UNIVERSITY OF CALIFORNIA, SAN DIEGO

Life-like agents:
Internalizing local cues for reinforcement learning and evolution

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Computer Science and Cognitive Science

by

Filippo Menczer

Committee in charge:

Professor Richard K. Belew, Chairperson
Professor Garrison W. Cottrell
Professor Russell Impagliazzo
Professor Ted J. Case
Professor John D. Batali
Professor Jeffrey L. Elman

1998

The dissertation of Filippo Menczer is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

1998

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am grateful to my advisor, Rik Belew, and the professors in my doctoral committee (Gary Cottrell, Russell Impagliazzo, John Batali, Jeff Elman, and Ted Case), as well as Christos Papadimitriou, Charles Elkan, Mike Gilpin, Fran Berman, Bill Griswold, Jeanne Ferrante, and Domenico Parisi, for support and guidance through the years.

Many thanks also go to past and present members of the groups and labs in which I have had the fortune to belong over the last ten years: the Cognitive Computer Science Research Group at UCSD (especially Tom Kammeyer, Mark Land, Chris Rosin, Wolfram Willuhn, Amy Steier, Bill Hart, Craig Mautner, and John Hatton); the AI Lab at UCSD (especially Daniel Clouse, Tim Bailey, Dave Demers, Curtis Padgett, Bill Grundy, Matt Dailey, Chris Vogt, and Alvaro Monge); and the Artificial Life Group at the Institute of Psychology of the National Research Council in Rome (especially Federico Cecconi, Stefano Nolfi, Orazio Miglino, Dario Floreano, Luigi Pagliarini, Roberto Pedone, and Henrik Lund). They provided for discussions, suggestions, companionship, and help when I was stuck.

To my parents, Marina and Erico, my gratitude for giving me life and much more. To my wife, Colleen, my love for supporting and coping with me. To the rest of my family and friends, from Anchorage to San Diego, Duluth, and Rome, thank you for being there for me.

Financial support towards the Ph.D. was provided by the Rotary Foundation, the University of Florence, Apple, NATO, and the departments of Linguistics and Computer Science and Engineering at UCSD. External logistic and scientific support was provided by the Fulbright Foundation, the Institute of Psychology of the National Research Council in Rome, the Cognitive Science department at UCSD, and Encyclopaedia Britannica, Inc.

VITA

May 16, 1965	Born, Rome, Italy
1989–1991	Institute of Psychology, National Research Council, Rome
1991	Doctor in Physics, University of Rome “La Sapienza”
1991–1992	Fulbright and Rotary Foundation Scholar
1992–1994	Instructor, University of California, San Diego
1994	M.S., University of California, San Diego
1994–1996	Apple and NATO Fellow
1996	C.Phil., University of California, San Diego
1996–1997	Research Assistant, University of California, San Diego
1998	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

“Adaptive retrieval agents: Internalizing local context and scaling up to the Web.” Technical Report CS98–579, University of California, San Diego, 1998 (submitted to the Machine Learning Journal).

“Adaptive information agents in distributed textual environments.” Second International Conference on Autonomous Agents, 1998.

“Local selection.” Seventh Annual Conference on Evolutionary Programming, 1998.

“Artificial life meets the Web.” First International Conference on Virtual Worlds, 1998.

“ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery.” Fourteenth International Conference on Machine Learning, 1997.

“From complex environments to complex behaviors.” Adaptive Behavior, Vol. 4, No. 3/4, pp. 317-363, Winter-Spring, 1996.

“Internet: Vita da ragni.” Sistemi Intelligenti, Vol. VII, No. 3, pp. 421-442, December 1995.

"Latent Energy Environments." In: Adaptive Individuals in Evolving Populations: Models and Algorithms, edited by R.K. Belew and M. Mitchell, Santa Fe Institute Studies in the Sciences of Complexity, Vol. XXIII, Addison Wesley, 1996.

"Maturation and the evolution of imitative learning in artificial organisms." Adaptive Behavior, Vol. 4, No. 1, pp. 29-50, Summer 1995.

"Evolving sensors in environments of controlled complexity." Artificial Life IV, 1994.

"A neuro-ethological approach for the TSP: Changing metaphors in connectionist models." Journal of Biological Systems, Vol. 2, No. 3, pp. 357-366, 1994.

"Recombination and unsupervised learning: Effects of crossover in the genetic optimization of neural networks." Network - Computation in Neural Systems, Vol. 3, No. 4, pp. 423-442, December 1992.

"Evidence of hyperplanes in the genetic learning of neural networks." Biological Cybernetics, Vol. 66, No. 3, pp. 283-289, January 1992.

"A model for the emergence of sex in evolving networks: Adaptive advantage or random drift?" First European Conference on Artificial Life, 1992.

"Vita artificiale." Sapere, No. 943, pp. 41-51, December 1991.

"Trigger control in a high energy physics experiment." IEEE Transactions on Nuclear Science, Vol. 36, No. 5, pp. 1508-1513, October 1989.

FIELDS OF STUDY

Major Field: Computer Science
    Studies in Artificial Intelligence and Information Systems.
    Professors Richard K. Belew and Garrison W. Cottrell

    Studies in Algorithms.
    Professor Russell Impagliazzo
    Professor Christos Papadimitriou, University of California, Berkeley

Major Field: Cognitive Science
    Studies in Artificial Life.
    Professors Jeffrey L. Elman and John D. Batali
    Professor Domenico Parisi, National Research Council, Rome

    Studies in Ecological Theory.
    Professor Ted J. Case

ABSTRACT OF THE DISSERTATION

Life-like agents:
Internalizing local cues for reinforcement learning and evolution

by

Filippo Menczer
Doctor of Philosophy in Computer Science and Cognitive Science
University of California, San Diego, 1998
Professor Richard K. Belew, Chair

As computing and information environments become more complex, there is a need for tools that assist users in ways that scale with the growth of such environments and adapt to both the personal preferences of the user and the changes in user and environmental conditions. In this thesis I attempt to abstract some core principles from ecological adaptive processes, and cast them into a machine learning framework. The thesis focuses on two mechanisms — *local selection* and *internalization* — by which evolutionary algorithms can be: (i) efficiently parallelized, for adaptation in distributed environments; and (ii) integrated with both unsupervised and supervised learning, for adaptation at different spatial and temporal scales in dynamic and heterogeneous environments.

These methods have been applied in the construction of populations of adaptive information agents (InfoSpiders). Agents search online for information relevant to the user, by traversing links in distributed information environments such as the Web. I demonstrate feasibility of the approach by comparing the performance of InfoSpiders with both exhaustive search and other heuristics, on well-defined hypertext environments. InfoSpiders are capable of capturing essential features of their spatially and temporally local context, thus adapting to complex dynamic environments and user requests. InfoSpiders complement search engines

with respect to several of their difficulties. Engines provide global starting points, based on statistical features of the search space (words); InfoSpiders use topological features (links) to guide their subsequent search.

The same principles have also been applied to build better models of ecological adaptation. I have developed a model and simulator called LEE (Latent Energy Environments), whereby agents adapt by a genetic algorithm based on local selection. This simple individual-based model results in the emergence of density-dependent evolution, and allows the simultaneous study of adaptation at the individual and population level. LEE has been used to analyze the roles of different types of environmental structure in shaping the selective pressures that allow complex collective behaviors to evolve. I also report on the results of other experiments, aimed at observing the interactions between different evolving traits — behaviors, morphology, and maturation — and different forms of individual plasticity — prediction, reinforcement learning, and imitation.

Chapter I

Introduction

I.A Motivation

As computing and information environments become more complex, there is a need for tools that assist users in ways that scale with the growth of such environments and adapt to both the personal preferences of the user and the changes in user and environmental conditions. The situation is not unlike the one faced by ecologies of organisms adapting in natural environments. In fact, attempts to model the evolution of biological organisms have increasingly focused on the roles of the environment, and computational simulations have allowed to make such models more realistic.

I.A.1 The "artificial life bridge"

My research interests lie on a bridge between the natural sciences and engineering. In one direction, I envision a flow of ideas from the way problems are solved in the natural world to algorithmic solutions of complex computational and information challenges. Genetic algorithms and reinforcement learning are two examples of powerful and general machine learning methods derived from processes observed in nature. As our computational environments become more complex and life-like, adaptive techniques will mature and become integrated into the core of computer science.

In the other direction of the bridge, I advocate the use of efficient and rigorous computational tools to test new models and theories about natural systems. We must understand a system before we can build a paradigm of any usefulness around it. If we are to take inspiration from nature as we attempt to tackle new computational challenges, we must be humble in the face of nature's accomplishments.

I have begun to build the pillars of my bridge around some of the questions that are already making the two sides visible to each other. Consider the wide class of problems in which some "agents" must adapt to a complex environment, using only local, "on-line" interactions to determine appropriate strategies. Internet-based applications and robotics, for example, typically deal with environments that can be very large, dynamic, noisy, unreliable, heterogeneous, and/or distributed. The ideal agents would have to capture regularities at multiple spatial and temporal scales, identify those environmental features that best correlate with their performance, and internalize such features into their behaviors based only on local context. Individual-based learning algorithms are limited in their capability to capture global features and cover heterogeneous domains. Genetic algorithms are prone to coarseness of scale and premature convergence. I have extended and combined both approaches into a common algorithmic framework that overcomes these limitations by allowing single agents and populations to integrate environmental signal over different times.

How can we expand on the current adaptive and machine learning techniques to incorporate nature's solutions to problems such as these? Biologists are undertaking problems that seriously challenge the boundaries of mathematical tractability and the adequacy of available analytical tools. What computational wisdom should be offered to the natural sciences as they embark in the study of natural systems of unprecedented complexity?

I.A.2 From nature to technology

One of the exciting challenges brought about by the explosive development of the Internet is the problem of helping users cope with the vast amounts of unorganized, distributed information on the Web. There is a real need for tools that allow users to quickly locate the documents they need. Traditional “information retrieval” methods cannot keep up with the rapid changes of this environment. We need to complement them with new algorithms that scale with the growing number of documents. Further, users need personalized tools that can refine their requests based on context and adapt to their changing, long-standing queries and profiles.

The difficulties posed by building “intelligent” agents for the Web make this environment an obvious candidate for exploring the machine learning issues outlined above. Not only is the environment very large, truly distributed and highly dynamic, but the feature space — hundreds of thousands of words — severely pushes the limits of current learning techniques. Part of this dissertation focuses on the application of adaptive, situated agents to information discovery, retrieval, and filtering in distributed environments such as the Web.

One of the first issues that comes into light is how to model the environment; what are its assets from a learning agent’s perspective? Statistical features such as word frequencies are of course crucial dimensions. I argue that the “linkage topology” structure imposed by information providers upon the organization of documents is another important resource. Even in unstructured information environments, authors tend to cluster documents about related topics by letting them point to each other.

Another issue is how to choose agent representations allowing for the emergence of effective strategies. It is desirable to build agents who can internalize environmental features that appear locally correlated with their fitness, at different spatial and temporal scales. We discuss how to achieve this goal by using a novel evolutionary strategy, in combination with a form of reinforcement learning whereby both the population fitness and each individual's reinforcement are derived from the same environmental signals, sampled over different time scales. Additionally, these signals are seamlessly influenced by the user's relevance feedback (when available). We would like to arm our agents with the capability to internalize these signals as well.

I.A.3 From technology to nature

General simulation tools attempting to model the behaviors of natural populations interacting in real environments often fail in one of two ways. If they are rigorous, they may be too simplistic to capture the complexities of the interactions they aim to model. If they are broad in scope, they may be too convoluted to allow for formal analysis.

To correct this situation, we need models that allow for: flexible individual representation; both population evolution and individual learning; user-defined environmental of quantifiable complexity; and the potential for open-ended behaviors. The thesis describes several steps taken in these directions.

I.B Overview

In this thesis we attempt to abstract some core principles from ecological adaptive processes, and cast them into a machine learning framework. Distributed evolutionary algorithms are a recent paradigm springing from models developed in the artificial life community. One of the defining characteristics of the paradigm, on which this thesis focuses, is the use of *local selection* schemes. The basic mechanism is extremely simple: agents in a distributed population use and collect energy from local interactions with the environment. An agent dies when it runs out of energy, and reproduces when its energy passes some fixed threshold. Energy is a conserved, distributed, finite resource and is shared by the population. This indirect interaction is the main form of communication and competition among agents.

Chapter II introduces and discusses local selection. We characterize its advantages and shortcomings. First, its local flavor provides the natural setting for truly distributed adaptive algorithms. Local selection on one hand frees the evolutionary computation model from the bottleneck of any centralized, global control, allowing for efficient distributed implementations. Second, local selection is better suited for tasks such as multimodal optimization, requiring a cover of the adaptive landscape rather than optimizing convergence. In these cases, local selection is both effective and more efficient than other methods (e.g., fitness sharing) that have been put forth to deal with such problems.

This approach is not without its limitations, of course: local selection has thus far proven unsuccessful for combinatorial optimization problems, due to its minimal selective pressure; it is also infeasible for environments that cannot be "marked" or "queried" as data structures by the agents; and finally, it may be difficult to apply, depending on the isomorphism necessary between the problem and the environmental model required for the algorithmic implementation.

The characteristic of local selection that makes it a most appealing abstraction from a machine learning perspective is that it creates a tight coupling between adaptation at the population and individual levels. Local selection and reinforcement learning can be seen as two adaptive forces feeding on the same environmental signals, integrated at different temporal and spatial scales. This allows local interactions between learning and evolution to be driven by the environment in a natural way. Single agents can *internalize* the features of the environment as they are perceived in the local areas where they are situated, simplifying the problems they face.

These issues are explored in the simple context of abstract graph environments. These are randomly constructed with given statistical properties, so that the mechanisms of local selection and internalizations can be evaluated most easily.

Once we have achieved a better understanding of such abstractions in controlled machine learning experiments, the next goal of the thesis is to apply the principles of local selection and internalization to construct more realistic models of adapting biological entities, without sacrificing tractability. We have developed a model and simulator called LEE (Latent Energy Environments), whereby agents adapt by a steady-state genetic algorithm based on local selection. This simple mechanism alone results in the emergence of density-dependent evolution, capturing the quantitative behavior of this class of population growth models with an individual-based model.

LEE is introduced and described in Chapter III. While striving to preserve tractability, we have enriched LEE with flexible individual representation via user-specified sensors, motors, and neural networks; unsupervised learning by reinforcement or sensory prediction; user-defined, physics-grounded specification of environments of graduated complexity; and open-ended behaviors by user-defined metabolic interactions.

We have used these features to explore several adaptation questions. In one set of experiments, we analyzed the roles of different types of environmental structure in shaping the selective pressures that allow complex collective behaviors to evolve. We showed, for example, that complementary metabolic systems can lead to the preservation of biodiversity, and that seasonal fluctuations can divide individuals into subpopulations with different behaviors exploiting seasonal niches.

In a second set of simulations, we observed the coevolution of behaviors with morphological traits, speculating whether learning could favor the evolution of informative sensors.[1] We found that the signals provided by the environment for reinforcement learning are in fact effective, when the sensory apparatus can internalize the external features whose correlation with fitness has to be discovered by evolution. Finally, we modeled the interactions between maturation and evolution,

---

[1] We assume non-Lamarckian evolution, so that phenotypic changes due to learning are not inherited by offspring.

finding empirical support for the hypothesis that the development of behavioral skills prior to reproductive maturity can contribute to delayed maturation.

The more applied side of this thesis deals with the transfer of artificial life-inspired algorithms from ecological adaptation to distributed computation and information management problems. Chapter IV discusses the application of these methods in the construction of populations of adaptive information agents (InfoSpiders).

Agents search online for information relevant to the user, by traversing links in distributed information environments such as the Web. The idea is to complement search engines with respect to two of their difficulties. (i) Scaling: the static character of the index behind any search engine cannot keep up with the rapid dynamics of the growing and changing Web. (ii) Personalization: the general character of the index building process cannot exploit the differences between the needs of different users. We think that InfoSpiders could start up where search engines stop. Engines provide global starting points, based on statistical features of the search space (words); InfoSpiders can use topological features (links) to guide their subsequent search.

InfoSpiders employ both unsupervised and supervised adaptation. Driven by the user's initial query and interactions with the environment, via local selection and reinforcement learning, the population forages on information perceived as relevant. Agents identify relevant niches and exploit their structure. The evolutionary process, at the population level, lets agents evolve internal representations allowing them to heuristically choose links in an intelligent and autonomous way. The representation forces each agent to focus on a small number of word features that appear locally correlated with relevance. The presence and position of such words in documents aid InfoSpiders in deciding where to go next. Each agent can learn by reinforcement, during its lifetime, how to combine these features most effectively. This task is feasible precisely because only a small number of word features are selected by each agent — those that are most useful where the agent is situated. The user's relevance feedback, if available, provides InfoSpiders with the supervision they need in order to adapt their search behaviors to the (possibly changing) personal preferences of the user.

The results reported in Chapter IV, based on theoretical analysis, simulations, and actual experiments on Web-based corpora, are very encouraging. We first show that linkage topology can indeed be detected and exploited by distributed agents. We then demonstrate feasibility of the approach by comparing the performance of InfoSpiders with both exhaustive search and other heuristics, on a well-defined hypertext environment. The InfoSpiders algorithm outperforms exhaustive search by an order of magnitude, and its distributed nature allows it to beat centralized algorithms as well in resource-bound experiments.

The contributions of the thesis are recapitulated in Chapter V, especially with respect to the large body of related research in the different fields that this thesis touches on. Several directions in which this research program can be continued are also discussed.

This thesis reflects an interdisciplinary array of research interests, and therefore touches on such diverse areas as machine learning, genetic algorithms, artificial life, theoretical ecology, and information retrieval. Here is a map of the main general subjects concerning subsequent chapters.

Chapter II Evolutionary and genetic algorithms, reinforcement learning, neural networks, graph-theoretic algorithms, combinatorial optimization

Chapter III Artificial life, evolution, learning, life histories, ecology

Chapter IV Information retrieval, machine learning, multi-agent architectures, World Wide Web, distributed systems

Chapter V All of the above.

Chapter II

Theory

Environmental features play central roles in the adaptation of agents *situated* in complex environments. Whether we are interested in studying the behaviors of (natural or engineered) agents or in devising algorithms through which agents can perform tasks autonomously and efficiently, we cannot make much progress without considering the context in which such behaviors and algorithms are evaluated.

As our computation and information environments become more complex, there is a need for tools that assist users in ways that scale with the growth of such environments and adapt to both the personal characteristics of the user and the changes in user and environmental conditions. The situation is not unlike the one faced by ecologies of organisms adapting in natural environments. Such natural agents have to adapt to the topology of their environments, internalizing into their behaviors (via evolution or learning during life) those environmental signals and cues that they perceive as leading to their well-being. The environmental features that are best correlated (or anti-correlated) with fitness are the most useful internalizations.

Consider for example an ant's environment: its association between pheromone and food is internalized into the ant's pheromone-following behavior. Further, this behavior can be implemented by any ant without need for centralized control; finally, the ant can still resort to a behavior, such as a random walk, that can be implemented in the absence of any appropriate environmental signals. These capabilities of an ant colony — internalization of environmental signals, distributed control, and integration of externally driven and endogenous behaviors — are also highly desirable properties for agents carrying out tasks in complex artificial environments.

Much of the machine learning research of the recent years has in fact focused on ways to capture environmental features — e.g., spatial and/or temporal regularities such as gradient information and statistical correlations — and internalize them into adaptive algorithms. This chapter focuses on a number of machine learning abstractions inspired by ecological models, and characterizes them with respect to efficiency, feasibility, and effectiveness in general application classes. The following chapters will evaluate the ideas introduced here in more specific modeling and application domains.

II.A Background: Machine learning

Consider an agent situated in a complex environments and performing some task. Typically, the agent will have access to some sensory information about the environment. This input can be elaborated, and combined with state information accumulated in the past, to compute an output. The agent will typically have some motor organs to convert its outputs into motor actions. Changes in the environment, together with the results of the agent's actions, will produce a dynamic stream of inputs and outputs. If we are *observing* an agent to study how it functions and what task it performs, we can characterize this process — the way in which the agent interacts with the environment — as the *behavior* of the agent. If we are instead *constructing* an agent in order to perform some specified task or solve some problem, then we can characterize these interactions as a *strategy* or *algorithm* implemented by the agent.

Most realistic agents (animals, robots, and software agents) have limited access to environmental information; this is determined by the location of the agent and by some proximity metric. For a robot, physical two-dimensional space and Euclidean distance normally determine proximity. For a bird, three-dimensional space is more appropriate. For an electronic mail agent, the environment is best described by a computer network. Irrespective of whether we have a descriptive or a normative agenda, interactions between agents and environments are at the very core of understanding how agents can perform their tasks.

II.A.1 Evolutionary algorithms

Evolutionary algorithms (EAs) are the general class of algorithms inspired by the evolutionary paradigm of adaptation by survival of the fittest, and repeated selection and reproduction. An initial population of candidate individuals, each representing a solution of the problem at hand, or a strategy to perform the task at hand, is generated randomly. Then the population is repeatedly subjected to a cycle in which: (i) individuals are evaluated with respect to their fitness in solving the problem or performing the task; (ii) the best individuals are selected for reproduction; (iii) clones of the selected individuals are created and then subjected to some modification operators; and finally, (iv) a new generation is formed with the newly created individuals. The algorithm is halted when some stopping condition is met, for example when the population reaches *convergence,* i.e., it becomes too homogeneous to warrant any additional progress.

The are numerous variants of this basic algorithm. For example, many problem-independent or problem-specific operators can be applied to offspring to improve on the process by maintaining diversity, combining information from different candidate solutions, or performing local search to locate the nearest local optimum solution. Different representations for candidate solutions can be used, such as bit strings, real-valued vectors, or rule-based systems.

Evolutionary algorithms employing binary representations and crossover operators to recombine individuals and form solutions from sub-solutions are called genetic algorithms (GAs). The theory behind GAs rests on the so-called *schemata theorem* [67], showing that for binary representations the algorithms exploits the regularities of the search space in a more or less optimal way. A *schema* is a set of solutions obtained by replacing bits in a solution string by wild-card characters. The crossover operator recombines solutions by preserving short, above-average schemata. The schemata theorem depends on the validity of the *building blocks hypothesis* — the idea that partial solutions can be combined to produce more and more complete solutions. When this hypothesis holds, selection and crossover are said to allocate individuals in the population to schemata in a near-optimal way by implicitly testing an exponential number of schemata in parallel. For non-binary encodings the role of crossover is less clear, although the building blocks hypothesis has been extended to real-valued representations [125]. A detailed description of evolutionary and genetic algorithms is out of the scope of this thesis, and interested readers are referred to an extensive literature [68, 53, 132].

Selection and locality

Selection and recombination are the major sources of interactions among individuals in EAs. These two mechanisms can be seen as ways to implement competition and cooperation among solutions, respectively. They are not independent of each other, and each can play an important role in optimizing the fitness function, by exploiting the information in the population and exploring promising alternatives. In this thesis, however, we will mainly focus on *selection* alone.[1] In so doing, we will miss the discussion of the possibly crucial interactions between selection and recombination, and of the potential contributions of crossover when the building blocks hypothesis holds. However, this choice will simplify the discussion by considering the issue of communication due to selection in isolation of other sources of interaction. While neglecting crossover may impose

---

[1] Crossover will be applied in problems where the representation makes it obviously feasible.

a serious limitation on the performance of EAs (e.g., in combinatorial optimization problems), this issue has received extensive attention elsewhere [58, 86].

A selection scheme is a map from a distribution of fitness values to a distribution of offspring numbers, over the population.[2] There are many selection schemes that have been adopted in EAs. These can be deterministic or stochastic, threshold, linear or nonlinear functions, based on rank or actual fitness values, or more complicated mechanisms. In general, however, selection maps share the feature of being computed in a centralized, *global* manner. For example, if population size is to be kept constant, then the offspring distribution must be scaled by a normalization factor that takes into account the whole population; or if rank is used, then all individuals must be evaluated to assess their relative position. There are selection schemes that do not require the whole population to be evaluated to determine the number of offspring of a single individual. For example, in *tournament selection,* an individual need only be compared with $k-1$ other individuals, where k is the size of the tournament ($k \geq 2$). However, an individual must be compared with at least one other.

Let us refer to the *locality* of a selection scheme as the lack of dependencies in the computation of the selection map. The locality of selection is a topic of increasing interest in the EA/GA community [54]. One set of reasons has to do with the problem of premature convergence, and ways to prevent it in the presence of crossover by maintaining diversity and/or restricting mating [43].

Parallelization

The convergence issue is related with the general concern for the parallelization potential of GAs, and performance implications [56, 37]. Local selection in parallel GAs usually stems from imposing geographic constraints on genetic search. This may be a useful way of assisting the formation of diverse subpopu-

---

[2] Here we use “fitness” in the sense on EAs; in biology, fitness is commonly defined as the number of offspring per individual.

lations [58, 33]. The motivation however is in avoiding the communication overhead imposed by global selection schemes; different processors are allocated to subpopulations and we want to minimize inter-process dependencies to improve performance.

The implications of local and global selection on convergence in massively parallel GAs have been studied extensively. The poor match between parallel implementations and the standard GA notion of optimization by convergence is noted for example by McInerney [110], who distinguishes between *convergence* — all individuals converging on the best solution — and *cover* — all good solutions being represented in the population — as measures of successful termination. Parallel GAs are more amenable to cover optimization than to standard convergence criteria, due to the limited communication inherent in most parallel implementations. This limited communication normally implies that there are isolated subpopulations. Applying a parallel GA to optimize a two-mode fitness function, the traditional selection schemes — truncation, linear rank, and proportional selection — cause the population to rapidly converge to one mode of the fitness function or the other, while localized selection strategies generate two separate populations that have converged, each to a separate peak [110].

Niching

Independently of parallel implementations, the problems of ill-convergence exhibited by global selection schemes for multimodal fitness functions is a general issue related to aspects of natural adaptation — niching and speciation — that have seldom been included in formal treatments of GAs. According to Goldberg, we may view a niche intuitively as an organism's job or role in an environment, and we can think of a species as a class of organisms with common characteristics. He writes:

> As reproduction, crossover, and mutation proceed, the population climbs the [fitness] hills, ultimately distributing most of the strings near the top of one hill [...] This ultimate convergence on one peak

> or another without differential advantage is caused by genetic drift — stochastic errors in sampling caused by small population sizes. Somehow we would like to reduce the effect of these errors and enable stable subpopulations to form around each peak.
>
> We also might like to modify the performance of simple genetic algorithms in multimodal problems where the peaks are not all of the same magnitude [...] Perhaps we would even like to allocate subpopulations to peaks in *proportion* to their magnitude [...][3]

Goldberg also points out that, although there is a well-developed biological literature in both niching and speciation [65], its transfer to the artificial genetic search has been limited. Standard GAs are ineffective for multi-niche or multimodal function optimization, due to high selective pressure and premature convergence [44].

Fitness sharing

Several methods have been devised to deal with this problem. The most notable are *crowding* [36] and *fitness sharing* [55]. In both of these methods, fitness is somehow scaled by some measure of similarity among individuals. Shortcomings of both methods are problem-dependency and inefficiency; if p is the population size, selection requires time $O(p)$ rather than $O(1)$ *per individual*. While the time is still polynomial, the slowdown becomes important for practical cases with large populations. Moreover, it turns out that the population size required to maintain the population across niches is estimated to grow super-linearly (with a large constant) with the number of niches (assuming this is known a priori) [107]. Furthermore, computing similarity imposes a large communication overhead for parallel implementations.

All methods for multimodal optimization imply looking at fitness as a finite resource to be shared by the population. A typical application of these methods is game playing, where we attempt to evolve strategies that defeat many

---

[3] From Goldberg [53], pages 185–197.

opponents, rather than just the best opponent. In this domain, since fitness comes from competitions between players, the shared resource can be represented by a population of test cases (opponents) coevolving with the population of solutions (strategies) [146]. This is a special case of local selection, because a solution is tested against the resource rather than the rest of the (solution) population.

Dealing with real problems often leads to modifications of the selection mechanism aiming at maintaining diversity. One example is to tune the selective pressure by adaptively scaling the fitness function in roulette-wheel selection [127]. More generally, developmental processes such as maturation and learning can interact with fitness evaluation to adapt the selection process to the local characteristics of the function being optimized [64]. The role of selection for genetic multimodal optimization remains an active area of research in the GA community [60, 108].

Steady-state

An aspect of EAs that is indirectly related to locality is the *generation gap.* If all the individuals in the population are evaluated together, and the previous population is completely replaced with a new one, we say that the GA proceeds in *lock-step* generations. If instead a subset of the population is replaced at a time, then generations are interleaved through time and the generation gap is reduced. In the limit of minimal generation gap, one individual is evaluated and replaced at a time; in this case we have a so-called *steady-state* EA. Decreasing the generation gap has been shown to preserve diversity within the population and slow down convergence [38].

The relation between generation gaps and locality is that a steady-state algorithm with local selection can be executed in a distributed fashion, with individuals being evaluated asynchronously. Thus steady-state and local selection are both necessary conditions for distributed implementations of evolutionary algorithms. One important consequence of this distributed model is that the population size does not remain constant, and extinction is possible.

We can emulate a distributed EA by sequential calls to the individuals. The order of these calls can be randomized so as to minimize the spurious interferences of the sequential emulation [71]. However, the reverse is not possible; in a distributed system, where each individual may be evaluated on a different machine and there is no centralized control or synchronization, an individual must remain independent of the rest of the population.

II.A.2 Endogenous fitness

Localizing selection implies that the evaluation of an individual fitness becomes an *endogenous* process. Lacking a central, *exogenous* control mechanism to normalize fitness and maintain a constant population size, an individual may internally generate decisions as to its fitness and reproductive value. Much interest has been devoted in the recent years to the *endogenous fitness* paradigm, especially in the artificial life community [2, 175, 22, 13]. The main motivation is to model open-ended evolution. Mitchell and Forrest [133] point out that, although it is relatively easy to implement endogenous fitness strategies, there is virtually no theory describing the behavior of genetic algorithms under these circumstances. The lack of an explicit, exogenous fitness function allows adaptation to emerge naturally from interactions between organisms and their environment (which includes other organisms). It is the complexity of the environment that creates the selective pressures shaping the adaptive landscape.

Both the terms open-ended and endogenous fitness capture an important aspect of the distinctive features that discriminate these models from classic GAs: namely, the fact that the fitness function is not explicitly defined in terms of behavior. The stress therefore is on the creative power of endogenous fitness models, which is in fact one of the main motivating factors for the artificial life model described in Chapter III.

Another aspect of endogenous fitness, even more central to this thesis, is that it can be computed locally. From an engineering point of view, this allows for distributed implementations as described in the previous section. From a modeling point of view, this results in a stronger role of the environment in guiding the evolutionary process. With endogenous fitness, selective pressure results indirectly from the local competition for the finite resources in the shared environment. Ecologists study just such situations in natural systems. If the individuals interact exclusively by sharing resources, they say that fitness is *density dependent* [150]. The lack of a constant selective pressure allows individuals to explore the adaptive landscape without competing with others who are not part of their own local environment.

II.A.3 Reinforcement learning

Most learning schemes proposed for agents who must obtain all their input by interacting with the environment are based on some form of *reinforcement.* Reinforcement is a weak, unsupervised learning paradigm and thus appropriate for autonomous, situated agents that can use environmental cues as reinforcement signals [79]. Reinforcement learning algorithms are generally based on a simple idea: repeat actions that result in positive reinforcement, and inhibit those that cause negative reinforcement. When actions are motor responses to sensory input about the state of the environment, reinforcement learning is called *associative* [8].

Different reinforcement learning techniques have been proposed that focus on different aspects of the learning problem: temporal difference [172], prediction [138], and the successor representation [34] deal with temporal characteristics of the sequence of environmental input states; evolutionary reinforcement learning [1, 11] considers the case in which the reinforcement signal itself has to be evolved by the adapting agents; Q-learning [183] and the adaptive heuristic critic [9] attempt to build models of the environment while learning the task [96, 140].

Because of the active role played by the environment, reinforcement learning is particularly suitable as a model of individual adaptation, as compared to species-level adaptation, when we look at interactions between learning and evolution in an adaptive population (see [97] for a survey of simulation work on the subject). Evolving agents are better adapted to non-stationary environments when they are allowed to learn by reinforcement [98]. Individuals are also able to pick up local environmental cues at smaller time scale than populations [2].

Reinforcement is also one way to model the *internalization* of environmental information into an adaptive agent. The reinforcement signal shapes the adaptive response of the agent. Internalization can occur in any adaptive process, be it based on an individual or a population. A dynamic correlation may exist between performance (or fitness) and local features of the environment, signaled by the sensory system of the agent. Locality may be spatial or temporal: since a situated agent has bounded velocity and size, the concepts of space and time locality are in fact dual.

In evolutionary computation, we can think of fitness as a reinforcement signal for a population representing a distribution of solutions. Therefore the idea of internalization is found in extensive GA literature as well, pointing to ways in which information can be transferred from environmental cues to search strategies. E.g., the topology and smoothness of the data space determine appropriate search parameters and operators [23, 125]. Fitness autocorrelation can also be used as a measure of the difficulty of the task [77, 103, 46]. An analogous idea is that of *self-learning* or *second-level learning* of strategy parameters, as a powerful mechanism of internal adaptation of the algorithm with respect to the objective function topology, in the framework of Evolution Strategies [161]. While the real world is complicated and somewhat unpredictable, natural environments also exhibit a great deal of structure that a properly designed agent can depend upon and even actively exploit [70].

II.B Local selection

The mechanism that we associate in this thesis with local selection (LS) and endogenous fitness is extremely simple: agents in a (possibly distributed) population use and collect energy from local interactions with the environment. An agent dies when it runs out of energy, and reproduces when its energy passes some energy threshold.

The algorithm in Figure II.1 serves to illustrate the basic LS idea. The framework is basically that of a steady-state EA. Note (step 3) that actions result in energetic benefits only inasmuch as the environment has sufficient energetic resources; if these are depleted, no benefits are available until the environmental resources are replenished. Energetic costs are incurred by any action. Costs and replenishment determine the *carrying capacity* of the environment that in turn determines population size. An agent reproduces or dies based on the comparison between its current energy level and the θ, ω thresholds (step 5). Energy intake is the currency by which we measure the success or failure of behaviors with respect to the environment. The only form of communication among agents is their shared use of the finite energy resources.

In a standard EA implementation based on the pseudocode of Figure II.1, the input and output steps correspond to the evaluation of a candidate solution. They may also include a local search phase, if warranted by the problem/algorithm. The net energy intake is the individual's fitness. The reproduction step of course comprises cloning, random mutations, and optionally other genetic operators such as crossover and/or problem-specific local search. In a non-distributed task/algorithm, the environment may reduce to a single global process with the trivial role of ensuring balanced allocation of computational resources to the individuals.

In a multimodal or distributed task, the environment models the problem space and the resources that are locally available to individual solutions. In

```
initialize population of agents, each with energy $E_0$
while there are alive agents
  for each agent $i$
    1. input: sense environment
    2. output: compute action $a$
    3. update energy:
       $E_{envt} \leftarrow E_{envt} - benefit(a)$
       $E_i \leftarrow E_i + benefit(a) - cost(a)$
    4. optional learning
    5. selection:
       if ($E_i > \theta$)
         reproduce(i)
         $E_{offspring} \leftarrow \frac{E_i}{2}$
         $E_i \leftarrow \frac{E_i}{2}$
       else if ($E_i \leq \omega$)
         die(i)
       end
  end
  $E_{envt} \leftarrow E_{envt} + E_{replenish}$
end
```

Figure II.1: Pseudocode of EA with local selection.

such cases the distinction between local and global interactions among individuals becomes important; the selection mechanism and environmental resource model capture the nature of such interactions. In a standard EA, an individual is selected for reproduction based on how its fitness compares with the rest of the population. For example, proportional selection can be modeled by a choice of thresholds $\theta = \omega = \langle E \rangle$, where $\langle \cdot \rangle$ indicates population average; the reproduction rate would then be proportional to how rapidly an agent accumulates energy, with respect to the rest of the population. Likewise, binary tournament selection can be modeled by $\theta = \omega = E_r$ where the subscript r indicates a randomly picked individual.

Let us define LS as a scheme in which θ and ω are independent of the rest of the population. In the rest of the thesis we will use $\theta = 2E_0 = const > 0$ and $\omega = 0$. This way, interference among individuals is reduced to the sharing of local resources — the amount of interaction allowed by the distributedness or multimodality of the task at hand. Energy is consumed by action costs, created by

	Global Selection	**Local Selection**
reproduction threshold	$\theta = f(E_1, \ldots, E_{pop})$	$\theta = const$
fitness normalization	relative (ranked population)	absolute (shared environment)
conserved quantity	selective pressure	entropy
search bias	exploitation	exploration
convergence	unimodal	multimodal
adaptive criterion	optimization	cover
solutions	best (fragile)	good (robust)
biological equivalent	r-selection	K-selection

Table II.1: Schematic comparison between the local and global selection schemes.

environmental replenishment, and conserved at reproduction (parents share energy with offspring) and death (agents die as soon as they run out of energy). Table II.1 illustrates schematically the main implications of the different selection schemes. Some of the entries in the table will be justified in later chapters.

The removal of selection's centralized bottleneck allows for parallel EA implementations. It is also evident that LS is an implicitly niched scheme and therefore it naturally enforces the maintenance of population diversity. These factors in our opinion make LS a central issue in today's EA community [37, 108]. It is then essential to characterize the problem domains in which LS is applicable and/or advantageous, and estimate the costs involved. In section II.E we discuss two problem domains, one where LS proves advantageous over global selection schemes and one where it does not. We conclude the chapter by discussing the consequences of these experimental observations for a fair evaluation of LS methods.

II.C Internalization

A crucial feature of local selection that can act both to its advantage and detriment, is its weak selection pressure. Consider the energy level throughout an agent's lifetime depicted in Figure II.2. The agent reproduces around time 35,

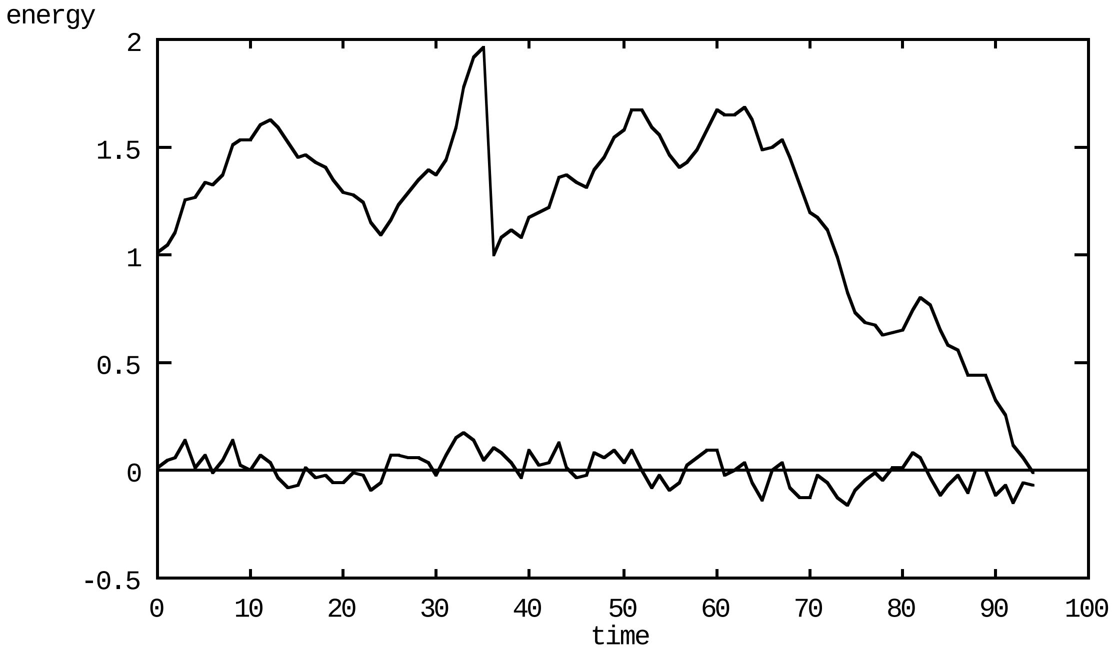


Figure II.2: Typical energy dynamics of an agent during its lifetime. The higher curve plots the level of accumulated energy as a function of time, resulting from the instantaneous changes in energy plotted by the lower curve. The selection threshold is $\theta = 2$. With the exception of the reproduction event, the energy level is the integral of the lower curve.

giving half of its energy to the offspring. Finally the agent runs out of energy and dies shortly after time 90. Such selection events are rare. As long as the energy level fluctuates between 0 and θ, there is no selective pressure.

While an agent interacts with the environment, its sensors may pick up many signals from the environment, Some of these may be pure noise, others may contain information correlated with the performance of the agent. If the agent has access to reinforcement signals from the environment that somehow assess the appropriateness of its actions, then such signals can be used as reward or penalty signals to adjust the agent's behavior during its life. This is the basis of the reinforcement learning framework [140].

Agents performing in real, complex environments have access to many signals. The problem is not so much whether some of these signals contain useful information, but rather how to identify those among the many cues that best correlate with performance and may allow the agent to discriminate between good and bad behaviors. We will discuss in the following chapters how both natural agents in physical environments and software agents in networked information environments can deal with this problem. For now assume that an agent has identified, or learned to compute, a "good" signal, positively correlated with its performance. Imagine this is the energy change computed in step (3) of the algorithm in Figure II.1.

Such instantaneous energy change corresponds to the time derivative of the agent's energy level, and is also plotted in Figure II.2. This signal is continuously available to the agent, and any reinforcement learning scheme can be used to adjust the behavior of the agent so that actions perceived to lead to rewards are reinforced, and action perceived to lead to penalties are discouraged. One such simple reinforcement learning policy, using the sign of the energy change as a reward/penalty signal, would be as follows:

$$\begin{array}{ll} \text{if } benefit(a) - cost(a) > 0 \text{ then} & \text{increase } prob(a) \\ \text{else} & \text{decrease } prob(a). \end{array}$$

With this type of individual learning scheme implemented in step (4) of the algorithm in Figure II.1, we imagine that the consistent rewards between time 0 and 12 in Figure II.2 might have warranted changes leading to an earlier reproduction; likewise, the prevailing penalties incurred between time 70 and 90 might have advised changes leading to a delayed death.

Others have shown that evolving agents can build their own reinforcement signal and then use it to become better adapted to their environment than possible with evolution or learning alone [2, 11]. We consider these phenomena as examples of internalization rather then second-level learning (cf. Section II.A.3) because the information being internalized is not about the learning algorithm, but about the environment in which the task is to be learned.

It is normally difficult for GAs to select for features that do not have a direct effect on fitness. Second-order interactions with fitness, such as those involved in second-level learning, are often overcome by genetic drift [126]. The above example of internalization, as well as other cases that will be illustrated in this thesis, occur with evolutionary algorithms that employ some form of endogenous or local fitness. This is no coincidence: since selective pressure is weaker with LS than with global selection, environmental signals whose correlation with fitness is relatively weak can be detected before being obscured by random drift, so that they can be internalized by the adaptive process. We therefore hypothesize that *local selection is a necessary condition for an evolving population of situated agents to internalize local features of their dynamic environment into their adaptive behavior.* Evidence in support of this hypothesis will be given throughout the remainder of the thesis.

Internalization is related to other aspects of learning that have been studied extensively. Suppose an agent has to learn to produce good responses to environmental inputs; suppose further that the environment is dynamic, noisy, unreliable, inconsistent, or some combination of these. If the agent could "observe" the environment from the outside, studying the most appropriate actions in any situation and analyzing the long-term consequences of each action, while remaining shielded from any risk, then it could wait until it has accumulated sufficient confidence. Then and only then, it would use its knowledge, or environmental model, to construct a policy or behavior.

However, such ideal conditions rarely exist in realistic environments. Often the agent will not be able to wait and test actions without paying their potential consequences. For example, if an action is lethal, the agent cannot learn this without losing the opportunity to put such knowledge to use! The situation in which an agent must learn by trying, and immediately apply its experience to shape its policy, is called *on-line learning.*

Another limitation for realistic agents is that they may not be able to freely choose the environmental conditions they want to explore and test; an action will typically affect the environmental inputs that the agent will experience following that action. This situation is called *active learning* and is related to the *exploitation/exploration* dilemma. The latter refers to the problem faced by agents who have to continuously decide between performing actions that are believed to give good reinforcement, and actions that are not well understood in order to gain information about their value.

Finally, an agent may not see the immediate consequences of its actions. In such cases, referred to as *delayed reinforcement,* it is difficult to assess how past decisions affect and lead to the current situation. Further complication arises when temporal dependencies make it possible for *combinations* of actions to affect the future.

It easy to see that all these learning problems — on-line learning, active learning, and learning from delayed reinforcement — may add significantly to the difficulty of the task faced by adaptive agents in complex environments. In the presence of many environmental signals detectable by the agent's sensory inputs, such problems make it even more crucial for an agent to discriminate among the signals and detect those that represent the best cues.

In summary, simple reinforcement learning algorithms can be effective when the agent can easily identify the reinforcement learning signal. In many complex learning environments, agents do not have such luxury. The problem then becomes to learn to identify useful environmental cues and internalize them into the learning algorithm as reinforcement signals. Which signals represent the best cues may depend upon spatial and temporal locality, so that the agent must continuously adapt not only its policy, but its choice of internalized signals as well. Internalization is made both more complicated and more necessary when agents must learn on-line, actively, and/or from delayed reinforcement.

II.D Duality of reinforcement learning and evolution

An effective internalization of environmental signals in conjunction with reinforcement learning during the lifetime of an agent can obviate the problem of weak local selective pressure. More generally, it can allow agents to pick up cues that are just not detectable at the time scales of evolutionary adaptation. Conversely, local selection is based on the same reinforcement signals used for individual learning, but these are integrated over longer times so that short-term and short-range fluctuations are averaged out, producing agent behaviors that are more robust in the face of noisy, inconsistent, or unreliable cues.

Evolution by local selection and learning by internalization of reinforcement signals are thus two faces of the same coin. Each process uses information from the environment to improve on its "unit of adaptation." For reinforcement learning, the unit is the agent and the signals are sampled instantaneously; for evolution, the unit is the population and the signals are integrated over time. One might view local selection as the extension of reinforcement learning from individuals to populations, or as *reinforcement evolution.* The two forms of unsupervised adaptation can be combined together to cover the different temporal and spatial scales at which useful cues can be detected.

We are interested in evolution and learning as paradigms of adaptation. These two processes are not easy to distinguish in general, and in fact interactions between the two are at the center of abundant debate (see, e.g., [17]) and of this thesis as well. Locality is one dimension along which we can discriminate between these two processes.

Evolution is "global" both in space and in time. A population of individuals samples a possibly large portion of the environment and collects fitness data accordingly. Evolution can work in the absence of local information in the fitness surface and thus can be used as a "method of last resort." But the adaptive value of an individual's behavior is unimportant as a contribution to genetic adaptation, unless its selective advantage is such as to grant a large enough differential reproductive success to affect the genetic pool of the population. If a behavior's usefulness is limited to a local portion of the environment, evolution can hardwire such behavior into the genotype only at the cost of producing behaviors that are maladaptive in other parts of the environment.

Along the time axis, the age of mature individuals is a lower bound for the time scale of evolutionary changes. This is how frequently, on average, selected individuals carry information across generations. Tracking faster environmental changes at the genetic level is only possible for species with accordingly short developmental periods. The consequences are bounds on the complexity of mature individuals, giving rise to limitations of adaptability to complex environments.

A single individual has access to local information about the environment that is not available at the population level. The capability to become adapted to such local features through individual plasticity, or learning, complements the limitations of evolution outlined above.[4] Learning can increase an individual's survival chances by adjusting behaviors and strategies in a manner that may not be appropriate elsewhere in the environment, or at other times. Spatial correlations between fitness and behaviors may be too short-ranged to be detectable at the coarse grain of populations. Individuals are capable of sampling regularities of the environment that disappear at the scale of the average distance between individuals, but are crucial to the survival of the individual due to its limited speed.

Temporal locality is just as important: individuals may be capable of adjusting to environmental conditions characteristic of the particular time (hour, season, age) in which they are constrained by their limited lifetime. Remembering

---

[4] If such phenotypic changes are not to be transmitted to offspring, they can be accomplished without the complications involved by reverse transcription into the genotype. The issue of Lamarckian versus non-Lamarckian inheritance, although extremely intriguing, is out of the scope of this thesis. For a discussion see [62].

recent history can assist adaptation to a changing environment in a way impossible from observations at the longer time scale of generations [190].

Locality is defined above with respect to the dynamic interactions between agents and environment. Therefore the important space and time scales are defined by the many constraints of these interactions: limited lifespan, limited velocity, limited sensory discrimination, size of environmental patches, auto- and cross-correlation distances of important environmental signals, etc. We should point out there are other dimensions in which we can define locality as well. An important one is genetic space; genetic and physical spaces constrain each other in the adaptive process, through the mediation of sensory systems and behaviors [84]. Niche size and resource abundance are crucial in determining carrying capacity and thus population dynamics, which in turn affect genetic diversity and adaptability. Conversely, the genetic pool determines the availability of mutations which may affect the capability to move and thus alter spatial relationships, for example changing the effective size of the environment or making alternative resources available.

As mentioned above, one of the reasons for looking at local selection as a condition for evolutionary internalization is the observation that local selection is actually the natural extension of adaptation by reinforcement learning from the level of individuals to that of populations. As reinforcement learning allows an individual to adapt to its environment within its lifetime, and internalize those regularities of the environment that are both observable and useful at that time scale, so local selection allows a population to do the same on evolutionary time scales. And as reproduction (or death) may be too delayed a reinforcement signal for behaviors whose adaptive value is of importance within a lifetime, so short-term effects of interactions with the environment may be too volatile to determine the evolutionary fitness of a population.

In this view, evolution and learning become two well-integrated parts of the same adaptive process in models where both population and individuals adapt, and we refer to them as reinforcement learning and reinforcement evolution. Evolution takes the fitness measures used for selection without imposing a global optimum; only individuals sharing common local resources compete against each other, so that evolution preserves the usefulness of what is learned by local interactions with the environment. In turn, learning can reinforce behaviors that help an individual to compete with those who share its local resources. Both evolution and learning are thus mediated by the environment, their effects being coupled as a result. The integration of evolutionary algorithms based on local selection with reinforcement learning algorithms based on internalization of environmental cues seems appropriate for problem domains in which performance is measured with respect to the interactions between situated agents and their complex environments. We attempt to characterize such a class of problems in the next section.

