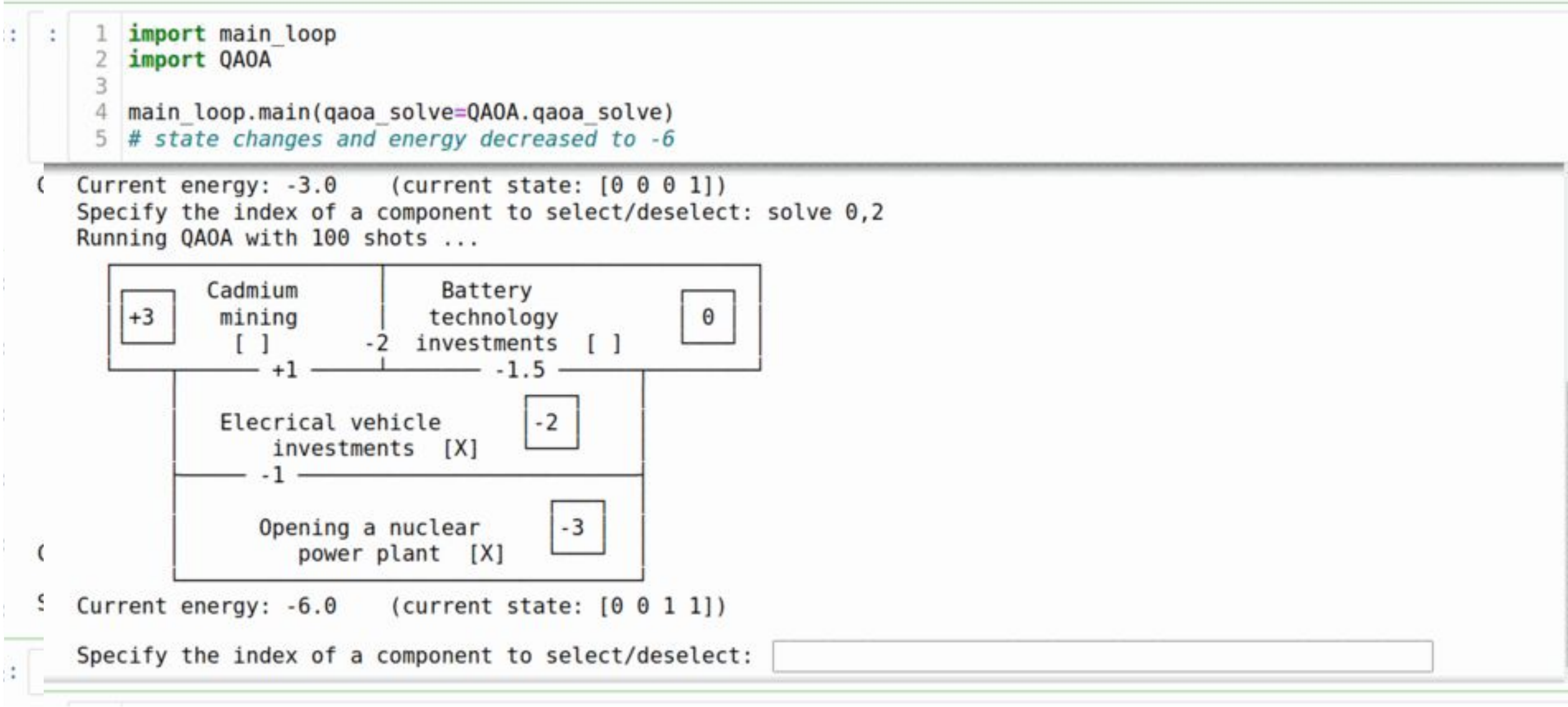
```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# we start with current energy of -3 and current state
# we need to provide a subcluster that QAOA will solve for us
```

```
Game mode: 1) standard, 2) cooperative: 2
 ┌────────────────────┬──────────────────────────┐
 │ ┌───┐  Cadmium     │    Battery          ┌───┐ │
 │ │+3 │   mining     │   technology        │ 0 │ │
 │ └───┘    [ ]     -2  investments   [ ]  └───┘ │
 └────┬──── +1 ───────┴────── -1.5 ──────┬───────┘
      │                                   │
      │    Elecrical vehicle      ┌───┐  │
      │                           │-2 │  │
      │        investments  [ ]   └───┘  │
      ├──── -1 ───────────────────────────┤
      │                                   │
      │    Opening a nuclear       ┌───┐ │
      │                            │-3 │ │
      │       power plant  [X]     └───┘ │
      └───────────────────────────────────┘
Current energy: -3.0    (current state: [0 0 0 1])

Specify the index of a component to select/deselect: solve 0,2
```



```python
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# we try to solve the subgraph of spins 0, 2
```

```
Game mode: 1) standard, 2) cooperative: 2
 ┌─────────────────────┬─────────────────────────────────┐
 │ ┌──┐   Cadmium      │   Battery                 ┌─┐   │
 │ │+3│   mining       │  technology               │0│   │
 │ └──┘    [ ]       -2  investments  [ ]          └─┘   │
 └────┬──── +1 ────────┴──────── -1.5 ─────┬─────────────┘
      │                           ┌──┐     │
      │  Elecrical vehicle        │-2│     │
      │     investments  [ ]      └──┘     │
      ├──── -1 ────────────────────────────┤
      │                             ┌──┐   │
      │    Opening a nuclear        │-3│   │
      │      power plant  [X]       └──┘   │
      └────────────────────────────────────┘
Current energy: -3.0    (current state: [0 0 0 1])

Specify the index of a component to select/deselect: solve 0,2
```

