Hierarchical Chunking in Classifier Systems

Gerhard Weiß
Institut für Informatik, Technische Universität München
D-80290 München, Germany
weissg@informatik.tu-muenchen.de

Abstract

Two standard schemes for learning in classifier systems have been proposed in the literature: the bucket brigade algorithm (BBA) and the profit sharing plan (PSP). The BBA is a local learning scheme which requires less memory and lower peak computation than the PSP, whereas the PSP is a global learning scheme which typically achieves a clearly better performance than the BBA. This "requirement versus achievement" difference, known as the locality/globality dilemma, is addressed in this paper. A new algorithm called hierarchical chunking algorithm (HCA) is presented which aims at synthesizing the local and the global learning schemes. This algorithm offers a solution to the locality/globality dilemma for the important class of reactive classifier systems.

The contents is as follows. Section 1 describes the locality/globality dilemma and motivates the necessity of its solution. Section 2 briefly introduces basic aspects of (reactive) classifier systems that are relevant to this paper. Section 3 presents the HCA. Section 4 gives an experimental comparison of the HCA, the BBA and the PSP. Section 5 concludes the paper with a discussion and an outlook on future work.

Motivation

The foundations for classifier systems (CSs for short) were laid by Holland (1975) and Holland and Reitman (1978). CSs are parallel, message-passing, rule-based systems that are capable of environmental interaction and of reinforcement learning through credit assignment and rule modification. Up to now two different learning schemes for credit assignment in CSs have been proposed: the *bucket brigade algorithm* (BBA for short, e.g. Booker, 1982; Holland, 1985, 1986; Riolo, 1988) and the *profit sharing plan* (PSP for short, e.g. Grefenstette, 1988; Holland & Reitman, 1978). These two schemes significantly differ from each other in that the BBA is a local learning scheme which incrementally assigns credit whenever the CS interacts with its environment, whereas the PSP is a global learning scheme which episodically assigns credit only when the CS receives a reinforcement signal from its environment. A consequence of this difference, known as the *locality/globality dilemma*, is that the BBA requires less memory and less peak computation than the PSP, but the PSP typically achieves a better performance level than the BBA. Roughly, this is because the PSP needs to maintain detailed information about the past activities carried out by the CS, whereas the BBA has difficulties in generating long activity sequences that are both useful and stable.

There is a lot of work centered around the locality/globality dilemma in the context of the BBA and the PSP; for instance, see the performance comparisons of the BBA and the PSP described in (Grefenstette, 1988; Weiß, 1992) and the investigations and considerations on the formation and maintenance of activity sequences presented e.g. in (Holland, 1985; Riolo, 1987, 1989; Robertson & Riolo, 1988; Wilson, 1987). However, despite this work it is still an open and challenging research issue to develop a local algorithm like the BBA that possesses the learning abilities of a global algorithm like the PSP. This issue has been addressed by the work reported in this paper. A new learning algorithm called hierarchical chunking algorithm is presented which offers a solution to the locality/globality dilemma for the important class of reactive CSs, that is, CSs whose activity is, at each time, exclusively triggered by the information they have about the actual environmental state.

An Introduction to Classifier Systems

This section gives a brief introduction to basic aspects of CSs. For a more comprehensive introduction the reader is referred to (Booker, Goldberg, & Holland, 1989; Goldberg, 1989; Wilson & Goldberg, 1989).

The prototypical organization of a CS can be described as follows. Structurally, a CS is composed of four *major components*:

- An input interface which consists of at least one detector providing information about the environment in the form of messages.
- An output interface which consists of at least one effector enabling the system to interact with the environment.
- A classifier list which consists of *condition/action* rules called classifiers. The condition part specifies

the messages that satisfy the classifier, and the action part specifies the messages to be sent when the classifier is activated. Associated with each classifier is a quantity called its strength.

- A message list which contains the messages sent by the detectors and the classifiers.

Functionally, the overall activity of a CS results from the repeated execution of the following *major cycle*:

1. Activation of the input interface: The actual detector messages are added to the message list.
2. Activation of the classifier list: The system decides which classifiers are allowed to produce new messages. This is done by running a strength-based competition between all satisfied classifiers.
3. Activation of the output interface: The system interacts with its environment in dependence on the contents of the message list.
4. Credit assignment: Strength-update rules are applied to adjust the classifier strengths such that they reflect the classifiers' relevance to goal attainment.
5. Rule modification: Some classifiers are modified by a genetic algorithm.

