Convergence Control in ACO

Bernd Meyer
bernd.meyer@acm.org

Dept. of Computer Science, Monash University, Australia

Abstract. Ant Colony Optimization (ACO) is a recent stochastic meta-heuristic inspired by the foraging behaviour of real ants. As for all meta-heuristics the balance between learning based on previous solutions (intensification) and exploration of the search space (diversification) is of crucial importance. The present paper explores a novel approach to diversity control in ACO. The common idea of most diversity control mechanisms is to avoid or slow down full convergence. We suggest to instead use a fast converging search algorithm that is artificially confined to the critical phase of its convergence dynamics. We also analyze the influence of an ACO parameter that does not seem to have received sufficient attention in the ACO literature: α, the exponent on the pheromone level in the probabilistic choice function. Our studies suggest that α does not only qualitatively determine diversity and convergence behaviour, but also that a variable α can be used to render ant algorithms more robust. Based on these ideas we construct an Algorithm *cc*AS for which we present some encouraging results on standard benchmarks.

1 Introduction

Ant Colony Optimization (ACO) [DDC99] is a relatively recent model-based meta-heuristic [ZBMD04]. It is successfully applied to a growing number of industrial and academic constraint optimization problems and reaches state-of-the-art results for several important problem classes [CHS02]. The progenitor of all ACO algorithms, Ant System (AS) [Dor92], was very directly inspired by the pheromone-based trail-laying-trail-following behaviour of real ants [CD01].

In ACO a number of agents ("ants") independently construct solutions in parallel by iteratively augmenting partial solutions. Consider a TSP, where a partial solution corresponds to a partial path. Every construction step extends a partial path to a new city. In ACO the ants make the choice of the next city based on a so-called "pheromone value", which models the preference for a particular choice and is cooperatively learned by the ants during the search. In a TSP the pheromone value is commonly attached to a pair of cities c_i, c_j and models the preference to go from city c_i to c_j. The pheromone values are learned through a reinforcement strategy in which each agent, after the complete solutions have been constructed, reinforces the choices it has made during the solution construction with an amount of reinforcement that depends on the solution quality obtained. The AS algorithm for TSP is given in Figure 1.

Modern ACO algorithms deviate significantly from the simple AS schema for the sake of a more efficient search. One of the core aspects of this quest is to achieve a better balance between exploration and exploitation or learning, i.e. between maintaining solution diversity and intensifying the search in the vicinity of good solutions in the search space. Achieving this balance requires the algorithm designer to adjust the convergence dynamics, where converging means that an increasing number of ants starts to reproduce the same solution.

```
$\forall i,j : \tau_{i,j} = \tau_0$                    /* initialize pheromone matrix
$\forall i,j : \eta_{i,j} = d_{i,j}^{-1}$                    /* heuristic $\eta$=inverse distance
$l_{gb} := +\infty$; $T^{gb} := nil$;                    /* initialize global best
for $t := 1$ to max_iterations do
    for $k := 1$ to number_of_ants do
        $T^k = nil$                    /* intialize tour of ant $k$ as empty
        mark all cities as unvisited by ant $k$
        $i :=$ random city not yet visited by ant $k$
        for $n := 2$ to number_of_cities
            mark city $i$ as visited by ant $k$
            $C :=$ set of cities not yet visited by ant $k$
            choose next city $j$ to be visited by ant $k$ with probability
                $p_j := \frac{\tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{j \in C} \tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}}$
            $T^k := append(T^k, (i,j))$
            $i := j$
        end
        $l_k := length(T^k)$
    end
    $ib := argmin_k(l_k)$                    /* best tour index
    if $l_{ib} < l_{gb}$ then begin $T^{gb} := T^{ib}$; $l_{gb} := l_{ib}$ end
    $\forall i,j : \tau_{i,j} := (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}$                    /* evaporate and reinforce
    where $\Delta\tau_{i,j} = \sum_{k=1}^{number\_of\_ants} \Delta\tau_{i,j}^k$ and $\Delta\tau_{i,j}^k = \begin{cases} Q \cdot l_k^{-1} & if\ (i,j) \in T^k \\ 0 & otherwise \end{cases}$
end.
```

Fig. 1. The Original Ant System Algorithm for TSP

In the case of static optimization problems early convergence to a small region of the search space leaves large sections of the search space unexplored. On the other hand, very slow convergence cannot sufficiently concentrate the search in the vicinity of good solutions and therefore renders the search inefficient. In the case of dynamic problems and reactive optimization diversity control takes an even more important role, as continued solution diversity is required to be able to quickly adapt to problem changes.

