VCG Auction Experiment

1. Problem Overview

The given problem is a combinatorial auction where advertising spaces on lecture slides are being auctioned off. The professor offers two types of ad spaces: a top banner and a sidebar. There are different types of bidders, some of whom are interested only in one space, while others require both. The auction uses the Vickrey-Clarke-Groves (VCG) mechanism for allocation and pricing, which incentivizes truthful bidding. Additionally, a Second Price Auction (SPA) mechanism is implemented for comparison.

The goal is to determine:

- The allocation of the ad spaces to bidders using the VCG auction.
- The payments and total revenue for the auctioneer in VCG and SPA scenarios.
- The evaluation of VCG truthfulness and revenue properties by introducing experiments, including agents with multiple identities.

2. VCG Auction (Part A)

2.1 Code Explanation

- The class VCGAuction is responsible for managing bids and determining the allocation of ad spaces.
- The find_optimal_allocation method identifies the allocation that maximizes the overall value (social welfare). It compares two cases:
    1. Allocating both spaces to a single bidder interested in both spaces.
    2. Allocating the top banner and sidebar to two separate bidders.

The allocation yielding the highest value is chosen.

- The compute_vcg_payments method calculates the payments for the winners. For each winning bidder:
    1. The auction is run again without this bidder to determine the social welfare without them.
    2. The difference in social welfare with and without the bidder determines their VCG payment.

2.2 Input Example

The input JSON file (example_input.json) contains the bids from different agents:
json

```
{
  "bids": [
    {"bidder_id": "A1", "value": 100, "space_type": "b"},
```

    {"bidder_id": "B1", "value": 60, "space_type": "t"},
    {"bidder_id": "C1", "value": 50, "space_type": "s"},
    {"bidder_id": "D1", "value": 80, "space_type": "b"}
  ]
}

- A1 and D1 bid for both spaces (b), while B1 bids for the top banner (t) and C1 bids for the sidebar (s).

2.3 VCG Auction Results

When the VCG auction is executed, the following allocation is found:
- B1 is allocated the top banner.
- C1 is allocated the sidebar.

The VCG payments are calculated based on the value difference when each bidder is removed:
- Bidder B1 pays 50.
- Bidder C1 pays 40.

The total revenue from the VCG auction is 90.

```
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2> python .\vcg_auction.py
Experiment Results:
{
  "vcg_allocation": {
    "top": "B1",
    "sidebar": "C1"
  },
  "vcg_payments": {
    "B1": 50,
    "C1": 40
  },
  "vcg_revenue": 90,
  "spa_winner": "A1",
  "spa_payment": 80
}
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2>
```

2.4 Experiment Results

The following results were obtained:
json
{

"vcg_allocation": {
  "top": "B1",
  "sidebar": "C1"
},
"vcg_payments": {
  "B1": 50,
  "C1": 40
},
"vcg_revenue": 90
}

```
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2> python .\action_test_suite.py

Key Findings:

1. Sybil Attack Analysis:
Original VCG Revenue: 50
Sybil Attack VCG Revenue: 110

2. Revenue Comparison:
Minimum SPA/VCG Revenue Ratio: 0.58
Average SPA/VCG Revenue Ratio: 1.18

3. Truthful Bidding Analysis:
True Value: 150
Truthful Bid Payment: 100
Overbid Payment: 100
Underbid Payment: 100
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2>
```

3. Second Price Auction (Part C)

3.1 Code Explanation

The SecondPriceAuction class simulates an auction where the highest bidder wins both ad spaces but pays the amount of the second-highest bid.

- The run_auction method sorts the bids and allocates both spaces to the highest bidder. The payment is equal to the second-highest bid.

3.2 SPA Results

In the given experiment:
- Bidder A1 wins the auction (highest bid of 100 for both spaces).
- A1 pays the second-highest bid of 80.

The output of the second price auction is:
json

{
  "spa_winner": "A1",
  "spa_payment": 80
}

4. Analysis (Part B and Part C Experiments)

4.1 Non-Dominant Strategy Truthfulness

By experimenting with a bidder submitting multiple identities, we can test whether VCG is dominant-strategy truthful. In scenarios where an agent creates a second identity and bids using both identities, they might achieve a higher utility than when bidding truthfully with only one identity. This demonstrates that the VCG mechanism may not be dominant-strategy truthful when such identity manipulation is possible.

