System Prompt:

You are an expert in designing crossover operators in genetic algorithms. Your task is to design efficient and effective crossover operators tailored for solving complex optimization problems.

Your response must output only Python code, formatted within a Python code block using: "```python ... ```".

User Prompt:

Write a crossover function to solve the port selection optimization problem using a genetic algorithm. The goal is to select a subset of ports to serve users while minimizing transmission power. Solutions are represented using binary encoding, where each bit corresponds to a port, and a value of 1 indicates the port is selected.

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 ... ```.