A Graph–based Ant System and its Convergence

Walter J. Gutjahr
Department of Statistics, Operations Research and Computer Science
University of Vienna

Abstract: A general framework for solving combinatorial optimization problems heuristically by the Ant System approach is developed. The framework is based on the concept of a construction graph, a graph assigned to an instance of the optimization problem under consideration, encoding feasible solutions by walks. It is shown that under certain conditions, the solutions generated in each iteration of this Graph–based Ant System converge with a probability that can be made arbitrarily close to one to the optimal solution of the given problem instance.

Keywords: Heuristic, ant system, ant colony optimization, combinatorial optimization, Markov process.

1 Introduction

It is well–known that many optimization problems arising in computer science, engineering, management, administration or other fields cannot be solved exactly within reasonable time limits, at least not for problem instance sizes of practical interest. For this reason, heuristics have been invented to find high quality solutions for these problems in reasonable time. Some of them are not restricted to specific problem types, but may be applied, with suitable modifications, to a broad class of optimization problems. Often, these "general–purpose" algorithms are called *metaheuristics*; prominent examples are Simulated Annealing (SA) (see, e.g., [1]), Genetic Algorithms (GA) (see, e.g., [16, 18]), or Tabu Search (TS) (see, e.g., [12]). Also some traditional and well–established heuristic optimization techniques such as Random Search (RS), Local Search (LS) (see, e.g., [19], ch. 19), or the class of Greedy Heuristics (GH) (see, e.g., [19], section 12.3) may be considered as metaheuristics.

In a rough classification, we may distinguish between *iterative* heuristics, which start with a complete feasible solution and change this solution in an iterative process in order to improve the objective function value, and *constructive* heuristics, where a solution is generated "from scratch" by successive additions of certain elements or components, with or without backtracking (that is, removal of components that have been added at an earlier step). A typical representative of iterative heuristics is LS, where the changes of the current solution (which is always complete and feasible) follow a so–called neighborhood structure; a typical representative of constructive heuristics is GH, where the final solution is successively built up in a linear process without backtracking, governed by the "gains" of the components that are allowed to be added at a certain step.

In a second classification, we may distinguish between *single–run* heuristic procedures, where the algorithm stops as soon as a certain internal termination condition is satisfied (e.g., having reached a local optimum in the case of LS, or having finished the construction process in the case of GH), and *repetitive* heuristic procedures, where the user may control the amount of computation time he/she wants to invest, and where the quality of the solution improves as a function of the invested computation time. The second type of heuristics, which usually exploits *randomness*, includes heuristics such as GA or SA. Single–run heuristic procedures are typically fast, but often provide only moderate solution quality, whereas repetitive heuristic procedures may reach considerable solution quality at the price of (sometimes) very large runtime.

Combining both classifications, we get four possible combinations. Surprisingly, most metaheuristics fall into only three of these four combinations:

1. iterative, single–run (e.g., LS),

2. constructive, single–run (e.g., GH),

3. iterative, repetitive (e.g., SA or GA).

(A possible exception is Greedy Randomized Adaptive Search [10], which is a repetitive heuristic with constructive features. However, since it uses LS as an essential part, it should rather be considered as a hybrid between an iterative and a constructive heuristic.)

Considering that Greedy Heuristics are available for most practical optimization problems and often produce good results, it seems, in these cases, less expensive with regard to development costs to further improve their solution quality by extending them to *repetitive* procedures than to replace them by iterative heuristics which follow completely different optimization strategies. So it seems desirable to have a constructive and repetitive metaheuristic including GH as a special (boundary) case.

An essential step in this direction was the development of *Ant System* (AS) by Dorigo, Colorni, and Maniezzo (see [8, 4, 9]), a new type of heuristic inspired by analogies to the foraging behavior of real ant colonies, which has proven to work successfully in a series of experimental studies. Diverse modifications of AS have been applied to many different types of discrete optimization problems and have produced very satisfactory results (for a survey, see [5]). Recently, the approach has been extended by Dorigo and Di Caro to a full discrete optimization metaheuristic [6], called the Ant Colony Optimization (ACO) metaheuristic.

The aim of the present article is to show the theoretical soundness of the ant–based optimization approach by deriving a *convergence* result: under certain conditions, the current solution of the system converges, with a probability that can be made arbitrarily close to one, to the optimal solution. As a formal framework for our investigation, we introduce an ant–based metaheuristic, called *Graph–based Ant System*, which is somewhat more specific than the ACO metaheuristic, but suffices for covering the entire range of static combinatorial optimization problems.

The remainder of the paper is structured as follows: Section 2 gives a short introduction into the key ideas of ant–based optimization. In Section 3, our extended version of AS is presented. Section 4 presents the mathematical convergence result, and Section 5 contains final considerations and references to possible future work.

2 The Basic Ideas of Ant–Based Optimization

Before we start to describe Graph–based Ant System in formal terms, it may be helpful to provide the reader with an intuition of how ant algorithms work. For this purpose, it is convenient to trace ant–based optimization back to its roots and give a short outline of the historically first ant algorithm, *Ant System*, for the solution of the traveling salesman problem [8, 4]. We follow the presentation in [6].

Ant algorithms are multi–agent systems inspired by the behavior of colonies of real ants which has been studied by biologists. It is known that ant colonies are able to solve shortest–path problems in their natural environment by relying on a rather simple biological mechanism: each ant leaves information on which path it has traversed by depositing a chemical substance, called *pheromone*, on the ground. Ants have a tendency to follow these pheromone trails. Within a fixed period, shorter paths between nest and food can be traversed more often than longer paths, and so they obtain a higher amount of pheromone, which, in turn, tempts a larger number of ants to choose them and thereby to reinforce them again.

The (artificial) Ant System for solving the Traveling Salesman Problem, that is the problem of finding a shortest closed tour visiting each node of a given graph with given edge lengths exactly once, is based on very similar principles: first, the agents — as the "artificial ants" may be called — are positioned at some node of the graph. Then, each agent performs a series of random moves to neighbor nodes, controlled by suitably defined transition probabilities. As soon as an agent has visited all nodes, the length of the agent's tour is evaluated, and the "pheromone" values assigned to the arcs of his path are increased by an amount proportional to the quality of the tour. This procedure is repeated many times. As a counter–effect to the accumulation of pheromone, an "evaporation" mechanism reduces periodically the pheromone values by some given factor. The probability of a transition along a specific arc is computed from two parameters: (a) the pheromone value assigned to this arc, and (b) the length of the arc. The higher the pheromone value and the smaller the length, the higher is the probability that the agent follows this arc in his next move.

The success of the approach in solving the Traveling Salesman Problem has stimulated its transfer to a large number of other combinatorial optimization problems. For some of them, e.g., the quadratic assignment problem (see [17]), the feasible solutions have a similar structure (being representable as permutations) as those of the Traveling Salesman Problem, such that they can easily be interpreted as results of walks in a directed graph. The idea of AS may be adapted in these cases without essential changes. Nevertheless, in order to be able to treat arbitrary combinatorial optimization problems, the graph in which the agents perform their walks must be given a more general, abstract interpretation, and the transition mechanism must also be generalized. In the following section, we develop a formal framework for this purpose.

3 The Algorithm

Our extension of Ant System is based on the representation of a feasible solution as a walk in a directed graph, which we call the "construction graph":

Definition 3.1. Let an instance of a combinatorial optimization problem be given. By a *construction graph* for this instance we understand a directed graph $\mathcal{C} = (\mathcal{V}, \mathcal{A})$ together with a function Φ with the following properties:

(1) In $\mathcal{C}$, a unique node is marked as the so–called *start node*.

(2) Let $\mathcal{W}$ be the set of (directed) walks w in $\mathcal{C}$ satisfying the following conditions:

(i) w starts at the start node of $\mathcal{C}$.

(ii) w contains each node of $\mathcal{C}$ at most once.

(iii) The last node on w has no successor node in $\mathcal{C}$ that is not already contained in w (that is, w cannot be prolonged without violating (ii)).

Then Φ maps the set $\mathcal{W}$ onto the set of feasible solutions of the given problem instance. In other words: To each walk w satisfying (i) – (iii), there corresponds (via Φ) a feasible solution, and to each feasible solution, there corresponds (via Φ^{-1}) at least one walk satisfying (i) – (iii).

