Cross-Entropy Guided Ant-Like Agents Finding Cyclic Paths in Scarcely Meshed Networks

Otto Wittner and Bjarne E. Helvik

Department of Telematics, Norwegian University of Science and Technology
7491 Trondheim, Norway
{wittner,bjarne}@item.ntnu.no
http://www.item.ntnu.no/~wittner

Abstract. Finding paths in networks is a well exercised activity both in theory and practice but still remains a challenge when the search domain is a dynamic communication network environment with changing traffic patterns and network topology. To enforce dependability in such network environments new routing techniques are called upon. In this paper we describe a distributed algorithm capable of finding cyclic paths in scarcely meshed networks using ant-like agents. Cyclic paths are especially interesting in the context of protection switching, and scarce meshing is typical in real world telecommunication networks. Two new next-node-selection strategies for the ant-like agents are introduced to better handle low degrees of meshing. Performance results from Monte Carlo Simulations of systems implementing the strategies are presented indicating a promising behavior of the second strategy.

1 Introduction

Finding paths in networks is a well exercised activity both in theory and practice. Still it remains a challenge especially when the search domain is a dynamic communication network environment with changing traffic patterns and network topology. The internet is such an environment, and as an increasing number of applications demanding QoS guarantees is beginning to use internet as their major communication service, efficient and dependable routing in the network becomes more important than ever.

Protection switching [1] is a well known technique for improving dependability in communication networks and commonly used in larger SDH- and ATM-networks. To enable fast recovery from link or network element failures two (or more) disjunct independent paths from source to destination are defined, one primary and one (or more) backup path. Loss of connectivity in the primary path triggers switching of traffic to the backup path. Good dependability is achieved by allocating required resources for the backup path prior to the occurrence of failures in the primary path. However maintaining the necessary mesh of backup paths in a dynamic network with a large number of active sources and destinations is a complex task [2]. Grover & al. [3,4] propose to use simple cyclic paths (“p-cycles”) as a means for dependable routing in meshed networks. Protection rings are common in SDH based transport networks and guarantee protection against single link failures in the ring (assuming the ring has duplex links)[5]. All network elements on the ring can continue to communicate with each other

M. Dorigo et al. (Eds.): ANTS 2002, LNCS 2463, pp. 123–134, 2002.

after a single link failure by routing all traffic over the “healthy” curve-section of the ring (Figure 1). Thus a cyclic path can provide a dependable communication service for a set of sources and destinations. Assuming p-cycles can be found, the number of necessary cycles to be maintained in a network providing a dependable communication service to a set of network elements, is likely to be far less than the number of traditional backup paths required to provide the same service.

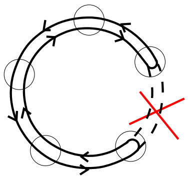

Fig. 1. Protection switching in a ring network.

In this paper we describe an algorithm, founded in rare event theory and cross entropy, able to find cyclic paths in networks. The fundamentals of the algorithm has previously be published in [6]. This paper enhances the original algorithm by enabling it to find cyclic paths in networks with low degrees of meshing, a common property of real world telecommunication networks. The algorithm is fully distributed with no centralized control which are two desirable properties when dependability is concerned. The algorithm can conveniently be implemented using simple (ant-like) mobile agents [7]. Section 2 introduces the foundations of the original algorithm and motivates the use of it. Section 3 describes the original next-node-selection strategy for the mobile agents as well as two new strategies. Emphasis is put upon search performance when searching for cycles in scarcely meshed networks. Section 4 presents results from Monte Carlo simulation of systems based on the strategies described in section 3. Finally section 5 summarizes, concludes and indicates future work.

2 Agent Behavior Foundations

The concept of using multiple mobile agents with a behavior inspired by foraging ants to solve routing problems in telecommunication networks was introduced by Schoonderwoerd & al. in [8] and further developed in [9,10,11]. Schoonderwoerd & al.’s work again builds on Dorigo & al.’s work on Ant Colony Optimization (ACO) [12]. The overall idea is to have a number of simple ant-like mobile agents search for paths between a given source and destination node. While moving from node to node in a network an agent leaves markings imitating the pheromone left by real ants during ant trail development. This results in nodes holding a distribution of pheromone markings pointing to their different neighbor nodes. An agent visiting a node uses the distribution of pheromone markings to select which node to visit next. A high number of markings pointing towards a node (high pheromone level) implies a high probability for an agent to continue its itinerary toward that node. Using trail marking agents together with a constant evaporation of all pheromone markings, Schoonderwoerd and Dorigo show that after a relatively short period of time the overall process converges towards having the majority of the agents following a single trail. The trail tends to be a near optimal path from the source to the destination.

