# Distributed mining of the Internet for novel news: Evolutionary community of news foragers

Zsolt Palotai
Department of Information Systems,
Eötvös Loránd University,
Pázmány Péter sétány 1/C,
Budapest, Hungary H-1117.
E-mail: zspalotai@vnet.hu

Sándor Mandusitz
Department of Information Systems,
Eötvös Loránd University,
Pázmány Péter sétány 1/C,
Budapest, Hungary H-1117.
E-mail: santi@sent.com

András Lőrincz
Department of Information Systems,
Eötvös Loránd University,
Pázmány Péter sétány 1/C,
Budapest, Hungary H-1117.
E-mail: lorincz@inf.elte.hu.

*Abstract*— **We populated a huge scale-free portion of Internet environment with news foragers. They evolved by a simple internal selective algorithm: selection concerned the memory components, being finite in size and containing the list of most promising supplies. Foragers received reward for locating not yet found news and crawled by using value estimation. Foragers were allowed to multiply if they passed a given productivity threshold. A particular property of this community is that there is no direct interaction (here, communication) amongst foragers. It is found that, still, fast compartmentalization, i.e., fast division of work can be achieved.**

## I. Introduction

We have developed a novel artificial life (Alife) method with intelligent individuals (agents) to detect 'breaking news' type information on a prominent and vast WWW domain. We turned to Alife to achieve efficient division of labor under minimal communication load (i.e., no direct interaction) between individuals. All components, but direct interaction, seem necessary in our algorithm to achieve efficient cooperation. There are many different aspects of this work, including evolution (for reviews on relevant evolutionary theories, see, e.g., [1], [2] and references therein), the dynamics of self-modifying systems (see, e.g., [3], [4] and references therein), and swarm intelligence (for a review, see [5]).

A particular feature of this study is that evolution occurs in a scale-free world, the WWW [6], [7]. A graph is a scale-free network if the number of incoming (or outgoing or both) edges follows a power-law distribution ($P(k) \propto k^{-\gamma}$, where $k$ is integer, $P(k)$ denotes the probability that a vertex has $k$ incoming (or outgoing, or both) edges and $\gamma > 0$). The direct consequence of the scale-free property is that there are numerous URLs or sets of interlinked URLs, which have a large number of incoming links. Intelligent web crawlers can be easily trapped at the neighborhood of such junctions as it has been shown previously [8], [9]. The intriguing property of scale-free worlds is their abundance in nature [10]: Most of the processes, which are considered evolutionary, seem to develop and/or to co-occur in scale-free structures.

The other feature of our study is that fitness is not determined by us and that fitness is implicit. Similar concepts have been studied in other evolutionary systems, where organisms compete for space and resources and cooperate through direct interaction (see, e.g., [11] and references therein.) Here also, fitness is determined by the external world *and* by the competing individuals, together. Also, our agents crawl by estimating the long-term cumulated profit using reinforcement learning (for a review, see, e.g., [12]). In our estimation, function approximation and temporal difference learning are utilized. Reinforcement learning has also been used in concurrent multi-robot learning, where robots had to learn to forage together via direct interaction [13]. The lack of explicit measure of fitness, the lack of our control over the environment, the value estimation executed by the individuals, and the *lack of direct interaction* distinguish our work from most studies using genetic, evolutionary, and reinforcement learning algorithms.

First, the applied methods are described in details (Section II). Experimental results are provided in Section III followed by a discussion and a summary (Section IV and V, respectively).

## II. Methods

In our model, individuals are *foragers*, 'populating' a continuously changing world, where the rate of the emergence of new resources is limited. An approximate overview of the full algorithm is provided in high-level pseudo-code in Fig. 1.

**Environment.** The domain of our experiments, the CNN news sites, was scale-free according to our measurements. The distributions of both incoming and outgoing links show a power distribution (Fig. 2(a)). Foragers 'live' and 'move' in this environment. A forager may make a step and arrive to an URL. Links and the document of the URL are investigated by the forager upon the step is made. The forager visits URL 'A', downloads the not-yet visited part of the environment (documents of URLs, which URLs have not been visited yet by that forager and are linked from URL 'A'). Downloading is followed by evaluation of the documents and 'decision making' about which URL is to be visited next. The inset shows the links and documents investigated by a forager when it takes a step. The distributions, shown in the figure, correspond to links investigated by the foragers.