An important class of restricted CSs is that of *reactive CSs*. In these systems only a single classifier is selected during each major cycle, and this selection is guided only by the actual detector messages (and not by internal messages). Reactive CSs have been extensively used for theoretical and experimental studies (e.g., see Wilson, 1985; Grefenstette, 1988), and they are also taken as a basis for the work described in this paper.

As mentioned in section 1, the BBA and the PSP have been proposed as *credit assignment schemes*. In its elementary form, the BBA locally updates the classifier strengths as follows. Whenever a competition runs, each satisfied classifier C_j makes a bid Bid_j,

$$Bid_j = b \cdot Str_j \cdot Spec_j \tag{1}$$

where b is a small constant called risk factor, Str_j is C_j's strength (initialized with a constant Str^{init} for all classifiers) and $Spec_j$ is C_j's specificity (a quantity expressing the classifier's relevance to particular environmental situations). The probability that a bidding classifier C_j wins the competition is given by

$$\frac{Bid_j}{\sum_{C_l \in \mathcal{B}} Bid_l} \tag{2}$$

where $\mathcal{B}$ is the set of all bidding classifiers. A winning classifier reduces its strength by the amount of its bid, and hands this amount back to its predecessors, that is, to those classifiers whose preceding activities enabled it to become active. (The winning classifiers pay for the privilege of being active, and the predecessors are rewarded for appropriately setting up the environment.) Formally, if C_j is a winning classifier and $\mathcal{P}_j$ is the set of its predecessors, then the strengths are modified according to the following rules:

$$Str_j = Str_j - Bid_j \quad \text{and} \tag{3}$$

$$Str_i = Str_i + \frac{Bid_j}{|\mathcal{P}_j|} \quad \forall C_i \in \mathcal{P}_j \quad . \tag{4}$$

Additionally, if an external reward is received from the environment, then it is equally distributed among the classifiers that sent the effector-activating messages. The idea underlying the BBA is to internally reward classifiers that are useful in achieving specific goals but that are not active when the external reward is obtained.

The PSP in its elementary form updates the classifier strengths as follows. Bidding and selection of the winning classifiers is done according to (1) and (2), respectively. In contrast to the BBA, the PSP globally rewards sequences of active classifiers. At the end of each episode (i.e., whenever an external reward Ext is received) the strength Str_j of each classifier C_j that was active at least one time during this episode is modified according to rule

$$Str_j = Str_j - Bid_j + b \cdot Ext \tag{5}$$

where b is the risk factor used in bid calculation.

There are many variants of the BBA – e.g., see (Dorigo, 1991; Huang, 1989; Riolo, 1990; Weiß, 1991; Wilson, 1985, 1987) – as well as of the PSP – e.g., see (Grefenstette, 1988; Holland & Reitman, 1978; Weiß, 1992). However, none of these variants solves the locality/globality dilemma.

The Hierarchical Chunking Algorithm

Chunking is an experience-based learning mechanism which was originally proposed within the frame of a psychological model of memory organization (Miller, 1956). According to this model, chunking refers to the process of correlating pieces of knowledge or sensory input in such a way that they can be treated and used as a single memory unit or "chunk" on its own. If it is explicitly assumed that already existing chunks can be used for building new ones, then this process is referred to as *hierarchical* chunking. Hierarchical chunking has received much attention in psychology as well as in artificial intelligence; for instance, see (Chase & Simon, 1973; Chi, 1978; Newell & Rosenbloom, 1981; Rosenbloom, 1983; Rosenbloom & Newell, 1986). In the following, a new algorithm called *hierarchical chunking algorithm* (HCA for short) is described which was designed to solve the locality/globality dilemma for reactive CSs; this algorithm synthesizes the local (BBA-type) and the global (PSP-type) learning paradigms by applying the mechanism of hierarchical chunking to successful sequences of active classifiers.