The common idea behind all published diversity control modifications is essentially to avoid full convergence or at least to slow it down significantly. These control mechanisms can be classified as either modifications of the pheromone deposit function or modifications of the probabilistic decision function. The earliest such refinement was introduced with Ant Colony System (ACS) [DG96]. The basic idea of ACS is to use pheromone update only on the elitist path and to complement the normal evaporation $\forall i,j : \tau_{i,j} := (1-\rho)\tau_{i,j}$ applied to the elitist path, which takes place after all ants have completed a tour construction, by local evaporation: Each ant takes a small amount of pheromone off each edge used as soon as it traverses this edge. In consequence the edge becomes less desirable for subsequent ants and thus solution diversity is increased. MaxMin AS [SH00] achieves increased diversity by imposing artificial minimum and maximum limits on the amount of pheromone on each edge, so that the likelihood of being selected does not ever become vanishingly small or overwhelmingly large for any path. In the refined form of MaxMin AS with pheromone trail smoothing (PTS), pheromone that exceeds a certain threshold is redistributed among incident edges. All these modifications have lead to significantly more effective search procedures [DG96,SH00]. An alternative to modifying the pheromone deposit mechanism is to change the path selection mechanism. This has been investigated mainly by introducing additional randomization, resembling a mutation operation in a GA, into the decision function [NA01].

A final possibility is to tune parameters of the decision function. We will analyze the influence of α, the exponent on the pheromone level in the selection function, which has received surprisingly little attention in the literature. Our studies suggest that α does not only qualitatively determine diversity and convergence behaviour, but also that a variable α can be used to improve the efficiency of the search.

Following this, we explore a radically different approach to diversity control in ACO. Instead of avoiding convergence we suggest to use a fast converging search algorithm that is artificially confined to the critical phase of its convergence dynamics. Based on these ideas we construct an Algorithm *cc*AS and analyze its performance for standard benchmarks. The results appear encouraging.

2 The Influence of α on ACO Convergence Dynamics

While ACO research has a track record of successful applications, only few papers have analyzed ACO theoretically. Even though convergence proofs exist for some restricted forms of ACO [SD02,Gut02], the general understanding of convergence behaviour and system dynamics is still in its infancy. This is particularly true for the function of α, the exponent on the pheromone level in the choice function of the original Ant System algorithm (Figure 1). In most approaches it is taken to be 1, so that the selection probability is linear in the pheromone level. The fact that α has not received a more systematic investigation is somewhat surprising, as its value has a marked influence on the convergence behaviour.

Experimental support for the idea that α has important influence in ACO can be drawn from some recent studies in ACO. In [BB98] the authors attempted to find optimum parameters for ACS applied to standard TSP benchmarks. A genetic algorithm was used to "breed" the optimum parameters for the ACS algorithm. Training was performed on one problem instance (Oliver30) and testing was performed with a different problem instance of comparable size (Eil51). The optimum parameter set found was able to outperform previous search methods on Eil51. Interestingly, the optimum α value was not 1 but 0.39.

In another recent study [RT02] different forms of varying α during the search process are experimentally evaluated. The authors find that some forms of non-constant α appear to outperform the same algorithm with a constant α. Both studies mentioned were purely experimental, as were, to the best of the author's knowledge, all other studies investigating α. The obvious question is, whether the influence of α can be understood in a more systematic manner. While a full theoretical analysis currently still seems beyond reach, as the system dynamics of ACO is not even fully understood for fixed α, it is possible to gain more insight by investigation of ACO convergence dynamics on some very simple graphs.

To achieve an analytical understanding of ACO system dynamics, we start from its simplest form, AS without a heuristic function ($\beta = 0$), and calculate expected values instead of the discrete update performed by a finite number of ants. This is a common simplification and a comparison with numerical experimental data on real AS will later be used to validate the investigation.

We can gain some initial insight by looking at the AS behaviour on the simplest possible graph (Figure 2) in which we can search for a Hamiltonian path. Ants must obviously start at the west-most node and can either proceed first north-east and then south (pathlength l_1) or start south-east and then go north (pathlength l_2), $l_1 < l_2$. Let p_i^* be the steady state of the probability p_i for the path with length l_i to be selected. Let τ_1 be the pheromone level on the edge leading north-east and τ_2 the one on the edge leading south-east. At steady state we must have: $\rho \cdot \tau_i = \frac{Q}{l_i} \cdot \ n \cdot p_i^*$ By substituting the definition of p_i from Figure 1 and dividing both p_i we find $p_1^* = \left(1 + \left(\frac{l_1}{l_2}\right)^{\frac{\alpha}{1-\alpha}}\right)^{-1}$

