Available online at www.sciencedirect.com
SCIENCE DIRECT
Electronic Notes in Discrete Mathematics 18 (2004) 181–186
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

An Ant Colony Algorithm for the Capacitated Vehicle Routing

Silvia Mazzeo[1] Irene Loiseau[2]

Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires,Argentina.

Abstract

The Vehicle Routing Problem (VRP) requires the determination of an optimal set of routes for a set of vehicles to serve a set of customers. We deal here with the Capacitated Vehicle Routing Problem (CVRP) where there is a maximum weight or volume that each vehicle can load. We developed an Ant Colony algorithm (ACO) for the CVRP based on the metaheuristic technique introduced by Colorni, Dorigo and Maniezzo. We present preliminary results that show that ant algorithms are competitive with other metaheuristics for solving CVRP.

Keywords: Capacitated Vehicle Routing Problem, Ant Colony, Metaheuristics

1 Introduction

The Vehicle Routing Problem (VRP) requires the determination of a optimal set of routes for a set of vehicles to serve a set of customers. The problem as it appears on real life may have several classes of additional constraints, as limit on the capacity of the vehicles, time windows for the customer to be served, limits on the time a driver can work, limits on the lengths of the routes, etc.. The problem was first introduced by Dantzig and Ramser in 1959. Due to the intrinsic interest as difficult combinatorial optimization problem and to the economical importance of applications VRP has received a lot of attention and many algorithms both exact and heuristics have been

[1] Email: smazzeo@quasar-arg.com.ar

[2] Email: irene@dc.uba.ar. Partially supported by grants UBACYT EX 036, PICT 11-09112.

doi:10.1016/j.endm.2004.06.029

developed since then to solve general problem and real world cases and so literature on the subject is very extensive. For a recent complete review on models and algorithms for VRP see [8]. There are also several commercial packages available designed to give good solutions for real world problems. We deal here with the Capacitated Vehicle Routing Problem (CVRP), that is we have: a depot where vehicles start and end their routes, a set of clients and their demands, a set of vehicles with a maximum weight or volume that each one can load, and costs or distances between clients and between clients and the depot. We want to define routes for the vehicles starting and ending at the depot that satisfy the clients demand at a minimum total cost. As most VRP problems, CVRP is known to be NP-hard.

2 Ant Colony Algorithms

Ant colony algorithms are inspired on an analogy with real life behavior of a colony of ants when looking for food. In their search they mark the trails they are using by laying a substance called pheromone. The amount of pheromone in a path lets other ants to know if it is a promising path or not. This observation inspired Colorni, Dorigo and Maniezo [3] for proposing a metaheuristics technique: ants are procedures that build solutions to a optimization problem. According to how the solution space is explored some values are recorded in a similar way as pheromone acts, and objective values of solutions are associated with food sources. An important issue of this algorithms is parallelism: several solutions are built at the same time and they interchange information during the procedure and use information of previous iterations.

At each iteration of the basic Ant Colony method, each ant builds a solution of the problem step by step. At each step the ant makes a move in order to complete the actual partial solution choosing between elements of a set A_k of expansion states, following a probability function. For each ant k probability $P_k(i,j)$ of moving from present state i to another state j is calculated taking into account:

- attractiveness η of the move according to the information of the problem.
- level τ of pheromone of the move that indicates how good the move was in the past.
- a $Tabu_k$ list of forbidden moves

In the ant algorithm original version formula for $P_k(i,j)$ is:

$$P_k(i,j) = \begin{cases} \frac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum_{(i,z)\notin Tabu_k} [\tau(i,j)]^\alpha [\eta(i,j)]^\beta} & if (i,j) \notin Tabu_k \\ 0 & if (i,j) \in Tabu_k \end{cases}$$

where α, β are parameters that are used to establish the relative influence of η versus τ. After iteration t is complete, that is when all the ants have completed their solutions, the pheromone levels are updated to:

$$\tau(i,j) = \varphi.\tau(i,j) + \Delta\tau(i,j)$$

where φ is a coefficient representing the level of pheromone persistence and $\Delta\tau(i,j)$ represents the contributions of all ants that chose move (i,j) in their solution.

Successful ant algorithms have been developed for several combinatorial optimization problems.

