
Modeling the Dynamics of Ant Colony Optimization

Daniel Merkle

merkle@aifb.uni-karlsruhe.de

Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany

Martin Middendorf

martin.middendorf@ku-eichstaett.de

Computer Science Group, Catholic University of Eichstätt-Ingolstadt, D-85072 Eichstätt, Germany

Abstract

The dynamics of Ant Colony Optimization (ACO) algorithms is studied using a deterministic model that assumes an average expected behavior of the algorithms. The ACO optimization metaheuristic is an iterative approach, where in every iteration, artificial ants construct solutions randomly but guided by pheromone information stemming from former ants that found good solutions. The behavior of ACO algorithms and the ACO model are analyzed for certain types of permutation problems. It is shown analytically that the decisions of an ant are influenced in an intriguing way by the use of the pheromone information and the properties of the pheromone matrix. This explains why ACO algorithms can show a complex dynamic behavior even when there is only one ant per iteration and no competition occurs. The ACO model is used to describe the algorithm behavior as a combination of situations with different degrees of competition between the ants. This helps to better understand the dynamics of the algorithm when there are several ants per iteration as is always the case when using ACO algorithms for optimization. Simulations are done to compare the behavior of the ACO model with the ACO algorithm. Results show that the deterministic model describes essential features of the dynamics of ACO algorithms quite accurately, while other aspects of the algorithms behavior cannot be found in the model.

Keywords

Ant colony optimization, ACO algorithms, ACO model, dynamic behavior, fixed points.

1 Introduction

Ant Colony Optimization (ACO) is a metaheuristic that has been applied successfully to several optimization problems (for an overview, see Dorigo and Di Caro (1999); the first ACO algorithm was proposed by Dorigo (1992) and Dorigo et al. (1996)). Since ACO algorithms are based on sequences of random decisions of artificial ants that are usually not independent, it is difficult to analyze the behavior of ACO algorithms theoretically. Except for convergence proofs for types of ACO algorithms with a strong elite principle (Gutjahr, 2000, 2002; Stützle and Dorigo, 2000), not much theoretical work has been done. Usually ant algorithms have been tested on benchmark problems or real-world problems. In this paper, we propose and analyze a deterministic model for ACO algorithms that allows us to derive numerically exact results on optimization problems with a simple structure. The analytical results are complemented with empirical studies that compare computations done with the ACO model with test runs of the ACO algorithm.

Modeling has been done in the field of *genetic algorithms* (GAs) by several authors in order to better understand GA behavior. One line of modeling is to use an infinite population, which is often easier to handle than a finite population, since important properties of an infinite population do not fluctuate due to a finite number of random events (Vose and Liepins, 1991; Vose, 1999). Each possible gene string of a GA can then be characterized by its fraction in the population instead of the total number of its occurrences. Another method for modeling is to characterize the population by a few parameters that capture only the important aspects of the population instead of dealing with a concrete population. Examples for such parameters are the mean and the variance of the fitness distribution in the population or the percentage of individuals with a given fitness value. This approach was used by Prügel-Bennett and Rogers (2001), Prügel-Bennett and Shapiro (1994), and van Nimwegen et al. (1999). It was motivated by ideas from statistical mechanics. The expected behavior of a population was used by Bruce and Simpson (1999) to model GAs. Mostly in these studies, GAs are modeled on simple problems that have some characteristic features of more complicated and real-life problems. Examples are the Royal Road functions, where the number of ones in a gene string has to be maximized in a certain way (Mitchell and Forrest, 1997).

The modeling approach used in this paper is to define a deterministic model for ACO algorithms that is based on the expected decisions of the ants and does not fluctuate due to the outcome of a few random decisions. We are especially interested in the changes of the pheromone values during the run of an ACO algorithm. In the proposed model, the pheromone update in every iteration is done by adding for each pheromone value the expected update for a random generation of ants. This means that the effect of an individual ant in a run is averaged out. The behavior of the model in one iteration is the same as when an infinite number of iterations of the algorithm is run independently using the same pheromone information, and afterwards, the pheromone update is done as the average of all ants that are allowed to update in all iterations.

The paper is structured as follows. In Section 2, we define the permutation problems that are used in this paper. The ACO algorithm that is modeled is described in Section 3. The deterministic ACO model is defined in Section 4. In Section 5, we discuss how the model can be applied to permutation problems that consist of independent subproblems. A fixed point analysis of pheromone matrices that helps to explain the behavior of the ACO model and ACO algorithms is done in Section 6. In Section 7, we analyze the dynamic behavior of the ACO model. Simulation results are described in Section 8 and used to compare the behavior of the ACO model and ACO algorithms. Conclusions are given in Section 9.

2 Permutation Problems

Although the general approach of our ACO model does not depend on a specific type of optimization problem, we offer a more elaborate description of the ACO model for permutation problems. Such problems are also used as test problems for the ACO model and the ACO algorithms.

In particular, we use the following type of permutation problem known as the *linear assignment* problem. Given are n items $1, 2, \dots, n$ and an $n \times n$ cost matrix $C = [c(i, j)]$ with integer costs $c(i, j) \geq 0$. Let \mathcal{P}_n be the set of permutations of $(1, 2, \dots, n)$. For a permutation $\pi \in \mathcal{P}_n$, let $c(\pi) = \sum_{i=1}^n c(i, \pi(i))$ be the cost of the permutation. Let $\mathcal{C} := \{c(\pi) \mid \pi \in \mathcal{P}_n\}$ be the set of possible values of the cost function. The problem is to find a permutation $\pi \in \mathcal{P}_n$ of the n items that has minimal costs, i.e., a permutation with $c(\pi) = \min\{c(\pi') \mid \pi' \in \mathcal{P}_n\}$.

It should be mentioned that the Linear Assignment problem is not an NP-hard problem but can be solved efficiently, for example, with the Hungarian method (see for example, Papadimitriou and Steiglitz (1982)).

3 The ACO Algorithm

An ACO algorithm consists of several iterations where in every iteration each of m ants constructs a solution for the optimization problem. In this section, we describe a basic ACO algorithm for permutation problems. The literature on ACO algorithms is growing, and we cannot consider here all the variants and improvements that have been proposed in recent years.

For the construction of a solution (i.e., a permutation), every ant selects the items in the permutation one after the other. For the selection of an item, the ant uses pheromone information that stems from former ants that have found good solutions. In addition, an ant may also use heuristic information for its decisions. The pheromone information τ_{ij} and heuristic information η_{ij} indicate the advantage of having item j at location i of the permutation. The heuristic values are problem specific. Since we want to study ACO algorithms in general and not for some specific problem, we omit the heuristic values in the rest of this paper. The matrix $(\tau_{ij})_{i,j \in [1:n]}$ of pheromone values is called the *pheromone matrix*.

The next item is chosen by an ant from the set of items not yet placed \mathcal{S} according to the following probability distribution (see for example, Dorigo et al. (1996)) that depends on the pheromone values in row i of the pheromone matrix:

$$p_{ij} = \frac{\tau_{ij}}{\sum_{h \in \mathcal{S}} \tau_{ih}} \quad (\text{local evaluation}) \quad (1)$$

It is a characteristic feature of the original ACO metaheuristic (Dorigo and Di Caro, 1999) that the decisions of the ants are done locally (like real ants) and consider only the pheromone values τ_{ij} for $j \in \mathcal{S}$. Therefore we call this method *local evaluation* (of the pheromone information). Merkle and Middendorf (2000) showed that it is profitable for some problems when ants make decisions that do not consider only the local pheromone values. Different such global evaluation methods have been successfully used to solve scheduling problems with ACO algorithms (Merkle and Middendorf, 2000, 2001b). One of these methods, called *summation evaluation*, performs particularly well for solving permutation problems where the good solutions (permutations) satisfy the following similarity property. For every two good permutations, it holds that for every item, its places in both permutations differ little. For summation evaluation, the probability distribution to decide which item to put on place i is based on the sums of the pheromone values in each column of the pheromone matrix up to row i , i.e., the values $\sum_{k=1}^i \tau_{kj}$, $j \in \mathcal{S}$. The probabilities are defined by

$$p_{ij} = \frac{\sum_{k=1}^i \tau_{kj}}{\sum_{h \in \mathcal{S}} (\sum_{k=1}^i \tau_{kh})} \quad (\text{summation evaluation}) \quad (2)$$

It was shown by Merkle and Middendorf (2000) that the summation method gives better results than the standard method for the Single Machine Weighted Total Tardiness problem. Meanwhile, it has been shown that the summation method is also useful for the Job Shop problem (Teich et al., 2001) and the Flow Shop problem (Rajendran and Ziegler, 2001). For the Resource-Constrained Project Scheduling problem, the summation method performed better than the standard method, but a combination of both

methods was best (Merkle et al., 2002).

Before the pheromone update is done, a percentage of the old pheromone is evaporated according to the formula $\tau_{ij} = (1 - \rho) \cdot \tau_{ij}$. Parameter ρ , $0 < \rho < 1$ allows us to determine how strongly old pheromone influences future decisions. Then, for every item j of the best permutation found in this iteration, some amount of pheromone Δ is added to element τ_{ij} of the pheromone matrix where i is the place of item j . When there are several best solutions found in the iteration, one of them is selected randomly for pheromone update. The algorithm stops when some stopping criterion is met, for example, a certain number of generations has been done. Pseudocode 1 shows the ACO algorithm.

For ease of description, we assume that the sum of the pheromone values in every row and every column of the matrix is always one, i.e., $\sum_{i=1}^n \tau_{ij} = 1$ for $j \in [1 : n]$ and $\sum_{j=1}^n \tau_{ij} = 1$ for $i \in [1 : n]$ and $\Delta = \rho$. Observe that for permutation problems, this property of a pheromone matrix will not be destroyed when pheromone update is applied. We also assume that all pheromone values are initialized with a positive value. This implies that pheromone values never become zero during a run of the algorithm.

Pseudocode 1 ACO algorithm

```

repeat
  for ant  $k \in \{1, \dots, m\}$  do
     $S \leftarrow \{1, 2, \dots, n\}$  {set of available items}
    for  $i = 1$  to  $n$  do
      choose item  $j \in S$  with probability  $p_{ij}$  (according to Equation 1 or 2)
       $S \leftarrow S - \{j\}$ 
    end for
  end for
  for all  $(i, j)$  do
     $\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$  {evaporate pheromone}
  end for
  for all  $(i, j) \in$  best solution (of the  $m$  solutions of this iteration) do
    /* when there are more than one best solution one is selected randomly
     $\tau_{ij} \leftarrow \tau_{ij} + \Delta$  {update pheromone}
  end for
until stopping criterion is met

```

4 ACO Model

In order to investigate the behavior of ACO algorithms analytically, we propose a deterministic model for ACO. We are especially interested in the changes of the pheromone values that occur during the run of an ACO algorithm. In the proposed model, the pheromone update of a generation of ants is done by adding for each value in the pheromone matrix the expected update for that iteration. This means that the effect of decisions of an individual ant in a run of the algorithm are averaged out in the model. The behavior of the model in one iteration is the same as when an infinite number of iterations of the algorithm is run independently using the same pheromone information, and afterwards, the update is done as the average of all ants that are allowed to update. Similar ideas have led to the study of GAs with an infinite population (for example, Vose and Liepins (1991) and Vose (1999)). Such infinite population models of

GAs allow us to overcome the difficulties that arise from the sampling bias of finite populations.

Since the update values in the ACO algorithm are always only zero or $\Delta = \rho$, the ACO model does not exactly match the expected behavior of an ACO algorithm, on the average, over more than one generation.