As it can be seen from this definition, a construction graph $(\mathcal{C}, \Phi)$ specifies a particular encoding of the feasible solutions as "walks". The objective function value of the walk is set equal to the objective function value of the corresponding feasible solution of the original problem. We assume throughout this paper that the optimization problem under consideration is formulated as a *minimization* problem, that is, the objective function represents *costs*. Usually, of course, there are several ways to design a construction graph for a given problem, and the choice of the construction graph may have great impact on the performance of the algorithm (cf. the discussion in Section 5).

Remark 3.1. For some applications, it is more convenient to extend Definition 3.1 by admitting the case that not all walks in the construction graph satisfying (i) – (iii) correspond (via Φ) to *feasible* solutions. Then, Φ is a function mapping only a *subset* of $\mathcal{W}$ onto the set of feasible solutions. Our formalism does not really change in this case, except that the cost value of a walk corresponding to an infeasible solution is set equal to infinity (or to a value larger than any cost of a feasible solution). Permitting such walks during the *execution* of the algorithm, however, would reduce the efficiency, so they should be "locked" by an operation described in component 5 below.

Remark 3.2. Using the extension of Definition 3.1 described in Remark 3.1, it is always possible to design a construction graph for the given problem instance with a number of nodes linear in the number of bits needed for the representation of a solution, and a number of arcs quadratic in this number of bits: see the procedure outlined in Remark 3.3 below. Sometimes, however, larger construction graphs will produce a better solution quality.

Let us now describe *Graph–based Ant System* as an extension of *Ant System* ([8, 4, 9]). Graph–based Ant System contains the following components:

1. A construction graph $(\mathcal{C}, \Phi)$ according to Definition 3.1.

2. A set $A_1, \dots, A_S$ of *agents* (in the literature usually called "ants"). Each agent performs a random walk with carefully chosen transition probabilities (see component 3 below) on the construction graph. In a multiprocessor system, the walk of each agent may be computed on a separate processor (cf. [3]). In a single–processor system, the moves of $A_1, \dots, A_S$ are computed sequentially. A time period in which each agent performs a walk (consisting of several single moves) through the construction graph will be called a *cycle*. An application of Ant System consists of several cycles $1, \dots, M$; the number M of cycles may be fixed in advance or be determined at a later time during the execution of the algorithm.

3. *Transition probabilities* for the random moves of the agents during each cycle. Let $u = (u_0, \dots, u_{t-1})$ denote the partial walk an agent has already traversed before its tth transition step in a fixed cycle m, where $u_0, \dots, u_{t-1}$ are node indices in the construction graph (u_0 referring to the start node). We write $l \in u$ if node l is contained in the partial walk u, and $l \notin u$ otherwise. Moreover, let $\mathcal{A}$ be the set of arcs in the construction graph. Then the general form of the transition probabilities is

$$p_{kl}(m,u) = \frac{[\tau_{kl}(m)]^\alpha \cdot [\eta_{kl}(u)]^\beta}{\sum_{r \notin u, (k,r) \in \mathcal{A}} [\tau_{kr}(m)]^\alpha \cdot [\eta_{kr}(u)]^\beta}, \tag{1}$$

if $l \notin u$ and $(k,l) \in \mathcal{A}$, and

$$p_{kl}(m,u) = 0 \tag{2}$$

otherwise. Therein, $p_{kl}(m,u)$ denotes the probability that a fixed agent having already traversed a partial walk $u = (u_0, \dots, u_{t-2}, u_{t-1} = k)$ in the current cycle m, moves from node k (its current position) to node l. (Note that this probability is only defined if $k = u_{t-1}$.) The numbers $\tau_{kl}(m)$ are called "pheromone values" (see component 4 below), and the numbers $\eta_{kl}(u)$ are called "desirability values" (see component 5 below). α and β are parameters.

At the beginning of each cycle, each agent is positioned at the start node of the construction graph. In the tth transition step of the current cycle, each agent A_s performs a single random move in the construction graph, controlled by the transition probabilities $p_{kl}(m,u)$, where u is the partial walk A_s has already traversed (that is, each agent has a particular u). If, for a fixed agent A_s, $p_{kl}(m,u) = 0$ holds for all nodes l before the tth transition step, then agent A_s has completed his walk in the current cycle m. According to Definition 3.1, this walk determines a solution of the given optimization problem.

4. An array of *pheromone values* τ_{kl}, where τ_{kl} is assigned to arc (k,l) in the construction graph. The pheromone values usually change from cycle to cycle (see below), so their dependence on the cycle index m can be represented in the form $\tau_{kl}(m)$. At the beginning of cycle 1, we set $\tau_{kl} = 1/(\text{number of arcs})$ for each arc (k,l). At the end of each cycle m ($m = 1, 2, \dots, M-1$), the following update rule is applied. First, for each agent A_s and each arc (k,l), a value $\Delta\tau_{kl}^{(s)}$ is determined as a function of the

solution assigned to the walk of A_s in the current cycle m. Suppose this solution has a cost value (objective function value) f_s. We set for each arc (k, l):

$$\Delta\tau_{kl}^{(s)} = \begin{cases} \varphi(f_s), & \text{if agent } A_s \text{ has traversed arc } (k,l), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Therein, φ is a nonincreasing function which may depend on the walks of the agents in the cycles $1, \ldots, m-1$. Let

$$C = \sum_{(k,l)\in\mathcal{A}} \sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}. \tag{4}$$

Now, if $C = 0$, we set

$$\tau_{kl}(m+1) = \tau_{kl}(m)$$

for all arcs (k, l), that is, the values τ_{kl} are the same in cycle $m + 1$ as in cycle m. If, on the other hand, $C > 0$, we set

$$\tau_{kl}(m+1) = (1-\rho)\tau_{kl}(m) + \rho\Delta\tau_{kl}, \tag{5}$$

where

$$\Delta\tau_{kl} = \frac{1}{C}\sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}. \tag{6}$$

The number ρ is usually called the *evaporation factor* (see [9]). It is easily verified from (5), (6) and (4) that the sum of pheromone values, $\sum_{(k,l)\in\mathcal{A}} \tau_{kl}(m)$, always remains equal to one. (Forcing the sum of the values $\tau_{kl}(m)$ to be one is not done in other ant–based algorithms, and it is not essential for our convergence result below. It serves simply as a re–normalization which favors the numerical stability of the algorithm.)

The pheromone update rule above may be interpreted as follows: If no walk is rewarded, everything remains constant. Otherwise, due to evaporation, only an amount of $1 - \rho$ of pheromone remains on the graph after cycle m. The remaining amount of ρ is the "budget" for rewarding the walks traversed in cycle m according to their respective objective function values. Each agent A_s can be imagined to report by which "bonus" $\Delta\tau_{kl}^{(s)}$ he wants to reward his walk in cycle m. The *actual* pheromone increments result by distributing the total budget of ρ proportionally to the reported values $\Delta\tau_{kl}^{(s)}$.

By the described update rule, the pheromone on "successful" arcs of the construction graph is increased, such that they are traversed more often by the agents in the future. By setting $\rho = 0$, the influence of the cost function on the walks of the agents is switched off. If, in addition, the influence of the desirability values $\eta_{kl}(u)$ is switched off by setting $\beta = 0$, then we obtain pure *Random Search*, which, in this way, turns out to be a special case of Graph–based Ant System.

5. An array of *desirability values* η_{kl}, where η_{kl} is assigned to arc (k, l) in the construction graph. The desirability values may depend on the partial walk $u = (u_0, \ldots, u_{t-2}, u_{t-1} = k)$ the current agent has already traversed, that is, on the whole "history", so they

can be written as $\eta_{kl} = \eta_{kl}(u)$. Typically, the value $\eta_{kl}(u)$ is obtained from a *Greedy Heuristic* for the combinatorial optimization problem under consideration; in this view, it may also be interpreted as the value of a so–called *greedy function* (see [10]): Assume that a GH is given. It specifies the stepwise construction of a "good" (but in general not optimal) solution of the problem. In our formalism, this construction can be represented by a walk on the construction graph. The GH defines "weights" for all feasible arcs (k, l) leaving node k, and determines the next node l of the walk by the "greedy principle" that the weight of (k, l) is maximum. We may now consider the weight of arc (k, l) as the "desirability" of the transition from node k to node l, that is, we may set $\eta_{kl}(u) =$ weight (k, l). Alternatively, another way of defining the desirability values may also be chosen: Set $\eta_{kl}(u) = 1$, if weight(k, l) is maximum among all successor nodes of node k, and $\eta_{kl}(u) = 0$ otherwise.

