Convergence Proof for a Monte Carlo Method for Combinatorial Optimization Problems

Stefka Fidanova

IPP – BAS, Acad. G. Bonchev str. bl.25A, 1113 Sofia, Bulgaria
stefka@parallel.bas.bg

Abstract. In this paper we prove the convergence of a Monte Carlo (MC) method for Combinatorial Optimization Problems (COPs). The Ant Colony Optimization (ACO) is a MC method, created to solve efficiently COPs. The Ant Colony Optimization (ACO) algorithms are being applied successfully to diverse heavily problems. To show that ACO algorithms could be good alternatives to existing algorithms for hard combinatorial optimization problems, recent research in this area has mainly focused on the development of algorithmic variants which achieve better performance than previous one. In this paper we present ACO algorithm with Additional Reinforcement (ACO-AR) of the pheromone to the unused movements. ACO-AR algorithm differs from ACO algorithms in several important aspects. In this paper we prove the convergence of ACO-AR algorithm.

1 Introduction

Some time it is more important to find quickly good although not necessarily optimal solution. In this situation, the heuristic methods are with big efficient. For some difficult Combinatorial Optimization Problems (COPs) one or more months is needed to find an optimal solution on powerful computer and only some minutes to find solution by heuristic methods, which is very close to optimal one. Typical examples of practical COPs are the machine-scheduling problem, the net-partitioning problem, the traveling salesman problem, the assignment problem, etc.. Monte Carlo methods have been implemented to efficiently provide flexible and computerized procedures for solving many COPs.

ACO [1,2,3] is a MC method, created to solve COPs. It is a meta-heuristic procedure for quickly and efficiently obtaining high quality solutions to complex optimization problems [9]. ACO algorithm can be interpreted as parallel replicated Monte Carlo systems [11]. MC systems [10] are general stochastic simulation systems, that is, techniques performing repeated sampling experiments on the model of the system under consideration by making use of a stochastic component in the state sampling and/or transition rules. Experimental results are used to update some statistical knowledge about the problem, as well as the estimate of the variables the researcher is interested in. In turn, this knowledge can be also iteratively used to reduce the variance in the estimation of the described variables, directing the simulation process toward the most interesting state space

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3039, pp. 523–530, 2004.

regions. Analogously, in ACO algorithms the ants sample the problem's solution space by repeatedly applying a stochastic decision policy until a feasible solution of the considered problem is built. The sampling is realized concurrently by a collection of differently instantiated replicas of the same ant type. Each ant "experiment" allows to adaptively modify the local statistical knowledge on the problem structure. The recursive retransmission of such knowledge determines a reduction in the variance of the whole search process the so far most interesting explored transitions probabilistically bias future search, preventing ants to waste resources in not promising regions of the search.

In this paper, the basic ACO algorithm has been modified and a convergence proof is presented. The ACO algorithms were inspired by the observation of real ant colonies [1,2,4]. Ants are social insects, they live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. An interesting behavior is how ants can find the shortest paths between food sources and their nest. While walking from a food source to the nest and vice-versa, ants deposit on the ground a substance called pheromone. Ants can smell pheromone and then they tend to choose, in probability, paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest).

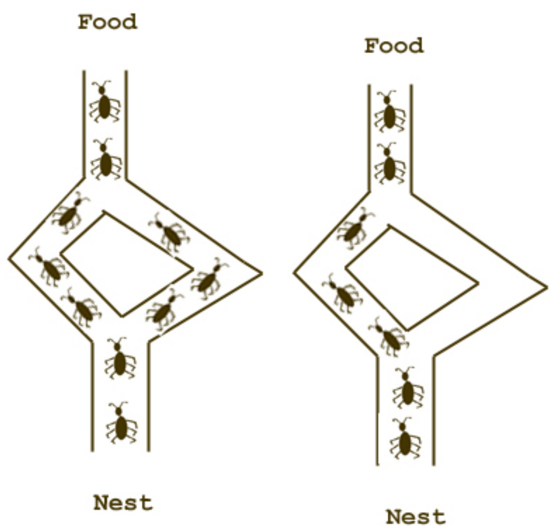

Fig. 1. Behavior of real ants at the beginning of the search and after some minutes

Figure 1 shows how ants can exploit pheromone to find a shortest path between two points. In this, figure ants arrive at a decision point in which they have to decide to turn on the left or on the right. The ants turning on the left first achieve the food sours. When they return back there is a pheromone only in a left side and they choose it and double the pheromone. Thus, after a short transition period the difference in the amount of pheromone on the two paths is sufficiently large and the new ants will prefer in probability to choose the left path, since at the decision point they receive a greater amount of pheromone on the left path. Very soon all ants will be using the shorter path.