In order to determine the expected update for a generation of random ants, the probabilities for the various decisions by the ants have to be determined. Let $M = (\tau_{ij})$ be a pheromone matrix. Then the probability that a random ant selects item j for place i can be computed as described in the following. Let \mathcal{P}_n be the set of possible solutions, i.e., the set of permutations of $(1, 2, \dots, n)$. The probability σ_π that a random ant selects a solution $\pi \in \mathcal{P}_n$ is

$$\sigma_\pi = \prod_{i=1}^n \frac{\tau_{i,\pi(i)}}{\sum_{j=i}^n \tau_{i,\pi(j)}} \quad (\text{when using local evaluation}) \quad (3)$$

$$\sigma_\pi = \prod_{i=1}^n \frac{\sum_{k=1}^i \tau_{k,\pi(i)}}{\sum_{j=i}^n \sum_{k=1}^i \tau_{k,\pi(j)}} \quad (\text{when using summation evaluation}) \quad (4)$$

Unless mentioned otherwise, we use local evaluation in this paper. The probability σ_{ij} that item j is put on place i by a random ant is

$$\sigma_{ij} = \sum_{\pi \in \mathcal{P}_n} g(\pi, i, j) \cdot \sigma_\pi \quad (5)$$

where

$$g(\pi, i, j) = \begin{cases} 1 & \text{if } \pi(i) = j, \\ 0 & \text{otherwise} \end{cases}$$

In order to model a generation of m ants, the expected behavior of the best of m ants has to be determined. Given a permutation problem P with a corresponding cost matrix and a pheromone matrix, let $\sigma_\pi^{(m)}$ be the probability that the best of m ants has found solution $\pi \in \mathcal{P}_n$ (if the best found quality was found by several ants, one of these ants is chosen with uniform probability to be the best one). Let $\sigma_{ij}^{(m)}$ be the probability that the best of m ants in a generation selects item j for place i . Let $\mathcal{P}_{\min}(P, \pi_1, \dots, \pi_m)$ be the multiset of permutations in $\{\pi_1, \dots, \pi_m\}$ that have lowest costs for problem P , i.e., $\mathcal{P}_{\min}(P, \pi_1, \dots, \pi_m) = \{\pi_i, i \in [1 : m] \mid c(\pi_i) = \min\{c(\pi_j) \mid j \in [1 : m]\}\}$. Note that in the following, $\mathcal{P}_{\min}(P, \pi_1, \dots, \pi_m)$ and its subsets are multisets that may contain some permutations more than once. Probability $\sigma_\pi^{(m)}$ can be computed as a sum over all possible choices (π_1, \dots, π_m) of m ants as follows

$$\sigma_\pi^{(m)} = \sum_{(\pi_1, \dots, \pi_m), \pi_i \in \mathcal{P}_n} \frac{|\{\pi' \in \mathcal{P}_{\min}(P, \pi_1, \dots, \pi_m) \mid \pi' = \pi\}|}{|\mathcal{P}_{\min}(P, \pi_1, \dots, \pi_m)|} \cdot \prod_{k=1}^m \sigma_{\pi_k} \quad (6)$$

where the fraction equals the portion of permutations in the best solutions in (π_1, \dots, π_m) that equal π , and $\prod_{k=1}^m \sigma_{\pi_k}$ is the probability that the ants have chosen (π_1, \dots, π_m) . Similarly, probability $\sigma_{ij}^{(m)}$ can be computed as

$$\sigma_{ij}^{(m)} = \sum_{(\pi_1, \dots, \pi_m), \pi_i \in \mathcal{P}_n} \frac{|\{\pi \in \mathcal{P}_{min}(P, \pi_1, \dots, \pi_m) \mid \pi(i) = j\}|}{|\mathcal{P}_{min}(P, \pi_1, \dots, \pi_m)|} \cdot \prod_{k=1}^m \sigma_{\pi_k} \quad (7)$$

At the end of an iteration that simulates one iteration with m ants of the algorithm, the pheromone update is done in the ACO model as follows (see Pseudocode 2):

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \sigma_{ij}^{(m)}$$

The sum of the pheromone values in every row and every column of the pheromone matrix is one. Same as for the algorithm, this property will not be destroyed when pheromone update is applied.

Pseudocode 2 ACO model

```

repeat
  for all (i,j) do
     $\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$  {evaporate pheromone}
     $\tau_{ij} \leftarrow \tau_{ij} + \rho \cdot \sigma_{ij}^{(m)}$  {update pheromone}
  end for
until stopping criterion is met

```

In the following, an alternative way to compute the selection probabilities for the best of m ants is described. It gives a better understanding of the behavior of ACO algorithms by considering the probability of selecting solutions with different qualities. Let \mathcal{C} be the set of possible cost values for a permutation, or in other words, the set of possible solution qualities. Let $\xi_x^{(m)}$ be the probability that the best of m ants in a generation finds a solution with quality $x \in \mathcal{C}$. (Recall that in the case where the best found quality was found by several ants, one of these ants is chosen with uniform probability to be the best one.) Let $\omega_{ij}^{(x)}$ be the probability that an ant that has found a solution with quality $x \in \mathcal{C}$ has selected item j for place i . Then

$$\sigma_{ij}^{(m)} = \sum_{x \in \mathcal{C}} \xi_x^{(m)} \cdot \omega_{ij}^{(x)} \quad (8)$$

where

$$\xi_x^{(m)} = \sum_{\pi \in \mathcal{P}_n, c(\pi)=x} \sigma_{\pi}^{(m)}$$

and

$$\omega_{ij}^{(x)} = \sum_{\pi \in \mathcal{P}_n, c(\pi)=x} g(\pi, i, j) \cdot \frac{\sigma_{\pi}}{\sum_{\pi' \in \mathcal{P}_n, c(\pi')=x} \sigma_{\pi'}}$$

An interesting aspect of Equation (8) is that the pheromone update that is performed at the end of an iteration is obtained as a weighted sum over the possible solution qualities. For each possible solution quality, the update value is determined by the probabilities for the decisions of a single ant when it chooses between all possible solutions with that same quality. The effect of the number of ants m is only that the weights $\xi_x^{(m)}$ of the different qualities in this sum changes. Since for each possible solution the probability to select it is never zero, it is clear that the more ants in an iteration, the higher the weight of the optimal quality.

5 Simple ACO Model for Restricted Permutation Problems

Many real-world problems consist of subproblems that are more or less independent from each other. In order to study the behavior of ACO algorithms on such problems, this situation is modeled in an idealized way. We consider restricted permutation problems that consist of several independent smaller problems. In practice, when the subproblems can be easily identified and it is known that they are independent, it is usually better to solve them independently. But it often might be difficult to identify the subproblems, or they may not be completely independent.

Define for a small permutation problem P of size n_0 a restricted permutation problem P^q that consists of q independent instances of P . As an example, consider the following problem P_1 with cost matrix

$$C_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \quad (9)$$

This example problem can be seen as a simple instance of a restricted single machine total lateness scheduling problem, where a set of jobs all with the same processing time and given due dates has to be scheduled on a single machine such that the sum of the deviations from the due dates is minimal. In the example, there are three jobs, each has processing time one, and job i has due date i , $i \in [1 : 3]$. The corresponding permutation problem P_1^2 has the following cost matrix

$$C_1^{(2)} = \begin{pmatrix} 0 & 1 & 2 & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty & \infty \\ 2 & 1 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & 1 & 2 \\ \infty & \infty & \infty & 1 & 0 & 1 \\ \infty & \infty & \infty & 2 & 1 & 0 \end{pmatrix} \quad (10)$$

Formally, let C_1, C_2, \dots, C_q be the cost matrices of q instances of P , and denote by $c_{ij}^{(r)}$ element (i, j) of matrix C_r , $r \in [1 : q]$. Let $C^{(q)} = (c_{ij})_{i,j \in [1:q \cdot n_0]}$ be the corresponding cost matrix of the instance of problem P^q where $c_{(r-1) \cdot n_0 + i, (r-1) \cdot n_0 + j} = c_{ij}^{(r)}$ for $i, j \in [1 : n_0]$, $r \in [1 : q]$ and $c_{ij} = \infty$ otherwise. Note, that our definition of restricted permutation problems does not allow an ant to make a decision with cost ∞ . This is accomplished by initializing all pheromone values with zero that correspond to ∞ -elements in the cost matrix. We call P the elementary subproblem of P^q and the q instances of P that form an instance of P^q the elementary subinstances of P^q . When all cost matrices C_1, C_2, \dots, C_q are equal, i.e., $C = C_1 = C_2 = \dots = C_q$ for some cost matrix C , we call P^q a homogeneous restricted permutation problem and denote the cost matrix of P^q by $C^{(q)}$. Otherwise it is called an *inhomogeneous restricted permutation problem*.

In the following, we show how the behavior of the ACO algorithm for a (possibly inhomogeneous) restricted permutation problem P^q can be approximated using the behavior of the ACO model for the elementary subproblem P . An example of this material and the material in the next subsection is elaborated in Subsection 5.2.

Consider an arbitrary instance of the q elementary subinstances of problem P^q — say the r th subinstance — and the quality of solutions that m ants in an iteration have found on the other elementary subinstances (which form an instance of problem P^{q-1}). Clearly, ants that have found better solutions on subproblem P^{q-1} have better

chances of ending up with the best found solution on problem P^q . Since the decisions of an ant on different subinstances are independent, the ant algorithm for problem P^q is modeled only on the elementary subproblem. But to model the pheromone update, the solution quality that an ant found on all other subinstances has to be considered. To model this situation for the elementary subinstance, we assume that some ants in an iteration — those ants that did not find the best solution on P^{q-1} in that iteration — receive a malus (a penalty value) that is added to the cost of the permutation they find on the elementary subinstance. An ant with a malus is allowed to update only when the cost of its solution plus the malus is better than the solution of every other ant plus its malus (if it has a malus). Formally, for $i \in [1 : m]$, let $d_i \geq 0$ be the malus of ant i . We always assume without loss of generality that ant 1 has no malus, i.e. $d_1 = 0$. The notation from Section 4 is now extended to include ants that have a malus. Let $\xi_x^{(m;d_2,\dots,d_m)}$ be the probability that the best of m ants, where ant $i \in [2 : m]$ has a malus d_i , has found a solution of quality $x \in \mathcal{C}$. For this case, the probability that the best of m ants selects item j for place i is

$$\sigma_{ij}^{(m;d_2,\dots,d_m)} = \sum_{x \in \mathcal{C}} \xi_x^{(m;d_2,\dots,d_m)} \cdot \omega_{ij}^{(x)} \quad (11)$$

Without loss of generality, assume that the quality of solution found by ant i on problem P^{q-1} is at least as good as the solution found by ant $i+1$, $i \in [1 : m-1]$. The assumption implies $0 \leq d_2 \leq \dots \leq d_m$. Let d_{max} be the maximum difference between two solutions on the r th subproblem. Formally, let \mathcal{C}_r be the set of all solution qualities of the r th subproblem and $d_{max} := \max\{x \mid x \in \mathcal{C}_r\} - \min\{x \mid x \in \mathcal{C}_r\}$. Clearly, an ant that has found a solution on P^{q-1} that is worse by more than d_{max} compared to the best ant on P^{q-1} has no chance to end up with the best solution on problem P^q and to update the pheromone information. Therefore, we do not consider malus values greater than $d_{max} + 1$. Formally, let d_i , $i \in [2 : m]$ be the minimum of $d_{max} + 1$ and the difference of the cost of the permutation found by ant i on P^{q-1} and the cost of the permutation found by ant 1 on P^{q-1} . Define $\phi_{d_2,\dots,d_m}^{(m)}$, $0 \leq d_i \leq d_{max} + 1$, $i \in [2 : m]$ as the probability that for m ants on problem P^{q-1} , the difference of the costs of the solutions found by the i th best ant and the best ant is d_i when $d_i \leq d_{max}$, and when $d_i = d_{max} + 1$, it is the probability that this difference is $> d_{max}$, $i \in [2 : m]$. Let \mathcal{D} be the set of all possible malus vectors (d_2, \dots, d_m) with integers $d_2 \leq \dots \leq d_m$, $0 \leq d_i \leq d_{max} + 1$, $i \in [2 : m]$. Then for the r th elementary subproblem of P^q we obtain

$$\begin{aligned} \sigma_{ij}^{(m)} &= \sum_{(d_2,\dots,d_m) \in \mathcal{D}} \phi_{d_2,\dots,d_m}^{(m)} \cdot \sigma_{ij}^{(m;d_2,\dots,d_m)} \\ &= \sum_{(d_2,\dots,d_m) \in \mathcal{D}} \phi_{d_2,\dots,d_m}^{(m)} \cdot \sum_{x \in \mathcal{C}} \xi_x^{(m;d_2,\dots,d_m)} \cdot \omega_{ij}^{(x)} \\ &= \sum_{x \in \mathcal{C}} w_x \cdot \omega_{ij}^{(x)} \end{aligned} \quad (12)$$

with weight $w_x = \sum_{(d_2,\dots,d_m) \in \mathcal{D}} \phi_{d_2,\dots,d_m}^{(m)} \cdot \xi_x^{(m;d_2,\dots,d_m)}$. This result is interesting because it shows that the effect of the subproblem P^{q-1} on the remaining elementary subinstance P_r of P^q is to change the weights between the influence of the different solution quality levels when compared to Equation (8) for solving only the elementary subproblem $P = P_r$.