**The foragers.** 'Food' of foragers corresponds to novel news. Positive reward is delivered only to the first sender of a given news item. Foragers have two components of

0-7803-8359-1/04/$20.00 ©2004 IEEE

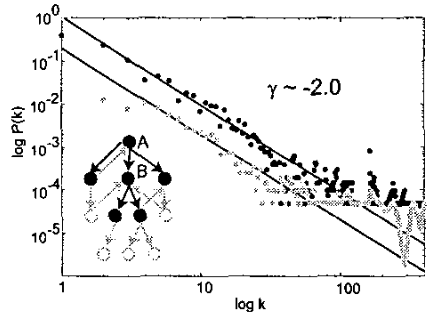
```
Initialize two foragers
    Set one of them to state 'next'
Repeat
    start next forager
        select next step (always avoid loops)
        either
            move to expected best document
        or
            save the previous good URL's in the weblog
            and move to a new start url of the weblog
        download neighboring documents
            and add those to found documents
        send new documents, measure immediate reward
            and tune value estimation
        estimate values of found documents
    this forager goes to sleep if its time period is over
    multiply (delete) forager if it is good (inefficient)
    decide, which forager is next
Until time is over
```

Fig. 1. **High level pseudo-code of the algorithm**

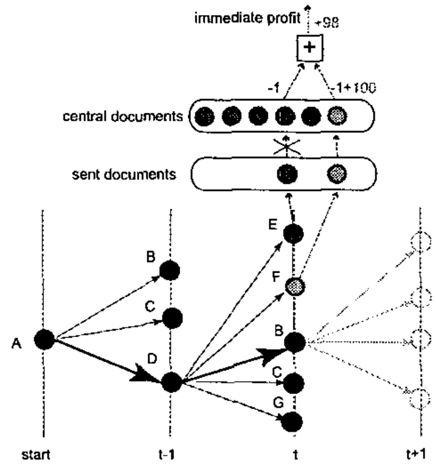
long-term memory, an adaptive component and a selective component. The adaptive component serves decision making and determines the *behavior* of the forager. Foragers with different behaviors may make different decisions at the same URL. The other component is a finite-sized memory, that we call weblog, which is subject to evaluation and selection. This memory contains the list of *food rich areas*, that is, good seed links, good starting URLs. First, this adaptive component and the related machine learning algorithms will be detailed.

**Reward system.** Foragers are searching for 'food', which is novel news. At each visited page, the forager downloads the neighboring documents and determines whether a document has a time stamp of the current (actual) date. If it does, this document is sent to the center (these documents will be referred as 'sent' documents). A central reinforcing agent administers rewards and costs. Positive reward is delivered only to the first sender of a given news item only if the document's time stamp is not older than a day according to GMT. Then reward $c_+$ ($= 100$ in arbitrary units (a.u.)) is 'provided'. Each sending of a document, costs $c_-$ ($= -1$ a.u.) for the forager. The (immediate) profit is the difference of rewards and costs at any given step. Figure 2(b) shows the reinforcement of a step in an example path: at start (time $t-2$), the agent is at URL 'A', where documents of neighboring URLs 'B', 'C' and 'D' are downloaded. URL 'D' is visited next. Documents of URLs 'E', 'F' and 'G' are downloaded. Document of URL 'G' has an obsolete date. Documents of URLs 'E' and 'F' are sent to the center. Document of URL 'F' is novel to the center, so it is rewarded. In turn, profit 98 is received by the forager. The forager maintains a list of neighbors of visited URLs, called *frontier* (see later). It can only visit URLs of the *frontier.*

**Long term cumulated profit. (LTP)** Immediate profit is a myopic characterization of an URL. Foragers have an adaptive continuous value estimator and follow the *policy* that maximizes the expected LTP instead of the immediate profit. Such estimators can be easily realized in neural systems [12], [14], [15]. Policy and profit estimation are interlinked concepts: profit estimation determines the policy, whereas policy influences choices and, in turn, the expected LTP. (For a review, see [12].) Here, choices are based on the greedy LTP policy: The forager visits the URL, which belongs to the *frontier* and has the highest estimated LTP. Visited URLs form a *path* and each path is limited to 100 steps.