2.1 The Cross Entropy Method

In [13] Rubinstein develops a search algorithm with similarities to Ant Colony Optimization [12,14]. The collection of pheromone markings is represented by a probability matrix and the agents' search for paths is a Markov Chain selection process generating sample paths in the network. ("Path" and "trail" are equivalent in this paper and will be used interchangeably.)

In a large network with a high number of feasible paths with different qualities, the event of finding an optimal path by doing a random walk (using a uniformly distributed probability matrix) is rare, i.e. the probability of finding the shortest Hamiltonian cyclic path (the Traveling Salesman Problem) in a 26 node network is $\frac{1}{25!} \approx 10^{-26}$. Thus Rubinstein develops his algorithm by founding it in rare event theory.

By importance sampling in multiple iterations Rubinstein alters the transition matrix and amplifies probabilities in Markov chains producing near optimal paths. Cross entropy (CE) is applied to ensure efficient alteration of the matrix. To speed up the process, a performance function weights the path qualities (two stage CE algorithm [15]) such that high quality paths have greater influence on the alteration of the matrix. Rubinstein's CE algorithm has 4 steps:

1. At the first iteration $t = 0$, select a start transition matrix $P_{t=0}$ (e.g. uniformly distributed).
2. Generate N paths from P_t using some selection strategy (i.e. avoid revisiting nodes, see section 3). Calculate the minimum Boltzmann temperature γ_t to fulfill average path performance constraints, i.e.

$$\min \gamma_t \text{ s.t. } h(P_t, \gamma_t) = \frac{1}{N} \sum_{k=1}^{N} H(\pi_k, \gamma_t) > \rho \tag{1}$$

   where $H(\pi_k, \gamma_t) = e^{-\frac{L(\pi_k)}{\gamma_t}}$ is the performance function returning the quality of path π_k. $L(\pi_k)$ is the raw cost of path π_k (e.g. delay in a telecommunication network). $10^{-6} \leq \rho \leq 10^{-2}$ is a search focus parameter. The minimum solution for γ_t will result in a certain amplification (controlled by ρ) of high quality paths and a minimum average $h(P_t, \gamma_t) > \rho$ of all path qualities in the current batch of N paths.
3. Using γ_t from step 2 and $H(\pi_k, \gamma_t)$ for $k = 1, 2..., N$, generate a new transition matrix P_{t+1} which maximizes the "closeness" to the optimal matrix, by solving

$$\max_{P_{t+1}} \frac{1}{N} \sum_{k=1}^{N} H(\pi_k, \gamma_t) \sum_{ij \in \pi_k} \ln P_{t,ij} \tag{2}$$

where $P_{t,ij}$ is the transition probability from node i to j at iteration t. The solution of (2) is shown in [13] to be

$$P_{t+1,rs} = \frac{\sum_{k=1}^{N} I(\{r,s\} \in \pi_k) H(\pi_k, \gamma_t)}{\sum_{l=1}^{N} I(\{r\} \in \pi_l) H(\pi_l, \gamma_t)} \tag{3}$$

which will minimize the cross entropy between P_t and P_{t+1} and ensure an optimal shift in probabilities with respect to γ_t and the performance function.

4. Repeat steps 2-3 until $H(\widehat{\pi}, \gamma_t) \approx H(\widehat{\pi}, \gamma_{t+1})$ where $\widehat{\pi}$ is the best path found.

2.2 Distributed Implementation of the Cross Entropy Method

Rubinstein's CE algorithm is centralized, synchronous and batch oriented. All results output from each step of the algorithm must be collected before the next step can be executed. In [6] a distributed and asynchronous version of Rubinstein's CE algorithm is developed. A few approximations let (3) and (1) be replaced by the autoregressive counterparts

$$P_{t+1,rs} = \frac{\sum_{k=1}^{t} I(\{r,s\} \in \pi_k) \beta^{t-k} H(\pi_k, \gamma_k)}{\sum_{l=1}^{t} I(\{r\} \in \pi_l) \beta^{t-k} H(\pi_l, \gamma_l)} \tag{4}$$