The values $\eta_{kl}(u)$ can also be used for preventing walks corresponding to infeasible solutions, if the extended version of Definition 3.1 according to Remark 3.1 is used: If w is such a walk, let u be the longest partial walk (starting at the start node) on w that can still be prolonged to a feasible walk (i.e, that did not contradict a feasibility condition so far), and let (k, l) be the first arc on w not belonging to u. Then w can be "locked" (without locking any feasible walk) by setting $\eta_{kl}(u) = 0$.

In our formalism presented above, Ant Systems are natural stochastic generalizations of Greedy Heuristics: Note that if the parameter α is set equal to zero, and the alternative way described above of defining the desirability values is chosen, the behavior of the agents is exclusively governed by the greedy principle, so GH is another special case of Graph–based Ant System.

Remark 3.3. It should be obvious that Graph–based Ant System is applicable to all combinatorial optimization problems with a finite solution space per problem instance: the construction graph could simply consist of a start node, a termination node, a specific node v_x for each feasible solution x, and arcs from the start node to each v_x and from each v_x to the termination node. Of course, this implementation would be highly inefficient. A more efficient alternative is the following:

(a) Encode each feasible solution by a binary string of fixed length (as it is done in the classical GA approach);

(b) design a construction graph with (i) a start node, (ii) a termination node, (iii) a completely interconnected subgraph, containing a node for each possible bit position (the visited nodes are then the 1–bits), (iv) arcs leading from the start node to each other node, and (v) arcs leading from each other node to the termination node;

(c) exclude infeasible binary strings by locking the corresponding walks via the process described in component 5 above.

In continuous optimization and sometimes also in dynamic optimization, the solution space is not finite anymore. Such problems are not within the area of application of Graph–based Ant System.

Remark 3.4. Recently, Dorigo and Di Caro [6] have developed the concept of the *ACO metaheuristic*, which allows the treatment of a very broad range of optimization problems by an ant–based approach. The variety of features incorporated into this metaheuristic makes it extremely flexible in application, and it is just because of this flexibility and generality why theoretical results such as the one proven here are hard to obtain for the ACO metaheuristic in total. Graph–based Ant System described above is more specific and hence less powerful, but in designing it, we have been very careful not to be too restrictive as far as the range of possible applications is concerned. A compromise between descriptive power and theoretical tractability had to be found. Let us mention some features of the ACO metaheuristic (ACO–MH) lacking in Graph–based Ant System (GAS), and shortly outline why we omitted these features in our framework:

1. Contrary to ACO–MH, *time–dependent evaluation* of the costs assigned to a walk, as it is convenient for some dynamic optimization problems (e.g., the routing problem), is not supported by GAS. In GAS, we have decided to restrict ourselves to static combinatorial optimization.

2. ACO–MH allows the formulation of *explicit constraints* for walks to be feasible. Since the same effect can also be obtained by the "locking"–construction described in component 5 of GAS, we have dispensed with this option.

3. In ACO–MH, agents can (a) *start their walks* not only at a certain start node, but at *arbitrary nodes*, and (b) *terminate them* triggered by *arbitrary conditions*. The more specific assumptions of GAS can be circumvented (a) by adding a unique dummy start node, and (b) by adding a dummy termination node and forcing conditional termination via the values $\eta_{kl}(u)$.

4. ACO–MH allows an (optional) *step–by–step pheromone update*, which is not supported by GAS, since global update is more usual in ant–based approaches.

5. Also with regard to the *delayed pheromone update*, as it is supported both by ACO–MH and GAS, the ACO–MH version is more flexible: different arcs belonging to the current walk may obtain different pheromone increments. However, as remarked in [6] (footnote 8), most ACO implementations do not exploit this possibility, so it seemed dispensable.

6. So–called *daemon actions* (centralized actions not assigned to specific agents) are not supported by GAS. We had to omit this feature because it enables hybrids between ant–based and other optimization techniques, whereas the aim of this article is to investigate the capabilities of a *pure* ant–based strategy.

7. The *scheduling of activities* is more flexible in ACO–MH than in GAS. Let us emphasize, however, that also GAS does not restrict the scheduling of the walks of the agents within a cycle: both serial and parallel (or even mixed) execution is allowed.

4 Convergence

In this section it will be shown that under some conditions, the current solutions of Graph–based Ant System (the feasible solutions corresponding to the walks of agents $A_1, \ldots, A_S$ in cycle m) converge with a probability that can be made arbitrarily close to one to the *optimal* solution. The conditions are the following:

(a) The parameter α in Equation (1) is chosen as $\alpha = 1$.

(b) There is only one optimal walk in $\mathcal{W}$, that is, the optimal solution is unique, and it is encoded by only one walk in $\mathcal{W}$.

(c) Along the optimal walk w^*, the desirability values satisfy $\eta_{kl}(u) > 0$ for all arcs (k, l) of w^* and the corresponding partial walks u of w^*.

(d) Let $f^* = f^*(m)$ be the lowest cost value observed in the cycles $1, \ldots, m-1$, that is, the lowest objective function value f_s corresponding to a walk of an agent A_s in these $m-1$ cycles. (It is easy to store and update the current value of f^* during the execution of the algorithm.) In the case of cycle $m = 1$, let $f^* = \infty$. Let the function φ chosen for the definition of the values $\Delta\tau_{kl}^{(s)}$ at the beginning of cycle $m+1$ (see eq. (3)) have the following properties:

    (i) $\varphi(f_s) > 0$ for $f_s \leq f^*$,

    (ii) $\varphi(f_s) = 0$ for $f_s > f^*$.

    In other words: Only walks that are at least as good as the best found walk up to now get a positive increment $\Delta_{kl}^{(s)}$. For positive cost functions, e.g., $\varphi(f_s)$ may be chosen as $\varphi(f_s) = 1/f_s$ if $f_s \leq f^*$ and $\varphi(f_s) = 0$ otherwise. Note that φ is allowed to depend on the "history" (the cycles $1, \ldots, m-1$), and hence in particular on the value f^*.

Let us briefly discuss the conditions (a) – (d). Condition (a) has merely technical reasons; it allows an easier treatment of the "normalization factor" in the denominator at the right hand side of Equation (1). It does not imply an essential restriction, since the main purpose of the parameters α and β is not to "shape" pheromone values and desirability values (this could be done by other means), but to have easy control on their impact relative to each other; so we may fix one of these two parameters as long as the other remains free.

Condition (b) is more restrictive, but we have preliminary results indicating that it can be dropped. Because of their mathematical complexity, we must defer this topic to a later publication [15].

Condition (c) may appear nasty, because it seems to require the knowledge of the optimal walk. In practice, however, this condition makes no problems at all; it can easily be satisfied by arbitrarily slight changes of the desirability values: for each *feasible* continuation (k, l) of a partial walk u with $\eta_{kl}(u) = 0$, replace $\eta_{kl}(u) = 0$ by $\eta_{kl}(u) = \delta$ with a small $\delta > 0$. Let us remark that as long as global optimization is intended, it makes sense anyway not to exclude any feasible continuation of a partial walk *a priori*. — Although not very restrictive, condition (c) is essential because otherwise it may happen that a specific arc belonging to the optimal walk is inaccessible for the agents, just because of an unlucky choice of the desirability values.

Condition (d), finally, is a version of what is called *elitist strategy* in prior publications on Ant System (see [9]): Only the best walks are rewarded; walks that are dominated by another already traversed walk do not get pheromone increments anymore. The first ant–based systems where only the best tour found so far is rewarded by pheromone increments, were Ant-Q [13] by Gambardella and Dorigo (they call this update strategy "Global–best") and ACS [7] by the same authors. Whether a combination with a "non–elitist" pheromone update strategy, as it has been used in diverse experimental studies, destroys the theoretical convergence property or not, is another open problem.

