1

**KONERU LAKSHMAIAH UNIVERSITY**

**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**

**PROJECT REPORT**

**On**

**Ant colony optimization & Goromy cut plane method**

**SUBMITTED BY:**

2100031707 S. Mohammed Ibrahim

**UNDER THE ESTEEMED GUIDANCE OF**

**Dr. Enireddy Vamsidhar**

**CSE**

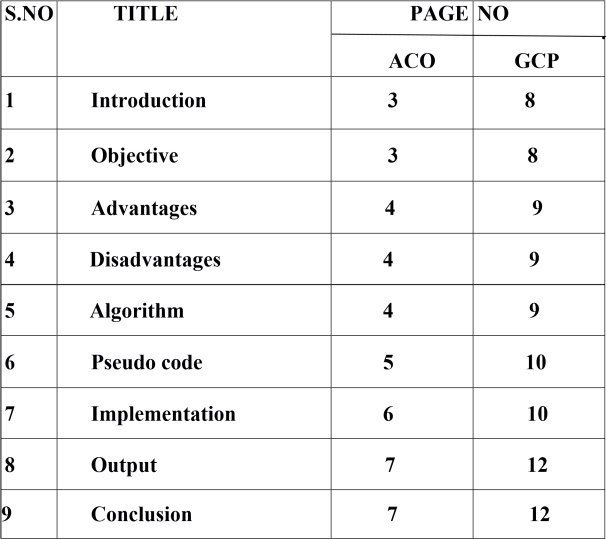


KL UNIVERSITY

Green fields, Vaddeswaram – 522 502 Guntur Dt., AP, India.

2

**INDEX**



# Ant Colony Optimization(Introduction):

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. Artificial ants stand for multi-agent methods inspired by the behavior of real ants. The pheromone-based communication of biological ants is often the predominant paradigm used. Combinations of artificial ants and local search algorithms have become a method of choice for numerous optimization tasks involving some sort of graph, e.g., vehicle routing and internet routing.

As an example, ant colony optimization[3] is a class of optimization algorithms modeled on the actions of an ant colony. Artificial 'ants' ( ex- simulation agents) locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions. One variation on this approach is the bees algorithm, which is more analogous to the foraging patterns of the honey bee, another social insect.

This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. From a broader perspective, ACO performs a model-based search[8] and shares some similarities with estimation of distribution algorithms.

## Objective:

The basic idea of Ant Colony Optimization (ACO) [7] is to model the problem to solve as the search for a minimum cost path in a graph, and to use artificial ants to Page 3 search for good paths.

## Advantages:

They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time.

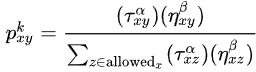
## Disadvantages:

Three main limitation of the algorithm are the stagnation phase, exploration and exploitation rate and convergence speed of the algorithm. The next section describes the simulation runs of ACO algorithm on TSP by taking different ant colony size and varied parameters.

## Algorithm:

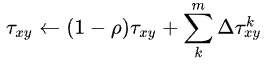
**procedure** ACO\_MetaHeuristic **is while not** terminated **do** generateSolutions() daemonActions() pheromoneUpdate() **repeat end procedure**

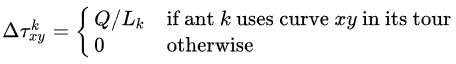
In general, the kth ant moves from state x to state y with probability:



## Pheromone update:

Trails are usually updated when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively. An example of a global pheromone updating rule is





## Pseudocode:

Procedure AntColonyOptimization:

