1) In the Simplex approach, show the following: (here recall that ˆcj = cj − P i∈B cidij).

(a) Let y be a solution to the constraints Ay = b, not necessarily a basic feasible solution. Let ˆc(B) be the relative cost vector defined w.r.t. a basis B. Show that ˆc T y = g − f where f is the cost of the current solution associated with B and g is the cost of y.

We start by recalling that, in the simplex method, beginning from a basic feasible solution represented by a basis B, we solve the linear system Ax = b to obtain the basic variables x_B in terms of the non-basic variables x_N, and then compute the reduced cost vector ˆc_N = c_N - c_B * B^-1 * A_N, where c_B are the cost coefficients of the basic variables, and A_N is the matrix containing the columns of A corresponding to the non-basic variables.

Now, let y be a solution to the constraints A = b, not necessarily a basic feasible solution, and let B be a basis corresponding to a basic feasible solution. Then we can write y = x_B + B^-1 * A_N * x_N, where x_N are the values of the non-basic variables.

The cost of y is given by g = c * y = c_B * x_B + c_N * x_N. The cost of the current basic feasible solution associated with B is given by f = c_B * x_B.

The reduced cost vector with respect to the basis B is ˆc(B) = c_N - c_B * B^-1 * A_N. Then we have:

ˆc(B) * y_N = (c_N - c_B * B^-1 * A_N) * x_N
= c_N * x_N - c_B * B^-1 * A_N * x_N
= c_N * x_N - ˆc(B) * B * x_B

The last equality follows from the fact that B * x_B = A_N * x_N - b, since Ay = b.

Therefore, we can rewrite the cost of y as:

g = c_B * x_B + c_N * x_N
= f + (c_N * x_N - ˆc(B) * B * x_B)
= f + ˆc(B) * (y_N - x_B)

where y_N - x_B is the change in the nonbasic variables from the current basic feasible solution to y.

Therefore, we have shown that ˆc(B) * (y_N - x_B) is the difference between the cost of y and the cost of the current basic feasible solution associated with B. Thus, ˆc_T y = g - f, as desired.

(b) Suppose the Simplex is at basis B and selects a basis B' by choosing a column j s.t. ˆcj < 0. Is the cost of the solution guaranteed to decrease? Under which condition would it not.

If the Simplex algorithm selects a basis B' by choosing a column j such that $\hat{c}_j < 0$, then the algorithm proceeds by selecting a row k such that $A_{kj} > 0$ and the quotient $x_{B(k)}/A_{kj}$ is minimized among all such k. The variable corresponding to the basic outgoing variable $x_{B(k)}$ is replaced by the non-basic variable x_j, and the new basis becomes $B' = (B - \{k\}) \cup \{j\}$.

Under certain conditions, the solution cost is guaranteed to decrease. Specifically, if the selected row k, corresponds to a positive change in the objective function, i.e., the reduced cost of variable j is negative and the ratio $x_{B(k)}/A_{kj}$ is positive, then the solution cost is guaranteed to decrease. This is because the objective function is a linear combination of the basic variables with coefficients given by the cost coefficients of the basic variables, and the pivot operation decreases the value of the outgoing basic variable and increases the value of the incoming non-basic variable, thus decreasing the objective function.

However, under some conditions, the solution cost may not decrease. For example, if the reduced cost of variable j is negative, but all values of A_{kj} are non-positive, then the pivot operation cannot be performed and the algorithm terminates without changing the solution. Another example is when the ratio $x_{B(k)}/A_{kj}$ is non-positive, which implies that increasing the value of the variable x_j will cause a decrease in the value of the incoming variable $x_{B(k)}$ and thus the solution cost may not decrease. In such cases, the algorithm may terminate without finding an optimal solution, or it may enter an infinite loop if the problem is unbounded.

2) Investigate the following and report your findings:

(a) How does the Simplex method find the first Basic Feasible solution

By using the simplex method, linear programming problems can be solved. One of the most important steps of the algorithm is finding the first basic feasible solution, which is a feasible solution that can be used as a starting point for the iterative optimization process. The simplex approach is based on choosing a basic feasible solution but not-optimal one, in order to start an iterative process which will end up with a feasible, and also optimal solution.

To find the first basic feasible solution, the simplex method uses a technique called "phase I". The objective of phase I is to introduce some artificial variables, and then solve a modified version of the original problem, so that:

The modified problem has a feasible solution if, and only if the original problem has a feasible solution. Therefore, if there is not feasible solution for the original problem, the introduced artificial variables will not allow the modified system to be feasible.