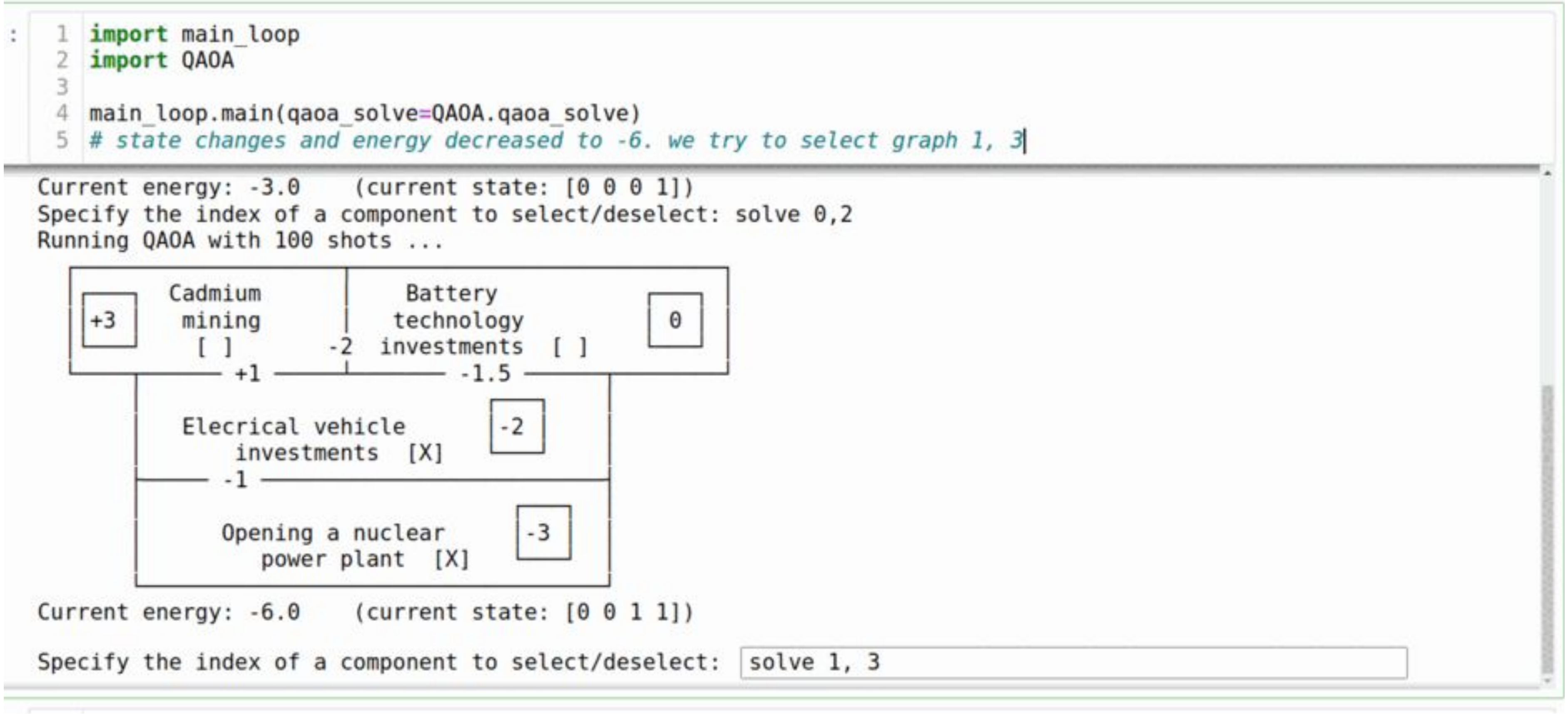


```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# state changes and energy decreased to -6
```

```
Current energy: -3.0    (current state: [0 0 0 1])
Specify the index of a component to select/deselect: solve 0,2
Running QAOA with 100 shots ...
```