and

$$\min \gamma_t \text{ s.t. } h_t^{'}(\gamma_t) \ > \rho \tag{5}$$

respectively where

$$\begin{aligned} h_t^{'}(\gamma_t) &= h_{t-1}^{'}(\gamma_t)\beta + (1-\beta) H(\pi_t, \gamma_t) \\ &\approx \frac{1-\beta}{1-\beta^t} \sum_{k=1}^{t} \beta^{t-k} H(\pi_t, \gamma_t) \end{aligned}$$

and $\beta < 1$, step 2 and 3 can immediately be performed when a single new path π_t is found and a new probability matrix P_{t+1} can be generated.

The distributed CE algorithm may be viewed as an algorithm where search agents evaluate a path found (and calculate γ_t by (5)) right after they reach their destination node and then immediately return to their source node backtracking along the path. During backtracking relevant probabilities in the transition matrix are updated by applying $H(\pi_t, \gamma_t)$ through (4).

The distributed CE algorithm resembles Schoonderwoerd & al.'s original system. However Schoonderwoerd's ants update probabilities during their forward search. Dorigo & al. realized early in their work on ACO that compared to other updating schemes, updating while backtracking results in significantly quicker convergence towards high quality paths. Dorigo & al.'s AntNet system [9] implements updating while backtracking, thus is more similar to the distributed CE algorithm than Schoonderwoerd & al.'s system. However none of the earlier systems implements a search focus stage (the adjustment of γ_t) as in the CE algorithms.

2.3 P-Cycles, Hamiltonian Cyclic Paths, and CE Algorithms

Grover's "p-cycles" [3] provide protection against a single link failure on any link connecting the nodes which are on the path defined by the p-cycle. This includes both on-cycle links (links traversed by the path) as well as straddling links (links not traversed but having their end nodes on the path). Intuitively a Hamiltonian cyclic path, which by definition visits all nodes once in a network, would provide a cycle potentially able to protect against any single link failure. This is also argued in [16].

The CE algorithms from both [13] and [6] show good performance when tested on optimal Hamiltonian cyclic path search problems as long as the network environment is fully meshed (all nodes have direct duplex connections). Real world telecommunication networks are seldom fully meshed. An average node degree much lager than 5 is uncommon. Finding a single Hamiltonian cyclic path in a large network with such scarce meshing can itself be considered a rare event.

In the section 3.1 we describe the selection strategy (used in CE algorithm step 2) implemented in the original CE algorithms (both [13] and [6]). They strategy struggles to find Hamiltonian cyclic paths in our 26 node test network shown in Figure 2. In section 3.2 and 3.3 we suggest two new selection strategies intended to better cope with a network topology with scarce meshing.

3 Selection Strategies

3.1 Markov Chain without Replacement

The CE algorithms in [13] and [6] implement a strict next-hop selection strategy termed *Markov Chain Without Replacement* (MCWR) in [13]. No nodes are allowed to be revisited, except for the home node when completing a Hamiltonian cyclic path.

Let

$$X^i_{t,r}(s) = I(s \notin \mathbf{V}^i_t \lor ((\mathbf{G}_{t,r} \subseteq \mathbf{V}^i_t) \land s = hn^i))$$

where $I(\ldots)$ is the indicator function, $\mathbf{V}^i_t$ is agent i's list of already visited nodes, $\mathbf{G}_{t,r}$ is the set of neighbor nodes to node r and hn^i is agent i's home node. Thus $X^i_{t,r}(s)$ is 1 if node s has not already been visited by agent i, or if all neighbor nodes of r have been visited by agent i and s is agent i's home node.

When MCWR is applied $P_{t,rs}$ from (4) is weighted by $X^i_{t,r}(s)$ and renormalized giving a new next-hop probability distribution

$$Q^i_{t,rs} = \frac{[I(t > D)P_{t,rs}(1-\epsilon) + \epsilon]\, X^i_{t,r}(s)}{\sum_{\forall k} [I(t > D)P_{t,rs}(1-\epsilon) + \epsilon]\, X^i_{t,r}(k)}$$

where D is the number of path samples required to be found to complete the initialization phase of the system (step 1). The random noise factor ϵ is set to a small value, e.g. 10^{-60}. During the initialization phase agents are forced to explore since the next-hop probability vector $\mathbf{Q}^i_{t,r}$ will have a uniform distribution over the qualified ($X^i_{t,r}(s) = 1$) neighbor nodes. See [6] for more details about the initialization phase.