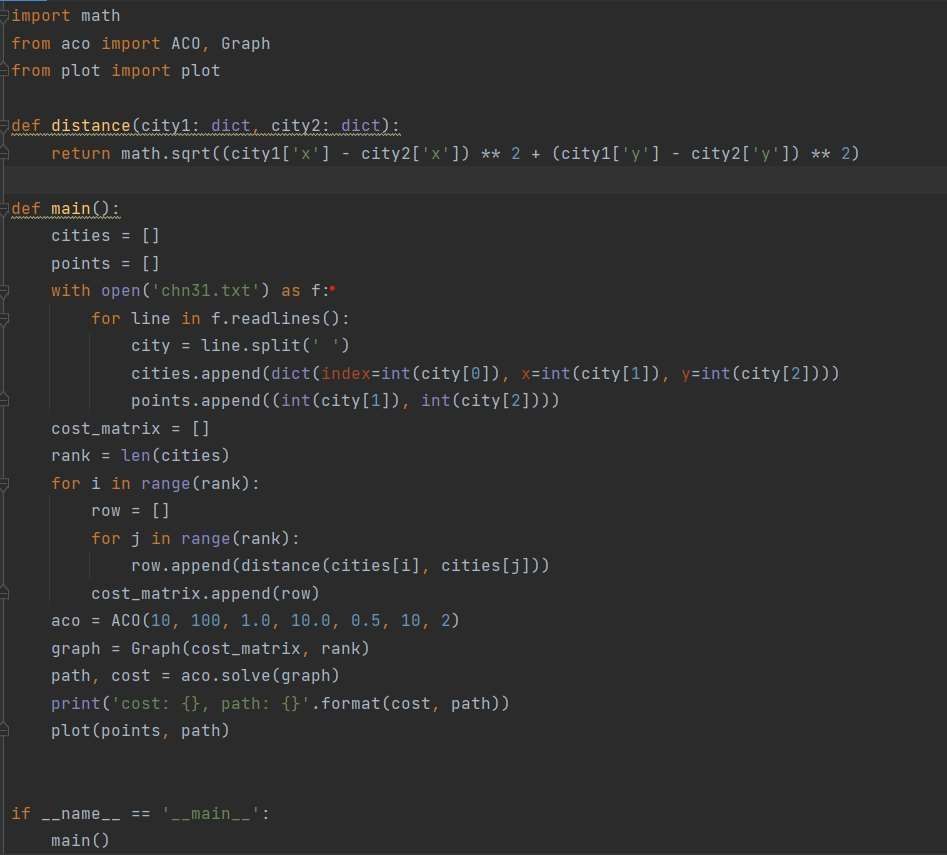
Initialize necessary parameters and pheromone trials; while not termination do:

Generate ant population;

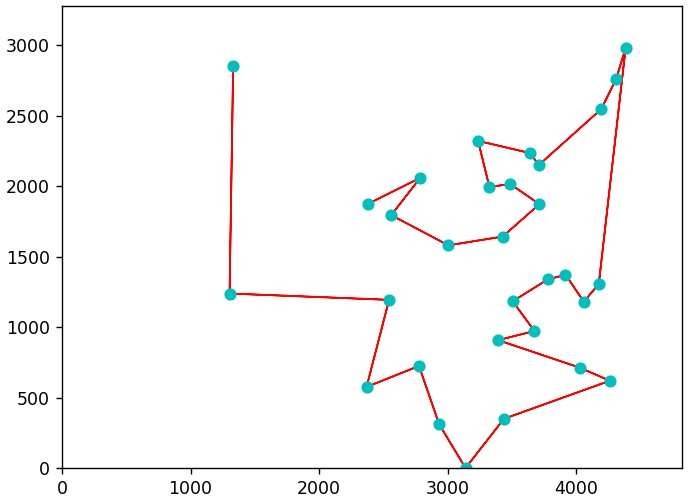
Calculate fitness values associated with each ant; Find best solution through selection methods;

Update pheromone trial; end while end procedure

## Execution:



The output after the execution :

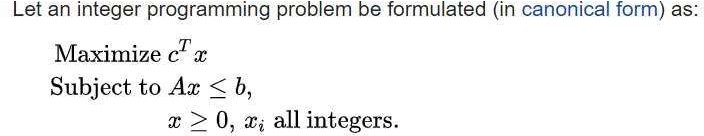


## Conclusion:

The conclusion that this algorithm will definitely converge to the optimal solution under the condition of 0<q 0<1 was proved true. In addition, the influence on its convergence caused by the properties of the closed path, heuristic functions, the pheromone and q 0 was analyzed.