4.2 Comparison of Revenue

The second-price auction consistently generates at least half the revenue of the VCG auction when bidders submit truthful reports. In the current experiment:

- VCG revenue = 90
- SPA revenue = 80

5. Conclusion

- The VCG auction provides a truthful mechanism (without identity manipulation) and optimizes social welfare.
- The second-price auction, though simpler, yields comparable revenue and maintains a dominant strategy of truthful bidding.

```
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2> python .\auction_analysis.py

Key Findings:

1. Sybil Attack Analysis:
Original VCG Revenue: 50
Sybil Attack VCG Revenue: 110

2. Revenue Comparison:
Minimum SPA/VCG Revenue Ratio: 0.58
Average SPA/VCG Revenue Ratio: 1.18

3. Truthful Bidding Analysis:
True Value: 150
Truthful Bid Utility: 100
Overbid Utility: 100
Underbid Utility: 100
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2>
```

Summary of the key findings from output:

1. Sybil Attack Analysis:
    - Original VCG Revenue: 50
    - Sybil Attack VCG Revenue: 110
    - This result suggests that the Sybil attack significantly increased the revenue for the auctioneer. In the original scenario, the VCG revenue was only 50, but after splitting the identity of a bidder, the revenue more than doubled. This demonstrates how vulnerable the VCG mechanism can be to such attacks.

2. Revenue Comparison:
    - Minimum SPA/VCG Revenue Ratio: 0.58
    - In one of the test cases, the second-price auction generated only 58% of the revenue compared to the VCG auction. This shows that, in certain scenarios, the second-price auction may result in much lower revenue.
    - Average SPA/VCG Revenue Ratio: 1.18
    - On average, the second-price auction generated about 18% more revenue than VCG across the test cases. This indicates that while VCG typically optimizes for efficiency, the second-price auction may sometimes produce better monetary outcomes.

3. Truthful Bidding Analysis:
    - True Value: 150
    - Truthful Bid Utility: 100
    - Overbid Utility: 100
    - Underbid Utility: 100
    - In this case, it seems that bidding truthfully, overbidding, or underbidding resulted in the same utility (payment of 100). This may indicate that in this specific test scenario, the second-price auction was resilient to deviations in bidding strategy, reinforcing the idea that truthful bidding is a dominant strategy in such auctions.

Insightful results for the Sybil attack, revenue comparison, and truthful bidding scenarios.

1. Sybil Attack Analysis:
   - Original VCG Revenue: 50
   - Sybil Attack VCG Revenue: 11
   - In this case, introducing a Sybil attack (where a single entity splits into two identities) increased the VCG revenue, showcasing a vulnerability where bidders can manipulate the system by splitting bids.

2. Revenue Comparison:
   - Minimum SPA/VCG Revenue Ratio: 0.58
   - The lowest second-price auction revenue was 58% of the VCG revenue, indicating that sometimes the second-price auction generates significantly less revenue.
   - Average SPA/VCG Revenue Ratio: 1.18
   - On average, second-price auctions yielded slightly more revenue than VCG auctions, suggesting they can sometimes outperform VCG, depending on the bidding scenarios.

3. Truthful Bidding Analysis:
   - True Value: 150
   - Truthful Bid Payment: 100
   - Overbid Payment: 100
   - Underbid Payment: 100
   - In second-price auctions, whether the bidder truthfully bids their value, overbids, or underbids, the payment remains the same. This confirms the property of second-price auctions where bidders are incentivized to bid truthfully because the payment does not change based on their bid.

```
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2> python .\experiments.py

Sybil Attack Experiment Results:
{
  "truthful": {
    "allocation": {
      "top": "B1",
      "sidebar": "C1"
    },
    "payments": {
      "B1": 30,
      "C1": 20
    }
  },
  "sybil": {
    "allocation": {
      "top": "B1",
      "sidebar": "C1"
    },
    "payments": {
      "B1": 60,
      "C1": 50
    }
  }
```

```
Revenue Comparison Experiment Results:
[
  {
    "vcg_revenue": 80,
    "spa_revenue": 80,
    "revenue_ratio": 1.0
  },
  {
    "vcg_revenue": 125,
    "spa_revenue": 140,
    "revenue_ratio": 1.12
  },
  {
    "vcg_revenue": 90,
    "spa_revenue": 70,
    "revenue_ratio": 0.7777777777777778
  }
]
PS C:\Users\sharm\Desktop\B21CS095_CSL7650_A2>
```

The test suite worked well in illustrating the core concepts, including potential weaknesses in VCG auctions due to strategic behavior like Sybil attacks.