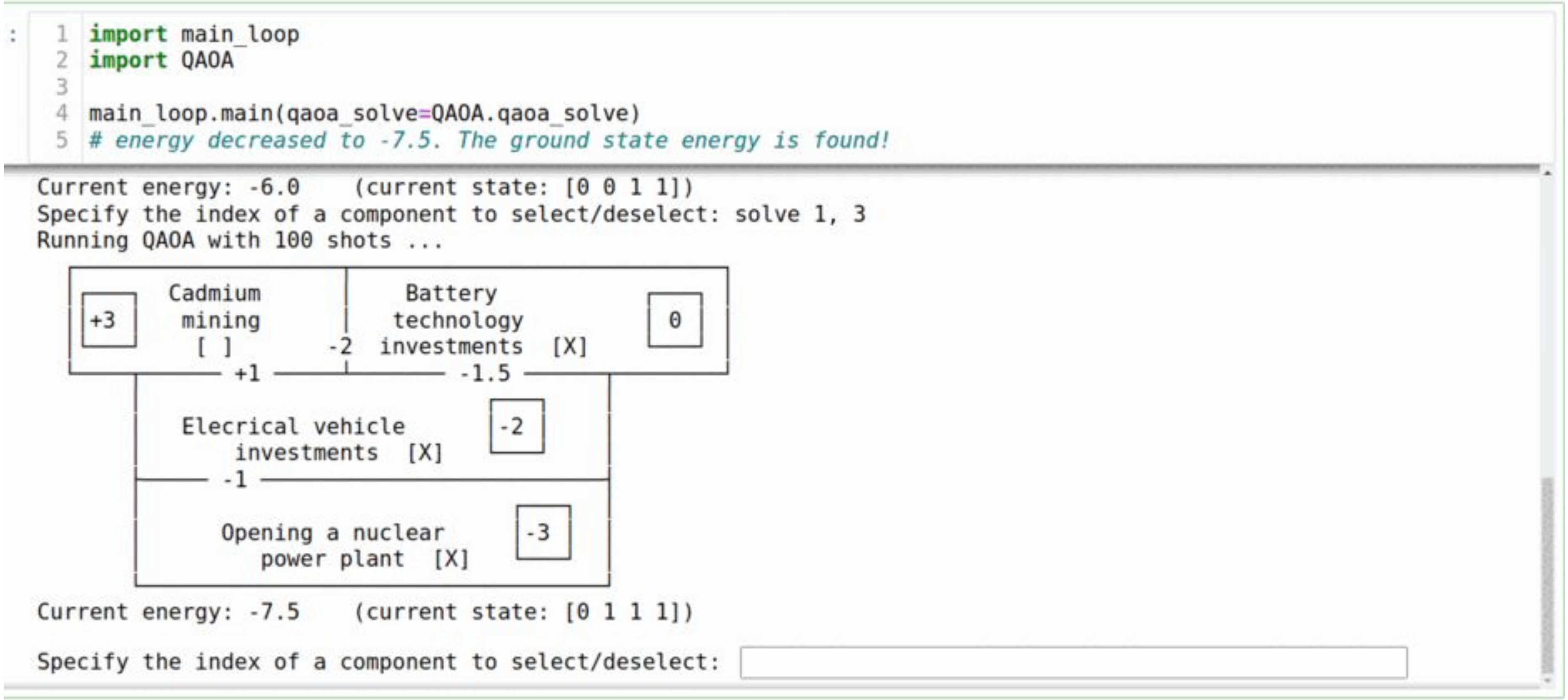
```
+-------------------------+-------------------------------+
| +3    Cadmium           |      Battery              0   |
|        mining           |     technology                |
|         [ ]          -2 |  investments  [ ]             |
+-------- +1 -------------+------------- -1.5 --+---------+
    |                                           |
    |   Elecrical vehicle           -2          |
    |       investments   [X]                   |
    |------- -1 --------------------------------|
    |                                           |
    |      Opening a nuclear          -3        |
    |         power plant   [X]                 |
    +-------------------------------------------+
```

```
Current energy: -6.0    (current state: [0 0 1 1])

Specify the index of a component to select/deselect:
```



```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# state changes and energy decreased to -6. we try to select graph 1, 3
```

```
Current energy: -3.0    (current state: [0 0 0 1])
Specify the index of a component to select/deselect: solve 0,2
Running QAOA with 100 shots ...
  ┌─────────────────────┬──────────────────────────┐
  │┌───┐    Cadmium     │      Battery        ┌───┐│
  ││+3 │     mining     │    technology       │ 0 ││
  │└───┘      [ ]      -2  investments  [ ]  └───┘│
  └────┬─────── +1 ─────┴─────── -1.5 ──────┬──────┘
       │                                    │
       │    Elecrical vehicle       ┌───┐   │
       │        investments  [X]    │-2 │   │
       │                            └───┘   │
       ├───── -1 ───────────────────────────┤
       │                                    │
       │       Opening a nuclear      ┌───┐ │
       │          power plant  [X]    │-3 │ │
       │                              └───┘ │
       └────────────────────────────────────┘
Current energy: -6.0    (current state: [0 0 1 1])

Specify the index of a component to select/deselect: solve 1, 3
```



```