(a) Power law




(b) Reinforcement

Fig. 2. **Scale-free properties of the Internet domain and reward system.** **(a):** Log-log scale distribution of the number of (incoming and outgoing) links of all URLs found during the time course of investigation. Horizontal axis: number of edges (log $k$). Vertical axis: relative frequency of number of edges at different URLs (log $P(k)$). Black (gray) dots: incoming (outgoing) edges of URLs. Slope of the straight lines $-2.0 \pm 0.3$. *Inset:* method of downloading. Black (gray) link: (not) in database. Solid (empty) circle: document (not) in database. **(b):** Central reinforcement system. Black (gray) circles: (not) novel documents. Positive (negative) numbers: reward and profit (cost). Vertical dashed lines: consecutive time steps. Dots on the $(t+k)^{th}$ dashed line: documents available at time step $t+k-1$.

In the particular simulation each forager has a $k(= 50)$ dimensional PrTFIDF text classifier [16] generated on a previously downloaded portion of the Geocities database. 50 clusters were created by Boley's clustering algorithm [17] from the downloaded documents. The PrTFIDF classifier was trained on these clusters plus an additional one, the $(k+1)^{th}$, representing general texts from the internet. The PrTFIDF outputs were non-linearly mapped to interval [-1,+1] by a hyperbolic-tangent function. The classifier was applied to reduce the texts to a small dimensional representation. When the forager visits URL 'A' displaying document $d_a$, the output vector of the classifier is $\mathbf{s_a} = (s_a(1), \ldots, s_a(k))$. (The $(k+1)^{th}$ output was dismissed.)

A linear function approximator is used for LTP estimation. It encompasses $k$ parameters, the *weight vector* $\mathbf{w} = (w(1), \ldots, w(k))$. The LTP of document $d_a$ is estimated as the scalar product of $\mathbf{s_a}$ and $\mathbf{w}$: $L(d_a) = \sum_{i=1}^{k} w(i) s_a(i)$. This weight vector is part of the forager's *long-term memory*. The weight vector of each forager is tuned by temporal difference learning [18], [14], [15]: Let us denote the document to be visited next by $d_n$, the output of the classifier by $\mathbf{s_n}$ and the estimated LTP of the document by $L(d_n) = \sum_{i=1}^{k} w(i) s_n(i)$. Assume that leaving the actual document $d_a$ and arriving to the next document, we have immediate profit $r_n$. Our estimation is perfect if $L(d_a) = L(d_n) + r_n$. Future profits are typically discounted in such estimations: $L(d_a) = \gamma L(d_n) + r_n$, where $0 < \gamma < 1$. In turn, the error of value estimation is

$$\delta(a,n) = r_n + \gamma L(d_n) - L(d_a).$$

Throughout the simulations $\gamma = 0.9$ was used. At each actual step $d_a \to d_n$ the weights of the value function were tuned to decrease the error of value estimation for the visited documents. This estimation error was used to correct the parameters: the $i^{th}$ component of the weight vector $w_i$ was corrected by

$$\Delta w_i = \alpha\, \delta(a,n)\, s_a(i) \quad .$$

with $\alpha = 0.1$ and $i = 1, \ldots, k$.

**URL lists and decisions.** Each forager maintains a selective long-term memory component, the *weblog*, which is a list of 100 URLs. *Starting points* are the first 10 elements of the weblog. At the start of a path the forager makes a random choice amongst starting points and visits that URL. After a path is finished the forager selects a new starting point for the next path.

For an URL 'A', the cumulated reward is the sum of immediate rewards collected during the path after visiting URL 'A'. Denoting the cumulated reward of URL 'A' by $R_{path}(A)$, when a path is completed, the *value of the URL 'A'*, denoted by $V(A)$ is estimated as follows:

$$V_{new}(A) = (1-\beta)V_{old}(A) + \beta R_{path}(A)$$

where $\beta$ was set to 0.3. If URL 'A' did not have a value before, then $V_{new}(A)$ is set to $R_{path}(A)$. These values are then used to update the weblog after each path. URLs are ordered by decreasing value and the list is clipped at the $100^{th}$ URL to form the new weblog.