II.E Problem domains

What are the features of a problem that make it amenable to local selection algorithms? What conditions must be met in order for local selection algorithms to be applicable? Addressing these questions is the goal of this section.

LS algorithms can be used whenever the fitness function is evaluated by an external environment, in the sense that the environment provides appropriate data structures for maintaining shared resources correlated with fitness. Consider, for example, evaluating a robot in a physical environment: the environment itself holds information about its state. The robot prompts for some of this information through its sensors; storing such information would be by far less efficient than simply prompting for the same information again as needed. If the robot holds some finite memory, it can choose which important information to store — e.g., most recent events. But for an agent, or even a distributed population of agents, it would be simply impossible to store *all* relevant observations about a distributed, dynamic environment.

The environment therefore can take the role of a data structure, to be queried inexpensively for current environmental state. Of course, this is not the case for any adaptation problem. For example, classic applications of GAs such as function optimization have an obviously defined exogenous fitness function that is static and may be expensive to evaluate. No environment can do the job: the shape of the solution space must be determined by the state of the agent or population of agents. Such information is not needed for solving the problem in different conditions (positions or times); rather, it only matters inasmuch as it helps to determine the one global optimum. Local selection would actually hinder this process.

When the problem makes the adoption of LS schemes suitable for situated agents, however, multiple benefits may follow. In the framework of evolutionary computation, the case of distributed populations of agents situated in an environment corresponds to multimodal fitness. As we have seen in Section II.A.1, existing GA approaches to multimodal optimization, such as fitness sharing, are inefficient; the time complexity of fitness evaluation and selection is quadratic in the population size, and lower bounds on population sizes for preventing convergence are super-linear in the complexity of the fitness function. With local selection, instead, the population size self-adjusts to the complexity of the adaptive landscape without any prior knowledge of fitness characteristics; evaluation and selection can be carried out in time linear in the population size; and additional speedup can be obtained by parallel/distributed implementations.

II.E.1 Graph search

Let us first outline a number of experiments pointing to the feasibility and performance of local selection and internalization for an abstracted class of graph environments. The problem can be broadly described as searching large graphs in sublinear time. Imagine a very large graph, where each node is associated with some payoff. The population of agents visits the graph as agents traverse its edges. The goal is to maximize the collective payoff of visited nodes, given that there is only time to visit a fraction of the nodes in the graph. The framework is well studied theoretically in the case of random-walkers. The problem of finding the optimal path to an unknown node in a weighted graph is NP-complete [82]. Agents can do no better than heuristically searching on-line through the graph.

The problem is interesting because typically the graph is distributed, so that agents are charged costs for using its resources, e.g., traversing edges and evaluating nodes' payoff. The issue of distributed algorithms is therefore central for this problem class. Furthermore, the graph search task is general enough that it can be reduced to several interesting special cases. If we use nodes to model hypertext documents, edges for hyperlinks, and payoff for some measure of relevance, then the problem is that of networked information retrieval; we can explore different search strategies in simulated information environments, given a model of relevance [179]. We will in fact return to this characterization of the problem in Chapter IV. Alternatively, the graph could be used to model a 2-dimensional environment in which agents have to sense their position and move to reach some goal. This would be a typical task for situated robots.

In our instances of the graph search task, each node is assigned a payoff p from a uniform probability distribution in the unit interval. Furthermore, each link l is annotated with a "feature vector" with N_f real components $f_1^l, \ldots, f_{N_f}^l \in [0,1]$. The idea is that these features, if properly interpreted, can guide agents by allowing them to predict the payoff of a node based on the features of a link pointing to that node.

To make this possible, each agent's genotype comprises a single-layer neural net or perceptron, i.e., a vector of weights $w_1, \ldots, w_{N_f+1} \in \Re$. An agent receives in input, for each outgoing link from the node where it is currently situated, the link's feature vector (step (1) of the algorithm in Figure II.1). It then uses its

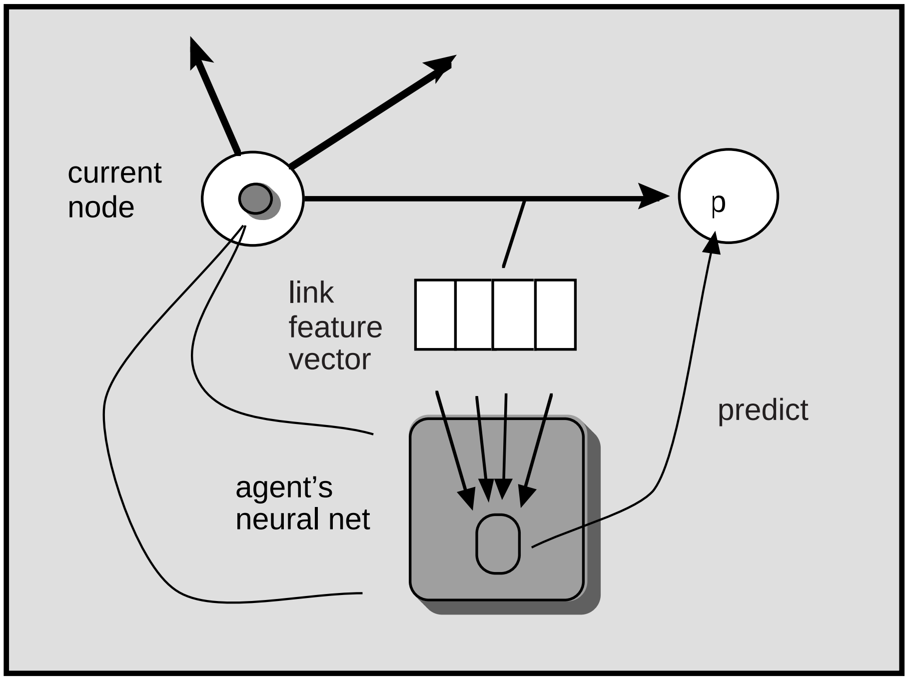


Figure II.3: Schematic representation of an agent's behavior in the graph search problem.

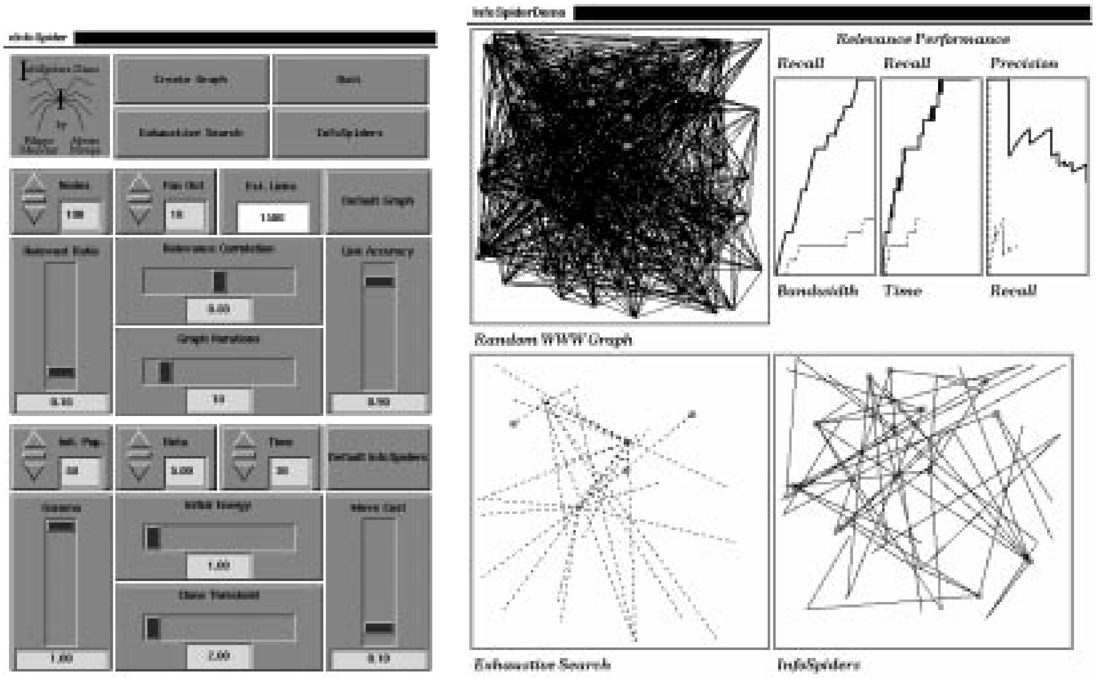


Figure II.4: Graphical interface of the graph search simulator.

neural net to compute

$$o(l) = \frac{1}{1 + e^{-\left(w_{N_f+1} + \sum_{i=1}^{N_f} w_i f_i^l\right)}},$$

i.e., its prediction of the payoff $p(l)$ of the node that l points to. The situation is illustrated in Figure II.3. Finally (step (2)) the agent follows a link that is picked by a stochastic selector among the links from the current node, with probability distribution

$$\Pr[l] = \frac{e^{\beta o(l)}}{\sum_{l' \in node} e^{\beta o(l')}}$$

where the β parameter is a component of the agent's genotype describing the importance attributed to link predictions.

There exists by construction an optimal weight vector such that the corresponding neural net predicts payoff within accuracy A (a parameter). Agents with such a genotype can follow the best links and thus achieve optimal fitness (maximum payoff intake). The energetic benefit of an action is the payoff of the newly visited node, provided it had not been previously visited by any agent (step

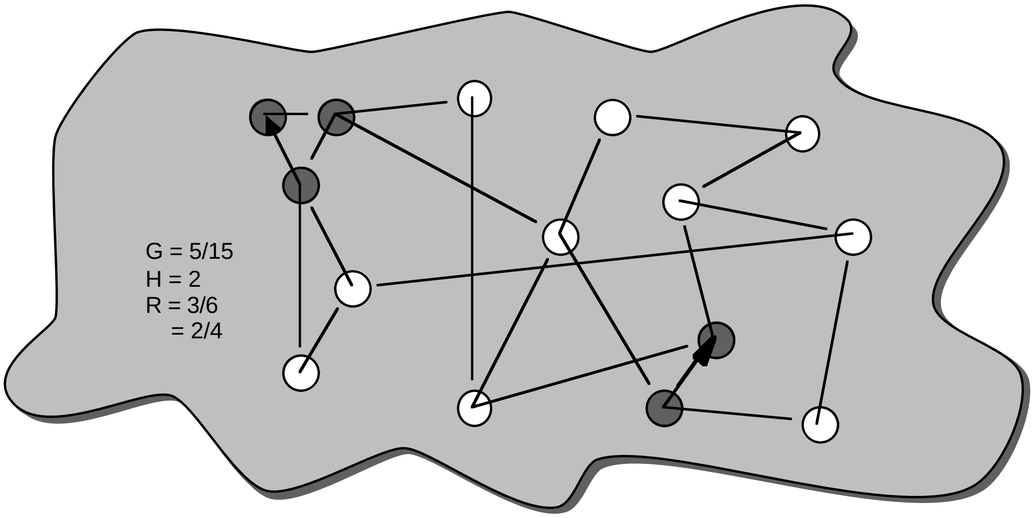


Figure II.5: The major parameters of artificially constructed search graphs.

(3); nodes are "marked" to keep track of used resources). The environment is not replenished; a node yields energy only once. A constant energy cost is charged for any new node visited. A smaller cost is also charged for previously visited nodes, to prevent endless paths through visited nodes. At reproduction (step (5)), an agent's genotype is cloned and mutated to obtain the offspring genotype. Both β and some of the weights are mutated by additive uniform noise (with the constraint $\beta \geq 0$). To study the effect of local selection in isolation from other factors, no recombination operator is applied.

Figure II.4 shows an early graphical user interface of the graph search simulator. This allows one to generate random graphs at run-time, with topology built in accordance to a number of user-specified parameters. We will focus in particular on the following parameterizations (cf. Chapter IV):

accuracy $A = 1 - \mathit{noise}$, where *noise* is the minimum achievable error in payoff prediction

generality G equals the density of “relevant” nodes, i.e. those whose payoff is above some threshold (cf. carrying capacity in Section II.B)[5]

multimodality H is the number of clusters in which relevant nodes are grouped, each with a distinct optimal weight vector; the irrelevant background has yet another optimal weight vector

autocorrelation R is defined as the conditional probability that a relevant node is linked to other nodes in the same cluster ($R \geq G$; cf. relevance autocorrelation in Chapter IV)

These parameters are illustrated in the example of Figure II.5. Unless otherwise stated, the graphs constructed for the experiments described in this section have $N = 1000$ nodes, an average fan-out of 5, and $N_f = 16$ features constructed with an accuracy $A = 0.99$.

LS vs. breadth-first-search

The first experiment is aimed at comparing the basic local selection algorithm with breadth-first-search, as a baseline graph search algorithm. The latter is a typical exhaustive search algorithm that will visit every node (and thus all the good ones) in time $\Theta(N)$. Yet, in realistic graphs such as the Web, N may be too large for an exhaustive search. The goal is then to find as many relevant nodes as possible in time $\ll N$, and this is the basis of our comparison.

Breadth-first-search can also be viewed as a sort of local algorithm, in the sense that new nodes are always visited by following edges from previously visited ones. However, the search cannot focus adaptively on promising areas and therefore remains uniformly shallow.[6] While we expect any local search algorithm to take advantage of topological features such as autocorrelation, we expect that the adaptive character of LS algorithms will optimally exploit such local structure.

---

[5] The term is borrowed from information retrieval[157].

[6] Depth-first-search would suffer from the analogous problem of remaining uniformly narrow.

Since breadth-first-search is not an adaptive algorithm, we are only interested here in exploring the search bias of LS as compared to the "blind" search order of breadth-first-search. To this end, in this experiment we do away with the neural net part of the agent representation. Instead we assume that an agent receives in input the link estimates directly, and uses the stochastic selector to decide which link to follow. The behavior is completely specified by the value of the β gene. $\beta = 0$ means that the agent considers the estimates as noise and picks a random link with uniform probability; high β basically implements a localized best-first-search.

The way that individual agents' search strategies adapt in the LS algorithms is by selective reproduction and mutation of β:

$$\beta_{offspring} \leftarrow \beta_{parent} + \Delta\beta$$

where $\Delta\beta$ in this experiment is drawn from a uniform distribution in $[-\beta_{parent}, +\beta_{parent}]$. Evolutionary adaptation cannot, however, capture regularities over portions of the environment whose scale is significantly smaller than the distance traveled over a lifetime. For this reason, learning could be considered just as crucial a mechanism for adapting individual strategies to cope with environmental variations (step (4) in Figure II.1). One possibility would be to use the sign of the change in energy following an action as a reward/penalty signal, as in:

$$\begin{aligned} \beta &\leftarrow \beta + \Delta\beta \\ \Delta\beta &= \begin{cases} \eta_+\beta & \text{if } \Delta E \geq 0 \\ -\eta_-\beta & \text{otherwise} \end{cases} \end{aligned}$$

where $0 \leq \eta_+, \eta_- \ll 1$ are small learning rates. In this experiment no reinforcement learning will be used; in the LS case β will adapt by mutations alone.

Since we are modeling a distributed environment, i.e., a domain in which there is a cost associated in accessing information about the search space, we can assume that search algorithms will keep such expensive information in their own data structures to avoid duplicate tolls. For example, caches are normally used to limit bandwidth waste in accessing documents over the Web. We model this by assuming that both LS and other search algorithms used for comparison employ such caches. Therefore we consider the number of *new* nodes visited (for the first time) as our measure of time.

Figure II.6 shows a comparison between breadth-first-search and LS in three random graphs. We explore the implications of different models of information structure and link estimation accuracy (modeled by R and A, respectively) for evolutionary adaptation. For the top experiment only the information structure counts ($R = 0.75$), the link estimates being essentially noise. Evolving β values are of no consequence. The recall-time plots agree with the prediction that the information structure (high R) is a sufficient condition for LS to outperform non-adaptive algorithms. In the middle experiment the situation is reversed: there is no information structure in the graph, but the links provide useful hints for the agents. These can be exploited by strategies with evolving β values. The plots show that this environment provides LS with cues that are sufficient to secure an even better performance. Finally, the bottom experiment attempts to model a more realistic situation in which the information is somewhat structured (less than in the top experiment) and the link estimates are somewhat reliable (less than in the middle experiment). Once again, LS significantly outperform breadth-first-search. Figure II.6 does not show the case of graphs with neither information structure ($R = G$) nor link accuracy ($A = 0$). Predictably, the performance of LS in these conditions is no better than that of breadth-first-search.

LS vs. global selection

We have then used the graph environments to compare local and global selection [123]. Binary deterministic tournament selection was chosen as the global scheme for the comparison because of its steady-state nature. Step (5) of the algorithm in Figure II.1 is modified for tournament selection by using the energy level of a randomly chosen member of the population in place of both θ for reproduction

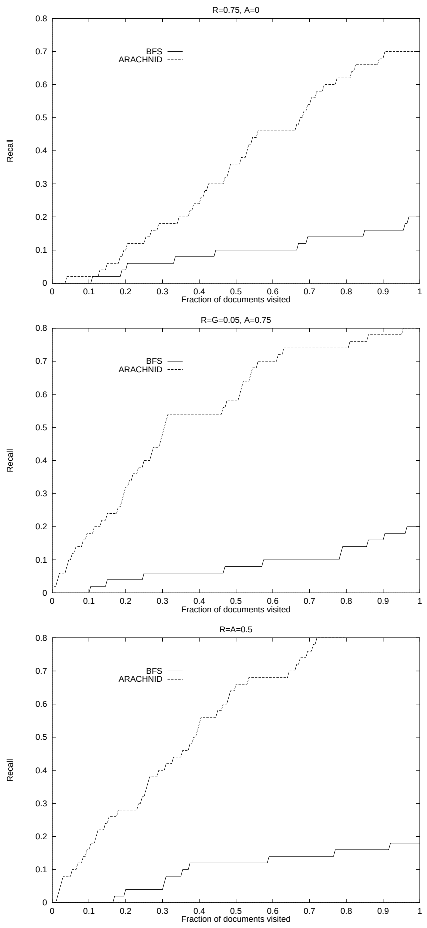


Figure II.6: LS performance versus breadth-first-search (BFS) on three random graphs with $\langle \text{fan-out} \rangle = 10$, and $G = 0.05$. The LS algorithms is marked "ARACHNID" in the legend after the name of an adaptive retrieval system modeled by the simulator. In the top experiment $R = 0.75$ and $A = 0$; in the middle one, $R = G = 0.05$ and $A = 0.75$; in the bottom experiment, $R = A = 0.5$. The plots show recall as a function of the fraction of nodes (documents) visited, out of a total of 200. Recall, defined here as the fraction of relevant nodes that have been visited, is a measure normally used in information retrieval (cf. Chapter IV).

G	0.025	0.05	0.1	0.2
R	0.2	0.4	0.6	0.8
H	1	2	4	8

Table II.2: Parameterizations of the graph search problem.

and ω for death.

The algorithm is stopped when 50% of the nodes have been visited. Figure II.7 illustrates the difference in performance typically observed between the two selection schemes for eight example graphs. The recall level (fraction of relevant nodes visited so far) is plotted as a function of time (fraction of *all* nodes visited so far). Local selection populations continue to discover a constant rate of good nodes, while tournament populations tend to converge prematurely. Notice that for $R = G$ (top right graph), the graph topology is not informative of payoff and in fact this situation correspond to the worst performance for the local selection algorithm, which performs only slightly better than a random walk (a random walker's expected recall would be equal to the fraction of nodes visited).

The same experiment was repeated 64 times, over a wide range of graph parameters, shown in Table II.2. Across all graph parameterizations, local selection significantly and consistently outperforms tournament selection. The improvement varies depending on the graph parameters, but is generally between two- and ten-fold.

Varying G, the density of good nodes, does not result in any noticeable trend across all other experimental conditions. Figure II.8 shows the base case of lowest H (unimodal graphs) and R (unstructured graphs). The recall level plotted is recorded after 500 nodes have been visited.

Increasing R, the correlation among good nodes, is equivalent to increasing the importance of locality; where an agent is situated has greater consequence in determining how well it will do in the future. We therefore expect local selection's performance to improve accordingly. Figure II.9 shows the case of unimodal graphs and intermediate G. We observe that the performance of tournament se-

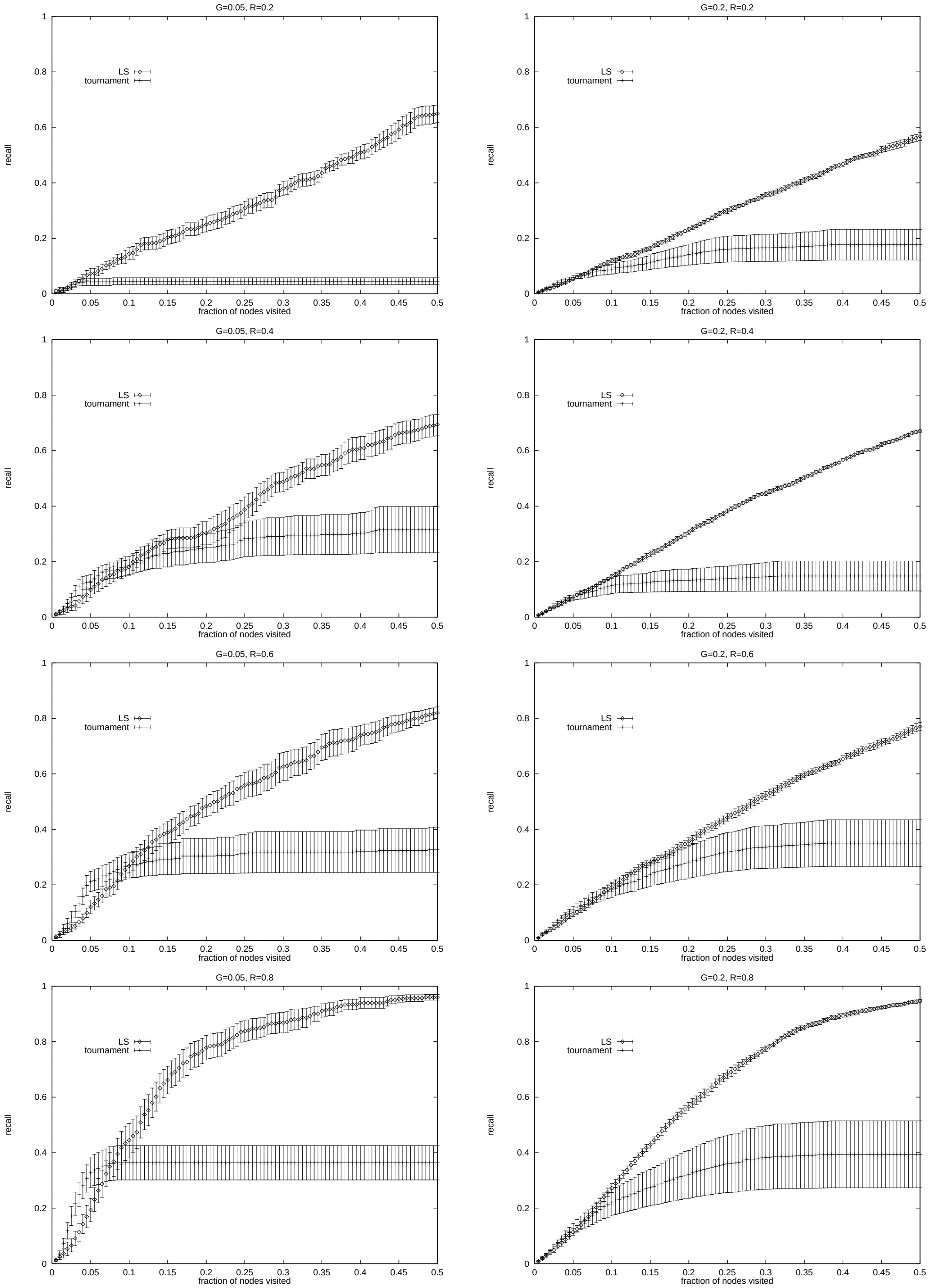


Figure II.7: Performance of local selection vs. tournament selection on typical random graphs with $H = 2$ and various values of G and R (shown above the plots). Error bars indicate standard errors across multiple runs with the same algorithm and graph parameters.

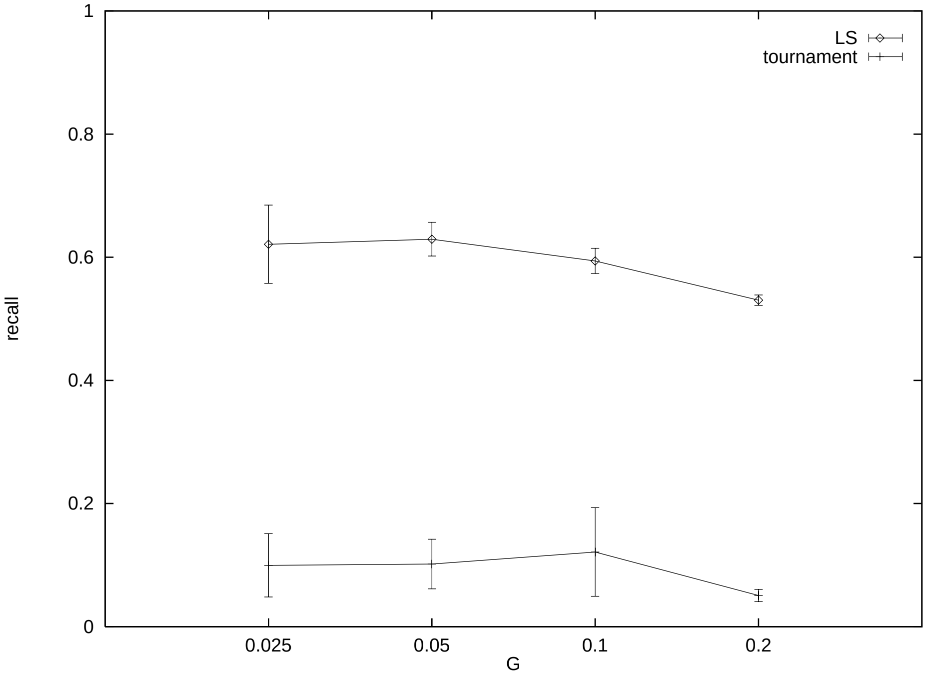


Figure II.8: Performance of different selection schemes searching graphs with $H = 1$, $R = 0.2$, and various values of G.

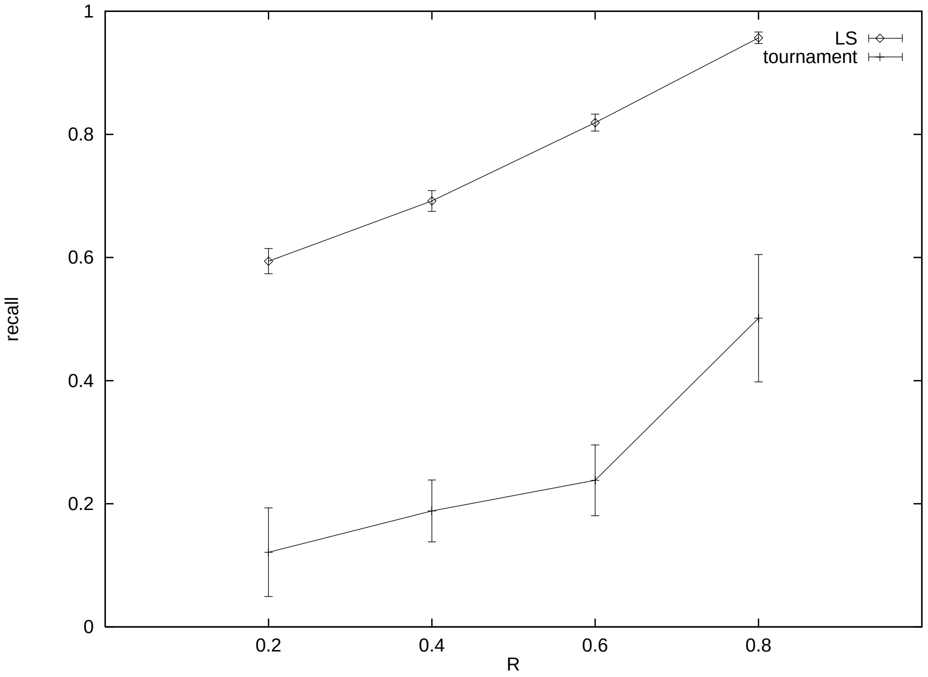


Figure II.9: Performance in graphs with $H = 1$, $G = 0.1$, and various R values.

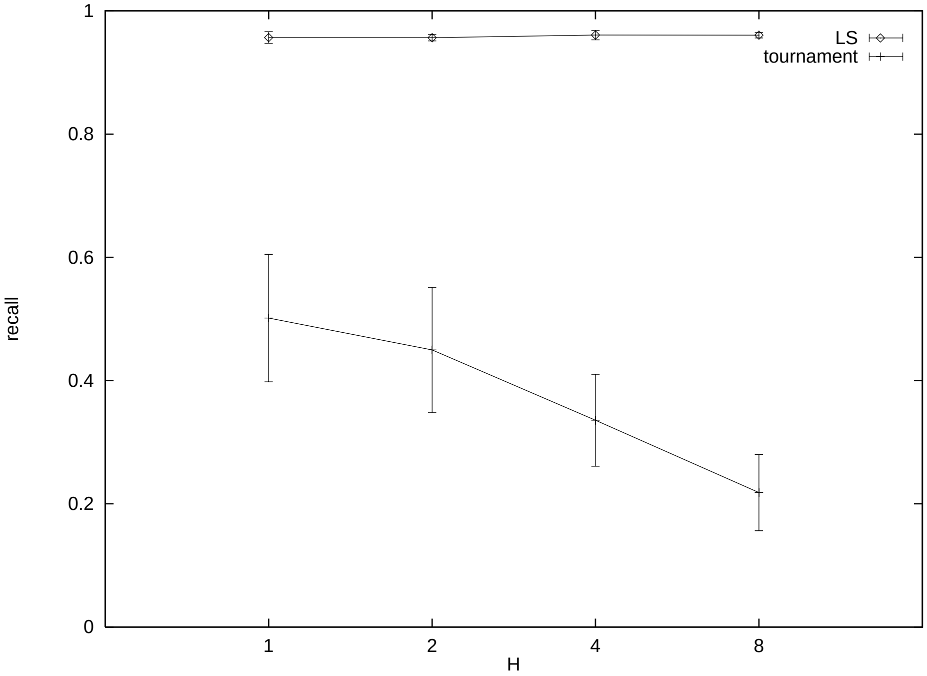


Figure II.10: Performance in graphs with $R = 0.8$, $G = 0.1$, and various values of H.

lection also increases with R, yielding a consistent advantage in favor of local selection.

Finally, increasing H makes the problem multimodal and therefore we expect tournament selection to degrade in performance due to premature convergence. Figure II.10 illustrates this trend in the case of high R and intermediate G. The advantage in favor of local selection increases with H as predicted.

Internalization of global cues

Another important question is, Do agents really become adapted to the particular environment in which they are situated? I.e., is evolution allowing agents to detect important features of the information space and adapt their strategies accordingly? The goal of a third set of experiments was to test the capability of agents evolving by the local selection algorithm to internalize *global* environmental cues [114]. The signal considered was the accuracy of payoff predictions based on

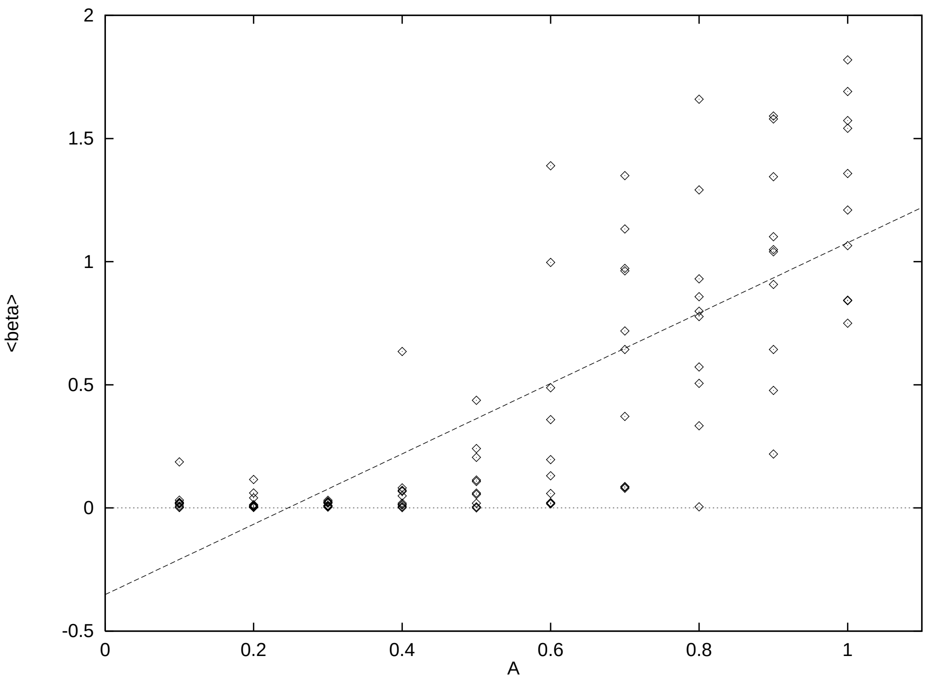


Figure II.11: Correlation between link prediction accuracy A and evolved β parameters, averaged over agents in the population. Linear regression is also shown.

link cues, i.e., the potential accuracy of optimally evolved agents. For high A, the optimal agent strategy is best-first-search; for low A, it is random-walk. Thus internalization of link prediction accuracy implies evolving β values corresponding to the appropriate strategies as implemented by agents via their stochastic selectors.

We ran ten experiments with graphs having an average fan-out of 10 links, $G = 0.1$, $R = 0.75$, and $H = 1$. Each experiment, consisting of 10 runs, used a different value of A between 0.1 (very noisy predictions) and 1.0 (perfectly accurate predictions). In each of these runs, β was initialized with uniform distribution in the range $[0, 5]$ and measured after 750 node accesses. As Figure II.11 shows, the β values evolved by the population were indeed well correlated with the accuracy of the environmental cues. The positive correlation between A and β (with a coefficient of 0.77) indicates that the population successfully internalized environmental cues about accuracy into its agent behaviors.

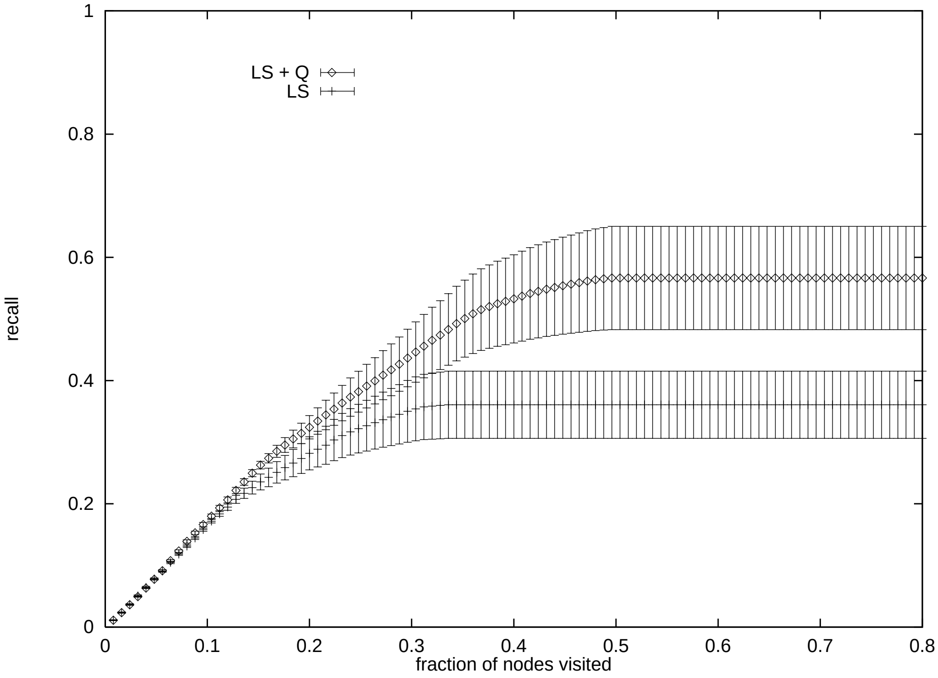


Figure II.12: Effect of Q-learning on recall performance.

Internalization of local cues

The last experiment with graphs is aimed at testing whether local environmental cues can be internalized by reinforcement learning occurring over the lifetime of individual agents. To this end, we have endowed agents with the capability to adjust their neural nets by Q-learning (step (4) of the algorithm in Figure II.1). This algorithm was chosen because it is model-free and easy to implement within the connectionist framework of the agent representation [96]; agents' neural nets are naturally used as Q-value function approximators. An agent compares the payoff of the current node with the prediction based on the features of the link that was followed to visit the node. Perceptron weights are adjusted by the delta rule to improve the accuracy of the link predictor. The net instantaneous energy change (payoff minus cost) is used as reinforcement signal, with future values discounted by a factor of 0.5. Learned changes to the weights are inherited by offspring at reproduction.

We ran two experiments with graphs having $N = 10000$ nodes, $G = 0.25$,

$R = 0.5$, and $H = 1$. In the two experiments Q-learning was disabled and enabled, respectively. Figure II.12 shows that Q-learning yielded a significant improvement in performance. Agents were able to learn, based on where they are situated, the appropriate network weights allowing them to correctly predict payoff. Therefore they could internalize the different local features of the relevant cluster versus the irrelevant background.

II.E.2 Combinatorial optimization

The second problem domain in which we explore local selection is combinatorial optimization. We consider two NP-hard optimization problems, TSP and SAT. In each case steps (1) and (2) of the algorithm in Figure II.1 reduce to evaluating a new candidate solution. In order to apply LS, we need to implement step (3) through some model of environmental resources and energy benefits and costs. We compare the performance of two evolutionary algorithms, with local and global selection, respectively. As in Section II.E.1, deterministic binary tournament selection is chosen as a representative of global selection schemes mainly because it does not require global operations such as averaging, and thus it fits naturally within the steady-state framework of the algorithm.

Traveling salesman problem

The traveling salesman problem (TSP) is to find the shortest tour through all the nodes in an undirected graph. The length is given by the sum of the weights of all the edges through the tour. The Euclidean TSP is the special case in which each node corresponds to a point on the plane, connected by edges to all other nodes, and the weight of an edge is given by the Euclidean distance between the two points connected by the edge.

We generate a Euclidean TSP instance by distributing points uniformly in the unit square. An agent's genotype represents a tour, i.e., a permutation of the order in which points are to be visited. While no crossover is used, two ad-hoc mutation operators are applied: (i) swapping two random points, (ii) reversing the subtour between two random edges. This operation, called *2-Opt,* is a well-known local search strategy for the TSP [75].

For tournament selection, the tour length is used to compute fitness. For local selection, edges between points represent the shared resources. Every time an agent tests a tour, a usage count associated with each traversed edge is incremented. The agent is then charged an energy cost based on the accumulated usage counts, and receives an energy benefit based on how good (short) the tour is. At replenishment, usage counts are redistributed uniformly across edges and decreased by a constant amount that determines carrying capacity. This model resembles the Ant Colony system [40], in which agents deposit pheromone on used edges; edges are chosen based on both length and amount of pheromone accumulated by the passage of previous agents. However, the way of associating resource consumption with solutions in the Ant Colony algorithm is different in that agents are encouraged to agree upon, rather than diversify, their use of edges.

Figure II.13 illustrates the performance of the two selection schemes. Performance is plotted as percentage excess tour length over the Held-Karp bound. The latter is an approximate, analytically derived lower bound for Euclidean tours in the unit square. The plot shows that the comparison between LS and tournament selection performance on the TSP is in clear favor of the global scheme. Local selection does not seem to apply sufficient selective pressure.

In the run of Figure II.13, agent are initialized with random tours. In another run, we initialized agents with good starting tours using the nearest-neighbor heuristic, which produces Euclidean tours whose average length is 29.3% above the Held-Karp lower bound [75]. The results are shown in Figure II.14. LS in this case significantly outperforms tournament selection; the latter consistently converges prematurely to less optimal tours.

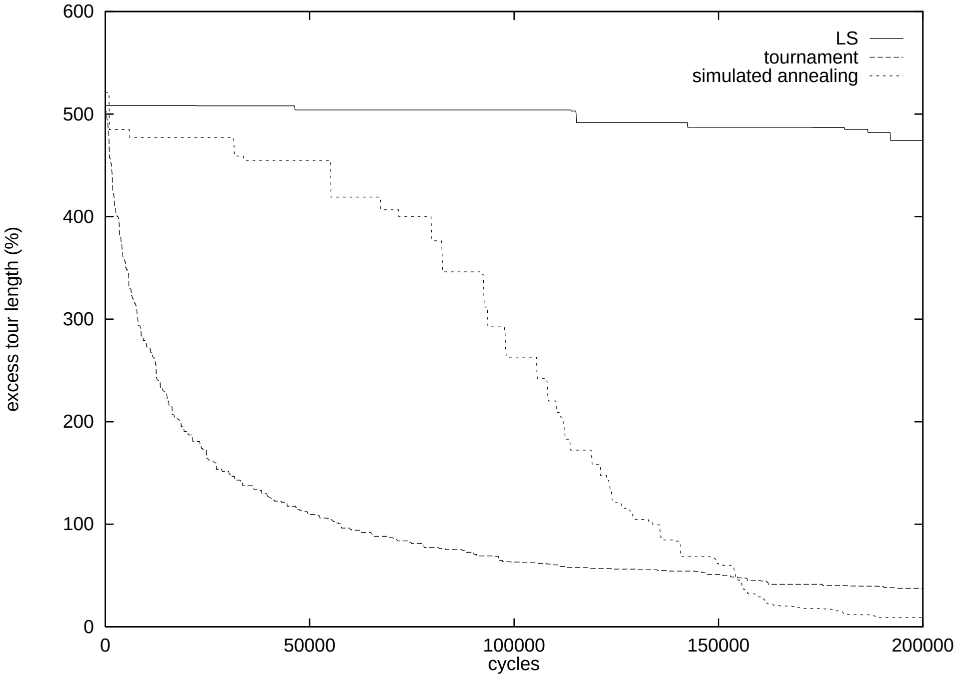


Figure II.13: Single-run performance of EAs with local and tournament selection on a 100-point Euclidean TSP with random initial tours. The performance of a simple simulated annealing algorithm, with geometric annealing schedule, is also shown for sake of comparison.

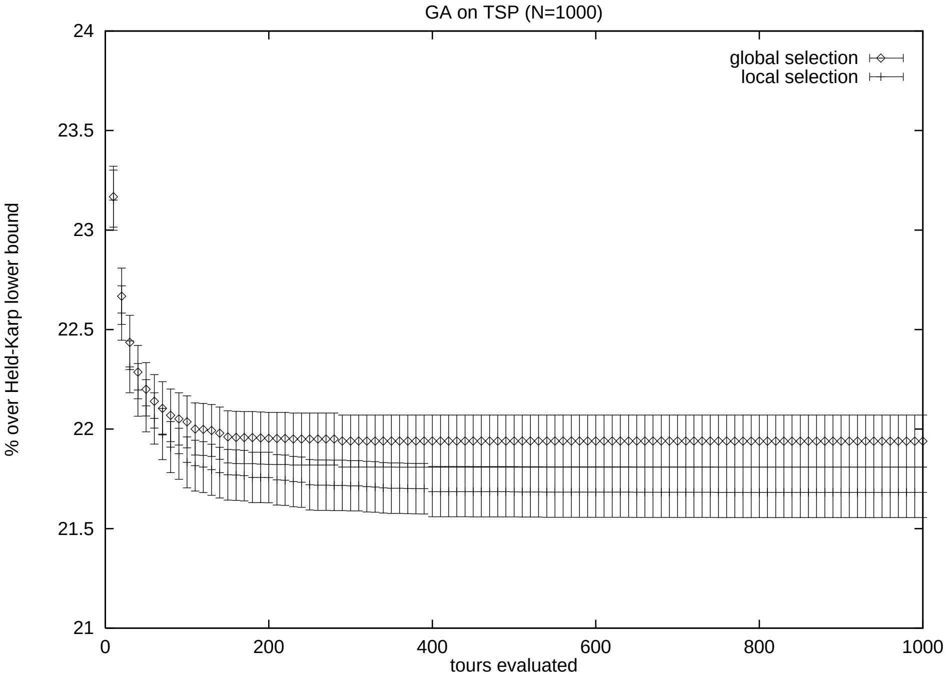


Figure II.14: Performance of EAs with local and tournament selection on 1000-point Euclidean TSP instances with greedy initial tours.

Satisfiability

For the second combinatorial optimization problem, MAX-3-SAT, we generate random CNF formulas in which each clause contains 3 literals, and seek truth assignments that maximize the number of satisfied clauses. The number of clauses in the formulas is a multiple of the number of variables, chosen so as to make the problem maximally difficult [162]. Individual genotypes contain binary truth assignments that are randomly initialized. Due to this binary representation, crossover can be applied in this problem; we use point crossover with panmictic mating — a parents can be recombined with any other member of the population. A local search strategy is applied at each step; a random literal is selected from one of the unsatisfied clauses and negated. This is an optimal strategy for SAT [141], and we also use it as a mutation operator with probability 1.

For tournament selection, fitness is simply the fraction of satisfied clauses. For LS, the energy cost charged for each evaluation is a constant. The energy

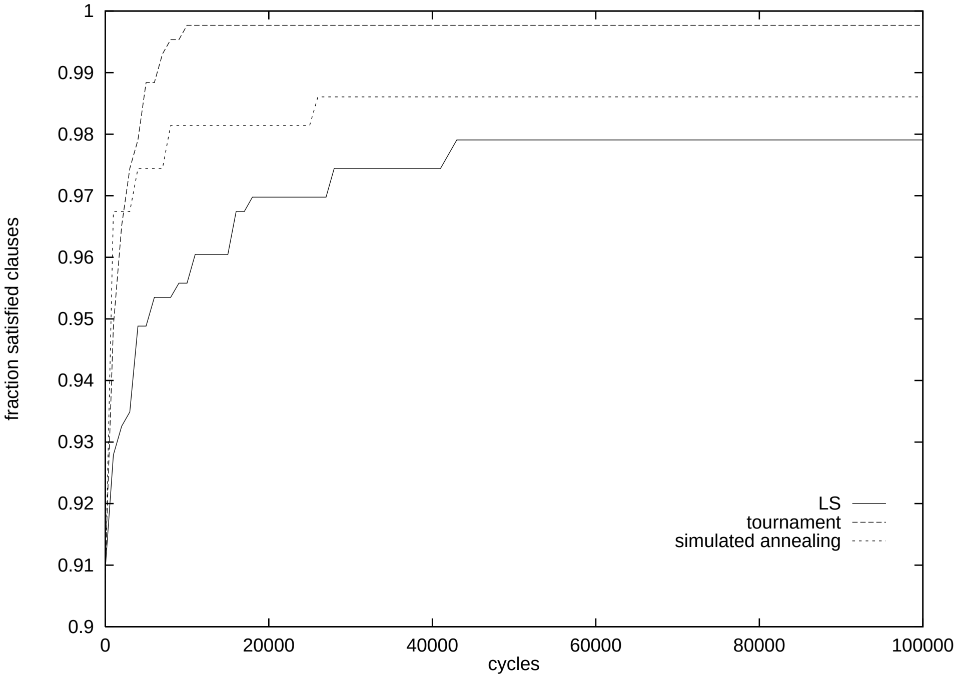


Figure II.15: Single-run performance of EAs with local and tournament selection on a 100-variable, 430-clause MAX-3-SAT problem. The performance of a simple simulated annealing algorithm, with geometric annealing schedule, is also shown for sake of comparison.

benefit, however, depends on the fraction of clauses that are satisfied by the agent's truth assignment, and for which there are available resources. Environmental resources are associated with each clause, and decremented every time an agent satisfies that clause. At replenishment, all clause resources are incremented by a constant that determines carrying capacity.

Figure II.15 shows that for SAT with random initial solutions, as for the TSP, tournament selection performs better than local selection. Again, the latter does not seem to exercise a sufficient selective pressure.

II.E.3 Remarks

The performance of local selection strategies is related to the tension between exploration and exploitation, as for evolutionary algorithms in general. The interplay between these two opposite forces determines the EA's capability to make progress toward good solutions without prematurely converging to suboptimal ones. The appropriate balance, of course, is problem dependent. LS is a very weak selection scheme, so it ensures better performance in tasks where the maintenance of diversity within the population is more important than a speedy convergence to the optimum. This is the case for multimodal optimization and sublinear graph search, as we have shown. LS does in fact maintain population diversity at a much higher level than tournament selection.

The maintenance of diversity by local selection is illustrated in Figure II.16, where population entropy is plotted over time for the two selection schemes in TSP problems with greedily constructed initial tours. Entropy is computed by

$$S = -\sum_{l} f_l \log(f_l)$$

where f_l is the frequency with which edge l is traversed by agents in the current population. Since all of the tours making up the population are quite good in these runs, tournament selection exploits the information too quickly, excluding good solutions and eventually converging prematurely to suboptimal tours.

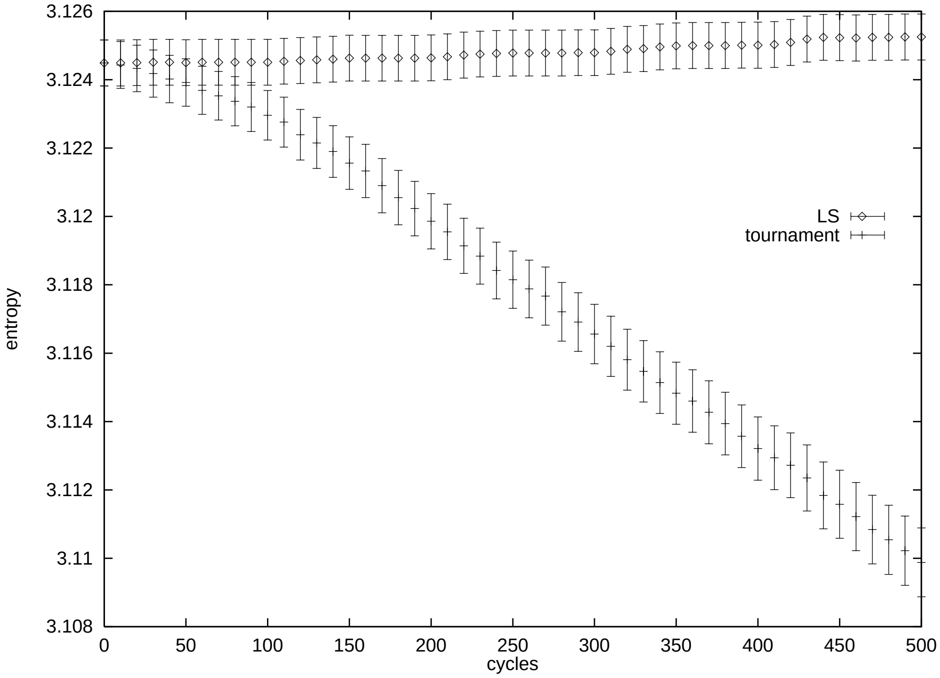


Figure II.16: Entropy of population solutions for 1000-point Euclidean TSP instances with greedy initial tours.