We consider the case of $m = 2$ ants in more detail. The probability that the best of two ants selects item j of the r th elementary subproblem, i.e., item $(r - 1)n_0 + j$, $r \in [1 : q]$, $j \in [1 : n_0]$ at place $(r - 1)n_0 + i$, $i \in [1 : n_0]$ for problem P^q can be computed as follows

$$\sigma_{(r-1)n_0+i, (r-1)n_0+j}^{(2)} = \phi_{>d_{max}}^{(2)} \cdot \sigma'_{ij} + \sum_{d=1}^{d_{max}} \phi_d^{(2)} \cdot \sigma_{ij}^{\prime(2;d)} + \phi_0^{(2)} \cdot \sigma_{ij}^{\prime(2)} \quad (13)$$

where σ' refers to probabilities for the elementary subinstance P . The equation shows the interesting fact that part of the probability $\sigma_{(r-1)n_0+i, (r-1)n_0+j}^{(2)}$ to select the item is just the probability σ'_{ij} of a single ant to select item j at place i for the elementary subproblem P . This is the case when the quality of the solutions of both ants differ by more than d_{max} . When the qualities of both solutions are the same, the probability $\sigma_{ij}^{\prime(2)}$ to select item j at place i equals the probability that the better of two ants on problem P selects item j at place i . All other cases correspond to the situation where one ant on problem P has a malus.

Since decisions of an ant on different elementary subproblems are independent, it follows that the larger the number q of subproblems, the larger the probability $\phi_{>d_{max}}^{(2)}$. An important consequence is that the (positive) effect of competition between the two ants for finding good solutions becomes weaker, and a possible bias in the decisions of a single ant has more influence.

5.1 The Homogeneous Case

Assume in this subsection that the restricted permutation problem is homogeneous, i.e., the submatrices of the cost matrix of problem P^q , which correspond to the elementary subinstances, are all equal.

The main difficulty in computing probability $\sigma_{(r-1)n_0+i, (r-1)n_0+j}^{(m)}$ is computing the probability $\phi_{d_2, \dots, d_m}^{(m)}$ that the malus situation (d_2, \dots, d_m) occurs. It can be done using homogeneous Markov chains, where the state space is defined as the set of all tuples (d_2, \dots, d_m) . One step in the Markov chain corresponds to the evaluation of one elementary subproblem for each of the m ants. The transition matrix defines all probabilities to change from one malus situation to another by using the m outcomes on the elementary subproblem. For example, the probability of change within one step from state $(d_2, \dots, d_m) = (0, 0, \dots, 0)$ to the same state again is exactly the probability that all of the ants found a solution with the same quality on the subproblem. For the example problem P_1 , this probability is $(\xi_0^{\prime(1)})^m + (\xi_2^{\prime(1)})^m + (\xi_4^{\prime(1)})^m$, if $\xi_x^{\prime(1)}$ is the probability that a single ant finds a solution with quality x on problem P . The probability to be in a given state after $q - 1$ steps can be calculated by a recursive function that is recursive in $m - 1$ variables and has $q - 1$ recursion levels. For small systems with a few ants, the calculation can be done in a few seconds (for example, for $m = 3$ and $q = 64$, it takes less than 0.3 seconds on a PC with a 500 MHz Pentium III processor).

For the case of $m = 2$ ants, the computation of $\phi_d^{(2)}$ can be done easily in an alternative way that is described in the rest of this subsection. Let ψ_d (ψ'_d) be the probability that the difference of the solution quality found by the first ant minus the solution quality found by the second ant on subproblem P^{q-1} (respectively P) is d (here we do not assume that the first ant finds the better solution). The value of this difference on subproblem P^{q-1} can be described as the result of a generalized, one-dimensional, random

walk of length $q - 1$. Define for $d \in [-d_{max} : d_{max}]$

$$I_d = \{(k_{-d_{max}}, k_{-d_{max}+1}, \dots, k_{d_{max}}) \mid q - 1 = \sum_{i=-d_{max}}^{d_{max}} k_i, d = \sum_{i=-d_{max}}^{d_{max}} i \cdot k_i\} \quad (14)$$

where k_i is the number of elementary subinstances of P^{q-1} , where the difference between the first and the second ant is $i \in [-d_{max} : d_{max}]$. Then ψ_d can be computed as follows

$$\psi_d = \sum_{(k_{-d_{max}}, \dots, k_{d_{max}}) \in I_d} \frac{(q-1)!}{k_{-d_{max}}! \cdot \dots \cdot k_{d_{max}}!} \cdot (\psi'_{-d_{max}})^{k_{-d_{max}}} \cdot \dots \cdot (\psi'_{d_{max}})^{k_{d_{max}}} \quad (15)$$

Clearly, $\phi_0^{(2)} = \psi_0$, and due to symmetry, $\phi_d^{(2)} = 2 \cdot \psi_d$ for $d \neq 0$.

5.2 An Example

As an example for the ACO model introduced in this section, consider an elementary problem P_1 with cost matrix given in (9). The possible solution qualities for problem P_1 are 0, 2, and 4, and the optimal solution is to put item i on place i for $i \in [1 : 3]$. Consider the following pheromone matrix for P_1

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.6 & 0.1 & 0.3 \\ 0.3 & 0.6 & 0.1 \end{pmatrix} \quad (16)$$

Then the probability for a random ant to put, for example, item 2 on place 2 can be computed according to (5) as $\sigma_{22} = \sigma_{\pi_1} + \sigma_{\pi_2} = 0.1 \cdot 0.1 / (0.1 + 0.3) + 0.6 \cdot 0.1 / (0.1 + 0.6) \approx 0.111$ with permutations $\pi_1 = (1, 2, 3)$ and $\pi_2 = (3, 2, 1)$. The matrix of selection probabilities for one ant on problem P_1 is

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \approx \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.714 & 0.111 & 0.175 \\ 0.186 & 0.589 & 0.225 \end{pmatrix}$$

Since the optimal solution is to place item i on place i for $i \in [1 : 3]$, it might be conjectured that the corresponding selection probabilities become larger with competition between two ants in an iteration compared to the case of a single ant in an iteration. This is true for the probability that the better of two ants puts item 1 on place 1 and item 3 on place 3. But our example shows that this conjecture does not hold in general. In the example, the probability to place item 2 on place 2 is $\sigma_{22}^{(2)} = 0.109$ (computed according to (8)) and slightly smaller than $\sigma_{22} = 0.111$. The matrix with the selection probabilities for the better of two ants on problem P_1 is

$$\begin{pmatrix} \sigma_{11}^{(2)} & \sigma_{12}^{(2)} & \sigma_{13}^{(2)} \\ \sigma_{21}^{(2)} & \sigma_{22}^{(2)} & \sigma_{23}^{(2)} \\ \sigma_{31}^{(2)} & \sigma_{32}^{(2)} & \sigma_{33}^{(2)} \end{pmatrix} \approx \begin{pmatrix} 0.175 & 0.405 & 0.420 \\ 0.695 & 0.109 & 0.196 \\ 0.130 & 0.486 & 0.384 \end{pmatrix} \quad (17)$$

When one of two ants has a malus, the selection probabilities are mostly in between the case of two ants per iteration and a single per iteration. But again, the probability to place item 2 on place 2 is a counterexample. Computed according to (11) the value of $\sigma_{22}^{(2;2)}$ is 0.118, and thus $\sigma_{22}^{(2)} < \sigma_{22} < \sigma_{22}^{(2;2)}$. When one ant has a malus of 2, the matrix of the selection probabilities of the better ant on problem P_1 is

$$\begin{pmatrix} \sigma_{11}^{(2;2)} & \sigma_{12}^{(2;2)} & \sigma_{13}^{(2;2)} \\ \sigma_{21}^{(2;2)} & \sigma_{22}^{(2;2)} & \sigma_{23}^{(2;2)} \\ \sigma_{31}^{(2;2)} & \sigma_{32}^{(2;2)} & \sigma_{33}^{(2;2)} \end{pmatrix} \approx \begin{pmatrix} 0.146 & 0.351 & 0.503 \\ 0.698 & 0.118 & 0.184 \\ 0.156 & 0.531 & 0.313 \end{pmatrix}$$

When one ant has a malus of 4, the matrix of the selection probabilities of the better ant on problem P_1 is

$$\begin{pmatrix} \sigma_{11}^{(2;4)} & \sigma_{12}^{(2;4)} & \sigma_{13}^{(2;4)} \\ \sigma_{21}^{(2;4)} & \sigma_{22}^{(2;4)} & \sigma_{23}^{(2;4)} \\ \sigma_{31}^{(2;4)} & \sigma_{32}^{(2;4)} & \sigma_{33}^{(2;4)} \end{pmatrix} \approx \begin{pmatrix} 0.109 & 0.299 & 0.593 \\ 0.708 & 0.118 & 0.174 \\ 0.183 & 0.583 & 0.234 \end{pmatrix}$$

Although not true in every single case, it can be observed that the selection probabilities for the better ant become more similar to the matrix of the selection probabilities for a single ant the higher the malus is.

The probability that one ant finds a solution with cost x for the elementary problem P_1 can be derived using (3) and summing over σ_π for all permutations with $c(\pi) = x$. For $x \in 0, 2, 4$ this gives for problem P_1 : $\xi_0^{(1)} = 0.1 \cdot (0.1/0.4) = 0.025$, $\xi_2^{(1)} = 0.3 \cdot (0.6/0.9) + 0.1 \cdot (0.3/0.4) = 0.275$, and $\xi_4^{(1)} = 0.6 \cdot (0.6/0.7) + 0.6 \cdot (0.1/0.7) + 0.3 \cdot (0.3/0.9) = 0.7$, respectively.