Interestingly, p_1^* depends on the ratio of path lengths and on α, with a singularity at $\alpha = 1$. A full fix point analysis for p_1^* is given in Figure 3 (stable fixpoints marked with solid lines, unstable ones with dashed lines). The two trivial fixpoints correspond to "all traffic on the better path" and "all traffic on the inferior path", i.e. $p_1^* = 1$ and $p_1^* = 0$. The non-trivial fixpoint corresponds to a proportional distribution of the traffic depending on the ratio of path lengths $\frac{l_1}{l_2}$. For $0 < \alpha < 1$ only the non-trivial fixpoint is stable, i.e. the system is guaranteed to reach a proportional traffic distribution. At $\alpha = 1$ the only stable fixpoint is $p_1^* = 1$, i.e. the system is guaranteed to end up with the entire traffic on the better path. For $\alpha > 1$ the system behaviour is definitely not optimizing. The third (proportional) fixpoint corresponds to inverted proportions and becomes unstable and both (!) trivial fixpoints become stable. This means that now $p_1^* = 0$, i.e. all traffic on the inferior path, is stable. It depends on the initialization whether the system reaches the correct fixpoint or the inverted one.

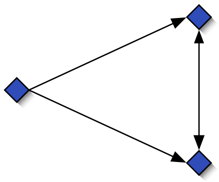

Fig. 2. Simple Model

This shows that $\alpha > 1$ is not a good choice, as the system can converge totally to the wrong path. More importantly the analysis also suggests that if we want to keep diversity in the search $\alpha = 1$ may not be the ideal choice as it forces total convergence towards a single path. Does this observation extend to realistically sized problems tackled with a finite number of ants?

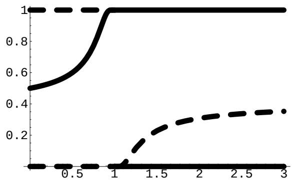

Fig. 3. Fixpoints

The theoretical considerations can easily be verified in a numerical simulation. We use randomly generated 20 city problems with an edge probability of $p_e = 0.5$. The histograms in Figures 4 show a typical run result for AS with 20 ants using simple visibility heuristics $\eta_{ij} = d_{ij}^{-1}$ where d_{ij} is the distance between cities i and j. Each bar represents an interval of solution quality (with shorter pathlengths on the left). The height of the dark bars in the foreground represents the relative frequency of solutions in this interval after 1000 iterations. The gray bars in the background represent the relative path length frequencies of all unique paths encountered during the entire search. $\alpha = 2$ converges to a single path, but not to the best path encountered during the search. With $\alpha = 1$ the situation looks somewhat better: the system converges towards a single path in the range of the good solutions. However, as the small number of ants cannot

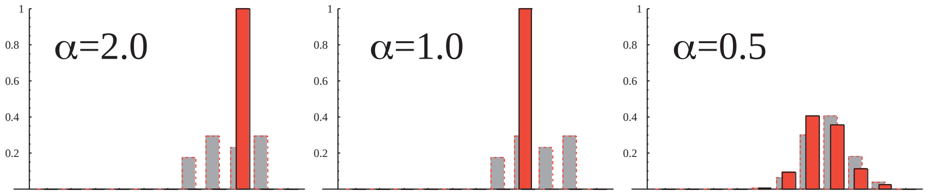

Fig. 4. Solution Histograms for 20 City Random Problem

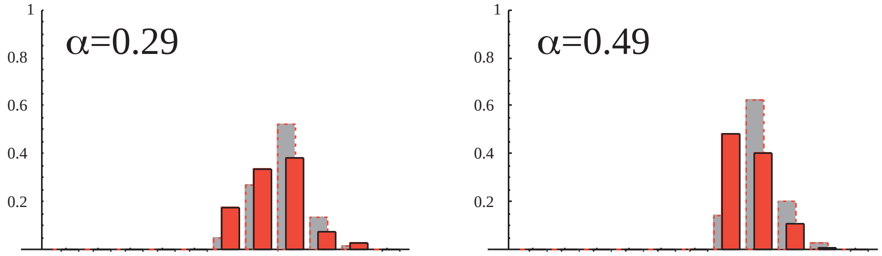

Fig. 5. α Determines Distribution Center and Shape

sample the entire search space and the system converges too quickly, it does not converge reliably to the best path, rather just to some path in the range of good solutions. Which path wins is dependent on the initialization of the pheromone levels and on random events early in the search. In contrast, with $\alpha = 0.5$ the system does not converge to a single path. Instead it maintains a distribution of paths that during the search shifts into better regions. Given sufficient time, this enables the system to reliably find the optimal path.

It is also instructive to look at the shape of the distributions generated by different α. From Formula 2 and $p_2^* = 1 - p_1^*$ we obtain $\frac{p_1^*}{p_2^*} = \left(\frac{l_2}{l_1}\right)^{\frac{\alpha}{1-\alpha}}$. For $\alpha = 0.5$ the probability ratio mirrors the path length ratio exactly, while $\alpha > 0.5$ over-emphasizes path length differences and $\alpha < 0.5$ under-emphasizes these.