Our main result is the following:

Theorem 4.1. Let conditions (a) – (d) be satisfied, and let P_m denote the probability that a fixed agent, say agent A_1, traverses the optimal walk in cycle m. Then the following two assertions are valid:

1. For each $\epsilon > 0$ and for fixed parameters ρ and β, it can be achieved by the choice of a sufficiently large number S of agents that $P_m \geq 1 - \epsilon$ holds for all $m \geq m_0$ (with an integer m_0 depending on ϵ).

2. For each $\epsilon > 0$ and for fixed parameters S and β, it can be achieved by the choice of an evaporation factor ρ sufficiently close to zero that $P_m \geq 1 - \epsilon$ holds for all $m \geq m_0$ (with an integer m_0 depending on ϵ).

Let us briefly outline the plan of the proof:

- First (Proposition 4.1), we show that the search procedure of Graph–based Ant System can be understood as a *Markov process*, that is, a stochastic process where the probability distribution of the state at cycle m only depends on the state at time $m-1$. Usually, a probabilistic algorithm admits different valid descriptions in terms of a stochastic process (cf. Remark 4.2 below); by establishing the Markov process interpretation, we fix the mathematical framework for the notions introduced in the proof and give them an unambiguous meaning from the viewpoint of probability theory.

- Next, a lower bound for the probability that at least one agent traverses the optimal walk in a fixed cycle m is derived (Lemma 4.1). Closer inspection shows (Corollary) that the obtained lower bound, although depending on m, holds independently of the "history" of the process (the events in the previous cycles).

- Then, it is shown that, provided that the optimal walk is traversed at least once by some agent at some time, the pheromone values assigned to the arcs of the optimal walk get closer and closer to 1/(length of the optimal walk), whereas the pheromone values assigned to other arcs tend to zero (Lemma 4.2).

- Again under the condition that the optimal walk is traversed at least once, the last–mentioned phenomenon has the consequence that the computed transition probability values responsible for the traversal of the arcs of the optimal walk get closer and closer to unity (Lemma 4.3 and Corollary).

- Still under the mentioned condition, it follows that the probability for the event that a *fixed* agent traverses the optimal walk gets closer and closer to unity during the execution of the algorithm (Lemma 4.4).

- Finally, the entire theorem is proven by combining the last observation with an estimation of the probability that *no* agent will ever traverse the optimal walk: using the lower bound of Lemma 4.1, it turns out that this probability can be made arbitrarily small either by increasing the number of agents or by decreasing the evaporation factor.

It may be useful to indicate where our convergence conditions (a) – (d) enter into the proof: Condition (a) is applied merely as a simplification of the pheromone update rule in the proof of Lemma 4.1. By condition (b), the proof can be reduced to an investigation of what happens on the unique optimal walk. Condition (c) enables the lower bound estimation of Lemma 4.1. Condition (d), finally, is used to avoid premature convergence to a suboptimal solution.

It should be mentioned that our proof also works in the special case $\beta = 0$, that is, the case where the values $\eta_{kl}(u)$ are not used at all (cf. Remark 4.3 below). The case $\rho = 0$, on the other hand, must be excluded: for Lemma 4.1, $\rho > 0$ is required. So our result does not cover Random Search.

As announced, we start by interpreting the iterative solution process of Graph–based Ant System as a *Markov process* (see, e.g., [11]) in discrete time. The states of this Markov process are the triples

$$(\underline{\tau}(m), \underline{w}(m), f^*(m)) \quad (m = 1, 2, \ldots),$$

where

- $\underline{\tau}(m)$ is the vector of the pheromone values $\tau_{kl}(m)$ for all arcs (k, l) during cycle m,

- $\underline{w}(m)$ is the vector of the walks $w^{(s)}(m)$ $(s = 1, \ldots, S)$ of the agents $A_1, \ldots, A_S$ in cycle m,

- $f^*(m)$ is the best found cost value corresponding to the walk of any agent in cycle $1, \ldots, m-1$ (that is, the value f^* controlling the update of the pheromone values at the end of cycle m according to condition (d) at the beginning of this section). For cycle $m = 1$, we set $f^*(1) = \infty$.

Proposition 4.1. The state variables $(\underline{\tau}(m), \underline{w}(m), f^*(m))$ $(m = 1, 2, \ldots)$ form a Markov process.

Proof. The Markov property is satisfied if the distribution of the state in cycle m, $(\underline{\tau}(m), \underline{w}(m), f^*(m))$, only depends on the state in cycle $m-1$, $(\underline{\tau}(m-1), \underline{w}(m-1), f^*(m-1))$. This is indeed the case, since the state transitions are given as follows:

- $\underline{\tau}(m)$ results deterministically from $\underline{\tau}(m-1)$, $\underline{w}(m-1)$ and $f^*(m-1)$ according to the update rule for the pheromone values.

- The probability distribution of $\underline{w}(m)$ only depends on $\underline{\tau}(m)$ (the values $\eta_{kl}(u)$ are deterministic!) and is hence determined by $(\underline{\tau}(m-1), \underline{w}(m-1), f^*(m-1))$.

- $f^*(m)$ results deterministically from $\underline{w}(m-1)$ and $f^*(m-1)$.

□

Remark 4.1. The reader should notice that in the given Markov process interpretation, the numbers $p_{kl}(m,u)$ defined by (1) are functions of $\underline{\tau}(m)$ and therefore also functions of the state of the process in cycle $m-1$. In particular, the numbers $p_{kl}(m,u)$ are *random variables.* From this point of view, their interpretation as probabilities is only an indirect one: by the state transition rule of the considered Markov process, they determine the distribution of the vector $\underline{w}(m)$ of the walks in cycle m.

Remark 4.2. We could describe the search process also by a non–Markovian stochastic process, e.g., by capturing the current state by the triple $(\underline{\tau}(m), \underline{w}(m), f_{opt}(m))$ instead of $(\underline{\tau}(m), \underline{w}(m), f^*(m))$, where $f_{opt}(m)$ denotes the best found cost value *in cycle* m. Our choice, however, facilitates the formulation of the proof. For example, in the proof of Lemma 4.4 below, only the state in cycle $m'-1$ has to be considered instead of the whole history of the process.

In the sequel, the following abbreviations shall be used:

- w^* denotes the (unique) optimal walk.

- L denotes the length (number of arcs) of w^*.

- Pr is written for the probability measure on the Markov process defined above.

- $E_m^{(s)}$ denotes the event that $w^{(s)}(m) = w^*$, that is, the event that agent A_s traverses in cycle m the optimal walk.

- B_m is an abbreviation for $\neg E_m^{(1)} \wedge \ldots \wedge \neg E_m^{(S)}$, that is, for the event that $w^{(s)}(m) \neq w^*$ for all $s = 1, \ldots, S$ (the event that no agent traverses the optimal walk in cycle m).

- F_m is an abbreviation for $B_1 \wedge \ldots \wedge B_{m-1} \wedge \neg B_m$, that is, for the event that the optimal walk is traversed by some agent in cycle m, but by no agent in the cycles $1, \ldots, m-1$. Obviously, the events $F_1, F_2, \ldots$ are mutually exclusive.

- A is an abbreviation for $F_1 \vee F_2 \vee \ldots$, that is, for the event that there is an m and an s such that $w^{(s)}(m) = w^*$ (the event that the optimal walk is traversed by some agent in some cycle).

Furthermore, the notation $(k,l) \in w$ shall be used for the assertion that arc (k,l) lies on the walk w, and a walk w shall sometimes also be denoted by the sequence of nodes lying on it.

Because of condition (c) at the beginning of this section and the fact that there are only finitely many arcs (k,l) and only finitely many feasible partial walks u, we have

$$\gamma = \min \{[\eta_{kl}(u)]^\beta \mid (k,l) \in w^*,\ u \text{ partial walk of } w^*\} \ > \ 0 \tag{7}$$

and

$$\Gamma = \max\, [\eta_{kl}(u)]^\beta < \infty.$$

(Note that if there were infinitely many arcs or walks, the minimum resp. maximum could be undefined, and the infimum resp. supremum could be 0 resp. ∞.) Multiplying all desirability values $\eta_{kl}(u)$ by a fixed constant does not change the transition probabilities (1). So it can be assumed without loss of generality that the values $\eta_{kl}(u)$ are normalized in such a way that $\Gamma = 1$, that is,

$$[\eta_{kl}(u)]^\beta \le 1 \tag{8}$$

for all arcs (k,l) and all partial walks u.