Under the HCA each classifier C_j is assumed to be of the *generalized form* $Cond_j/Act_j$, where

$$Cond_j = \langle c_{j1}, \ldots, c_{jr_j} \rangle \tag{6}$$

specifies the tuples $(m_1, \ldots, m_{r_j})$ of messages m_k that satisfy C_j and

$$Act_j = \langle a_{j1}, \ldots, a_{js_j} \rangle \tag{7}$$

specifies the sequence $(m_1, \ldots, m_{s_j})$ of messages m_k to be sent when C_j wins the competition ($r_j, s_j \in \mathbb{N}$ for all j). Each message sent within a sequence of messages is immediately processed by the effectors; with

that, a CS is able to carry out several environmental interactions (instead of just a single interaction) within one major cycle.[1] In the following, the length of Act_j of a classifier C_j is called the level of C_j, and is denoted by L_j; furthermore, if $L_j = 1$, then C_j is said to be an elementary classifier, and if $L_j > 1$, then C_j is said to be an extended classifier or macro-classifier or a chunk. As an illustration of this generalized view of classifiers, consider a CS which has to navigate from the start state S to the goal state G in the maze shown in figure 1. Assuming that the CS is able to interact with its environment by moving, in each location, to one of the neighbouring locations, an elementary classifier might represent the behavioral rule "If the current location is in the upper-left area, then move one step to the right", and an extended classifier might represent the rule "If the current location is in the middle area, then first move one step to the left and then one step down".

The HCA arranges a *hierarchical competition* between the classifiers for the right to produce new messages. Among all satisfied classifiers, only the highest-level classifiers are allowed to make bids and to compete against each other. More exactly, if $\mathcal{S}$ is the set of all satisfied classifiers (in the actual cycle), then only the classifiers being contained in the set $\mathcal{B}$,

$$\mathcal{B} = \{C_j : C_j \in \mathcal{S} \text{ and } L_j \geq L_l \ \forall \ C_l \in \mathcal{S}\}, \quad (8)$$

are allowed to participate in the competition. Each classifier $C_j \in \mathcal{B}$ calculates a bid according to (1), and the probability that C_i wins the competition is given by (2).

The HCA modifies the classifier strengths similar to the implicit BBA proposed by Wilson (1985). Compared to the general BBA described in section 2, the HCA takes a simplified point of view of a classifier's predecessor which bases on the assumption that the temporal order of active classifiers is imposed by the environment. If the classifiers C_i and C_j won the competition in the previous and the actual cycle, respectively, then C_i is considered to be the only predecessor of C_j (i.e., $\mathcal{P}_j = \{C_i\}$), and their strengths are adjusted according to (3) and (4). In this way a linkage is established between time-adjacent classifiers.

At the beginning (i.e., before learning takes place), the classifier list is assumed to contain only elementary classifiers. The extended classifiers are dynamically formed and dissolved under the HCA in the course of environmental interaction. The formation and dissolution of extended classifiers correspond to the formation and dissolution of chunks, respectively, and make up the core of the HCA. The *criteria* used for triggering formation and dissolution are conceptually similar to (and, in fact, have been inspired by) the group-development criteria proposed by Weiß (1993a, 1993b) in the context of BBA-based multi-agent learning. Formally, the formation and dissolution criteria are as follows. Let C_i be the preceding winning classifier, C_j the actual winning classifier, $\mathcal{B}$ the set of all actual bidding classifiers, and $\mu = \frac{1}{|\mathcal{B}|}\sum_{C_l \in \mathcal{B}} Str_l$ the average strength of all classifiers contained in $\mathcal{B}$. A new (extended) classifier $Cond_i/Act_i \circ Act_j$ with

$$Act_i \circ Act_j = \langle a_{i1}, \ldots, a_{is_i}, a_{j1}, \ldots, a_{js_j} \rangle \quad (9)$$

is formed out of C_i and C_j, if and only if

$$Str_j \geq \mu + \sigma \cdot \sqrt{\frac{1}{|\mathcal{B}|} \sum_{C_l \in \mathcal{B}} (Str_l - \mu)^2} \quad (10)$$

where σ is a constant called formation factor. The strength of the new classifier is initialized with Str_j. Conversely, an (extended) classifier $C_k \in \mathcal{B}$ is dissolved and removed from the classifier list, if and only if

$$Str_k \leq \mu - \rho \cdot \sqrt{\frac{1}{|\mathcal{B}|} \sum_{C_l \in \mathcal{B}} (Str_l - \mu)^2} \quad (11)$$