In a graph of realistic size the distribution shape should also be expected to depend on α. Again, this can be confirmed with a numerical experiment. Figure 5 shows typical final path length distributions for the random 20 city model for $\alpha = 0.29$ and $\alpha = 0.49$ after 400 iterations. Clearly, the distribution for the higher α is distorted more strongly towards better path lengths.

The insight we have gained so far suggest to keep α in the range $0 < \alpha < 1$. A value closer to 1.0 will emphasize better paths but reduce diversity, while lower α will keep more diversity but reduce selective pressure. However, it appears impossible to fix a universally best α.

3 Alpha Annealing

The previous section suggests that an adaptive or dynamic α that changes throughout the search could be advantageous. We will term this procedure, which loosely resembles the idea of simulated annealing,"α-annealing". The idea is to change the α value according to some annealing schedule (to be defined later) between steps 4 and 5 of the AS algorithm (Figure 1). Increasing α slowly throughout the search can keep diversity in the beginning and gradually increase the selective pressure to cover better regions of the search space in the later phases. The first question to ask is, is it possible to establish tighter bounds for a reasonable α-range than the interval $(0,1]$. We shall first look at the lower bound. It turns out that cycles have interesting implications for this question.

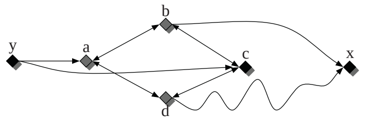

Fig. 6. Circuit Trinket

Figure 6 shows the simplest possible graph that contains a bidirectional cycle, a root node and at least two paths with different costs (lengths). All edges are assumed to have unit cost with the exception of the wavy edge, which has a higher cost. To perform a complete tour, an ant obviously has to start at y and has to terminate at x. From y it can either perform the tours $ycdabx$ with $cost(ycdabx) = 5$, $yabcdx$ with $cost(yabcdx) > 5$ or the tours $yadcbx$ with $cost(yadcbx) = 5$ and $ycbadx$ with $cost(ycbadx) > 5$. Obviously, ants starting with ya should prefer $yadcbx$ and those starting with yc should prefer $ycdabx$ and reward the corresponding edges more strongly. Ants starting with ya will at node c reinforce the decision to go to node b, while ants starting with yc will reinforce the contrary decision to go to node d from c. If ants are equally likely to start with ya or yc, it can be expected that the reinforcements annihilate each other and no preferred orientation is learned. As the rewards for both orientations are identical, this should be expected when the general system behaviour converges towards a proportional distribution according to path cost. On the other hand, if the system behaviour dictates convergence towards a single solution, we should expect one of the orientations to ultimately attract all of the traffic, i.e. we expect *symmetry breaking* through reinforcement of some initial random fluctuation towards one of the orientations.

Figure 7 shows the development of the path choice probabilities for AS with expected value update on the graph in Figure 6. With $\alpha = 1$ the entire traffic converges on only one of the two optimal paths (symmetry breaking). With $\alpha = 0.49$ both circuit orientations converge towards a probability of 0.5 and the traffic distribution on the two exit edges bx and dx is proportional. As the total path probability is the product of the edge probabilities, the resulting path probability for the optimal paths becomes very low. More extensive simulations show that the symmetry breaking only sets in for $\alpha > 0.5$.

The system works much more efficiently when symmetry breaking is performed as the total number n of ants contributing to the system is higher. With $\alpha = 1$ all n ants choose the same single optimal path (here $ycdabx$). With $\alpha = 0.49$ (no symmetry breaking) less than $n/2$ ants are on the two possible op-

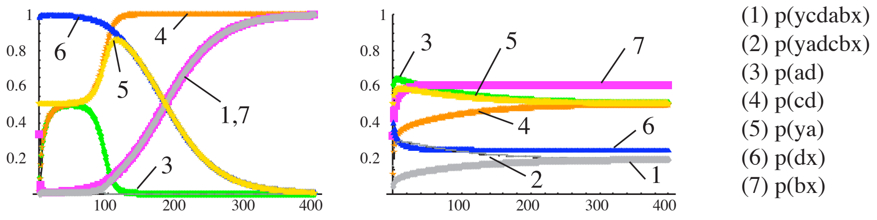

Fig. 7. Path Choice Probabilities for the Circuit Trinket (left: $\alpha = 1.0$, right: $\alpha = 0.49$)

timal paths. Even for a proportional traffic distribution we should have expected more than $n/2$ as two out of the total four paths have optimal length. The loss is due to missing symmetry breaking: contradictory reinforcements annihilate one another and conspire to let some proportion of ants get stuck. We can therefore assert that symmetry breaking is in general desirable. As bi-directional circuits can always be a component of the problem graph, we can establish $\alpha > 0.5$ as a safe (but not necessarily tight) lower bound.