On the other hand, exploiting information is also necessary to guarantee progress. For problems requiring effective selection pressure, LS may just be too weak. This may explain its failure at the NP-hard combinatorial optimization problems that we have considered. The only selection pressure that LS can apply comes from the sharing of resources. Therefore the way in which environmental resources are coupled with the problem space in a particular implementation of LS is crucial to its success. We have explored a few alternatives for TSP and SAT, but LS has been consistently outperformed by tournament selection.

Local selection algorithms can be used whenever the environment provides for appropriate data structures for maintaining the resources associated with fitness. At a minimum, in order for LS to be feasible, an environment must allow for "marking" so that resources may be shared and in finite quantities. In the graph search problem, visited nodes are marked so that the same node does not yield payoff multiple times. If marking is allowed and performed in constant time, the time complexity of LS is also $O(1)$ per individual. This is a big win over fitness sharing, the obvious alternative for distributed or multimodal optimization (cf. Section II.A.1). Further, in a distributed implementation there is no communication overhead and thus the parallel speedup can be as large as the number of agent processes.

However, all problem spaces do not lend themselves to being used as data structures. For example, a task may have a fitness function that is static and expensive to evaluate. And for problems such as continuous function optimization, marking the environment would imply a discretization of the search space; arbitrary precision would make LS at least as expensive as fitness sharing.

In applications such as robotics, the marking problem would not arise because physical state can be "queried" by sensors to provide a situated agent with history information (e.g., an object was moved, or a resource was consumed). Software agents, on the other hand, must be provided by their execution environment with capabilities such as access to local storage. These capabilities may require the solution of several system-related problems, such as dynamic resource management, security, and service payment transactions.

Finally, it may be difficult to devise an appropriate isomorphism of an arbitrary problem such that the environmental resource model can be applied successfully. For example, associating environmental resources to partial solutions of a combinatorial optimization problem may require a decomposition property that the problem is unknown to possess. In the particular TSP and SAT problems where LS has failed to prove advantageous, it is not clear whether the poor performance is to be attributed to inadequate resource models or to weak selection pressure. More experiments might shed light on this issue and help us better characterize the problem domains in which local selection is feasible and successful.

Chapter III

Latent Energy Environments

Evolutionary theory and ecological theory are rich in data and analytical models, and in recent years have increasingly taken advantage of the modeling tools made available by computational methodologies. Such tools provide new opportunities, through numerical simulation, for dealing with systems whose complexity is too great to be captured by common mathematical tools, such as systems of differential equations. In this chapter we study evolutionary and ecological models of natural adaptive systems based on local selection and internalization.

III.A Background: Issues and models

The work reported in this chapter relates to several previous attempts to characterize the role of environmental complexity in guiding the evolution of adaptive behaviors. General modeling tools aimed at explaining the behaviors of natural populations interacting in real environments are often limited in one of two respects. Analytical models are rigorous, but may be too simplistic to capture the complexities of the interactions they are meant to describe. Simulation tools are often broad in scope, but too convoluted to allow for formal analysis. In this section we review different aspects of such models, both analytical and numerical, with respect to a number of open issues in evolutionary and ecological theory.

III.A.1 Artificial life

Roughgarden *et al.* [151] have discussed the role and usefulness of computer simulations in biological theory. They identify two classes of minimal models in which the contribution of computer simulation can be most effective and relevant for biologists. *Idea models* need not possess the plausibility of a biological theory; they only need to point to the *possibility* of a certain idea, which can in turn inspire biologists to devise experiments in the field to test for more formal hypotheses. *Minimal system models,* though still very simple, are more plausible and aim to provide an explanation for some observed phenomenon; their assumptions and consequences must be testable in the field, so that cooperation with biologists is more essential.

Several open questions lend themselves to be studied via minimal models using computer simulations. We want to list some of them for which we believe that local selection models, because of their non-optimizing selection pressure, can become useful tools of analysis and interpretation of the biological data. Many questions revolve around the interactions between learning and evolution: How do the genotype and the environment interact to produce the phenotype? How does this interaction affect evolution? What determines the degree of learning versus genetic determination involved in the development of a trait, with respect to environmental variability? How does learning affect the population-genetic selection on a trait, altering its evolution?

Computational models from the artificial life community have already been applied to some of these problems: for example Hinton and Nowlan [66] have proposed a well-known idea model to explain the Baldwin effect, by which learning can influence evolution. Many other researchers have used simulation tools to build idea models that address related issues [2, 78, 77, 97]. The evolution of metabolism as a self-sustained, adaptive, complex chemical system is another fascinating open problem that has been studied with artificial life models [148, 5, 81].

III.A.2 Environment

A different set of issues, closer to the interests of ecologists, has also been the target of many interdisciplinary studies by artificial life practitioners: the relationship and interplay between environmental complexity and the emergence of adaptive behaviors. Levin *et al.* [90] point out that there is a need for computational models to deal with the collective dynamics of heterogeneous ensembles of individuals, especially with respect to the ways in which their interactions with the environment scale from small to large spatial regions.

Genetic algorithm modelers have generally assumed a fairly direct correspondence between the genotypic data structure manipulated by genetic operators (mutation, crossover, etc.) and the phenotype's fitness (i.e., number of offspring), downplaying the role of the environment in determining such fitness. But the appropriateness of phenotypic behaviors is obviously conditional upon environmental context, and the definition of "adaptive" behavioral features must therefore depend on a complex interplay of genetic and environmental circumstances.

What is the role of space in shaping the selective pressures leading to adapted individuals? How can we compare the efficiency of behaviors in different (natural or artificial) environments? How can we characterize different environments in such a way as to predict what types of behavior would be optimal or even appropriate? Can adaptive algorithms lead an animal to behave (forage, choose habitat) optimally, as predicted by optimal foraging theory? How does an animal learn behavioral interactions such as where to make its home or how to court another? How can interactive signaling influence learning, truth in signaling, and cooperative equilibria? What determines the size of a social colony or other complex societies? To what degree can we model (intra-species or inter-species) interactions among individuals in terms of environmental interactions? What are the limits in the temporal scales of environmental dynamics within which populations can adapt?

Several models have been put forth to begin to address these issues. Echo

[68, 69, 78] is a notable example; it is a class of models that attempts to be inclusive with respect to the many different levels of interactions that can exist between agent and environmental resources, between different (cooperating or competing) agents within same species, and between agents of different (cooperating or competing) species.

Sugarscape [42] is a model aimed at the development of a "bottom up" social science. It simulates the behavior of agents located on a landscape of a generalized resource (sugar). A remarkable range of social phenomena, such as group formation, cultural transmission, combat, and trade are seen to "emerge" from the interactions of individual agents following simple local resource-approaching rules. Many other simulations models have been proposed to address more or less specific questions in this domain [185, 175, 176, 12, 151, 10, 177, 47, 46].

In the following sections of this chapter we describe an artificial life framework within which both minimal system and idea models can be designed easily to explore the classes of evolutionary and ecological question mentioned above.

III.A.3 Ecology

Local selection and more generally endogenous fitness [133] are different aspects that we consider key to modeling populations adapting to an environment — as opposed to explicitly optimizing some predetermined trait. A general reason behind this belief comes from a common criticism of genetic algorithms made by evolutionary biologists. It is pointed out that in nature we often observe "sub-optimal" adaptation while GAs are more suitable to perform optimization. Gould and Lewontin [57], for example, assume *a priori* that evolved behaviors are not optimal. Less extreme criticisms of optimality are better supported by experimental observations; for example, apparently optimal behaviors for the individual can lead to resource depletion and extinction of the group under density dependence of fitness [24]. A robust model of group selection has been proposed to account for the evolution of traits that are sub-optimal from the point of view of individual selection [51]. While we don't deny the usefulness of optimality analysis, we propose that local selection, eliminating the global mechanism responsible for the optimizing behavior of standard GAs, may allow our computational models to approach the richness and complexity of population biology.

In evolutionary biology, since the classic works of Fisher on populations (e.g., [45]), models of selection have mirrored the role of the environment on selective pressures and population dynamics. Two general classes of models are used commonly for fitness and selection, while evolutionary computation methods until recently seem to have been inspired only by one of them. Biologists instead can point to the environmental conditions that make one or the other class of models better suited for capturing realistic mechanisms of adaptation.

When environmental resources are abundant, or their exploitation is so poor that increases in efficiency make the resources appear as though they were infinite with respect to an existing population, it is possible to optimize resource usage without apparent bounds. These conditions lead to an exponential growth in population, and successful individuals become dominant in proportion to their reproduction rate. The population grows geometrically over time:

$$p = p_0 e^{rt}$$

where p_0 is the initial population size. The intrinsic rate of population increase, r, is therefore used as fitness. This is appropriate for expanding populations, with negligible crowding, where density has no influence on birth and death rates. A fitting example is the situation in which death rates are driven by seasonal climate changes [91]. This model is referred to as geometric growth or *r-selection.*

The situation is fundamentally different when the usage of resources by the population is at a level near the bounds imposed by the environmental replenishment rate. In these cases the *carrying capacity* of the environment, K, can be used as fitness:

$$p = \frac{K}{1 + (\frac{K}{p_0} - 1)e^{-rt}}.$$

This is appropriate for crowded populations in which individuals must compete for finite resources. This model is called *K-selection.* Fitness is said to be *density dependent* when individuals do not have much interaction other than by sharing resources, and selective pressure is proportional to the density (or size, assuming fixed territory) of the population.

Roughgarden [150] showed that r- and K-selection need not use different fitness definitions. In fact, using Sewall Wright's *selective value* [189] as a universal fitness measure, both types of selection follow. Evolution in either case favors genes producing phenotypes with the highest fitness. Depending on the harshness of environmental conditions and on population crowding, either of the two extreme selection/fitness models may be appropriate, as well as a stable coexistence between them.[1]

We can easily notice the correspondence between the models of selection used in evolutionary biology and evolutionary computation. The fitness of classic genetic algorithms corresponds to r-selection. For example, this is the basis of Holland's initial argument in support of GAs (the *schemata theorem* [67]) and of much of the later work analyzing the behavior and performance of GAs. The limit of finite resources imposed on computational models by finite computer memory and time (hence finite populations) is also the limit of these analyses, that consequently treat genetic drift as a sort of "sampling error" rather than a first-class mechanism of evolutionary pressure.

We conclude that when standard GAs have been used as computational models of biological adaptation, they have only captured one class of environmental conditions — namely, unbounded resources. On the other hand, biologists have long studied analytical models of broader classes of natural selection mechanisms, including density dependent selection [102].

Recent computational models of adaptive populations have reduced the

---

[1] Both r- and K-selection models are further complicated by direct interactions among individuals, e.g., communication. But in the present discussion we limit our observations to the simpler case in which selection is the only — or main — form of interaction.

gap with biological models and increasingly made use of fitness measures defined in biology to characterize the role of the environment in their simulations. Well-known models focusing on adaptation to the environment rather than optimization include Echo [68] and ERL [2] among others. The SPO model [175, 177] considers reproductive behaviors that make selection more a matter of individual "choice" than a response to metabolic/energy considerations. Endogenous fitness and local selection have introduced the ideas of open-ended adaptation and density dependence into models based on evolutionary computation algorithms [133, 123]. In short, artificial life's growing interest in simulation of evolution's creative, open-ended aspects can draw on a growing body of literature for mathematical and computational analyses of the behavior [22].

III.A.4 Space

Spatial features of the environment have been identified long ago as crucial in characterizing environmental complexity for both natural and artificial systems [150, 149]. Ecologists recognize the importance of spatial models, in which relationships between members of a population and their environmental resources are mediated by the distribution of such resources in space [131]. They point out the lack of studies on how to relate phenomena across scales [90]. Ideal free distributions, critical patch size, habitat variability, food gradients, metapopulations and niche adaptation are all examples of important spatial aspects of the environment [152, 50].

Characterizing the ecological consequences of physical constraints in artificial and natural environments is an active area of research. The effects of geographic structure on populations have been studied extensively in theoretical biology since the theory of demic selection and shifting balance [188]. Such effects shed light on the *genetic* dynamics of populations evolving in space, thus indicating ways to improve the performance of GAs through niching [110].

However, there is a lack of models that account for the potential influences of space on *ecological* dynamics. Environmental consequences of the spatial dimension are often treated by adding *ad-hoc* variables into partial differential equation models. Recently, theoretical explorations of spatial effects have involved, among others, models of dispersal, habitat fragmentation, patch turnover, reaction-diffusion, and cellular automata [80, 160, 173]. Yet, experimental investigations to test the major hypotheses emerging from such models remain missing because field studies are difficult, expensive, or time-consuming. Simulations can assist in this regard. The EcoBeaker program, for example, lets users explore spatially explicit ecological models on the Macintosh [111].

III.B LEE overview

The acronym "LEE" stands for Latent Energy Environments. LEE is an artificial life modeling framework and is described in this section. The name refers to the fact that in LEE, survival is determined by an agent's capability to realize the energy latent in the environment, by detecting and appropriately combining the resources present in the environment.

III.B.1 Motivation

Adaptation of ecological systems to their environments is often viewed through some explicit fitness function defined *a priori* by the experimenter, or measured *a posteriori* by estimations based on population size and/or reproductive rates. These approaches have been used in the field as well as in computational methods. In the former case, evolutionary ecologists must choose some measurable trait to monitor and characterize the individuals in a population, perhaps finding which traits are best correlated with established measures of fitness such as reproductive rate or success, or environmental carrying capacity. Analogously, the modeler of an artificial life experiment has to define how to map some phenotypic trait, be it somatic or behavioral, to a mechanism of offspring allocation, through some fitness function and selection scheme.

These methods have the merits of having allowed for enormous advances in understanding the dynamics of ecological systems, and having enlarged the set of tools at the disposal of ecological theory from purely analytical mathematics (such as differential equations) to explorative idea and system models (such as computer simulations). However, the relationship between adaptive population and environment continues to look like a "black box" function mapping phenotypic traits to reproductive fitness. It is difficult, in the field, to refine the black box to the point of characterizing the way in which the environment shapes the selective pressures acting simultaneously on each trait while possibly interfering with each other. The analytical tools commonly used by the theorist rapidly become intractable as the spatio-temporal dynamics of the environment become even moderately complex.

Are we then hopeless in the face of the problem of characterizing the interactions between complex environments and adaptive populations? This chapter suggests that modeling techniques developed in the artificial life community may play an important part in refining the role of the environment in the adaptive process. Genetic algorithms have been used as computational models of evolution in a great many simulations of adapting populations. Since we want to focus on the role of the environment, we will consider variations of evolutionary algorithms in which local interactions between agents and the environment guide the adaptive process. In particular, we will pay special attention to local selection (cf. Chapter II) in order to study the effects of localizing the main global aspect of the classic genetic algorithm, i.e., selection.

III.B.2 Environmental complexity

Even disregarding issues of biological plausibility, the artificial life coupling of organisms with environments brings with it a major methodological problem: results reporting behaviors of different organisms in different environments are incommensurate. It is therefore difficult to assess whether an apparently superior behavior is the consequence of more sophisticated adaptive techniques, or is due to the relative complexity of the environments. It is very desirable to be able to define artificial environments of controlled complexity, within which a wide range of artificial life techniques might be directly compared.

Godfrey-Smith [52] provides us with a very good starting point for the definition of both environmental and behavioral complexity, equating complexity with *heterogeneity.* This useful simplification allows environments to be characterized in terms of the number of distinct states they present to an organism, the frequency at which these change, etc. But while Godfrey-Smith abstracts away from specific properties of organization, we want to ground our notion of complexity on observable measures of environmental organization.

There have been several attempts to define generic conditions on environmental complexity that are analytically tractable without constraining evolution's creative potential. Rössler [149] proposed food density as a simple complexity metric for the environment facing an organism. Even a random walk may be an adequate foraging technique if food is abundant, but as environments become more scarce, more coherent movement is required and the foraging organism may need to depend on landmarks, cognitive maps, etc. [187]. Food density is clearly a useful dimension for artificial life simulation, but this single dimension of environmental variability must be extended to include other factors if we are to be able to test the full repertoire of artificial life models.

Latent Energy Environments are a modest step from the single dimension of food density towards richer models. To accomplish a useful measure of environmental complexity without compromising analytical tractability, we extend the standard model of "food" as a spatially localized element of the environment, required for survival. Food is replaced in LEE with a series of inert environmental "atomic elements" that must be combined by organisms in order to realize the *energy* they require for survival. Consider a simple discrete world with cells placed on a two-dimensional grid. Let the only source of (positive or negative) energy be via binary reactions; i.e., combining two elements results in an energy gain or loss, and possibly other by-product element(s). Let the base elements belong to an artificial set of types $\{A_1, ..., A_t\}$. Furthermore, let elements of each of these types be generated (or replenished) according to some spatio-temporal distribution. All the possible reactions can be represented by a symmetric matrix of reactions indexed by element types:

$$A_i + A_j \longrightarrow E_{ij} + P_{ij}. \tag{III.1}$$

A reaction occurs when two reactive elements come into contact.[2] A reaction can only happen when an individual catalyzes it, since reactive elements never occupy the same cell. In the example III.1, two generic elements A_i and A_j yield energy $E_{ij} \in \Re$ ($E_{ij} > 0$ for exothermic reactions, $E_{ij} < 0$ for endothermic ones) and a list of by-product elements indicated by P_{ij}. The reaction matrix and the spatio-temporal distributions of elements together represent the laws of physics and chemistry regulating the interactions between an individual and its environment and among individuals, since the environment of an individual includes the rest of the population. In short, the matrix of reactions and the distributions of elements are the parts of the LEE model which characterize the environment.

We now need some definitions. For any world configuration, there may be many possible combinations of existing elements according to the possible reactions in the matrix. After one of these reactions takes place, two elements are consumed and some new elements (by-products) may appear, giving rise to a new configuration. We can think of this process as a decision tree, where each reaction leads to a new node down the tree, and a path along the tree represents a reaction chain. Eventually, the chain terminates if no possible reactions exist among the remaining elements: a leaf is reached in the tree. Every reaction chain has a corresponding potential energy, given by the arithmetic sum of energies released and/or absorbed by the reactions along the path. For each world configuration, we

---

[2] Not all elements need be reactive: an empty entry in the reaction matrix indicates non-reactive elements.

call *latent energy* the set of potential energies corresponding to all reaction chains starting from that node. The energy remains latent until it is realized as *potential energy* by choosing a particular reaction chain. By specifying the interaction energies in the matrix and controlling the rates at which elements of each type are introduced, the amount of energy available in its latent form can be precisely regulated.

Recall now that a collection of the various elements is distributed across the two-dimensional space of the LEE world, and define the *work* required to release latent energy associated with a pair of elements to be the *distance* one element must be moved in order to occupy the cell of the other. Then, by controlling the spatial distributions of elements in the world, we can dynamically regulate the amount of work required to combine them.

The next step is to introduce organisms into this abiotic environment, by considering their ability to move through the environment and mix its elements as their most fundamental behaviors. We also endow the basic LEE individual with an internal body cavity (*gut*) and assume ingestion of an element into this gut occurs any time an organism occupies a cell containing one. An element consumed in this fashion is then carried with the organism as it moves. Organisms therefore become the agents actively mixing the otherwise static elements of the LEE environment.

In the experiments reported in this chapter, individuals incur constant metabolic costs simply for being alive. That is, there is no additional energy cost for moving (as opposed to remaining in one cell of the environment) or for carrying elements. The work associated with a pair of elements, based on the Euclidean distance between them in the environment, therefore becomes a lower bound on the energy cost an efficient individual would incur in moving to realize their latent energy.

Subtracting work from latent energy and summing over all pairs of elements in the environment, we can estimate optimal energy efficiencies, and from this the maximum *carrying capacity* of an environment. In short, by controlling how much latent energy is available and how much work must be done in order to realize this energy, we can obtain a quantitative measure of how difficult it is for an organism (or ecology of organisms) to survive.

The above definitions — latent energy, work, carrying capacity — allow us to control parameters of the environment without specifying just how latent energy is to be realized by one or more species. The monitoring of population dynamics makes it possible to compare ecologies in different environments, in terms of how efficiently each exploits the energy latent in its own environment. In a single shared environment, competition for the finite resources intrinsically defines the ground on which adaptive success can be gauged.

The simple models of metabolism and environment — achieved by the range of available reactions and by the spatio-temporal distribution of resources, respectively — provide a space in which to explore the effects of discoveries due to more or less complex and efficient exploitations of the energy available in its latent form. The principle of energy conservation and the fixed physics permit us to do so without sacrificing analytical tractability. Energy is conserved in this system because we can always verify that incoming energy (from resource replenishment) equals outgoing energy (from work) plus changes in the latent energy of the environment. The physical laws of the system are fixed because they are embodied by the set of available reactions, and these cannot change during an experiment — unless the experimenter uses changes in the reaction matrix to model dynamic aspects of the environment.

III.B.3 Algorithm

The evolutionary process is modeled in LEE by the steady-state, local selection algorithm of Figure II.1. The algorithm is reproduced in Figure III.1, for ease of reference and with more details about the specific LEE model. As discussed in Chapter II, one important consequence of this model is that the population size does not remain constant throughout an experiment, and extinction is possible. We

```
initialize population of agents, each with energy $\frac{\theta}{2}$
while there are alive agents
  for each agent $i$
    1. input: gather information from sensors
    2. output: use motors to move to new positions
    3. update energy:
      ingest atoms
      catalyze reaction(s)
      $E_i \leftarrow E_i + E(\texttt{catalyzed reaction(s)}) - cost$
    4. optional learning
    5. selection:
      if ($E_i > \theta$)
        clone(i)
        mutate genotype(offspring)
        develop phenotype(offspring)
        $E_{offspring} \leftarrow \frac{E_i}{2}$
        $E_i \leftarrow \frac{E_i}{2}$
      else if ($E_i \leq 0$)
        die(i)
      end
  end
  replenish environment
end
```

Figure III.1: Pseudocode of LEE local selection algorithm.

show in Section III.C that the population size becomes stable spontaneously and robustly when the environmental conditions allow it, without this being imposed externally.

Environmental energy represents the endogenous fitness by which individuals survive and are locally selected for reproduction. Energy in a latent energy environment can be released only through the behaviors of organisms. Behaviors induce reaction sequences, thus catalyzing the transformation of latent energy into usable energy. The notion of behavior — actions taken by the organism that change the world and/or the organism's relationship to the world — immediately binds an organism to its environment in an intrinsic way. The complexity of an environment, defined by its physics, must be matched by the population in order to avoid extinction.

A behavior is defined by a function mapping input corresponding to the organism's stimuli to output controlling its actions. Input (step (1) of the algorithm) comes from a set of genetically-specified sensors, and output (step (2)) drives a set of genetically-specified motors. The behavioral map is implemented in LEE by a feed-forward neural network representing the organism's brain. These different components of an agent's representation are illustrated in Section III.B.4.

The reactions(s) catalyzed by an agent result in energy losses or gains (step (3)) that can be used to guide learning during its lifetime (step (4)). Learning only affects the phenotype, while it is the genotype that is passed on to offspring. At reproduction (step (5)), the genotype of the parent is cloned into that of the offspring, and the latter undergoes mutations. Then the offspring phenotype is obtained by making a copy of the mutated genotype. This distinction between genotype and phenotype allows one to consider non-Lamarckian phenotypic plasticity, whereby learned behaviors are not hereditary. This issue is explored in Section III.E.

In LEE's local selection evolutionary algorithm, adaptation results from the *local competition for the finite resources in the shared environment.* The interactions between individuals and the environment create selective pressures that can be as diverse as the environmental conditions occurring in different places. The lack of a constant selective pressure allows individuals to explore the adaptive landscape without competing with others who are not part of their own local environment. Since the organisms do not interact other than by sharing resources, fitness is density dependent. Genetic drift is strong due to the weak selection pressure of the algorithm, as will become clear in the remainder of this chapter.

III.B.4 Individual representation

In any adaptive model, an individual's behavior is strongly dependent upon its representation. Each individual in LEE is represented by a body and a brain. The body comprises sensory and motor apparati, and internal reservoirs

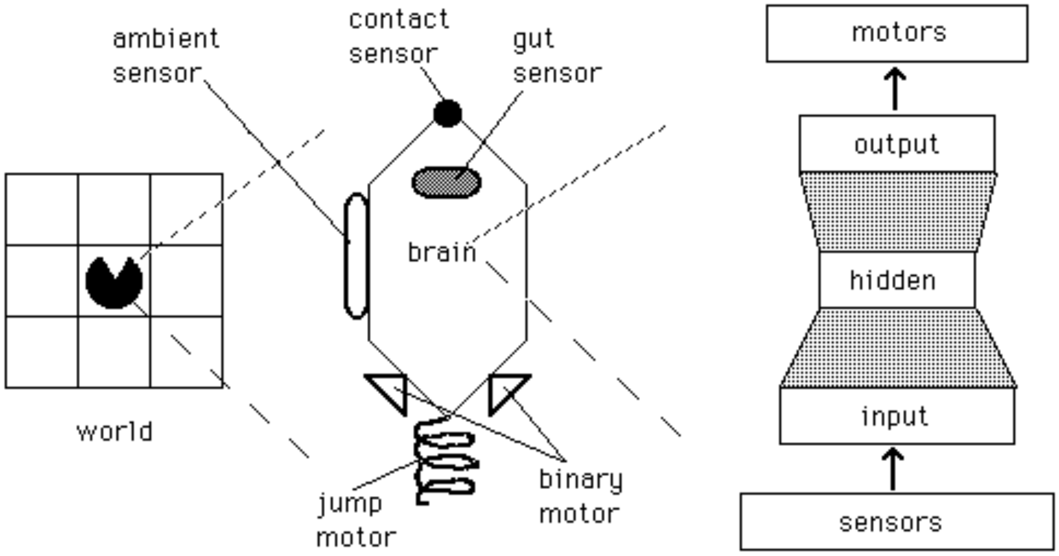


Figure III.2: The body of a LEE organism. On the left, the organism is displayed situated and oriented in the environment; in the middle, the body of the organism is expanded and different possible sensors and motors are shown; on the right, the architecture of a neural net modeling the organism's brain is illustrated. Not shown in the figure, a learning feedback loop can occur either by reinforcement or sensory prediction.

for energy and environmental resources. The brain consists of a neural network that models the behavior of the individual given its environmental conditions. The fact that the individual is embodied makes its interactions with the environment crucial in determining the degree to which behaviors are well adapted.

Body

The brain interacts with the environment external to the body through a sensory-motor system. Figure III.2 provides an illustration of an organism's body structure with a typical sensory-motor system. The body also determines the placement and orientation of an organism's sensors and motors.

The sensory system is composed of a set of sensors with different characteristics. These collect information from either the external world or the internal body of the organism, and map it onto the brain input. Sensors may differ in range, directionality, sensitivity, resolution, specificity, accuracy, etc. Examples of external sensor types used in the experiments reported in this thesis are *contact* and *ambient* sensors. Contact sensors provide a binary presence/absence indication of some element (or complex of elements) in the space directly adjacent to the sensor. They are sufficient to support avoidance behaviors, but not approach. Ambient sensors signal the presence of some element (or complex), summed over cells in a neighborhood of the sensor and weighted inversely according to their distance. Ambient sensors can underlie approaching behavior only if the brain possesses either a memory to compare temporal differences in a single sensor, or multiple ambient sensors placed and oriented differently to compare spatial differences. Two examples of sensory configurations are shown in Figure III.3.

The sensory information is elaborated by the brain to produce an output, interpreted as an action (movement) in the world. Motors function as the output effectors of the organism, changing its location and/or orientation with respect to a fixed environment referential frame. Motors may differ in energy efficiency, power, orientation, accuracy, etc. Examples of motor types used in the experiments reported in this thesis are *binary* and *jump* motors. A binary motor is very simple and has been used in many other artificial life experiments (e.g., by Nolfi *et al.* [138]). It allows an organism to move to the cell immediately ahead, turn 90 degrees right or left, or stand still. A jump motor moves the organism ahead by some distance (determined by the output) and randomly changes its facing direction. In Figure III.4 we illustrate the operation of these motor systems.

Each organism also has an internal cavity we call its *gut.* A gut is capable of holding a number of atoms specified by the organism's genotype. An organism automatically ingests atoms over which it travels, causing these atoms to be placed in its gut (irrespective of gut capacity). If the gut already contains some atoms that are reactive with the new atoms, one or more reactions take place and as a result

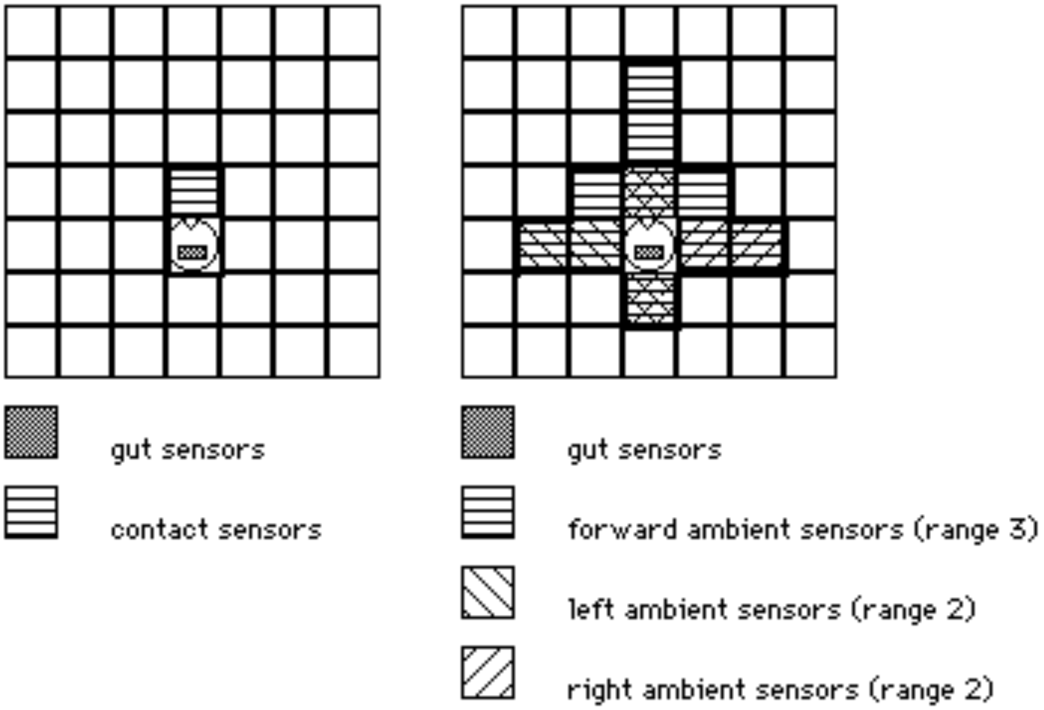


Figure III.3: Two examples of sensory configurations. Both systems use internal gut sensors. One uses only contact external sensors, while the other uses three ambient sensors with different ranges and orientations. Each sensor can be designed to signal the presence of any complex of different elements.

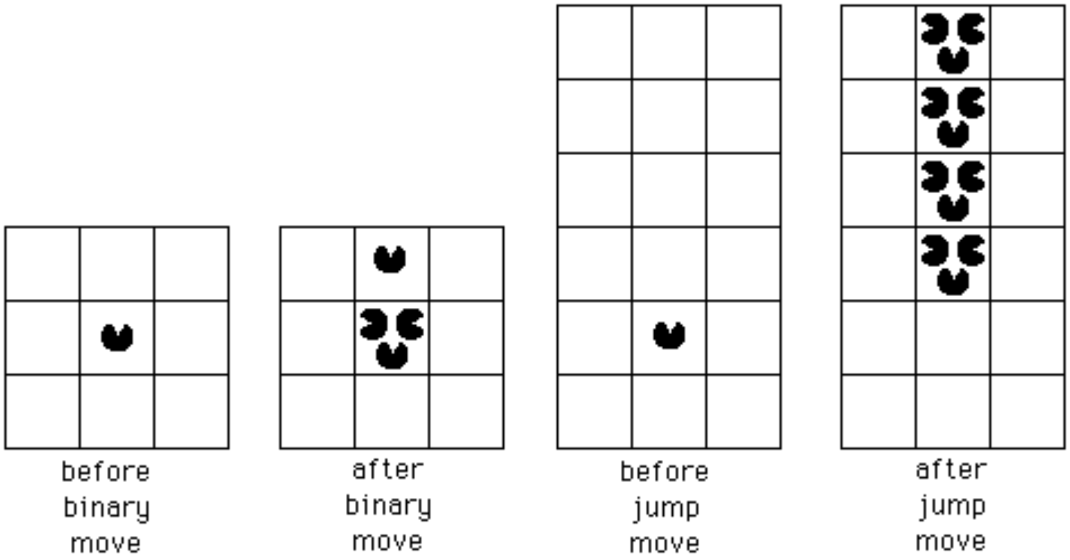


Figure III.4: The functioning of two motor systems. The positions and orientations that are possible after a move are shown for a binary and a jump motor (with power 1 and 4, respectively). Note that different positions and orientations are not equally probable. It is possible for LEE organisms to have more than one motor, in which case the moves are additive.

the organism either accrues (exothermic reaction) or loses (endothermic reaction) energy. If at the end of these reactions the gut is still filled with non reactive atoms beyond its capacity, some atoms are randomly expelled before the next ingestion, leaving a "trace" of the organism's presence in the environment. This can be used as a primitive form of communication among individuals. Furthermore, the gut can function as an important form of "memory" by the use of internal gut sensors (cf. Figure III.3). Some, all, or none of these somatic characteristics may be allowed to evolve in LEE experiments.

Brain

The brain receives input corresponding to the organism's stimuli and produces output controlling its behaviors. We use a well-studied type of neural network as our computational model of an organism's brain [154]. The signals from each sensor are used as the inputs to a feed-forward neural network with zero or more hidden layers of units, and the neural net's output units are used to control the motors (cf. Figure III.2). The network mapping sensory states to motor actions is part of an organism's genome and thus evolves by selection and mutations of the neural net's connection weights, represented as real numbers. Mutations are obtained by adding a random deviate (uniformly distributed in some user-specified interval) to a fraction of the connection weights.

A different way in which behaviors can adapt to the environment, within the lifetime of an individual, is by learning. The role of three mechanisms of unsupervised learning that do not assume the presence of any *external* teacher — learning by prediction of sensory states, by reinforcement, and by parental imitation — are explored in Section III.E.

III.B.5 Implementation

The LEE software tool was developed to be used for efficient simulations within the framework of the LEE model. The execution of the basic cycle of the

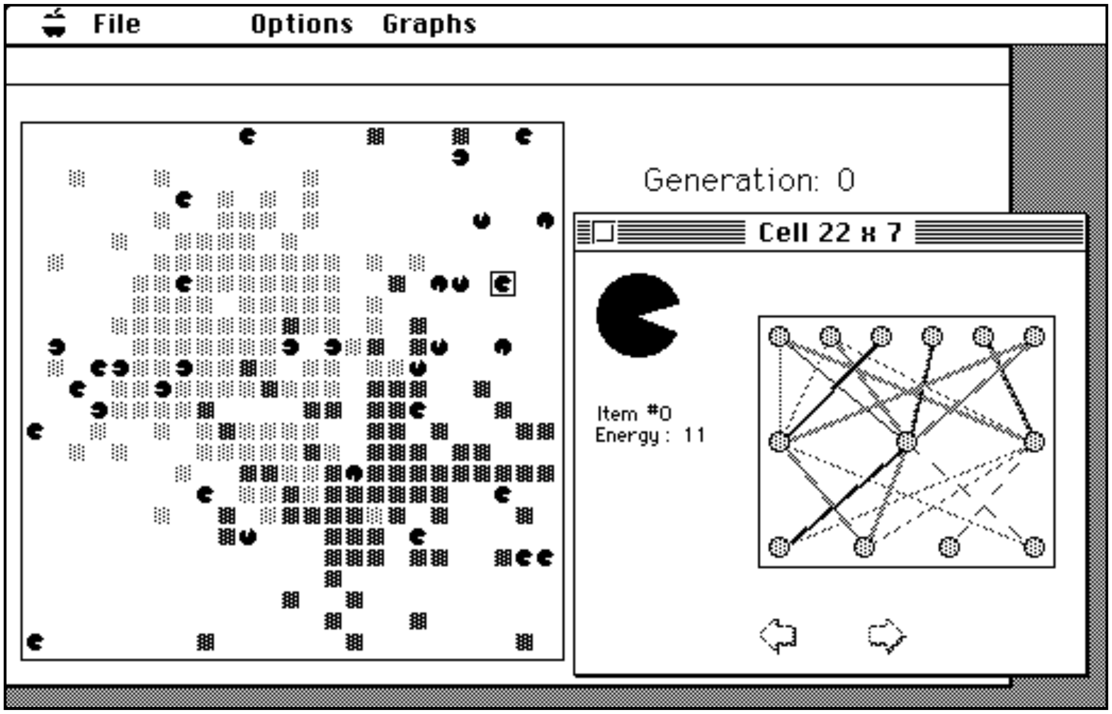


Figure III.5: Graphical interface of the LEE simulator.

local selection algorithm by each agent in the population is an intrinsically parallel process; in our sequential implementation, it is simulated via randomly ordered calls to the agents, to minimize spurious interferences.

The latest LEE release and documentation is available via World Wide Web at URL `http://www.cs.ucsd.edu/users/fil` or anonymous `ftp://cs.ucsd.edu/pub/LEE`. The source code is ©University of California and its use is free, except for commercial purposes. Copies of the software are also available at mirror sites on ALife Online, ENCORE/SAFIER, CMU/AI, and other repositories as well as on CD-ROM freeware.

The LEE package consists of approximately 7,000 lines of C code and runs on both UNIX and MacOS platforms. The documentation contains directions on how to compile LEE for different systems. The only machine dependencies are in the input and output interfaces. While most of the default and user-driven I/O is implemented through files, an interactive graphical interface exists for the

Macintosh. The typical graphical output is shown in Figure III.5. Complete state files allow for checkpointing and continuations of experiments across platforms.

III.C Local selection and carrying capacity

By observing some emerging properties of LEE populations, we show that the task of analyzing the interactions between individuals and their environment is greatly facilitated by relying on well-founded constraints such as the conservation of energy, enforced in local selection models. In this section we illustrate the roles that different measures of fitness (population size, behavior optimality, carrying capacity) play in latent energy environments.

There is a connection between the size of an evolving population and the resources available from its environment. Such resources impose a limit on how many individuals can be sustained. However, different behaviors may result in different efficiencies in using the resources of the same environment. Thus it is the combination of population behavior and environment that characterizes the carrying capacity of that environment. For example, extinction indicates that the carrying capacity is insufficient to support a population large enough to withstand stochastic fluctuations.

LEE allows one to quantitatively estimate adaptedness of behaviors by monitoring the population size throughout an experiment. To see how, let us analyze the relationship between latent energy and population size in a particular example. The experiments reported in the rest of the chapter will make use of analogous derivations to predict and evaluate evolved behaviors. Consider the following reaction energy matrix for a simple environment in which two types of atomic elements, A and B, are distributed uniformly:

$$\begin{array}{c|cc} & A & B \\ \hline A & -\beta E & E \\ B & E & -\beta E \end{array} \tag{III.2}$$

where E and β are environmental parameters subject to the constraints:

$$\begin{aligned} E &> 0 \\ 0 &< \beta < 1 \end{aligned} \tag{III.3}$$

and there are no by-products. Matrix III.2, along with the spatio-temporal distributions of the elements, determines the environmental complexity. Conditions III.3 make the environment non-zero-sum. The minimal set of elements that can be found in the environment and completely transformed into energy by alternative behavioral strategies is $\{A, A, B, B\}$. In fact an organism, assuming its sensory apparatus can perfectly discriminate between these elements, can catalyze the following sets of reactions:

$$(A + A), (B + B) \rightarrow -2\beta E \tag{III.4}$$

$$(A + B), (A + B) \rightarrow +2E \tag{III.5}$$

where strategy III.5 is clearly more advantageous than III.4, given the conditions III.3. If ν is the expected rate of replenishment for both A and B elements, then a set $\{A, A, B, B\}$ is produced every $2/\nu$ cycles. Thus strategies III.4 and III.5 produce energy changes

$$\Delta E_- = \frac{-2\beta E}{2/\nu} = -\nu\beta E \tag{III.6}$$

$$\Delta E_+ = \frac{2E}{2/\nu} = \nu E \tag{III.7}$$

per unit time (cycle), respectively. Let us now consider the situation at equilibrium. On average, two conditions are verified: first, by definition, the population size p remains constant; second, energy is consumed by organisms in the population at the same rate at which it is produced (otherwise the population size would change). Using $\langle\cdot\rangle$ to indicate time averages, we can write:

$$\langle \Delta p \rangle = 0 \tag{III.8}$$

$$\langle \Delta E \rangle = 0. \tag{III.9}$$

Since energy is always conserved, the only consumed energy is that lost in the form of work, that is, used for moving in the world. At any time step, all individuals get to make a move. For simplicity we set the cost of all moves equal to a constant that we call c. Then, using Equation III.8, the average energy used by the population per unit time is pc.

To calculate how much energy is produced, we must know the strategy used on average by the population for combining elements. In other words, we need to determine how efficiently the latent energy contained in the elements is transformed into usable energy. It is important to note that resources cannot build up forever in the environment, otherwise by statistical arguments the equilibrium would become less and less stable. Therefore, by choosing an appropriate time unit, we can make the additional assumption (verified in every simulation) that elements are transformed into energy at the same rate as they are replenished by the environment. Strategies III.4 and III.5 provide the connection between elements and produced energy. We then introduce a probability distribution over these two strategies: let η be the probability of strategy III.5, so that $(1-\eta)$ is that of strategy III.4. The average energy produced per unit time by the population is obtained summing the energy changes of Equations III.6 and III.7, weighted by the corresponding probabilities:

$$\eta \Delta E_+ + (1-\eta)\Delta E_- = \nu[\eta E + (\eta - 1)\beta E]$$

so that we can finally rewrite Equation III.9 as

$$\langle \Delta E \rangle = \nu E[\eta + \eta\beta - \beta] - pc = 0. \tag{III.10}$$

Equation III.10 provides the link between efficiency of behavioral strategies, expressed through the probability distribution over catalyzed reactions (η), and population size (p). The remaining variables (ν, E, β, c) are environmental parameters. If η is known, we can solve Equation III.10 for the expected population size:

$$p = \frac{\nu E}{c}(\eta + \eta\beta - \beta). \tag{III.11}$$

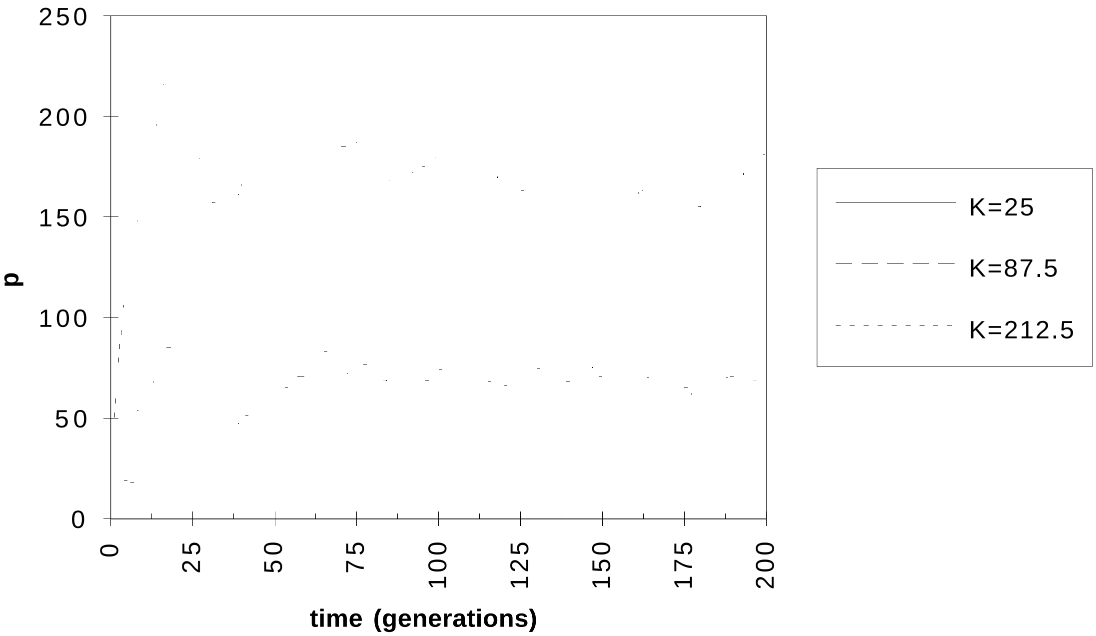


Figure III.6: Population dynamics in three different environments. The lowest carrying capacity is such that stochastic fluctuations bring the population to extinction. The two more benign environments allow the population to quickly reach the different carrying capacities corresponding to the same simple (random) behavior.

In particular, the case of optimal behavior, $\eta = 1$, corresponds to the maximum sustainable population:

$$p_{max} = \frac{\nu E}{c}. \tag{III.12}$$

The converse case is useful for estimating the optimality of a population's behavior at equilibrium, by measuring the population size and solving Equation III.10 for η:

$$\eta = \frac{\frac{pc}{\nu E} + \beta}{1 + \beta}. \tag{III.13}$$

As an illustration of this simple analysis, in Figure III.6 population size is plotted versus time for three simulations with different entries (β and E values) in reaction matrix III.2. No evolution is allowed in these runs. Disregarding the first damped oscillations (due to an initial abundance of atoms) and the following noise fluctuations, we readily observe a steady-state regime in which the population size depends on the rate at which energy is introduced into the world. Simulations are labeled by the quantity K, which is the carrying capacity for a population of individuals with random behaviors (obtained from Equation III.11 with $\eta = 1/2$):

$$K = p_{\eta=1/2} = \frac{\nu E}{2c}(1 - \beta). \tag{III.14}$$

We can use these equations to make predictions about the outcomes of the simulations, and to compare behaviors in the different environments. The smallest K corresponds to a random behavior population smaller than the amplitude of the fluctuations, so extinction occurs rapidly. Larger K values result in different stable population levels. Substituting the measured population sizes into Equation III.13, we find that the corresponding strategies are not significantly different ($\eta \approx 1/2$, or random behavior, in both cases). Therefore the observed difference is to be attributed to the different environments, rather than to different behaviors.

Of course, the linear relation III.10 holds only at equilibrium and for this simple example: the more general non-equilibrium case and more complex environments may yield systems of differential equations that are difficult to solve analytically. In the remainder of this chapter, environments will be designed so as to keep their analytical characterization simple. We want to stress that carefully designed latent energy environments allow us to maintain an accurate connection between environmental and behavioral complexity.

Since individuals interact solely by sharing finite environmental resources, fitness is density dependent and thus affected by the size of the population. Under such conditions, carrying capacity is commonly considered in biology to be the best measure of fitness [168]. We now want to point to the strong role of the environment under density dependent fitness, by showing that the carrying capacity of certain environments can be predicted very reliably from knowledge about the environment alone, without considering behaviors. One such environment, even simpler than the one described in matrix III.2, is easily characterized: assume that there is just one element with, say, uniform distribution and rate of replenishment ν; let each

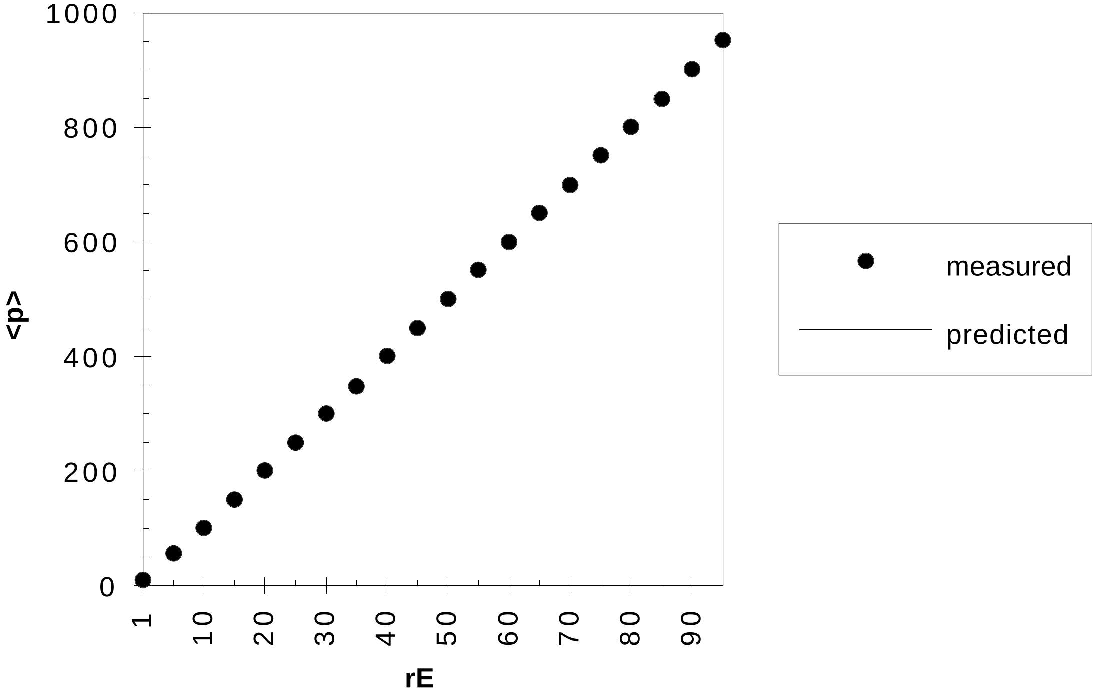


Figure III.7: Correlation of population size with environmental carrying capacity. Measures of population size are time averages in the steady-state regime, for the first stable behavior; the predicted population sizes are calculated from Equation III.15 for random behaviors. The x-axis spans a wide range of environmental parameterizations.