If $\sum_{s \in \mathbf{G}_{t,r}} X^i_{t,r}(s) = 0$, agent i has reach a dead end and in the MCWR strategy it is terminated. When the event of finding a Hamiltonian cyclic path is rare due to scarce meshing in a network, most agents will reach such dead ends. Thus only a few “lucky” agent will be able to contribute with a path in step 2 of the CE algorithm. This will slow down the search process significantly since CE algorithms require a “smooth” search space, i.e. many suboptimal solutions should exist in addition to the optimal solutions.

3.2 Markov Chain Depth First

Instead of immediately terminating agents when dead ends are reached a “retry mechanism” can be implemented. We have tested what we call the *Markov Chain Depth First* (MCDF) strategy which allows agents to backtrack and retry searching. An MCDF-agent performs a depth first search [17] from its home node, i.e. it tries to visit nodes in such an order that when a dead end is met (a leaf node is found) all nodes have been visited only once and the home node is a neighbor of the leaf node. If a dead end is reached and either all nodes has not been visited or the home node is not a neighbor node, the agent backtracks along its path one step before continuing the search.

Let

$$X^{i*}_{t,r}(s) = I(s \notin (\mathbf{V}^i_t \cup \mathbf{D}^i_{t,r}) \ \vee \ ((\mathbf{G}_{t,r} \subseteq \mathbf{V}^i_t) \ \wedge \ s = hn^i))$$

where $\mathbf{D}^i_{t,r}$ is the set of neighbor nodes of r leading to dead ends for agent i. Thus $X^{i*}_{t,r}(s)$ is 1 if node s has not already been visited by agent i and s does not lead to a dead end, or (as for $X^i_{t,r}(s)$) if all neighbor nodes of r have been visited by agent i and s is agent i’s home node.

All $\mathbf{D}^i_{t,r}$ (for $\forall_r$) are stored in a stack managed by agent i. When a fresh next node r_{+1} is chosen $\mathbf{D}^i_{t,r_{+1}} \equiv \emptyset$ is pushed onto the stack. If a dead end is reached at node r_{+1} agent i backtracks to the previously visited node r, removes (pops) $\mathbf{D}^i_{t,r_{+1}}$from the stack and adds r_{+1} to $\mathbf{D}^i_{t,r}$ (which is now on the top of the stack).

When MCDF is applied $P_{t,rs}$ is weighted by $X^{i*}_{t,r}(s)$ in the same way $X^i_{t,r}(s)$ is for MCWR. Results in section 4 show simulation scenarios for MCDF-agents both with unlimited and limited backtracking. Unlimited backtracking implies never terminating agents but letting them search (in depth first fashion) until they find Hamiltonian cyclic paths. Limited backtracking implements a quota of backtracking steps in each agent, i.e. a certain no of “second chances” or “retries” are allow for an agent before termination.

A method with similarities to MCDF is presented in [18].

3.3 Markov Chain with Restricted Replacement

By relaxing the agent termination condition even more, we arrive at what we call the *Markov Chain with Restricted Replacement* (MCRR) strategy. The fundamental difference between this strategy and both MCWR and MCDF is a less strict condition concerning revisits to nodes. Revisits are simply allowed, but only when dead ends are reached. To ensure completion of cycles the home node is given priority when a dead end is reached and the home node is a neighbor node.

Let

$$\begin{aligned} X^{i**}_{t,r}(s) = \overline{I(}((\mathbf{G}_{t,r} \not\subseteq \mathbf{V}^i_t) \ \wedge \ s \in \mathbf{V}^i_t) \vee \\ ((\mathbf{G}_{t,r} \subseteq \mathbf{V}^i_t) \ \wedge \ hn^i \in \mathbf{G}_{t,r} \ \wedge \ s \neq hn^i)) \end{aligned}$$

where $\overline{I}(\ldots)$ is the inverse indicator function. Thus $X^{i*}_{t,r}(s)$ is zero if an unvisited neighbor node to r exists and s has already been visited, or if all neighbor nodes have been visited and the home node is a neighbor node but s is not the home node. As for MCWR and MCDF $X^{i**}_{t,r}(s)$ weights $P_{t,rs}$.