To establish an upper bound for α we ask whether there is a maximum α below which full convergence to a single path (SPC in the following) is always avoided. We construct the simplest possible DAG in which two paths intersect at a node and cause contradictory reinforcements (Figure 8).

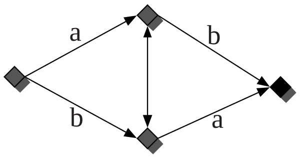

Fig. 8. Bridge Trinket

To complete a tour in this graph, an ant has to start at the leftmost node and terminate at the rightmost node. It can either follow the a-path ($a - down - a$), which receives reward μ_a, or the b-path, which receives reward μ_b. Assuming uniform pheromone initialization all edges on the a-path (b-path) receive the same update $\Delta\tau_a$ ($\Delta\tau_b$, respectively). Because of symmetry the sum of all pheromones on outgoing edges at the two middle nodes is also identical. From $\rho \cdot \tau_i^* = \Delta\tau_i$ at the fixpoint τ_i^* ($i \in \{a, b\}$) we immediately obtain for the non-trivial fixpoint $\frac{\tau_b^*}{\tau_a^*} = \left(\frac{\mu_a}{\mu_b}\right)^{\frac{1}{2\alpha-1}}$. At the singularity $2\alpha = 1$ the non-trivial fixpoint becomes unstable and only the trivial fixpoints ($p_a = 1$ or $p_b = 1$) become stable. In other words, as α passes 0.5 the system switches to SPC.

Combining these two observations we see that it is not possible to establish an α-range independently of the network: symmetry breaking dictates $\alpha > 0.5$, while avoiding SPC dictates $\alpha < 0.5$. Thus we cannot expect to avoid SPC.

It is important to note that even though our argument relies on very specific simple small graphs, any larger graph can always contain these as subgraphs, so that the argument directly extends to arbitrary larger graphs.

However, α-annealing can still make sense as it will emphasize diversity in the beginning of the search, thereby giving the algorithm a chance to widely sample the search space, and will subsequently gradually increase the selective pressure in favour of better solutions until convergence to a single path is reached. This potentially balances learning and exploration better than just a constant α setting that would either lead to rapid SPC (possibly leaving large amounts of interesting parts of the search space unexplored) or would favour exploration but distort the probability distribution unreasonably towards inferior solutions. The importance of this becomes particularly obvious when we take heuristics into account. The task of the heuristic factor $\eta_{i,j}$ is to bias the (initial) probability distribution in the search space towards better solutions. If an ACO method that performs rapid SPC is using a heuristics that biases the search towards sub-optimal regions of the search space, it is likely that it will not converge towards the optimal solution. α-annealing has the advantage that it can render the algorithm more robust against such sub-optimal guidance received from the heuristics, as it favours diversity in the initial search phases.

We demonstrate this effect with an α-annealing Ant System of 20 ants applied to the same 20 city random problems as above. Recall that a standard Ant

System with constant $\alpha = 1.0$ did generally converge on paths that are close to optimum but sub-optimal. Those problem sets which AS with $\alpha = 1$ fails to solve to optimality in the majority of trials were selected for processing with α-annealing AS. The annealing schedule starts from $\alpha = 0.3$ and increases α every 50 iterations by 0.1 to a maximum of $\alpha = 1.0$. α-annealing AS solves these problem instances reliably to optimality within 400 iterations.

While this first test does not by any means prove the superiority of α-annealing, it is encouraging. However, even though α-annealing appears to achieve an improved search behaviour and more robustness against the choice of a heuristics, we are essentially still faced with a variant of the original problem. As α-annealing AS is in the limit a standard AS, it is fully convergent in the later phases of the search. To sustain diversity throughout the search, even in its later phases, we need to investigate other modifications.

4 Exploiting the Critical Phase of Convergence

We revisit the simple model introduced in Figure 2 to inspect its convergence dynamics. Figure 9 shows the development of path choice probabilities. It is striking that there is little change for a long time, until the fixpoint is assumed in a very rapid transition (around 600 iterations). The qualitative dynamics is not unique for the trivial model graph. The left-hand graph in Figure 10 shows the typical development of solution diversity during a 200 iteration run of a standard 20 ant AS on the 51 city benchmark Eil51. Fine diversity is measured as the number of different paths followed by the 20 ants in a single iteration. It is clearly visible that the same type of dynamics governs the system: after a long period of full maximal diversity the system abruptly converges to a single path in what looks like a phase transition.