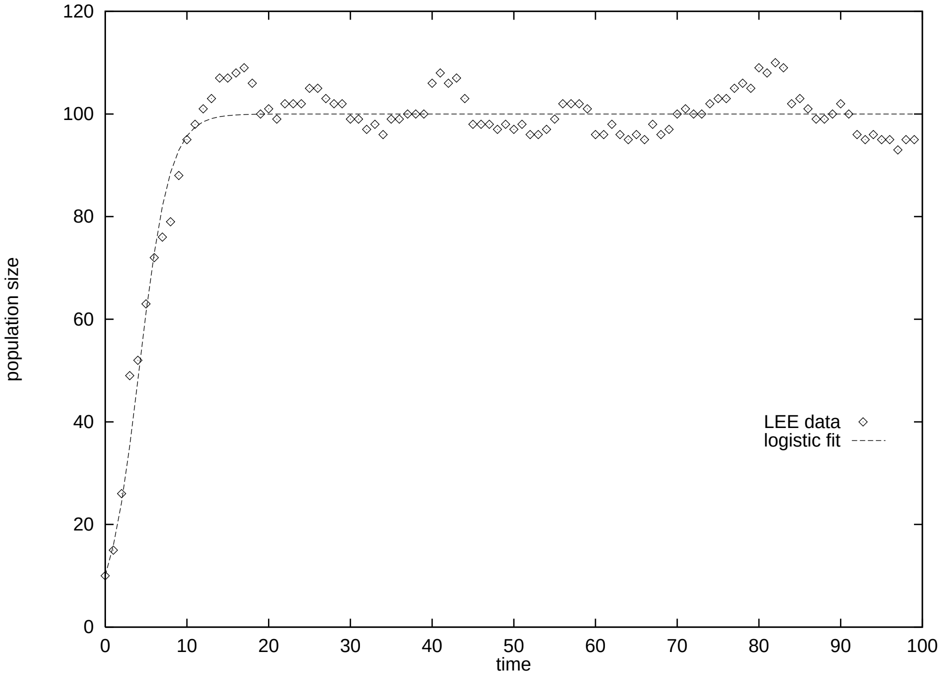


Figure III.8: Least-squares fit of LEE population dynamics to the logistic growth model, with carrying capacity $K = 100$ derived from Equation III.15 and initial population size $p_0 = 10$. The growth rate of the best fit is $r \simeq 0.53$. The fit between data and model is excellent (r-squared 0.9357, $p < 0.0001$).

atom be associated with energy E (again, no by-products).[3] Equation III.8 still holds at equilibrium, and Equation III.9 for energy conservation yields in this case:

$$K \equiv p = \frac{\nu E}{c} \tag{III.15}$$

where the carrying capacity K corresponds to the size of the population, whatever the behavior. Equation III.15 is different from Equation III.11 in that there is no independent variable connected with behavioral effects. The prediction, by Equation III.15, of perfect correlation between population size and environmental carrying capacity (parameterized by νE) is confirmed by the measures illustrated in Figure III.7.

---

[3] In the LEE simulator, the oxymoron "unary reaction" refers to the utilization of energy from a single atom, without the need for combinations with other atoms.

Given how well we can analytically predict the carrying capacity of a latent energy environment, based on its reaction matrix and temporal distribution of resources, it is natural to ask whether the local selection algorithm yields such a good individual-based model of K-selection, or density dependence, as we have claimed. At the population level, density dependence is analytically modeled in the continuous case by the logistic growth equation:

$$\frac{dp}{dt} = rp(1 - \frac{p}{K}) \tag{III.16}$$

whose solution has the form

$$p = \frac{K}{1 + (\frac{K}{p_0} - 1)e^{-rt}} \tag{III.17}$$

where r is the growth rate and p_0 is the initial population size [83]. To see how well LEE models logistic growth we have fitted the population dynamics of a run from Figure III.7 to Equation III.17. We use the carrying capacity from Equation III.15 in Equation III.17, so that the growth rate r is the only free parameter in the fit. As Figure III.8 clearly shows, the fit is excellent; LEE is indeed a very good model of density dependent selection.

It is reassuring to know that even though the individual agents in LEE act independently of each other and without any centralized control, their sharing of an environment with finite-resources results in the well-known logistic growth model. However, we have seen only one half of the picture — behaviors in the above environment are inconsequential. When organisms are allowed to evolve behaviors that may use the latent energy more efficiently than by a random walk, an increase in average age and in population size is observed. The situation is illustrated in Figure III.9, where measures are from LEE simulations with reaction matrix III.2. During these non-equilibrium phases Equation III.8 does not hold, and the environment is no longer sufficient to predict carrying capacity. Population size, on the other hand, can still be used as a fitness measure at the population level. Its increase is due to the evolution of better behaviors, that bring η from 0.5 to approximately 0.73 (from Equation III.13).

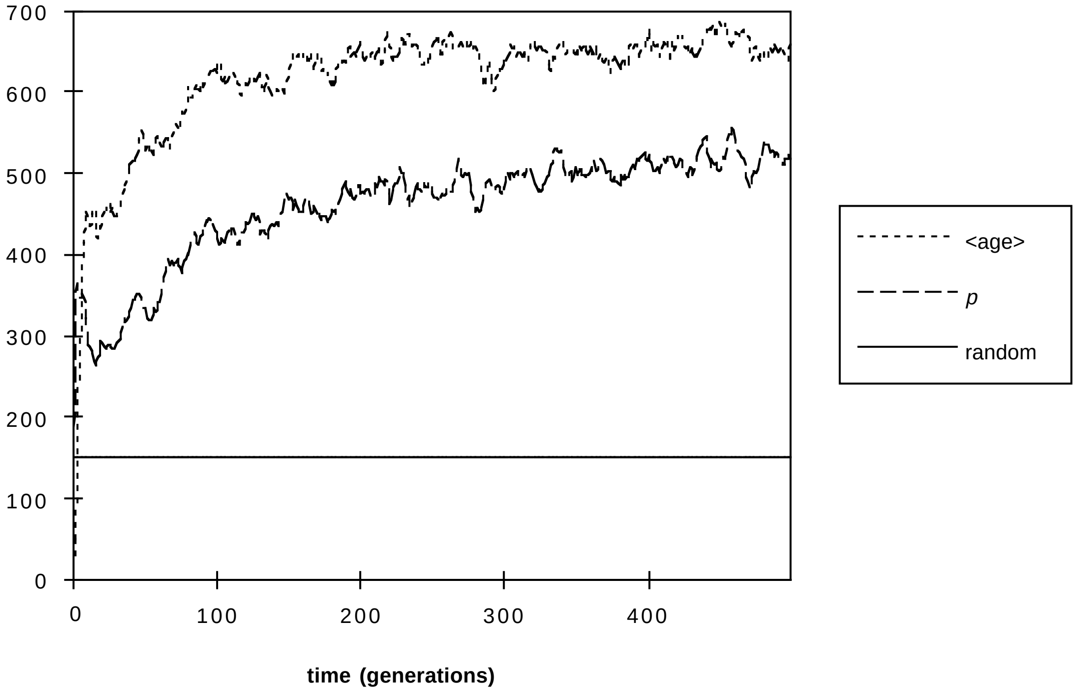


Figure III.9: The role of behavior in environmental carrying capacity and fitness. Here Equation III.12 yields $p_{max} = 900$, whereas the size of a population with random behaviors (from Equation III.14) is 150 (marked “random”). Average age is also shown to be correlated with population size in this experiment (age units are cycles).

Just as η in the simple environments described by III.2 can vary between 0.5 and 1, so p can — in any latent energy environment — range between some low level, corresponding to the carrying capacity of random initial behaviors, and some maximum carrying capacity corresponding to optimal behaviors. It turns out that population size is well correlated with both age and *expected reproductive success,* a fitness measure used in biology under density dependence. Therefore we are justified in considering population size our (dependent) measure of fitness in LEE, and in using it to analyze the interplay between environments and behaviors of evolving populations.

III.D From complex environments to complex behaviors

In this section we outline the main results of three LEE experiments. Each experiment explores a different dimension of environmental complexity, by a set of environments designed to characterize and vary such complexity. We then analyze the evolving population to find the ways in which it responds and becomes adapted to the different complexities. These experiments are aimed at identifying the degree to which the environment can create and shape the selective pressures driving the adaptive process. We first focus on the evolutionary aspect of adaptation, considering the internalizations of environmental complexity into agent behaviors that can be achieved by the local selection algorithm alone; learning will be considered in Section III.E.

Noise plays an important role in the LEE simulator, and this is reflected in stochastic fluctuations of all the monitored variables. By repeating a simulation for several runs, each with different initial conditions, we can average out such noise and find statistically significant effects. The results reported here are all found to be statistically significant across runs, but will be visualized more easily by plotting results of single runs.

III.D.1 Patchiness

Experimental setting

In the first experiment, the latent energy of the environment is as described in the reaction matrix III.2 of the previous section. Since the spatial dimension is crucial for environmental characterization, we have explored this dimension by observing the response of an evolving population to variations in the *spatial* distributions of elements in the world.

The environment is replenished with atoms of both elements at a constant rate and according to pseudo-gaussian distributions along the two spatial axes. We obtain a pseudo-gaussian probability density in LEE by adding one or more uniform probability densities on some interval along each axis. The more of these uniform probability densities are added, the more peaked the resulting distribution (i.e., the smaller its variance). This is a standard statistical method, derived from the central limit theorem.

We consider four such environments of increasing patchiness, each obtained by adding between one and four uniform probability densities, for each element and for each axis. These are shown in Figure III.10. This LEE world is 25 by 25 cells wide; however, due to its toroidal edge conditions, the environment appears as arbitrarily large and periodic. The patch centers of the two elements are offset diagonally. The first, uniform environment has no spatial structure and thus no patch size; in this sense, it is the simplest. The patchy environments have increasing spatial structure, which we consider a source of complexity. The patches overlap largely in the second environment, less in the third, and are completely separated in the fourth and last environment. Therefore the patch size decreases from the entire world size to about half of the world's diagonal.

To signal the presence of either element in its neighborhood, each organism has two ambient sensors, oriented forward (cf. Figure III.3). Like all ambient sensors used in this section, they have a range of 5 cells. By comparing the two

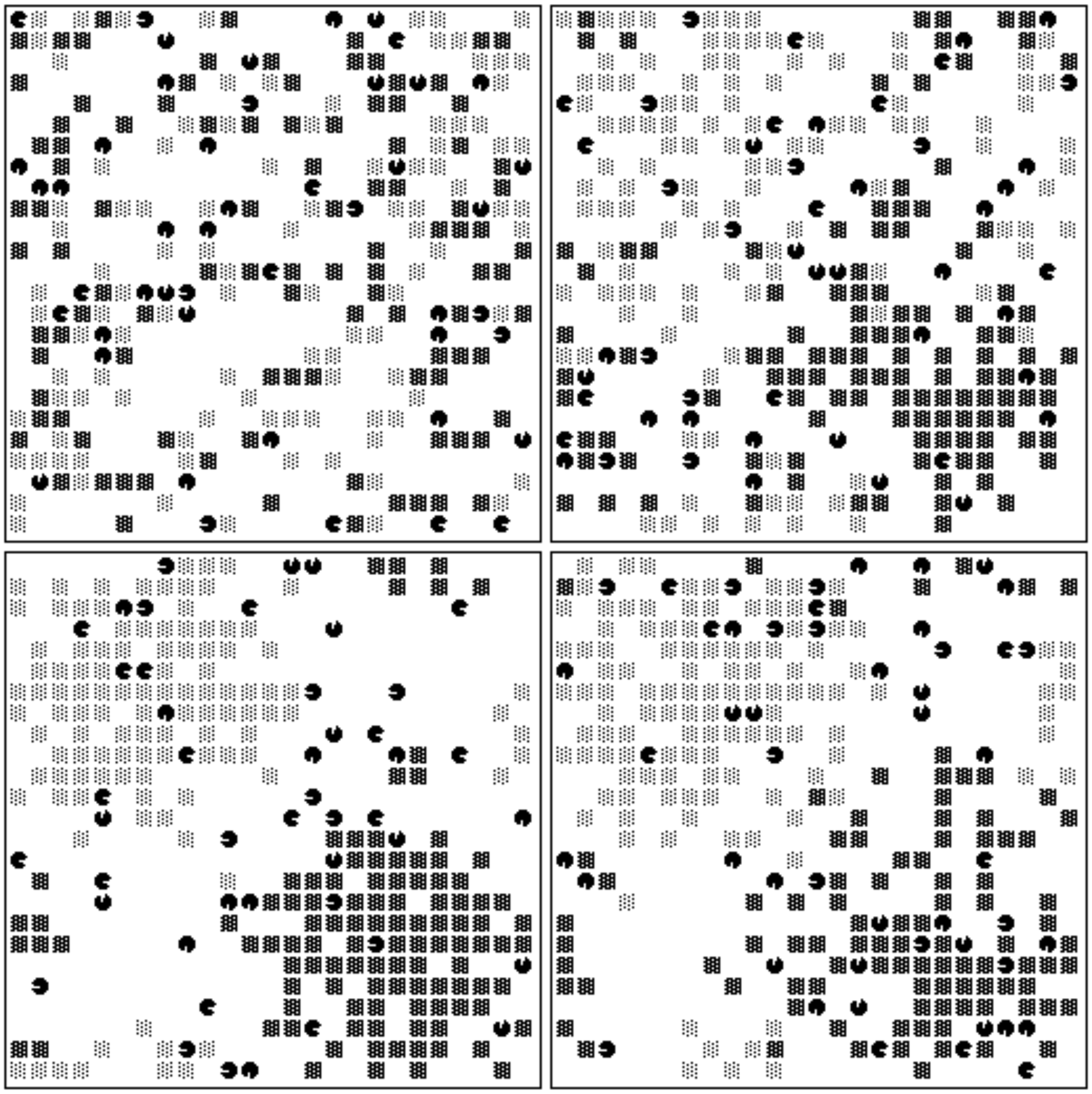


Figure III.10: Snapshots of four environments used for the patchiness experiment. They are ordered (clockwise from top left) according to increasing patch complexity: from the simplest case of uniform element distributions to the most structured environment with sharply separated element patches. A and B atoms are marked as squares of different shades of gray.

Contact sensors	Gut sensors	Jump motor
A	A	Not nearest
B	A	Nearest
A	B	Nearest
B	B	Not nearest
Any condition	Empty	Any move

Table III.1: Behavior of maximum energy efficiency for uniform environments in the patchiness experiment. No cell contains both A and B atoms because they would react together, so that contact signals cannot be ambiguous. "Nearest" means jumping to the cell immediately facing the organism, i.e., the one sensed by contact sensors.

signals, an organism can discern whether it is located closer to an A or B patch. There are also two contact sensors, identifying the content of the cell facing the organism. Finally, two gut sensors make it possible to compare external patch signals with previously ingested atoms and decide whether they are of the same or different elements. The four sensors provide input to the neural net, which has six hidden units and one output. The latter drives a single jump motor (cf. Figure III.4). Like all jump motors used in this section, this has power 10 — an agent can jump a distance of at most 10 cells.

Given this representation, adaptive individuals can evolve behaviors yielding the different carrying capacities obtained by Equation III.11. The parameters (ν, E, c) of the simulations are set in such a way that a population with random behaviors, catalyzing all reactions with equal probabilities, has a size $p_{\eta=1/2} = 50$ (from Equation III.14). Given the characterization of the environments in this experiment, it is easy to see that the most efficient behavior consists of jumping far from patches of the same element as an atom in the gut, and otherwise foraging with small jumps: this way exothermic reactions are catalyzed. The carrying capacity of this collective behavior is $p_{max} = 200$ (from Equation III.12).

Up to a point, patch size of the environment determines how complex such efficient behavior can be. In uniform environments, contact sensors alone

Ambient sensors	Gut sensors	Jump motor
$A > B$	A	Far
$B > A$	A	Near
$A > B$	B	Near
$B > A$	B	Far
Any condition	Empty	Any move

Table III.2: Behavior of maximum energy efficiency for clustered environments in the patchiness experiment. Hidden units must perform the comparisons between ambient signals. "Far" and "near" mean jumping distances greater or smaller than patch size, respectively.

provide all the information needed to implement such behavior, as shown in Table III.1. No patch size information is available, because there are no patches. Nothing is to be lost from jumping too far or too near, as long as endothermic reactions are avoided. In environments with spatially clustered elements, on the other hand, space is more structured and thus more complex. The population needs to evolve an "understanding" of environmental structure (e.g., patch size) to gauge appropriate moves. Organisms need to make use of the more ambiguous information provided by ambient sensors — and perform preliminary comparisons between their signals — to decide how far to jump (recall that the direction of a jump cannot be decided). Due to the two additional requirements (comparison of inputs and gauging patch size) this is a more complex behavior, as illustrated in Table III.2.

Of course, the complexity of the task would be greatly affected by changes in the sensors, as discussed in Section III.E.1. With the given sensory system, however, such is the direction in which we expect environmental complexity to drive the evolution of adaptive behaviors in this experiment. More complicated spatial organizations (e.g., checker-board patterns and other layouts with even higher spatial order statistics) can reasonably be expected to require still greater sophistication from individuals' cognition.

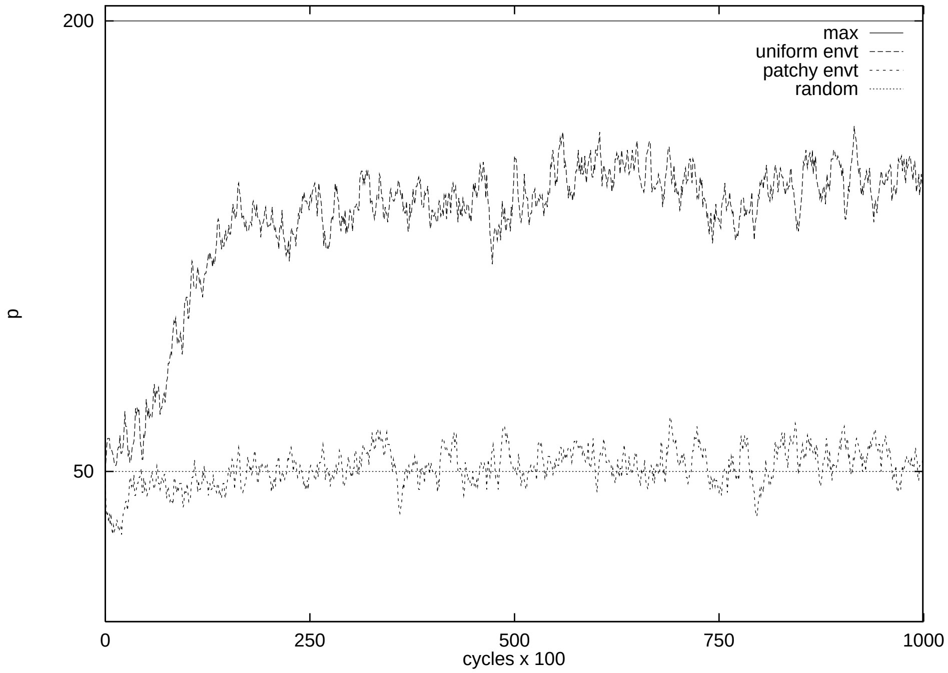


Figure III.11: Population size for two fixed environments. These are the first (uniform) and the last (most patchy) worlds of Figure III.10. The predicted carrying capacities corresponding to random and most efficient behaviors are also shown.

Results

We have run four simulation with different environmental conditions. The first two simulations have stationary environments, whose complexity remains constant through evolutionary time. The last two have changing environments, whose complexity increases in different ways.

Populations adapting in the two fixed environments reach carrying capacities shown in Figure III.11. The patchy environment is more complex and the population is unable to evolve any behavior superior to the random one, as seen by comparing population size with the predicted carrying capacity of the random behavior. The uniform environment is more benign and (a good part of) the population can evolve the more efficient behavior of Table III.1, resulting in a carrying capacity about three times as large ($\eta \sim 0.83$ from Equation III.13). The residual selective pressure does not appear sufficient for convergence to the optimal strategy

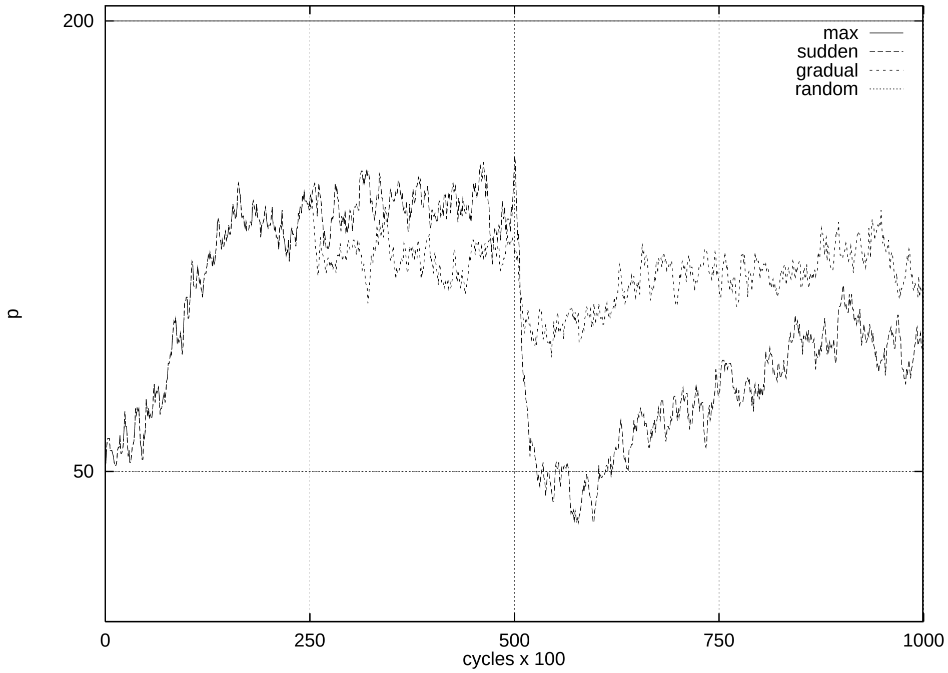


Figure III.12: Population dynamics for two changing environments. The “sudden” change is a single transition, at 50,000 cycles, from the uniform to the most patchy worlds of Figure III.10. The “gradual” change is made of three transitions, every 25,000 cycles (see vertical grid), through all of the four worlds of Figure III.10 in order of increasing patchiness. The predicted carrying capacities corresponding to random and most efficient behaviors are also shown.

within the observed evolutionary time, i.e., to push the population to its maximum size.

Figure III.12 illustrates what happens to the populations evolving in the two environments of increasing complexity. When the environment suddenly becomes very patchy, the population continues to apply the evolved behavior of Table III.1 with catastrophic consequences; the population size decreases to the random level. Since the individuals are essentially behaving the same way right before and right after the environmental transition, we could say that the carrying capacity of the environment has dropped. However, the population then grows again, becoming significantly higher than random ($\eta \sim 0.63$). This is evidence that (part of) the population is able to evolve a behavior similar to the one of Table III.2. In the simulation where the environment becomes gradually more patchy, adaptation is not so catastrophic. There are small performance degradations that are quickly offset by the evolution of improved behaviors, so that finally the population can cope quite well with the most patchy environment — over doubling with respect to the population starting in the patchy environment ($\eta \sim 0.7$).

Discussion

The above results demonstrate, in the first place, that patchiness can provide a useful measure of environmental complexity. Further, spatial heterogeneity affects evolved behaviors. A uniform world is simple because there is nothing to be learned about different parts of it: it is the same everywhere. Here, as shown by Figure III.11, the simple behavior of Table III.1 is discovered rapidly by the evolving population. Conversely, a patchy world is complex because of its spatial structure. If such complexity is too large for the population to make sense of it, then no behavior evolves to exploit its resources better than by moving at random. The evolutionary leap from random behavior to the strategy of Table III.2, in terms of individual neural networks, is much larger than to the strategy of Table III.1. We conclude that in the most patchy environment, the selective pressure away from the random behavior is too small to discover more complex behaviors within the observed evolutionary time.

Simpler environments have additional advantages for the adaptive individuals. As Figure III.12 indicates, behaviors adaptive in the more complex environments can indeed be evolved but only if the population has previously adapted to less complex environments. Thus a gradual increase in environmental complexity generates an adaptive response in the collective behavior evolved.

If the environment undergoes a large increase in complexity, any advantage of previously evolved behaviors seems lost: after the sudden transition occurring at 50,000 cycles, the population goes down to the random level, evidence that the simple behavior appropriate for the uniform environment (cf. Table III.1) is not adaptive in the patchy environment. This catastrophe is only illusory, however: the fact that population size grows again in the following phase represents evidence that the more complex behavior (cf. Table III.2) has evolved. This did not happen when the environment was patchy to start with (Figure III.11); the adaptive behavior evolved in the milder environmental phase therefore provides an evolutionary stepping point. This phenomenon is sometimes referred to as *pre-adaptation.*

III.D.2 Mutualism

Experimental setting

The second experiment explores the role of environmental complexity in enforcing ecological balance. The environment permits the population a more robust survival when this is structured in such a way to maximize its *biodiversity.* The population is divided into two "species" distinguishable by a genetic marker; parents pass their species gene to offspring. Some degree of ecological stability might be provided if it were possible for this genes to mutate occasionally. However, in this experiment we are interested in the conditions under which biodiversity is maintained by environmental pressures alone. Therefore the species gene is not mutated. Since reproduction occurs asexually by cloning, species are completely determined by phylogeny. Let us call a and b the alleles of the species gene. Let us further assume that the two species have different metabolisms, given by the following *mutualistic* reaction matrices:

$$\begin{array}{c|cc} a & A & B \\ \hline A & E+2B & \\ B & & -\beta E \end{array} \tag{III.18}$$

$$\begin{array}{c|cc} b & A & B \\ \hline A & -\beta E & \\ B & & E+2A \end{array} \tag{III.19}$$

for species a and b respectively. Both A and B elements are uniformly distributed and replenished at the same, low rate. In this experiment, the world can fill up with unused elements. If this occurs, replenishment of those elements is suspended.

Since different elements are never reactive, and conditions III.3 still hold, all individuals have only two (non-exclusive) possible actions. This simplifies the range of behaviors that we can explore in this experiment. The first action is to ingest atoms of the element whose symbol is the (capitalized) letter of their species allele, and gain energy; the second possibility is to ingest atoms of the other element, and lose energy. Notice, however, that in the former case there are by-products which may still react in the gut and consume part of the energy that was acquired with the original action. This will happen half of the time on average, due to the digestion algorithm (cf. Section III.B.4); gut contents are shuffled, and then reactions occur in the order determined by a single pass through the atoms in the gut. The effect is easily accounted for in the computation of environmental carrying capacities.

The "optimal" (most efficient) behavior requires that species a and b catalyze exothermic reaction by foraging for A or B atoms, respectively, and avoid other atoms by turning away. Note that agents do not have access to their species markers, so they have to internalize this information from their interactions with the environment. The consequences of the optimal collective behavior are conditional upon the biodiversity of the population; in fact, if one species goes extinct, the optimal carrying capacity halves because the remaining species can no longer use the other's by-products. The random behavior is to always move ahead, irrespective of input, and eat every atom along the way. Biodiversity has no effect on a population with random behavior.

The representation of individuals is quite simple. Each organism has two contact sensors signalling the presence of A or B atoms in the facing cell. Its neural net has four hidden units and two outputs driving a single binary motor (with power 1). The gut's size is 1 — that is, it can hold at most one previously ingested atom between actions. Since only same-element atoms react, there is no need of gut sensors.

Results

We have run two simulations for this experiment, using β to model two environments of different complexity. In the first "easy" environment, $\beta = 0.1$. The parameters are such that the carrying capacity achievable in this environment by the optimal collective behavior described above is $p_{max} = 95$. The population cannot go beyond this limit. This environment is quite benign and thus there is small pressure to push the population above carrying capacity corresponding to the random behavior ($p_{\eta=1/2} = 42.75$). In fact, as shown in Figure III.13, the population does not go beyond such level within the observed 50,000 cycles.

In the second "hard" environment, $\beta = 0.5$ and thus the situation is more harsh. The maximum carrying capacity of this environment, for an optimal population, is $p_{max} = 75$; the random carrying capacity is $p_{\eta=1/2} = 18.75$. The ratio between the former and the latter is almost twice as large as in the easy environment, therefore there is larger selective pressure toward more efficient behaviors. Figure III.14 shows the population dynamics in this simulation. After 50,000 cycles, the population has evolved a collective behavior consistently superior to the random one ($\eta \sim 0.69$).

A more interesting statistics in this experiment is the biodiversity of the population, based on the species gene. We have plotted in Figure III.15 the evolving biodiversity of the two simulations with easy and harsh environment. In the former the biodiversity goes to zero, which is an absorbing state due to the lack of mutations on the species gene. Therefore one of the two species has gone extinct for good. In the latter, however, the biodiversity is maintained at its maximum

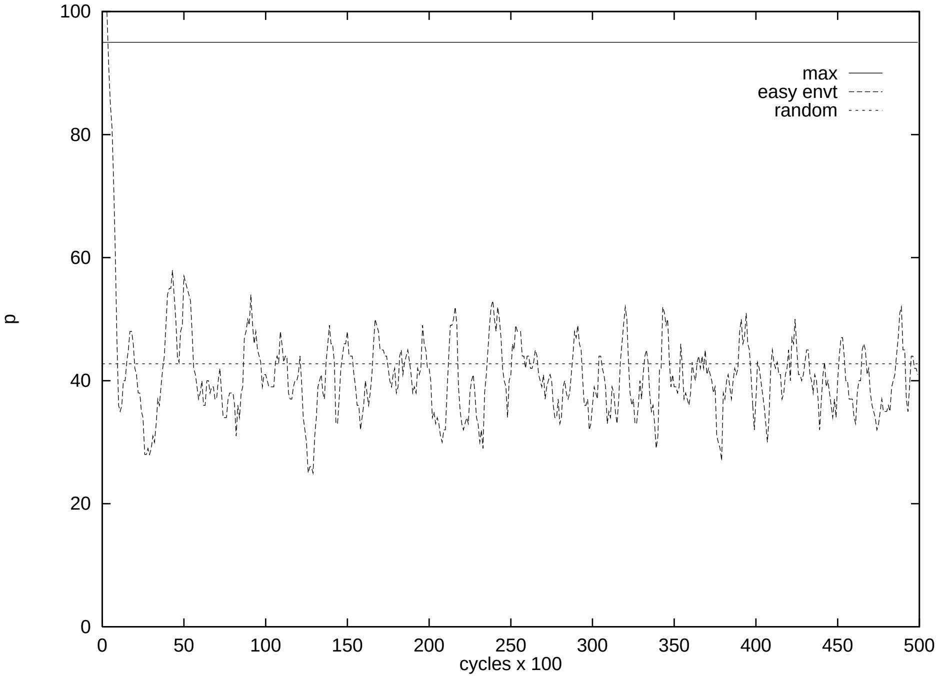


Figure III.13: Population size in the easy ($\beta = 0.1$) environment. The predicted carrying capacities corresponding to random and most efficient behaviors are also shown.

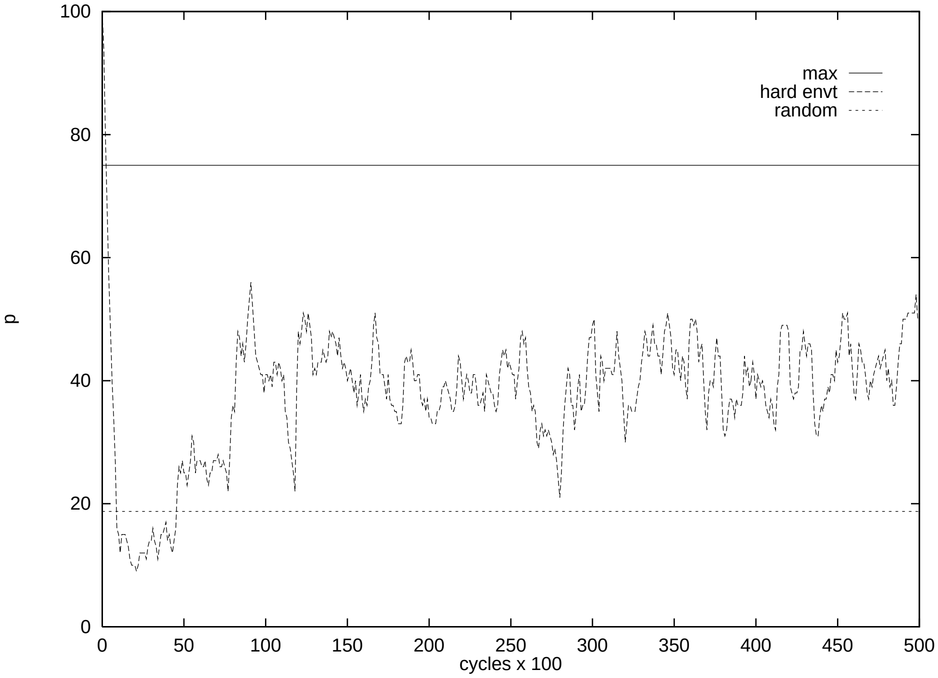


Figure III.14: Population size in the hard ($\beta = 0.5$) environment. The predicted carrying capacities corresponding to random and most efficient behaviors are also shown.

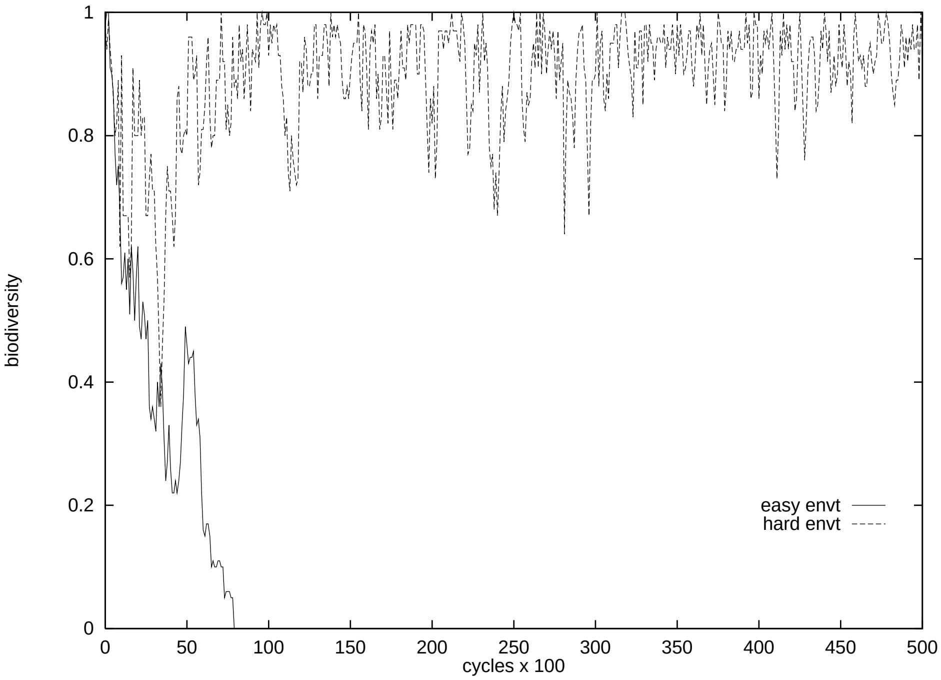


Figure III.15: Biodiversity of the population in the two environments of the mutualism experiment. This is measured based on the species gene and normalized so that 1, the maximum value, means that half of the total population belongs to each species; 0 means that all individuals belong to one and the same species, while the other has gone extinct.

(not an absorbing state) showing perfect balance between the two species thanks to their capability to consume and replenish each other's resources.

We have arbitrarily killed the biodiversity in a variant of the hard environment simulation, by setting all species genes to the same allele after 50,000 cycles. This a poor man's model of some "environmentally unsound" external action, like an oil spill or some other disaster causing the fracture of the food chain. Figure III.16 shows that the consequence is catastrophic; complete extinction follows swiftly.

We have extended the simulation in the harsh environment to compare the consequences of self-preserved vs. killed biodiversity on population dynamics, but also to see whether the population size can further increase toward the optimal

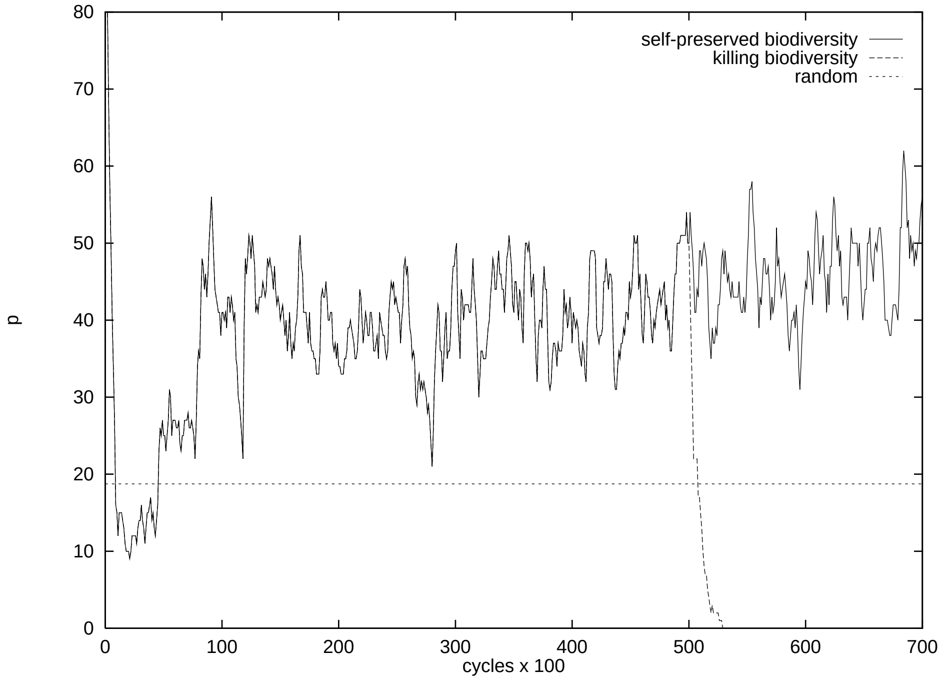


Figure III.16: Population size in part of the extended simulations with the hard environment (cf. Figure III.14). The normal case with evolving biodiversity is compared with the one in which the biodiversity is killed after 50,000 cycles (see text).

level if given sufficient evolutionary time. The result of a 500,000 cycle-run (shown only in small part in Figure III.16, for clarity) is a very slow improvement that eventually reaches a population size over 2.5 times larger than the random behavior level ($\eta \sim 0.78$).

Discussion

This experiment also shows how the environment creates the selective pressures for the evolution of adaptive behaviors. Environmental complexity is measurable here by the ratio between the energetic outcomes of the different reactions. This heterogeneity of selective pressures has a direct impact on the carrying capacity of random behaviors: the more complex the environment, the harder to survive without adaptation.

When the environment is so benign that the random behavior has a carrying capacity close enough to that of the optimal behavior, there is not enough pressure for organism to improve their efficiency. Biodiversity in this experiment is not a genetic trait that can be selected for; it is a collective property of the population. Therefore it is subject to genetic drift in the course of evolution. Since zero is the only absorbing state — one from which the population cannot escape — the biodiversity drifts to zero after about 8,000 cycles, as shown in Figure III.15. From this moment on, all individuals are of the same species, so only one element can provide positive energy. The carrying capacity of the optimal behavior, as it turns out, decreases to only about 11% higher than the random one, explaining why nothing else happens (Figure III.13).

Harsher environments make a random population smaller. Just like too much environmental complexity prevented evolution in the patchiness experiment, too large a value for β would lead to extinction in this experiment. A moderate increase in selective pressure, however, causes better than random behaviors to evolve, as Figure III.14 illustrates. At least part of the population catalyzes exothermic reactions, creating by-products that help the other species. This collective behavior, requiring the presence of both species, is effectively a form of mutualism and increases the carrying capacity thanks to the extra energy from by-products. This indirectly pushes biodiversity to a maximum, as it happens (Figure III.15): each species needs the other.

In other words, we see two selective pressures reinforcing each other. The presence of individuals with efficient foraging behaviors in a certain area creates selective pressure in favor of individuals of whatever species is less frequent in that area (they have more food available), thereby increasing biodiversity. The ecological balance between species maintains carrying capacity at its maximum and thus creates selective pressure in favor of individuals with efficient foraging behaviors. This “selective loop” occurs locally in the environment, mediated by the resources, yet we observe a larger role of the individuals locally shaping each other’s environment.

Once this coupled adaptive process is “bootstrapped” by a sufficiently complex initial environment, it gives rise to behaviors that depend upon the diverse ecosystem. In fact, as Figure III.16 shows, the biodiversity annulment has a much more catastrophic consequence in the complex environment — extinction. This phenomenon is due to the population no longer behaving randomly; each species relies on the existence of the other for providing the resources that can sustain its size. When those resources disappear, the carrying capacity of the environment drops dramatically and the violent population fluctuation that follows leads to extinction. Therefore the environment determines the necessity of mutualism and biodiversity. Ecological balance can provide for robust adaptation in complex environments.

III.D.3 Seasonality

Experimental setting

In the third experiment, we consider continuously changing environments. Under such conditions it makes no sense to speak of stable carrying capacity in the sense of Section III.C; instead, we expect environmental fluctuations to drive the adaptive process in an equally continuous fashion. In particular, we want to model seasons and therefore choose environments characterized by periodic, sinusoidal patterns of change. To this end, consider a world with two uniformly distributed elements, with constant replenishment rate, that give off energy as follows:

$$A \rightarrow E_A = E_0 + \lambda \sin(2\pi t/\tau) \qquad \text{(III.20)}$$

$$B \rightarrow E_B = E_0 - \lambda \sin(2\pi t/\tau) \qquad \text{(III.21)}$$

without need of binary reactions and without by-products. As in the previous experiment, if the world fills up with unused elements, replenishment of those elements is suspended.

Here t represents time (measured in cycles) and τ is a seasonal time constant that we will assume to be fixed throughout a simulation. We can think of the elements as two food sources, e.g., fruits, whose caloric contents vary with the passing of seasons, one reaching its high peak in the summer and the other in the winter. Note that if $\lambda > E_0$, each food becomes “poisonous” in its bad season. We are interested in observing how the population adapts to the seasons when either environmental resource alone is insufficient to guarantee survival.

Each individual has two contact sensors, signalling A and B atoms respectively in the facing cell. Because of the “unary” reactions, there is no need for gut sensors. There are two hidden units and two outputs driving a binary motor. As in the previous experiment, ignoring input leads to the random behavior of always moving forward, eating all atoms. We can easily derive a “time-averaged” carrying capacity for this behavior and the seasonal environment, in the same form of Equation III.15, but setting the value of E as follows:

$$E = E_A + E_B = 2E_0. \tag{III.22}$$

This behavior, however, neglects the complexity of the environment provided by the seasonal fluctuations.

At any given moment (except the instants when $E_A = E_B = E_0$) one element is more energetic, so one possible action would be for individuals to avoid the other element. As long as the other element remains exothermic, this is a less efficient behavior than the random one; but as soon as the other element becomes endothermic, such poison-avoidance strategy is a winner. It requires, however, that organisms be able to “track” the environment to tell the energetic content of an atom before ingesting it, a capability that sensors do not provide in our experiment. It is instead possible for the population (or a part of it) to converge on a behavior avoiding one of the two elements (say, B, the one that first goes through winter). This (sub)population would have a carrying capacity again given by Equation III.15, but with $E = E_A$. The rest of the population (or a part of it) could in theory converge on the opposite behavior (say, avoiding A) with carrying capacity following Equation III.15 with $E = E_B$.

Following Roughgarden [150], we expect the adaptive process by endogenous fitness to favor individuals who are robust in the face of environmental change. Given the sensory limitations, and the fact that a single individual’s behavior is fixed throughout its lifetime, selective pressure will shift seasonally on those individuals making the best use of the current season’s resources. We therefore predict the formation of subpopulations of individuals adapted to the different seasons, shifting in relative size with seasonal changes.

Results

In the seasonality experiment we analyze two simulations. The amplitude of environmental oscillations remains constant in the first, a simpler condition that makes it possible to interpret the behavior evolved by the population. In the second simulation the oscillations become more and more violent, putting the robustness of the collective behavior to the test.

In the first simulation $\lambda = E_0$, i.e., the energetic value of each element has a minimum of zero. The situation is illustrated in Figure III.17. The fact that the population oscillates, well correlated with A energy, indicates that it is differentiated. Inspection confirms that one subpopulation follows the B-avoidance strategy and a second one the random behavior. These are adaptive because both realize the higher energy provided by A atoms; individuals do not evolve the A-avoidance behavior because initially it is maladaptive to avoid energetic A atoms. Since all A is consumed, the population level is very close to the case in which there is no B; and since some B is consumed as well (by the subpopulation with random behavior), it is slightly higher. However, many B atoms are left over by the B-avoidance subpopulation and later consumed by the random behavior subpopulation when their energy content is higher. This joint strategy by the two subpopulations turns out to be a very robust collective behavior, because the available resources are used efficiently. Another source of robustness is the energy contained in the individual reservoirs. The observed oscillations of its population average reflect seasonal fluctuations, as is also illustrated in Figure III.17: energy accumulates in good times and then is spent slowly in bad times.

In the second simulation $\lambda = \epsilon t$, i.e., the amplitude of the energy oscillations in this environment increases linearly with time. The situation is illustrated in Figure III.18. At first, when the environment is almost stationary ($\lambda \ll E_0$), the random behavior is good enough. As the environment becomes increasingly harsh, however, individuals evolve into subpopulations with differentiated behaviors just like in the previous simulation. Eventually the two elements become in turn poisonous and their ingestion extremely dangerous. Nevertheless, the population survives for a long time after $\lambda > E_0$, providing further evidence for the robustness of the evolved collective behavior. Finally, after 150,000 cycles, the

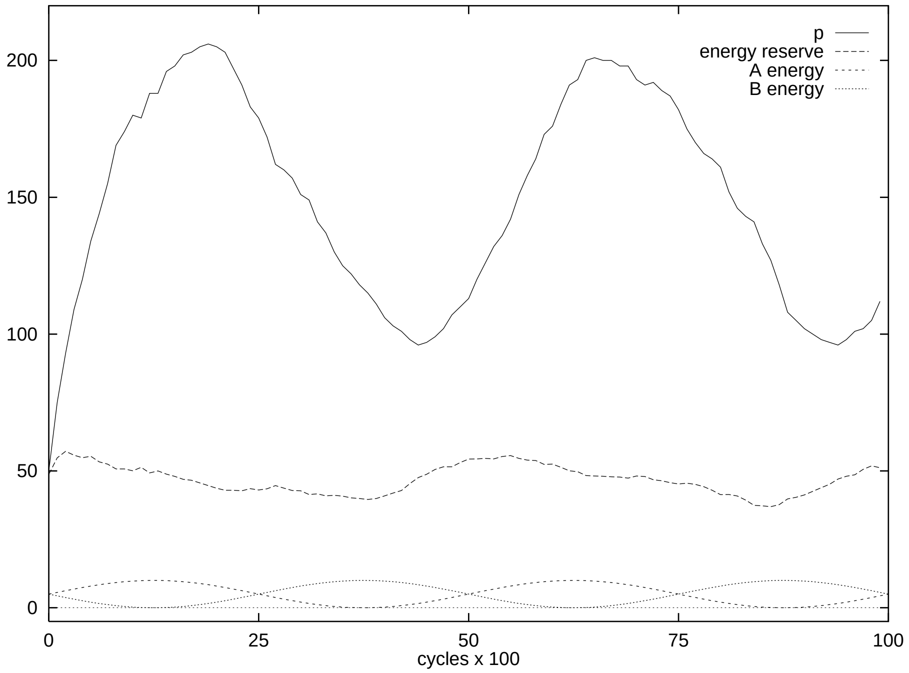


Figure III.17: Results of the simulation with fixed-amplitude energy oscillations. The energy of the two elements is shown on an energy scale (bottom). $E_0 = 5$, as in the following simulation. The period is $\tau = 5000$. A and B atoms have equal rates of replenishment. The resulting population size is plotted, along with the average energy reservoir levels.

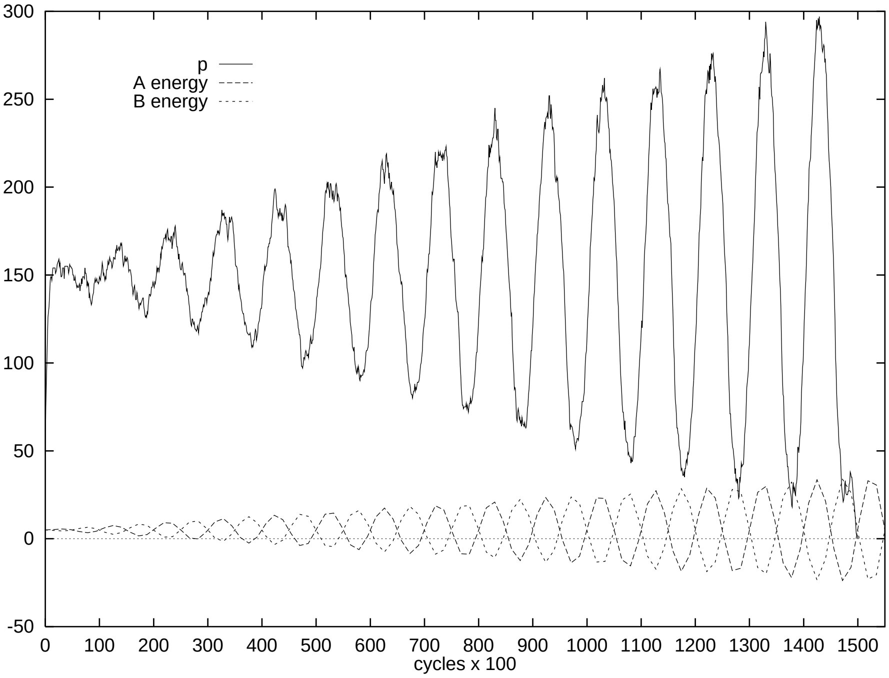


Figure III.18: Population dynamics of the simulation with increasing-amplitude energy oscillations. The slope of the amplitude is $\epsilon = 2 \times 10^{-4}\text{cycles}^{-1}$. The energy of the two elements, replenished at equal rates, is shown on an energy scale (bottom). The period is $\tau = 10000$ cycles.

environment is just too violent and the population goes extinct.