Lemma 4.1. The probability $Pr(\neg B_m)$ that at least one agent traverses the optimal walk in cycle m is larger or equal to $1-(1-c^{m-1}p)^S$, where $c = (1-\rho)^L$ and $p = \gamma^L \prod_{(k,l)\in w^*} \tau_{kl}(1)$ with γ defined by (7).

Proof. Since $\Delta\tau_{kl} \ge 0$ and $\rho > 0$, eq. (5) implies

$$\tau_{kl}(m+1) \ge (1-\rho)\, \tau_{kl}(m) \tag{9}$$

in the case $C > 0$, and again because of $\rho > 0$, this holds also in the case $C = 0$. Repeated application of (9) yields

$$\tau_{kl}(m) \ge (1-\rho)^{m-1}\, \tau_{kl}(1). \tag{10}$$

Because of (8) and $\sum_{(k,l)} \tau_{kl}(m) = 1$,

$$\sum_{r\notin u,(k,r)\in\mathcal{A}} \tau_{kr}(m) \cdot [\eta_{kr}(u)]^\beta \ \le \ \sum_{r\notin u,(k,r)\in\mathcal{A}} \tau_{kr}(m) \ \le \ 1.$$

Therefore, the transition probabilities $p_{kl}(m,u)$ given by (1) satisfy, for a node l with $l \notin u$, the inequality

$$p_{kl}(m,u) = \frac{\tau_{kl}(m)\, [\eta_{kl}(u)]^\beta}{\sum_{r\notin u,(k,r)\in\mathcal{A}} \tau_{kr}(m)\, [\eta_{kr}(u)]^\beta} \ \ge \ \tau_{kl}(m)\, [\eta_{kl}(u)]^\beta. \tag{11}$$

Let $w^* = (v_0, \ldots, v_L)$. Then by (11), (7), and (10),

$$Pr(E_m^{(s)}) = \prod_{i=0}^{L-1} p_{v_i v_{i+1}}(m,(v_0,\ldots,v_i)) \ \ge \ \prod_{i=0}^{L-1} \tau_{v_i v_{i+1}}(m)\, [\eta_{v_i v_{i+1}}]^\beta \ \ge \ \gamma^L \prod_{i=0}^{L-1} \tau_{v_i v_{i+1}}(m)$$

$$\ge \gamma^L \prod_{i=0}^{L-1} (1-\rho)^{m-1}\, \tau_{v_i v_{i+1}}(1) \ = \ \gamma^L\, (1-\rho)^{L(m-1)} \prod_{(k,l)\in w^*} \tau_{kl}(1) \ = \ c^{m-1} p.$$

Since the walks of the S agents are independent, this implies

$$Pr(B_m) \leq (1 - c^{m-1}p)^S,$$

whence the assertion follows. $\square$

Corollary. Also the *conditional* probability $Pr(\neg B_m \mid B_1 \wedge \ldots \wedge B_{m-1})$ that at least one agent traverses in cycle m the optimal walk, given that no agent has traversed the optimal walk in one of the previous cycles, is larger or equal to $1 - (1 - c^{m-1}p)^S$.

Proof. The proof is a repetition of the proof of Lemma 4.1. Inequality (9) and therefore also inequality (10) hold in *any* case, independently of what happens in the cycles $0, \ldots, m-1$, and therefore also independently of the condition $B_1 \wedge \ldots \wedge B_{m-1}$.

$\square$

In the following lemmas, assertions on conditional probabilities, conditional on the event F_m, shall be made. We denote these conditional probabilities in the usual way by writing $Pr\{\ldots | F_m\}$.

Lemma 4.2. For each $\epsilon > 0$ and each $m \in \mathbb{N}$ there is an integer $d(\epsilon, m) \in \mathbb{N}$ such that

$$Pr\left\{|\tau_{kl}(m') - 1/L| < \epsilon \text{ for all } (k,l) \in w^* \mid F_m\right\} \geq 1 - \epsilon$$

for all $m' \geq m + d(\epsilon, m)$, and

$$Pr\left\{\tau_{kl}(m') < L\epsilon \text{ for all } (k,l) \notin w^* \mid F_m\right\} \geq 1 - \epsilon$$

for all $m' \geq m + d(\epsilon, m)$.

Proof. We restrict ourselves to the event F_m, that is, to the event where the optimal walk w^* is traversed for the first time in cycle m. Consider a cycle $m' > m$. There are two possible cases:
Case (a): At the end of cycle m', $C = 0$. Then $\tau_{kl}(m'+1) = \tau_{kl}(m')$ for all arcs (k,l).
Case (b): At the end of cycle m', $C > 0$. This is only possible if $\Delta\tau_{kl}^{(s)} > 0$ for some agent A_s in cycle m'. In view of (3) and condition (d) at the beginning of this section, this means that some agent A_s has traversed a walk with corresponding cost of at most $f^*(m')$. Because of $m' > m$, $f^*(m')$ is already the minimal cost value f^* (the cost value corresponding to w^*). So case (b) implies that at least one agent A_s has traversed w^* in cycle m'. Without loss of generality, assume that just the agents A_s for $s = 1, \ldots, S'$ $(1 \leq S' \leq S)$ have traversed w^* in cycle m'. Then for an arc $(k,l) \notin w^*$, $\Delta\tau_{kl}^{(s)} = 0$ and hence also $\Delta\tau_{kl} = 0$ is set at the end of this cycle, whereas for an arc $(k,l) \in w^*$,

$$\Delta\tau_{kl}^{(s)} = \varphi(f^*) \quad (s \leq S')$$

and

$$\Delta\tau_{kl}^{(s)} = 0 \quad (s > S').$$

We obtain

$$C = \sum_{(k',l')\in\mathcal{A}} \sum_{s=1}^{S'} \Delta\tau_{k'l'}^{(s)} = \sum_{(k',l')\in w^*} S' \cdot \varphi(f^*) = L \cdot S' \cdot \varphi(f^*).$$

Hence for $(k,l) \in w^*$,

$$\Delta\tau_{kl} = \frac{1}{C} \sum_{s=1}^{S'} \Delta\tau_{kl}^{(s)} = \frac{S' \cdot \varphi(f^*)}{L \cdot S' \cdot \varphi(f^*)} = \frac{1}{L}, \tag{12}$$

and therefore

$$\tau_{kl}(m'+1) = (1-\rho)\,\tau_{kl}(m') + \rho/L \tag{13}$$

or

$$\tau_{kl}(m'+1) - \tau_{kl}(m') = \rho\,(1/L - \tau_{kl}(m')). \tag{14}$$

In particular: If $\tau_{kl}(m') < 1/L$, then $\tau_{kl}(m'+1) > \tau_{kl}(m')$. If, on the other hand, $\tau_{kl}(m') \geq 1/L$, then (because of (13)) $\tau_{kl}(m'+1) \geq 1/L$. In other words: in case (b), $\tau_{kl}(m')$ (for $(k,l) \in w^*$) either increases, or it stays above $1/L$.

Now combine case (a) and case (b). We obtain the assertion that for $(k,l) \in w^*$ and $m' > m$,

$$\tau_{kl}(m') \;\geq\; \min\left(\tau_{kl}(m+1),\, 1/L\right) \;\geq\; \alpha_{kl}(m) > 0,$$

where $\alpha_{kl}(m)$ is defined as $(1-\rho)^m\,\tau_{kl}(1)$.

Let now $(k,l) \in w^*$, and let $u = (u_0, \ldots, u_{t-1} = k)$ be the partial walk from the start node to node k on w^*. By (11) and (7),

$$p_{kl}(m', u) \geq \gamma\,\tau_{kl}(m') \geq \gamma\,\alpha_{kl}(m).$$

So, for a fixed agent A_s and $m' > m$,

$$Pr(E_{m'}^{(s)} \mid F_m) \geq \prod_{(k,l)\in w^*} \gamma\,\alpha_{kl}(m) = \gamma^L \prod_{(k,l)\in w^*} \alpha_{kl}(m) = a(m) > 0,$$

the number $a(m)$ being independent of m'.