where ρ is a constant called dissolution factor. With equations (10) and (11), formation as well as dissolution take place if the strength of a classifier is not within the "standard range" that can be expected given the average strength and the strength deviation of the bidding classifiers. Because both criteria are defined over the strength values, extended classifiers are formed and dissolved in an experience-based and goal-directed manner. Furthermore, because strength adjustment, formation and dissolution are strongly interrelated and mutually influence each other, the HCA endows a reactive CS with highly dynamic adaptation and learning abilities. (It should be noted that the HCA does not require more information for realizing learning than the BBA; in particular, the HCA forms and dissolves classifiers on the basis of local information and, in contrast to the PSP, without the need of an episodical trace of all – useless and useful – winning classifiers.)

Experimental Analysis

As an initial learning domain a navigation task first introduced by Sutton (1990) has been chosen. This type of task captures the essential features of the locality/globality dilemma, and it is well suited for experimentally comparing the HCA, the BBA and the PSP. Subsequently experiments on the task of learning to navigate through the maze shown in figure 1 are described.[2] The maze is a 10 by 7 grid of locations, where the shaded locations are obstacles that cannot be entered. In each location the CS can move to each of the neighbouring locations, except where such a movement would take the system into an obstacle or outside the maze. The CS has to learn to move from each possible location of the maze to a fixed location called goal

[1] The definition of $Cond_i$ corresponds to the traditional notion of a classifier's condition part. Against that, the definition and interpretation of Act_i establishes a rather unconventional notion of a classifier's action part which opens up new possibilities in the coupling between a CS and its environment.

[2] A number of experiments with other mazes has been performed, varying the grid size, the directions in which the system can move, and the number of possible starting states, the position of the goal state, and the number and positions of the obstacles. The results of these experiments are qualitatively similar to those presented below.

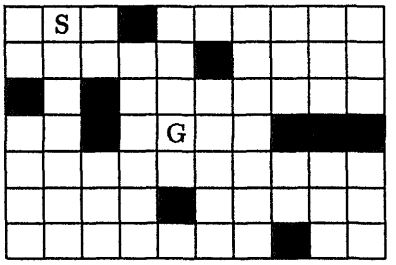

Figure 1: Maze.

state (G). If and only if the goal state is reached, then a non-zero external reward is provided, a new location called start state (S) is randomly chosen and the next episode starts.

Some implementational details. A problem of every system that works with an internal representation of its environment is the mapping problem, that is, the problem that the system can produce discontinuous mappings from input to output even if the environment is continuous (and vice versa). This problem also exists for CSs (Wilson & Goldberg, 1989), and in order to avoid or at least strongly reduce it, the following domain-specific *decimal coding* is used. If the CS is in the location (x, y), then the actual detector message simply is of the form (x, y). Furthermore, each classifier is of the form $\langle u, v\rangle/\langle w_1, \ldots, w_s\rangle$, with $\langle u, v\rangle$ being its condition part and $\langle w_1, \ldots, w_s\rangle$ being its action part ($u \in \{1, \ldots, 10\}$, $v \in \{1, \ldots, 7\}$ and $w_i \in \{0, \ldots, 7\}$). Associated with each classifier C_j is an integer M_j called its matching radius. M_j is randomly chosen from the integer interval $[0, \ldots, M^{max}]$, and is used to define C_j's specificity as $Spec_j = \frac{1}{1+M_j}$. (The smaller a classifier's matching radius, the higher is its specificity, and reversely.) A classifier C_j having $\langle u, v\rangle$ as its condition part matches each detector message (x, y) with $x \in [u - M_j, \ldots, u + M_j]$ and $y \in [v - M_j, \ldots, v + M_j]$. A classifier having $\langle w_1, \ldots, w_s\rangle$ as its action part codes the activity sequence "First go to direction w_1, then to direction w_2, ..., and finally to direction w_s", where direction "0" is interpreted as "north", "1" as "northeast", "2" as "east", and so on. (As an illustration, consider the classifier $C_j = \langle 9, 6\rangle/\langle 5, 6, 4\rangle$, and assume that $M_j = 1$. This classifier matches the detector messages (8,5), (8,6), (8,7), (9,5), (9,6), (9,7), (10,5), (10,6), and (10,7), and codes for the activity sequence "First go one step southwest, then one step west, and finally one step south".)[3]

