DOI: 10.20396/revpibic2720192195

XXVII Congresso de Iniciação Científica
Unicamp
16 a 18 de outubro de 2019 - Campinas | Brasil

Metaheuristics For Solving Mathematical Programming Problems

Vitor H. Washiya*, Aurelio R. L. Oliveira

Abstract

This undergraduate research aims to study metaheuristics for solving mathematical programming problems. In optimization's context, metaheuristics are strategies which use problem specific knowledge to, stochastically use solutions worse than the current in order to avoid local maximum or minimum. The literature about metaheuristics is huge and continues in expansion, mostly by its commercial interest, once most real problems are far too large to be solved by exact algorithms. This project considered a wide variety of metaheuristics and focused on a particular one: The Ant Colony Optmization applied to the Symmetric Travelling Salesman Problem (TSP).

Key words:

Ant Colony Optimization, Mathematical Programming, Metaheuristics

Introduction

The Travelling Salesman Problem (TSP) can be stated as: "Given a list of cities and the distances between each pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point". More generally, the cities are nodes in a complete graph and the distances are the arcs connecting them. In the symmetric case, the cost to go from node i to j and from node j to i is the same.

Currently, the TSP is considered a NP-hard problem, in other words, there is no polynomial time exact algorithm to solve it. Therefore, other strategies as metaheuristics are sought. Although, metaheuristics do not guarantee optimality, they quickly find very good solutions.

This project studied the ACO metaheuristic applied to the TSP with a C implementation of the Ant System. The parameters were calibrated by random generated graphs. After that, its efficiency was tested using instances from the opensource library TSPLIB. Furthermore, we verified the quality of the random numbers from the C library rand() by the rescaled range R/S analysis, the Hurst exponent.

Results and Discussion

ACO metaheuristics have three main phases: Initialization (setting the parameters), building ants' solutions, and updating pheromone trail.

In the phase that ants build their solutions, each ant has a starting node and acordingly to the probability:

$$P_{ij}^k = \begin{cases} \dfrac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{l \in N_i^k} (\tau_{il})^\alpha (\eta_{il})^\beta}, & if\ j \in N_i^k \\ 0, & Otherwise \end{cases}$$

chooses the next node to be visited. After every ant has built their solution, the pheromone trail phase starts. In this phase, we first evaporate the pheromone trail to "forget" bad decisions, i. e., $\tau_{ij} \leftarrow (1-\rho)\tau_{ij}$.

Then, each ant deposits pheromone in the arcs belonging to its solution, the amount deposited on each arc is:

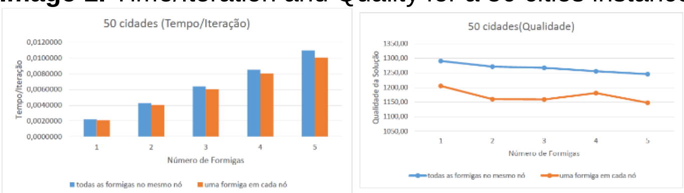
$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L^k}, & if\ (i,j) \in M^k \\ 0, & Otherwise \end{cases}$$

The two stopping criteria are the number of iterations and number of iterations without improving the solution.

To calibrate the parameters $\alpha, \beta, \rho, \tau_0$ we used a random generated instance of 50 cities. Based on results given on [1], we tried $m = 50,\ \tau_0 = m/C^{nn} \approx 0{,}02$ and the intervals $\alpha \in \{0{,}7; 0{,}8; \ldots; 1{,}3\}$, $\beta \in \{0{,}5; 1{,}0; \ldots; 1{,}5\}$, $\rho \in \{0{,}001; 0{,}005; \ldots; 1{,}5\}$, $Q \in \{0{,}5; 1{,}0; 1{,}5\}$ and by exhaustion comparing averages of time/iterations and quality of solution we found that a good set of parameters for this instance is: $\alpha = 1, \beta = 1, \rho = 1, Q = 1$.

For the number of ants m we analysed whether they should start at the same node or start with one ant on each node. From graphics like the one bellow, it seems like m = n and ants spread around the graph give better results. To higher order, this advantage is smaller.

Image 1. Time/Iteration and Quality for a 50 cities instance



Using these parameters we ran some tests from the TSPLIB using 10 000 iterations without improvement as stopping criteria, here is a table with partial results:

Chart 1. Sample from the TSPLIB tests

Instance	Best Solution	Solution Found
Gr17	2085	2085
Brazil58	25395	26656
Brg180	1950	4800
Gr48	5046	5046

Conclusions

Considering the data shown in the table we can see that ACO metaheuristics are a powerful tool to solve the TSP problem, for it can be applied to large instances to find good solutions in a feasible time.

[1] Dorigo, M.;Stützle, T. Ant Colony Optimization. Italy: MIT Press, 2004.
[2] Dorigo, M. Optimization, learning and natural algorithms. Italy: PhD Thesis, Politecnico di Milano, 1992.
[3] Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: optimization by a colony of cooperating agentes. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), v. 26, n. 1, p. 29-41, 1996.
[4] Gendreau, M.; Potvin, J. Handbook of Metaheuristics. London: Springer, second edition, 2010.
[5] Hurst, Harold Edwin. Long-term storage capacity of reservoirs. Trans. Amer. Soc. Civil Eng., v. 116, p. 770-799, 1951.
[6] Mandelbrot, Benoit B.; Wallis, James R. Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence. Water Resources Research, v. 5, n. 6, p. 967-988, 1969.