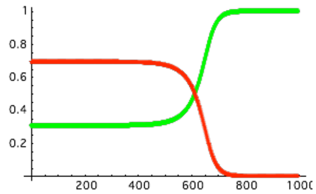

Fig. 9. Path Probabilities

The best solutions are typically found at the end of the transition phase, shortly before full convergence to the single solution path. The question arises in how far this single path is already determined by the pheromone matrix before the phase transition. If not, i.e. if small random fluctuations can still drive the system to converge onto different solutions, it could be the case that *just before the transition all the diversity that we were aiming to keep is still present and represented in the pheromone matrix*. This suggests that re-running the transition phase could be an effective way to explore this diversity.

To enable us to locate the phase transition we use two different measures of diversity: *fine* diversity and *coarse* diversity. As above, fine diversity is measured as the number of different complete paths generated in a single iteration, i.e. during n parallel tour constructions without update of the pheromone matrix. *Coarse* diversity counts the number of different complete paths that have been generated in a fixed number of k successive iterations (here 10). The maximum fine diversity therefore is n and the maximum coarse diversity $k \cdot n$. The typical development of these two measures is shown in Figure 10. After a long initial period of maximal diversity the coarse diversity starts to fall off and then both measures rapidly fall to minimum. The coarse diversity starting to decrease before the fine diversity is due to the fact that the system still produces very

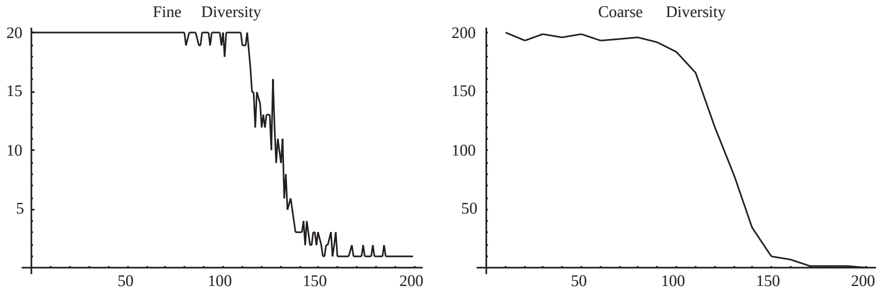

Fig. 10. Diversity Development for Eil51

close to n solutions per iteration, but that some solutions are repeated between iterations. We can use this observation as an "early warning" mechanism that the transition is about to happen. To repeatedly cycle through the transition phase we save the pheromone matrix after each iteration as long as the coarse diversity is still high enough and we reset the pheromone matrix to this saved value as soon as the fine diversity has fallen to minimum. This is performed between steps 20 and 21 of Figure 1. As the transition is generally very sharp it is easy to choose appropriate values for the thresholds very close to minimum and maximum. We use 97% coarse diversity as the threshold for determining the onset of the transition and 10% fine diversity as the threshold to determine full convergence.

The proposed Ant System algorithm, which we term *critical cycle Ant System* (*cc*AS) is essentially akin to a multistart procedure, with the crucial difference that the subsequent runs do not start from scratch but from a pheromone matrix that has already developed into a good representation of the distribution of solutions in the vicinity of the optimum. Therefore a substantial amount of computational effort is saved compared with a simple multi-start procedure. Does this algorithm really produce the required diversity? The best indication of this is given by the cumulative count of unique new solutions in the vicinity of the optimum produced during restarts, which is shown on the left-hand side of Figure 11 for a typical test run of 1000 iterations of α-annealing *cc*AS with 20 ants on the Eil51 benchmark. Comparing with the coarse diversity plot on the right-hand side it is apparent that the first few solutions in the vicinity of the optimum are discovered directly before the first restart. The curve flattens out immediately due to SPC and no further solutions would be discovered without cycling. With cycling, however, new good solutions in the vicinity of the optimum are continuously discovered after every restart (which approximately happen at iteration 500, 700 and from then on every 50 iterations) at a rate of 5 solutions per iteration. Further indication of this is given by Figure 12. The left-hand side shows a scatter plot of the cost of all solutions against the iteration number in which they are generated, the right-hand side shows iteration best and global best at every iteration. These figures show that the algorithm is effective in maintaining diversity in the proximity of the optimum.