Consider the homogeneous problem P_1^3 . For elementary subproblem P_1 , the possible differences between the quality of two solutions are $-4, -2, 0, 2, 4$ so that $d_{max} = 4$. Hence according to (13), for $r \in [1 : 3]$,

$$\sigma_{3(r-1)+i, 3(r-1)+j}^{(2)} = \phi_{>4}^{(2)} \cdot \sigma'_{ij} + \sum_{d=2,4} \phi_d^{(2)} \cdot \sigma'_{ij}^{(2;d)} + \phi_0^{(2)} \cdot \sigma'_{ij}^{(2)}$$

where σ' corresponds to the subproblem P_1 . According to Equation (15), the corresponding probabilities that the solutions of two random ants on P_1 differ by $-4, -2, 0, 2, 4$ are $\psi_0 \approx 0.566$, $\psi_{-2} = \psi_2 \approx 0.199$, $\psi_{-4} = \psi_4 \approx 0.018$ with the set

$$I_0 = \{(0, 0, 0, 0, 2, 0, 0, 0, 0), (0, 0, 1, 0, 0, 0, 1, 0, 0), (1, 0, 0, 0, 0, 0, 0, 0, 1)\}$$

and the sets I_2 and I_4 are defined analogously (see (14)). The probability that two ants find a solution with the same quality for $P_1^{(2)}$ is

$$\phi_0^{(2)} = \psi_0 = (\psi'_0)^2 + \psi'_{-2}\psi'_2 + \psi'_2\psi'_{-2} + \psi'_{-4}\psi'_4 + \psi'_4\psi'_{-4} \approx 0.401$$

and similarly, $\phi_2^{(2)} = 2\psi_2 \approx 0.466$, $\phi_4^{(2)} = 2\psi_4 \approx 0.119$, and $\phi_{d>4}^{(2)} = 2\psi_{>4} \approx 0.015$. Thus for $r \in [0 : 2]$,

$$\begin{pmatrix} \sigma_{3r+1, 3r+1}^{(2)} & \sigma_{3r+1, 3r+2}^{(2)} & \sigma_{3r+1, 3r+3}^{(2)} \\ \sigma_{3r+2, 3r+1}^{(2)} & \sigma_{3r+2, 3r+2}^{(2)} & \sigma_{3r+2, 3r+3}^{(2)} \\ \sigma_{3r+3, 3r+1}^{(2)} & \sigma_{3r+3, 3r+2}^{(2)} & \sigma_{3r+3, 3r+3}^{(2)} \end{pmatrix} \approx \begin{pmatrix} 0.152 & 0.366 & 0.482 \\ 0.699 & 0.114 & 0.187 \\ 0.149 & 0.520 & 0.331 \end{pmatrix}$$

Then the probability $\xi_x^{(2)}$ that the better of two ants finds a solution with cost $x = 0$ for one elementary subproblem P_1 of P_1^3 is $\xi_0^{(2)} \approx 0.152 \cdot 0.114 / (0.114 + 0.187) \approx 0.058$. This is smaller than the corresponding probability that the better of two ants finds a solution with cost 0 on problem P_1 alone, which is $0.175 \cdot 0.109 / (0.109 + 0.196) \approx 0.063$, and can be computed from the matrix in (17).

6 Fixed Points

Since the pheromone values of an ACO algorithm reflect the frequencies of decisions that resulted in good solutions, it is a desirable property of the algorithm that the selection probabilities used by an ant are equal to their corresponding pheromone values. (Recall that we do not use heuristic values for the decisions of an ant and assume that the pheromone values in every row of the matrix are one and can be interpreted as probabilities.) As observed by Merkle and Middendorf (2001a, 2001b), this property might often be violated since the decisions of an ant for constructing a solution are not independent. We say that there is a selection bias when the probability of a random ant to choose an item is different from the corresponding pheromone value (i.e., from the proportion of pheromone it has with respect to the total pheromone in the row). In this section, we analyze the situation in more detail.

A pheromone matrix with the property that the probability of a random ant to choose an item is the same as the corresponding pheromone value is called a *selection fixed point matrix*. A selection fixed point matrix will not change in the ACO model when using only $m = 1$ ant per iteration. It can only change when there are $m \geq 2$ ants in an iteration. This is a desirable property because it means that the change of the matrix is driven by competition and is not due to a bias in the decisions of a single ant. The question arises: Which matrices are selection fixed point matrices for a permutation problem, and how many fixed point matrices exist? As an example, we consider a permutation problem of size $n = 3$, where the pheromone matrix is of the form

$$\begin{pmatrix} \tau_{11} & \tau_{12} & 1 - \tau_{11} - \tau_{12} \\ \tau_{21} & \tau_{22} & 1 - \tau_{21} - \tau_{22} \\ 1 - \tau_{11} - \tau_{21} & 1 - \tau_{12} - \tau_{22} & \tau_{11} + \tau_{12} + \tau_{21} + \tau_{22} - 1 \end{pmatrix}$$

We are interested in selection fixed points of the ACO model with such matrices, i.e., matrices for which $\tau_{ij} = \sigma_{ij}$ for all $i, j \in [1 : 3]$. Clearly, the selection probabilities for the items in the first row are always equal to their corresponding pheromone values. Therefore we consider these values as constant in this section. Since the pheromone matrix is determined by the four values τ_{11} , τ_{12} , τ_{21} , and τ_{22} , it remains to determine the probability of choosing the first and the second item in the second row. The selection probabilities for these items are

$$\begin{aligned} \sigma_{21} &:= \frac{\tau_{12} \tau_{21}}{1 - \tau_{22}} + \frac{(1 - \tau_{11} - \tau_{12}) \tau_{21}}{\tau_{21} + \tau_{22}} \\ \sigma_{22} &:= \frac{\tau_{11} \tau_{22}}{1 - \tau_{21}} + \frac{(1 - \tau_{11} - \tau_{12}) \tau_{22}}{\tau_{21} + \tau_{22}} \end{aligned}$$

Solutions of the equations $\sigma_{21} - \tau_{21} = 0$ and $\sigma_{22} - \tau_{22} = 0$ are

1. $\tau_{21} = 0, \tau_{22} = (\tau_{11} + \tau_{12} - 1)/(\tau_{11} - 1)$
2. $\tau_{21} = (\tau_{11} + \tau_{12} - 1)/(\tau_{12} - 1), \tau_{22} = 0$
3. $\tau_{21} = (\tau_{12})/(\tau_{11} + \tau_{12}), \tau_{22} = (\tau_{11})/(\tau_{11} + \tau_{12})$
4. $\tau_{21} = 1 - 2 \cdot \tau_{11}, \tau_{22} = 1 - 2 \cdot \tau_{12}$

Analyzing the eigenvalues of the Jacobian matrix of the vector function $[f_1, f_2] := [\sigma_{21} - \tau_{21}, \sigma_{22} - \tau_{22}]$ that determines the change of $[\tau_{21}, \tau_{22}]$ from one iteration to the next, the stability of the selection fixed point matrices can be determined. For example, for Case (1), the Jacobian matrix is

$$\left(\begin{array}{cc} \frac{\partial f_1}{\partial \tau_{21}} & \frac{\partial f_1}{\partial \tau_{22}} \\ \frac{\partial f_2}{\partial \tau_{21}} & \frac{\partial f_2}{\partial \tau_{22}} \end{array} \right) \bigg|_{\begin{array}{l} \tau_{21}=0 \\ \tau_{22}=\frac{(\tau_{11}+\tau_{12}-1)}{(\tau_{11}-1)} \end{array}} = \left(\begin{array}{cc} 1 - 2\tau_{11} & 0 \\ \frac{(2\tau_{11}^2 - 3\tau_{11} + \tau_{11} \cdot \tau_{12} + 1)}{(\tau_{11} - 1)} & \tau_{11} - 1 \end{array} \right)$$

with eigenvalues $1 - 2 \cdot \tau_{11}$ and $\tau_{11} - 1$. A selection fixed point matrix is stable when all eigenvalues of the Jacobian matrix are negative. The analysis of the Jacobian matrix shows that for every pair of possible values τ_{11} and τ_{12} , exactly one of the selection fixed point matrices is stable and attracting (with respect to changes of (τ_{21}, τ_{22})) in the range of possible pheromone values. Cases (1), (2), and (3) are symmetric: for $\tau_{11} > 0.5$, the selection fixed point (1) is stable, for $\tau_{12} > 0.5$, the selection fixed point (2) is stable, and for $1 - \tau_{11} - \tau_{12} > 0.5$, the selection fixed point (3) is stable. In every other case, the selection fixed point (4) is stable. Thus, there always exists exactly one stable selection fixed point matrix. Some of the three unstable selection fixed points might lay outside of the allowed parameter range $\tau_{21}, \tau_{22}, \tau_{23} \in (0, 1), \tau_{21} + \tau_{22} + \tau_{23} = 1$.

The analysis above has shown that the selection fixed points depend only on the pheromone values in the first row of the pheromone matrix. Figure 1 shows for different pheromone vectors $(\tau_{11}, \tau_{12}, \tau_{13})$ the corresponding selection fixed points. The situation is symmetric and therefore only the case $\tau_{11} > 1/3$ is considered. The figure shows that the pheromone values in row 2 of the selection fixed point matrices are dependent on the pheromone values in row 1. For selection probabilities $(\tau_{11}, \tau_{12}, \tau_{13}) = (1/3, 1/3, 1/3)$, the ACO model has a stable selection fixed point with $(\tau_{21}, \tau_{22}, \tau_{23}) = (1/3, 1/3, 1/3)$. This is the pheromone matrix that would typically be used as the initialization matrix for an ACO algorithm. Hence when starting with this matrix, all dynamic in the first iteration of the ACO model stem from competition effects and not from a selection bias. The corresponding three unstable selection fixed points lay at the edges of the inner triangle with values for $(\tau_{21}, \tau_{22}, \tau_{23})$ of $(1/2, 1/2, 0)$, $(0, 1/2, 1/2)$, and $(1/2, 0, 1/2)$.

For other values of $(\tau_{11}, \tau_{12}, \tau_{13})$, the value τ_{1j} of the stable selection fixed point and its value τ_{2j} show a converse behavior as can be seen in the Figure 1. For example, for pheromone vector $(\tau_{11}, \tau_{12}, \tau_{13})$ with $\tau_{11} > 0.5$, the stable selection fixed point has value $\tau_{21} = 0$. For $1/3 < \tau_{11} < 0.5$ and $\tau_{12} < 1/3$, the stable selection fixed point has values $\tau_{21} < 1/3, \tau_{22} > 0.5$. When pheromone vector $(\tau_{11}, \tau_{12}, \tau_{13})$ lays in the inner triangle in Figure 1, then on every borderline of the triangle of possible pheromone values there is a vector $(\tau_{21}, \tau_{22}, \tau_{23})$ of an unstable selection fixed point. For $\tau_{11} > 1/2$, one of the unstable selection fixed points lies outside of the triangle with a value $\tau_{21} < 0$.

The directions of the vector field for changes of pheromone vector $(\tau_{21}, \tau_{22}, \tau_{23})$ when $(\tau_{11}, \tau_{12}, \tau_{13}) = (1/3, 1/3, 1/3)$ are shown in Figure 2. The left part of the figure shows the case of local pheromone evaluation and the right part the case of summation evaluation with selection probabilities determined according to Equation (4). In this case, both vector fields are symmetric with respect to rotations of 60 degrees around the fixed point $(\tau_{21}, \tau_{22}, \tau_{23}) = (1/3, 1/3, 1/3)$. It is interesting to observe that for local evaluation in some areas of the vector field there are points $(\tau_{21}, \tau_{22}, \tau_{23})$ with a value $\tau_{2i} > 1/3$ for an $i \in [1 : 3]$ that becomes even larger when there is only one ant per iteration. This is an unexpected effect of the selection bias but which shows that the

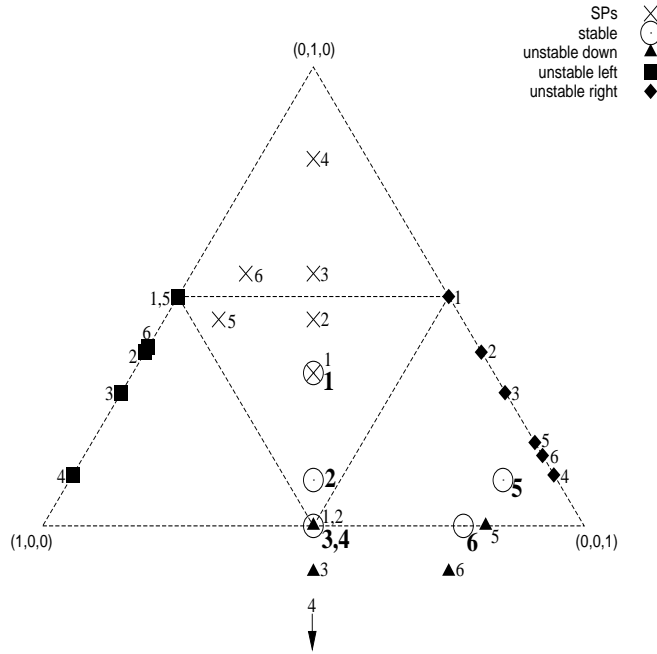


Figure 1: Stable and unstable selection fixed points for six different pheromone vectors $(\tau_{11}, \tau_{12}, \tau_{13})$: given are the selection probabilities (SP) $(\tau_{11}, \tau_{12}, \tau_{13})$ and the corresponding fixed point described by the vector $(\tau_{21}, \tau_{22}, \tau_{23})$ (these values determine the fixed point, since pheromone values $\tau_{11}, \tau_{12}, \tau_{13}$ are equal to the selection probabilities and do not change). To read the figure, note that every point in the plane is defined by a triple of coordinates that denote their shortest distances from the three lines that are defined by the three edges of the outer equilateral triangle (signs indicate on which side of a line a point lies, for points in the same halfspace as the triangle the corresponding sign is a plus, otherwise it is a minus); points within the (outer) triangle with edge lengths 1 have positive distances with sum 1; the pheromone vectors $(\tau_{11}, \tau_{12}, \tau_{13})$ and $(\tau_{21}, \tau_{22}, \tau_{23})$ are represented by points in the triangle as follows: distance from right line defined by $(0, 1, 0)$ and $(0, 0, 1)$, τ_{11}, τ_{21} ; distance from bottom line defined by $(1, 0, 0)$ and $(0, 0, 1)$, τ_{12}, τ_{22} ; distance from left line defined by $(0, 1, 0)$ and $(1, 0, 0)$, τ_{13}, τ_{23} ; the inner triangle contains the points with all values $\leq 1/2$; numbers denote the corresponding SPs and fixed points.