By the same argument as in the proof of the Corollary to Lemma 4.1, it follows that the estimation above holds irrespectively of what has happened in cycle $m+1, m+2, \ldots, m'-1$, that is, it is also valid if the considered probability is additionally conditioned on any possible event in any of these cycles. Straightforward considerations show then that the probability (conditional on F_m) that no agent traverses w^* in g fixed subsequent cycles after cycle m is smaller or equal to $(1-a(m))^{S\cdot g}$. By choosing g large enough, this probability can be made arbitrarily small. As an immediate consequence, the probability (conditional on F_m) that case (b) above occurs less than h times between cycle m and cycle $m+d$ (that is, that event $\neg B_{m'}$ occurs for less than h indices m' between m and $m+d$) can be made arbitrarily small by choosing d large enough: Choose, for $0 < \delta < 1/2$, the number $g(\delta)$ such that

$$Pr\,\{\text{no agent traverses } w^* \text{ in } g(\delta) \text{ fixed successive cycles with indices } \geq m \mid F_m\} \;\leq\; \delta.$$

Now, for any $0 < \epsilon < 1$, set $\delta = \epsilon/(2h) < 1/2$. The probability (conditional on F_m) that case (b) occurs at least once in each of h successive periods, each period consisting of $g(\delta)$

successive cycles, is then larger than $(1-\delta)^h \geq 1 - 2\delta h = 1 - \epsilon$. So at least with probability $1-\epsilon$ (conditional on F_m), case (b) occurs at least h times in these $d = g(\delta) \cdot h$ cycles.

Now choose $h = h(\epsilon)$ in such a way that $(1-\rho)^h < \epsilon$, and $d = d(\epsilon, m)$ in such a way that

$$Pr\,\{\,\text{case (b) occurs less than } h \text{ times between cycle } m \text{ and cycle } m+d \mid F_m\,\} \;<\; \epsilon.$$

Consider an index m' for which case (b) holds. Then by (13),

$$\tau_{kl}(m'+1) - \frac{1}{L} = (1-\rho)\,\left(\tau_{kl}(m') - \frac{1}{L}\right)$$

for $(k,l) \in w^*$, that is, for such an arc, the distance between $\tau_{kl}(m')$ and $1/L$ decreases each time when case (b) occurs by the factor $1-\rho$ (and remains constant each time when case (a) occurs). By repeated application, we obtain that with a probability (conditional on F_m) larger or equal to $1-\epsilon$,

$$\left|\tau_{kl}(m+d) - \frac{1}{L}\right| \leq (1-\rho)^h \,\left|\tau_{kl}(m) - \frac{1}{L}\right| \leq (1-\rho)^h < \epsilon.$$

Hence with a probability (conditional on F_m) larger or equal to $1-\epsilon$, also

$$\left|\tau_{kl}(m') - \frac{1}{L}\right| \leq \left|\tau_{kl}(m+d) - \frac{1}{L}\right| < \epsilon$$

for all $(k,l) \in w^*$ and for arbitrary $m' \geq m+d$, which proves the first part of the lemma.

The second part follows from the equation

$$\sum_{(k',l') \in \mathcal{A}} \tau_{k'l'}(m') = 1 :$$

According to the first part, the sum of the pheromone values on the optimal walk w^* is, with probability (conditional on F_m) of at least $1-\epsilon$, larger than

$$L \cdot \left(\frac{1}{L} - \epsilon\right) = 1 - L\epsilon$$

for $m' \geq m + d(\epsilon, m)$. So an arc not lying on w^* cannot have an pheromone value larger than $L\epsilon$. □

Lemma 4.3. Let $u^*(k)$ denote the partial walk on w^* leading to node k ($k \in w^*$). Then for each $\epsilon > 0$ and each $m \in \mathbb{N}$ there is an integer $d'(\epsilon, m) \in \mathbb{N}$ such that

$$Pr\,\{\,p_{kl}(m', u^*(k)) \geq 1-\epsilon \text{ for all } (k,l) \in w^* \mid F_m\,\} \;\geq\; 1-\epsilon$$

for all $m' \geq m + d'(\epsilon, m)$.

Proof. By Lemma 4.2, for each $m' \geq m + d(\tilde{\epsilon}, m)$, with a probability (conditional on F_m) of at least $1-\tilde{\epsilon}$,

$$\left|\tau_{kl}(m') - \frac{1}{L}\right| \leq \tilde{\epsilon} \quad \text{for all } (k,l) \in w^* \tag{15}$$

and

$$\tau_{kr}(m') \leq L\tilde{\epsilon} \quad \text{for all } (k,r) \notin w^*. \tag{16}$$

Let $(k,l) \in w^*$, and let $u = u^*(k)$. We obtain

$$p_{kl}(m',u) = \frac{\tau_{kl}(m')\,[\eta_{kl}(u)]^\beta}{\sum_{r\notin u, r\neq l, r\in\mathcal{A}} \tau_{kr}(m')\,[\eta_{kr}(u)]^\beta + \tau_{kl}(m')\,[\eta_{kl}(u)]^\beta}.$$

Set $\eta = [\eta_{kl}(u)]^\beta > \gamma$ for abbreviation ($\eta \leq 1$ according to (8)). With ν denoting the maximal outdegree of a node in $\mathcal{C}$, the eqs. (15) and (16) imply

$$p_{kl}(m',u) \;\geq\; \frac{(1/L - \tilde{\epsilon})\,\eta}{\nu L\tilde{\epsilon} + (1/L + \tilde{\epsilon})\,\eta} = \frac{1 - L\tilde{\epsilon}}{1 + \tilde{\epsilon}\,(\nu L^2/\eta + L)}.$$

Since $(1+x)^{-1} \geq 1 - x$ for $x \geq 0$,

$$p_{kl}(m',u) \;\geq\; (1 - L\tilde{\epsilon})\,(1 - \tilde{\epsilon}\,(\nu L^2/\eta + L)) \;\geq\; 1 - (2L + \nu L^2/\eta)\,\tilde{\epsilon} \;\geq\; 1 - (2L + \nu L^2/\gamma)\,\tilde{\epsilon}.$$

The assertion follows by setting

$$\tilde{\epsilon} = \frac{\epsilon}{2L + \nu L^2/\gamma} < \epsilon.$$

□

Corollary. With the notation in Lemma 4.3, let

$$Y_{m'} = \prod_{(k,l)\in w^*} p_{kl}(m', u^*(k)). \tag{17}$$

Then, for each $\epsilon > 0$ and each $m \in \mathbb{N}$, there is an integer $d''(\epsilon, m) \in \mathbb{N}$ such that

$$Pr\,\{\, Y_{m'} \geq 1 - \epsilon \,|\, F_m \,\} \;\geq\; 1 - \epsilon$$

for all $m' \geq m + d''(\epsilon, m)$.

Proof. Follows immediately from Lemma 4.3: replacing ϵ by $\epsilon/(2L)$, we obtain that, except with a probability (conditional on F_m) of at most $\epsilon/(2L) \leq \epsilon$, each factor in (17) can be made larger or equal to $1 - \epsilon/(2L)$, and since $\epsilon/(2L) \leq 1/2$ (note that without loss of generality, $\epsilon \leq 1$), we have

$$\left(1 - \frac{\epsilon}{2L}\right)^L \geq 1 - 2 \cdot \frac{\epsilon}{2L} \cdot L = 1 - \epsilon.$$

□

Lemma 4.4. For each $\epsilon > 0$ there is an integer $d'''(\epsilon, m) \in \mathbb{N}$, such that for fixed s,

$$Pr\left(E_{m'}^{(s)} \,|\, F_m\right) \;\geq\; 1 - \epsilon$$

for all $m' \geq m + d'''(\epsilon, m)$.