The forager also maintains two short-term memory lists during each path. One of the lists contains URLs visited during the path to avoid loops. The other list is the *frontier*, which contains the URL's of pages directly accessible from the visited pages, excluding the visited URLs themselves. Forager selects its next step from this list. If the list is empty or the actual path has length of 100 steps then this path is finished.

**Multiplying by bipartition and extinction.** Every forager has a value of 100 at start. The value is reduced by 0.05 for each document sent to the center and increased by 1 if a sent document is reinforced. Once the forager's value reaches 200, the forager multiplies by bipartition and value 100 is assigned to both descendants. The weblog of the parent is randomly separated into two 50 element lists. The original weight vector of the parent and the partial weblogs are passed on to the descendants. On the other hand, if the forager's value hits 0 then it dies out.

**Foraging periods.** Foragers run sequentially in a prescribed order for approximately equal time intervals on one PC (the analysis of the last visited URL (downloading neighboring URLs and LTP estimation) is allowed). The foraging period is the time interval while all foragers run once. Unfinished paths are continued in the next run.

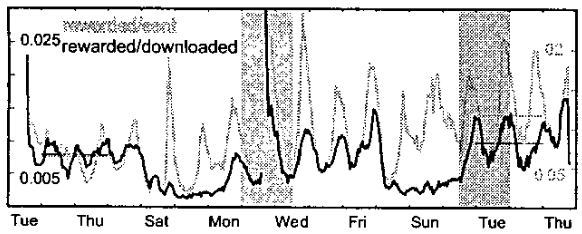
## III. Experimental results

Multiple two-week long experiments were conducted over a three month long period. Apart from short breaks, monitoring was continuous for each two week period. Most of the figures in this article represent a single two week period. The parameters of this experiment are representative to the entire series: The number of downloaded, sent and reinforced documents was 1,090,074, 94,226 and 7,423, respectively. The used Internet bandwidth (1Mbps on average) was almost constant, decreasing slowly by 10 % towards the end of the experiment. The number of foragers increased from 2 to 22. Experiments were run on a single computer. Within each foraging-period, the allocated time of every forager was 180 s. The actual time was always lager by 50 ± 30s, because foragers were always allowed to complete the analysis of the last URL after their allocated time expired. The net duration of a 100 step path was 350s.

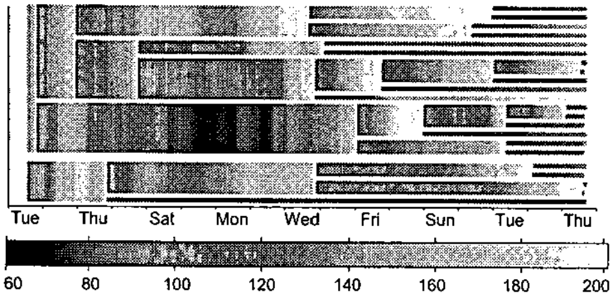
### A. Time lag and multiplication

In the presented experiments we applied the cost/reward ratio described in Section II. Larger cost/reward ratio increased the foraging areas and we noted a sudden increase in extinction probability.

Time-lag between publishing news and finding those decreases already after a few days (see Fig. 3(a)): the ratio of maxima to minima of the curves increases; and also, fewer news published on Friday were picked up on Saturday, a day late, during the second weekend of the experiment than during the first. Further gains in downloading speed are indicated by the relative shift between lighter gray and darker gray peaks. The darker gray peaks keep their maxima at around midnight GMT, lighter gray peaks shift to earlier times by about 6 hours. The shift is due to changes in the number of sent documents. The minima of this number shifts to around 6:00 P.M. GMT, when it is around 3:00 A.M. in Japan. (Identical dates can

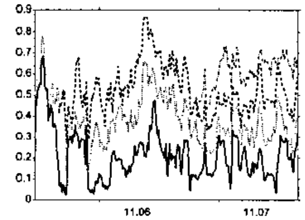



(a) Forager efficiency

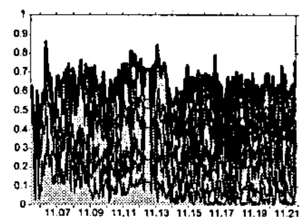



(b) Population value vs. time

Fig. 3. **Experimental results. (a):** The rate of sent and rewarded documents showed daily oscillations. Lighter (darker) gray curve: the ratio of rewarded to sent (rewarded to downloaded) documents. Horizontal lines: average values of the two curves during two workdays. Light (darker) gray regions: number of foragers is 6 (number of foragers increases from 14 to 18). **(b):** Forager population and forager values vs. time. The scores of the foragers are given with different gray levels. Starting of white horizontal line means forager multiplication. (There was a break in our Internet connection in the middle of the second week.)