import main_loop
import QAOA

main_loop.main(qaoa_solve=QAOA.qaoa_solve)
# energy decreased to -7.5. The ground state energy is found!
```

```
Current energy: -6.0    (current state: [0 0 1 1])
Specify the index of a component to select/deselect: solve 1, 3
Running QAOA with 100 shots ...
 ┌────────────────────┬─────────────────────────────┐
 │ ┌──┐   Cadmium     │     Battery          ┌──┐   │
 │ │+3│    mining     │   technology         │ 0│   │
 │ └──┘     [ ]      -2  investments   [X]   └──┘   │
 └───┬──────── +1 ────┴──────── -1.5 ─────┬─────────┘
     │                             ┌──┐   │
     │  Elecrical vehicle          │-2│   │
     │      investments   [X]      └──┘   │
     ├─────── -1 ─────────────────────────┤
     │                               ┌──┐ │
     │     Opening a nuclear         │-3│ │
     │        power plant   [X]      └──┘ │
     └────────────────────────────────────┘
Current energy: -7.5    (current state: [0 1 1 1])

Specify the index of a component to select/deselect:
```

Outlook

Outlook

- Hamiltonian specified by a domain expert (scenario as close as possible to reality);

Outlook

- Hamiltonian specified by a domain expert
  (scenario as close as possible to reality);

- leaderboard to compete against each other in cooperative mode
  (context for innovative solution to real word resource allocation problems);

Outlook

- Hamiltonian specified by a domain expert
  (scenario as close as possible to reality);

- leaderboard to compete against each other in cooperative mode
  (context for innovative solution to real word resource allocation problems);

- … technicalities like a proper GUI;

Outlook

- Hamiltonian specified by a domain expert
  (scenario as close as possible to reality);

- leaderboard to compete against each other in cooperative mode
  (context for innovative solution to real word resource allocation problems);

- … technicalities like a proper GUI;

- real hardware is currently too slow for a real-time Opti-maze.