In our simulations we consider only paths found by MCRR-agents which have visited all nodes when they return to their home node. Several of these agents will find closed acyclic paths (with loops), i.e. none Hamiltonian cyclic paths. The search space is now "smoother" and a range of suboptimal solutions exists (most none Hamiltonian). This enables step 2 in the CE algorithm to be executed with close to the same efficiently as for a fully meshed network.

However when using MCRR-agents there is no longer guaranteed that the best path found when the system converges is a Hamiltonian cyclic path. Since the agents visit all nodes, the length of acyclic closed paths where nodes have been revisited, are likely to be longer than Hamiltonian cyclic paths. Thus finding the shortest Hamiltonian cyclic path (minimization) may still be achievable.

We realize that the above statement does not hold in general since a network topology may be constructed having its shortest Hamiltonian cyclic path longer than one or more closed acyclic paths visiting all nodes. However in the context of p-cycle design closed acyclic paths may still provide protection against single link failures.

Results in section 4 are promising. MCRR outperforms both MCWR and MCDF when it comes to speed of convergence, and do in all simulation scenarios converge to a Hamiltonian cyclic path.

4 Strategy Performance

As for the simulation scenarios in [6] we have used an active network enabled version of the *Network Simulator* version 2 [19] to test the three selection strategies. Our test network topology is show in Figure 2, a 26 node network with an average number of outgoing links (degree) per node equal to 5. The low average degree implies existence of far less Hamiltonian cyclic paths compared to a fully meshed 26 node network with number of Hamiltonian cyclic paths equal to $25! \approx 10^{26}$. The network topology was generated by the *Tier 1.1* topology generator [20] with the parameter vector "1 0 0 26 0 0 9 1 1 1 1 9".

All scenarios had equal parameter settings (except for different selection strategies): $D = 19$, $\mu = 26$, $\beta = 0.998$, $\rho = 0.01$, $\rho^* = 0.95$, where μ is the number of agents operating concurrently and ρ^* the ρ reduction factor. See [6] for further descriptions of the parameters.

Table 1 and 2 compare results form simulation scenarios for the different selection strategies. Results shown are values recorded after 100, 1500 and 10 000 seconds of simulation time. Time has been chosen as the scale of progress rather than number of iterations since the time spent per iteration (per search) vary significantly between the different strategies. Simulation time is expected to be approximately proportional to real time in a network.

Columns 2, 3 and 4 in Table 1 show the total number of Hamiltonian cyclic paths found including re-discoveries. Values are averaged over 10 simulations and reported

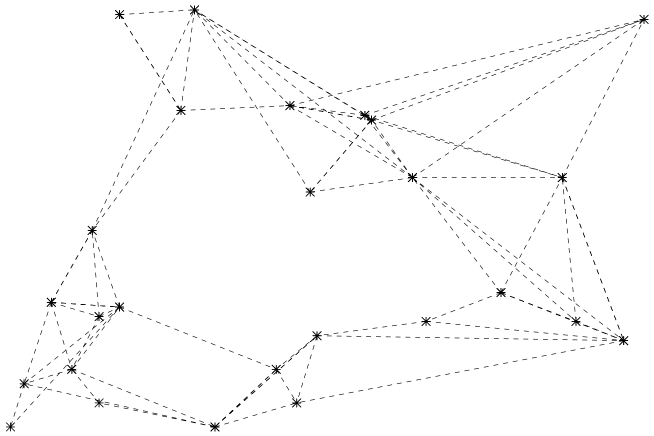

Fig. 2. Network topology used in simulation scenarios. The topology is generated by the *Tiers 1.1* topology generator [20].

Table 1. Number of paths found in simulation scenarios using the three different selection strategies. Values are averaged over 10 simulations and reported with standard deviations (prefixed by ±). Numbers in bracket are the number of agents which have finished their initialization phase.