(a) Early development



(b) Two week period

Fig. 4. **Division of work.** Horizontal axis in 'month.day' units. **(a):** Number of sites visited by only one forager relative to the number of all visited sites in a finite time period ($\approx 75mins$). Contribution of a single forager is superimposed on cumulated contributions of older foragers. The difference between 1.0 and the upper boundary of the curves corresponds to the ratio of sites visited by more than one forager. Duration: about three days. **(b):** Same for 16 day period.

be found for a 48 hour period centered around noon GMT.) Maxima of the relevant documents are at around 11:00 P.M. GMT (around 6:00 P.M. EST of the US). During the first week, the horizontal lines (average values of the corresponding curves during 2 workdays) are very close to each other. Both averages increase for the second week. The ratio of sent to reinforced documents increases more. At the beginning of the darker gray region, the relative shift of the two curves is large, at the end it is small, but becomes large again after that region, when multiplication slows down.

The multiplication of the foragers is shown in Fig. 3(b). Gray levels of this figure represent the value of the foragers, the range goes from 60 (darker) to 200 (lighter). In the time region studied, the values of the foragers have never fallen below 60. Upon bipartition new individuals are separated by horizontal white lines in the figure.

### *B. Compartmentalization*

Division of work is illustrated by Fig. 4. According to Fig. 4(b) large proportion of the sites are visited exclusively by not more than one forager. Only about 40% of the sites is visited by more than one forager. Figure 4(a) demonstrates that new foragers occupy their territories quickly. Figure 4(b) shows that similar data were found for few (2-4) and for many (22) foragers (upper boundary is mainly between 0.6 and 0.7 throughout the figure). The figures depict the contributions of individual foragers: as new foragers start, they quickly find good terrains while the older ones still keep their good territories. The environment changes very quickly, cca. 1200 new URLs were found every day. Time intervals of about 75 minutes were investigated.

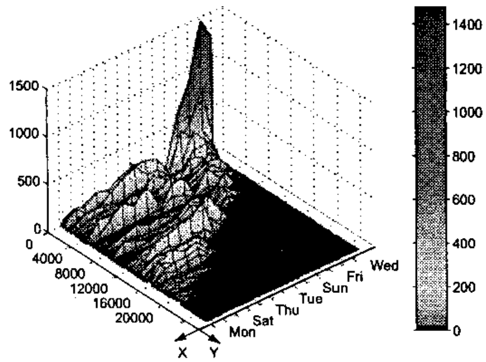
A relatively large number of sites were visited by different foragers. The question is if these foragers were collecting similar or different news. If they had the same 'goal' then they presumably made similar decisions and, in turn, their next step was the same. We have found that the ratio of such 2 step trajectories drops quickly at the beginning of the experiment and it remains between 0.2 and 0.3. Given that the increase of the number of foragers gave rise to the decrease of individual foraging time, the actual numbers are only indicative. Nevertheless, the fast decrease at the beginning and the small ratio for few as well as for many foragers provides support that foragers followed different paths, that is, foragers developed different behaviors. Differences between hunting/foraging territories and/or differences between consumed food are called compartmentalization (sometimes called niche formation) [1], [3], [2], [4]. Figure 4 demonstrates that compartmentalization, is fast and efficient in our algorithm.

The population of news foragers can be viewed as a rapidly self-assembling and adapting breaking news detector. The efficiency and speed of novelty detection is increasing while the structure of the environment is changing very quickly as it is shown in Fig. 5: The number of newly discovered URLs increases about linearly by time and new starting points keep emerging continuously. These drastic changes are followed by the breaking news detector, which continuously reassembles itself and maintains its monitoring efficiency. In fact, the efficiency of the detector increases with time; the quality of the assembly is improving.