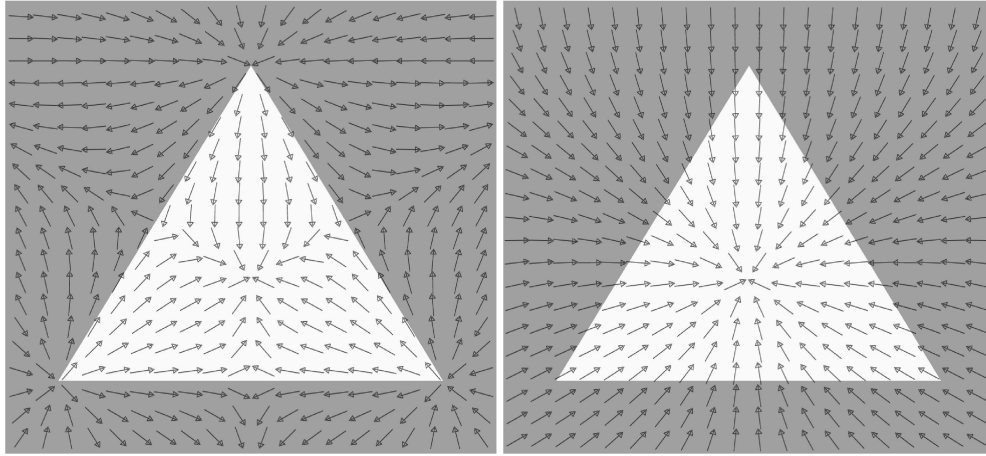


Figure 2: Direction of the vector field for changes of pheromone vectors $(\tau_{21}, \tau_{22}, \tau_{23})$ for the ACO model with a single ant and $\tau_{11} = \tau_{12} = \tau_{13} = 1/3$. Local pheromone evaluation (left); summation evaluation (right); possible pheromone vectors lay in the white area; distance from right line is τ_{21} ; distance from bottom line is τ_{22} ; distance from left line is τ_{23} .

effects of the selection bias can be complex even for small problems. For summation evaluation, the vectors point much more straight in the direction of the fixed point. This is interesting because it is an additional hint that summation evaluation could be a good method for certain types of permutation problems.

The selection fixed point analysis has shown that for this small system there is always only one stable selection fixed point matrix for the given first row of the pheromone matrix. This is an indication that the selection bias could be an important factor for understanding the optimization behavior of ACO algorithms. The selection bias in the decisions of random ants will drive the system toward its (actual) stable selection fixed point when the random ants are allowed to update the pheromone information.

7 Dynamic Behavior

In this section, we study the dynamic behavior of the ACO model on small (general) permutation problems and on larger restricted permutation problems. One aspect of the study is to find out whether the selection bias that was investigated in the last section can have a significant influence on the dynamic behavior of the ACO model under competition, i.e., when there is more than one ant in an iteration. As example problems we use restricted permutation problems of different size with problem P_1 as the elementary subproblem of size $n_0 = 3$. Recall that the cost matrix is given in (9) in Section 5.

Figures 3–6 show the dynamic behavior of the ACO model for the restricted permutation problem P_1^q with different values of q and $m = 2$ or $m = 3$ ants. The pheromone evaporation parameter ρ is 0.1 and the initial pheromone matrix is the same as in (16) in the example of Section 5, i.e., the pheromone matrix is determined by $\tau_{11} = 0.1$, $\tau_{12} = 0.3$, $\tau_{21} = 0.6$, $\tau_{22} = 0.1$. Each figure contains curves for numbers of elementary subproblems $q = 2, 4, 8, 16, 32, 64$, and 128. In addition, Figures 3–5 contain

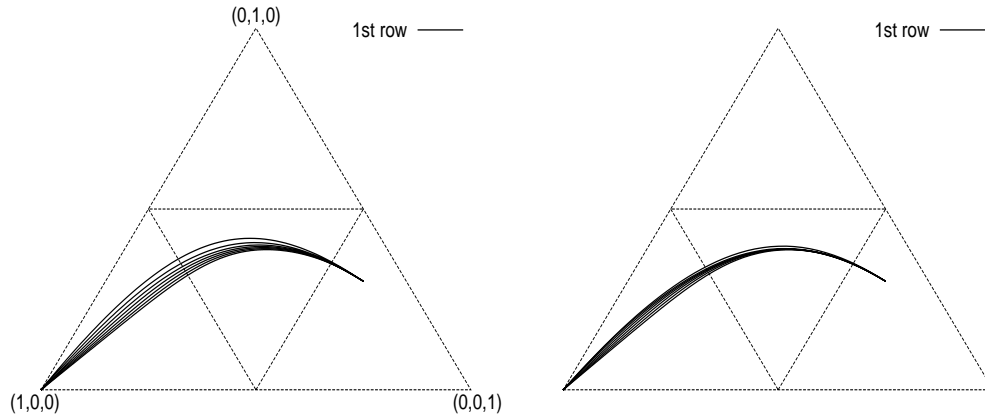


Figure 3: ACO model for P_1^q with size $n_0 = 3$ of P_1 , $q = 2, 4, 8, 16, 32, 64, 128$, initial pheromone matrix as in (16); $m = 2$ ants (left); $m = 3$ ants (right); change of pheromone values $\tau_{11}, \tau_{12}, \tau_{13}$ (values in the first row of the corresponding fixed point matrix are identical): starting point is $(\tau_{11}, \tau_{12}, \tau_{13}) = (0.1, 0.3, 0.6)$.

the curves for the corresponding positions of the stable selection fixed point, i.e., the stable fixed point for the case when there is only one ant per iteration. Recall that the vector of pheromone values in the first row always equals the values in the first row of the selection fixed points.

In contrast to the situation of one ant (where the system converges to its stable selection fixed point), the ACO model converges in this example to the optimal solution. But during convergence the pheromone vectors in the ACO model do not follow a straight path toward the values corresponding to the optimal solution. The actual positions of the selection fixed point have a strong influence on the system. Note that for the first row of the matrix the pheromone values always equal the corresponding selection probabilities. Hence the dynamic of the first row is influenced only indirectly by a selection bias. Therefore, the pheromone values in Figure 3 approach the optimal values on an almost straight path, and there is no big difference between 2 ants and 3 ants.

This is clearly different for the pheromone vectors of row 2 (see Figure 4) and even more for row 3 (see Figure 5). It can be seen that during most of the generations the stable selection fixed point has a large influence on the system. In rows 2 and 3 the system moves often more in direction of the stable selection fixed point than in direction of the optimal solution. In the middle of the run, the system even moves away from the optimal solution considering these rows. Note that every path of the selection fixed points in Figures 4 and 5 consists of 3 subpaths that correspond to the 3 subpaths of the pheromone values in the first row, which move through three of the smaller triangles. In Figure 4 the first and the last of the three subpaths lay on the border of the triangle. In Figure 5 there is a sharp bend at the edges of the inner triangle between different subpaths.

In Figure 5 most paths (those with $q \geq 8$ for $m = 2$ ants and with $q \geq 16$ for $m = 3$ ants) contain a loop that is clearly influenced by the turn of the selection fixed point. For $m = 3$ ants, where the influence of competition is stronger than for $m = 2$ ants,

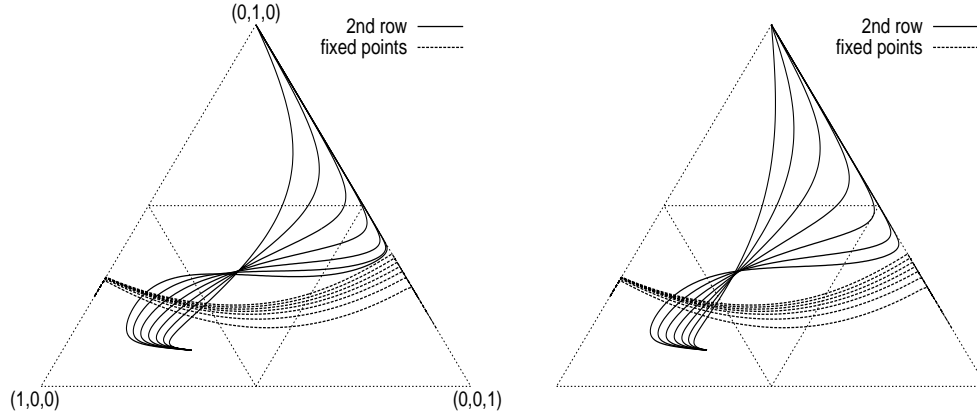


Figure 4: ACO model for P_1^q with size $n_0 = 3$ of P_1 , $q = 2, 4, 8, 16, 32, 64, 128$, initial pheromone matrix as in (16); $m = 2$ ants (left); $m = 3$ ants (right); change of pheromone values $\tau_{21}, \tau_{22}, \tau_{23}$ and corresponding values of the second row of the fixed point matrix: starting point is $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$ and the corresponding fixed point values are $(0.75, 0.25, 0.0)$; pheromone curves (fixed point curves) that are lower (higher) in the right half of the triangle correspond to higher q values.

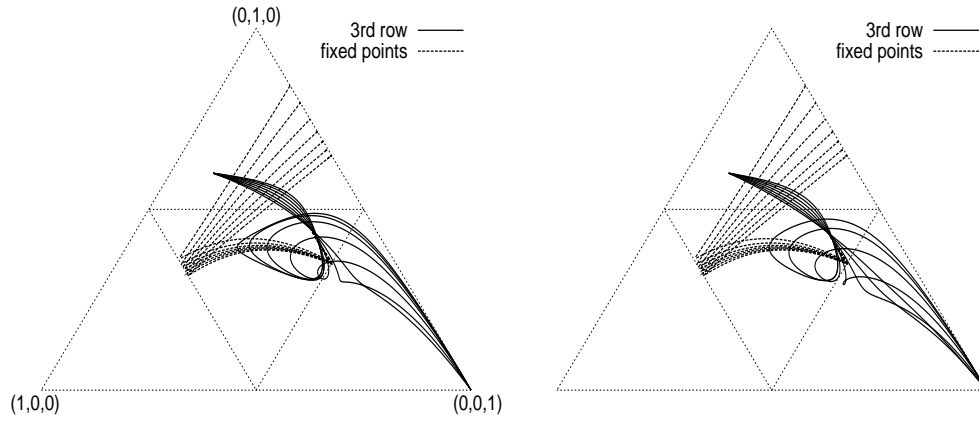


Figure 5: ACO model for P_1^q with size $n_0 = 3$ of P_1 , $q = 2, 4, 8, 16, 32, 64, 128$, initial pheromone matrix as in (16); $m = 2$ ants (left); $m = 3$ ants (right); change of pheromone values $\tau_{31}, \tau_{32}, \tau_{33}$ and corresponding values of the third row of the fixed point matrix: starting point is $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$ and the corresponding fixed point values are $(0.15, 0.45, 0.4)$; pheromone curves (fixed point curves) that are higher (lower) in the upper half of the triangle correspond to higher q values.