3 An Ant Colony algorithm for CVRP

We resume here the main characteristics of our ACO algorithms for CVRP. We tried several alternatives for each component of the algorithm:

(i) **Routes building :** at each iteration of ACO each ant builds a solution for the CVRP, moving to next client (state in the general ACO scheme) according to transition rules based on a combination of the amount of pheromone at each arc and length of it (see (2) below). The role of Tabu list mentioned at previous section is taken here by a set of already visited neighbors, forbidden for the ant at current iteration. We implemented two versions of the order in which routes are determined:

- *Sequential*: each ant start its solution determining the route for the first vehicle till its capacity is complete. Then it continues with others vehicles till complete the solution . Each ant starts its solution from a different client.
- *Parallel*: each ant designs the route for all vehicles at the same time. At each iteration of the algorithm only one client is chosen, according to transition rule. Then best tour is extended.

(ii) **Transition rules**:

- *Random-Proportional rule*: a neighbor client is randomly chosen according to probability $P_k(i,j)$ calculated as described in section 2.
- *Pseudo-Random-Proportional rule*: this rule for choosing next client to visit combines random selection with best option. Let q_0 such that $0 \leq q_0 \leq 1$, we generate q a random number in [0,1], then next client j is chosen according to:

$$j = \begin{cases} max_{u \in \Gamma(i)}[\tau(i,u)^{\alpha}][\eta(i,u)]^{\beta} \; if q \leq q_0 & \textit{(explotation)} \\ J \qquad\qquad\qquad\qquad if q > q_0 & \textit{(exploration)} \end{cases}$$

where J is randomly selected according to $P_k(i,j)$.

(iii) **Pheromone actualization:** we tried the following alternatives that appear at the ant colony literature:

(a) **Global actualization**:is done after each ACO iteration is completed.

- *All the solutions*: is the one proposed in the original version of the algorithm, where pheromone levels are actualized at each iteration after all ants complete their routes.
- *Elite Ants*: only ants that obtained the best solutions are take into account for the actualization:

$$\tau(i,j) = \varphi.\tau(i,j) + \sum_{\mu}^{\sigma-1} \Delta\tau_{\mu} + \sigma.\Delta\tau^{*}(i,j)$$

where φ is the factor de pheromone persistence and σ is the number of elite ants,

$$\Delta\tau_{\mu}(i,j) = \begin{cases} \frac{(\sigma-\mu)}{L\mu} & \textit{if (i,j) is part of a solution of the } \mu\textit{-best ant} \\ 0 & \textit{otherwise} \end{cases}$$

where $L\mu$ is the solution of the μ-best ant, and

$$\Delta\tau^{*}(i,j) = \begin{cases} \frac{1}{L^{*}} & \textit{if (i,j) is part of the best solution} \\ 0 & \textit{otherwise} \end{cases}$$

where L^{*} is the value of best solution.

- *Best solution*: pheromone is updated using only information of best solution of the previous iteration.

(b) **Local actualization**: It is done each time an ant moves from one client i to the next j to decrease the amount of pheromone of an used edge (i,j), in order to diversify solutions obtained by the ants.

$$\tau(i,j) = \varphi.\tau(i,j) + (1-\varphi)\Delta\tau(i,j)$$

Several ways to determine $\Delta\tau(i,j)$ where tested:

- *Q-learning*: inspired in Q-learning, a method for automatic learning, we define:

$$\Delta\tau_{k}(i,j) = \gamma.max_{z\in\Gamma(j)}\tau(j,z) \ (0 \leq \gamma < 0)$$

- *Initial pheromone*: we use $\Delta\tau(i,j) = \tau_0$ where τ_0 is the initial pheromone level.
- *Proportional to distance*: $\Delta\tau(i,j) = \frac{\tau^{0}}{d(i,j)}$
- *Evaporation*: $\Delta\tau(i,j) = 0$, that is $\tau(i,j) = \varphi.\tau(i,j)$

(iv) **Reduced neighbor list:** when the problem is too big to explore all the potential moves of the ant, a reduced list of best candidates is used.

(v) **Improving heuristics** are used to modify ant solutions after each iteration. Details of the several simple and multiple interchange procedures implemented, based on original ideas of [4] and [6], can be read at [5] .