4.1 Benchmarking

We perform testing on some standard TSP benchmarks [Rei01]. Comparison figures are taken from [SH00] with the exception of AS and ACS [DG96]. Blank

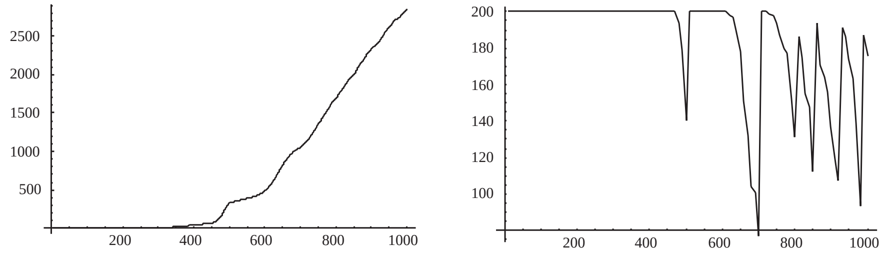

Fig. 11. Cumulative Count of Unique New Solutions within 10% of Optimum

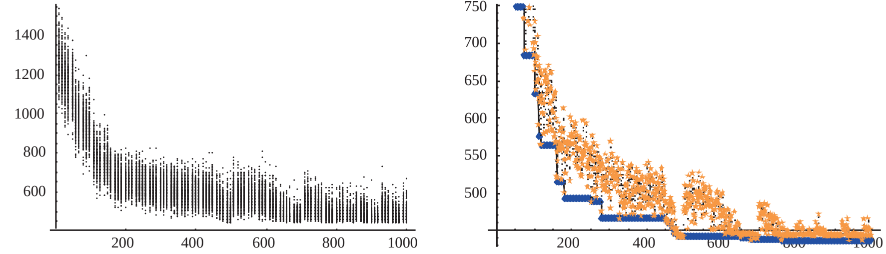

Fig. 12. Left: Development of Solution Distribution, Right: Iteration Best, Global Best

fields indicate missing data in the literature. The parameters used are standard ACS parameters that have been reported as producing good results for similar size problems in [SH00,DG96,RT02,BB98]: $Q = k = \beta = 1, \rho = 0.05$. The figures are averages over 50 runs (10,000 iterations with 96 ants for RY48P, 10,000 iterations with 51 ants for Eil51). While MaxMin AS and ACS clearly perform better, these results show that the combination of α-annealing and cycling has brought *cc*AS roughly into the range of these algorithms and that these mechansims may be useful as the basis of a different form of ACO.

We further explore the algorithm performance with some refinements. (1) To achieve increased diversity *between runs* we initialize the pheromone matrix with random values from the interval $(0, 0.1)$. α-annealing ensures robustness against outliers in the random initialization. (2) If there is no improvement of the global best over a large number of iterations (here: 2000) we restart the search with a complete re-initialization of the pheromone matrix. (3) The reward factor used is $\frac{Q}{l_k - l_{gb} + 1}$ as $\frac{Q}{l_k}$ does not produce sufficient selective pressure and pure elitist reward often caused the search to get trapped in local optima in pilot experiments.

To isolate the effects of α-annealing, cycling and restarts, we run six different versions of the algorithm on Eil51 and RY48P: plain AS, cAS (AS+cycling), αAS

Algorithm	RY48P			Eil51	
	Best	Avg	StdDev	Avg	StdDev
AS		16,845	783	442.2	9.04
ACS	14,422	14,625	142	428.1	
AS-rank+pts		14,645		434.5	
AS-elite+pts		14,658		428.3	
MaxMin AS+pts		14,523		427.1	
ccAS	14,422	14,782	160	429.5	1.85

Measurement	AS	cAS	αAS	αrAS	αcAS	ccAS
Eil51 - Best	428	430	428	427	427	427
Eil51 - Avg	442.2	439.2	436.9	433.2	431.4	429.5
Eil51 - Stddev	9.04	6.36	5.50	3.57	3.37	1.85
RY48P - Best	16,053	15,064	15,504	14,765	14,758	14,422
RY48P - Avg	16,845	15,334	15,836	14,930	15,279	14,782
RY48P - Stddev	783.6	224.2	357.1	134.3	379.4	160.0
FT53 - Best	7,796	7,188	7,320	7,170	7,177	7,170
FT53 - Avg	8,144.6	7,520.2	7,479.6	7,257.3	7,337.8	7,241.6
FT53 - Stddev	413.9	239.4	106.8	36.7	158.9	45.88

(AS+α-annealing), αrAS (AS+α-annealing+restarts), αcAS (AS+α-annealing+cycling), ccAS (AS+α-annealing+cycling+restarts). We show a performance comparison for these versions on the standard benchmarks Eil51 and RY48P. The results are averages over 20 runs with the same parameters as above. The ranking of the results clearly shows that the combination of all three features in *cc*AS yields the best results.

