Project Report I

Aryan Singh 2018320
Aminesh Kumar Rai 2018314
Prakash Prajapt 2018350

Solving Bin Packing Problem Using Ant Colony Algorithm

Under the Guidance of Dr. Kusum Kumari Bharti

December 04, 2020

Abstract

Improved ant colony optimization (ACO) was applied to solve the bin-packing problem. A tabu matrix based on goods is established for ant colony optimization (ACO). Tabu matrix's every column corresponds to goods. Matrix has two rows, first row corresponds to goods load visit state, and second row corresponds to vehicle that ferries the goods. Ant state transfer probability is improved to solve the bin-packing problem. According to ants tabu table, all the goods are searched by ants and satisfy the vehicle constraint. Illustrated with a practical case, ACO is feasible for bin-packing problems.

Introduction

The bin packing problem (BPP) is a NP-hard combinatorial optimization problem where the primary aim is to pack a finite number of items using the least bins possible. It involves a number of practical constraints and requirements to be satisfied, such as weight, centre of gravity, irregularly shaped bins and priority items. BPP has many important applications such as multiprocessor scheduling, resource allocation, transportation planning, real-world planning, and packing and scheduling optimization problems. Many algorithms are used to deal with BPP, such as improved approximation algorithms, particle swarm optimization, and ant colony optimization, genetic approach.

Ant Colony Optimisation

ACO algorithms were originally inspired by the ability of real ants to find the shortest path between their nest and a food source. The key to this ability lies in the fact that ants leave a pheromone trail behind while walking. Other ants can smell this pheromone, and follow it. When a colony of ants is presented with two possible paths, each ant initially chooses one randomly, resulting in 50% going over each path. It is clear, however, that the ants using the shortest path will be back faster. So, immediately after their return there will be more pheromone on the shortest path, influencing other ants to follow this path. After some time, this results in the whole colony following the shortest path. AS is a constructive meta-heuristic for the TSP based on this biological metaphor. It associates an amount of pheromone τ (i, j) with the connection between two cities i and j. Each ant is placed on a random start city, and builds a solution going from city to city, until it has visited all of them. The probability that an ant k in a city i chooses to go to a city j next is given by equation 3:

$$p_k(i,j) = \begin{cases} \frac{[\tau(i,j)].[\eta(i,j)]^\beta}{\sum_{g \in J_k(i)} [\tau(i,g)].[\eta(i,g)]^\beta} & \text{if } j \in J_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In this equation, τ (i, j) is the pheromone between i and j and η(i, j) is a simple heuristic guiding the ant. The value of the heuristic is the inverse of the cost of the connection between i and j. So the preference of ant k in city i for city j is partly defined by the pheromone between i and j, and partly by the heuristic favourability of j after i. It is the parameter β which defines the relative importance of the heuristic information as opposed to the pheromone information. Jk(i) is the set of cities that have not yet been visited by ant k in city i. Once all ants have built a tour, the pheromone is updated. This is done according to these equations:

$$\tau(i,j) = \rho.\tau(i,j) + \sum_{k=1}^{m} \Delta\tau_k(i,j) \tag{4}$$

$$\Delta\tau_k(i,j) = \begin{cases} \frac{1}{L_k} & \text{if } (i,j) \in \text{ tour of ant } k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Equation (4) consists of two parts. The left part makes the pheromone on all edges decay. The speed of this decay is defined by ρ, the evaporation parameter. The right part increases the pheromone on all the edges visited by ants. The amount of pheromone an ant k deposits on an edge is defined by Lk, the length of the tour created by that ant. In this way, the increase of pheromone for an edge depends on the number of ants that use this edge, and on the quality of the solutions found by those ants.

BPP MODEL STRATEGY BASED ON ANT COLONY OPTIMIZATION

The ant colony optimization (ACO) is a kind of simulant evolution algorithm based on real ant colony which is proposed by Italy experts M.Dorigo, V.Maniezzo and A. Colomi.

An important and interesting behavior of ants in colonies is their foraging behavior and in particular how they can find the shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called a pheromone, forming in this way a pheromone trail. Ants can smell a pheromone, and when choosing their way they tend to choose paths marked by strong pheromone concentrations. When several paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back. ACO is a multi agent heuristic search approach to difficult combinatorial optimization problems.

A. Parameter Representation

In ACO, one dimensional table tabuk denotes depots state which will be visited by ant k, depots state is 0 when the depot was visited, otherwise is 1. Tabu table needs to be modified to solve BPMDVSP.