In order to guarantee the system's capacity to act, a variant of Wilson's (1985) *create operation* has been implemented as follows. Whenever the system enters a location (x, y) whose associated detector message is not matched by any classifier, then a new elementary classifier $C_j = \langle u, v\rangle/\langle w_1\rangle$ is created, where u is randomly chosen from the interval $[x - M_j, \ldots, x + M_j]$, v is randomly chosen from $[y - M_j, \ldots, y + M_j]$, and w_1 is randomly chosen from $\{0, \ldots, 7\}$. With that, the system never stops moving around and searching for the goal state.

In the experiments a slightly modified, more *"reactive" PSP* has been used: instead of adjusting the strengths of all classifiers that won during an episode, only the strengths of the last 4 winning classifiers are adjusted according to (5). This modification is consistent with the general notion of a reactive system; in particular, it is realistic to assume that a purely reactive CS is only "aware" of the last few actions, no matter when the last external reward was received.

Finally, some details on the implemented *genetic algorithm*. The genetic algorithm is applied with probability 0.04 at the end of each episode. If applied, 5 percent of the classifiers, which are selected with probability proportional to the inverse of their strengths, are replaced by new classifiers. The new classifiers are created as follows. Until no further classifier is required, a classifier C_j is selected with probability proportional to its strength and mutated, resulting in a new classifier C'_j. If C_j is of the form $\langle u, v\rangle/\langle w_1, \ldots, w_s\rangle$ and M_j is its matching radius, then C'_j is of the form $\langle u', v'\rangle/\langle w'_1, \ldots, w'_s\rangle$ with $u' = u + a$, $v' = v + b$ and $w'_k = (w_k + c_k)\text{mod}8$ for all $k \in \{1, \ldots, s\}$, where a and b are randomly chosen from the integer interval $[-M_j, \ldots, +M_j]$ and c_k is randomly chosen from $[-1, 0, +1]$. The matching radius of C'_j is randomly chosen from $[0, \ldots, M^{max}]$. No crossover operator is applied. (In other experiments not described in this paper we found that the standard crossover operators are rather inefficient for the learning domain under consideration, since they typically produce classifiers which represent illegal moves.)

Figure 2 shows the performance profiles of the PSP, the HCA, the BBA and a random-walk algorithm (i.e., an algorithm which randomly and with uniform probability selects, in each location, a legal direction and moves one step in this direction). The parameter setting was as follows: $b = 0.1$, $Str^{init} = Ext = 1000$, $\sigma = \rho = 2$, and $M^{max} = 3$. (The classifier system turned out to robust over a broad range of the parameters, and the learning effects reported here are not restricted to exactly this setting.) Each curve shows, averaged over 100 runs, for each of the episodes 1 to 1000 the number of decisions (cycles) required to reach the goal state. In each run the CS was initialized with a set of 100 randomly generated classifiers. At the beginning of learning, each learning algorithm started at the random performance level. Each of the three learning curves rapidly falls within the first 30 episodes. (Interestingly, with that the PSP, the HCA and the BBA led to an early behavioral improvement much like the dynamic-programming approaches investigated by Sutton (1990) did for the same type of task.) After about episode 40, the curves of the PSP and the HCA continuously decrease; the curve of the BBA requires a longer period to become smooth,

[3]Traditionally, a tertiary coding over the alphabet $\{0, 1, \#\}$ is used, where # acts as a don't-care symbol. Obviously, this is a very problematic, discontinuous coding for the task domain under consideration, because in this case the condition part of a classifier can match detector messages that represent completely different, non-adjacent locations.