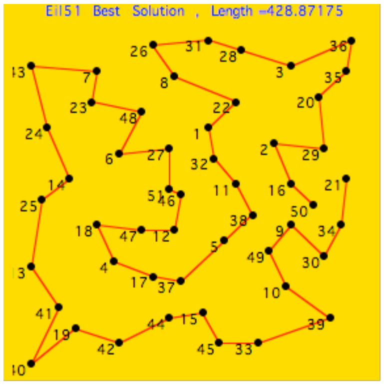

Fig. 13. Best Solution for Eil51

Finally, we give a quick comparison of *cc*AS with local optimization (2-opt), which is encouraging, however, not statistically significant. Using 2-opt the performance increases significantly and matches MaxMin AS+pts on Eil51 with an average of 427.3. We should therefore compare *cc*AS with other ACO algorithms that use local optimization. In [BB98] ACS with 2-opt was used with a parameter set optimized by a GA (see Section 2) and applied to EIL51 (real-valued). The study found a number of new solutions that improved on the optimum published in TSPLIB (429.9833). To the best of our knowledge, their best solution (length 429.118) is the best solution that has been reported for ACO algorithms. *cc*AS+2-opt found two new solutions to Eil51 (real-valued) that improve on these results. These have the lengths 428.982 and 428.871 (see Figure 13). The average solution quality generated by *cc*AS+2-opt over 20 runs was also quite remarkable (429.61, StdDev=0.721).

4.2 Conclusions and Future Directions

Diversity control is a crucial aspect of meta-heuristics and becomes even more important when continuous adaptiveness to dynamic problems is required. We have investigated two aspects of diversity control: the influence of α and the critical phase of the convergence behaviour. Based on this we have proposed to use a dynamic α and a dynamic restart method to maintain diversity throughout the search. Complementary to the standard diversity control mechanisms, which prevent complete convergence, we have shown how the critical phase of the system dynamics of a fully convergent algorithm can be utilized to effectively explore the search space in the vicinity of the optimum. The effectiveness of these mechansims has been demonstrated by embedding them into AS. These modifications seem a promising terrain for further investigation, but more extensive testing of *cc*AS on larger problems is obviously required to fully analyze the performance of this algorithm and to understand whether it may be able to compete directly with ACS and MaxMin AS when its parameters are tuned carefully.

However, the basic effects of these mechansims are interesting independently of the absolute performance of *cc*AS, as they may help us to better understand the convergence behaviour of related reinforcement-based search methods.

An important question for practical applications is whether the suggested diversity control mechanisms can directly be embedded into more effective types of ACO, like MaxMin AS or ACS. Our feeling is that α-annealing is likely to be useful. The case of cycling is more difficult as ACS and MaxMin AS already incorporate their own mechanisms to avoid full convergence.

We would also like to gain a better understanding of the kind of fitness landscapes on which these mechanisms are effective. Intuitively it is clear that they, like other methods that exploit local diversity, must be more effective on less rugged landscapes. How this relation can be captured formally is a challenging question for future research.

References

[BB98] H.M. Botee and E. Bonabeau. Evolving ant colony optimization. *Advances in Complex Systems*, 1:149—159, 1998.

[CD01] S. Camazine and J.L. Deneubourg. *Self-organization in biological systems.* Princeton University Press, Princeton, NJ, 2001.

[CHS02] O. Cordon, F. Herrera, and T. Sttzle. A review on the ant colony optimization metaheuristic. *Mathware and Soft Computing*, 9(2–3):141—175, 2002.

[DDC99] M. Dorigo and G. Di Caro. The ant colony mea-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11—32. McGraw Hill, London, 1999.

[DG96] M. Dorigo and L.M. Gambardella. Ant colony system. Technical Report TR/IRIDIA/1996-5, Universite Libre de Bruxelles, 1996.

[Dor92] M. Dorigo. *Optimization, learning and Natural algorithms.* PhD thesis, Poltecnico di Milano, 1992.

[Gut02] W.J. Gutjahr. ACO algorithms with guaranteed convergence to the optimal solution. *Information Processing Letters*, 82(3):145—153, 2002.

[NA01] Y. Nakamichi and T. Arita. Diversity control in ant colony optimization. In *Proc. Inaugural Workshop on Artificial Life (AL'01)*, pages 70–78, Adelaide, Australia, December 2001.

[Rei01] G. Reinelt, 2001. WWW repository. `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`.

[RT02] M. Randall and E. Tonkes. Intensification and diversification strategies in ant colony system. *Complexity International*, 9, 2002.

[SD02] T. Stützle and M. Dorgio. A short convergence proof for a class of ant colony optimization algorithms. *IEEE Transactions on Evolutionar Computation*, 6(4):358–365, 2002.

[SH00] T. Stützle and H. Hoos. MAX-MIN ant system. *Future Generation Computer Systems*, 16(8):889—914, 2000.

[ZBMD04] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo. Model-based search for combinatorial optimization: A critical survey. *Annals of Operations Research*, 2004. to appear.