Test	*No of Hamiltonian cyclic paths found*			*No of none Hamiltonian cyclic paths*		
scenarios	*100s*	*1500s*	*10 000s*	*100s*	*1500s*	*10 000s*
MCWR	1.0 ±0 (0)	16.0 ±2.3 (0)	109 ±8.7 (0)	16.0e3 ±52.1	246e3±213	1.6e6±533
MCDF unlimited	0.0±0 (0)	1.0 ±0.0 (0)	2.57 ±1.3 (0)	201±30.6	1.5e3±79.1	9.3e3±190
MCDF quota=10	1.0 ±0 (0)	14.0 ±4.5 (0)	85.0 ±12.4 (0)	8.8e3±24.3	134e3±156	901e3±345
MCDF quota=5	2.0 ±0.9 (0)	16.0 ±4.4 (0)	95.0 ±10.8 (0)	10.1e3±34.1	155e3±151	1.0e6±402
MCDF quota=2	1.0 ±0 (0)	14.0 ±3.7 (0)	99.0 ±12.1 (0)	11.5e3±40.9	176e3±126	1.2e6±316
MCDF quota=1	2.0 ±1.0 (0)	16.0 ±5.5 (0)	105±12.9 (0)	12.3±35.5	188e3±101	1.3e6±367
MCRR	7.0 ±2.8 (26)	**52 800±4620 (26)**	(converged)	4.9e3±56.0	**15.7e3±1580**	(converged)

Table 2. Quality of paths found in simulation scenarios using the three different selection strategies. Values are based on 10 simulation. Standard deviations are prefixed by ± and the worst of the best values are given in brackets.

Test	*Best path found (Worst of best paths found)*			*Average path cost*		
scenarios	*100s*	*1500s*	*10 000s*	*100s*	*1500s*	*10 000s*
MCWR	0.199 (0.251)	0.197 (0.214)	0.193 (0.207)	0.231±0.015	0.232±0.004	0.231±0.004
MCDF unlimited	(no paths found)	0.250 (3.273)	0.215 (0.750)	(no paths found)	0.923±1.023	254.5±670.5
MCDF quota=10	0.202 (0.371)	0.202 (0.222)	0.194 (0.207)	0.260±0.058	0.254±0.008	0.254±0.008
MCDF quota=5	0.201 (0.240)	0.201 (0.221)	0.196 (0.209)	0.228±0.014	0.240±0.011	0.243±0.008
MCDF quota=2	0.198 (0.293)	0.194 (0.224)	0.194 (0.206)	0.244±0.028	0.241±0.006	0.241±0.005
MCDF quota=1	0.204 (0.258)	0.201 (0.210)	0.193 (0.204)	0.231±0.016	0.235±0.003	0.238±0.003
MCRR	0.203 (0.230)	**0.194 (0.202)**	(converged)	0.514±0.037	**0.206±0.015**	(converged)

with standard deviations (prefixed by ±). Numbers in bracket are the number of agents which have finished their initialization phase, i.e. changed search behavior from doing random walk guided only by a selection strategy to being guided both by a selection strategy and cross entropy adjusted probabilities (pheromones). Column 5, 6 and 7 show the total number of none Hamiltonian cyclic paths found, including dead ends and paths not visiting all nodes.

Column 2, 3 and 4 in Table 2 show the best of the best paths found (lowest cost) in 10 simulations with the worst of the best in brackets. And finally columns 5, 6 and 7 show path cost averaged over 10 simulations and reported with standard deviations (prefixed by ±).

4.1 Markov Chain without Replacement

As expected in the MCWR scenario few Hamiltonian cyclic paths are found even after 1500 simulation seconds. The scarce meshing in the test network results in many agent reaching dead ends. The fraction of feasible to infeasible paths found is as low as $6.5 \cdot 10^{-5}$ after 1500 seconds. The paths found are of relatively good quality, but still after 10 000 seconds none of the agents have managed to collect enough paths samples to finish their initialization phases.

These results indicate the need for a different selection strategy when searching for Hamiltonian cyclic paths in networks with scarcely meshed topologies.

4.2 Markov Chain Depth First

The results for the MCDF scenarios are not promising. When unlimited backtracking is enabled even fewer path are found than for the MCWR scenario. The path quality is low because the total search time (including backtracking) is registered as path cost. This is not surprising since unlimited backtracking implies that every agent is doing an exhaustive search for Hamiltonian cyclic paths. The very reason for introducing stochastic search techniques in the first place is to avoid the need for such exhaustive searches.

When the MCDF strategy is limited by quotas of 10, 5, 2 or 1 retry the performance improves compared to unlimited backtracking but is not better than the original MCWR strategy. Also with this strategy not enough valid paths have been found after 10 000 seconds to enable any agents to complete their initialization phase. Best and average path costs are similar to the values for MCWR and no convergence is observed due to the overall low number of valid paths found.

