

ASSIGNMENT-3

Annam Deddeepya Purnima (cs22btech11006)
Mane Pooja Vinod (cs22btech11035)
Surbhi (cs22btech11057)

Question:

Assumption

1. Rank of A is n

Implement the simplex algorithm to maximize the objective function, You need to implement the method discussed in class.

Input: CSV file with m+2 rows and n+1 column.

- The first row excluding the last element is the initial feasible point z of length n
- The second row excluding the last element is the cost vector c of length n
- The last column excluding the top two elements is the constraint vector b of length m
- Rows third to m+2 and column one to n is the matrix A of size $m \times n$

Output: You need to print the sequence of vertices visited and the value of the objective function at that vertex

REPORT

Code implements the **Simplex algorithm** for solving linear programming problems. It takes a set of linear constraints and a cost function to minimize, adjusts an initial point to a feasible solution if necessary, includes features for handling degeneracy and avoiding numerical instability

→ Parsing the input file to extract:

- Initial point (z)
- Cost vector (c)
- Constraints matrix (A) and vector (v)

→ The **is_feasible_point** function validates if the initial point satisfies the inequality constraints $Ax \leq v$

- The **adjust_to_basic_feasible** function moves the initial point to a feasible region by finding active constraints ($\mathbf{Ax} = \mathbf{v}$) and adjusting the point along the null space of active constraints.
- **handle_degeneracy** functions handles scenarios where the algorithm encounters degenerate vertices by identifying valid entering directions and step sizes. It will print warnings when no valid directions or steps are available.
- The **simplex** function iteratively explores feasible solutions. It identifies active constraints, computes search directions and step sizes, updates the solution (z), and tracks visited vertices with costs. The process stops when no valid direction exists (optimal solution) or after a maximum iteration limit.
- Uses pseudo-inverse (**np.linalg.pinv**) for handling rank-deficient matrices.
- We took **input as t.csv at line 127**, we can change according to our test cases.

OUTPUT:

```
(base) surbhi@surbhi-SS:~/Desktop/L0$ python3 3.py
Initial point: [0. 0.], Initial cost: 0.0
Feasible point: [-2.  1.], Feasible cost: -8.0

Sequence of vertices visited:
Step 1: Vertex = [-2.  1.], Cost = -8.0

Final Optimal Point: [-2.  1.]
Final Optimal Cost: -8.0
```