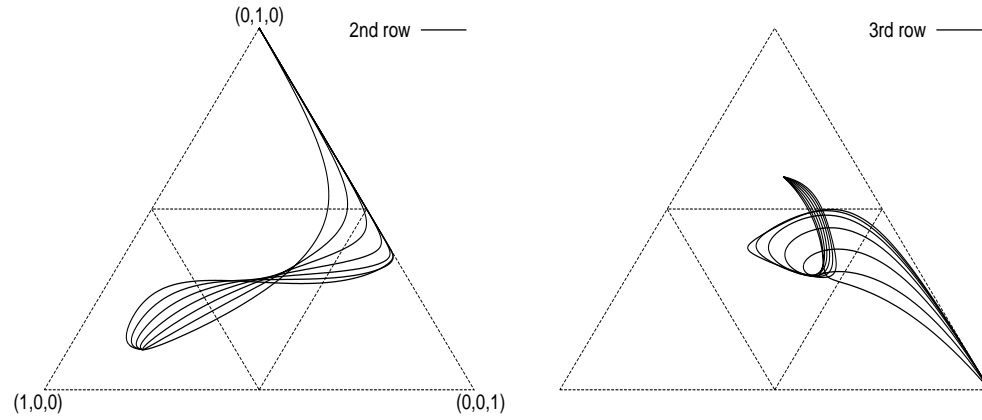


Figure 6: ACO model for P_1^q with size $n_0 = 3$ of P_1 , $q = 2, 4, 8, 16, 32, 64, 128$, initial pheromone matrix as in (16) and $m = 2$ ants; change of selection probabilities $\sigma_{21}, \sigma_{22}, \sigma_{23}$ in second row (left) and selection probabilities $\sigma_{31}, \sigma_{32}, \sigma_{33}$ third row (right): starting point of probabilities is $(13/70, 33/56, 9/40)$ in row 2 and $(13/70, 33/56, 9/40)$ in row 3.

the curves are slightly more straight. For example, in Figure 4 the curves are driven more to the right in the upper part than the curves for 3 ants. In Figure 5 the curve for $q = 4$ and 3 ants makes only a smooth bend while there is sharp bend for 2 ants. The larger the number q of elementary subproblems, the stronger is the deviation from a straight line. For example, in Figure 2 the path for $q = 2$ is only slightly bent whereas the curve for $q = 128$ has a zigzag form that in the middle follows the horizontal path of the stable fixed point. The reason is that a high number of elementary subproblems leads to a small influence of competition (see Section 5).

Figure 6, when compared with the left parts of Figures 4 and 5, shows that the selection probabilities for the case of $m = 2$ ants change similarly during the run as the corresponding pheromone values. But they are not exactly the same, which means that there is a selection bias. For example, compare the different starting points of the pheromone curves and the corresponding curves for the selection probabilities.

The exact position of the pheromone vectors of the ACO model and the corresponding fixed points at different iterations is shown in Figures 7 and 8 for problem P_1^q with $q = 64$ elementary subproblems. Again, it can be seen that the system moves in direction of the selection fixed points for about 200 iterations (slightly more for $m = 2$ and slightly less for $m = 3$) until the influence of competition seems to become more important.

Figure 9 shows the changes of the pheromone values in rows 2 and 3 of the pheromone matrix when the summation evaluation method is used for $m = 2$ ants. The results for row 1 are so similar to the results for local evaluation that they are not depicted. The figure shows, when compared to the left parts of Figures 4 and 5, that the system moves much more straight to the optimal solution than when using local pheromone evaluation. This behavior was expected since the example problem is a single machine total lateness problem (see Section 5) that falls into the problem class for which summation evaluation was originally proposed (cmp. Section 3).

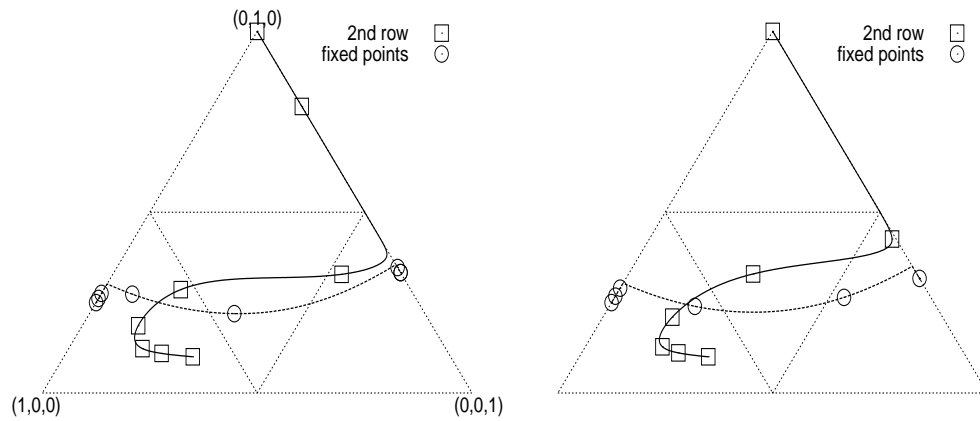


Figure 7: ACO model for P_1^q with size of P_1 $n_0 = 3$, $q = 64$, $m = 2$ ants (left) and $m = 3$ ants (right) at iterations 0, 10, 20, 50, 100, 200, 500, and 1000; fixed points and phomone values $\tau_{21}, \tau_{22}, \tau_{23}$ starting at $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$; when less than 8 values are visible, the final values are the same.

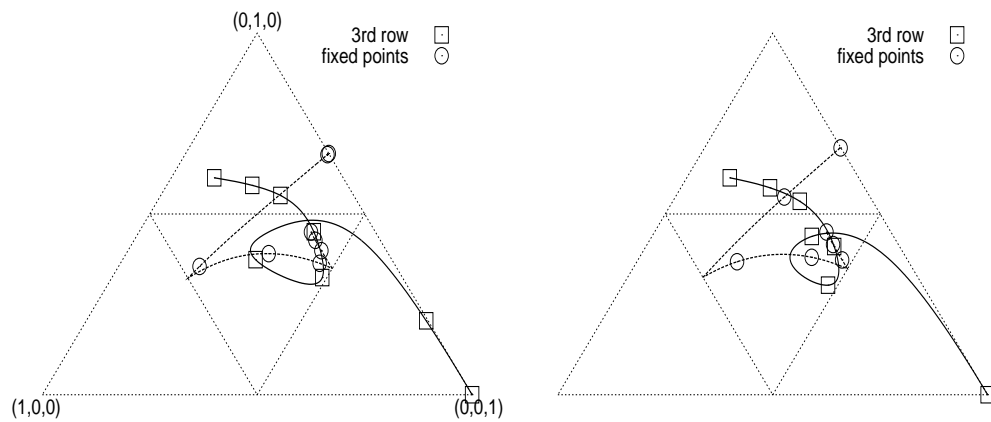


Figure 8: ACO model for P_1^q with size of P_1 $n_0 = 3$, $q = 64$, $m = 2$ ants (left) and $m = 3$ ants (right) at iterations 0, 10, 20, 50, 100, 200, 500, and 1000; fixed points and phomone values $\tau_{31}, \tau_{32}, \tau_{33}$ starting at $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$; when less than 8 values are visible, the final values are the same.

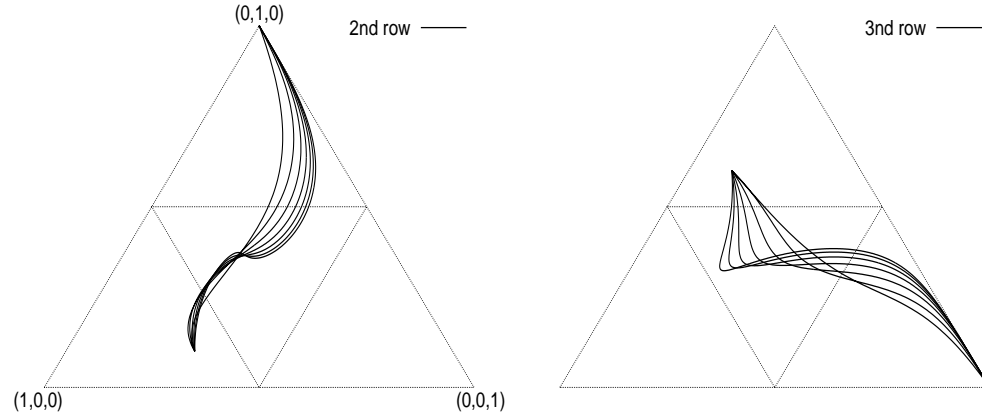


Figure 9: ACO model with summation evaluation for P_1^q with size of P_1 $n_0 = 3$, $q = 2, 4, 8, 16, 32, 64, 128$, $m = 2$ ants. Left: change of pheromone values $\tau_{21}, \tau_{22}, \tau_{23}$ starting at $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$, curves that are more right in the upper half of the triangle correspond to higher q values. Right: change of pheromone values $\tau_{31}, \tau_{32}, \tau_{33}$ starting at $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$, curves that are more left in the upper half of the triangle correspond to higher q values.

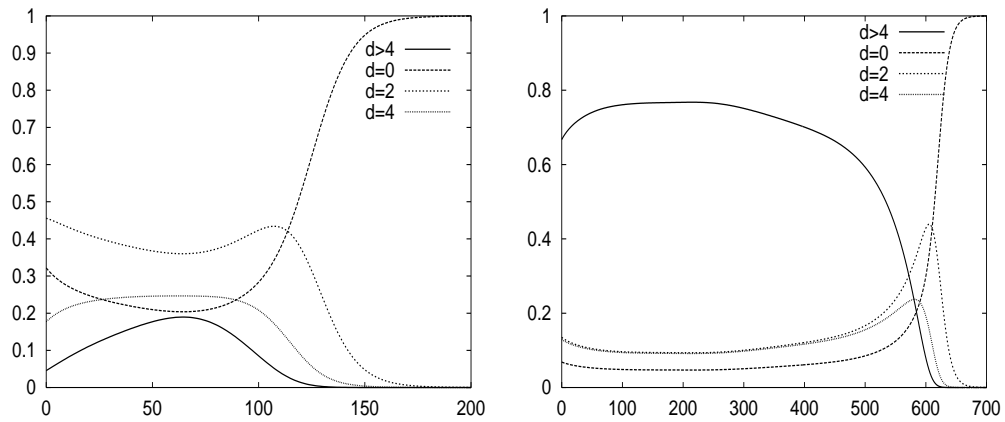


Figure 10: ACO model for P_1^q with size $n_0 = 3$ of P_1 and 2 ants. $q = 4$ (left); $q = 64$ (right); change of the probabilities ϕ_d for several differences d in solution quality between the two ants on subproblem P_1^{q-1} : $\phi_0^{(2)}, \phi_2^{(2)}, \phi_4^{(2)}, \phi_{>4}^{(2)}$.