4.3 Markov Chain with Restricted Replacement

The last row in Table 1 and 2 which presents results from MCRR scenarios, stand out from the other results. Already after 100 simulation seconds all 26 agents have completed their initialization phases, and after 1500 seconds the average path cost has converged to a value close to the best path found (see bold values). Since full convergence is experienced already around 1500 seconds, no values are given for 10 000 seconds. For all 10 simulations the best path found is a true Hamiltonian cyclic path. Figure 3 show how the ratio of Hamiltonian to none Hamiltonian cyclic paths found increases during the search process. Not long after 800 seconds most agents start finding Hamiltonian cyclic paths.

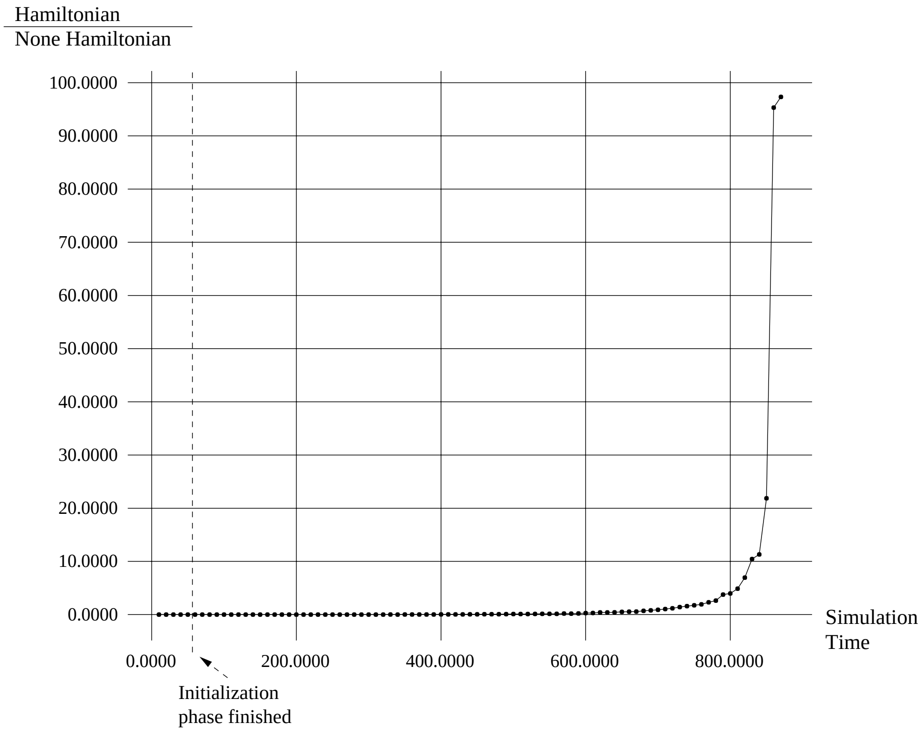

Fig. 3. Ratio of Hamiltonian to none Hamiltonian cyclic paths found when using the MCRR selection strategy.

We believe these results indicates the value of using infeasible paths (none Hamiltonian cyclic paths) as intermediate solutions during the iterative search process of the CE algorithm. The question of accepting or rejecting infeasible solutions is well known to designers of evolutionary systems [21,22]. When feasible solutions are located far from each other in the search domain a significant speedup in the search process can be achieved by letting infeasible solutions act as “stepping stones” between suboptimal and optimal feasible solutions.

5 Concluding Remarks

As an increasing number of applications demanding QoS guarantees are accessing internet, dependable routing is becoming more important than ever. This paper examines and enhances a distributed cross entropy founded algorithm designed to find cyclic paths (Hamiltonian cyclic paths) in networks. Such paths are argued to be good candidate paths when protection switching is to be implemented in meshed networks using p-cycles [16]. The algorithm presented is well suited for distributed implementation, for instance using mobile agents or active networks technology.

Previous versions of the algorithm [13,6] struggle to find Hamiltonian cyclic paths in scarcely meshed network, very much because of the strict selection strategy (Markov Chain without Replacement) in operations during the search process. In this paper we compare the original selection strategy with two new strategies. The first new strategy, *Markov Chain Depth First*, proves to be as inefficient as the original, while the second, *Markov Chain with Restricted Replacement*, outperforms both the other strategies. However, using the second strategy convergence towards feasible solutions (i.e. Hamiltonian cyclic paths) is not guaranteed. Even so results from simulation scenarios indicate a high probability of converging towards near optimal Hamiltonian cyclic paths.