The artificial variables in the modified problem will be chosen in order to result in an optimal solution with the artificial variables equal to zero, which corresponds to a basic feasible solution of the original problem.

The modified problem is obtained by adding a new constraint for each variable that has a lower bound (for example, a minimum value it can take). The new constraint has the form "variable >= lower bound", and introduces a non-negative dummy (‘slack variable’) variable to represent the difference between the variable and its lower bound. For example, if the original problem is

Maximize:
2x + 3y

Subject to:
x >= 0
y >= 0
x + y <= 4
2x + y >= 3

The modified problem for phase I is

Maximize:
0a + 0b + 0c + 0d + Mw + Nx + Py

Subject to:
x - w + a = 0
y - x + b = 0
4 - y + c = 0
3 - 2x - y + d = 0
w, x, y >= 0
a, b, c, d >= 0

In the modified problem, the dummy variables (a, b, c, d) represent the slack variables of the lower bound constraints, and the variable w represents the maximum violation of the lower bound constraints. The constant M is a large positive number that helps the artificial variables to have a higher cost than the original variables, so that they are minimized to zero in the optimal solution.

Once the modified problem is constructed, the simplex method is applied to find an optimal solution which has null values for the artificial variables. If such a solution is found, it corresponds to a basic feasible solution of the original problem. Otherwise, the original problem is infeasible.

After phase I, the artificial variables and the w variable are eliminated from the problem, and the simplex method is applied to the original problem to find the optimal solution. Since the starting point is a basic feasible solution, the algorithm is guaranteed to converge to an optimal solution, bearing in mind that the problem is not unbounded.

(b) Can the Simplex Method cycle, i.e. repeat a basis? under what conditions.

Yes, the simplex method has the ability to cycle, which means that it can repeat a base and enter an infinite loop, which would not lead to finding the best solution. This may occur under the next circumstances:

1. ***There are alternative ideal solutions to the problem:*** In this specific case, the simplex method can cycle between multiple optimal solutions that share the same objective function value. The method always seeks to minimize an objective function, therefore, if there exists more than one solution with the same optimized minimal value, the infinite loop could happen. This usually happens when there are multiple optimal solutions at the vertices of the feasible region, or when the objective function is parallel to one of the constraints. This is not a huge problem due to the fact that any of the optimal solutions of the loop would be valid. So once this situation is detected, if it is due to several optimal solutions sharing a common objective function value, any of the solutions would be fine.

2. ***The problem has degeneracy:*** In this case, one or more of the basic variables have a value of zero in the optimal solution, which causes the simplex method to have multiple choices for the input variable. If the method chooses the same input variable multiple times, it can enter an infinite loop. Degeneracy can occur when the feasible region has redundant constraints, or when the problem has multiple optimal solutions that share a common vertex.

3. ***The method makes rounding errors:*** In some cases, the simplex method may encounter rounding errors that cause it to choose the wrong input or output variable, or to terminate too early. This can occur when the problem has coefficients of large magnitude, or when the floating point arithmetic used by the computer is not sufficiently accurate.

To overcome cycling, several techniques can be used, such as slightly perturbing the problem to break the degeneracy, using a different pivot rule to avoid cycling, or implementing a cycle detection mechanism that monitors the number of iterations and the bases encountered.

(c) What is the complexity of the Simplex method.

<u>Time complexity:</u>

The simplex method is a well-known algorithm for solving linear programming problems. Its worst-case time complexity is exponential in the number of variables, although in practice it often runs much faster. But if a conservative approach is needed for the worst case scenario, the complexity would be exponential.

Specifically, the simplex method has a worst-case exponential time complexity in the number of variables, which means that as the number of variables increases, the time required to solve the problem grows exponentially. However, in practice, the simplex method is usually much faster than this worst-case limit, more or less polynomial-time complexity for real-life problems. Especially, when the problem is well solved (i.e., has a small number of constraints or a well-solved geometry).

In addition, there are many variants of the simplex method designed to improve its performance, such as the dual simplex method and the revised simplex method.

The dual simplex method has the same worst-case exponential time complexity as the simplex method in the number of variables. However, in practice it is usually faster than the simplex method.
The revised simplex method has a worst-case time complexity that is polynomial in the number of variables, and is often faster in practice than the simplex method and the dual simplex method.