Figure III.19 shows the average age of the population in the same simulation. Observing this changing life-history trait assists us in understanding the dynamics of the adaptive process. From 0 to about 20,000 cycles, age increases steadily. In this phase there is a transition from the initial uniform population with random behavior to a structured population. The following general trend of decreasing age is an indication of shorter lives caused by increasing environmental harshness.

A more interesting observation comes from inspecting the finer details of age dynamics. Between approximately 20,000 and 90,000 cycles, average age fluctuates following a single wave form, anticorrelated with A energy and population size (cf. Figure III.18). As we find out by inspection, this indicates that

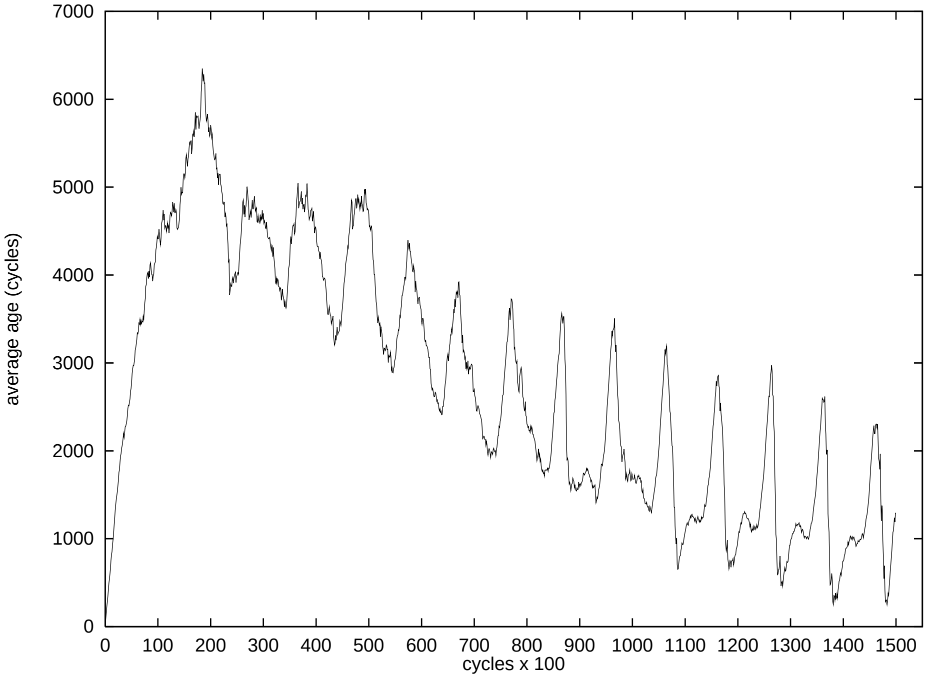


Figure III.19: Average age of the population evolving in the environment with increasing-amplitude energy oscillations.

the population has split into two subpopulations with random and B-avoidance behaviors, respectively; average age decreases when population increases (more A births than deaths) and vice versa. After about 90,000 cycles the age fluctuations become more convoluted and reveal a complexity that was not detectable from the population plot. In fact, two low peaks emerge during each period; new individuals are born during both A and B seasons! The explanation is that there are now three subpopulations. The environment has created selective pressure for the A-avoidance behavior to emerge as well — this is as adaptive as the symmetric B-avoidance behavior. Of course, all subpopulations must eat at least one element and thus eventually become extinct during the season in which it is poisonous.

Discussion

In the seasonality experiment, the increase of environmental complexity (from stability to fluctuations of growing amplitude) drives the differentiation of individuals into a composite collective behavior. The complexity of the evolving population could be defined as the number of subpopulations, each with a unique individual behavior, of which it is composed.

We first observe this (Figure III.17) when two subpopulations evolve in a relatively benign environment with energies oscillating always above zero and with fixed amplitude. The synergy of the two subpopulations provides for a more robust collective behavior than either strategy alone could accomplish. Survival is guaranteed because the best possible use is made of available resources — given the limitations of the sensory system.

It is worth noting that for a uniform population to accomplish the same behavior, it would be necessary for evolution to adapt behaviors at each season, with an ever-young population always behaving optimally in its season and dying — replaced by a new generation — when the next season sets in. This might have been possible had we used a much lower reproduction threshold, thus increasing the reproductive rate. Such a behavior could also emerge through the use of learning during each agent's lifetime, thus possibly detecting environmental changes at a smaller temporal scale than evolution alone can do. However, it is virtually impossible with the parameters and mutation rates used in the experiment, due to the short duration ($\tau/2$) of the seasons. On the contrary, some individuals belonging to each subpopulation survive during their bad season, thanks to the energy accumulated in their internal reservoir during their good season. This is reminiscent of the way some mammals hibernate during low-resource seasons.

Figure III.18 illustrates a similar differentiation occurring when the amplitude increases with time and brings the oscillating energies below zero. Further analysis of the age statistics (Figure III.19) has shown that as the fluctuations grow and the environment becomes less stable, the population splits again and a new subpopulation fills the one niche that had not yet been taken advantage of — energy from B atoms. This does not improve the overall carrying capacity, since B atoms were already being consumed by the other two subpopulations. In fact, no change is observed in the trend of the population curve (Figure III.18). Yet we speculate that the emergence of the last behavior has the beneficial effect of prolonging the survival of the population, based on the observation that the long-term decreasing trend in average age becomes less severe in the last 40,000 cycles (see Figure III.19).

III.E Interactions between evolution and learning

To the best of our knowledge, LEE is the only ecological simulation model allowing for the study of adaptation at both the population level (evolution) and the individual level (learning) simultaneously. This feature makes it possible to use LEE to explore some of the many possible interactions between learning and evolution. In this section we outline a number of such interactions elicited by latent energy environments.

In particular, we consider two types of internalization of environmental signals: one with respect to morphological traits, and one with respect to life history traits. The individual plasticity models used in these experiments are three types of reinforcement learning, driven by sensory predictions, associative reward-penalty, and parental imitation, respectively. In each case there is a "weak supervision" in the sense that agents are provided with reinforcement signals by the environment or by other agents — never by an external teacher. Learned changes only affect phenotypes, as learning is strictly non-Lamarckian.

III.E.1 Evolving efficient sensors

A critical aspect of all artificial life models is how the organism/environment "interface" is defined. This specification effectively cleaves the modeler's problem into two, a model of the internal "cognitive" system the organism uses to control its behavior (the brain in LEE), and a model of environmental change, due in part to these behaviors. In many artificial life models, the complexity of the behaviors displayed by the organism is evaluated without taking into account the role played by the sensory system and motor system engineered by the modeler. Rather than assuming any such *a priori* division, we propose to investigate the evolution of the organism/environment interface itself.

The sensory system in particular provides a strong coupling between environmental complexity and difficulty of the survival task. Intelligent behavior may result from complex sensors and trivial processing of this information, or simple sensors and clever processing [127, 175, 30]. In fact, experiments by Miglino and Parisi [130] in which the arbitrary specification of the sensory interface appeared to have important consequences for learning helped to motivate our work.

Sensors — transducers from external environmental signals to the internal cognitive system — therefore represent a crucial link between two distinct adaptive forces. They are simultaneously phenotypic features shaped by evolutionary forces to define a species' relationship with its environment, and the "input" channels on which a cognitive system's abilities to act and learn are entirely dependent. Features of the sensory system such as determinism, reliability, information content, computational complexity, signal/noise level, locality, etc. determine the difficulty of the task to be performed by the organism. The same holds for the motor system, but in this section we will limit our attention to the sensory system.

Experimental setting

In order to quantitatively estimate optimality of behaviors and sensory systems, we will use the following reaction matrix for three atoms, A, B and C:

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & -\beta E & E & \\ B & E & -\beta E & \\ C & & & \end{array} \tag{III.23}$$

with no byproducts. This environment is similar to the one described by matrix III.2, and in fact they are equivalent from an energetic point of view, so that the full analysis of Section III.C holds. The only difference of matrix III.23 is in the *potential* presence of the C element. This is non-reactive and thus makes no contribution to the carrying capacity of the environment. In fact, since the presence of C atoms is irrelevant, we leave them out of the environment entirely. Therefore we set $\nu_C = 0$, and $\nu_A = \nu_B = \nu$. However, C's presence in matrix III.23 can affect an agent's performance because if a sensor is devoted to detecting the presence of C rather than A or B elements, it will effectively detect noise; the agent would in such a case waste its sensory resources on a useless signal, rather than detecting signals (A or B) that might allow for efficient foraging strategies.

To study the evolution of sensory systems, we allow mutations of genetically specified sensor complexes in addition to neural net weights. Each sensor signals the presence of a complex of (possibly multiple) elements. In the experiments described here, a sensor complex corresponds to a single element. Random mutations at reproduction can change the complexes of an offspring's sensors with respect to its parent's. The probability that any sensor complex is mutated, or sensor mutation rate, is set to 0.05.

Let us now consider an agent's architecture. All organisms are endowed with a single binary motor and a pair of gut sensors. The rest of the body and brain specification depends on the learning model that we choose. We consider two variants of the basic architecture of Figure III.2, each allowing for a different source of feedback for reinforcement learning. The sensory configurations of both variants are illustrated in Figure III.3.

With the exception of the simulations using predictive learning (described ahead), an organism uses a pair of contact sensors and its neural net has a hidden layer with two neurons. If reinforcement learning is applied during the life of such an individual, it follows an extension of the A_{rp} *(associative reward-penalty)* algorithm [8]. Network weights are modified immediately after actions catalyzing non-zero energy reactions. If a reaction is exothermic (reward), the teaching input corresponds to the motor action just performed. If a reaction is endothermic (penalty), the teaching input is the binary negation of the action just performed. The weights are then updated by standard back-propagation of error [153]. This algorithm is equivalent to complementary reinforcement back-propagation, or CRBP [1].

In the simulations using predictive learning, an agent uses three pairs of ambient sensors and its neural net has eight hidden units. This architecture may seem unnecessarily complicated, but is justified by the use of a *predictive learning* algorithm. Following the work of Nolfi *et al.* [138], the neural net's output is extended to produce not only a motor action, but also a prediction of the sensory input it will receive after that action. This prediction is compared with the actual sensory information following the action, and the prediction error is then minimized by back-propagation.

Predictive learning demands a *difference,* in either space or time, between sensory inputs. The reason is the same as for approaching (cf. Section III.B.4). To allow a spatial comparison of signals, we use two ambient sensors directed towards opposite sides of the organism, each with a range of 2 moves; to break the lateral symmetry of these two sensors, we add a third ambient sensor with range 3, oriented frontally. Each sensor is replicated twice to allow full information (as will become clear in the next subsection), for a total of six ambient sensors and two gut sensors. Note also that in this architecture, the weights from the hidden layer to the motor units cannot be modified by learning; they can only change due to evolutionary forces.

Control runs

Using only contact sensors, it is impossible to approach atoms. Atoms must be encountered by some type of "blind" foraging pattern. Given the toroidal structure of the LEE world and the binary motors, always moving straight ahead

```
if  the gut is empty, go forward
else if  the forward cell's content is the same as the content of the gut, turn
else  go forward
```

Figure III.20: Optimal behavior for full-information sensors.

Gut sensors	Contact sensors	Move
A	A	Turn
A	B	Forward
B	A	Forward
B	B	Turn

Table III.3: Implementation of the optimal behavior for full-information sensors.

is typical. With only blind foraging allowed, and for the environment described by matrix III.23, the *optimal* behavior is shown in Figure III.20.

To implement this optimal strategy, the sensory system should ignore C's and discriminate A from B with both gut and external sensors. Assuming the availability of gut and contact sensors giving accurate discriminating information, an (example of) optimal neural net function can be represented as in Table III.3.

Note that this table simplifies the space of potential behaviors from the entire repertoire of available sensors in order to focus on the four consequential decisions. To make the discriminations necessary for this optimal strategy, one of the gut's two sensors must sense A and the other B, and the same must be true of the two external sensors.

If some sensors give information that is only partially discriminating, then only suboptimal strategies are possible. For example assume that the gut's sensors are as above but the environmental sensors both sense A. With this limited information, the best possible strategy is shown in Figure III.21. The neural net implementation of this strategy is shown in Table III.4.

```
if  the gut is empty, go forward
else if  both the gut and the forward cell's contents are A, turn
else  go forward
```

Figure III.21: Suboptimal behavior for partial-information sensors.

Gut sensors	Contact sensors	Move
A	A	Turn
A	?	Forward
B	A	Forward
B	?	Forward

Table III.4: Implementation of the suboptimal behavior for partial-information sensors.

Critically, this *sub-optimal* strategy is a function very much like "inclusive OR," which is known to be much simpler than the "exclusive OR" function underlying the optimal strategy above.

To verify that full information sensors can in fact afford better adapted behaviors, we first run two control simulations aimed at measuring the baseline and best fitness (carrying capacity) obtainable by evolving the network connections alone. There is no learning during life, and the sensor configurations are fixed. The worst possible sensors are those that leave the organism blind, i.e., without any information whatsoever about the gut or the external environment. This is done by setting all sensors to C, so that the inputs of the network are always zero: the adapted weights must implement constant actions, independent of the environment, i.e., a "random choice" behavior. The best possible sensors are those allowing organisms to discriminate between A and B atoms. This is done by setting both sensor pairs to (A,B): the adapted weights may use these "full information" signals to implement the optimal strategy of Figure III.20, for example in the way shown in Table III.3.

The results are shown in figure III.22. Random choice, as expected, cannot show any improvement past the very initial phase in which the strategy to always move sets in. Full information sensors, on the other hand, allow near-optimal convergence. To see this, note that the population size at the end of the simulations corresponds to $\eta \sim 0.87$ (from Equation III.13). Inspection confirms that the majority of organisms at the end of the simulations exhibit behaviors equivalent to the optimal strategy.

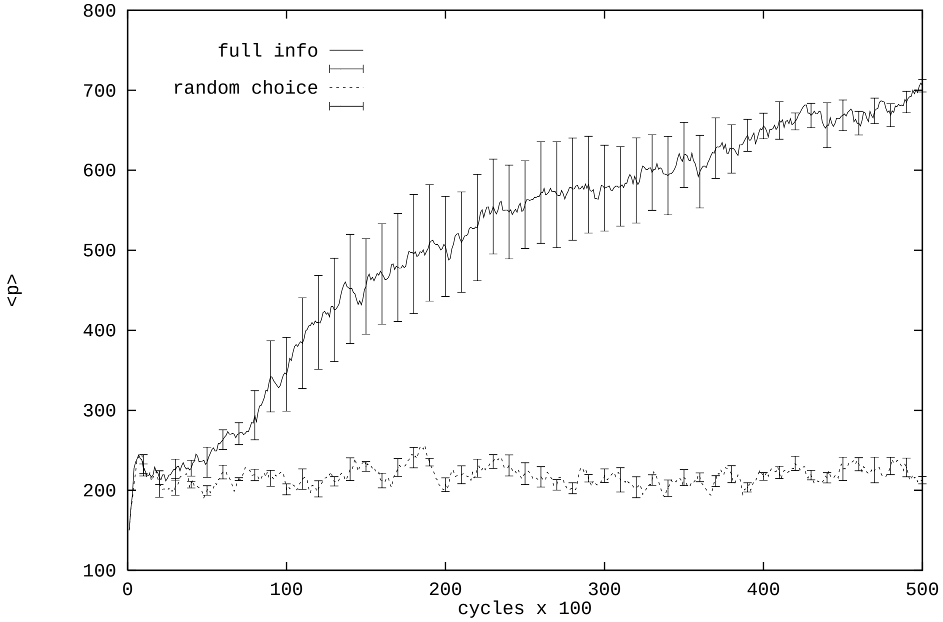


Figure III.22: Population size in two control simulations with blind sensors (random choice) and full information sensors. Averages and standard errors are computed over different runs.

Given that informative sensors have higher fitness than non-informative ones, as measured by carrying capacity, we expect selective pressure in favor of efficient sensory systems when these are allowed to evolve. The only gut and contact sensor pairs which afford the information necessary for the optimal strategy are (A,B) and (B,A). In either case, we expect the population to converge to optimal percentages of 50% for A (and B) sensors, and 0% for C sensors.

Figure III.23 plots the evolving sensor complexes in simulations with evolving networks and sensors, by plotting percentages of A and C summed over all sensors in the population. Although the trend is consistent with our expectation, the large error bars indicate the importance of random genetic drift in these runs. Upon repeating the simulation for different runs, sensor percentages converge to highly dispersed values. Our interpretation of such a large drift is that only about 5% of the possible sensor configurations (4 out of the total $3^4 = 81$) allow for optimal strategies to evolve. Furthermore, these configurations do not all share the same coadapted connection weights, so the resulting behavior is very fragile in the face of mutations on the sensory system.

In contrast, a suboptimal strategy like the one shown in Table III.4 can be implemented with $28/81 \approx 35\%$ of the sensor configurations. While this strategy results in somewhat inferior performance, it is more robust because not only more configurations can achieve it, but more importantly, many of them function with the same network weights. This follows naturally from some of the sensors being redundant with respect to the suboptimal strategy; these sensors can be mutated without affecting the network behavior. Such redundancy is also the source of the observed genetic drift: *robustness* and *drift* are two faces of the same selective pressure against optimality. To be sure, optimal sensors are generated from mutations at all times, but they are destroyed before the connections evolve to the coadapted weights. The fact that the suboptimal strategy is the winning one is also confirmed by inspecting organisms at the end of runs.

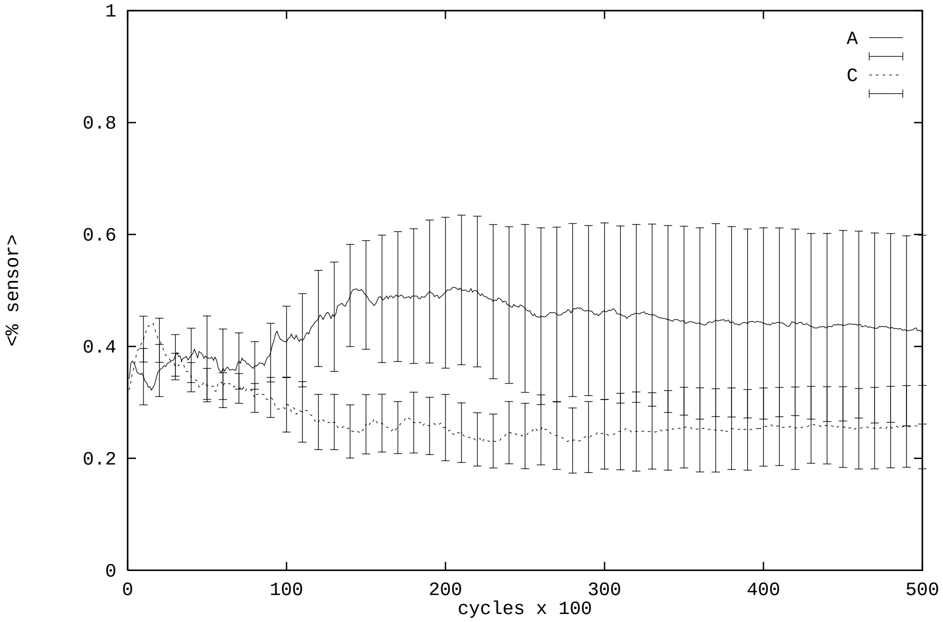


Figure III.23: Evolving sensor percentages in simulations with evolution alone (no learning), with the contact sensors configuration used with associative reinforcement learning. The B percentage is not shown for clarity (it can be obtained by subtracting those of A and C from 100%, and its expected optimum is identical to that of A). Averages and standard errors are computed over different runs.

Prediction learning

The so-called “Baldwin effect” describes an indirect mechanism by which learning can increase fitness of a population, short of direct, Lamarckian inheritance of the knowledge acquired via learning [7, 182, 17]. It is reasonable to hypothesize, then, that learning may increase selective pressure towards more informative sensors and consequently reduce the drift observed in the previous section, facilitating the evolution of informative sensors.

Our first choice for an individual adaptation mechanism is reinforcement learning by sensory prediction. The architecture is that described earlier, and we use learning rate 0.2. Evolving sensor percentages are plotted in Figure III.24. Inspection reveals that such sensors induce behaviors performing some approaching; but as far as combining elements, the evolved behaviors correspond to suboptimal strategies. Good approaching strategies can improve fitness only to a limited extent in our environments, therefore these results basically show that prediction learning provides no significant advantage over evolution alone, neither for fitness improvement nor for genetic drift reduction.

This apparently disappointing result is actually of assistance in understanding the nature of the information afforded by ambient sensors and their interaction with the prediction learning algorithm. One difficulty related to these sensors is the *ambiguity* of their signals, because the function of the environment state that they compute integrates signals over the neighborhood and thus is non-invertible (many-to-one). In this sense, the information they provide is harder to use than for contact sensors, even though theoretically it allows prediction because it is less local. Another difficulty related to prediction learning is that the prediction task is *not well correlated* with the survival/reproduction task. For example, one reliable prediction in these environments is the absence of C atoms, i.e., a zero signal from all inputs corresponding to C sensors. Therefore the learning process drives phenotypes toward behaviors that are not useful because they are based on blind (C) sensors.

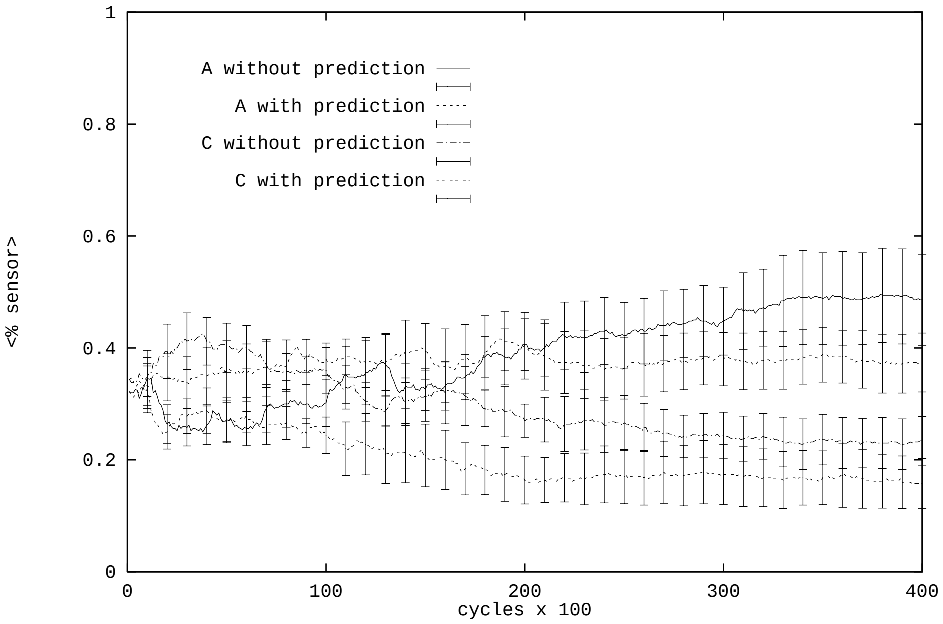


Figure III.24: Evolved sensor percentages in simulations with ambient sensors, with and without prediction learning. Averages and standard errors are computed over different runs.

Associative reward-penalty

From the result of the previous simulations we certainly cannot draw the conclusion that learning in general cannot interact constructively with evolution, but rather the observation that ambient sensory prediction is not the appropriate cue for our experiment in internalization. Therefore we need another learning algorithm, with the same motivation as for prediction learning: to determine whether the Baldwin effect can facilitate the evolution of informative sensors. However, we seek a reinforcement signal better correlated with the fitness of the survival task. For this we turn to the CRBP algorithm, as described earlier.

A number of authors have explored the use of associative reinforcement learning in conjunction with neural networks, genetic algorithms, and artificial life [1, 186, 2]. Let us use instantaneous energy changes as the reinforcement signal, as discussed in Chapter II. Energy's direct correlation with LEE's local fitness should provide agents with appropriate cues to be internalized for evolving efficient sensors.

Going back to the simpler architecture with only contact and gut sensors, the only actions that determine fitness are those choosing whether or not to move forward when a food is sensed by the contact sensor, based on the gut content. Therefore weights are updated only immediately after actions catalyzing reactions.[4]

Figures III.23 and III.25 show the results of adapting sensors with evolution alone and with associative reinforcement learning, respectively. A significant reduction in drift and increased selective pressure toward informative sensors (e.g., against C sensors) is observed in the simulations with learning. This reflects the fact that in a greater fraction of the runs the population converges to full information sensors (approximately 50% A, 50% B, and no C) and optimal behaviors. The result supports our hypothesis that the Baldwin effect can facilitate the evolution of sensors, if the appropriate environmental cues are internalized.

---

[4] Learning rate 0.1.

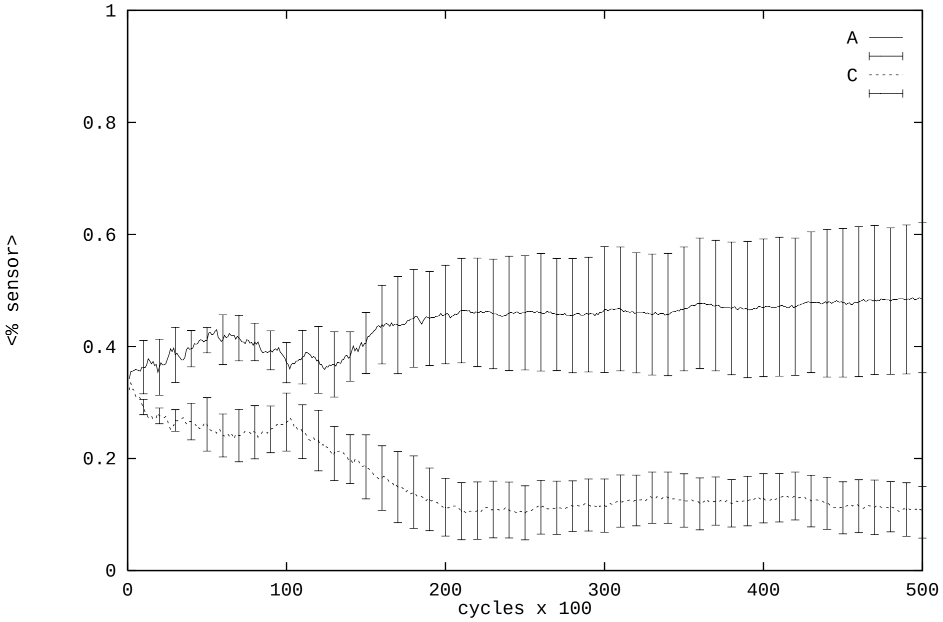


Figure III.25: Evolving sensors using evolution and associative reinforcement learning. Learned network weights are not transmitted genetically, yet learning provides additional selective pressure in favor of optimal sensors. Averages and standard errors are computed over different runs.

III.E.2 Evolving maturation age

To many computational modelers using neural networks as models of learning, it is second-nature to provide as much training experience as possible, since almost always this leads to increased performance. But when these neural nets are used in conjunction with genetic algorithms as biologically realistic models of the evolution of learning individuals, it is necessary to consider the problem as it occurs in "first Nature:" Is the increased time required by extended training worth the evolutionary costs of providing it? The fact that many organisms spend their prolonged immaturity as part of family units that can — potentially — shape the experience of the learning juvenile in predictable, heritable ways makes the role of the learning period especially important as one begins to contemplate proto-cultural effects on cognitive development. In the last experiment of this chapter we address restricted versions of these questions, focusing exclusively on imitative types of learning between parent and child.

In the study of the evolution and adaptation of life history traits in animals, the commonly accepted theory states that any particular trait accomplishes a trade-off between the different selective pressures acting simultaneously upon the phenotypic variants of that trait. One such trait that is central in behavioral and developmental psychology as well as in theoretical biology is the age at which an individual reaches reproductive maturity, or *maturation age.* The selective pressures concerning the evolution of age at maturity considered in theoretical biology are the adaptive costs and benefits associated with anticipating or delaying maturation. Typically, the costs of delayed maturation include: (i) lower population reproductive fitness due to longer generation time; (ii) lower individual reproductive fitness due to decreased probability to reach the mature stage; and (iii) parenthood cost due to longer immature period requiring parental care. Conversely, the benefits of delayed maturation typically include: (i) higher fecundity of the parent who can grow for a longer time and better endure the reproductive effort; and (ii) lower instantaneous juvenile death rates due to better quality of offspring or parental care [168].

Ecological field studies attempting to quantify the magnitude of selective pressures toward delayed maturation necessarily focus on easily measurable life traits, such as body size and weight. On the other hand, animal and human psychology studies concerned with social cognition emphasize the improvement of offspring phenotypes taking place through cultural learning. It would be desirable to be able to quantitatively correlate such behavioral advantages with life history models of delayed maturation. However, the difficulty in measuring phenotypic traits associated with behavioral development, necessary to apply analytical trade-off models, causes any behavioral benefit of delayed maturity to be neglected in studies of the evolution of maturation.

We propose to use LEE to model the advantage of parental care as the only benefit of delayed maturation. Lower death rates may result from an improvement of phenotypic behavior before the adult stage is reached. In the model, this improvement is acquired by the offspring through learning by *imitation* of its parent. Notice that it makes sense to model learning only before maturation, because parents must act as teachers to their immature offspring. Johnston [76] has associated learning with immaturity in the study of costs and benefits of phenotypic plasticity as a life history trait. While it is not our intention to propose imitation as a universal mechanism for cultural transmission, Tomasello *et al.* [178] identify imitative learning as the first of the three forms in which cultural learning manifests itself during ontogeny. Therefore, we use imitation to model one possible mechanism by which parents may confer a cultural advantage to their immature offspring.

Experimental setting

We illustrate quantitative evidence in support of the hypothesis that the fitness improvement of phenotypes by means of their learned behavior plays an important role in the evolution of maturation age. We simulate the evolution of

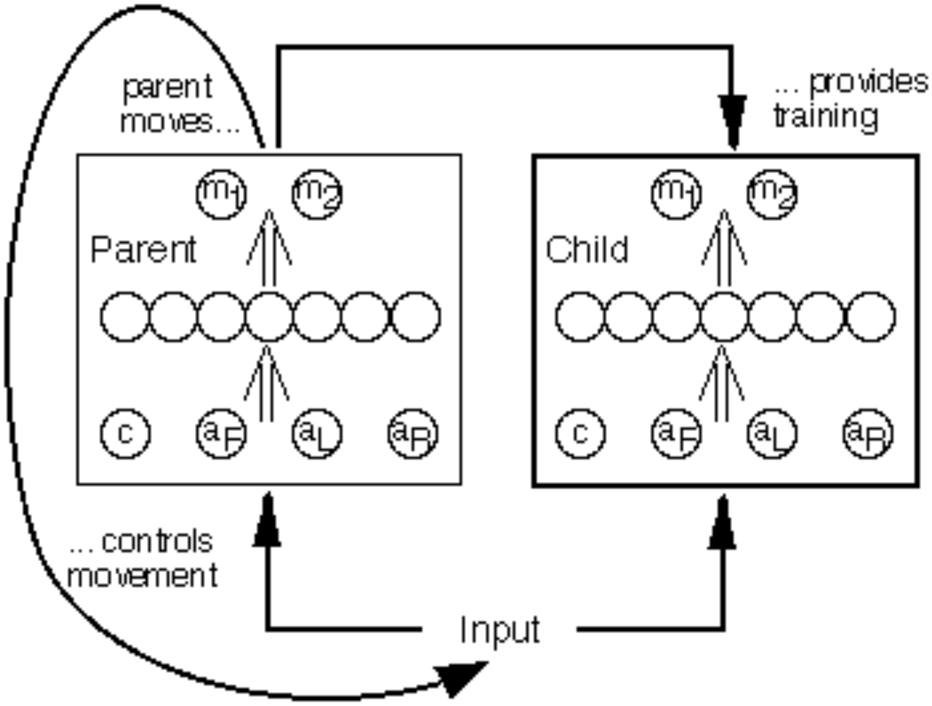


Figure III.26: Imitative learning architecture. The offspring sits on the parent's shoulders, experiencing the same sensory inputs and being trained to imitate the parent's motor responses.

age at maturity as a genotypic trait regulating the duration of phenotypic learning by imitation. The model exhibits all of the costs of delayed maturation enumerated above. We hypothesize that the ability to learn imitated behaviors is adaptive, and hence expect that there be selective pressure toward delayed maturation to allow such learning to occur.

In this experiment agents have three ambient sensors, two with a range of 3 oriented to the sides of the organism, and one with a range of 5 oriented straight ahead. There is also one contact sensor. The motor system is made of a binary motor controlled by two binary output units. The neural net connecting sensors to motors has a single hidden layer with seven units. The environment has one element type (cf. Equation III.15 in Section III.C).

To study maturation age, the life of organisms is divided into two distinct stages separated by an "age of maturity" represented by the value of a new gene, called a_m. Values of a_m are between zero (no immaturity) and the age at death (no maturity). During its immature phase, an organism differs in exactly two respects from its mature form. First, the immature phenotype is not allowed to reproduce. Its energy at birth is conserved until maturity, and its moving costs are entirely transferred to its parent. Second, juveniles undergo a process of learning by imitation. The details of this construction are suggested by Figure III.26. Parent and child experience identical input stimuli from the world, but the output computed by the parent's neural net is used to determine both organisms' movement. The motor units of the parent's network are also used as training for the child, based on which error signals are generated and weights are updated by error back-propagation.[5] Our metaphor for this stage is to think of the offspring as being carried on the shoulders of its parent. When an organism reaches age a_m it detaches from the parent and becomes an adult, normal member of the population.

Let us review the way in which the three costs of delayed maturation are modeled. First, longer generation time is implicit in the LEE model: longer immature stages correspond to shorter times in which offspring can be generated. Second, increasing the immature stage decreases the probability to reach maturity, due to the constant expected probability of death of the parent per life cycle. When a parent dies, its "orphan" immature offspring are dropped into the environment before becoming adults. An orphan cannot move until it reaches its mature age, but may run out of energy and die before becoming adult. The probability of survival to mature age is higher the closer to maturation the offspring is when the parent dies. The third cost of the immature stage is associated with parental care: a parent pays an additional energy toll equal to the cost of a move for each offspring it carries on its shoulders. Energy is still conserved, since the immature offspring

---

[5] The learning rate is set to a high value (0.8) because the networks of immature organisms are give a relatively short training experience (the duration of the immaturity stage) [16].

incur no living costs as long as they are carried on their parents' shoulders.

Learning is the only benefit of delayed maturity in our model. Thus if a delay of maturation age is observed in simulations, it can only be attributed to the advantage provided by learned behaviors. Note that this benefit is conferred upon the offspring and not the parent. In evolutionary ecology, similar benefits for the offspring are mainly attributed to morphologic development, such as offspring size. The benefit modeled here is less explicit because mediated through the behavior of the offspring: juvenile mortality can be decreased thanks to the experience accumulated by offspring during their immature life stage, by way of parental imitation.

The a_m gene is allowed to evolve, being selected together with the rest of the organisms' genotypes. It may be mutated (with probability 0.1) at reproduction by a random additive deviate uniformly distributed in the interval $[-a_m, +a_m]$. Negative values are clamped to $a_m = 0$. Connection weights are mutated with probability 0.15 by random additive deviates uniformly distributed in the range $[-2.5, +2.5]$.

Results

Simulations are run for 150,000 cycles. In Figure III.27 population size as a function of time is plotted for two single runs in which a_m is held at the constant values 0 (no immaturity) and 100 cycles, respectively. The first case ($a_m = 0$) provides a baseline with neither the costs nor advantages of an immature period. The observable increase in population corresponds to an improvement in the approaching behaviors of the organisms, due to the evolution of their network weights. The second case ($a_m = 100$) gives us a measure of the magnitude of the cost of delayed maturation: this is large enough to drive the population to extinction in less than 25,000 cycles. Thus we expect strong selective pressure against delayed maturation. The expected size of a population of agents with random behaviors (from Equation III.15) is also shown for comparison.

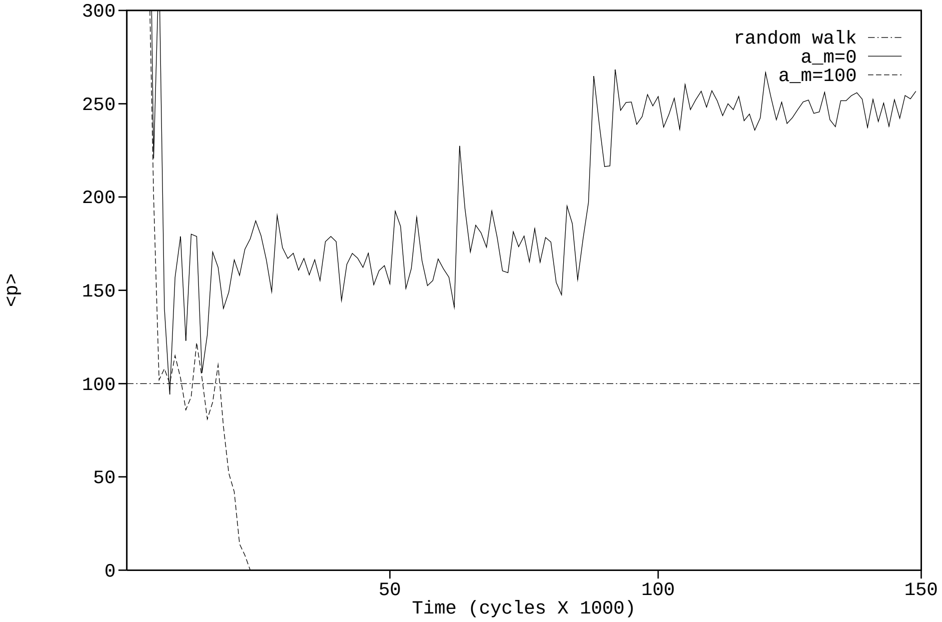


Figure III.27: Population size for different fixed ages at maturity. Data are collected form two single simulation runs. Extinction occurs in the case of non-zero maturation age. “Random walk” refers to the random behavior of Equation III.15. At the beginning of the simulations, a large amount of environmental resources results in an exponential peak in population size (not shown in the plot for clarity). Then density dependence sets in.

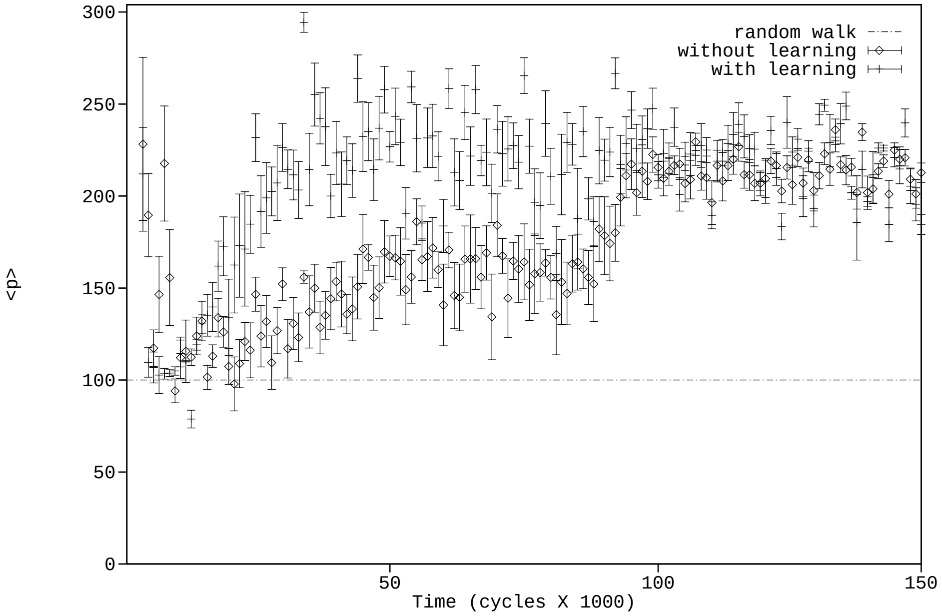


Figure III.28: Population size with evolving maturation age gene. The case with learning from imitation results in faster fitness improvement. Averages and standard errors are computed over repeated simulation runs.

In order to study the evolution of age at maturity, the next experiments add a_m to the genome of the evolving organisms. The population is initialized with a_m uniformly distributed over the interval $[0, 100]$. Figure III.28 shows the population sizes for two populations, one in which imitative learning is enabled during the immature phase and one in which it is not. In both cases, after the initial stochastic fluctuations, the populations are able to evolve individuals with behaviors significantly better adapted than random. However, with imitative learning, the evolution of good approaching behaviors is significantly accelerated.

Figure III.29 plots the population's average value for the evolved a_m gene in the two experiments of Figure III.28, again with and without imitative learning, respectively. In the absence of learning, other costs of immaturity dominate and a_m rapidly evolves to zero. When the young are also allowed to learn via imitation, however, a significant delay in the extinction of the immature phase is

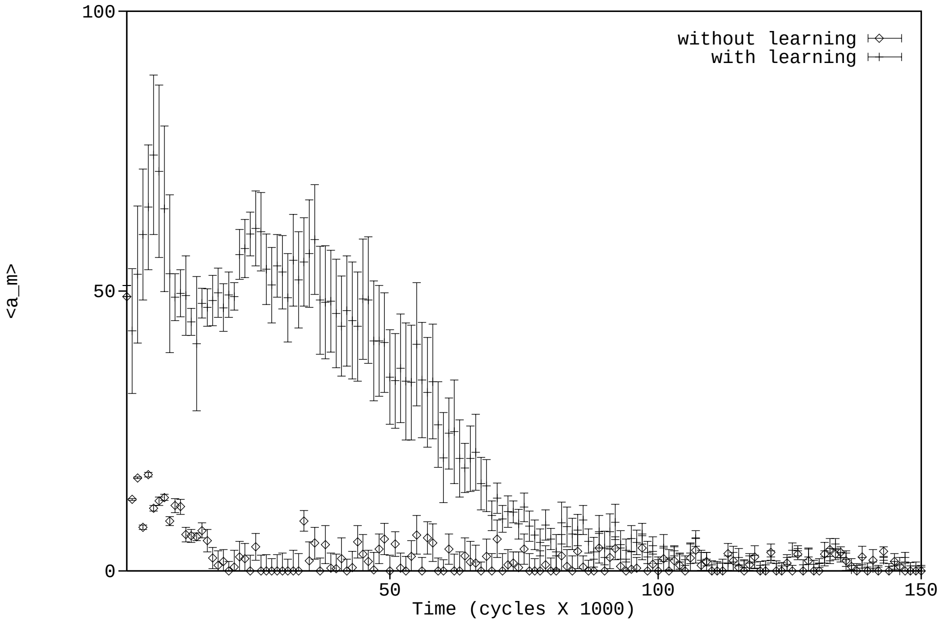


Figure III.29: Evolution of the maturation age gene. When imitation occurs, maturation is delayed noticeably; the disappearance of immaturity (due to genetic assimilation) is also delayed evolutionarily. Averages and standard errors are computed over repeated runs.

observed. It is important to note that this is an evolutionary delay experienced by the species, not to be confused with the developmental delay controlled by the a_m and experienced by individuals. It is also important to recognize that the inclusion of imitative learning does not keep a_m from becoming zero; it only prolongs this process.

In summary, the delay observed represents evidence that there is a trade-off between costs and advantages of delayed maturation. This trade-off is quantitatively estimated by the value — slightly above 50 cycles — around which the population's average maturation age oscillates initially (see Figure III.29). Since in this model the only advantage can be that of offspring learning via parental imitation, we conclude that cultural transmission of adapted behaviors is one of the evolutionary factors creating selective pressure in favor of longer immature stages.

Discussion

The role learning and other forms of phenotypic plasticity can play in accelerating evolutionary change is increasingly well explored. The Baldwin effect, as discussed in Section III.E.1, has come to describe any mechanism by which a learner, prohibited from encoding the consequences of learning directly onto the genome (as Lamarck proposed), can nevertheless enjoy a selective advantage as a direct consequence of its learning [7, 182, 17]. The ability of an organism to adapt within its lifetime towards beneficial characteristics of its environment (i.e., learn) increases the probability that other genetic traits serendipitously correlated with this same benefit will increase in frequency. A range of evidence for this important, albeit subtle connection between learning and evolution has been found by a number of investigators using computer simulations [66, 14, 2, 59, 138]. The above results are consistent with these accounts. The learning organisms' ability to explore a range of behavioral strategies within their lifetimes means that the evolutionary process is capable of exploiting much more information about adaptively favorable characteristics of the environment.

But if learning by imitation during immaturity provides selective pressure toward delayed maturation, why does a_m eventually converge to zero, as shown in the last phase of its evolutionary course in Figure III.29? The Baldwin effect gives us a simple interpretation of this fact as well. During the juvenile period, modifications that learning causes upon the phenotype are eventually re-discovered by the evolutionary process and thereby affect the genotype. Once this happens, learning is no longer useful because offspring at birth are already capable of the behaviors that in earlier generations they could only acquire by imitating their parents. Therefore learning no longer confers an advantage to immature offspring. The costs of delayed maturation remain the same, and the missing benefit causes the immature period to disappear. This phenomenon is called *genetic assimilation.* Its occurrence in the above experiment is confirmed by the strong positive correlation observed between evolved values of a_m and two indicators of behavior quality: (i) the amount of change per unit time undergone by neural net weights as a result of learning; and (ii) the fraction of optimal moves made by new-borns.

Learning by imitation is a relatively simple paradigm within which we have been able to model the costs of delayed maturity on both mature organisms (parents) and immature ones (offspring). The adaptive advantage of transferring a parent's experience onto its offspring by "carrying them on its shoulders" has been shown to be large enough to permit a significant delay in maturation age, at least until the adaptive behaviors are transferred to the genotypes of the population via the Baldwin effect. Our data, therefore, supports behavioral and cultural models of maturation age as a life history trait. Since imitation seems to be a very basic form of cultural transmission in social animals, this result not surprisingly points to one account for the long immaturity stages in these species. Clearly, many other factors determine the trade-off for optimal age at maturity, but critical data about cultural learning are available for only a very few species, e.g., apes and humans [178]. We hope our results stimulate field studies aimed at providing experimental evidence for (or against!) the behavioral benefits of delayed maturation illustrated here.

Chapter IV

Adaptive Information Agents

The trend of the recent years in distributed information environments is a good example of the life-like complexity that we expect to observe in most aspects of information and computational science. The explosion of the Web and electronic mail, multiplying the number of information providers and consumers many times over and bringing the Internet inside the average home, has created formidable new opportunities and challenges in almost every area of computer and information science.

In an effort to address such problems, researchers in artificial intelligence and information retrieval have already been successful in developing agent-based techniques to automate many tedious tasks and facilitate the management of the growing amounts of information flooding users. But the work has just begun. There is still much need for tools to assist users in ways that scale with the growth of the Web, and adapt to both the personal preferences of the user and the changes in user and environmental conditions.

Learning and other forms of adaptation are essential for *situated* and *autonomous* agents. A situated agent is a system whose performance is characterized by a tight coupling with an external environment [79]. The environment is unpredictable with certainty by the agent, both because it has a dynamic nature, changing with time and locations, and because the agents perception of the environment is mediated by a sensory apparatus that may be noisy, incomplete, inconsistent, and/or unreliable. An autonomous agent is a system that must perform, on behalf of the user, some task whose solution is not completely specified a priori [104]. The agent must autonomously make some choices based on context, i.e., its perception of conditions judged relevant to the problem. Therefore both situated and autonomous agents must learn some of the regularities about their operating environments and adapt their behaviors to perform their tasks successfully [105].

The situation is not unlike the one faced by ecologies of organisms adapting in natural environments. The capabilities of such *natural agents* illustrated in Chapter III — local adaptation, internalization of environmental signals, distributed control, integration of externally driven and endogenous behaviors, etc. — represent desirable goals for the next generation of *autonomous, intelligent, distributed, adaptive, artificial agents.* This chapter focuses on the applications of local selection and internalization algorithms, inspired by the artificial life models discussed in the previous chapters, to the search and retrieval of information distributed across networked environments.

IV.A Background: Information systems and agents

IV.A.1 Machine learning

We have shown in Chapter II that evolutionary algorithms based on local selection are not suitable in every domain (e.g., combinatorial optimization). However, distributed search in networked environments is a multimodal problem that presents many of the characteristics making it an ideal target for such an approach. This task requires a heterogeneous cover of the search space rather than a convergence to the perceived global optimum. Indeed it can easily be cast into the graph search framework discussed in Section II.E.1, in which local selection algorithms have proven very effective.

There is plenty of data available on-line, and although it may be noisy and inconsistent compared with manually constructed relevance assessments, adaptive algorithms must take advantage of what is cheap and realistic in the actual search environment [171]. Neural networks have been used to learn probability distributions for text retrieval by logistic regression [134].

If relevance assessments from the user are available, active learning should take advantage of them because it has been shown that they can considerably improve the performance of retrieval systems [92]. This suggests the integration of unsupervised adaptation with "adaptation by examples," driven by relevance feedback. Users need not interact directly with agents (especially if these are executing on remote hosts); the environmental model behind local selection allows relevance feedback to interact asynchronously with mobile agents. Relevance feedback is a selfish process from the user's standpoint [171], but it provides agents with modified rewards that can improve on their models of relevance and therefore on their performance.

Large, distributed text collections are a typical example of massive data sets that challenge machine learning techniques due to their huge feature space dimensionality [93]. Feature selection techniques are often suggested to deal with the curse of dimensionality [18, 143], although the unsupervised case remains largely unaddressed. In distributed applications where agents are situated in temporally or spatially local portions of heterogeneous or dynamic environments, feature selection should be strongly context-dependent.

IV.A.2 Information retrieval

Unsupervised learning has been applied extensively in information retrieval, especially for automatic classification and clustering [179, 159]. Typically, the model of the document collection is global and static. Any two documents are assumed to reside on the same physical locations (disk, LAN, or server) or in locations with equivalent access costs. Further, a documents is not expected to change over time. Therefore locality (temporal and spatial context) is neither a concern nor an asset. Whatever the goal (classification, clustering, ranking), the learned strategy is supposed to be equally applicable to the whole collection.

Hypertext brings a new dimension to text classification, as non-local features (links) may extend the local (words) context and consequently reduce the descriptive power of the text local to any single documents. For example, a document may consist of a list of links to other sources, or be broken into many smaller documents (sections). These issues are just beginning to be taken into account to improve classification performance [27].