More excessive parameter tuning of the algorithm is required. Also the possibility of including heuristics to speed up the convergence process even more should be investigated.

Currently a new version of the algorithm is under development which allows several species of agents compete in finding quality paths in a network. By this a set of cyclic paths providing a overall high quality p-cycle design may potentially be found.

References

1. S. Aidarous and T. Plevyak, ed., *Telecommunications Network Management into the 21st Century*. IEEE Press, 1994.
2. Otto Wittner and Bjarne E. Helvik, "Cross Entropy Guided Ant-like Agents Finding Dependable Primary/Backup Path Patterns in Networks," in *Proceedings of Congress on Evolutionary Computation (CEC2002)*, (Honolulu, Hawaii), IEEE, May 12-17th 2002.
3. W.D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proceedings of IEEE International Conference on Communications*, vol. 1, pp. 537 –543, 7-11 June 1998.
4. D. Stamatelakis and W.D. Grover, "Rapid Span or Node Restoration in IP Networks using Virtual Protection Cycles," in *Proceedings of 3rd Canadian Conferance on Broadband Research (CCBR'99)*, (Ottawa), 7 November 1999.
5. M. Decina and T. Plevyak (editors), "Special Issue: Self-Healing Networks for SDH and ATM," *IEEE Communications Magazine*, vol. 33, September 1995.
6. Bjarne E. Helvik and Otto Wittner, "Using the Cross Entropy Method to Guide/Govern Mobile Agent's Path Finding in Networks," in *Proceedings of 3rd International Workshop on Mobile Agents for Telecommunication Applications*, Springer Verlag, August 14-16 2001.
7. Vu Anh Pham and A. Karmouch, "Mobile Software Agents: An Overview," *IEEE Communications Magazine*, vol. 36, pp. 26–37, July 1998.
8. R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based Load Balancing in Telecommunications Networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1997.
9. G. D. Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, Dec 1998.
10. T. W. B. P. F. Oppacher, "Connection Management using Adaptive Mobile Agents," in *Proceedings of 1998 International Conference on Parallel and Distributed Processing Techniques and Applications (PDAPTA'98)*, 1998.

11. J. Schuringa, “Packet Routing with Genetically Programmed Mobile Agents,” in *Proceedings of SmartNet 2000*, (Wienna), September 2000.
12. Marco Dorigo and Gianni Di Caro, “Ant Algorithms for Discrete Optimization,” *Artificial Life*, vol. 5, no. 3, pp. 137–172, 1999.
13. Reuven Y. Rubinstein, “The Cross-Entropy Method for Combinatorial and Continuous Optimization,” *Methodology and Computing in Applied Probability*, pp. 127–190, 1999.
14. M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, “Model-based Search for Combinatorial Optimization,” IRIDIA IRIDIA/2001-15, Universite Libre de Bruxelles, Belgium, 2000.
15. Reuven Y. Rubinstein, “The Cross-Entropy and Rare Events for Maximum Cut and Bipartition Problems - Section 4.4,” *Transactions on Modeling and Computer Simulation*, To appear.
16. D. Stamatelakis and W.D. Grover, “Theoretical Underpinnings for the Efficiency of Restorable Networks Using Preconfigured Cycles ("p-cycles"),” *IEEE Transactions on Communications*, vol. 48, pp. 1262–1265, August 2000.
17. Kenneth A. Ross and Charles R.B. Wright, *Discrete Mathematics*. Prentice Hall, 2nd ed., 1988.
18. V. Maniezzo, “Exact and Approximate Nondeterministic Tree-Search Procedures for the Quadratic Assignment Problem,” *INFORMS Journal on Computing*, vol. 11, no. 4, pp. 358–369, 1999.
19. DARPA: VINT project, “UCB/LBNL/VINT Network Simulator - ns (version 2).” http://www.isi.edu/nsnam/ns/.
20. Calvert, K.I. and Doar, M.B. and Zegura, E.W. , “Modeling Internet Topology ,” *IEEE Communications Magazine*, vol. 35, pp. 160 –163, June 1997.
21. Goldberg, D., *Genetic Algorithms in Search, Optimization and MachineLearn ing*. Addison Wesley, 1998.
22. Z. Michalewicz, *Genetic algorithms + Data Stuctures = Evolution Programs*. Springer Verlag, second ed., 1996.