# Goromy cut plane method(Introduction):

Cutting planes were proposed by Ralph Gomory in the 1950s as a method for solving integer programming and mixed-integer programming problems. However, most experts, including Gomory himself, considered them to be impractical due to numerical instability, as well as ineffective because many rounds of cuts were needed to make progress towards the solution. Things turned around when in the mid-1990s Gérard Cornuéjols and coworkers showed them to be very effective in combination with branch-and-bound (called branch-and-cut) and ways to overcome numerical instabilities. Nowadays, all commercial MILP solvers use Gomory cuts in one way or another. Gomory cuts are very efficiently generated from a simplex tableau, whereas many other types of cuts are either expensive or even NP-hard to separate. Among other general cuts for MILP, most notably lift-andproject dominates Gomory cuts.



# Objective:

The basic idea of the cutting-plane (CP) method is to add consecutive constraints to the initial relaxed LP formulation of the considered MIP problem (in the initial relaxed formulation and also in the consecutively constructed problems the integrality requirements for variables are totally skipped) in order to eliminate consecutive nonintegral optimal solutions. More precisely, when we solve the initial continuous problem with the simplex algorithm, and the optimal vertex x\* is integral, then we are done; otherwise, we need to look for an additional inequality (defining a hyperplane) that will diminish the initial solution polytope (or polyhedron) (5.1.1) by cutting off vertex x\*. Then, we add this inequality (defining a hyperplane called the cutting plane) to the original relaxed problem, and continue the procedure. You may note that for the problem in Example 5.5, the first cutting plane could be defined by inequality x1 ≤ 5.

The issue of how the consecutive planes should be selected is difficult, as no systematic way of generating the linear equations or inequalities defining the minimal convex hull of all the integer points contained in a given convex polyhedron is known. In fact such a convex hull can contain an enormous number of facets, even for MIP problems with only few variables and constraints. In this context, the main breakthrough in the CP method was the discovery by R.E. Gomory [Gom60] of a special class of CPs leading to finitely convergent CP algorithms

(note that in general the number of generated planes must be exponential with the size of the problem, as the CP method aims at solving NP-complete problems).

# Advantages:

The first, called Gomory cuts, generates cuts from any linear programming tableau. This has the advantage of ``solving'' any problem but has the disadvantage that the method can be very slow. The second approach is to use the structure of the problem to generate very good cuts.

# Disadvantages:

There are two ways to generate cuts. The first, called Gomory cuts, generates cuts from any linear programming tableau. This has the advantage of ``solving'' any problem but has the disadvantage that the method can be very slow. The second approach is to use the structure of the problem to generate very good cuts.

# Algorithm:

Integer simplex method (gomory's cutting plane method) Steps (Rule)

Step-1: a. Formulate the integer LP problem

b. If any constraint contains non-integer coefficient then convert it into integer.

c. Solve the given problem using Simplex (BigM) method, ignore the integer

Step-2: a. Examine the optimal solution. if all the basic variables have integer values, then terminate the process