Relevance feedback also belongs to the mainstream of text retrieval research, as a way to estimate word probability distributions through supervised learning [145, 165]. Again, the use of users' relevance assessments is traditionally disjoint from local context, although it can exploit a *personal* context, for example to adjust ranking functions or make suggestions based on user preferences and interest profiles. Relevance feedback is also typically used to select word features for query expansion [158, 61]. But when space, time, and user needs constitute highly dynamic and heterogeneous contexts, no query expansion is necessarily appropriate everywhere or forever.

In general, there is a need to address some of the new challenges posed by text classification to machine learning, especially the need to extend information retrieval to deal with time-varying documents and user needs [93] and with large, dynamic, and heterogeneous collections such as the Web [94]. On-line search makes the classification problem both simpler and harder: simpler, because there are only two classes (relevant and irrelevant with respect to the current query); and harder, because the relevant class can be heterogeneous (as for long-standing user profiles) and because class membership can change over time with the user's shifting interests. We must consider both of these aspects as assets in the construction of an agent-based retrieval system.

IV.A.3 Search engines

Exploiting the proven techniques of information retrieval, search engines have followed the growth of the Web and provided users with much needed assistance in their attempts to locate an retrieve information from the Web. Search engines have continued to grow in size, efficiency, performance, and diversity of services offered. Their success is attested by both their multiplication and popularity.

The model behind search engines draws efficiency by processing the information in some collection of documents once, producing an *index,* and then amortizing the cost of such processing over a large number of queries which access the same index. The index is basically an inverted file that maps each word in the collection to the set of documents containing that word. Additional processing is normally involved by performance-improving steps such as the removal of noise words, the conflation of words via stemming and/or the use of thesauri, and the use of the word weighting schemes.

This model, which is the source of search engines' success, is also in our opinion the cause of their limitations. In fact it assumes that the collection is static, as was the case for earlier information retrieval systems. In the case of the Web, the collection is highly dynamic, with new documents being added, deleted, changed, and moved all the time. Indexes are thus reduced to "snapshots" of the Web. They are continuously updated by *crawlers* that exhaustively visit and periodically revisit every Web page. At any given time an index will be somewhat inaccurate (e.g., contain stale information about recently deleted or moved documents) and somewhat incomplete (e.g., missing information about recently added or changed documents).

A recent study by Lawrence and Giles [89] confirms these observations on the limited coverage and recency of the data in any search engine's index. The best coverage achieved among the six most popular search engines was estimated to be around 34% of the Web's indexable pages. A significant portion of the information stored by these search engines was also found to be stale, as attested by the fraction (up to 5%) of broken links, i.e., pages deleted or moved since they had been indexed.

These problems, compounded by the huge size of the Web, hinder search engines' capability to satisfy user queries. Users are often faced with very large hit lists, low *recall* (fraction of relevant pages that are retrieved), even lower *precision* (fraction of retrieved pages that are relevant), and stale information. These factors make it necessary for users to invest significant time in manually browsing the neighborhoods of (some subset of) the hit list.

A way to partially address the indexing problems posed by the size and dynamic nature of the Web is by decentralizing the index-building process. Dividing the task into localized indexing, performed by a set of *gatherers,* and centralized searching, performed by a set of *brokers,* has been suggested since the early days of the Web by the Harvest project [21].

A step toward enriching search engines with topological information about linkage to achieve better precision has been suggested by the CLEVER[1] group at IBM Almaden Research Labs. The idea is to use hyperlinks to construct "hub" and "authority" nodes from the Web graph and it has proven effective in improving document retrieval and classification performance[28, 27].

IV.A.4 Information agents

Autonomous agents, or semi-intelligent programs making automatic decisions on behalf of the user, are viewed by many as a way of decreasing the amount of human-computer interaction necessary to manage the increasing amount of information available on-line [106]. Many such software agents, more or less intelligent

---

[1] Formerly known as HITS project.

and more or less autonomous, have been developed in the recent years. The great majority of them suffer from a common limitation: their reliance on search engines. The limited coverage and recency of search engines cannot be overcome by agents whose search process consists in submitting queries to search engines. However, many agents partially improve on the quality of any search engine's performance by submitting queries to many different engines simultaneously. This technique, originally called *metasearch* [129], has indeed proven to increase recall significantly [180].

Typical examples of agents who rely on search engines to find information on behalf of the users are homepage or paper finders. CiteSeer [19] is an autonomous Web agent for automatic retrieval and identification of publications. Ahoy [163] is a homepage finder based on metasearch engine plus some heuristic local search. WebFind [135] is a similar locator of scientific papers, but it relies on a different information repository *(netfind)* to bootstrap its heuristic search. To be sure, agents like CiteSeer, Ahoy and WebFind do perform some autonomous search from the pages returned by their initial sources, but this is limited to the servers of their starting points.

A different class of agents are designed to learn user interests from browsing for recommendations purposes. Syskill & Webert [142] is a system that identifies interesting Web sites from large domain-specific link lists by learning to rate them based on relevance feedback. WebWatcher [3, 74] is a tour guide agent that learns from experience of multiple users by looking over their shoulders while browsing. Then it provides users with suggestions about what links to follow next. Similarly, Letizia [95] is an autonomous interface agent that assists the user in browsing the Web by performing look-ahead searches and making real-time recommendations for nearby pages that might interest the user. WebMate [29] assists browsing by learning user preferences in multiple domains, and assists searching by automatic keyword extraction for query refinement. All these agents learn to predict an objective function on-line; they can also track time-varying user preferences. However, they need supervision from the user in order to work; no truly autonomous search is possible.

Fab [6] and Amalthaea [136] are multi-agent adaptive filtering systems inspired by genetic algorithms, artificial life, and market models. Term weighting and relevance feedback are used to adapt a matching between a set of discovery agents (typically search engine parasites) and a set of user profiles (corresponding to single- or multiple-user interests). These systems can learn to divide the problem into simpler subproblems, dealing with the heterogeneous and dynamic profiles associated with long-standing queries. However they share the weak points of other agents who perform no active autonomous search, and therefore cannot improve on the limitations of the metasearch engines they exploit.

Fish Search [35] is a search system inspired by some of the same ideas from artificial life that motivated the research in this thesis. Fish Search is based on a population of search agents who browse the Web autonomously, driven by an internally generated energy measure based on relevance estimations. The population is client-based, and uses a centralized cache for efficiency. While we believe that the algorithm could be extended to allow for distributed implementations, each agent cannot internalize local context. This is due to a fixed, nonadaptive strategy: a mixture of depth-first-, breadth-first-, and best-first-search, with user-determined depth and breadth cutoff levels. One difficulty of the Fish Search approach is in determining appropriate cutoff levels *a priori,* possibly resulting in load-unfriendly search behaviors. Therefore Fish Search suffers from limitations that are in a sense opposite to those of all the previously discussed agents; all it does is search, but it cannot adapt to user or environmental conditions.

IV.B Limitations of the state of the art

When the Web was much smaller than today, it was already clear that the fast pace of development in network technology would make the field of distributed information sources an ideal area of applications for autonomous, intelligent agents. In the words of Booker:

> Intelligent agents are needed to interact with humans in virtual environments, provide adaptive interfaces that model user's preferences, and locate information on wide-area computer networks. In applications such as these, an agent must interact with "environments" in which spatial structure is not necessarily the most significant basis for organizing behavior. Modeling the kinds of intelligent behaviors needed for these environments is an exciting challenge [...][2]

We have seen in the previous section only a small fraction of the many (more or less aware) answers to Booker's call. Yet there are still many limitations in the ways that the current state of the art applications use the concepts of agents and environments. We believe that the ideas illustrated in the previous chapters of this thesis can provide us with new directions to improve on such limitations. Agents must be endowed with mechanisms to exploit the spatial, temporal, and personal dimensions of their environments. Before discussing how local selection and internalization can move us closer to this goal, let us look more closely at some of the problems that we consider most urgent, and why we believe they should be viewed as assets rather than limitations — or as computer scientists would say, as features rather than bugs.

IV.B.1 Linkage topology

Indexing can be described as the process of building a *statistical topology* over a document space. In the vector representation [159], documents and queries are viewed as vectors in very large feature spaces where each word corresponds to a dimension. Two documents are similar, or a document is relevant with respect to a query, if the angle between their respective vectors is small. A search engine will show similar documents next to each other, effectively creating on the fly a

---

[2] From Booker [20], pages 1–2.

topology based on their word statistics.[3] This is a very useful model because the user can immediately make assumptions about the contents of retrieved documents, for example about the fact that they contain certain words.

However, networked information environments may contain additional structure information, which can be used to provide browsing users (or agents) with helpful cues. Here we focus on linkage information that is at the basis of hypertext markup languages such as those used in the Web. One cannot submit to search engines queries like "Give me all documents k links away from this one," because the space to store such information would scale exponentially with k.[4]

While much linkage information is lost in the construction of indexes, it is there to be exploited by browsing users, who in fact navigate from document to document following links. We have argued that *linkage topology* — the spatial structure in which two documents are as far from each other as the number of links that must be traversed to go from one to the other — is indeed a very precious asset on the Web. Even in unstructured portions of the Web, authors tend to cluster documents about related topics by letting them point to each other via links, as confirmed by bibliometric studies of the Web [88]. Such linkage topology is useful inasmuch as browsers have a better-than-random expectation that following links can provide them with guidance — if this were not the case, browsing would be a waste of time!

Let us quantify the notion of value added by linkage topology. We have conjectured that such value can be captured by the extent to which linkage topology "preserves" relevance (with respect to some query). Imagine a browsing user or agent following a random walk strategy.[5] Define R as the conditional probability that following a random link from the current document will lead to a relevant

---

[3] Similar arguments apply to Web information retrieval systems based on other representations.

[4] Several search engines now allow such queries for $k = 1$.

[5] We make the conservative assumption of random walk to obtain a lower bound for the value added of linkage topology.

document, given that the current document is relevant. We call R *relevance autocorrelation.* And define G as the probability that any document is relevant, or equivalently the fraction of relevant documents. We call G *generality* (of the query) [157].

For the random browser, the probability of finding a relevant document is given by

$$\nu = \eta R + (1 - \eta) G,$$

where η is the probability that the current document is relevant. If linkage topology has any value for the random browser, then browsing will lead to relevant documents with higher than random frequency. In order for this to occur the inequality

$$\nu / G > 1$$

must hold, which upon simplifying for η is equivalent to

$$R/G > 1. \tag{IV.1}$$

Conjecture IV.1 is equivalent to the cluster hypothesis [179] under a hypertext derived definition of association. We can then express the linkage topology value added by defining the quantity

$$\Theta \equiv R/G - 1.$$

As a reality check, we have measured Θ for a few queries from a couple of search engines [114]. Relevance autocorrelation statistics were collected by counting the fraction of links, from documents in each relevant set, pointing back to documents in the set. Generality statistics were collected by normalizing the size of the relevant sets by the size of the collections. These are quite gross measurements, since they are based on the assumption that the sets returned by the search engines correspond to the relevant sets. But a retrieved set can be viewed as the relevant set for *some* query. Our conjecture about the value added by linkage topology is confirmed by the large values of Θ shown in Table IV.1. Note that

Query	Θ (Lycos)	Θ (Britannica Online)
`color`	2.8×10^3	6.1×10^3
`blindness`	2.1×10^4	1.1×10^2
`photography`	3.5×10^3	1.8×10^2
`einstein`	5.2×10^3	6.2×10^1
`bach`	7.5×10^3	7.8×10^1
`carcinoma`	1.5×10^4	1.3×10^2
`cinema`	1.1×10^3	1.0×10^2
`internet`	4.0×10^3	7.2×10^1
`evolution selection`	2.3×10^3	1.8×10^1
`red wine`	2.5×10^4	3.7×10^1
Mean	$(9 \pm 3) \times 10^3$	$(7 \pm 6) \times 10^2$

Table IV.1: Measures of Θ for ten queries submitted to Lycos [101] and Britannica Online [41]. The minimum score parameter used for Lycos was 0.1. Only full (absolute) HTTP references were considered for Lycos, and only Micropaedia and Macropaedia article references for EB. Multiple-term queries represent AND Boolean searches.

even though the documents in the Encyclopaedia Britannica are better structured (higher R), the value of linkage topology is driven down by the smaller size of the collection (higher G). Yet, all Θ values are significantly positive.

Linkage topology also has been considered by others in the context of the Web, with different motivations. Links have been used for enhancing relevance judgments [144, 184], incorporated into query formulation to improve searching [4, 166], and exploited to determine "hub" and "authority" pages for document categorization and discovery [28, 27].

If links constitute useful cues for navigation, they can be exploited by autonomous *browsing* agents just as they are by browsing users — indeed, even the dumbest of agents (random walkers) can exploit linkage information. In fact, the random walk model may turn out to be more than just a lower bound for browsing behavior. Huberman *et al.* [72] argue that it is a very good predictive model of human browsing behavior. They assume that the *value* (e.g., relevance) of pages along the browsing path of a user follows a random walk of the form:

$$V_L = V_{L-1} + \xi_L \tag{IV.2}$$

where L is the depth along the path and ξ_L is a random variable drawn from a normal distribution $\aleph(\mu, \sigma^2)$. Equation IV.2 is a stronger than Equation IV.1, since it implies a positive correlation between V_L and V_{L-1} (equivalent to our relevance autocorrelation) for any $\mu > 0$. Huberman *et al.* find that the inverse-gaussian distribution of surfing depth (clicks per Web site) derived from Equation IV.2 accurately fits experimental data on surfing behavior, and therefore they call such a distribution a universal *law of surfing.* Although our conjecture on the value of linkage topology is more modest, it finds strong support in these findings. Furthermore, the random-walk assumption implies normative models for constructing browsing agents who make optimal local decisions about when to stop surfing, in much the same way in which real options are evaluated in financial markets [100]. Thus we feel justified in our confidence that browsing is not an unreasonable task for autonomous agents.

Note that the information encoded by statistical (word) versus linkage topologies are quite distinct, and arguably complimentary. Links, constructed manually to point from one page to another, reflect an author's attempts to relate his/her writings to others'. Word topology is an epiphenomenal consequence of word vocabulary choices made by many authors, across many pages. By making agents *perceptually* sensitive to word topology features and capable of *acting* by traversing link topology, we expect to find interesting relationships between the purposeful, manual linkage of Web authors and the words they use.

IV.B.2 Scalability

As we discussed in Section IV.A.3, scalability is a major issue limiting the effectiveness of search engines. The factors contributing to the problem are the large size of the Web, its rapid growth rate, and its highly dynamic nature. The

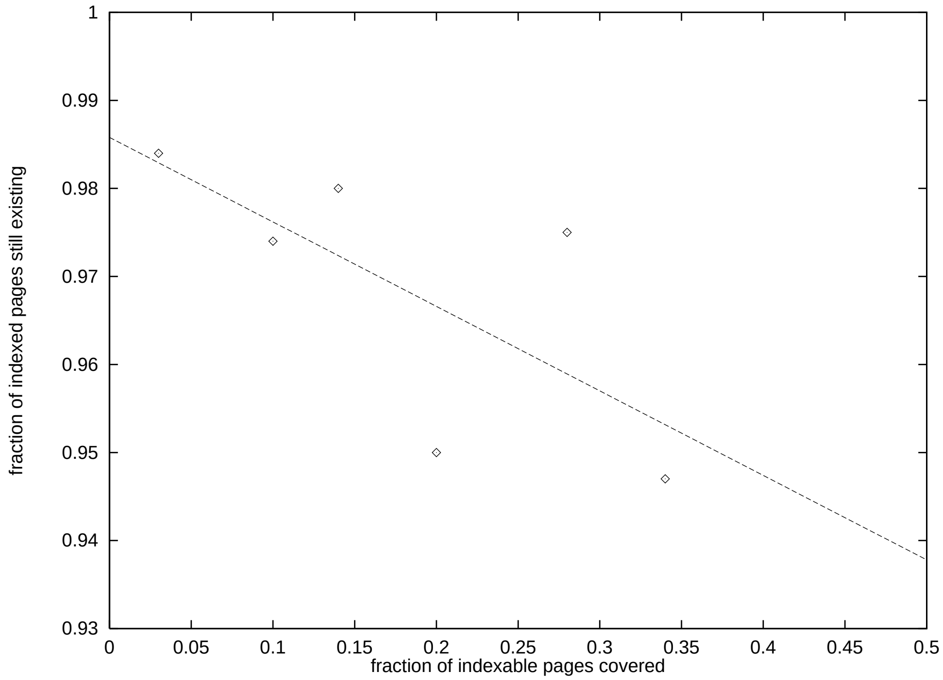


Figure IV.1: Scatter plot of coverage versus recency in six popular search engines: Alta Vista, HotBot, Northern Lights, Excite, InfoSeek, and Lycos. Data from Lawrence and Giles [89]. Linear regression is also shown. The correlation coefficient is -0.7.

scalability problem is quantified in the recent study by Lawrence and Giles [89]. Their estimates of the current size (over 320 million pages) and growth rate (1000% in a few years) of the Web attest to this environment's increasing complexity.

Lawrence and Giles also measure the coverage and recency of six among the most popular search engines. The coverage achieved by these search engines varies approximately between 3% and 34% of the Web's indexable pages. A higher bound on recency was obtained by counting the fraction of returned hits corresponding to broken URLs, i.e., pages that have been deleted or moved.[6] Among the search engines considered, the one with highest coverage is also the one with lowest recency, and viceversa — the engine with lowest coverage has the least bro-

---

[6] URLs with changed content do not appear broken, therefore this method only detects part of the stale information in an index.

ken links. This trade-off between coverage and recency is illustrated in Figure IV.1. Coverage and recency are indeed anti-correlated, as expected. Increasing the coverage of an index, given some limited bandwidth resource, imposes a search engine's crawler to "spread itself thin" and update pages less frequently, thus increasing the amount of stale information in the index.

In order to keep indexes as up-to-date as possible, crawlers have to revisit documents often to see if they have been changed, moved, or deleted. Further, crawlers have to try to exhaustively visit every new document to keep indexes as complete as possible. Such crawler behaviors impose significant loads on the net, as documents must be examined periodically. Heuristics are used to estimate how frequently a document is changed and needs to be revisited, but the accuracy of such statistics is highly volatile. The network load scales as n/τ, where n is the number of documents in the Web and τ if the time scale of the index, i.e. the mean time between visits to the same document. The longer τ, the more stale information in the index. If q is the number of queries answered by the search engine per unit time, then the amortized cost of a query scales as $n/q\tau$.

Agents searching the Web *on-line* do not have a scale problem because they search through the *current* environment and therefore do not run into stale information. On the other hand, they are of course less efficient than search engines because they cannot amortize the cost of a search over many queries. Assuming that users may be willing to cope with the longer wait for certain queries that search engines cannot answer satisfactorily, one might ask, *What is the impact of on-line search agents on network load?*

In our opinion, because of the scale effect, making an index less up-to-date can free up sufficient network resources to completely absorb the impact of on-line searches. Consider increasing the τ of a search engine by a factor of $(1+\epsilon)$, allowing the information in the index to become correspondingly more stale. Maintaining a constant amortized cost per query, we could now refine the results of each query with an on-line search using an amount of network resources scaling as

$$\frac{n}{q\tau} - \frac{n}{q\tau(1+\epsilon)} \sim \frac{n}{q\tau}\frac{\epsilon}{1+\epsilon}.$$

As an example, imagine visiting 100 Web pages on-line for each query, and accepting $\epsilon = 1$ (bringing τ, say, from one to two weeks). This could be achieved without impacting network load by satisfying the condition $n/q\tau = 200$. Assuming $q\tau$ (the number of queries posed over a constant time interval) is a constant, the current growth of the Web assures that the condition will be met very soon. For Alta Vista, at the time of this writing we estimate $n/q\tau \approx 5$ [39]; even at a conservative growth rate of a doubling per year, the condition would be met within about 5 years.[7] This simple argument, in our opinion, shifts the question: we should not ask what is the network impact of on-line search agents, but rather, *What* ϵ *achieves an appropriate balance between the network loads imposed by search engines crawlers and on-line agents?*

IV.B.3 Context

All samples of language, including the documents indexed by Web search engines, depend heavily on *shared context* for comprehension. A document's author makes assumptions, often tacit, about their intended audience and when this document appears in a "traditional" medium (conference proceedings, academic journal, etc.) it is likely that typical readers will understand it as intended. But one of the many things the Web changes is the huge new audience it brings for documents, many of whom will *not* share the author's intended context.

These vague linguistic concerns have concrete manifestation in the *global* word frequency statistics collected by Web search engines. The utility of an index term, as a discriminator of relevant from irrelevant items, can become a muddy

---

[7] If we consider the coverage factor of 3 due to the discrepancy between the n of the search engine and the actual size of the Web, the condition will be met even sooner.

average of its application across multiple, distinct sub-corpora within which these words have more focused meaning [170, 169].

Situated agents, on the other hand, can rely on local coherence in keyword distributions by exploiting their structural (link) proximity. Over time, agents may come to internalize the features that best describe the current documents and discriminate between relevant and other pages. For example, agents browsing through pages about "rock climbing" and "rock'n'roll" should attribute different weights to the word "rock" depending on whether the query they are trying to satisfy is about music or sports. Where an agent is situated in the environment provides it with the *local context* within which to analyze word meanings — a structured, situated approach to polisemy. Conversely, the words that surround links in a document provide an agent with valuable information to evaluate links and thus guide its path decisions — a statistical approach to action selection.

Indexes are also constructed without knowledge of the particular queries that they will answer, or of the users posing them. A universal ranking scheme may be generally good but probably will not be the best for each specific query or particular user. Conversely, *personal* agents may adapt to a user's interests, even if they change over time. They can internalize the user's preferences with respect to, e.g., vocabulary, word disambiguation, and relative importance of terms.

Localization and *personalization* are two aspects of context usage that search engines cannot provide. Situated, personal agents can capture features of local and personal contexts and adapt to them to improve on their search performance. The following sections discuss how to construct such agents.

IV.C InfoSpiders overview

Let us operationalize the ideas discussed in the previous section into an agent framework. Our goal is to address the limitations of search engines with respect to scalability, personalization, and personalization, by an agent-based algorithm and adaptive representation taking advantage of both the statistical and linkage topology of the distributed information environment. We have argued that such agents must be *autonomous, on-line, situated, personal browsers.*

Our approach to achieve these properties is based on the idea of a *multi-agent* system. The problem is decomposed into simpler subproblems, each addressed by one of many simple agents performing simple operations. The divide-and-conquer philosophy drives this view. Each agent will "live" browsing from document to document on-line, making autonomous decisions about which links to follow, and adjusting its strategy to both local context and the personal preferences of the user. Population-wide dynamics will bias the search toward more promising areas.

In this framework, both individual agents and populations must *adapt.* Individually learned solutions (e.g., by reinforcement learning) cannot capture global features about the search space or the user. They cannot "cover" heterogeneous solutions without complicated internal models of the environment; such models would make the learning problem more difficult. On the other hand, if we allowed for population-based adaptation alone (e.g., by an evolutionary algorithm), the system might be prone to premature convergence. Genetically evolved solutions would also reflect an inappropriate coarseness of scale, due to individual agents' incapability to learn during their life. These are the same reasons that have motivated the hybridization of genetic algorithms with local search [63], and reflect the general problem of machine learning techniques in environments with very large feature space dimensionalities (cf. Section IV.A.1).

The approach and methods introduced above have been applied in the construction of populations of adaptive information agents. The *InfoSpiders* system was implemented to test the feasibility, efficiency, and performance of adaptive, on-line, browsing, situated, personal agents in the Web. In this section we describe the InfoSpiders implementation and in particular discuss the details of the distributed evolutionary algorithm and agent representation used.

IV.C.1 Algorithm

InfoSpiders search on-line for information relevant to the user, by making autonomous decisions about what links to follow. How long should an agent live before being evaluated? What global decisions can be made about which agents should die and which should reproduce, in order to bias the search optimally? No answer to these questions would appear satisfactory. Fortunately, the local selection algorithm provides us with ways to remain agnostic about these questions. The InfoSpiders algorithm follows closely the model of Figure II.1. It is reproduced in Figure IV.2, for ease of reference and with more details about the specific InfoSpiders implementation of the local selection algorithm.

A central part of the system is the use of relevance feedback. The user may assess the relevance of (some of) the documents visited by InfoSpiders up to a certain point. Such relevance assessments take place asynchronously with respect to the on-line search, and alter the subsequent behaviors of agents on-line by changing the energy landscape of the environment. The process is akin to the replenishment of environmental resources; the user interacts with the environment to bias the search process. Let us first overview the algorithm at a high level, while representation-dependent details will be given in the next subsections.

The user initially provides a list of keywords and a list of starting points, in the form of a bookmark file. This list could typically be obtained by consulting a search engine. First, the population is initialized by pre-fetching the starting documents. Each agent is "positioned" at one of these document and given a random behavior (depending on the representation of agents) and an initial reservoir of energy.

In step (1), an agent "senses" its local neighborhood by analyzing the text of the document where it is currently situated. This way, the relevance of all neighboring documents — those pointed to by the hyperlinks in the current document — is estimated. Based on these link relevance estimates, in step (2) the agent "moves" by choosing and following one of the links from the current

```
initialize $p_0$ agents, each with energy $E = \frac{\theta}{2}$
while there are alive agents:
    for each alive agent $a$:
        1. pick link from current document
        2. fetch new document $D$
        3. $E_a \leftarrow E_a - c(D) + e(D)$
        4. Q-learn with reinforcement signal $e(D)$
        5. selection:
            if ($E_a \geq \theta$)
                $a' \leftarrow mutate(recombine(clone(a)))$
                $E_{a'} \leftarrow E_a/2$
                $E_a \leftarrow E_a/2$
            else if ($E_a \leq 0$)
                $die(a)$
            end
    end
    process optional relevance feedback from user
end
```

Figure IV.2: High-level pseudocode of the InfoSpiders evolutionary algorithm for distributed information agents.

document.

In step (3), the agent's energy is updated. Energy is needed in order to survive and move, i.e., continue to visit documents on behalf of the user. Agents are rewarded with energy if the visited documents appear to be relevant. The $e()$ function is used by an agent to evaluate the relevance of documents. There are two energy "sources" in the system. If a document D had previously been visited *and* assessed by the user, the user's assessment is used to compute $e(D)$ ($e(D) > 0$ or $e(D) < 0$); if the document had not been visited before, its relevance must be estimated to compute $e(D)$. If D had previously been visited but *not* assessed, $e(D) = 0$ because that resource has been consumed.

This mechanism is implemented via a cache, which also speeds up the process by minimizing duplicate transfers of documents. While in the current, client-based implementation of InfoSpiders this poses no problem, caching is a form of communications and thus a bottleneck for the performance of distributed agents. In a distributed implementation, we imagine that agent will have local caches. When using the current implementation to simulate the performance of distributed InfoSpiders, we will simply assume that no cache is used, setting the cache size to zero.

There is only one energy "sink" in the system: agents are charged energy costs for the network load incurred by transferring documents. The cost function $c()$ should depend on used resources, for example transfer latency or document size. For simplicity we will assume a constant cost for accessing any new document, and a (possibly smaller) constant cost for accessing the cache; this way stationary behaviors, such as going back and forth between a pair of documents, are discouraged.

Just as for graph search, instantaneous changes of energy are used, in step (4), as reward/penalty signals. This way agents adapt during their lifetime by Q-learning [183]. This adaptive process allows an agent to modify its behavior based on prior experience, by learning to predict the best links to follow.

In step (5), an agent may be killed or be selected for reproduction. In the latter case offspring are recombined by the use of one of two types of crossover. In *local crossover,* an agent can only recombine with agents residing on the same document, if there are any.[8] In *panmictic crossover,* an agent may recombine with any other agent in the population; in this case the mate is selected at random. Local crossover is appropriate for distributed InfoSpiders to minimize communication overhead and will be used in the remainder of the chapter. Offspring are also mutated, providing the variation necessary for adapting agents by way of evolution.

Finally, the user provides the system with relevance feedback. This process is optional and can take place without direct on-line interactions with the agents. The user may assess any visited document D with feedback $\phi(D) \in \{-1, 0, +1\}$. All the words in the document are automatically assessed by updating

---

[8] A more sensible alternative for distributed implementations would be to allow crossover between agents situated on the same server.

a "feedback list" of encountered words. Each word in this list, k, is associated with an integer count ω_k that is initialized with 0 and updated each time any document is assessed by the user:

$$\forall k \in D : \omega_k \leftarrow \omega_k + \phi(D).$$

The word feedback list is maintained to keep a global profile of which words are relevant to the user.

The output of the algorithm is a flux of links to document, ranked according to some relevance estimate — modulo relevance assessments by the user. The algorithm stops when the population goes extinct for lack of relevant information resources, or if it is terminated by the user.

IV.C.2 Agent architecture

Figure IV.3 illustrates the architecture of each InfoSpiders agent. The agent interacts with the information environment, that consists of the actual networked collection (the Web) plus data kept on local disks (e.g., relevance feedback data and cache files). The user interacts with the environment by accessing data on the local client (current status of a search) and on the Web (viewing a document suggested by agents) and by making relevance assessments that are saved locally on the client and will be accessed by agents as they subsequently report to the user/client. There is no direct interaction between the user and the agents after the initial submission of the query and starting points.

The InfoSpiders prototype consists of over 14,000 lines of C code and runs on UNIX and MacOS platforms. The Web interface is based on the `W3C` library [181]. Agents employ standard information retrieval tools such as a filter for noise words [48] and a stemmer based on Porter's algorithm [49]. Finally, agents store an efficient representation of visited documents in the shared cache on the client machine. Each document is represented by a list of links and stemmed keywords. If the cache reaches its size limit, the LRU (least recently used) replacement strategy is used. Figure IV.4 shows a screen shot of the graphical user interface for the

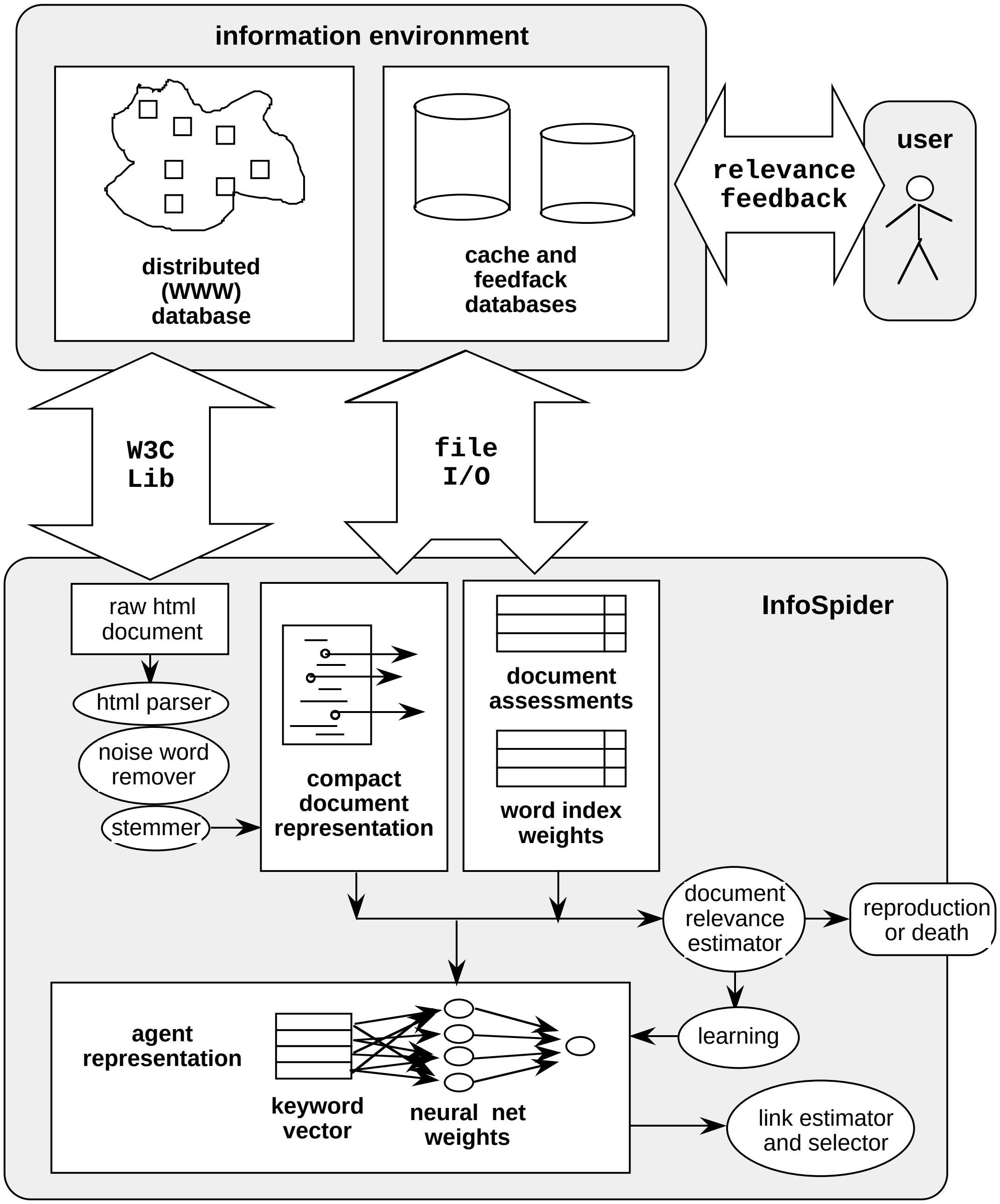


Figure IV.3: Architecture of an InfoSpiders agent.

Macintosh version of InfoSpiders.

IV.C.3 Adaptive agent representation

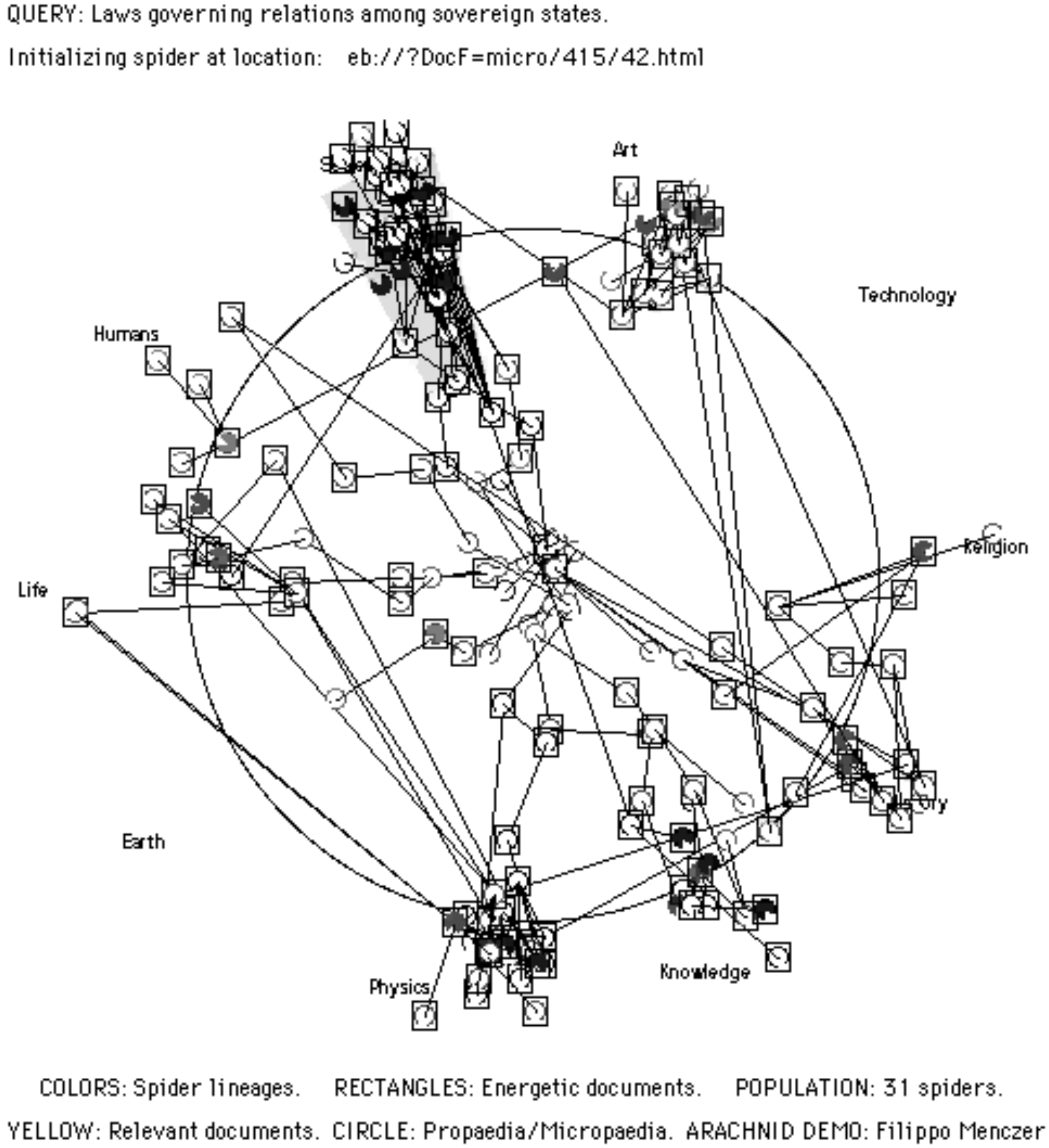
Figure IV.3 highlights the central dependence of the InfoSpiders system on agent representation. We now describe the adaptive representation of InfoSpiders. This consists of the genotype, that determines the behavior of an agent and is passed on to offspring at reproduction; and of the actual mechanisms by which the genotype is used for implementing search strategies.

The first component of an agent's genotype consists of the parameter $\beta \in \Re^+$. Roughly, it represents the degree to which an agent trusts the descriptions that a page contains about its outgoing links. β is initialized with β_0.

Each agent's genotype also contains a list of keywords, initialized with the query terms. Since feed-forward neural nets are a general, versatile model of adaptive functions, we use them as a standard computation device — just as we did in LEE. Therefore genotypes also comprise a vector or real-valued weights, initialized randomly with uniform distribution in a small interval $[-w_0, +w_0]$. The keywords represent an agent's opinion of what terms best discriminate documents relevant to the user from the rest. The weights represent the interactions of such terms with respect to relevance. The neural net has a real-valued input for each keyword in its genotype and a single output unit. We want to allow the inputs and activation values of the network to take negative values, corresponding to the possibly negative correlations perceived between terms and relevance. For this reason the network uses the hyperbolic tangent as its squashing function, with inputs and activation values in $[-1, +1]$. Let us now see how the different parts of the system are implemented, based on the this representation.

Action Selection

An agent performs action selection by first computing the relevance estimates for each link from the current document. This is done by feeding into the



QUERY: Laws governing relations among sovereign states.

Initializing spider at location: eb://?DocF=micro/415/42.html

COLORS: Spider lineages. RECTANGLES: Energetic documents. POPULATION: 31 spiders.

YELLOW: Relevant documents. CIRCLE: Propaedia/Micropaedia. ARACHNID DEMO: Filippo Menczer

Figure IV.4: Screen shot of the InfoSpiders window on a Macintosh. The search environment in this example is the one used for evaluation purposes (see Section IV.D.2).

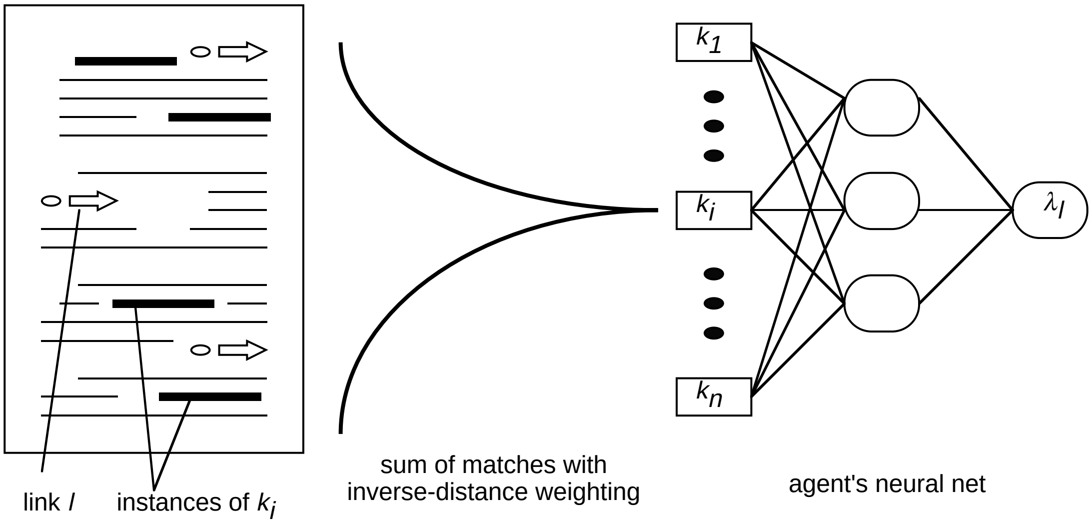


Figure IV.5: How an agent estimates each link from the current document. For each link in the document, each input of the neural net is computed by counting the document words matching the keyword corresponding to that input, with weights that decay with distance from the link.

agent's neural net activity corresponding to the small set of (genetically specified) keywords to which it is sensitive. Each input unit of the neural net receives a weighted count of the frequency with which the keyword occurs in the vicinity of the link to be traversed. In the experiments reported here, we use a distance weighting function which is biased towards keyword occurrences most close to the link in question.

More specifically, for link l and for each keyword k, the neural net receives input:

$$in_{k,l} = \sum_{i:dist(k_i,l)\leq\rho} \frac{1}{dist(k_i,l)}$$

where k_i is the ith occurrence of k in D and $dist(k_i,l)$ is a simple count of other, intervening links (up to a maximum window size of $\pm\rho$ links away). The neural network then sums activity across all of its inputs; each unit j computes activation

$$\tanh(b_j + \sum_k w_{jk} in_k^l)$$

, where b_j is its bias term, w_{jk} are its incoming weights, and in_k^l its inputs from the lower layer. The output of the network is the activation of the output unit, λ_l. The process is illustrated in Figure IV.5 and is repeated for each link in the current document. Then, the agent uses a stochastic selector to pick a link with probability distribution:

$$\Pr[l] = \frac{e^{\beta\lambda_l}}{\sum_{l' \in D} e^{\beta\lambda_{l'}}}.$$

Relevance estimation and feedback

After a link has been chosen and the corresponding new document has been visited, the agent has to determine the corresponding energy gain and loss; both depend on whether or not the document had been visited previously. If the document is in the cache, and the user has assessed its relevance, then the agent receives energy $e(D) = \phi(D)$, after which $\phi(D)$ decays according to

$$\phi(D) \leftarrow \gamma\phi(D)$$

where the decay factor γ is a parameter. This is done to avoid a population explosion due to non-conserved energy from relevance feedback.

If the user provided the system with relevance assessments, the word feedback list represents a profile of his/her interests that is both more current and more accurate than the original query. This list is used to estimate the relevance of previously unvisited or not assessed documents, so that the corresponding energy intake can be computed:

$$e(D) = \tanh\left(\sum_{k \in D} freq(k, D) \cdot I_k\right)$$

where $freq(k, D)$ is the frequency of term k in document D normalized by document size, and I_k is the weight of term k based on relevance feedback. The latter is an extension of the TFIDF (term frequency-inverse document frequency) index weighting scheme. If k is not in the word feedback list, then $I_k = 0$. The list is initialized with the query terms and weights $I = 1$. It changes only if the user provides InfoSpiders with relevance feedback. In this case the weights are updated according to the rule:

$$I_k \leftarrow \alpha \cdot I_k + (1 - \alpha) \cdot \omega_k \cdot \left[1 + \log\left(\frac{1}{C_k}\right)\right] \tag{IV.3}$$

where C_k is the fraction of cache documents containing k and α is an inertia term.

Such a weighting formula differs from more traditional TFIDF schemes [164] in at least two respects. First, it is not aimed at weighting terms based on how well they describe documents, but rather on how well they correlate with relevance. Therefore it employs algebraic term frequencies to account for negative contributions from documents that contain the term but are anti-correlated with relevance. Second, it is computed on-line and therefore uses document frequencies based on the contents of the cache rather than the entire collection. The hyperbolic tangent is used to normalize energy intakes into the appropriate range $[-1, +1]$ — the same range as the corresponding neural net's prediction.

Q-learning

The agent then compares the relevance (assessed or estimated) of the current document with the estimate of the link that led to it. By using the connectionist version of Q-learning [96], the neural net can be trained on-line to predict values of links based on local context. After the agent visits document D, $e(D)$ is used as an internally generated reinforcement signal to compute a teaching error:

$$\delta(D) = e(D) + \mu \cdot \max_{l \in D}\{\lambda_l\} - \lambda_D \tag{IV.4}$$

where μ is a future discount factor and λ_D the prediction from the link that was followed to get to D. The neural net's weights are then updated by back-propagation of error [153]. Learned changes to the weights are "Lamarckian" in that they are inherited by offspring at reproduction.

In the absence of relevance assessments, this reinforcement learning algorithm is unsupervised because it is the environment that provides the reinforcement signal $e(D)$. However, relevance feedback alters the function $e()$ under the supervision of the user, who modifies the environment by providing examples of relevance. Therefore InfoSpiders integrate unsupervised and supervised adaptation in the form of evolution, Q-learning, and relevance feedback.

Reproduction

At reproduction, the offspring clone is recombined with another agent. Two-point crossover is applied to the keywords of the clone, so that a subset of the mate's keywords is spliced into the offspring's keyword vector.

Then the offspring is mutated to provide the evolutionary algorithm with the necessary power of exploration. If a' is an offspring of a:

$$\beta_{a'} \leftarrow U[\beta_a(1-\kappa_\beta), \beta_a(1+\kappa_\beta)]$$

where $\kappa_\beta \in [0,1]$ is a parameter and U is the uniform distribution. The values of β are clipped to β_{max} to maintain some exploratory behavior. The neural net is mutated by adding random noise to a fraction ζ_w of the weights. For each network connection i:

$$w^i_{a'} \leftarrow U[w^i_a(1-\kappa_w), w^i_a(1+\kappa_w)].$$

The keyword vector is mutated with probability ζ_k. The least useful (discriminating) term $\arg\min_{k\in a'}(|I_k|)$ is replaced by a term expected to better justify the agent's performance with respect to the user assessments. In order to keep any single keyword from taking over the whole genotype, this mutation is also stochastic; a new term is selected with probability distribution

$$\begin{aligned} \Pr[k] &\propto freq(k,D)\cdot\delta_{<1}(|I_k|+\chi) \qquad \text{(IV.5)} \\ \delta_{<1}(x) &\equiv \begin{cases} x & \text{if } x<1 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

where D is the document of birth and $\chi \in [0,1]$ is a parameter. The first factor captures the local context by selecting a word that well describes the document that led to the energy increase resulting in the reproduction. The second factor captures the global context set by the user by selecting a word that well discriminates the user's preferences. The parameter χ regulates the amount of supervised (small χ) versus unsupervised (large χ) keyword mutation; if $\chi = 0$, only keywords important to the user can be internalized, while if $\chi > 0$ new keywords can be internalized based on local environmental context alone. Learning will take care of adjusting the neural net weights to the new keyword.

The evolution of keyword representations via local selection, mutation and crossover implements a form of *selective query expansion.* Based on relevance feedback and local context, the query adapts over time and across different places. The population of agents embodies a distributed, heterogeneous model of relevance that may comprise many different and possibly inconsistent features. But each agent focuses on a small set of features, maintaining a well-defined model that remains manageable in the face of the huge feature dimensionality of the search space.

IV.D Experimental setting

IV.D.1 Algorithmic parameters

In Section IV.E we report on results of experiments and analysis aimed at evaluating the performance of InfoSpiders in responding to queries by searching the Web on-line. Unless otherwise stated, the algorithmic parameters discussed in Section IV.C will take the default values shown in Table IV.2.

Unless otherwise stated, we will assume that the cache size is large enough to contain all the visited documents. The cost charged for visiting a document will depend on the experiment. For the sequential InfoSpiders implementation, The cost is higher for new documents (c_n) than for cached documents (c_o). For

Parameter	**Default**	**Description**
p_0	21	Initial population size
θ	2.0	Reproduction threshold
c_n	0.01	Energy cost per new document
c_o	0.0001	Energy cost per cached document
c_d	0.001	Energy cost per document (distributed case)
T_{max}	10,000	Max number of new pages visited per query
β_0	2.0	Initial β
κ_β	0.5	β mutation range
β_{max}	5.0	Max β
ρ	5	Half-size of link estimation sliding window
ζ_k	0.5	Keyword mutation rate
χ	0	Unsupervised keyword mutation factor
N_{layers}	2	Neural net layers (excluding inputs)
w_0	0.5	Initial neural net weight range
ζ_w	0.2	Neural net weight mutation rate
κ_w	0.25	Neural net weight mutations range
η	0.05	Neural net Q-learning rate
μ	0.5	Q-learning discounting factor
α	0.5	Inertia of word feedback weights
γ	0.9	Decay factor for document assessments
F_{max}	64	Max number of word feedback entries
$\|C\|$	T_{max}	Max cache size
ψ	0.1	Recall level for search length measurements

Table IV.2: InfoSpiders parameter descriptions and values for the experiments reported in Section IV.E.

distributed InfoSpiders (see Section IV.E.1) there is no difference; in this case the cost c_d is such that an agent can visit 1000 irrelevant documents before it will run out of energy. For each query, the search is stopped when the population meets the ψ success criterion explained in Section IV.E.1 (unless it goes extinct or visits a total of T_{max} new pages).

IV.D.2 EB search graph

The difficulty of evaluating on-line retrieval systems stems from multiple factors. The lack of a ranking function over all documents (in particular those not seen) is one problem that will be addressed in the next subsection. A more general difficulty is the lack of queries with available well-defined relevant sets. To overcome this problem, a special chunk of the Web has been selected as a test environment [170, 169]: the Encyclopaedia Britannica (EB) [41]. The advantage is that we can make use of readily available relevant sets of articles associated with a large number of queries.

Here we use a subset of the EB corpus, corresponding to the "Human Society" topic subtree — roughly one tenth of the whole collection. This environment is made of $N = 19427$ documents, organized in a hypertext graph. 7859 of these pages are articles constituting the *Micropaedia*. These, together with 10585 *Index* pages, form a connected graph. The remaining 983 nodes form a hierarchical topical tree, called *Propaedia*. Figure IV.6 show the in- and out-degree statistics of the search graph.