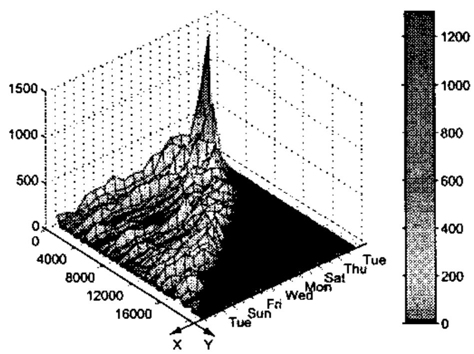
## IV. Discussion

We have introduced an evolutionary Alife community comprising the following main features:

*Feature 1.* Individuals of the community could multiply. Multiplication was kept as simple as possible. It was a bipartition




(a) $1^{st}$ month




(b) $3^{rd}$ month

Fig. 5. **Distribution of new starting points.** Newly discovered URLs were indexed. X axis: time in days, Y axis: index of URLs. Vertical axis: number of *new* starting points summed over equal time domains and equal domains of URL indices. As time goes on, the forager community continuously discovers new regions. Runs from the first (from the third) month of the three month period are shown in subfigure (a) (in subfigure (b))

process that randomly shared the properties of the parent to the descendants.

*Feature 2.* Individuals had two adaptive long-term memory components: (2.a): a discrete component, the list of good foraging places, subject to evaluation and selection and (2.b): a continuous component, the behavior forming value estimating module.

*Feature 3.* No fitness value was available, fitness was implicitly determined by the evolution of the community and by the environment, which was not under our control.

*Feature 4.* There was a central agent, that administered the immediate reward and cost.

Feature 2.a is advantageous in scale free small worlds[1]. In such structures there is a large number of highly clustered regions. In these regions, the escape probability is very small. Intelligent crawlers get easily trapped in these regions. This point has been studied thoroughly in our previous works [8], [9]. In fact, the well known PageRank algorithm makes use of stochastic restarts, too to avoid the oversampling of these regions [20]. In general, feature 2.a is useful only if the forager 'knows' how to get to the foraging place. Here, this was a simple matter of memorizing the http addresses. Feature 2.a is crucial for efficient sharing of space.

Without Feature 2.b, agents could still divide the world but may not develop different behaviors. Feature 2.a and 2.b, together, enable different individuals to develop different function approximators and to behave differently even if they are launched from the same html page. This property has been demonstrated in Fig. 4. That is individuals either live in different regions (i.e., have different weblogs), or behave differently (i.e., have different function approximators), or both. This is that we call *compartmentalization* in the context of our work: each individual has its environmental niche, where the word environment concerns the environment of the Alife community. However, these niches are apparent: those are determined jointly by the indirect competition, by the actual properties of the foragers and by the actual properties of the environment.

Our results indicate that agents with different behaviors have evolutionary advantages in the fast changing environment that we studied. Work sharing in the Alife community is efficient: Foragers found new Internet pages continuously on the vast Internet news domain that we studied. Probably, in an environment, which is changing slowly and where food is hard to come by, the division of space and expertise in searching the owned environments could be the winning strategy. In our model, such strategy is possible. Regions of the Internet, poor in novel information, are currently under study to support this claim.

The population of news foragers can be viewed as a rapidly self-assembling and adapting news detector. The efficiency and speed of novelty detection is increasing. This occurs in spite of the fact that the structure of the environment is changing very quickly: the number of newly discovered URLs was about constant versus time. Such drastic changes are followed by the news detector, which continuously reassembles itself and improves its monitoring efficiency.

Finally we note that the distribution of work *without* direct interaction makes this model a possible starting point to study the emergence of interaction; the emergence of communication in our case. The intriguing point is that (i) communication of all available information is impossible, (ii) communication of relevant information still scales badly if the number of participants increases and if the information is broadcasted, (iii) processing of communicated information consumes time, and all of these can be costly. On the other hand, the communication or the exchange of properly selected information that lowers the time of finding that information are of clear relevance in our distributed computational world.

[1]Note that the distinction between scale free worlds and scale free small worlds is delicate [19]. The Internet is considered scale free small world [6], [7]. Given that fitness is not provided by us, we see no straightforward way to compare our results to algorithms working on artificial grid worlds.