The above behavior of real ants has inspired ACO algorithm. ACO algorithm, which is a population-based approach, has been successfully applied to many NP-hard problems [3,7,8]. One of its main ideas is the indirect communication among the individuals of ant colony. This mechanism is based on an analogy with trails of pheromone which real ants use for communication. The pheromone trails are a kind of distributed numerical information which is modified by the ants to reflect their experience accumulated while solving a particular problem.

The main purpose of this paper is to use additional reinforcement of the pheromone to the unused movements and thus to effectively avoid stagnation of the search and to prove the convergence of ACO-AR to the global optimum. The remainder of this paper is structured as follows. Section 2 describes the developed ACO-AR algorithm, while section 3 investigates its convergence. Section 4 shows parameter settings. The paper ends with conclusions and some remarks.

2 The ACO Algorithm

The ACO algorithms make use of simple agents called ants which iteratively construct candidate solutions to a COPs. The ants' solution construction is guided by pheromone trail and problem dependent heuristic information. The ACO algorithms can be applied to any COP by defining solution components which the ants use to iteratively construct candidate solutions and on which they may deposit a pheromone. An individual ant constructs a candidate solution by starting with a random partial solution and then iteratively adding new components to their partial solution until a complete candidate solution is generated. We will call each point at which an ant has to decide which solution component to add to its current partial solution a choice point. After the solution is completed, ants give feedback on their solutions by depositing pheromone on the components of their solutions. Typically, solution components which are part of the best solution, or are used by many ants, will receive a higher amount of pheromone and hence will be more attractive by the ants in following iterations. To avoid the search getting stuck before the pheromone trails get reinforced pheromone trails are decreased.

In general, all ACO algorithms adopt specific algorithmic scheme as follows. After the initialization of the pheromone trails and control parameters, a main loop is repeated until the stopping criteria are met. The stopping criteria can be a certain number of iterations or a given CPU-time limit. In the main loop, the ants construct feasible solutions, then the pheromone trails are updated. More precisely, partial solutions are seen as follow: each ant moves from a state i to another state j of the partial solution. At each step, ant k computes a set of feasible expansions to its current state and moves to one of these expansions, according to a probability distribution specified as follows. For ant k, the probability p_{ij}^k of moving from state i to a state j depends on the combination of two values:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}\eta_{ij}}{\sum_{l \in allowed_k} \tau_{il}\eta_{il}} & \text{if } j \in allowed_k \\ \\ 0 & otherwise \end{cases} \tag{1}$$

where:

- η_{ij} is the attractiveness of the move as computed by some heuristic information indicating the a prior desirability of that move;
- τ_{ij} is the pheromone trail level of the move, indicating how profitable it has been in the past to make that particular move (it represents therefore a posterior indication of the desirability of that move).
- $allowed_k$ is the set of remaining feasible states.

Thus, the higher the value of the pheromone and the heuristic information, the more profitable it is to include state j in the partial solution. In the beginning, the initial pheromone level is set to τ_0, which is a small positive constant. While building a solution, ants change the pheromone level of the elements of the solutions by applying the following updating rule:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \Delta\tau_{ij} \tag{2}$$

where in the rule $0 < \rho < 1$ models evaporation and $\Delta\tau_{ij}$ is different for different ACO algorithms. In ant system [1], the first ant algorithm, all ants change the pheromone level depending to the quality of their solution. In ant colony system [2] extra pheromone is put on the elements of the best solution. In ACO algorithms with elitist ants [3] only small number of ants update the pheromone and so on.

Stagnation situation may occur when we perform the ACO algorithm. This can be happened when the pheromone trail is significantly higher for one choice than for all others. This means that one of the choices has a much higher pheromone level than the others and an ant will prefer this solution component over all alternatives. In this situation, ants construct the same solution over and over again and the exploration of the search space stops. The stagnation situation should be avoided by influencing the probabilities for choosing the next solution component which depend directly on the pheromone trails.