Articles are manually classified according to the Propaedia hierarchy by skilled human editors. By using the title of any Propaedia node as a query we have access to all the articles classified into that category by the editor, who has exhaustive knowledge of all the documents in that subject. Therefore we will use the set of Micropaedia articles associated with the *subtree rooted at the query node* as the relevant set corresponding to that query.

Agents are initially placed at the root of the "Human Society" tree. Since

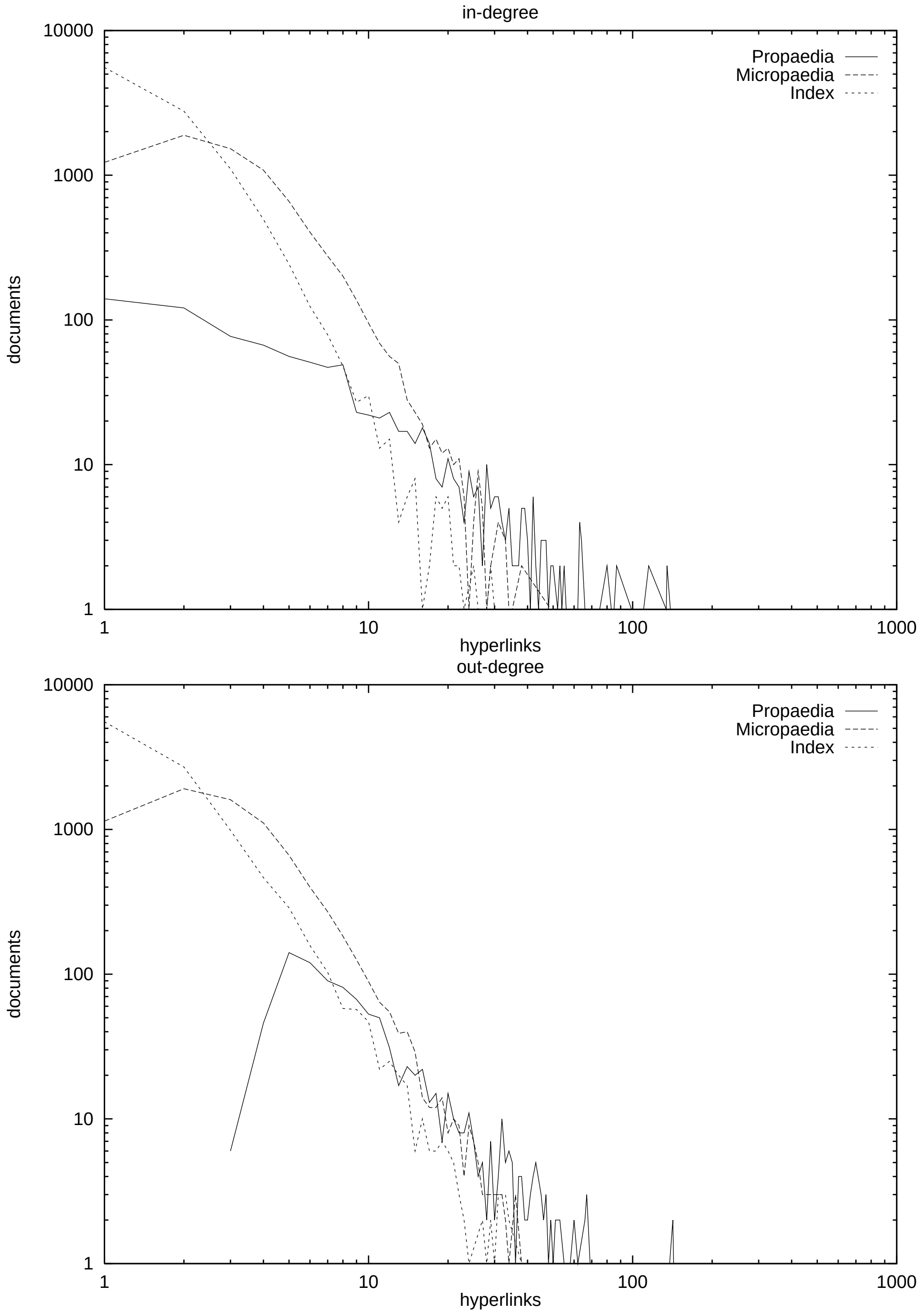


Figure IV.6: In-degree and out-degree distribution histograms for the EB Human Society graph.

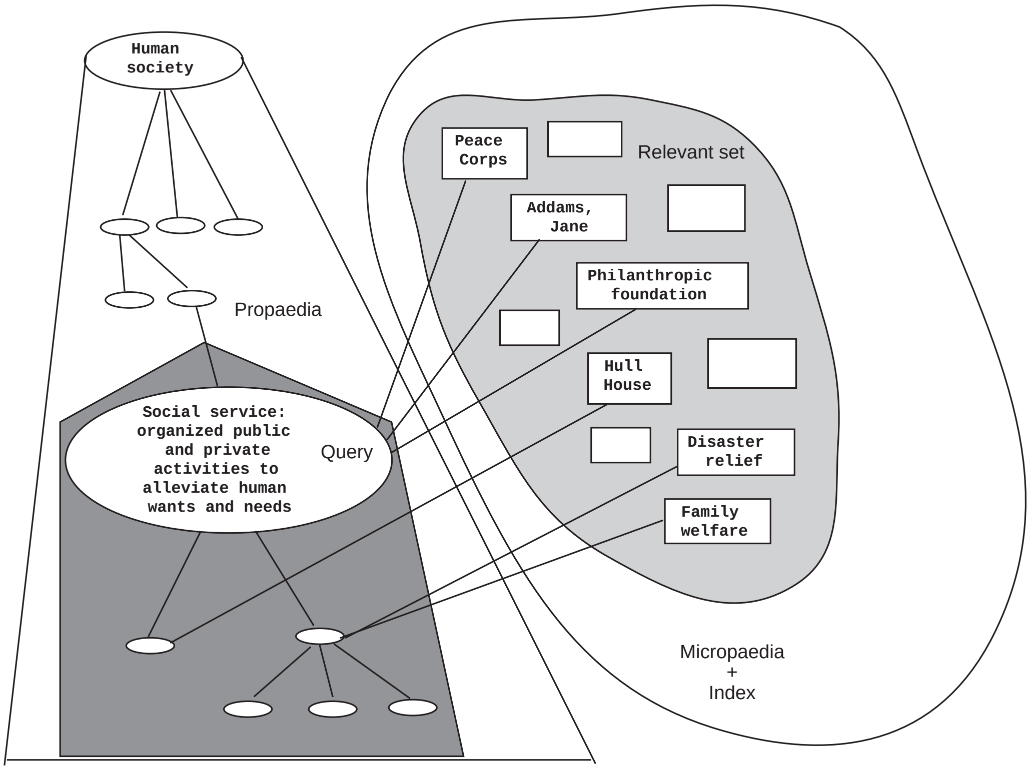


Figure IV.7: Schematic representation of the EB search space for an actual query, showing the titles of a few relevant articles. The relevant set is depicted in light gray. The subtree in dark gray is removed from the search space.

Depth	# Queries	# Measures	$\langle G \rangle$	Example
1	12	84	0.02	`Branches of Private Law, Substantive and Procedural`
2	72	144	0.003	`Laws governing economic transactions`
3	100	200	0.002	`Law of commercial transactions`
4	100	200	0.001	`Principal forms of business associations`
5	12	84	0.0007	`State and municipal corporations, quasi-public enterprises and utilities`

Table IV.3: Propaedia query statistics and examples. Generality statistics do not account for the removal of relevant Propaedia subtree nodes.

the Propaedia topology is used to define relevance, its "visibility" to agents during search would make the problem of navigating through the relevant set quite easy. Therefore, unless otherwise stated, we remove from the search space all the Propaedia nodes in the *relevant subtree;* agents cannot access relevant nodes directly from the Propaedia, but only from Index or other Micropaedia nodes. This is illustrated schematically in Figure IV.7 for the example query used in Section IV.E.2.

Table IV.3 shows some statistics and examples for the queries used in the experiments of this chapter (unless otherwise stated). Queries corresponding to nodes at the same depth in the Propaedia tree are grouped together. Queries span five different depths. If d is the depth of a query, the minimal distance from the staring node (the Human Society root) to the root of the relevant Propaedia subtree is $d+1$. However, since the latter is not part of the search graph, the actual shortest path to a relevant article is at least $d+3$. Deeper queries are harder for InfoSpiders because they are less general (lower carrying capacity) and their relevant sets are farther from the starting point. The number of different queries available also varies with depth. We selected queries with sufficiently large relevant sets to allow for statistically significant performance evaluations (see Section IV.E.1). In order to decrease standard errors in performance statistics, multiple measurements are obtained from a single query by restarting the search with different initial conditions (i.e., different seeds for the random number generator).

IV.E Experiments

Evaluation of a system like InfoSpiders is complex and must be carried out at different levels. At a high level, we must be confident that the system performs significantly better than simpler strategies against which it can be compared. For this type of *macro* analysis, we look at the collective performance of the population as a whole. At a lower level, we want to verify that agents internalize appropriate cues of their local environments into their behaviors. For this type of *micro* analysis, we consider the representations of individual agents.

IV.E.1 Macro analysis

How do we quantitatively evaluate the behavior of the InfoSpiders algorithm on the EB corpus? Since InfoSpiders do not have access to the whole corpus, but only to the subset of documents they actually visit, they cannot impose a ranking over the whole collection. This makes it impossible to use standard information retrieval performance analysis methods such as precision-recall curves, which require rank or similar measures as a control parameter [179].

Another metric lends itself better to assess the performance of on-line retrieval systems. *Search length* is normally defined as the number of irrelevant documents that appear, in some ordered list of retrieved documents, in front of some fraction of the relevant set [31]. We can easily extend this method by imagining that only visited documents appear in the list of retrieved documents, and that their ordering is given by visit time rather than rank — *length* then refers to waiting time. For an on-line retrieval system this means that we only need to wait until some fraction ψ of the relevant set is visited, and count the total number of documents visited up until that time. Search length is measured when $\lceil \psi \cdot G \cdot N \rceil$ relevant articles have been visited (cf. Table IV.2).

Another complication arises due to sequential versus potentially distributed implementations. In a sequential algorithm, all visited documents are retrieved from the servers on which they reside and analyzed locally on the client machine. In this case the central cache helps minimize the number of repeated document requests over the network. The performance of InfoSpiders will be compared with two sequential (centralized) algorithms. Like the sequential version of InfoSpiders, these can also take advantage of a centralized cache. Therefore they will be allowed to use an arbitrarily long central cache, so that only *previously unvisited documents* contribute to the measured search length. This will also be the case for *sequential InfoSpiders*.

However, we expect that the InfoSpiders algorithm will be implemented in a distributed fashion, and therefore can *simulate* its distributed execution. To this end, those parts of the system that require communication among agents must be disallowed, and more specifically: the shared access to the global cache, the differential costs for new and previously visited documents, and panmictic crossover. Therefore for (simulated) *distributed InfoSpiders* we disallow such interactions; we include each and every document visit into the search length statistics, whether the agent requests a new document across the network or loads a document from the cache.[9] Agents would then execute in parallel, and search length is given by the maximum number of documents visited by any agent *lineage*. This is measured in the experiments by keeping track of the number of links traversed by each agent, accumulated over all generations in the agent's ancestry since the start of a run. Local crossover occurs only if an agent reproduces at a location (document) where at least one other agent is situated.

---

[9] This is a worst-case scenario for InfoSpiders performance, since we can imagine that each agent could carry along a small local cache.

Notice that access to relevance feedback information could also be viewed as a source of indirect communication among agents, which could interfere with their parallelization potential. However, relevance assessments take place off-line and very infrequently from an agent's perspective. That is, their temporal scale is much longer than that of an agent's interaction with the environment. Therefore we imagine that agents will access relevance feedback information only when they periodically report to the client. Based on this assumption, we allow relevance feedback in distributed InfoSpiders.

InfoSpiders vs. breadth-first-search

In the first experiment search length is averaged across *all* queries in the "Human Society" tree, one run each (rather than multiple runs over the subset illustrated in Figure IV.3). The performance of sequential InfoSpiders is compared with that of a simple non-adaptive algorithm, *breadth-first-search.* Due to our implementation of breadth-first-search for this experiment, the search space is somewhat different than illustrated in Section IV.D.2. Index pages are excluded, and *all* Propaedia nodes are included. Breadth-first-search only traverses edges in the Propaedia tree, and stops when it finds the node corresponding to root of the relevant set (the query page). InfoSpiders are free to traverse all Propaedia and Micropaedia pages, but stop when the first relevant node is found, whether in the Propaedia or Micropaedia. Thus the search length corresponds to a recall level of $\psi = 1/GN$ (unlike the fixed value of Table IV.2).

InfoSpiders use reinforcement learning, but for a fair comparison no user feedback is provided. The results are shown in Figure IV.8: the improvement of InfoSpiders over breadth-first-search is as large as one order of magnitude. This result confirms that the InfoSpiders approach is viable and suggests comparisons with more elaborate algorithms than exhaustive search.

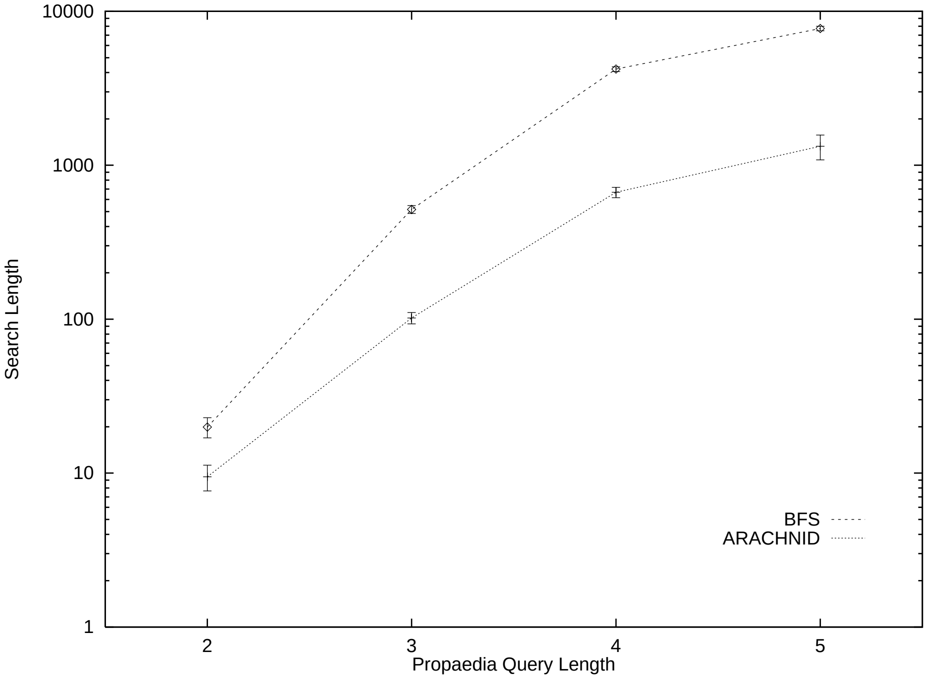


Figure IV.8: Search length of InfoSpiders and breadth-first-search for queries at different depths in the "Human Society" Propaedia category of Encyclopaedia Britannica. Error bars are standard errors of means over same-depth queries.

InfoSpiders vs. best-first-search

We now want to compare InfoSpiders with a good global search algorithm. When searching an annotated graph, the optimal strategy is given by the A^* algorithm [156]. However, in the case of the Web there is no admissible heuristic to apply A^*, since the only suitable lower bound on the distance from the closest relevant node is trivially zero. Therefore A^* reduces to *best-first-search.*

InfoSpiders can implement a search strategy similar to best-first-search by evolving high values for the β gene, but only from the local "perspective" of single agents, rather than with respect to the global search space. Nevertheless, assuming best-first-search as an upper bound for global search algorithms, we compare its performance with InfoSpiders.

We implemented best-first-search by means of a *priority queue.* Links are estimated in the same way in which they are by InfoSpiders, but then they are placed in a global queue, sorted by estimated relevance of the pointed documents; then documents are visited by following the queued links, in the order determined by their priority. The length of the priority queue is set equal to the initial size of the InfoSpiders population, p_0.

We have run experiments comparing the search length achieved by best-first-search with that of two variants of distributed InfoSpiders, one without relevance feedback and one with relevance feedback every 50 newly visited documents. In the latter case assessments of $\phi(D) = 1$ are automatically generated for all documents D visited so far that belong to the editor-defined relevant set. Everything else in the experiment is as described in Section IV.D.

Figure IV.9 shows the percentages of queries successfully completed by the different algorithms. These are the cases in which a fraction ψ of the relevant pages are found within the T_{max} limit on visited pages. Non-completed queries are those for which InfoSpiders run out of time, or go extinct; and for which priority queue runs out of time, or becomes engulfed in previously visited areas. As expected, performance degrades for queries of increasing depth. InfoSpiders

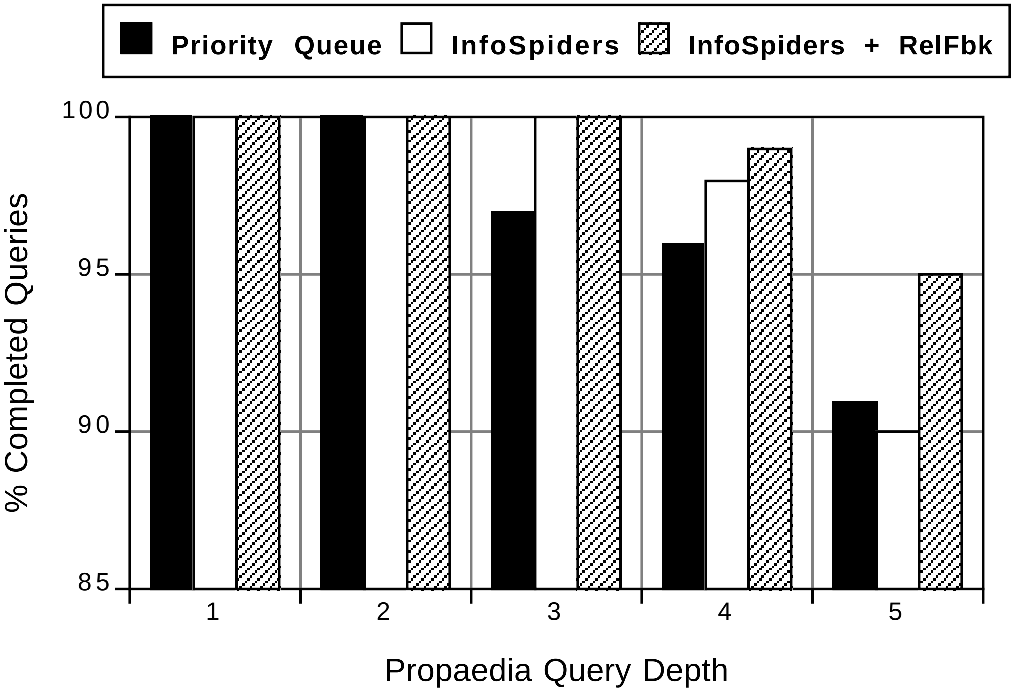


Figure IV.9: Percentage of queries successfully completed by best-first-search ("Priority Queue") and different versions of InfoSpiders.

however tend to have a higher success rate than best-first-search, with the exception of depth 5 for which best-first-search does better by 1%. Relevance feedback affords an improvement where unsupervised InfoSpiders fail to achieve 100% completion rate. This is due to the evolutionary reinforcement provided by relevance feedback: those agents that are moving in the right direction receive energy boosts and have a greater chance to reach the relevant clusters.

Figure IV.10 plots search length versus query depth. Search length is averaged over completed queries at the same depth. As the plots demonstrate, InfoSpiders' search length increases with depth while the global heuristic does not show such dependency within the observed depth range. As a result, best-first-search outperforms InfoSpiders for very deep queries. However, in practical cases we do not expect the search to start too far away from the desired pages thanks the use of search engines to seed the initial population of InfoSpiders. For the more realistic queries at lower depths, the distributed nature of InfoSpiders results in a

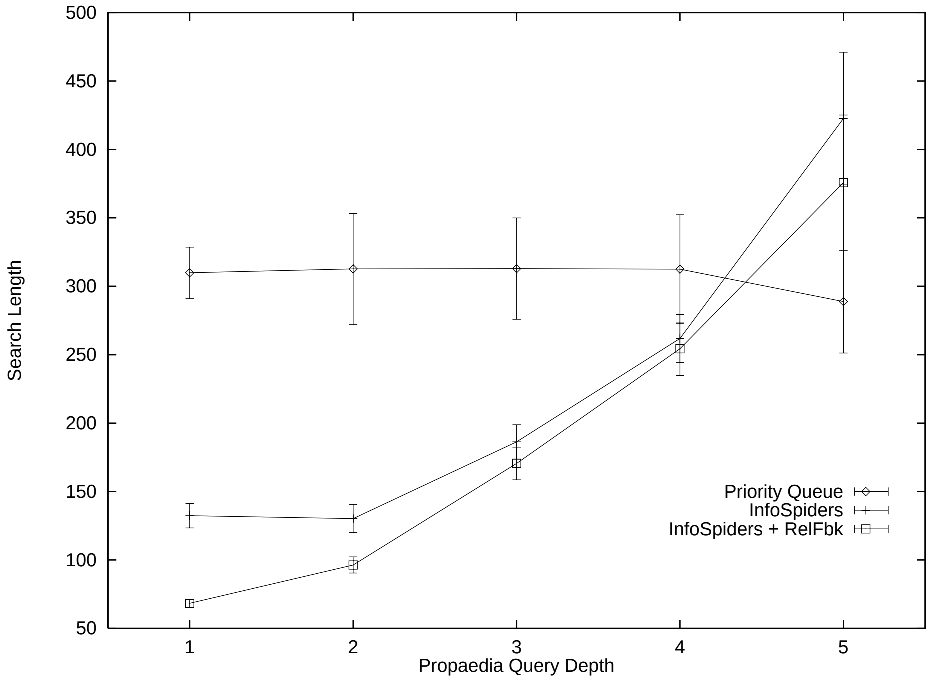


Figure IV.10: Performance of InfoSpiders (with and without relevance feedback) versus best-first-search. Error bars correspond to standard errors.

significantly shorter search time than required by best-first-search. Figure IV.10 also shows that when user relevance feedback is available it further accelerates the discovery of relevant documents, by pointing agents in the right direction. The improvement becomes less significant statistically for deep queries due to the fewer available measurements.

Population dynamics

Let us now look at the typical population dynamics of InfoSpiders in response to a single query. This is determined by the carrying capacity of the networked information environment, that is, by the size of the relevant set for the given query. Further, the different cost schemes associated with the sequential and parallel versions of InfoSpiders alter the selective bias of the algorithm and consequently the collective system behavior and performance. Finally, the size of the cache also affects the energetic balance of the environment. The cache works as

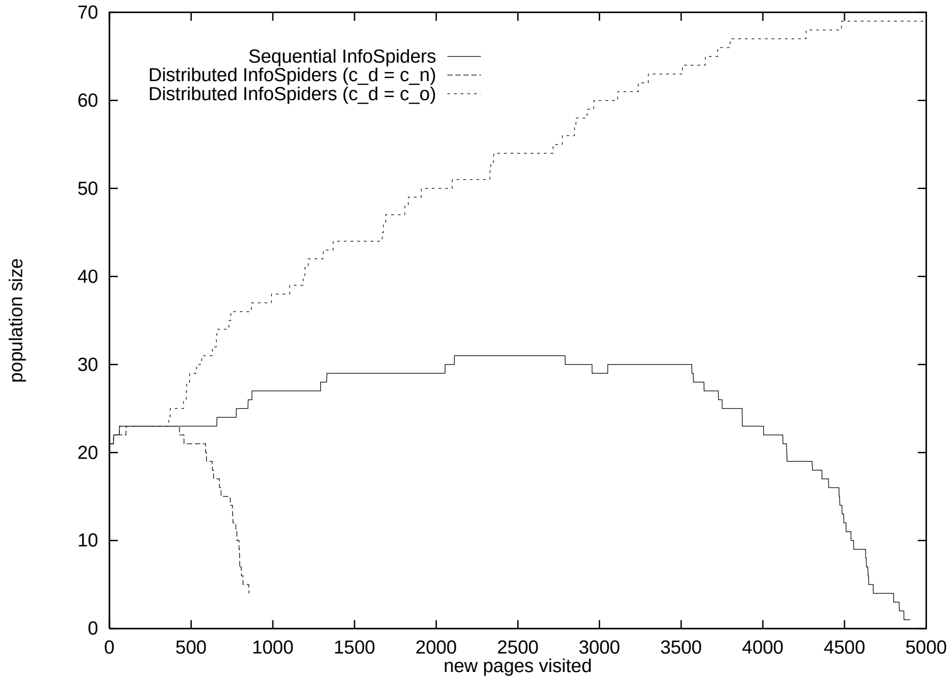


Figure IV.11: Population dynamics in three runs of sequential and distributed InfoSpiders with different cost parameters.

the collective memory of sequential InfoSpiders; when the system "forgets" pages that are replaced in the cache, the effect is equivalent to replenishing environmental resources in the local selection algorithm.

The query we consider is "`Social service: organized public and private activities to alleviate human wants and needs.`" For the experiments in this subsection we allow all Propaedia nodes to be visited.

To evaluate the effect of sequential versus distributed implementations of InfoSpiders on population dynamics and on performance, we compare three runs. In the first run, sequential InfoSpiders are employed; the ratio of the cost parameters is set to $c_o/c_n = 0.1$ (rather than 0.01 as in Table IV.2). In the second and third run, distributed InfoSpiders are employed; their cost parameter is set to $c_d = c_n$ and $c_d = c_o$ in the two cases, respectively. The value of c_n is set to 0.02.

Figure IV.11 plots population size as a function of the number of new pages visited. Predictably, the distributed case with $c_d = c_n$ has the smallest

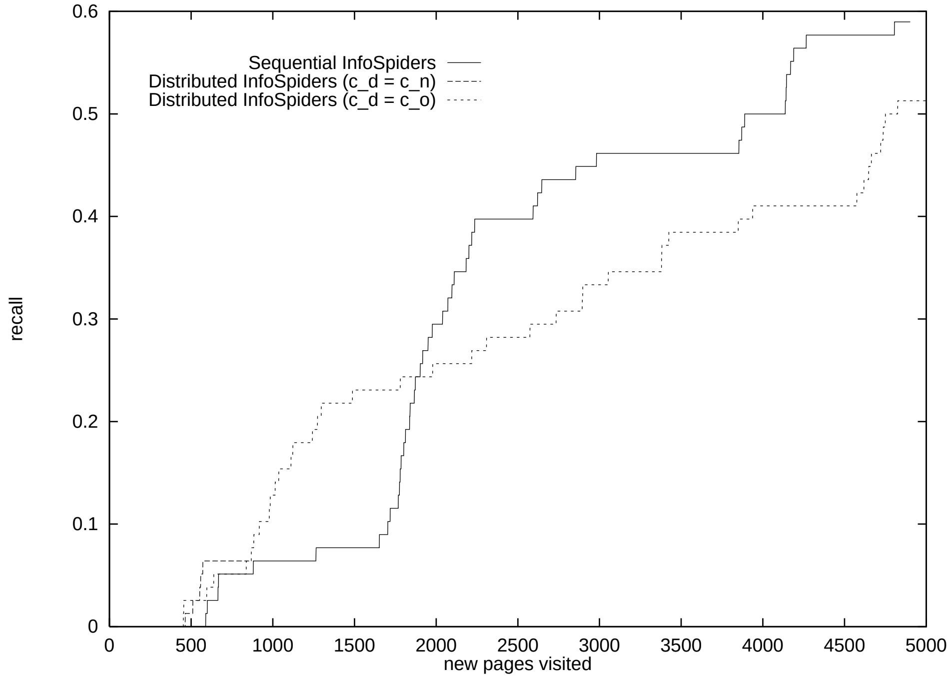


Figure IV.12: Recall in three runs of sequential and distributed InfoSpiders with different cost parameters.

carrying capacity and corresponds with the earliest extinction. The sequential case is intermediate because the two costs are charged depending on whether or not a page is already in the cache. The distributed case with smallest cost $c_d = c_o$ has largest carrying capacity and no extinction occurs up to time T_{max}. Assuming agents can save state on the servers where the pages reside, so that documents can yield energy only the first time they are visited (the environmental marking operated by the cache in our simulations), the population would eventually run out of resources and go extinct in this case as well.

Figure IV.12 shows recall versus new pages visited in the same three runs. Recall here is the fraction of relevant pages visited up to a point, and can be used as a measure of performance to be maximized. As the plot illustrates, population size is not an indicator of performance. Before extinction in the distributed case with $c_d = c_n$, recall does not seem to significantly differ from the other two cases. And the distributed case with $c_d = c_o$, in spite of its surviving population, has

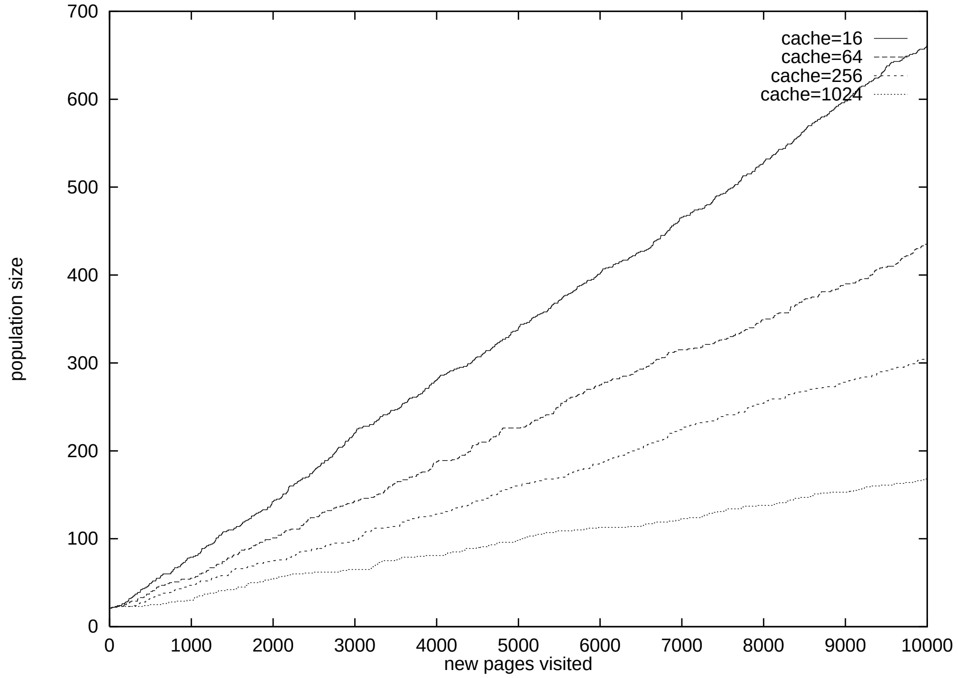


Figure IV.13: Population dynamics in four runs of sequential InfoSpiders with different cache sizes.

achieved lower recall than the sequential case. Note, however, that since recall is normalized by the total number of new pages visited, this measure of performance is intrinsically sequential — it does not capture the fact that distributed InfoSpiders can execute in parallel.

To evaluate the effect of cache size on population dynamics and on performance, we compare four runs of sequential InfoSpiders with finite cache size, rather than effectively infinite cache (as was the case when $|C|$ was set to T_{max}). We let $|C|$ equal 16, 64, 256, and 1024 in the four runs, respectively. The ratio of the cost parameters is still $c_o/c_n = 0.1$, but the value of c_n is set to 0.01 for this experiment. Figure IV.13 plots the population dynamics in the four cases. The population size grows at higher rates for smaller cache sizes, and viceversa. This is not surprising, because a smaller cache means that pages are replaced — and therefore resources are effectively replenished — more frequently.

As Figure IV.14 shows, the different population growths spurred by the

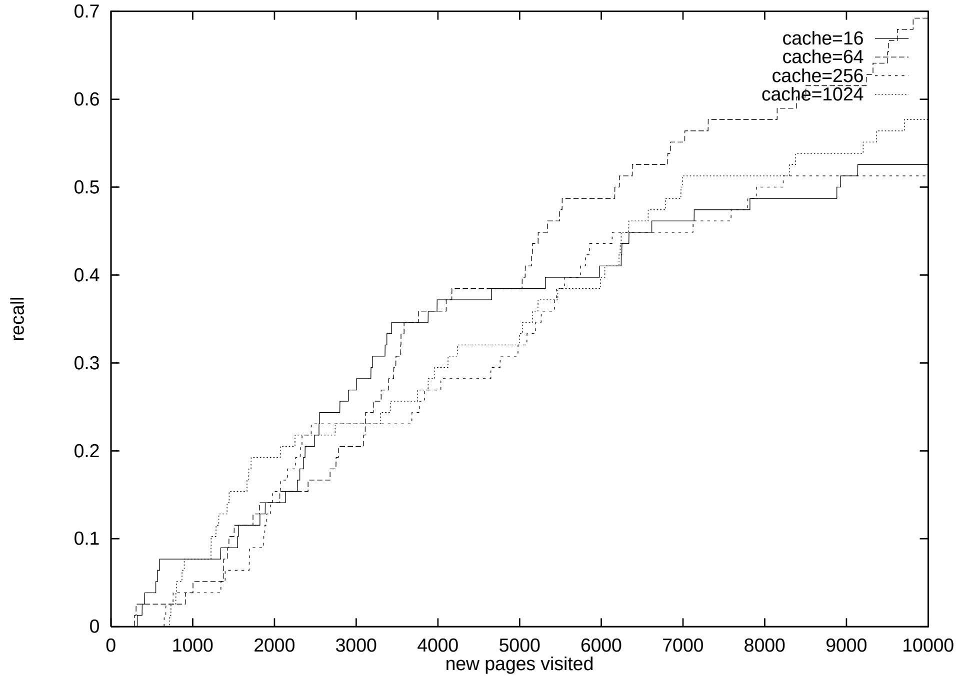


Figure IV.14: Recall performance in four runs of sequential InfoSpiders with different cache sizes.

various cache sizes do not necessarily imply any significant difference in performance. A mere change in growth rate, or in population size, does not alter the search bias of the population. The situation is equivalent to changing the reproduction threshold θ, or the cost c_n, by some scaling factor. So long as time is measured sequentially, as for the recall statistics, relevant pages are discovered at the same rate.

In order for performance to be affected by population dynamics, the search bias of the population must be altered so that certain directions become preferred over others. This can be achieved for example by relevance feedback. Figure IV.15 plots population size in two runs of distributed InfoSpiders with $c_d = 0.002$ and infinite cache size ($|C| = T_{max}$). In one run, no relevance feedback is provided; in the other, positive relevance assessments are generated automatically, for visited pages in the relevant set, every 100 newly visited pages. The energy provided by relevance feedback decays in the latter run with a geometric factor $\gamma = 0.5$.

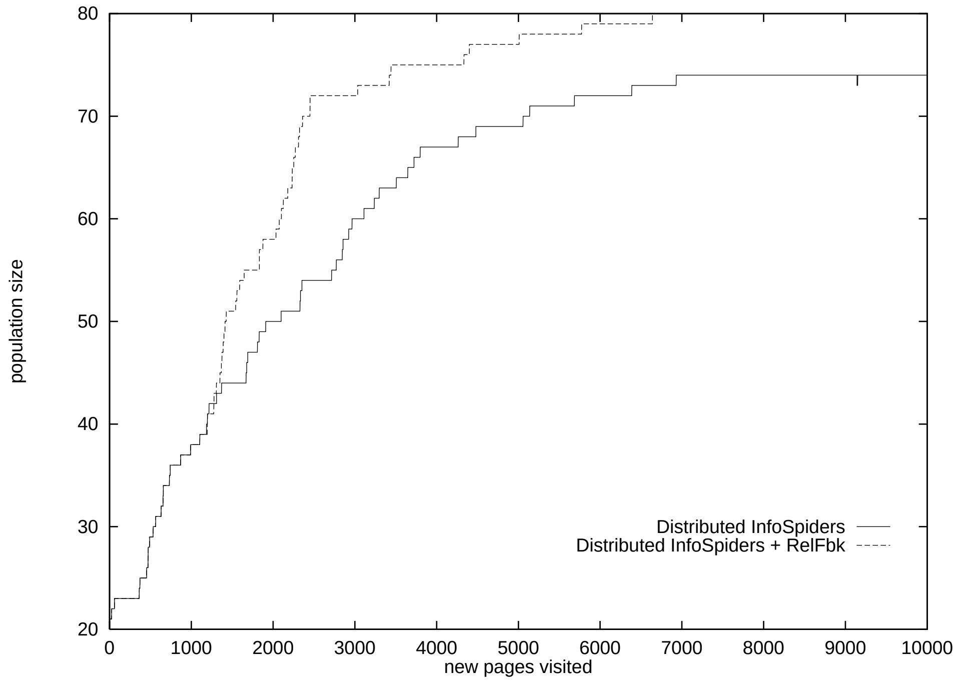


Figure IV.15: Population dynamics in two runs of distributed InfoSpiders with and without relevance feedback, respectively.

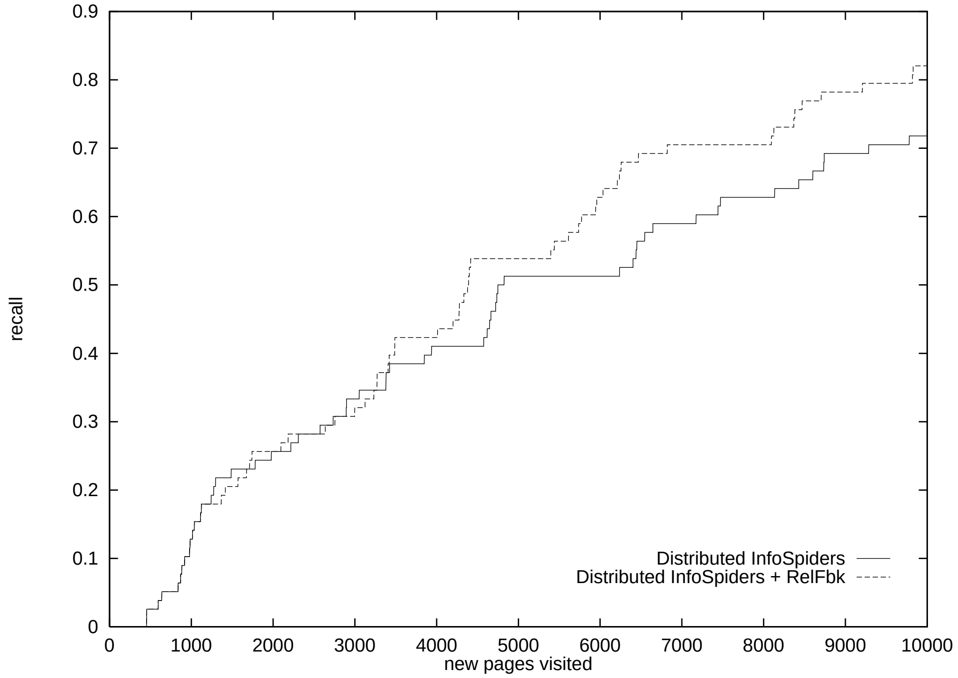


Figure IV.16: Recall in two runs of distributed InfoSpiders with and without relevance feedback, respectively.

k	I_k
`ORGAN`	1.0
`PUBLIC`	1.0
`PRIVAT`	1.0
`SERVIC`	1.0
`SOCIAL`	1.0
`HUMAN`	1.0
`ACTIV`	1.0
`ALLEVI`	1.0

Table IV.4: Default initial word feedback list corresponding to a query.

Figure IV.16 shows that in this case there is not a mere scaling of population growth rate, but a change in performance driven by the bias imposed by relevance feedback. The user is effectively changing the energy landscape and pushing the population in a preferred direction — one in which agents happen to find more relevant pages.

IV.E.2 Micro analysis

To illustrate how some of the goals that we set out for adaptive information agents are achieved by InfoSpiders, let us now look more closely at a few typical agents adapting within a single search. The query is the same one used to analyze population dynamics in the previous section; after removal of stop words and stemming, it results in the keyword vector shown in Table IV.4.

To simplify the analysis during this example run we use simple perceptrons to represent agent behaviors, so that an agent is completely described by a vector of 8 keywords and a vector of 9 weights (one per keyword plus a bias term), plus the β parameter. We set $\zeta = 1$, use panmictic crossover, provide the population with relevance feedback every 50 new pages, and run the search until the population had visited $T_{max} = 1000$ new pages.

Spatial context

How do InfoSpiders internalize environmental word features that are spatially local (in the sense of linkage topology)? Can they adapt to the spatial context in which they evolve? To answer these questions consider two agents, A and B, born at the same time but in different places. More precisely, A was born at time 554 in the sequential execution of the algorithm, i.e., after the population had collectively visited 554 new pages. B was born at time 580; A and B were effectively contemporaries because they had the same temporal context — the global information resulting from the relevance assessments of time 550, partially shown in Table IV.5.

As Table IV.5 shows, the original query words were displaced from their top positions and replaced by new terms. For example, `PRIVAT` and `ALLEVI` had relatively low weights, while `FOUNDAT` and `RED` appeared to have the highest correlation with relevance feedback at this time.

A's and B's keyword vectors are shown in Table IV.6. In the course of the evolution leading to A and B through their ancestors, some query terms were lost from both genotypes. A was a third generation agent; its parent lost `ALLEVI` through a mutation in favor of `HULL`. At A's birth, `PRIVAT` was mutated into `TH`. B was a second generation agent; at its birth, both `ALLEVI` and `PRIVAT` were replaced by `HULL` and `ADDAM`, respectively, via mutation and crossover. These keyword vectors demonstrate how environmental features correlated with relevance were internalized into the agents' behaviors.

The difference between A and B can be attributed to their evolutionary adaptation to spatially local context. A and B were born at documents D_A and D_B, respectively, whose word frequency distributions are partly shown in Table IV.7. `TH` represented well the place where A was born, being the second most frequent term there; and `ADDAM` represented well the place where B was born, being the third most frequent term there. By internalizing these words, the two situated agents are better suited to their respective spatial contexts.

rank	new	k	I_k
1	⋆	`FOUNDAT`	0.335
2	⋆	`RED`	0.310
3	⋆	`MISSION`	0.249
4		`SOCIAL`	0.223
5	⋆	`CROSS`	0.197
6	⋆	`HULL`	0.184
7	⋆	`HOUS`	0.183
8		`ORGAN`	0.161
		...	
15		`SERVIC`	0.114
16		`ACTIV`	0.112
		...	
23	⋆	`TH`	0.094
		...	
30		`PUBLIC`	0.087
		...	
32	⋆	`ADDAM`	0.079
		...	
37		`HUMAN`	0.075
		...	
41		`PRIVAT`	0.067
		...	
44		`ALLEVI`	0.065
		...	

Table IV.5: Part of the word feedback list and weights at time 550. Here and in Table IV.8, stars mark new terms not present in the original query. Note that `TH` does not correspond to the article “the,” which is a noise word and thus removed from all documents; rather, it corresponds to the “th” used for ordinal numbers and often associated with centuries.

A	B
`ORGAN`	`ORGAN`
`PUBLIC`	`PUBLIC`
`TH`	`ADDAM`
`SERVIC`	`SERVIC`
`SOCIAL`	`SOCIAL`
`HUMAN`	`HUMAN`
`ACTIV`	`ACTIV`
`HULL`	`HULL`

Table IV.6: Keyword vectors for agents A and B.

$rank_{D_A}$	k	$freq(k, D_A)$	$rank_{D_B}$	k	$freq(k, D_B)$
1	`WORKHOUS`	0.076	1	`HOUS`	0.043
2	`TH`	0.038	1	`HULL`	0.043
2	`POOR`	0.038	3	`ADDAM`	0.025
4	`SOCIAL`	0.030		...	
4	`CENTURI`	0.030	38	`AMERICAN`	0.004
	...			...	

Table IV.7: Most frequent terms in the documents where agents A and B were born. Word frequencies are normalized by the total number of words in each document.

rank	new	k	I_k
1		`SERVIC`	0.273
2		`SOCIAL`	0.268
3		`ORGAN`	0.238
4	⋆	`FOUNDAT`	0.152
5	⋆	`NATION`	0.148
6		`PUBLIC`	0.138
		...	
12		`ACTIV`	0.118
		...	
15	⋆	`HULL`	0.110
		...	
17		`PRIVAT`	0.098
		...	
25		`HUMAN`	0.087
		...	
31		`ALLEVI`	0.080
		...	
35	⋆	`AMERICAN`	0.077
		...	
56	⋆	`ADDAM`	0.053
		...	

Table IV.8: Part of the word feedback list and weights at time 950.

Temporal context

Let us now consider adaptation along the temporal dimension. How do InfoSpiders internalize features appropriate for their time? Can they capture the temporal context in which they evolve? To answer these questions consider again two agents: our acquaintance, B and a third agent, C. They were born in the same place (D_B; see Table IV.7) but at a different times; more precisely, C was born at time 965, and therefore its temporal context was the global information resulting from the relevance assessments of time 950, partially shown in Table IV.8. B's temporal context was given in Table IV.5.

Let us make a few observations about Table IV.8. After more relevant documents were discovered, the original query terms gained weight, with four of

B	C
`ORGAN`	`ORGAN`
`PUBLIC`	`PUBLIC`
`ADDAM`	`PRIVAT`
`SERVIC`	`SERVIC`
`SOCIAL`	`SOCIAL`
`HUMAN`	`AMERICAN`
`ACTIV`	`ACTIV`
`HULL`	`HULL`

Table IV.9: Keyword vectors for agents B (from Table IV.6) and C.

them in the top six positions. Their relative positions also changed; for example, `PRIVAT` surpassed `HUMAN`. Other words lost importance; among those with a presence in B's and C's birth page D_B, both `HULL` and `ADDAM` decreased their weights, but while the former maintained a relatively strong position, the latter did not. Finally, some new terms made their first appearance in the list, such as `AMERICAN` that was also represented in D_B but was not present in the word feedback list at time 550.

Table IV.9 shows the differences between the representations of agents B and C. Such differences reflect the times in which these agents were born. When B was born, `ADDAM` appeared better correlated with relevance then `PRIVAT`, while the converse was true when C was born. The internalization of the two terms by B and C, respectively, is consistent with this change in temporal context. Furthermore, at the time of C's birth `AMERICAN` had a small but positive global weight, so that the presence of this term in D_B could be picked up by C — something impossible for B due to the term's absence in the earlier word feedback list. By evolving to internalize these words, the two agents adapted to their respective temporal contexts; since the change over time was caused by the user's relevance assessments, these agents effectively achieved personalization.

k	w_k^D	w_k^E
`ORGAN`	0.22	-0.14
`PUBLIC`	0.75	-0.42
`PRIVAT`	0.20	0.81
`SERVIC`	0.22	-0.05
`SOCIAL`	-0.01	-0.21
`HUMAN`	0.07	-0.03
`ACTIV`	0.12	0.27
`ALLEVI`	0.29	-0.08
Bias	0.02	0.41

Table IV.10: Learned perceptron weights for agents D and E.

Experience

Another question is: Can InfoSpiders internalize their local context over smaller spatial scales and shorter time scales, during their lives? To answer, consider two agents, D and E, in the initial population. Both lived until the end of the run and were successful (with three and nine offspring, respectively). Although D and E were born at the same time and in the same place (the root of the "Human Society" tree), they searched through different paths and therefore had different life experiences.

Table IV.10 shows the weight vectors of D's and E's neural nets at the end of the run. The weights were adapted via Q-learning so that each agent would be able to estimate document relevance across links. For example, it is clear that the strategy learned by D paid special attention to `PUBLIC`. On the contrary, E's predictions were anticorrelated with the presence of `PUBLIC`. This demonstrates that the local contexts experienced by D and E during their lives were quite divergent with respect to this word. Through reinforcement learning, the two agents were able to internalize into their neural nets the perceived correlations between environmental features and relevance.

Unsupervised query expansion

In all experiments described up to this point, the keywords that agents are allowed to internalize into their representation are limited by the word feedback list. This is due to the fact that $\chi = 0$ in Equation IV.5. The result is that the query expansion process is supervised; a new keyword cannot be internalized unless it is contained in some document assessed (positively or negatively) by the user. All other terms have $I_k = 0$ and thus their probability of being selected during mutation is also zero.

For the last InfoSpiders demonstration we set $\chi = 1$ to explore the opposite extreme, i.e., completely unsupervised query expansion. The distribution of Equation IV.5 reduces in this case to

$$\Pr[k] \propto freq(k, D).$$

This way, word cues can be internalized from environmental interaction alone, even in the total absence of relevance feedback. At reproduction, a term in the offspring genotype can be mutated into a term that is frequent in the document where the offspring is born. Since reproduction occurs after a positive energy intake, this query expansion mechanism is driven by the detection of correlation between environmental signals (words) and fitness (estimated or assessed relevance).