(vi) **Stopping rules**: ACO procedure stops when there is not improvement on the solution after several iterations or when $nmax$ number of iterations is reached.

4 Computational results

We first carried on an extensive set of experiments that we can not present here because of lack of space. We used a set of problems to decide which options of implementation and parameters and combinations of them worked better (see detailed report at [5]). Some of the conclusions showed are:

- parallel building of the routes behave better than sequential.
- using best solution global updating was the best between all the global updates ideas already mentioned.
- use of local pheromone updating did not improve quality of solutions.
- use of improvement heuristics derived in significantly better final results.
- use of reduced (25%) candidate lists and of a smaller number of randomly located ants than clients, showed to diminished significantly running times.

Then we used our version of ACO with parallel vehicle assignation, without candidate lists, with improvement heuristics (3 cycles in each iteration) to compare it with other heuristics on the literature, in a different set of problems. Data of the problems, bounds, optimal solution, etc., where taken from [8] and from http://www.branchandcut.org/VRP/data. The algorithm obtained the optimum value for all the problems we run of less than 50 clients. For some bigger problems results are shown on next table where: E-n51-k5 means a problem with 50 clients and 5 vehicles and Cl.Wr(1) refers to Clarke and Wright Savings Algorithm, parallel version [2], Cl.Wr.3-Opt(2) to Clarke and Wright, 3-opt, FI version, Sim(4) to Simulated Annealing by Osman[7], Tabu(4) to Tabu Search by Osman[7], and ACO is the algorithm we developed. As ACO is a probabilistic algorithm results can vary a from a run to another. Here results of our algorithm correspond to the best value of only two or three runs. We found very few references of Ant Colony systems for the VRP. We only got results to compare with some few cases of [1]. Our algorithm obtained better solutions in some of the problems as in E-n51-k5 and slightly worse in others. Description of algorithm in [1] is not completely detailed but some differences we can see with ours are that they use different formula for η and only 2-opt improving heuristics.

Problem	Cl.Wr(1)	Cl.Wr.3Op(2)	Sim(3)	Tabu(4)	ACO	UBound	Gap
E-n51-k5	584.64	578.56	528	524.61	521	521	0
E-n76-k10	900.26	888.04	838.62	844	877	832	?
E-n101-k8	886.83	878.70	829.18	835	845	815	?
M-n101-k10	833.51	824.42	826	819.59	838	820	0
M-n121-k7	1071.07	1049.43	1176	1042.11	1189	1042.11	?
M-n151-k12	1133.43	1128.24	1058	1052	1105	1028.42	?
M-n200-k17	1395.74	1386.84	1378	1354	1606	1291.45	?

5 Conclusions

Preliminary results obtained show that performance of our ACO is very good in problems up to 50 nodes, and promising in bigger ones. Future work includes improvement on the implementation, in order to obtain better solutions for medium to big problems and better running times.

References

[1] B. Bullnheimer, R. F. Hartl and C. Strauss. “An improved ant system algorithm for the vehicle routing problem”. Annals of Operations Res., 89, 319-328, 1999.

[2] G. Clarke and J.V. Wright. “Scheduling of vehicles from a central depot to a number of delivery points”. Operations Research, 12, 568-581, 1964.

[3] A. Colorni, M. Dorigo, V. Maniezzo. “Distributed Optimization by Ant Colonies”. First European Conference on Artificial Life, 134-142, 1991.

[4] S. Lin, B.,W. Kernighan. “An effective heuristic algorithm for the traveling salesman problem”. Operations Research 21, 498-516, 1973.

[5] S. Mazzeo. “Un algoritmo de colonia de hormigas para el problema de ruteo de vehículos”, Tesis de Licenciatura, FCEN, UBA, 2003.

[6] I. Or.“Traveling Salesman-type combinatorial optimization problems and the relation to the logistics of regional blood banking”. Ph.D.dissertation. NorthWestern Univ.,1976.

[7] I. H. Osman. “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem”. Annals of Operations Res. 41, 421-451, 1993.

[8] P. Toth, D. Vigo(eds). “The Vehicle Routing Problem”. SIAM monographs on discrete mathematics and applications, 2002.