Proof. $E_{m'}^{(s)}$ is the event $w^{(s)}(m') = w^*$. For a fixed given state in cycle $m'-1$ of the Markov process, the probability of this event is the value $Y_{m'}$ as defined by (17). $Y_{m'}$ itself is a random variable with a certain distribution (cf. the remark after the proof of Proposition 4.1). So, in order to get the (conditional) probability of $E_{m'}^{(s)}$ without fixing the previous state, we have still to take the expected value with respect to the distribution of $Y_{m'}$. In particular,

$$Pr\,\{\,Y_{m'} \geq 1-\tilde{\epsilon}\,|\,F_m\,\} \;\geq\; 1-\tilde{\epsilon}$$

for $m' \geq m + d''(\tilde{\epsilon}, m)$ by the Corollary to Lemma 4.3. Hence for such an m',

$$Pr\,(\,E_{m'}^{(s)}\,|\,F_m) \;\geq\; Pr\,\{\,E_{m'}^{(s)} \wedge (Y_{m'} \geq 1-\tilde{\epsilon})\,|\,F_m\,\}$$

$$= Pr\,\{\,E_{m'}^{(s)}\,|\,(Y_{m'} \geq 1-\tilde{\epsilon}) \wedge F_m\,\}\;\cdot\;Pr\,\{\,Y_{m'} \geq 1-\tilde{\epsilon}\,|\,F_m\,\} \;\geq\; (1-\tilde{\epsilon})\;\cdot\;(1-\tilde{\epsilon}) \;\geq\; 1-2\tilde{\epsilon}.$$

The assertion is obtained by setting $\tilde{\epsilon} = \epsilon/2$. □

Now we are able to prove the main result:

Proof of Theorem 4.1. We have

$$P_m = Pr(E_m^{(1)}) = \ldots = Pr(E_m^{(S)}).$$

Furthermore,

$$Pr(B_1 \wedge \ldots \wedge B_m) = Pr(B_1) \cdot Pr(B_2|B_1) \cdot \ldots \cdot Pr(B_m|B_1 \wedge \ldots B_{m-1}),$$

so by the Corollary to Lemma 4.1,

$$Pr(B_1 \wedge \ldots \wedge B_m) \leq (1-p)^S(1-cp)^S \ldots (1-c^{m-1}p)^S = \left[\prod_{i=1}^{m}(1-c^{i-1}p)\right]^S.$$

Set

$$w(p,c,S) = \left[\prod_{i=1}^{\infty}(1-c^{i-1}p)\right]^S.$$

Because the event A can be represented as

$$A = \neg(B_1 \wedge B_2 \wedge \ldots),$$

one obtains

$$Pr(A) = 1 - \lim_{m\to\infty} Pr(B_1 \wedge \ldots \wedge B_m) \geq 1 - \lim_{m\to\infty}\left[\prod_{i=1}^{m}(1-c^{i-1}p)\right]^S = 1 - w(p,c,S).$$

On the other hand, $w(p,c,S)$ can be made arbitrarily small[1] either by choosing S sufficiently large, or by choosing ρ sufficiently small: because of $\log x \leq x-1$ for $0<x<1$,

$$\log w(p,c,S) = S\sum_{i=1}^{\infty}\log(1-c^{i-1}p) \leq -S\sum_{i=1}^{\infty}c^{i-1}p = -Sp\sum_{i=0}^{\infty}c^i = -\frac{Sp}{1-c},$$

that is,

$$w(p,c,S) \leq \exp\left(-\frac{Sp}{1-c}\right).$$

Since $0<c<1$, the expression on the r.h.s. tends to zero both (for fixed c) as $S\to\infty$, and (for fixed S) as $c\to 1$ (that is, as $\rho\to 0$).

Summarizing, we obtain the fact that it is possible by an appropriate choice of S or of ρ to achieve that $w(p,c,S)\leq \epsilon/4$ and hence $Pr(A)\geq 1-\epsilon/4$.

Because of

$$1 > Pr(A) = Pr(F_1 \vee F_2 \vee \ldots) = \sum_{m=1}^{\infty} Pr(F_m),$$

the partial sums $\sum_{m=1}^{K} Pr(F_m)$ of the series above converge as $K\to\infty$, so there is an integer $K=K(\epsilon)$ such that

$$\sum_{m=K+1}^{\infty} Pr(F_m) < \frac{\epsilon}{4}.$$

Then,

$$Pr(F_1\vee\ldots\vee F_K) = \sum_{m=1}^{K} Pr(F_m) \geq Pr(A) - \frac{\epsilon}{4} \geq 1-\frac{\epsilon}{2}.$$

By Lemma 4.4,

$$Pr(E_{m'}^{(1)}|F_m) \geq 1-\frac{\epsilon}{2} \tag{18}$$

for all $m' \geq m + d'''(\epsilon/2, m)$. Let

$$d(\epsilon) = \max\left(d'''\left(\frac{\epsilon}{2},1\right),\ldots,d'''\left(\frac{\epsilon}{2},K\right)\right),$$

and $m_0 = m_0(\epsilon) = K + d(\epsilon)$. Then for $m\leq K$, eq. (18) holds for all $m'\geq m_0$. So for all $m'\geq m_0$:

$$\begin{aligned} P_{m'} = Pr(E_{m'}^{(1)}) &= Pr(E_{m'}^{(1)}|F_1)\cdot Pr(F_1) + \ldots + Pr(E_{m'}^{(1)}|F_K)\cdot Pr(F_K) \\ &\quad + Pr(E_{m'}^{(1)}|\neg(F_1\vee\ldots\vee F_K))\cdot Pr(\neg(F_1\vee\ldots\vee F_K)) \end{aligned}$$

[1] The reader might conjecture that since $0<1-c^{i-1}p<1$ for all i, one has $w(p,c,S)=0$ anyway. Unfortunately, this is not the case, as the following counterexample shows: Let $c=1/2$, $p=1/2$, and $S=1$. Then $w(p,c,S)=\prod_{i=1}^{\infty}\left(1-(1/2)^i\right) = (1/2)\cdot(3/4)\cdot(7/8)\ldots$. Since

$$\log w(p,c,S) = \log\left(1-\frac{1}{2}\right) + \log\left(1-\frac{1}{4}\right) + \ldots \geq -2\cdot\frac{1}{2} - 2\cdot\frac{1}{4} - \ldots = -2,$$

we have $w(p,c,S)\geq e^{-2} = 0.1353\ldots > 0$.

$$\geq Pr(E_{m'}^{(1)}|F_1) \cdot Pr(F_1) + \ldots + Pr(E_{m'}^{(1)}|F_K) \cdot Pr(F_K)$$
$$\geq \left(1 - \frac{\epsilon}{2}\right) \left(Pr(F_1) + \ldots + Pr(F_K)\right) \geq \left(1 - \frac{\epsilon}{2}\right) \left(1 - \frac{\epsilon}{2}\right) \geq 1 - 2 \cdot \frac{\epsilon}{2} = 1 - \epsilon.$$

□

Remark 4.3. We do not claim that the theoretical bounds implicit in the proof of our result are helpful for deriving reasonable choices of the parameters "number of agents" and "evaporation factor" in practical applications. *Assuring* convergence to the optimum with high probability via these bounds would exceed the power of present computers. Our bounds are far too weak to admit such numerical conclusions. E.g., the attempt to achieve the requirement $Pr(A) \geq 1-\epsilon$ by the choice of a sufficiently large number S of agents, using the inequalities

$$Pr(A) \;\geq\; 1 - w(p, c, S) \;\geq\; 1 - \exp\left(-\frac{Sp}{1-c}\right)$$

would imply

$$S \;\geq\; \frac{(1-c)\,(-\log \epsilon)}{p},$$

which is usually a huge value, since p, as defined in Lemma 4.1, is typically extremely small (e.g., $p = 0.05^{10}$ for a construction graph with 20 arcs, an optimal walk of length 10, and $\gamma = 1$). Similarly, if S is fixed, the small value of p entails an unrealistically small evaporation factor ρ.

The reader should notice, however, that these extraordinary small values for p result from a *very* coarse estimation of the probability that an agent finds the optimal path. In order to be quite general, we have based our computation on the worst case that the use of the desirability values η_{kl} is not helpful at all, but can instead be even misleading. Nevertheless, in each *reasonable* application, these heuristic values will be of great value, such that the chance that at least one agent finds the optimal path in some cycle is drastically improved. Moreover, we have used several other rough bounds implying an underestimation of the real convergence probability by some orders of magnitude. Considering this, convergence to the optimum with high probability may already be obtained with a moderate number of agents.