The aim of the paper is to develop the functionality of the ACO algorithms by adding some diversification such as additional reinforcement of the pheromone. This diversification guides the search to areas in the search space which have not been yet explored and forces ants to search for better solutions. We will call the modified ACO algorithm with additional reinforcement [5,6]. If some movements are not used in the current iteration, additional pheromone reinforcement will be used as follows.

$$\tau_{ij} \leftarrow \alpha\tau_{ij} + q\tau_{max} \quad \alpha = \begin{cases} 1 \text{ if unused movements are evaporated} \\ \rho \text{ otherwise} \end{cases} \tag{3}$$

where $q \geq 0$, τ_{max} is the maximal value of the pheromone. Using ACO-AR algorithm the unused movements have the following features:

– they have great amount of the pheromone then the movements belonging to poor solutions.
– they have less amount of the pheromone then the movements belonging to the best solution

Thus the ants will be forced to choose new direction of search space without repeating the bad experience.

3 Convergence of the ACO-AR Algorithm

This section describes the convergence of the ACO-AR algorithm to the global optimum. We will use the Theorem 1 from [12], which proves that if the amount of the pheromone has a finite upper bound τ_{max} and a positive lower bound τ_{min}, then the ACO algorithm converges to the optimal solution. From the Proposition 1 (see [12]),the upper bound of the pheromone level is $g(s^*)/(1-\rho)$, where $g(s^*)$ is the maximum possible amount of a pheromone added after any iteration.

In some of ACO algorithms a pheromone is added on all used movements and in others ACO algorithms only on used movements which belong to the best solutions. The other possible movements are evaporated or stay unchanged. Thus, the lower bound of the pheromone level of some of them can be 0. After additional reinforcement of the unused movements a lower bound of their pheromone is greater then $q\tau_{max}$. The ACO algorithm for which a pheromone is added on used movements the lower bound of the pheromone value on used movements is greater or equal to τ_0. Thus after additional reinforcement of unused movements and by Theorem 1 from [12] the algorithm will converge to the optimal solution.

Let us consider an ACO algorithm with some elitist ants. In this case only a small number of ants update the pheromone belonging to their solutions. Thus, the big part of pheromone is only evaporated and its value decreases after every iteration. Assuming in the first iteration, the movement from a state i to a state j is unused and the movement from a state i to a state k is used and does not belong to the best solution. The probability to choose the state j and the state k are respectively:

$$p_{ij} = \tau_0\eta_{ij} / \sum_l \tau_{il}\eta_{il} \tag{4}$$

and

$$p_{ik} = \tau_0 \eta_{ik} / \sum_l \tau_{il} \eta_{il} \tag{5}$$

If the movement from i to k is used it means that $\eta_{ij} < \eta_{ik}$. After additional reinforcement, the pheromone level of the movement from i to j will increase and the pheromone level of the movement from i to k will decrease. Thus, after a transition period t_{0i} the probability to choose the movement from i to j will be greater than the probability to choose the movement from i to k. Also the movement from i to k will become unused and will receive additional reinforcement. Therefore, $\rho^{t_0}\tau_0 > 0$ is a lower bound of the pheromone value, where $t_0 = \max t_{0i}$.

Independently of used ACO algorithm, after additional reinforcement of unused movements the lower bound of the pheromone is greater than 0 and the Theorem 1 can be used. Thus, the convergence of ACO-AR algorithm to optimal solution have been proved.

We will estimate the length of the transition period t_{0i}. Let $\eta_j = \min_s(\eta_{is})$ and $\eta_k = \max_s(\eta_{is})$. At first iteration the pheromone level for all movements from arbitrary state i to any other state is equal to τ_0 and therefore the ants choose the state with greater heuristic. After number of iterations t_{0i} the pheromone of movements from the state i to a state with less heuristic information (i.e. unused movement) is:

$$\rho^{t_{0i}}\tau_0 + q\tau_{max}(1 - \rho^{t_{0i}})/(1 - \rho). \tag{6}$$

While the pheromone of the movement from the state i to a state with greater heuristic information (i.e. used movement) is $\rho^{t_{0i}}\tau_0$. From the above discussion it can be seen the used movements become unused if they have less probability as follows:

$$\rho^{t_{0i}}\tau_0\eta_k < \rho^{t_{0i}}\tau_0\eta_j + q\tau_{max}\eta_j(1 - \rho^{t_{oi}})/(1 - \rho) \tag{7}$$

The value of t_{0i} can be calculated from upper inequality.