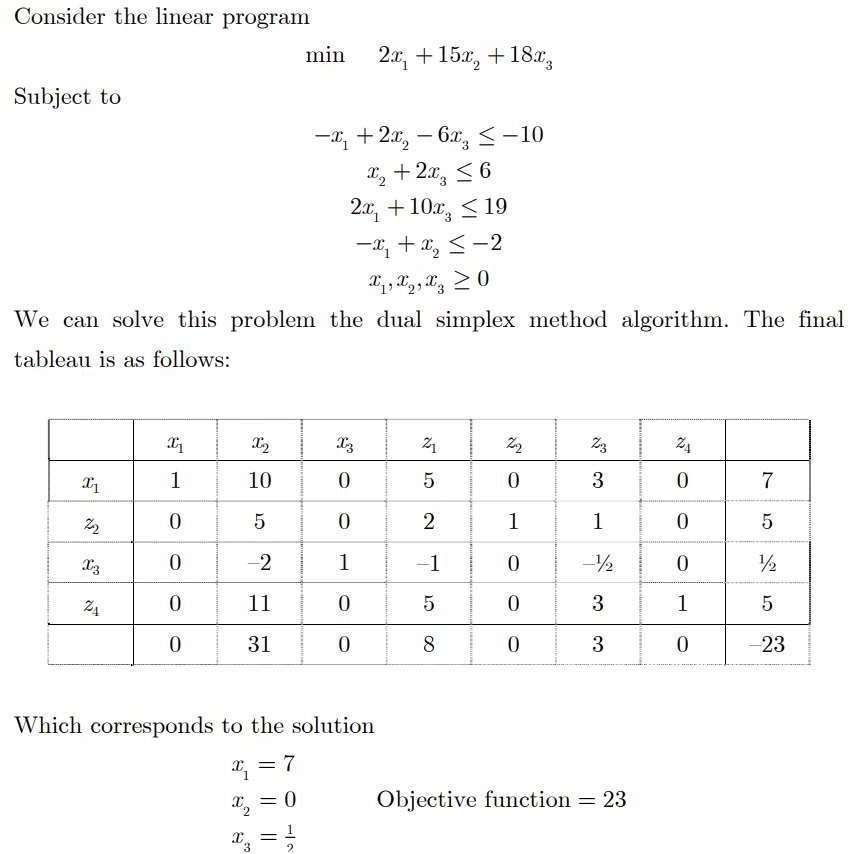
b. Otherwise, construct a Gomory's fractional cut from the row, which has the largest fractional part, and add it to the original set of constraints.

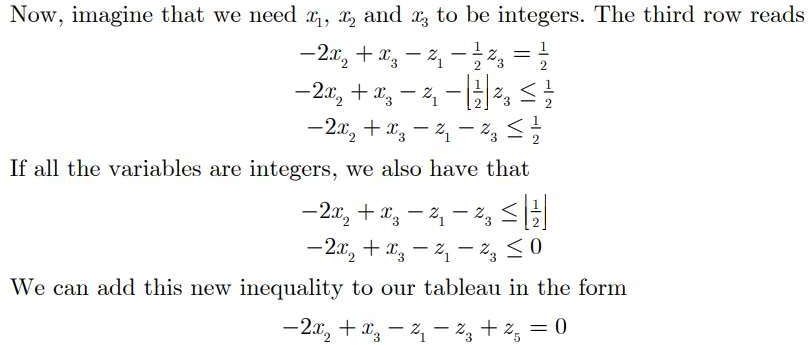
Gomory's constraint -fr=sg-∑frx

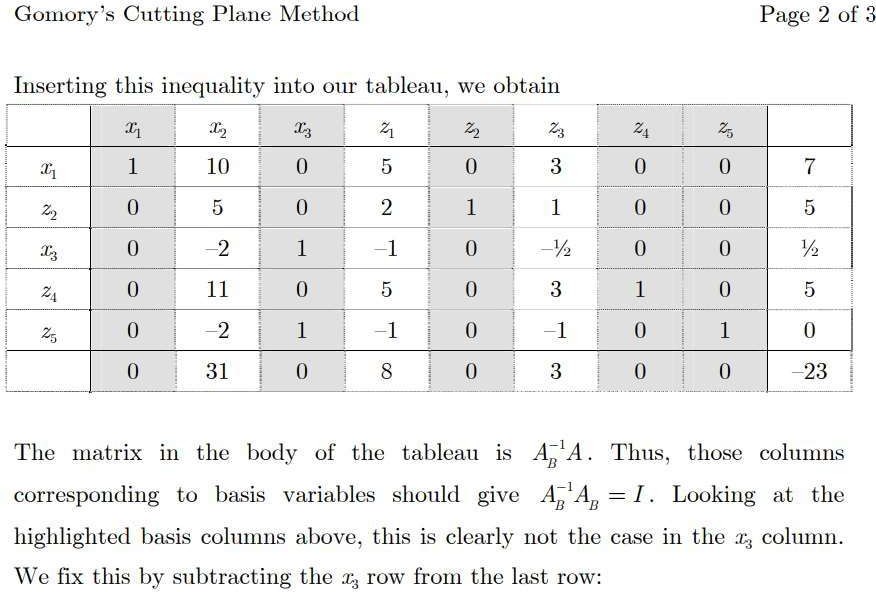
Step-3: a. Now add this constraint to optimal simplex table.

