Initial Population Prompt

System Prompt:

You are an expert in the domain of optimization heuristics. Your task is to design heuristics that can effectively solve optimization problems.

Your response outputs Python code and nothing else. Format your code as a Python code string: "```python ... ```".

User Prompt:

Write a crossover function for a crossover function for solving the port selection optimization problem using a genetic algorithm. The problem requires selecting a subset of ports from a given set of ports to serve users, with the goal of minimizing the transmission power. The encoding uses binary encoding, where the number of binary bits equals the number of ports, and a bit value of 1 represents the selection of that port. The solution generated after the crossover must be valid.

The `crossover` function takes as input a 2D NumPy array parents and an integer n_pop. The function performs a genetic crossover operation on parents to generate n_pop offspring. Use vectorized implementation if possible.

def crossover_v1(parents: np.ndarray, n_pop: int) -> np.ndarray:

n_parents, n_decap = parents.shape

Split genomes into two halves

left_halves = parents[:, :n_decap // 2]

right_halves = parents[:, n_decap // 2:]

Create parent pairs

```
parents_idx = np.stack([np.random.choice(range(n_parents), 2, replace=False)
for _ in range(n_pop)])
```

```
parents_left = left_halves[parents_idx[:, 0]]
```

parents_right = right_halves[parents_idx[:, 1]]

Create offspring

```
offspring = np.concatenate([parents_left, parents_right], axis=1)
```

return offspring

Refer to the format of a trivial design above. Be very creative and give

`crossover_v2`. Output code only and enclose your code with Python code block: ```python ... ```

Short-term Reflection Prompt

System Prompt:

You are an expert in the domain of optimization heuristics. Your task is to give hints to design better heuristics.

User Prompt:

Below are two crossover functions for write a crossover function for solving the port selection optimization problem using a genetic algorithm. The problem requires selecting a subset of ports from a given set of ports to serve users, with the goal of minimizing the transmission power. The encoding uses binary encoding, where the number of binary bits equals the number of ports, and a bit value of 1 represents the selection of that port. The solution generated after the crossover must be valid.

The `crossover` function takes as input a 2D NumPy array parents and an integer n_pop. The function performs a genetic crossover operation on parents to generate n_pop offspring. Use vectorized implementation if possible.

You are provided with two code versions below, where the second version performs better than the first one.

[Worse code]

[Better code]

You respond with some hints for designing better heuristics, based on the two code versions and using less than 20 words.

The comment: "[Worse code]" and "[Better code]" are followed by the crossover operator individuals with poor performance and the crossover operator individuals with good performance, respectively.

Crossover Prompt

System Prompt:

You are an expert in the domain of optimization heuristics. Your task is to design heuristics that can effectively solve optimization problems.

Your response outputs Python code and nothing else. Format your code as a Python code string: "```python ... ```".

User Prompt:

Write a crossover function for a crossover function for solving the port selection optimization problem using a genetic algorithm. The problem requires selecting a subset of ports from a given set of ports to serve users, with the goal of minimizing the transmission power. The encoding uses binary encoding, where the number of binary bits equals the number of ports, and a bit value of 1 represents the selection of that port. The solution generated after the crossover must be valid.

The `crossover` function takes as input a 2D NumPy array parents and an integer n_pop. The function performs a genetic crossover operation on parents to generate n_pop offspring. Use vectorized implementation if possible.

[Worse code]

[Better code]

[Reflection]

Please write an improved function `crossover_v2`, according to the reflection. Output code only and enclose your code with Python code block: ```python ... ```.

The comment: "[Worse code]" and "[Better code]" are followed by the crossover operator individuals with poor performance and the crossover operator individuals with good performance, respectively. "[Reflection]" refers to the improvement suggestions generated through short-term reflection.

Long-term Reflection Prompt

System Prompt:

You are an expert in the domain of optimization heuristics. Your task is to give hints to design better heuristics.

User Prompt:

Below is your prior long-term reflection on designing heuristics for write a crossover function for solving the port selection optimization problem using a genetic algorithm. The problem requires selecting a subset of ports from a given set of ports to serve users, with the goal of minimizing the transmission power. The encoding uses binary encoding, where the number of binary bits equals the number of ports, and a bit value of 1 represents the selection of that port. The solution generated after the crossover must be valid.

[prior_reflection]

Here are some recent insights.

[new_reflection]

Write constructive hints for designing better heuristics, based on prior reflections and new insights and using less than 50 words.

The comment: "[prior_reflection]" and "[new_reflection]" are followed by the historical reflection results and the latest reflection results, respectively.

Mutation Prompt

System Prompt:

You are an expert in the domain of optimization heuristics. Your task is to design heuristics that can effectively solve optimization problems.

Your response outputs Python code and nothing else. Format your code as a Python code string: "```python ... ```".

User Prompt:

Write a crossover function for Write a crossover function for solving the port selection

optimization problem using a genetic algorithm. The problem requires selecting a subset of ports from a given set of ports to serve users, with the goal of minimizing the transmission power. The encoding uses binary encoding, where the number of binary bits equals the number of ports, and a bit value of 1 represents the selection of that port. The solution generated after the crossover must be valid.

The `crossover` function takes as input a 2D NumPy array parents and an integer n_pop. The function performs a genetic crossover operation on parents to generate n_pop offspring. Use vectorized implementation if possible.

[Prior reflection]

[Code]

Please write a mutated function `crossover_v2`, according to the reflection. Output code only and enclose your code with Python code block: ```python ... ```.

The comment:"[Prior reflection]" and "[Code]" are followed by the previous reflection results and the crossover operator code block to be mutated, respectively.