4 Parameter Value for *q*

In this section we discus the value of the parameter q. Our aim is the diversification and exploration of the search space while keeping the best found solution. Let the movement from a state i to a state k belong to the best solution and the movement from a state i to a state j is unused. The aim is the pheromone level (τ_{ij}) of unused movements to be less than the pheromone level (τ_{ik}) of the movements that belong to the best solution (i.e. $\tau_{ij} \leq \tau_{ik}$). The values of τ_{ij} and τ_{ik} are as follows:

$$\tau_{ij} = \rho^{k_1+k_2}\tau_0 + \frac{1-\rho^{k_2}}{1-\rho}\rho^{k_1}g_1 + \frac{1-\rho^{k_1}}{1-\rho}g(s^*) \tag{8}$$

$$\tau_{ik} = \rho^{k_1+k_2}\tau_0 + \frac{1-\rho^{k_1+k_2}}{1-\rho} g(s^*) \tag{9}$$

where:

- k_1 is the number of iterations for which the movement from i to j belongs to poor solutions;
- k_2 is the number of iterations for which the movement from i to j is unused;
- g_1 is the maximal added pheromone to a movement that belong to poor solution;

From equations (8) and (9) and $0 < g_1 < g(s^*)$ follows that $q \leq \rho$. Evaporation parameter ρ depends of the problem.

5 Conclusion

Recent research has strongly focused on improving the performance of ACO algorithms. In this paper we have presented the ACO-AR algorithm to exploit the search space, which have not been exploited yet, and to avoid premature stagnation of the algorithm. We have shown that ACO-AR algorithm converges to the optimal solution when the algorithm run for a sufficiently large number of iterations. The main idea introduced by ACO-AR, the additional reinforcement of the unused movements, can be apply in a variety ACO algorithms. Our future work will be to apply ACO-AR to other NP-hard COPs and to investigate the search space exploration.

Acknowledgments. Stefka Fidanova was supported by the CONNEX program of the Austrian federal ministry for education, science and culture, and Center of Excellence BIS-21 grant ICA1-2000-70016.

References

1. M. Dorigo and G. Di Caro: The ant colony optimization metaheuristic, in: D. Corne,M. Dorigo and F. Glover,eds., New Idea in Optimization, McGrow-Hill (1999) 11–32.
2. M. Dorigo and L. M. Gambardella: Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1. (1999) 53–66.
3. M. Dorigo, G. Di Caro and Gambardella: Ant algorithms for distributed discrete optimization, Artificial Life 5. (1999) 137–172.
4. M. Dorigo,V. Maniezzo and A. Colorni: The ant system: Optimization by a colony of cooperating agents, IEEE Transaction on Systems, Man. and Cybernetics - Part B 26. (1996) 29–41.
5. S. Fidanova: ACO Algorithm with Additional reinforcement, In:M. Dorigo, G. Di Carro eds., From Ant Colonies to Artificial Ant, Lecture Notes in Computer Science 2542, Springer (2002) 292–293.

6. S. Fidanova: Ant Colony optimization and Pheromone Model, Int. conf. on Real-Life Applications of Metaheuristics, http://www.ruca.ua.ac.be/eume/workshops/reallife/programme.php (2003)
7. M. L. Gambardella, E. D. Taillard and G. Agazzi: A multiple ant colony system for vehicle routing problems with time windows, in: D. Corne, M. Dorigo and F. Glover, eds., New Ideas in Optimization, McGraw-Hill (1999) 63–76.
8. L. M. Gambardella, E. D. Taillard and M. Dorigo: Ant colonies for the QAP, J. of Oper. Res. Soc. 50. (1999) 167–176.
9. I.H. Osman and J. P. Kelley: Methaheuristic:An Overview, In: I. H. Osman and J. P. Kelley eds., Mathematics: Theory and Applications, Kluwer Academic Publishers (1996).
10. R. Y. Rubinstein: Simulation and the Monte Carlo Method John Wiley& Sons. (1981).
11. S. Streltsov and P. Vakili: Variance Reduction Algorithms for Parallel Replicated Simulation of Uniformied Markov Chains, J. of Discrete Event Dynamic Systems: Theory and Applications 6. (1996) 159–180.
12. T. Stützle and M. Dorigo: A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms, IEEE Transactions on Evolutionary Computation 6(4). (2002) 358–365.