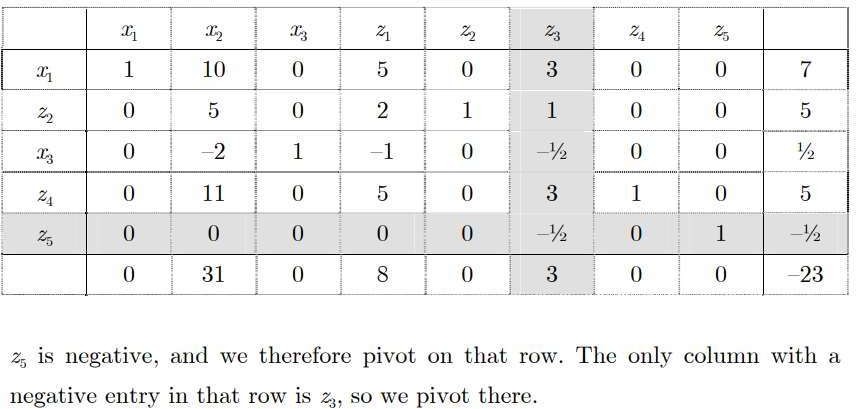
b. Find a new optimal solution using dual simplex method. and then goto step-2.

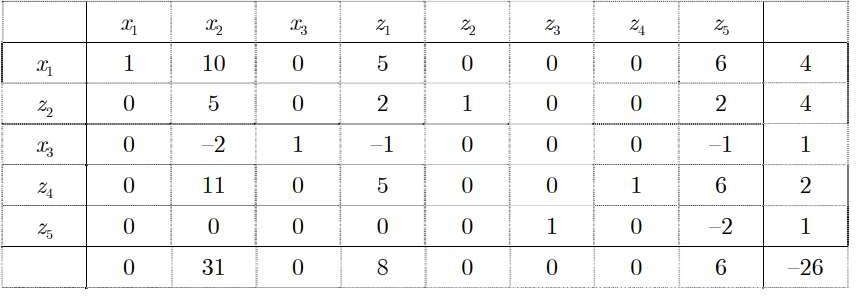
# Example:

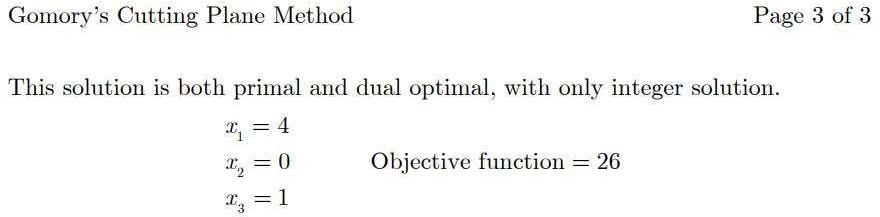












**Conclusion:**

The method proceeds by first dropping the requirement that the xi be integers and solving the associated linear programming problem to obtain a basic feasible solution.

Geometrically, this solution will be a vertex of the convex polytope consisting of all feasible points. If this vertex is not an integer point then the method finds a hyperplane with the vertex on one side and all feasible integer points on the other. This is then added as an additional linear constraint to exclude the vertex found, creating a modified linear program. The new program is then solved and the process is repeated until an integer solution is found.

13