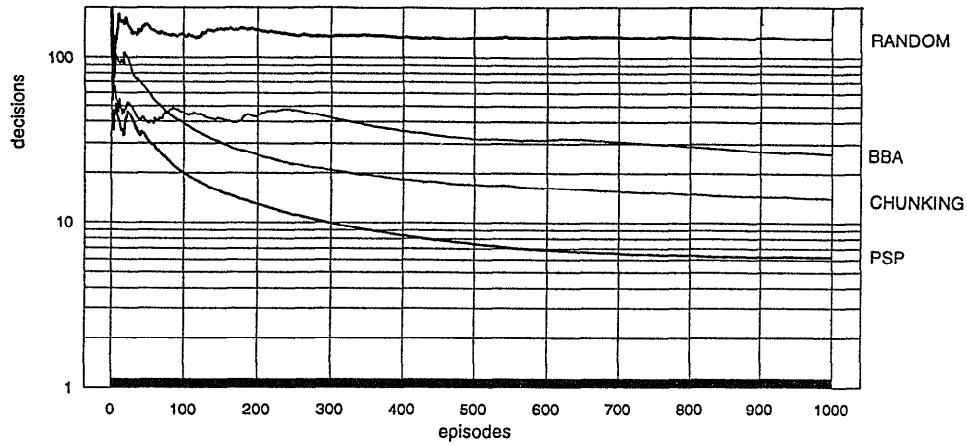

Figure 2: Performance Profiles.

namely about 230 episodes. Averaged over the last 100 episodes, the mean episode length achieved by the PSP, the HCA, the BBA and the random-walk algorithm is 6.2, 15.4., 27.3 and 142.8, respectively. (The behavior of the learning algorithms was observed up to episode 5000. After episode 1000 the performance levels of the three learning algorithms did not further improve and remained almost constant.) Obviously, each of the three learning algorithms performed significantly better than the random-walk algorithm. In particular, after about 85 episodes, the curve of the HCA runs between the curves of the BBA and the PSP: the HCA clearly outperformed the BBA and, at the same time, remained below the performance level of the PSP. This illustrates that hierarchical chunking is an appropriate mechanism for synthesizing local and global learning principles, and that the HCA successfully combines BBA-type and PSP-type learning.

(It is worth to note that even the best performing algorithm, the PSP, left room for improvement, since the minimal episode length, averaged over all legal positions, is 3.0. This shows that CSs, after more than 15 years of existence, still establish an open and challenging area of research on machine learning.)

Conclusion

The HCA attacks the locality/globality dilemma in the context of reactive CSs by bringing together local and global learning principles known from the BBA and the PSP, respectively. On the one side, the HCA retains the local strength adjustment rules of the BBA. On the other side, by introducing the concept of extended classifiers or chunks and by providing mechanisms for their formation and dissolution, the HCA achieves global adjustment qualities much like the PSP does. As a consequence, the HCA approaches to both the lower computational requirements of the BBA and the higher performance level of the PSP.

Wilson and Goldberg (1987) proposed to introduce higher organizational units in the learning and performance processes of a CS in order to cope with the *chaining problem* (i.e., the problem of generating and maintaining long chains of active classifiers) as well as with the *cooperator/competitor dilemma* (i.e., the dilemma that classifiers being active in a chain are cooperative w.r.t. strength adjustment but competitive w.r.t. the selection mechanism of the genetic algorithm). The HCA is much in the spirit of this proposal: the chunks formed and dissolved under the HCA act as such organizational units, since they eliminate (or at least greatly reduce) the need for long chains of elementary classifiers.

Like the standard PSP, the HCA in its present form is not applicable to general CSs. This is an important objection because general CSs, compared to reactive ones, allow multiple winning classifiers per cycle as well as the processing of internal messages, and, with that, achieve a higher degree of parallelism and cognitive plausibility. We think, however, that the HCA can be fully extended towards general CSs. In particular, in artificial intelligence there is a plenty of work on learning by chunking in rule-based systems and production systems (e.g., see (Laird, Rosenbloom & Newell, 1986) and the references therein), and this work is likely to be very stimulating and useful for constructing such an extension.

The work described in this paper shows new perspectives of several issues of current CS research, including the locality/globality dilemma, the cooperator/competitor dilemma, the chaining problem, the mapping problem, and the system-environment interaction. However, further investigations are needed in order to fully understand the merits and limitations of hierarchical chunking in CSs.

Acknowledgements

I want to thank Armin Wirth and Martin Eldracher for many stimulating and fruitful discussions on learning in classifier systems.

References

Booker, L.B. 1982. Intelligent behavior as an adaptation to the task environment. Ph.D. diss., Dept. of Computer and Communication Sciences, Univ. of Michigan.

Booker, L.B., Goldberg, D.E., and Holland, J.H. 1989. Classifier systems and genetic algorithms. *Artificial Intelligence* 40: 235–282.