Perhaps the bounds can be improved in the future by tighter estimations and suitable models for the influence of the desirability values. In any case, let us emphasize that it is not the intention of the present article to give hints for the choice of parameter values in applications, but rather to demonstrate the capability of the ant–based approach. Especially in view of future developments on the hardware sector, it seems important to distinguish between "optimization–capable" heuristics, where there is no *principal* obstacle to get closer and closer to the optimal solution, provided that more computer power is invested, and "non–optimizing" heuristics for which there are problem instances where the produced solution quality may always be at a certain distance from the optimum, no matter how much computation effort be spent and how powerful the hardware may be. For anybody using or developing ant–based algorithms, it might be reassuring to know that Graph–based Ant System belongs to the first class of heuristics. A thorough theoretical comparison of Graph–based Ant System with other heuristics of this first class is a task for the future.

5 Conclusion

We have developed a formal framework for extending *Ant System* ([8, 4]) to *Graph–based Ant System*, a metaheuristic which can treat arbitrary static combinatorial optimization problems. Graph–based Ant System is closely related to the ACO metaheuristic recently developed by Dorigo and Di Caro, but it focuses more on classical features of ant–based optimization systems, and is somewhat more restricted with respect to implementation alternatives.

For Graph–based Ant System, we have been able to derive a convergence result. Compared to the convergence result for another well–known optimization metaheuristic, Simulated Annealing (see [1], ch. 3), our Theorem 4.1 is weaker: we cannot demonstrate convergence (to the optimal solution) in distribution or with probability one, but only convergence with a probability that can be made arbitrarily close to one by a suitable choice of the parameters of the heuristic. In particular, it has been shown that there are two possible alternatives for increasing the convergence probability: increasing the number of agents, or decreasing the evaporation factor.

As most theoretical convergence results for heuristics, our Theorem 4.1 has to be interpreted cautiously. From the fact that a large number of agents and/or a small evaporation factor yield high probabilities of convergence to the optimum, it does not follow that an implementation of Graph–based Ant System is the better, the more agents are used and the smaller the evaporation factor is. The price for a (theoretically) favorable convergence behavior can be an explosion of the computation time: If, for example, a very large number of agents is simulated on a single–processor system, then it may take an infeasible large computation time before Graph–based Ant System turns from a merely "explorative" phase to an "optimizing" phase. So it may be better to reduce the number of agents, even if thereby also the chance of finding the global optimum is reduced. A very similar problem is known from SA applications, where the "theoretical" cooling schedule (implying convergence) must be modified to a much faster cooling schedule in order to get the procedure out of the first (random–search–like) phase with high temperature parameter (see [1]).

As to ant–based optimization in general, much remains to be done, both experimentally and theoretically. Up to now, only a small fraction of problem types seems to have been attacked by such approaches. In terms of the framework developed in this article, research should be done in designing suitable construction graphs for the diverse types of applications, and in evaluating them experimentally. For most problems, there are alternative ways of encoding feasible solutions by walks in corresponding construction graphs, and it should be found out which are the best. Also different ways of updating pheromone values (cf. the "elitist" scheme suggested by condition (d) at the beginning of Section 4) should be investigated in much more detail.

From a theoretical point of view, our results in Section 4 leave some interesting questions open. Which of our conditions (a) – (d) can be relaxed or dropped? Particularly challenging seems the question whether condition (b) can be dropped. Besides its theoretical interest, an answer to this question is also of practical value: if the answer whether condition (b) can be omitted should be "no", then one could conclude that redundancy in encoding feasible solutions should be avoided whenever possible, which would exclude certain types of construction graphs where several walks correspond to the same solution. As mentioned in Section 4, however, it seems at the moment that condition (b) can be completely dropped. Condition (d) should stimulate the investigation of the convergence behavior of mixtures between elitist and non–elitist pheromone update strategies. Moreover, also rank–based update strategies (see [2]) should be investigated.

Another interesting theoretical question concerns the speed of convergence. From the proof of our main theorem, some information about this topic could be derived, but the question deserves a much closer look. In total, it would be desirable to develop the theory of ant–based systems to such an extent as it has already been achieved for certain other metaheuristics, Simulated Annealing being an excellent example.

Acknowledgment. The author is indebted to B. Bullnheimer and C. Strauss for having drawn his attention to Ant System, to B. Bullnheimer for helpful remarks on a preliminary version, and to R. F. Hartl for a discussion on the conditions used in the convergence result. Furthermore, he wants to thank the anonymous referees for their profound comments which led to a considerable improvement of the article.

References

[1] E. Aarts and J. Korst, Simulated Annealing and Boltzmann Machines (Wiley, 1989).

[2] B. Bullnheimer, R.F. Hartl, and C. Strauss, A new rank–based version of the Ant System: A computational study, Central European Journal for Operations Research and Economics 7 (1999) 25–38.

[3] B. Bullnheimer, G. Kotsis, and C. Strauss, Parallelization Strategies for the Ant System, in: R. Leone, A. Murti, P. M. Pardalos, G. Toralo, eds., High Performance Algorithms and Software in Nonlinear Optimization (Kluwer Academic Publishers, Dordrecht, 1998) 87–100.

[4] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD Thesis, Dept. of Electronics, Politecnico di Milano, Italy, 1992.

[5] M. Dorigo, G. Di Caro, and L.M. Gambardella, Ant algorithms for distributed discrete optimization, Artificial Life 5(2) (1999) 137–172.

[6] M. Dorigo and G. Di Caro, The ant colony optimization meta-heuristic, in: D. Corne, M. Dorigo, and F. Glover, eds., New Ideas in Optimization (McGraw-Hill, London, UK, 1999) 11–32.

[7] M. Dorigo and L.M. Gambardella, Ant Colony System: A cooperative learning approach to the traveling salesman problem, IEEE Trans. on Evolutionary Computation 1 (1997) 53–66.

[8] M. Dorigo, V. Maniezzo, and A. Colorni, The Ant System: An Autocatalytic Optimization Process, Technical Report 91–016, Dept. of Electronics, Politecnico di Milano, Italy, 1991.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, The Ant System: Optimization by a colony of cooperating agents, IEEE Trans. on Systems, Man, and Cybernetics 26 (1996) 29–41.

[10] Th.A. Feo and M.G.C. Resende, Greedy randomized adaptive search procedures, J. of Global Optimization 6 (1995) 109–133.

[11] R.G. Gallager, Discrete Stochastic Processes (Kluwer, 1996).

[12] F. Glover, Tabu Search, Part I, ORSA J. on Computing 1 (1989) 190–206.

[13] L.M. Gambardella and M. Dorigo, Ant-Q: A Reinforcement Learning approach to the traveling salesman problem, in: Proc. ML-95, Twelfth Intern. Conf. on Machine Learning (Morgan Kaufman, Palo Alto, CA, 1995) 252–260.

[14] L.M. Gambardella and M. Dorigo, Solving symmetric and asymmetric TSPs by ant colonies, in: ICEC96, Proc. of the IEEE Conference on Evolutionary Computing (IEEE Press, 1996) 622–627.

[15] W.J. Gutjahr, A generalized convergence result for the Graph–Based Ant System metaheuristic, Technical Report 99-09, Dept. of Statistics, O.R. and Computer Sci., University of Vienna, 1999.

[16] J.H. Holland, Adaption in Natural and Artificial Systems (University of Michigan Press, 1975).

[17] V. Maniezzo, A. Colorni, and M. Dorigo, The Ant System applied to the quadratic assignment problem, Technical Report IRIDIA/94–28, Université Libre de Bruxelles, Belgium, 1994.

[18] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs (Springer, 1992).

[19] Ch.H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity (Prentice–Hall, 1982).

Correspondence:
Walter J. Gutjahr
Dept. of Statistics, O.R., and Computer Sci.
Universitaetsstrasse 5/9
A-1010 Wien, Austria
E-mail: walter.gutjahr@univie.ac.at