As an illustration, Figures IV.17 and IV.18 list the results of the keyword mutation and crossover operators in the course of a single InfoSpiders run. These are some of the reproduction events occurred until $T_{max} = 5000$ pages are visited. All other parameters are as in the run of the previous subsections, with the exceptions that the cache is limited to $|C| = 1000$, *no* relevance feedback is provided, and of course $\chi = 1$. Thus this search is completely unsupervised, and energy comes only from relevance estimates based on the query words (Table IV.4). The mutation and crossover operations shown are taken from the InfoSpiders log file. Figures IV.17 and IV.18 give a sense of the diversity that quickly propagates through the population based on the different pages visited by each agent.

```
/ref/ <- /activ/
/sponsor/ <- /social/
/legion/ <- /allevi/
/role/ <- /human/
/und/ <- /activ/
/associ/ <- /social/
/agre/ <- /ref/
/embol/ <- /role/
/religi/ <- /social/
/death/ <- /allevi/
/govern/ <- /allevi/
/chang/ <- /human/
/human/ <- /allevi/
/yaqui/ <- /chang/
/urban/ <- /allevi/
/variou/ <- /associ/
/social/ <- /yaqui/
/law/ <- /ref/
/class/ <- /law/
/subsidi/ <- /chang/
/characterist/ <- /public/
/human/ <- /social/
/chang/ <- /chang/
/societi/ <- /allevi/
/ideologi/ <- /chang/
/psychologi/ <- /activ/
/type/ <- /chang/
/concept/ <- /allevi/
/sanction/ <- /govern/
/human/ <- /variou/
/televis/ <- /activ/
/public/ <- /human/
/organ/ <- /activ/
/relat/ <- /und/
/centuri/ <- /chang/
/spanish/ <- /educ/
/schwinghosen/ <- /chang/
/chang/ <- /social/
/ag/ <- /class/
/languag/ <- /ref/
/polit/ <- /activ/
/relat/ <- /allevi/
/path/ <- /role/
/consider/ <- /death/
/elit/ <- /psychologi/
```

```
/quaker/ <- /human/
/human/ <- /allevi/
/subject/ <- /servic/
/centuri/ <- /allevi/
/design/ <- /activ/
/concept/ <- /human/
/april/ <- /allevi/
/societi/ <- /allevi/
/statu/ <- /human/
/futur/ <- /distribut/
/minor/ <- /subject/
/trade/ <- /allevi/
/human/ <- /societi/
/organ/ <- /legion/
/activ/ <- /allevi/
/rise/ <- /type/
/structur/ <- /centuri/
/polygami/ <- /intern/
/cereal/ <- /concept/
/educ/ <- /law/
/polish/ <- /activ/
/lee/ <- /chang/
/devolut/ <- /class/
/steril/ <- /govern/
/societi/ <- /minor/
/cold/ <- /social/
/urban/ <- /ag/
/institut/ <- /allevi/
/price/ <- /cold/
/human/ <- /role/
/languag/ <- /allevi/
/polyolefin/ <- /agre/
/structur/ <- /statu/
/daili/ <- /elit/
/parkinson/ <- /human/
/relationship/ <- /futur/
/rate/ <- /lee/
/outer/ <- /social/
/trelawni/ <- /societi/
/ku/ <- /chang/
/bank/ <- /privat/
/master/ <- /activ/
/rural/ <- /ref/
/abip/ <- /und/
/law/ <- /religi/
```

Figure IV.17: Mutations during a 5000-page run with unsupervised query expansion

```
/public privat servic associ human/ <- /public privat servic social human/
/servic social yaqui/ <- /servic social human/
/social human ref/ <- /social human activ/
/privat servic social chang/ <- /privat servic social human/
/public privat servic variou/ <- /public privat servic social/
/servic social human activ/ <- /servic social human law/
/activ/ <- /type/
/organ public privat servic social/ <- /organ public privat servic sponsor/
/subject human human/ <- /servic quaker quaker/
/human human/ <- /social human/
/servic social human type/ <- /servic social human organ/
/chang/ <- /activ/
/minor human human/ <- /servic human human/
/organ public privat servic cold/ <- /organ public privat servic social/
/public privat servic social statu/ <- /public privat servic social concept/
/social human/ <- /social subsidi/
```

Figure IV.18: Recombinations during a 5000-page run with unsupervised query expansion

Chapter V

Conclusions

In the previous chapters of this thesis we discussed research at the boundary between artificial life, adaptive computation, biology, and information systems. Chapter II focused on the common algorithmic themes that make up the framework for Chapters III and IV. The theoretical and experimental aspects of this framework have first been introduced elsewhere [118, 123].

Chapter III was concerned with the direction of the artificial life bridge going from computational methodologies to exploring questions in biological, evolutionary, and ecological theory. The LEE model was originally presented in a series of technical reports [115, 116, 112]. The connection between the observed population dynamics and the carrying capacity of Latent Energy Environments was discussed in a chapter of the book edited by Belew and Mitchell [119]. The results outlined in Chapter III are selected from experiments reported in a number of papers [117, 26, 118].

Chapter IV dealt with the direction of the artificial life bridge going from local selection algorithms, inspired by ecological systems, to applications in adaptive and autonomous information retrieval, especially in regard to distributed search over networked information environments. The idea of applying artificial life algorithms to adaptive information agents and the World Wide Web was first introduced when the Web was still in its infancy [128, 124, 113]. The first prototype of the InfoSpiders system was presented in 1997 [120]. The machine learning aspects of the algorithm, its later implementations, the distributed aspects of the approach, and many results outlined in Chapter IV were discussed in the most recent papers [114, 121, 122].

V.A Discussion of contributions

The first contribution of this thesis is an exploration of issues concerning a tight integration between adaptation at the population level (evolution) and at the individual level (learning). We have proposed and analyzed the local selection algorithm as a way to build an evolutionary loop around the same environmental signal(s) that drive reinforcement learning. This allows one to focus on the roles of reinforcement learning and evolution as two similar processes, acting at different temporal and spatial scales by sampling environmental cues over different intervals — a life cycle or a lifetime.

At the individual level, reinforcement learning biases an agent's behavior toward actions leading to a better knowledge of the environment and to increased payoff, or longer survival. At the collective level, reinforcement evolution biases the population toward sampling areas of the environment that are close to known resources, by increasing the density of agents near those sources.

The integration of local selection with reinforcement learning is quite different from hybrid models in which evolution and learning are used together [14, 16, 63, 87, 147, 86]. Typically, genetic algorithms are used in conjunction with local search as a way to improve on the evaluation of an individual's fitness. Such models, albeit successful in several domains, beg the question of how to allocate computational resources between evolving the population and searching locally around individual genotypes; the answers remain mostly empirical. Another major difference between the two approaches is that local selection only requires competition among agents sharing environmental resources, and thus is an implicitly distributed algorithm.

As was shown in Chapter II, evolutionary algorithms based on local selection are not necessarily feasible or appropriate in any problem domain. However, they are a useful paradigm for distributed agents situated in some (physical or software) environment.

As a model of ecological adaptation, Latent Energy Environments are little more than the local selection algorithm in conjunction with a specification for agents capable of learning during their lifetime (neural nets), and a specification for a class of physics-grounded environments. The result is the first simulation-based model of ecological processes allowing for the simultaneous study of evolution and individual plasticity. An immediate consequence of the local selection algorithm is the emergence of density dependence. We have shown in Chapter III that by specifying the complexity of an environment we can accurately predict the carrying capacities corresponding to the range of evolvable behaviors. The individual-based LEE model is an excellent fit to the analytical model of logistic population growth used in ecology for similarly density-dependent conditions.

We have reported in Chapter III on the use of LEE to investigate the coevolution of behaviors and morphology (sensors). This has turned out to be an unexpectedly difficult problem due to the interdependence between the neural net implementing a behavior and the sensory interface producing the information upon which the behavior must be based. We have shown that, if the cognitive modifications provided by learning are well correlated with those required for survival in a particular environment, then phenotypic learning can assist adaptation in that environment by facilitating the evolution of appropriate genotypes.

The integration between adaptation by reinforcement evolution and learning was also explored in Chapter IV, in the context of information agents applications. We have shown how to construct agents who can evolve to exploit global and local information about the environment in which they are situated, and learn during their lifetime to adjust their behaviors to best match their local environmental context. Our evaluation of the InfoSpiders collective performance provides us with encouraging support for the approach: the population can locate relevant documents in large distributed corpora faster than exhaustive search. Furthermore, thanks to the local nature of each agent's interactions with the information environment, distributed InfoSpiders can also beat globally optimum strategies.

The InfoSpiders system also gave us a chance to explore the question of internalization: How to select among the many environmental cues the ones that best correlate with fitness or survival? As the LEE experiments on the evolution of sensors illustrated, the appropriateness of reinforcement signals is crucial in determining the success of reinforcement learning and evolution.

In the InfoSpiders context, evolution selects words to be internalized into an agent's representation. These signals then drive the agent's link-following decisions. Finally the agent learns through its lifetime experience to improve its endogenous estimation function (neural net) to match the evaluation of visited nodes. The evaluation function depends on the same word cues that can be internalized by evolution. This closes the loop, and allows the two adaptive processes to support each other. We have shown in Chapter IV how local selection and internalization can lead agents to adapt to their local environmental context, over different spatial and temporal scales.

Another contribution of this thesis was to explore the interactions and integration of supervised and unsupervised adaptation. Evolution by local selection and individual learning by sensory prediction, associative reward-penalty, or Q-learning, are all examples of unsupervised adaptation. Like reinforcement learning, there may be a weak supervision in the sense that the environment provides adaptive agents with reinforcement or feedback cues. However, if the agents have to internalize the appropriate signals, we can consider these mechanisms unsupervised.

In the LEE context, we have considered parental imitation as a form of supervised learning, although the supervision occurs by evolved individuals rather than by the external experimenter. Nevertheless the interaction allowed for an interesting cost-benefit analysis of an evolving life-history trait, showing that behavioral improvements of the immature phenotype may be among the factors determining the trade-off between costs and benefits of delayed maturation. The simulations also demonstrated that the Baldwin effect can use this extended window of juvenile plasticity to good evolutionary advantage.

In the InfoSpiders context, we had good reason to exploit a more direct form of supervised adaptation, or learning from examples. While we don't want agents who *need* user feedback in order to function, we certainly want to make use of the user's relevance assessments if such precious resources are made available. Relevance feedback was used not only to bias the population toward promising areas, but also to improve on the relevance estimation function and to enable the internalization of local features into agents' adaptive behaviors. As a consequence, the performance of InfoSpiders received a boost from the synergy between unsupervised learning and relevance feedback.

The interplay between local selection, internalization, Q-learning, and relevance feedback provided us with a way to address another problem in this domain: Making the learning problem manageable in the face of the huge feature space dimensionality (easily as large as 10^5 words). This goal was achieved by *selective query expansion* — another way to describe internalization. The micro analysis in Chapter IV has enabled us to determine that single agents can in fact select the important local features of their environment, while the collective ecology captures a more heterogeneous snapshot of what features best correlate with user relevance. Agent representations and strategies evolve with time and change over an agent's lifetime; they are different from agent to agent depending on the temporal and spatial contexts in which they were born, and on what parts of the environment each has experienced. Therefore we expect this approach to be robust in the presence of time-varying documents and user profiles.

Finally, we have discussed the importance of an accurate account of the environment, its complexity, and its topology, in determining the adaptive behaviors that may emerge. The simple demonstrations of this principle outlined in this thesis could be seen as quantitative explorations of the thesis put forward by Godfrey-Smith [52]: "The function of cognition is to enable the agent to deal with environmental complexity."

More specifically, in our LEE experiments (Chapter III) we have characterized three forms of environmental complexity and shown how they create selective pressures for populations evolving different varieties of structure in their behaviors, matching those those of their environments. Gradual patchiness of resources causes efficient foraging behaviors to be evolved. Metabolisms supporting complementary food chains results in the preservation of biodiversity, but only in the presence of a sufficient carrying capacity differential in favor of mutual behaviors. Finally, dynamic environments with seasonal fluctuations divide the individuals into subpopulations with different behaviors exploiting seasonal niches. These collective behaviors are not necessarily optimal, but they are adaptive in that they guarantee robust survival of the population in the face of increasing environmental harshness.

In characterizing a "real" (although neither physical nor natural) environment such as the World Wide Web (Chapter IV), we have focused on the role of the topology induced by the hyperlinks manually created by information providers in their documents. When we started this endeavor, we had difficulty arguing in defense of our conjecture that such linkage topology could be detected against the noise background and constitute a precious resource in guiding autonomous software agents. Today, much corroborating evidence [88, 28, 27, 72, 100] makes our conjecture widely accepted and points to linkage topology as a resource as useful as the statistical topology induced by word usage and exploited by search engines.

Our results suggest that distributed, adaptive, on-line information browsing agents could complement current indexing technology by starting up where search engines stop. Engines provide global starting points, based on statistical features of the search space; agents can use topological features (links) to guide their subsequent search on-line. We have shown how this approach can extend the current state of the art by dealing with the problems of scalability, personalization, and localization.

V.B Directions for future work

V.B.1 From technology to nature

The LEE model has many limitations and possible extensions. To mention just a couple, recombination and development are missing in the current model. Asexual reproduction by cloning has several limits. Recombination may facilitate the adaptive process for many rich environments in which the number of possible behaviors is very large [127, 125]. Sexual reproduction would also enable us to use LEE for studying the evolution of speciation; sexual selection models have been proposed in evolutionary biology for this important yet poorly understood problem [85, 174]. In this thesis we have touched on some themes — biodiversity, niche selection — that are closely related to speciation.

Another desirable consequence of modeling behaviors is that it demands a more elaborate characterization of the genotype/phenotype distinction [15, 64]. Such a developmental process would strengthen the role of environment, from selective pressure on phenotypes to direct interactions with ontogeny.

Minimal models of behaviors dealing with space, such as foraging, habitat selection, signalling, etc., are buildable with LEE without resorting to many ad-hoc assumptions restricting the range of observables. It is also easy to distinguish between, and integrate across, the selective pressures determining within- and across-niche adaptation. The environments and behaviors studied in this thesis have given limited examples of this approach, that could be extended to include models of selection deriving from more direct interactions among individuals and populations, such as predation, mate choice, communication, and other social behaviors.

In keeping with the original goal of building tools to help biologists answer open questions about natural systems, the future emphasis should be on models that can be predictive of real biological data. A possible first step in this direction might be to study the different species of tropical oceanic fishes (tunas, billfishes, sharks) who share and co-exploit the same environment — the pelagic open ocean. LEE has been used to build a model in which the behavior of predator oceanic fishes is driven by the biotic environment [32]. The model is studied with simulations in which artificial fishes in an evolving population adapt their behaviors to a three-dimensional environment. The spatial distribution and temporal dynamics of prey in the simulated world are inspired by acoustic observations about horizontal and vertical movements of individuals of different species in French Polynesia. The movement patterns of many evolved behaviors have been analyzed to compare artificial individuals to real fishes (three species of tunas, three species of billfish and one species of shark) observed by acoustic telemetry, and to examine how the artificial fishes exploit their environment. Most of the artificial individuals have vertical and horizontal patterns similar to those exhibited by fishes in the wild. This approach exemplifies how models such as LEE can be effective tools for studying multi-behavior ecologies.

Sociality is the last dimension of environmental complexity. Direct interactions among organisms go beyond the characterization of environments grounded on physical space. Examples of social phenomena that could be addressed within the LEE framework include interactive signalling, cooperative equilibria, and optimal size of social colonies or other complex societies [151].

An important feature of local selection is the *robustness* it induces in the evolved populations and behaviors. For this reason, local selection algorithms could be used for modeling populations of parasitic individuals, such as cancer cells or AIDS viruses [137]. The resistance of these populations is believed to lie in their capability to quickly evolve adaptive mutants. The goal of such a model, from a therapeutic viewpoint as opposed to the experiments of this thesis and simulations of adaptive behavior in general, is to find ways to most efficiently kill off the population of adapting parasites. In this case it is crucial to characterize the factors that make the environment — the host — harsh enough to reduce the diversity of the population and thus weaken its ability to withstand extinction.

V.B.2 From nature to technology

There are many experiments, questions, and ideas around the InfoSpiders project that could not be explored in this thesis and that remain interesting open directions for the future. For example, our weighting scheme may be improved; it has been suggested that the use of IDF in the local relevance estimation mechanism (Equation IV.3) may be inappropriately biased toward global features [167].

Many parameters of the InfoSpiders algorithm (Table IV.2) have not received enough attention. For example, what are appropriate α values to weigh past versus new relevance assessments? Should this be a fixed parameter, a user-defined preference, or an evolved trait? And how quickly should the energy resources associated with relevance feedback decay (γ) in order to optimize performance? How should agents discount future payoffs in their Q-value function approximation? Should μ be a fixed or evolved parameter? How far should should an agent go to estimate a link? We have not studied the effect of varying the window size ρ. Further, how should we determine the trade-off between supervised and unsupervised query expansion? We have illustrated the behavior of the two extremes possibilities for χ, but intermediate values remain to be considered. And finally, what are appropriate neural net architectures for the agents? Are the linear functions learnable by perceptrons sufficient to capture the user needs, or do we need the sophistication of multi-layer networks?

Lukose and Huberman [100] have suggested that the InfoSpiders death mechanism could be modified according to their optimal stopping criterion inspired by Equation IV.2 and the way in which real options are evaluated in financial markets. This could conceivably be accomplished by changing the death threshold ω to take into account the Q-value used for reinforcement learning (cf. Equation IV.4). However, it would have to be determined empirically whether this approach would improve performance or hinder the ecological balance determined by energy conservation and carrying capacity.

Many models of interaction among agents are worth exploration in this domain. Agents learning from other agents, agent collaboration, and agent communication languages are all examples of very active research areas. The only form of *direct* agent interaction that we have touched on is crossover. An agent at reproduction can recombine its internal representation with that of a “nearby” agent. The two can internalize experiences that are now relevant to each other because of their proximity. But the experiments in this thesis have really not explored the impact of crossover on performance, nor the effects of different types of recombination.

The notion of *proximity* mentioned above must be regarded in the face of the communication overhead imposed by recombination. We have argued that distributed agents are best suited to perform tasks in networked environments. From this point of view agent interactions must be kept at a minimum. The two examples of recombination used by InfoSpiders are extreme cases in which interactions occur either at a completely global scale (panmictic crossover) or within a single document file (local crossover). Intermediate implementations should be considered to take advantage of recombination’s properties while maintaining the potential speed-up of distributed execution. One obvious possibility would be to allow interactions among agents situated on the same server, who could share information locally without network communication overhead.

These considerations call for further study of both *explicit* sources of communication, such as recombination, and interference mechanisms that are *implicit* in the current model. Such interactions are mainly caused by the use of centralized repositories for caching, resource sharing, and relevance feedback. The experiments on cache size in Chapter IV have only scratched the surface of these issues. The inclusion of further local state (memory) into an agent representation would not only allow for localized cache and relevance feedback images, but also reduce the reactive limitations of the current repertoire of agent behaviors.

While we have just begun to explore the issue of distributed implementations, mainly by simulation, the agents that we envision are mobile agents that execute on remote servers. It is not just information that must travel, as in current client-server protocols, but agent code as well. This way agents can use their intelligence to select the information to be sent back to the user's machine; information providers will in our opinion be interested in trading off CPU cycles for improved network bandwidth. Secure languages and protocols are surely needed before trusted autonomous agents will become a welcome reality; agent research is providing systems technology with the thrust that may very soon make such mobile agents possible [155].

In the experiments outlined in this thesis we have been most interested in the behavior of agents on a carefully controlled and structured corpus (EB). Therefore the full diversity we can reasonably expect from our agents as they interact with the real Web remains to be demonstrated. We have shown at least some divergence in the features that allow one agent to be successful within one topical area of the Encyclopedia and another, but the real purpose of open-ended evolutionary methods like those we propose is to adapt to the much wider variation found in Web media. We expect there to be roles for many different types of agents, sensitive to widely varying user demands, and effective at searching disparate corpora. Extensive evaluation of our approach on the actual Web is necessary to verify whether these goals can be met, making the problem of evaluation in open environments one of high priority.

Finally, the feasibility of integrating agent-based on-line search with index-based search engines must be put to the test. Hybrid systems can be constructed in which search engines provide agents with good staring points, based on the statistical (word-based) topology of the search space. This is crucial because, as we have shown in Chapter IV, the performance of on-line distributed search degrades with the distance between starting points and relevant clusters. Personal agents can then continue the search on-line, adapting to both user and current environmental context. Using a population of autonomous browsing agents as a front-end to a search engine can help us better understand the mutual benefits of the two approaches and the potential synergies that may ensue.

V.C Interdisciplinary issues

One of the goals of the "artificial life bridge" proposed by the research in this thesis is to strengthen the connections between machine learning, evolutionary algorithms, autonomous agents, distributed information retrieval, and ecology. The cross-fertilization of these areas shows great promise for furthering our understanding of complex natural systems as well as applying nature's solutions to the development of useful technology. Our artificial information environments already exhibit many of the real complexities of the natural world, creating challenges that warrant new, creative approaches. Living systems will in our opinion continue to inspire techniques allowing one to meet such growing challenges.

V.C.1 Relevance to natural systems

In one direction we have suggested ways in which computational science can provide tools to help biologists in their mission to study the natural world. We hope that LEE can provide for both a rich theoretical framework and a useful simulation tool not only for the artificial life community, but also for studying a broad range of issues in ecological theory as well as theoretical, behavioral, and evolutionary biology.

The methods that we have used to characterize different dimensions of complexity can be applied to understanding the dynamics of adaptive populations in other, perhaps more realistic, environments. We have seen that adequate ecological models dealing with the spatial structure of environments, as well as models to relate natural phenomena across scales, are missing [90]. Classic analytical tools, such as systems of differential equations, are limited in their capability to deal with complex environments. Minimal models such as LEE can contribute to moving in the right direction.

While biologists have had reason to resist to the use of genetic algorithms as biologically plausible models of adaptation in natural systems, because of their optimizing inclination, models based on local selection provide a framework within which ecological hypotheses can be tested through simulations before field experimentation.

A few observations should be made with respect to the experiments on the evolution of sensory systems and of age of maturity in Chapter III. In each of these it has traditionally been difficult to study the interactions between the evolution of morphological or life history traits on the one side, and of behavioral traits on the other. For example, prior quantitative approaches to the study of maturation age are often subject to the limitations of field experiments, and therefore these have mainly focused on morphological benefits of delayed maturation, such as body size, weight, etc. Animal psychology studies of cultural transmission, on the other hand, have difficulty in casting the phenotypic nature of learned behaviors into the more general evolutionary framework.

The simulation approach employed here provides a bridge between the two sides. This is important for biologists because even within the broad area of the natural sciences, there are disciplines where it is difficult to transfer data or models from one type of experiment to another, even if they may be relevant to each other. The models presented here attest to the potential of artificial life methodologies and interdisciplinary efforts.

V.C.2 Relevance to artificial systems

Artificial life is concerned with applying ideas from the natural biological world to engineering problems. In this vein, we have proposed local selection and internalization as general abstractions applicable to adaptation problems arising for agents situated in complex — large, noisy, dynamic, distributed, heterogeneous, inconsistent — environments. Such characteristics can be found in both the physical domains studied in robotics and the artificial, heterogeneous, distributed environments studied in information technology.

The local selection algorithm has been applied to another domain — image compression — in which statistical local features of the environment have to be captured by an adaptive algorithm. Cecconi *et al.* [25] have considered images as 2-D worlds in which a population of agents evolves by selecting areas rich with information that the user considers worthy of being kept. An extension of the JPEG compression algorithm uses the population to adapt compression factors to local environmental characteristics. Agents evolve collective behaviors that depend on the local texture of the image (e.g., follow edges); their spatial distribution ends up matching the features of the image. This approach saves up to 50% of storage robustly, i.e., preserving the most informative image areas. A very similar approach seems to show great promise in a wide range of image analysis applications [99].

When an embedded agent (a robot) has to learn appropriate actions by discovering important correlations between performance and environmental signals, the topology of the space biases input sequences and thus has tremendous influence on the time and space complexity of the learning algorithm [79]. The different complexity of "ideal" simulated worlds versus noisy, inconsistent, physical environments is of particular importance for evolutionary robotics [139, 73]. Local selection also enables the self-evaluation of evolving robot behaviors. Understanding the role of the environment in shaping the selective pressures giving rise to collective evolving behaviors is also relevant to the field of distributed robotics

[109].

Careful modeling of environments where multiple agents can interact by sharing finite resources, or even more directly by signalling or learning from each other, is important for the design of effective agents who must carry out autonomous tasks on behalf of the user [104, 105]. The world we have considered in Chapter IV, the Web, has many of the characteristics that make it interesting and challenging for autonomous software agents. Distributed information retrieval is only one class of applications where local selection and internalization can suggest interesting solutions. Any type of distributed search is arguably well suited for this methodology, whether the resource being sought by the population of agents is relevant information, a competitive price for some goods, a piece of software, an investment opportunity, or any other service.

Scalability will in our opinion become an even more central issue in the next generation of software agents. Assumptions such as completeness, availability, consistency, and recency — already false today — will become completely superseded. Agents will have to rely less on centralized repositories, develop more initiative, and become more independent. They will have to be able to continuously adapt to new conditions, create new models of their surroundings, and internalize new features. These goals appear more robustly attainable by large numbers of simple, dynamic, redundant agents such as InfoSpiders, than by large, complicated, static programs such as some of the systems currently classified as agents.

As agents will be endowed with additional capabilities, such as cash to engage in transactions, memory to keep local state, languages to communicate, and secure environments to execute remotely, users will trust them to perform more and more complex tasks on their behalf. Understanding how agents can locally interact with and adapt to their remote environments, while still satisfying their requirements at a collective level, will be key to the successful transition to the next, life-like stage of networked electronic environments.

Bibliography

[1] DH Ackley and ML Littman. Generalization and scaling in reinforcement learning. In DS Touretzky, editor, *Advances in Neural Information Processing Systes (NIPS2)*, pages 550–557. San Mateo, CA: Morgan Kaufmann, 1990.

[2] DH Ackley and ML Littman. Interactions between learning and evolution. In CG Langton, C Taylor, JD Farmer, and S Rasmussen, editors, *Artificial Life II*, pages 487–507, Reading, MA, 1992. Addison Wesley.

[3] R Armstrong, D Freitag, T Joachims, and T Mitchell. Webwatcher: A learning apprentice for the world wide web. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995.

[4] GO Arocena, AO Mendelzon, and GA Mihaila. Applications of a web query language. In *Proc. 6th International World Wide Web Conference*, 1997.

[5] RJ Bagley, JD Farmer, and W Fontana. Evolution of a metabolism. In C Langton, C Taylor, JD Farmer, and S Rasmussen, editors, *Artificial Life II*, pages 487–507, Reading, MA, 1992. Addison Wesley.

[6] M Balabanović. An adaptive web page recommendation service. In *Proc. 1st International Conference on Autonomous Agents*, 1997.

[7] JM Baldwin. A new factor in evolution. *American Naturalist*, 30:441–451, 1896.

[8] AG Barto and P Anandan. Pattern recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:360–375, 1985.

[9] AG Barto, RS Sutton, and CJCH Watkins. Learning and sequestial decision making. In M Gabriel and JW Moore, editors, *Learning and computational neuroscience.* MIT Press, 1990.

[10] J Batali. Small signalling systems can evolve in the absence of benefit to the information sender. In preparation, 1995.

[11] J Batali and WN Grundy. Modeling the evolution of motivation. *Evolutionary Computation*, 4(3):235–270, 1996.

[12] R Beckers, OE Holland, and JL Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In R Brooks and P Maes, editors, *Artificial Life IV*, Cambridge, MA, 1994. MIT Press.

[13] RD Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72:173–215, 1995.

[14] RK Belew. Evolution, learning, and culture: Computational metaphors for adaptive algorithms. *Complex Systems*, 4:11–49, 1990.

[15] RK Belew. Interposing an ontogenic model between genetic algorithms and neural networks. In J Cowan, editor, *Advances in Neural Information Processing Systems (NIPS5)*. San Mateo, CA: Morgan Kaufmann, 1993.

[16] RK Belew, J McInerney, and NN Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In CG Langton, C Taylor, JD Farmer, and S Rasmussen, editors, *Artificial Life II*, Reading, MA, 1992. Addison Wesley.

[17] RK Belew and M Mitchell, editors. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe Institute Studies in the Sciences of Complexity. Addison Wesley, Reading, MA, 1996.

[18] A Blum, L Hellerstein, and N Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50(1):32–40, 1995.

[19] KD Bollacker, S Lawrence, and CL Giles. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Proc. 2nd International Conference on Autonomous Agents*, 1998.

[20] LB Booker. Editorial. *Adaptive Behavior*, 3(1), 1994.

[21] CM Bowman, PB Danzig, U Manber, and MF Schwartz. Scalable internet resource discovery: Research problems and approaches. *Communications of the ACM*, 37(8):98–107, 1994.

[22] R Brooks and P Maes, editors. *Artificial Life IV*. MIT Press, Cambridge, MA, 1994.

[23] HM Cartwright and GF Mott. Looking around: Using cues from the data space to guide genetic algorithm searches. In RK Belew and LB Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.

[24] T Case. Personal communication, 1995.

[25] F Cecconi, S Mancuso, and D Parisi. Edge eaters: An artificial life approach to image compression. In *Proceedings of Adaptive Computing in Engineering Design and Control (ACEDC 96)*, 1996.

[26] F Cecconi, F Menczer, and RK Belew. Maturation and the evolution of imitative learning in artificial organisms. *Adaptive Behavior*, 4(1):27–48, 1996.

[27] S Chakrabarti, B Dom, and P Indyk. Enhanced hypertext categorization using hyperlinks. In *Proc. ACM SIGMOD*, Seattle, WA, 1998.

[28] S Chakrabarti, B Dom, P Raghavan, S Rajagopalan, D Gibson, and J Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. 7th International World Wide Web Conference*, 1998.

[29] L Chen and K Sycara. WebMate: A personal agent for browsing and searching. In *Proc. 2nd International Conference on Autonomous Agents*, 1998.

[30] D Cliff, P Husband, and I Harvey. Analysis of evolved sensory-motor controller. In *Second European Conference on Artificial Life (ECAL93)*, Brussels, 1993.

[31] WS Cooper. Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *Journal of the American Society for Information Science*, 19:30–41, 1968.

[32] L Dagorn, F Menczer, and P Bach. Tropical predator oceanic fishes and their biological environment: simulation of a multi-behavior ecology. To be submitted to Ecological Modeling.

[33] Y Davidor. A naturally occurring niche and species phenomenon: The model and first results. In RK Belew and LB Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.

[34] P Dayan. Improving generalization for temporal difference learning: the successor representation. *Neural Computation*, 5:613–624, 1993.

[35] PME De Bra and RDJ Post. Information retrieval in the world wide web: Making client-based searching feasible. In *Proc. 1st International World Wide Web Conference*, Geneva, 1994.

[36] KA De Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[37] KA De Jong and J Sarma. On decentralizing selection algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995.

[38] KA DeJong and J Sarma. Generation gaps revisited. In LD Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 19–28. San Mateo, CA: Morgan Kaufmann, 1993.

[39] Digital Equipment Corporation. http://altavista.digital.com.

[40] M Dorigo and LM Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1(1):53–66, 1997.

[41] Encyclopaedia Britannica, Inc. http://www.eb.com.

[42] JM Epstein and RL Axtell. *Growing artificial societies: Social science from the bottom up.* Brookings/MIT Press, 1996.

[43] LJ Eshelman and JD Schaffer. Preventing premature convergence in genetic algorithms by preventing incest. In RK Belew and LB Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.

[44] LJ Eshelman and JD Schaffer. Crossover's niche. In *Proceedings of the 5th International Conference on Genetic Algorithms*, 1993.

[45] RA Fischer, editor. *The genetical theory of natural selection.* Dover Publications, New York, NY, 1930.

[46] JA Fletcher, MA Bedau, and M Zwick. Effect of environmental structure on evolutionary adaptation. In C Adami, RK Belew, H Kitano, and CE Taylor, editors, *Proceedings of the Sixth International Conference on Artificial Life.* MIT Press, 1998.

[47] JA Fletcher, M Zwick, and MA Bedau. Dependence of adaptability on environmental structure in a simple evolutionary model. *Adaptive Behavior*, 4:283–315, 1996.

[48] C Fox. Lexical analysis and stop lists. In *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, 1992.

[49] WB Frakes. Stemming algorithms. In *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, 1992.

[50] M Gilpin. personal communication, 1995.

[51] ME Gilpin. *Group Selection in Predator-Prey Communities.* Monographs in Population Biology. Princeton University Press, Princeton, NJ, 1975.

[52] P Godfrey-Smith. *Complexity and the Function of Mind in Nature.* Cambridge University Press, 1996.

[53] DE Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[54] DE Goldberg and K Deb. A comparative analysis of selection schemes used in genetic algorithms. In G Rawlings, editor, *Foundations of Genetic Algorithms.* Morgan Kaufmann, 1991.

[55] DE Goldberg and J Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the 2nd International Conference on Genetic Algorithms*, 1987.

[56] VS Gordon and D Whitley. Serial and parallel genetic algorithms as function optimizers. In *Proceedings of the 5th International Conference on Genetic Algorithms*, 1993.

[57] SJ Gould and RC Lewontin. The spandrels of san marco and the panglossian paradigm: A critique of the adaptionist programme. *Proceedings of the Royal Society of London*, B205:581–598, 1979.

[58] PB Grosso. *Computer Simulation of Genetic Adaptation: Parallel Subcomponent Interaction in a Multilocus Model.* PhD thesis, University of Michigan, 1985.

[59] F Gruau and D Whitley. Adding learning to the cellular development of neural networks: Evolution and the baldwin effect. *Evolutionary Computation*, 1(3):213–233, 1993.

[60] G Harik. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995.

[61] D Harman. Relevance feedback and other query modification techniques. In *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, 1992.

[62] WE Hart. *Adaptive Global Optimization with Local Search.* PhD thesis, University of California, San Diego, 1994.

[63] WE Hart and RK Belew. Optimization with genetic algorithm hybrids that use local search. In RK Belew and M Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Santa Fe Institute Studies in the Sciences of Complexity. Addison Wesley, Reading, MA, 1996.

[64] WE Hart, TE Kammeyer, and RK Belew. The role of development in genetic algorithms. In *Foundations of Genetic Algorithms 3*, 1994.

[65] D Hartl and A Clarke. *Principles of Population Genetics.* Sinauer Associates, 1989.

[66] GE Hinton and SJ Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987.

[67] JH Holland. *Adaptation in Natural and Artificial Systems.* Univ. of Michigan Press, Ann Arbor, MI, 1975.

[68] JH Holland. *Adaptation in Natural and Artificial Systems.* Bradford Books (MIT Press), Cambridge, MA, 2 edition, 1992.

[69] JH Holland. Echoing emergence: Objectives, rough definitions, and speculations for echo-class models. In G Cowan, D Pines, and D Melzner, editors, *Integrative Themes*, volume XIX of *Santa Fe Institute Studies in the Sciences of Complexity.* Addison Wesley, Reading, MA, 1993.

[70] I Horswill. Analysis of adaptation and environment. *Artificial Intelligence*, 73, 1995.

[71] BA Huberman and CS Glance. Evolutionary games and computer simulations. *Proc. Nat. Acad. Science*, 1993.

[72] BA Huberman, PLT Pirolli, JE Pitkow, and RM Lukose. Strong regularities in world wide web surfing. *Science*, 280(5360):95–97, 1998.

[73] N Jacobi, P Husbands, and I Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Proceedings of the Third European Conference on Artificial Life*, 1995.

[74] T Joachims, D Freitag, and T Mitchell. WebWatcher: A tour guide for the world wide web. In *Proc. International Joint Conference on Artificial Intelligence*, 1997.

[75] DS Johnson. Local optimization and the traveling salesman problem. In *Proceedings of the 17th Colloquium on Automata, Languages and Programming*, pages 446–461, 1990.

[76] TD Johnston. Selective costs and benefits in the evolution of learning. *Advances in the study of behavior*, 12:65–106, 1982.

[77] T Jones and S Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995.

[78] T Jones and M Mitchell. Introduction to the echo model. Working paper 93-12-074, Santa Fe Institute, 1993.

[79] LP Kaelbling. *Learning in Embedded Systems.* MIT Press, Cambridge, MA, 1993.

[80] P Kareiva. Space: The final frontier for ecological theory. *Ecology*, 75(1), 1994. Special Feature.

[81] H Kitano. Evolution of metabolism for morphogenesis. In R Brooks and P Maes, editors, *Artificial Life IV*, pages 49–58, Cambridge, MA, 1994. MIT Press.

[82] E Koutsoupias, C Papadimitriou, and M Yannakakis. Searching a fixed graph. In *Proc. 23rd International Colloquium on Automata, Languages and Programming*, pages 280–289, Berlin, Germany, 1996. Springer-Verlag.

[83] CJ Krebs. *Ecology: the experimental analysis of distribution and abundance.* Harper & Collins, New York, NY, 1994.

[84] LA Krubitzer, RK Belew, CRB Boake, E Boncinelli, EA Brenowitz, S de-Belle, JS Edwards, WPM Geraerts, CP Kyriacou, GL Gabor Miklos, and F vonSchilcher. Group report: How do evolution and behavior interact? In RJ Greenspan and CP Kyriacou, editors, *Flexibility and Constraint in Behavioral Systems.* John Wiley and Sons, 1994.

[85] KN Laland. On the evolutionary consequences of sexual imprinting. *Evolution*, 48(2):477–489, 1994.

[86] M Land. *Global/Local Search Hybrids for Combinatorial Optimization.* PhD thesis, University of California, San Diego, 1998. Forthcoming.

[87] M Land, JJ Sidorowich, and RK Belew. Using genetic algorithms with local search for thin film metrology. In *Proc. Seventh International Conference on Genetic Algorithms*, 1997.

[88] RR Larson. Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Proc. 1996 Annual ASIS Meeting*, 1996.

[89] SR Lawrence and CL Giles. Searching the world wide web. *Science*, 280:98–100, 1998.

[90] SA Levin, B Grenfell, A Hastings, and AS Perelson. Mathematical and computational challenges in population biology and ecosystems science. *Science*, 275:334–343, 1997.

[91] R Levins. *Evolution in changing environments.* Princeton University Press, Princeton, NJ, 1968.

[92] DD Lewis. Active by accident: Relevance feedback in information retrieval. In *AAAI Fall Symposium on Active Learning*, 1995.

[93] DD Lewis. Information retrieval and the statistics of large data sets. In *Proc. NRC Massive Data Sets Workshop*, Washington, DC, 1996.

[94] DD Lewis. Challenges in machine learning for text classification. In *Proc. 9th Annual Conference on Computational Learning Theory*, New York, NY, 1997.

[95] H Lieberman. Autonomous interface agents. In *Proc. ACM Conference on Computers and Human Interface*, Atlanta, GA, 1997.

[96] L-J Lin. Self-improving reactive agents based on reinforcement learning, planning, and teaching. *Machine Learning*, 8:293–321, 1992.

[97] ML Littman. Simulations combining evolution and learning. In RK Belew and M Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and algorithms*, Santa Fe Institute Studies in the Sciences of Complexity. Addison Wesley, Reading, MA, 1995.

[98] ML Littman and DH Ackley. Adaptation in constant utility non-stationary environments. In RK Belew and LB Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.

[99] J Liu. Reactive agents for adaptive image analysis. In *Proc. 2nd International Conference on Autonomous Agents*, 1998.

[100] RM Lukose and BA Huberman. Surfing as a real option. In *Proc. 4th International Conference on Computational Economics*, 1998.

[101] Lycos. http://www.lycos.com.

[102] RH MacArthur. Some generalized theorems of natural selection. *Proc. Nat. Acad. Sci.*, 48:1893–1897, 1962.

[103] W Macready. Tailoring mutation to landscape properties. In *Proc. 7th Annual Conference on Evolutionary Programming*, San Diego, CA, 1998.

[104] P Maes, editor. *Designing autonomous agents: theory and practice from biology to engineering and back.* MIT Press, Cambridge, MA, 1990.

[105] P Maes, editor. *Designing autonomous agents.* CACM special issue. MIT/Elsevier, 1993.

[106] P Maes. Agents that reduce work and information overload. *Comm. of the ACM*, 37(7):31–40, 1994.

[107] SW Mahfoud. Population sizing for sharing methods. In *Foundations of Genetic Algorithms 3*, 1994.

[108] SW Mahfoud. A comparison of parallel and sequential niching methods. In *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995.

[109] MJ Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, 1992.

[110] J McInerney. *Biologically Influenced Algorithms and Parallelism in Non-Linear Optimization.* PhD thesis, University of California, San Diego, 1992.

[111] E Meir. *EcoBeaker*, 1994. Ecological Simulation Program, Pre-Release Version 0.8.

[112] F Menczer. Changing latent energy environments: A case for the evolution of plasticity. Technical Report CS94-336, University of California, San Diego, 1994.

[113] F Menczer. Internet: Vita da ragni. *Sistemi Intelligenti*, 7(3):421–442, 1995. In Italian.

[114] F Menczer. ARACHNID: Adaptive retrieval agents choosing heuristic neighborhoods for information discovery. In *Proc. 14th International Conference on Machine Learning*, 1997.

[115] F Menczer and RK Belew. Lee: A model for artificial life complexity. Technical Report CS93-298, University of California, San Diego, 1993.

[116] F Menczer and RK Belew. Lee: A tool for artificial life simulations. Technical Report CS93-301, University of California, San Diego, 1993.

[117] F Menczer and RK Belew. Evolving sensors in environments of controlled complexity. In R Brooks and P Maes, editors, *Artificial Life IV*, Cambridge, MA, 1994. MIT Press.

[118] F Menczer and RK Belew. From complex environments to complex behaviors. *Adaptive Behavior*, 4:317–363, 1996.

[119] F Menczer and RK Belew. Latent energy environments. In RK Belew and M Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Santa Fe Institute Studies in the Sciences of Complexity. Addison Wesley, Reading, MA, 1996.

[120] F Menczer and RK Belew. Arachnid software demo. Proc. 1st Intl. Conf. Autonomous Agents, 1997.

[121] F Menczer and RK Belew. Adaptive information agents in distributed textual environments. In *Proc. 2nd International Conference on Autonomous Agents*, Minneapolis, MN, 1998.

[122] F Menczer and RK Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. Technical Report CS98–579, University of California, San Diego, 1998. Submitted to Machine Learning Journal.

[123] F Menczer and RK Belew. Local selection. In *Proc. 7th Annual Conference on Evolutionary Programming*, San Diego, CA, 1998.

[124] F Menczer, RK Belew, and W Willuhn. Artificial life applied to adaptive information agents. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995.

[125] F Menczer and D Parisi. Evidence of hyperplanes in the genetic learning of neural networks. *Biological Cybernetics*, 66:283–289, 1992.

[126] F Menczer and D Parisi. A model for the emergence of sex in evolving networks: Adaptive advantage or random drift? In F Varela and P Bourgine, editors, *Toward a Practice of Autonomous Systems: Proc. First European Conference on Artificial Life*, Cambridge, MA, 1992. MIT Press.

[127] F Menczer and D Parisi. Recombination and unsupervised learning: effects of crossover in the genetic optimization of neural networks. *Network*, 3:423–442, 1992.

[128] F Menczer, W Willuhn, and RK Belew. An endogenous fitness paradigm for adaptive information agents. In *CIKM Workshop on Intelligent Information Agents*, 1994.

[129] Metacrawler. http://www.metacrawler.com.

[130] O Miglino and D Parisi. Evolutionary stable and unstable strategies in neural networks. In *Proc. International Joint Conference on Neural Netorks*, volume 2, pages 1448–1453, 1991.

[131] M. Milinski and GA Parker. Competition for resources. In JR Krebs and NB Davies, editors, *Behavioral Ecology*. Blackwell, Oxford, UK, 1991.

[132] M Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.

[133] M Mitchell and S Forrest. Genetic algorithms and artificial life. *Artificial Life*, 1(3), 1994.

[134] T Mitchell. *Machine Learning*, chapter 4. McGraw-Hill, New York, NY, 1997.

[135] AE Monge and CP Elkan. The WEBFIND tool for finding scientific papers over the worldwide web. In *Proceedings of the 3rd International Congress on Computer Science Research*, 1996.

[136] A Moukas and G Zacharia. Evolving a multi-agent information filtering solution in amalthaea. In *Proc. 1st International Conference on Autonomous Agents*, 1997.

[137] R Naviaux. Personal communication.

[138] S Nolfi, JL Elman, and D Parisi. Learning and evolution in neural networks. *Adaptive Behavior*, 3(1):5–28, 1994.

[139] S Nolfi, D Floreano, O Miglino, and F Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In *Artificial Life IV*, 1994.

[140] L Pack Kaelbling, ML Littman, and AW Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[141] CH Papadimitriou. On selecting a satisfying truth assignment. In *32nd FOCS*, 1991.

[142] M Pazzani, J Muramatsu, and D Billsus. Syskill & Webert: Identifying interesting web sites. In *Proc. National Conference on Artificial Intelligence (AAAI96)*, 1996.

[143] D Precup and RS Sutton. Exponentiated gradient methods for reinforcement learning. In *Proc. 14th International Conference on Machine Learning*, 1997.

[144] E Rivlin, R Botafogo, and B Shneiderman. Navigating in hyperspace: designing a structure-based toolbox. *Communications of the ACM*, 37(2):87–96, 1994.

[145] SE Robertson and K Spark Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, pages 129–146, 1976.

[146] CD Rosin and RK Belew. Methods for competitive co-evolution: Finding opponents worth beating. In *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995.

[147] CD Rosin, RS Halliday, WE Hart, and RK Belew. A comparison of global and local search methods in drug docking. In *Proc. Seventh International Conference on Genetic Algorithms*, 1997.

[148] OE Rössler. Design for autonomous chemical growth under different environmental constraints. In R Rosen and FM Snell, editors, *Progress in Theoretical Biology*. Academic Press, New York, NY, 1972.

[149] OE Rössler. Adequate locomotion strategies for an abstract environment — a relational approach. In M Conrad M, W Guttinger W, and MD Cin, editors, *Physics and Mathematics of the Nervous System*, pages 399–418. Sringer-Verlag, New York, NY, 1974.

[150] J Roughgarden. Density-dependent natural selection. *Ecology*, 52:453–468, 1971.

[151] J Roughgarden, A Bergman, S Shafir, and C Taylor. Adaptive computation in ecology and evolution: A guide for future research. In RK Belew and M Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison Wesley, Reading, MA, 1996.

[152] J Roughgarden, RM May, and SA Levin, editors. *Perspectives in Ecological Theory*. Princeton University Press, 1989.

[153] DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation. In DE Rumelhart and JL McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. Bradford Books (MIT Press), Cambridge, MA, 1986.

[154] DE Rumelhart and JL McClelland, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. Bradford Books (MIT Press), Cambridge, MA, 1986.

[155] D Rus, R Gray, and D Kotz. Transportable information agents. In *Proc. 1st International Conference on Autonomous Agents*, 1997.

[156] SJ Russell and P Norvig. *Artificial Intelligence: A modern approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[157] G Salton. The 'generality' effect and the retrieval evaluation for large collections. *Journal of the American Society for Information Science*, 23:11–22, 1972.

[158] G Salton and C Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[159] G Salton and MJ McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.

[160] NH Schumaker. Using landscape indices to predict habitat connectivity. *Ecology*, 77(4):1210–1255, 1996.

[161] H-P Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, NY, 1981.

[162] B Selman, DG Mitchell, and HJ Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81:17–29, 1996.

[163] J Shakes, M Langheinrich, and O Etzioni. Dynamic reference sifting: A case study in the homepage domain. In *Proc. 6th International World Wide Web Conference*, 1997.

[164] K Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:111–121, 1972.

[165] K Sparck Jones. Experiments in relevance weighting of search terms. *Information Processing and Management*, 15:133–144, 1979.

[166] E Spertus. Parasite: Mining structural information on the web. In *Proc. 6th International World Wide Web Conference*, 1997.

[167] P Srinivasan, 1998. Personal communication.

[168] SC Stearns. *The Evolution of Life Histories.* Oxford University Press, New York, NY, 1992.

[169] AM Steier. *Statistical semantics of phrases in hierarchical contexts.* PhD thesis, Computer Science and Engineering Department, U. C. San Diego, 1994.

[170] AM Steier and RK Belew. Exporting phrases: A statistical analysis of topical language. In R Casey and B Croft, editors, *2nd Symposium on Document Analysis and Information Retrieval*, 1994.

[171] R Sutton. Reinforcement learning and information access. In *AAAI Spring Symposium on Machine Learning and Information Access*, 1996.

[172] RS Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[173] PM Todd. Environment structure and behavior. *Adaptive Behavior*, 4(3/4), 1996. Special Issue.

[174] PM Todd and GF Miller. On the sympatric origin of species: Mercurial mating in the quicksilver model. In *Proc. Fourth International Conference on Genetic Algorithms*, 1991.

[175] PM Todd and SW Wilson. Environment structure and adaptive behavior from the ground up. In J-A Meyer, HL Roitblat, and SW Wilson, editors, *From Animals to Animats 2: Proc. Second Intl. Conf. on Simulation of Adaptive Behavior*, pages 11–20. Camridge, MA: MIT Press/Bradford Books, 1993.

[176] PM Todd, SW Wilson, AB Somayaji, and Y Yanco. The blind breeding the blind: Adaptive behavior without looking. In J-A Meyer, P Husband, and SW Wilson, editors, *From Animals to Animats 3: Proc. Third Intl. Conf. on Simulation of Adaptive Behavior*. Camridge, MA: MIT Press/Bradford Books, 1994.

[177] PM Todd and HA Yanco. Environmental effects on minimal behaviors in the minimat world. *Adaptive Behavior*, 4:365–413, 1996.

[178] M Tomasello, AC Kruger, and HH Ratner. Cultural learning. *Behavioral and Brain Science*, 16:495–511, 1993.

[179] CJ van Rijsbergen. *Information Retrieval.* Butterworths, London, 1979. Second edition.

[180] CC Vogt and GW Cottrell. Predicting the performance of linearly combined ir systems. In *Proceedings of the ACM SIGIR Conference*, 1998.

[181] W3C Reference Library. Libwww version 5.1. http://www.w3.org/Library, 1997.

[182] CH Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150:563–565, 1942.

[183] CJCH Watkins. *Learning from delayed rewards.* PhD thesis, King's College, Cambridge, UK, 1989.

[184] R Weiss, B Velez, M Sheldon, C Nemprempre, P Szilagyi, and DK Giffor. Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *Proc. Seventh ACM Conference on Hypertext*, 1996.

[185] GM Werner and MG Dyer. Evolution of communication in artificial organisms. In CG Langton, C Taylor, JD Farmer, and S Rasmussen, editors, *Artificial Life II*, volume X of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 659–687, Reading, MA, 1992. Addison Wesley.

[186] LD Whitley, S Dominic, and R Das. Genetic reinforcement learning with multilayer neural networks. In RK Belew and LB Booker, editors, *Fourth International Conference on Genetic Algorithms*, pages 562–569. San Mateo, CA: Morgan Kaufmann, 1991.

[187] SW Wilson. The animat path to ai. In J-A Meyer and SW Wilson, editors, *From Animals to Animats: First Intl. Conf. on Simulation of Adaptive Behavior*, pages 15–21. Cambridge, MA: Bradford Books (MIT Press), 1991.

[188] S Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proc. 6th Intl. Conf. on Genetics*, pages 356–366, 1932.

[189] S Wright. Physiological genetics, ecology of populations, and natural selection. *Perspect. Biol. and Med.*, 3:107–151, 1959.

[190] LA Zhivotovsky, A Bergman, and MW Feldman. A model of individual adaptive behavior in a fluctuating environment. In RK Belew and M Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Santa Fe Institute Studies in the Sciences of Complexity. Addison Wesley, Reading, MA, 1995.