## V. SUMMARY

An Alife algorithm has been introduced having individuals with two component memory. There is a discrete memory component providing a crude 'discretization' of the environment. This list develops together with the long-term cumulated value estimation memory component. The novel combination of these two components makes individuals unique and efficient. Individuals behave differently, they move along different paths. Still, the efficiency of work-sharing is high and increases by time. The algorithm contributes to the understanding of memory components in evolutionary systems and hierarchies. Our results concern scale free small worlds, abundant in mother nature. Most probably, the speed and the efficiency of work sharing is due to the highly clustered scale-free small world structure of the environment, the one that most evolutionary system shares.

The algorithmic model is minimal. Many other algorithmic components, most importantly e.g., direct interaction could be included. Our point is that individuals of the community compete with each other in an *indirect* fashion and thus competition gives rise to work sharing, which looks like collaboration. Such apparent collaboration needs to be separated when the advantages of direct interaction are to be considered. It seems to us that our algorithm is a good starting point to study the advantages and the evolution of interaction by *adding* new features to the agents, which enable the development of such skills.

*Acknowledgments:* This material is based upon work supported by the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory, under Contract No. FA8655-03-1-3036.Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory.

## REFERENCES

[1] C. Clark and M. Mangel, *Dynamic State Variable Models in Ecology: Methods and Applications*. Oxford UK: Oxford University Press, 2000.
[2] J. Fryxell and P. Lundberg, *Individual Behavior and Community Dynamics*. London: Chapman and Hall, 1998.
[3] V. Csányi, *Evolutionary Systems and Society: A General Theory of Life, Mind, and Culture*. Durham, NC: Duke University Press, 1989.
[4] G. Kampis, *Self-modifying Systems in Biology and Cognitive Science: A New Framework for Dynamics, Information and Complexity*. Pergamon, Oxford UK, 1991.
[5] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. Morgan Kaufmann, San Francisco, USA, 2001.
[6] A. Barabási, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: The topology of the world wide web," *Physica A*, vol. 281, pp. 69–77, 2000.
[7] J. Kleinberg and S. Lawrence, "The structure of the web," *Science*, vol. 294, pp. 1849–1850, 2001.
[8] I. Kókai and A. Lőrincz, "Fast adapting value estimation based hybrid architecture for searching the world-wide web," *Applied Soft Computing*, vol. 2, pp. 11–23, 2002.
[9] A. Lőrincz, I. Kókai, and A. Meretei, "Intelligent high-performance crawlers used to reveal topic-specific structure of the WWW," *Int. J. Founds. Comp. Sci.*, vol. 13, pp. 477–495, 2002.
[10] R. Albert and A. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47–91, 2002.
[11] E. Pachepsky, T. Taylor, and S. Jones, "Mutualism promotes diversity and stability in a simple artificial ecosystem." *Artificial Life*, vol. 8, no. 1, pp. 5–24, 2002.
[12] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.
[13] M. J. Mataric, "Reinforcement learning in the multi-robot domain," *Autonomous Robots*, vol. 4, no. 1, pp. 73–83, 1997.
[14] I. Szita and A. Lőrincz, "Kalman filter control embedded into the reinforcement learning framework." *Neural Computation*, 2003, (in press).
[15] W. Schultz, "Multiple reward systems in the brain." *Nature Review of Neuroscience*, vol. 1, pp. 199–207, 200.
[16] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," in *Proceedings of ICML-97, 14th International Conference on Machine Learning*, D. H. Fisher, Ed. Nashville, US: Morgan Kaufmann Publishers, San Francisco, US, 1997, pp. 143–151. [Online]. Available: citeseer.nj.nec.com/joachims96probabilistic.html
[17] D. Boley, "Principal direction division partitioning," *Data Mining and Knowledge Discovery*, vol. 2, pp. 325–244, 1998.
[18] R. Sutton, "Learning to predict by the method of temporal differences," *Machine Learning*, vol. 3, pp. 9–44, 1988.
[19] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of scale-free networks: Error and attack tolerance," *Physica A*, vol. 320, pp. 622–642, 2003.
[20] R. Lempel and S. Moran, "The stochastic approach for link-structure analysis (salsa) and the tkc effect." *Computer Networks*, vol. 33, 2000.