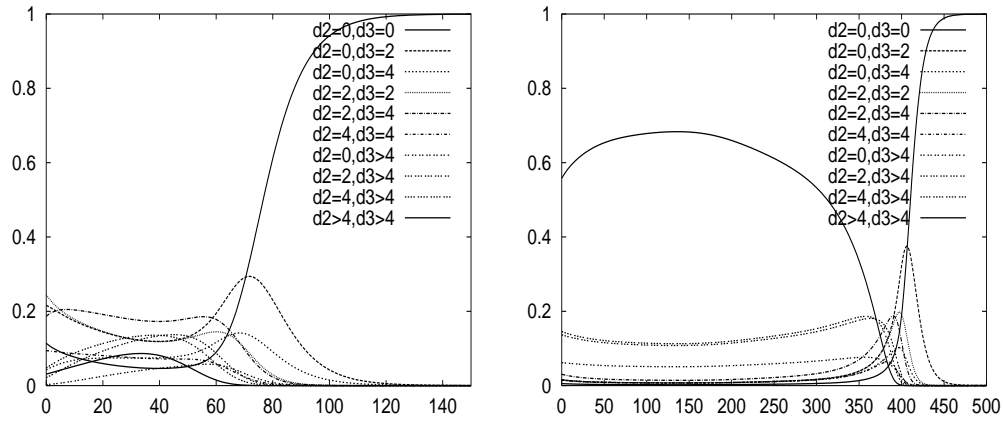


Figure 11: ACO model for P_1^q with size $n_0 = 3$ of P_1 and 3 ants. $q = 4$ (left); $q = 64$ (right); change of probabilities for difference in solution quality between ants on the problem P_1^{q-1} : $\phi_{d_2 d_3}^{(3)}$ with $d_2 \leq d_3$, $d_2, d_3 \in \{0, 2, 4, > 4\}$ (defining $4 < > 4''$).

In order to investigate the relative influence of selection, pure competition, and weak competition (where one or more ants have a malus), we computed the probabilities for the possible differences in solution quality between the two ants on the smaller problem P_1^{q-1} . Recall that the solution quality for the elementary subproblem P_1 can be 0, 2, or 4. Figure 10 shows the probabilities $\phi_0, \phi_2, \phi_4, \phi_{>4}$ for a small number of subproblems $q = 4$ compared to a large number of subproblems $q = 64$. For 3 ants per iteration, the corresponding probabilities are shown in Figure 11. Both figures show that for a large number of elementary subproblems the cases $\phi_{>4}$, respectively $\phi_{>4, >4}$, which correspond to selection by a single ant on the elementary subproblem, have a probability of more than 50% over most parts of the run. Only when the ACO model starts to converge is the model driven more by competition and the values $p_{d>4}$, respectively $\phi_{>4, >4}$, decline faster (as suggested by the analysis of the dynamics of the pheromone values). This is the case after about 550 iterations for the case of 2 ants per iteration and earlier after about 300 iterations when there is more competition by 3 ants per iteration.

In the following, we show that the influence of competition can be too weak compared to the influence of selection bias so that the ACO model is not able to find the optimal solution for $m > 1$ ants. This is true even for very small permutation problems. Consider the following problem P_2 with cost matrix

$$C_2 = \begin{pmatrix} 0 & 1 & 100 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (18)$$

Figure 12 shows the behavior of the ACO model on test problem P_2^q when the pheromone matrix is initialized with $\tau_{11} = 0.1$, $\tau_{12} = 0.2$, $\tau_{21} = 0.1$, $\tau_{22} = 0.7$ for different number q of elementary subproblems. For small numbers of 6, 10, and 14 subproblems, the system converges to the optimal solution. But for larger numbers of 18 and 22 subproblems the influence of the selection bias is so high compared to

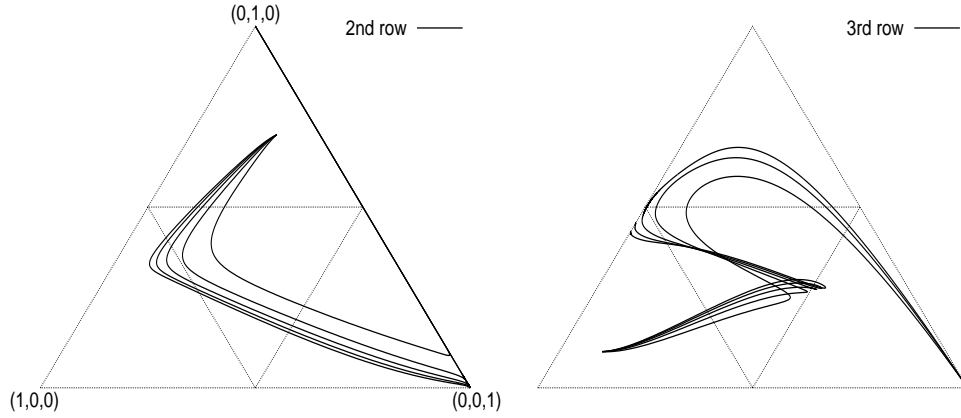


Figure 12: ACO model for problem P_2^q with $m = 2$ ants and $q = 6, 10, 14, 18, 22$ starting at initial matrix defined by $\tau_{11} = 0.1$, $\tau_{12} = 0.2$, $\tau_{21} = 0.1$, $\tau_{22} = 0.7$. Left: change of pheromone values $\tau_{21}, \tau_{22}, \tau_{23}$. Right: change of pheromone values $\tau_{31}, \tau_{32}, \tau_{33}$; for 6,10,14 elementary subproblems the system converges to the optimum with $(\tau_{21}, \tau_{22}, \tau_{23}) = (0, 1, 0)$ (difficult to see in the figure) and $(\tau_{31}, \tau_{32}, \tau_{33}) = (1, 0, 0)$, for 18 and 22 elementary subproblems it converges to a non-optimal solution with $(\tau_{21}, \tau_{22}, \tau_{23}) = (0, 0, 1)$ and $(\tau_{31}, \tau_{32}, \tau_{33}) = (x, y, 0)$.

the influence of competition that the system converges to a non-optimal solution. The figure shows that even for the small numbers of elementary subproblems the system is driven by a selection bias, but competition becomes stronger early enough to change the direction of convergence to a non-optimal solution, i.e., to a pheromone matrix of the following form with costs 2

$$\begin{pmatrix} x & 1-x & 0 \\ 0 & 0 & 1 \\ 1-x & x & 0 \end{pmatrix}$$

While these results are for a specific initial pheromone matrix, we also tested the system for all 666 matrices with a feasible combination of pheromone values $\tau_{ij} \in 0.1, 0.2, \dots, 0.8$ for $i, j \in [1 : 3]$. Figure 13 shows the number of initial matrices where the ACO model cannot find the optimal solution (which implies that none of the subproblems is solved optimal) with respect to the number of elementary subproblems. Even for the small problem P_2^2 with only two elementary subproblems for 83 of the 666 different initial matrices, the optimal solution cannot be found. This number increases with a larger number of elementary subproblems over 101 for P_2^3 up to 296 for P_2^{60} .

The discussion in this section has shown that even under competition, the dynamic behavior of the ACO model is strongly influenced by the selection bias. The influence can be so strong that even a small system consisting of independent subproblems where there is competition between more than one ant, converges to suboptimal solutions for a growing number of initial pheromone matrices when the number of subproblems increases. In general, the smaller the number of ants and the larger the problem (when there are independent subproblems), the stronger the influence of the selection bias.

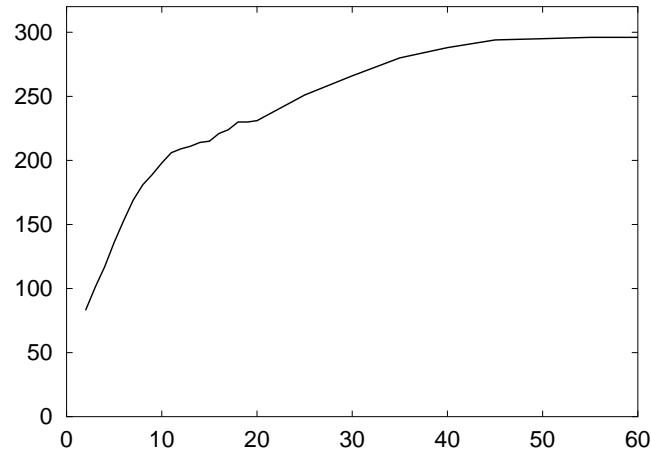


Figure 13: ACO model for P_2^q with size $n_0 = 3$ of P_2 , $q = 1, 2, \dots, 20, 25, \dots, 60$, and $m = 2$ ants: number of starting pheromone matrices where the elementary subproblems are not solved optimal; starting pheromone matrices are all 666 feasible matrices with $\tau_{ij} \in 0.1, 0.2, \dots, 0.8$, $i, j \in [1 : 3]$.

8 Simulation Results

In this section, we show the results of several test runs for the ACO algorithm on restricted permutation problems and compare them with results for the ACO model. It is obvious that the model will usually not reflect the behavior of the algorithm reasonably well when the optimization problem is small and the evaporation parameter ρ is large. In this case, few decisions of the ants have a strong influence on the ant algorithm, and even several runs of the ant algorithm will be very different from each other. In general, it is difficult to define in a reasonable way what the average over several runs of an ant algorithm is. Therefore we compared single runs of the ant algorithm with a run of the model. To make such a comparison meaningful we have to restrict the variation between several runs of the algorithm. Hence, a very small value of $\rho = 0.0001$ was chosen for the algorithm. For the ACO model, $\rho = 0.1$ was used because otherwise the computation times for the model are long for the large problems. When comparing the ACO algorithm with the ACO model at corresponding iterations, we compare iteration t of the model with iteration $1000t$ of the algorithm. Observe that for the chosen parameter values, the total amount of pheromone that is added in one iteration by the model is the same as the sum of the pheromone added in 1000 iterations by the algorithm. But this establishes an additional difference between the model and the algorithm. For the algorithm, the pheromone that evaporates during 1000 iterations is a mixture of new and old pheromone, whereas for the model, only old pheromone evaporates in one iteration. Moreover new pheromone that is added by the algorithm during 1000 iterations has an immediate influence on the following iterations.

Figure 14 shows the behavior of the ACO algorithm and the ACO model on our example problem P_1^q that is described in Section 4. Here a large problem with $q = 128$ elementary subproblem is used (the cost matrix is given in (9)). It can be seen that the solution quality found by a random ant of the ACO model is nearly the same as

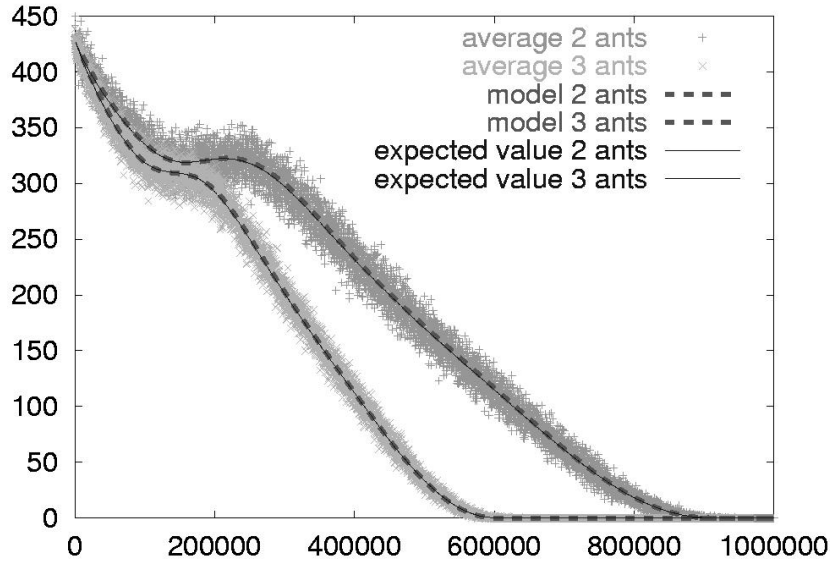


Figure 14: Solution quality of ACO model and ACO algorithm on problem P_1^{128} with $m = 2$ (upper curves) and $m = 3$ (lower curves) ants over 1000000 (respectively 1000) iterations. Average m ants: average solution quality found by m ants in the iteration of ACO algorithm; expected value m ants: expected quality of solution of a random ant with respect to the pheromone matrix in the generation of ACO algorithm; model m ants: solution quality of a random ant for ACO model. Note that the curves for the expected values of m ants and the model for m ants are nearly the same.

the expected behavior of a random ant on the observed pheromone matrices of the ACO algorithm (in the figure the corresponding curves are nearly identical). Since the number of ants per generation of the ACO algorithm was small ($m = 2$ or $m = 3$), the observed average quality of the solutions found by the $m = 2$ or $m = 3$ ants in the iterations of the algorithm fluctuates around the solution quality that can be expected from the pheromone matrix in that generation. An interesting observation is that the ACO model and the ACO algorithm for $m = 2$ show that the expected solution quality that is found during an iteration becomes worse during the run. Note, that this effect is not the result of some disadvantageous random decisions but is predicted by the model. With stronger competition, when there are $m = 3$ ants per iteration, the expected solution quality always improves from iteration to iteration in the example problem.