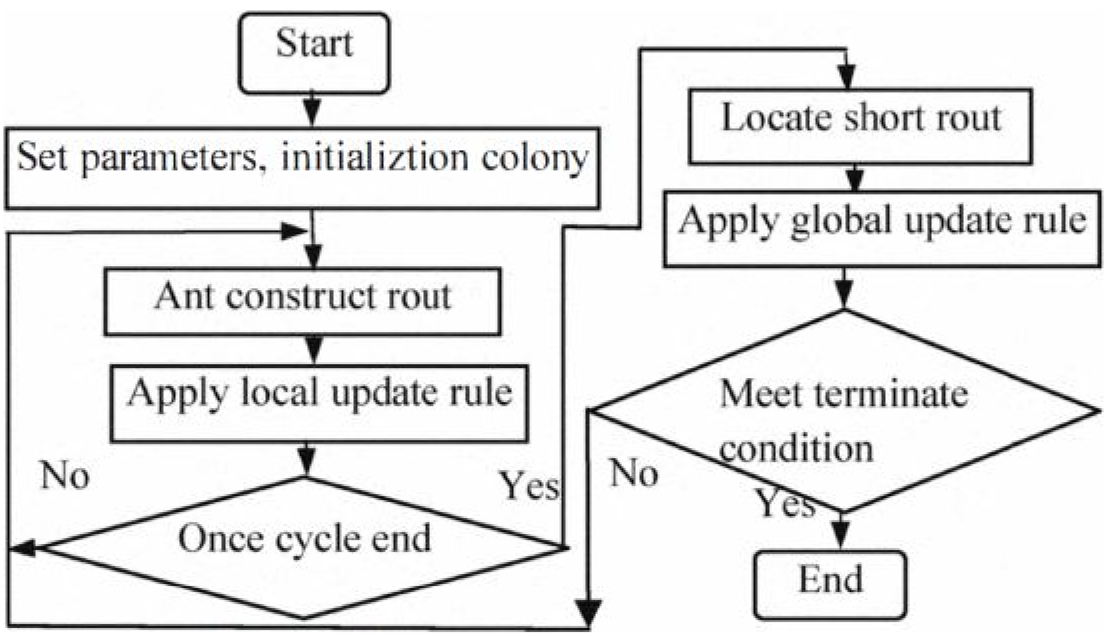
Considering BPP, a tabu matrix bases on goods is established for ant colony optimization (ACO). Tabu matrix's every column corresponds to goods. Matrix has two rows, first row corresponds to goods load visit state, and second row corresponds to vehicle that ferries the goods.

Vehicle serial number that ferry goods initialize as-1, when goods are loaded, goods visit state is 0, vehicle serial number is the vehicle that ferry goods, and otherwise goods visit state is 1. Tabu matrix expression is presented below.

$$\text{tabu}= \begin{array}{c} \text{goods} \\ \text{goods visit state} \\ \text{vehicle serial number that ferry goods} \end{array} \begin{array}{c} \begin{array}{cccccc} u_1 & u_2 & \cdots & u_n & \cdots & u_N \end{array} \\ \left[\begin{array}{c:c:c:c:c:c} 0 & 1 & & 0 & & 0 \\ B_k & -1 & \cdots & B_{k+2} & \cdots & B_k \end{array} \right] \end{array}$$

B. Step of the Algorithm

When initializing an ant colony, set ants randomly on goods, and the correlative tabu matrix elements need to be modified. A generic ACO procedure is given as follows:



CONCLUSIONS

Improved ant colony optimization (ACO) was applied to solve the bin-packing problem. A tabu matrix based on goods is established for ant colony optimization (ACO). Tabu matrix's every column corresponds to goods. Matrix has two rows, first row corresponds to goods load visit state, and second row corresponds to vehicle that ferries the goods. Ant state transfer probability is improved to solve the bin-packing problem. According to ants tabu table and vehicle subject function all the goods are searched by ants and satisfy the vehicle constraint. Illustrated result shows ACO is feasible for bin-packing problems.

References

[1] Ant Colony Optimization:
https://www.researchgate.net/publication/225265937_Ant_Colony_Optimization_Overview_and_Recent_Advances

[2] Bin Packing Problem
Review:https://www.researchgate.net/publication/245423006_Ant_colony_optimization_and_local_search_for_bin_packing_and_cutting_stock_problems