Space complexity:

The spatial complexity of the simplex method depends on the implementation details and the specific problem being solved. However, in general, the spatial complexity of the simplex method is proportional to the number of variables and the number of constraints in the problem.

In addition, some variants of the simplex method, such as the revised simplex method, require additional storage for constraint matrix factorizations or other auxiliary data structures, in exchange for the slighter time complexity of these methods. Hence, here the trade-off between ***space vs time*** complexity can be seen.

In general, the space complexity of the simplex method is usually much less than its time complexity, and the space required is usually manageable for problems of practical size.

3) In the system Ax = b, x ≥ 0 solved via Simplex, can two different basis in A result in the same vertex. If so, give an example. If not, prove your statement.

No, two different bases in A cannot give room to resulting in the same vertex when solving the Simplex algorithm with Ax=b, x ≥ 0.

Aiming to prove this statement, we can consider the fact that each vertex in the feasible region is related to a single basis feasible solution. This one, is a solution that satisfies the constraints, and has exactly n linearly independent active constraints, where n is the dimension of the variable vector x.

In the Simplex algorithm, each iteration moves from one vertex to an adjacent vertex along an edge of the feasible region. At each iteration, a new basic feasible solution is found by removing a variable from the base and adding another variable to the base. This new basis corresponds to a new set of n linearly independent active constraints, which defines a unique vertex.

Therefore, if two different bases in A result in the same vertex, then they must define the same set of n linearly independent active constraints, which is a contradiction. Therefore, two different bases in A cannot result in the same vertex in the Simplex algorithm.

This can also be illustrated by a simple example:

Consider the system

x1 + x2 = 2
x1 - x2 = 1

The feasible region is a line segment between (1.5, 0.5) and (0.5, 1.5). The two bases corresponding to these vertices are {x1, x2} and {x1-x2, x1}, respectively.

We can see that these two bases define different sets of linearly independent active constraints, and therefore cannot give rise to the same vertex in the Simplex algorithm.

4) Can a column vector in A that has just left the basis B when Simplex moves to B' return to the basis at the very next step?

When Simplex moves to the basis B′, a column vector of A cannot, in fact, be brought back to the basis in the next step. This is because, at each step, the Simplex algorithm swaps a variable that is leaving the base for one that is entering the base as it moves from vertex to vertex along an edge of the polytope that is defined by the constraints of the linear programming problem.

Owing to the fact that a variable that has left the base is no longer a base variable, it cannot re-enter it in the subsequent step. The Simplex algorithm picks the variable that makes the greatest improvements to the objective function and only takes into account non-basic variables that can enter the basis. As a result, in the subsequent step, a column vector in A that has just left the basis cannot return to it.

To put it another way, if a variable leaves the basis, it is replaced by a non-basis variable and taken out of the set of basis variables. After that, the Simplex algorithm looks for a non-basic variable that could be added to the basis to make the objective function better. If a variable that has just left the base is chosen to re-enter the base, it would have to take the place of another variable that is not the base, which would not result in the best solution.

In summary, when a vertex leaves the basis in the Simplex method, the coefficient in the objective row of the vertex becomes non-negative (0 or positive). On the next pivot, only vertices with a negative coefficient can be chosen for entering the basis. Hence, the vertex that has just left the basis in the algorithm cannot enter again, on the very next step.

5) A network problem is formulated for a directed graph G = (V, E) using the node-arc incidence matrix which represent the matrix A in the Simplex method. Show that a set of |V | − 1 columns is linearly independent if and only if the corresponding arcs,

considered as undirected edges in the undirected version of G, is a tree. Interpret the pivot step of the simplex method in the light of this fact

Let A be the node-arc incidence matrix of the directed graph G = (V, E) with |V| nodes and |E| arcs. The node-arc incidence matrix is a matrix |V| x |E| where each row represents a node and each column represents an arc. The entry in row i and column j is +1 if arc j starts at node i, -1 if arc j ends at node i, and 0 otherwise.

Consider the undirected version of G, where each arc of E is replaced by an undirected edge. Let T be a subgraph of this undirected version of G, which is a tree with |V|-1 edges. Let E(T) be the set of edges of T.

First, we will show that the corresponding set of columns in A, one for each edge in E(T), is linearly independent.

Suppose, by contradiction, that the corresponding set of columns in A corresponding to E(T) is linearly dependent. This means that there exists a nonzero vector x of length |E(T)| such that Ax = 0, where the entries of x correspond to the edges of E(T).