Figure 15 shows the results of test runs of the ACO algorithm on problems P_1^q with a different number q of elementary subproblems for $m = 2$ ants. Every curve in the figure stems from only one run of the algorithm because it is not clear how to average the results of several test runs in a reasonable way. Therefore the curves are not very smooth but show random fluctuations. Nevertheless the figure shows when compared to the left parts of Figures 4 and 5, that the ACO model predicts very well the development of the pheromone values of the ACO algorithm.

Figure 16 shows the pheromone values for all elementary subproblems of P_1^q with

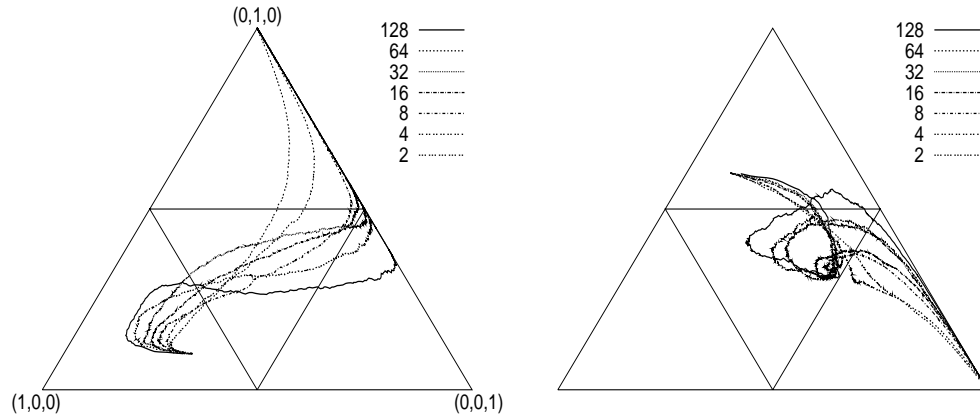


Figure 15: ACO algorithm for P_1^q with size $n_0 = 3$ of P_1 , $q = 2, 4, 8, 16, 32, 64, 128$, and $m = 2$ ants. Left: change of pheromone values $\tau_{21}, \tau_{22}, \tau_{23}$ starting at $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$; Right: change of pheromone values $\tau_{31}, \tau_{32}, \tau_{33}$ starting at $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$.

$q = 64$ and $m = 2$ ants for the ACO algorithm and the ACO model at different iterations. Clearly, there is some variation between the pheromone values of the subproblems in the run of the ACO algorithm, whereas for the ACO model the pheromone values for all subproblems are the same. It can be seen that the pheromone vectors of the ACO algorithm are clustered around the corresponding pheromone vector of the ACO model. For 10 and 20 iterations the ACO model seems to be driven somewhat slower than the algorithm for the corresponding numbers of 10000 and 20000 iterations.

Of course not all aspects of ACO algorithms behavior can also be observed in the ACO model. As an example consider the restricted homogeneous permutation problem P^q where the ACO model shows the same behavior on each of the elementary subproblems. The ACO algorithm in contrast behaves differently on the elementary subproblems due to random effects. While for problem P_2^q the ACO model either solves all or none of the elementary subproblems optimally, this is not true for the ACO algorithm. Table 1 shows the number of elementary subproblems that are solved optimally (i.e., the pheromone matrix converges to the optimal solution so that the probability to be chosen by an ant is one modulo the smallest floating point number) by the ACO algorithm with $m = 2$ ants when starting with an initial pheromone matrix defined by $\tau_{11} = \tau_{12} = \tau_{21} = \tau_{21} = 1/3$. The runs were done until the algorithm converged on all subproblems to one solution. The results show that for $q = 5$ all elementary subproblems were solved optimally. But the larger the number q of elementary subproblems, the smaller the proportion of elementary subproblems solved optimally by the algorithm. Only 14.7% of elementary subproblems are solved optimally for $q = 100$. The ACO model, when starting with the same initial pheromone matrix, solves all elementary subproblems optimally for every value of q that occurs in Table 1. But recall that the ACO model also shows the effect that for large q , the problem P^q becomes more difficult, because for large q , there are many other initial pheromone matrices for which the ACO model cannot solve the elementary subproblems optimally (cmp. Figure 13).

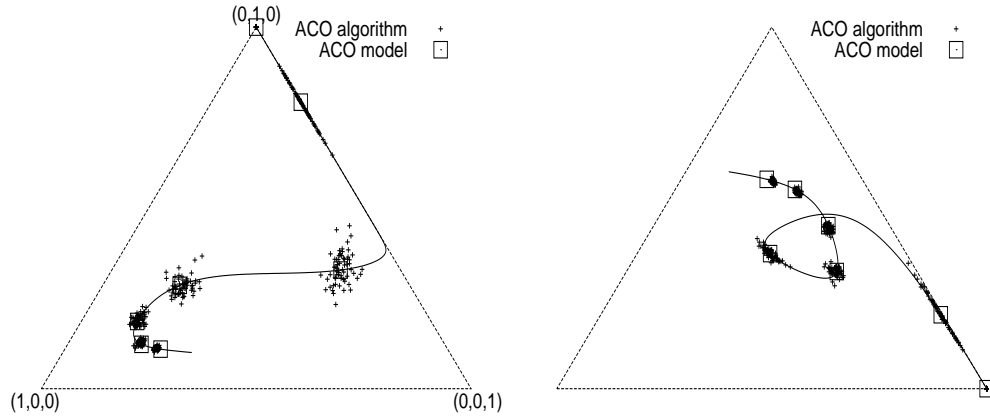


Figure 16: ACO algorithm and ACO model for the elementary subproblems of P_1^q with $q = 64$, $m = 2$ ants at iterations 10000, 20000, 50000, 100000, 200000, 500000, and 1000000 (respectively 10, 20, 50, 100, 200, 500, and 1000 — the corresponding curve for the ACO model from the left parts of Figures 4 and 5 over all iterations is also shown). Left: pheromone values $\tau_{21}, \tau_{22}, \tau_{23}$ starting at $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$; Right: pheromone values $\tau_{31}, \tau_{32}, \tau_{33}$ starting at $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$.

Table 1: ACO algorithm for problem P_2^q with $m = 2$ ants: average number of elementary subproblems that were solved optimal (i.e., the pheromone matrix converges to the optimal solution); average over 10 runs.

q	optimal	%	variance
5	5.0	100.0	0.0
10	8.7	87.0	0.6
15	10.1	67.3	0.9
20	10.8	54.0	2.4
25	12.0	48.0	2.4
30	12.1	40.3	1.7
40	12.5	31.3	2.7
50	14.1	28.2	3.7
75	14.6	19.5	2.4
100	14.7	14.7	2.2

9 Conclusion and Future Work

A deterministic model for ACO algorithms was proposed and analyzed. The ACO model uses a deterministic pheromone update mechanism that is based on the expected decisions of the ants instead of using random decisions as done in ACO algorithms. An interesting feature of the model is that it can be used to describe the behavior of an ACO algorithm through a combination of situations with different strength of competition between the ants.

The behavior of the ACO model was investigated analytically. A fixed point analysis of the pheromone matrices was done that gives insights into the occurrence of biased decisions by the ants. It was shown analytically that the positions of the fixed points in the state space of the system have a strong influence on its optimization behavior. Moreover, we analyzed how the number of independent subproblems of the permutation problems influences the composition of the system with respect to different levels of competition between the ants and therefore the system's optimization behavior.

It was shown by simulations that the model accurately describes essential features of the dynamic behavior of ACO algorithms. The test problems that were used are permutation problems that consist of several independent subproblems. We also discussed some limitations of the model and pointed out aspects of ACO algorithm behavior that are not represented in the model.

Our study suggests several directions for future work:

1. To establish mathematical bounds on how well the model approximates the algorithm. To investigate how much this depends on parameters like the evaporation parameter or the population size.
2. To study the ACO model on other types of problems like inhomogeneous restricted permutation problems or problems where other pheromone codings are used (for example the TSP).
3. To investigate various other variants of ACO algorithms with the ACO model, for example, ACO algorithms that use an elitist principle or heuristics.
4. To find alternative ways to model ACO algorithms.

Acknowledgments

We thank the anonymous referees for their helpful remarks.

References

- Bruce, I. D. and Simpson, R. J. (1999). Evolution Determined by Trajectory of Expected Populations: Sufficient Conditions, with Application to Crossover. *Evolutionary Computation*, 7(2):151–171.
- Dorigo, M. (1992). Optimization, Learning and Natural Algorithms (*in Italian*). Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.
- Dorigo, M. and Di Caro, G. (1999). The ant colony optimization meta-heuristic. In Corne, D., Dorigo, M., and Glover, F., editors, *New Ideas in Optimization*, pages 11–32, McGraw-Hill, London, UK.
- Dorigo, M., Maniezzo, V., and Colnari, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26:29–41.

- Gutjahr, W. (2000). A graph-based Ant System and its convergence, *Future Generation Computer Systems*, 16:873–888.
- Gutjahr, W. (2002). ACO algorithms with guaranteed convergence to the optimal solution. *Information Processing Letters*, 82:145–153.
- Merkle, D. and Middendorf, M. (2000). An Ant Algorithm with a new Pheromone Evaluation Rule for Total Tardiness Problems. In Cagnoni, S. et al., editors, *Real-World Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2000*, LNCS, 1803:287–296, Springer Verlag, Berlin, Germany.
- Merkle, D. and Middendorf, M. (2001a). A New Approach to Solve Permutation Scheduling Problems with Ant Colony Optimization. In Boers, E. J. W. et al., editors, *Applications of Evolutionary Computing: Proceedings of EvoWorkshops 2001*, LNCS, 2037:213–222, Springer Verlag, Berlin, Germany.
- Merkle, D. and Middendorf, M. (2001b). On the Behaviour of Ant Algorithms: Studies on Simple Problems. In *Proceedings of the Fourth Metaheuristics International Conference (MIC'20001)*, pages 573–577.
- Merkle, D., Middendorf, M., and Schmeck, H. (2002). Ant Colony Optimization for Resource-Constrained Project Scheduling. *IEEE Transactions on Evolutionary Computation*, in press.
- Mitchell, M. and Forrest, S. (1997). Royal Road Functions. In Bäck, T., Fogel, D., and Michalewicz, Z., editors, *Handbook of Evolutionary Computation*. Oxford University Press, Oxford, UK.
- van Nimwegen, E., Crutchfield, J. P., and Mitchell, M. (1999). Statistical dynamics of the royal road genetic algorithm. *Theoretical Computer Science*, 229:41–102.
- Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs.
- Prügel-Bennett, A. and Rogers, A. (2001). Modelling GA Dynamics. In Kallel, L., Naudts, B., and Rogers, A., editors, *Theoretical Aspects of Evolutionary Computing*, pages 59–86, Springer, Berlin, Germany.
- Prügel-Bennett, A. and Shapiro, J. L. (1994). An Analysis of Genetic Algorithms Using Statistical Mechanics. *Physical Review Letters*, 72:1305–1309.
- Rajendran, C. and Ziegler, H. (2001). Ant-Colony Algorithms for Flowshop Scheduling, *European Journal of Operational Research*, submitted.
- Stützle, T. and Dorigo, M. (2000). A short convergence proof for a class of ACO algorithms. Technical Report IRIDIA/2000-35, IRIDIA, Université Libre de Bruxelles, Belgium.
- Teich, T. et al. (2001). A new Ant Colony Algorithm for the Job Shop Scheduling Problem. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, page 803, Morgan Kaufmann, San Francisco, California.
- Vose, M. D. (1999). *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, Massachusetts.
- Vose, M. D. and Liepins, G. E. (1991). Punctuated equilibria in genetic search. *Complex Systems*, 5:31–44.