Let y be a vector of length |V| defined as follows: for each node i of V, yi is equal to the sum of the entries of x corresponding to the edges incident on node i (taking into account the sign of each entry).

Since T is a tree, each node of V has exactly one edge incident on E(T). Therefore, the entries of y corresponding to the edges in E(T) are exactly the entries of x, and hence yTx = 0.

Furthermore, for any edge e in G that is not in T, there exist two nodes i and j such that e is incident to i and j and there exists a single path from i to j in T that does not contain e. Therefore, the corresponding column of A for e is a linear combination of the columns of A corresponding to the edges in E(T). Therefore, Ay = 0.

Since y is a nonzero vector, this implies that the rows of A are linearly dependent, which contradicts the assumption that A is a node-edge incidence matrix. Therefore, the columns of A corresponding to E(T) are linearly independent.

Next, we will show that if the corresponding set of arcs in G is not a tree, then the columns of A are linearly dependent.

If the set of arcs corresponding to E(T) is not a tree, then there exists at least one cycle in the undirected version of G that contains at least one edge in E(T). Let C be a cycle with at least one edge in E(T).

For each edge e in C, there exists a node i such that e is incident to i and there exists a unique path in C from i to any other node in V. Let x be a vector of length |E| defined as follows: for each edge e in C, xe equals +1 if e is directed away from i and -1 otherwise, and xe equals 0 for all other edges.

Note that x is a nonzero vector since C contains at least one edge in E(T). Furthermore, Ax = 0, since the sum of the entries of x corresponding to any node of V is equal to zero. Therefore, the columns of A are linearly dependent.

Thus, we have shown that the corresponding set of columns of A, one for each edge in E(T), is linearly independent if and only if the corresponding arcs, considered as undirected edges in the undirected version of G, form a tree

In summary, explained in a easier way, if B is a set of |V| - 1 arcs in G, represented by linearly independent columns of A, suppose that B is in an orientable cycle. During the cycle in which we label each arc as ***'+1'*** or ***'-1'*** depending on if the arc points upwards or downwards when is traversed, the sum of the ***'+1'*** columns subtracted by the sum of the ***'-1'*** columns must equal 0. Therefore, B's columns would not be linearly independent. A set of |V| - 1 columns would only be linearly independent, if and only if, the corresponding arcs form a tree.

<u>The pivot step:</u>

The pivot step in the simplex method consists of selecting a pivot element and then performing row operations to eliminate all other non-zero entries in the pivot column. The objective of this step is to transform the current basis into a new basis that improves the current solution of the linear programming problem.

In the context of the node-arc incidence matrix, the pivot step can be interpreted as follows: given a current basis of |V|-1 linearly independent columns of the node-arc incidence matrix, the pivot step consists of selecting a pivot element (i.e., a nonzero entry in the column of the current basis corresponding to the incoming variable) and then performing row operations to eliminate all other nonzero entries in the pivot column.

If the incoming variable corresponds to an arc of the directed network G that is not in the current tree base, the pivot step consists of adding the incoming arc to the current tree base and removing another arc that is no longer part of the base. The result is a new base of |V|-1 linearly independent columns corresponding to a new spanning tree of the undirected version of G.

If the incoming variable corresponds to an arc of the directed network G that is already in the base of the current tree, the pivot step consists of updating the base by replacing the outgoing arc (i.e., the arc that is currently part of the base but will be removed) with the incoming arc. The result is a new basis of |V|-1 linearly independent columns corresponding to another spanning tree of the undirected version of G.

Thus, the pivot step of the simplex method can be interpreted as selecting a new basis corresponding to a new spanning tree of the undirected version of G, and improving the current solution of the linear programming problem by moving from one spanning tree to another.

Explained in a more understandable way, the pivot step of the simplex method consists of selecting a column of the matrix A that corresponds to a nonbasic variable, and then using row operations to remove the coefficients of that variable from the other rows. Geometrically, this corresponds to moving along an edge of the polytope defined by the linear constraints of the problem.

If the columns of A correspond to a set of arcs forming a tree, then the pivot step can be interpreted as moving from one vertex of the tree to an adjacent vertex along an edge of the tree. Specifically, the column that is selected for pivoting corresponds to an arc connecting the current vertex to an adjacent vertex in the tree, and the row operations that are performed correspond to updating the node potentials according to Kirchhoff's laws. This interpretation can help to guide the selection of pivots during the simplex method and to understand the geometry of the feasible region.