Chase, W.G., and Simon, H.A. 1973. Perception in chess. *Cognitive Psychology* 4: 55–81.

Chi, M.T.H. 1978. Knowledge structures and memory development. In R.S. Siegler ed. *Children's thinking: What develops?* Hillsdale, NJ: Erlbaum.

Dorigo, M. 1991. New perspectives about default hierarchies formation in learning classifier systems. Technical Report, No. 91-002. Dipartimento di Elettronica, Politecnico di Milano.

Goldberg, D.E. 1989. *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

Grefenstette, J.J. 1988. Credit assignment in rule discovery systems based on genetic algorithms. In *Machine Learning* 3: 225–245.

Holland, J. H. 1975. *Adaptation in natural and artificial systems.* Ann Arbor, MI: Univ. of Michigan Press.

Holland, J.H. 1985. Properties of the bucket brigade algorithm. In Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1–7. Pittsburgh, PA: Erlbaum.

Holland, J.H. 1986. Escaping brittleness: The possibilities of general-purpose learning algorithms to parallel rule-based systems. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell eds. *Machine learning: An artificial intelligence approach*, 593–632. Los Altos, CA: Morgan Kaufmann.

Holland, J.H., and Reitman, J.S. 1978. Cognitive systems based on adaptive algorithms. In D.A. Waterman and F. Hayes-Roth eds. *Pattern-directed inference systems*, 313–329. New York: Academic Press.

Huang, D. 1989. The context-array bucket-brigade algorithm: An enhanced approach to credit-apportionment in classifier systems. In Proceedings of the Third International Conference on Genetic Algorithms, 311–316. Fairfax, VA: Morgan Kaufmann.

Laird, J.E., Rosenbloom, P.S., and Newell, A. 1986. Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning* 1: 11–46.

Miller, G.A. 1956. The magic number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 63: 81–97.

Newell, A., and Rosenbloom, P.S. 1981. Mechanisms of skill acquisition and the law of practice. In J.R. Anderson ed. *Cognitive skills and their application.* Hillsdale, NJ: Erlbaum.

Riolo, R.L. 1987. Bucket brigade performance: I. Long sequences of classifiers. In Proceedings of Second International Conference on Genetic Algorithms, 184–195. Hillsdale, NJ: Erlbaum.

Riolo, R.L. 1988. CFS-C: A package of domain independent subroutines for implementing classifier systems in arbitrary, user-defined environments. Technical Report. Division of Computer Science and Engineering, Univ. of Michigan.

Riolo, R.L. 1989. The emergence of coupled sequences of classifiers. In Proceedings of the Third International Conference on Genetic Algorithms, 256–264. Fairfax, VA: Morgan Kaufmann.

Riolo, R.L. 1990. Lookahead planning and latent learning in classifier systems. In J.-A. Meyer and S. Wilson eds. *From Animals to Animats 1*. Cambridge, MA: The MIT Press.

Robertson, G.G., and Riolo, R.L. 1988. A tale of two classifier systems. *Machine Learning* 3: 139–159.

Rosenbloom, P.S. 1983. The chunking of goal hierarchies: A model of practice and stimulus-response compatibility. Technical Report, No. 83–148. Computer Science Institute, Carnegie–Mellon Univ.

Rosenbloom, P.S., and Newell, A. 1986. The chunking of goal hierarchies: A generalized model of practice. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell eds. *Machine learning: An artificial intellligence approach* (Vol. 2). Los Altos, CA: Morgan Kaufmann Publ.

Sutton, R.S. 1990. Reinforcement learning architectures for animats. In J.-A. Meyer and S. Wilson eds. *From Animals to Animats 1*, 288–296. Cambridge, MA: MIT Press.

Weiß, G. 1991. The action–oriented bucket brigade. Technical Report, FKI-156-91. Institut für Informatik, Technische Univ. München.

Weiß, G. 1992. Learning the goal relevance of actions in classifier systems. In Proceedings of the 10th European Conference on Artificial Intelligence, 430–434. Baffins Lane, Chichester: Wiley.

Weiß, G. 1993a. Action selection and learning in multi-agent environments. In J.-A. Meyer, H.L. Roitblat, and S.W. Wilson eds. *From Animals to Animats 2*, 502–510. Cambridge, MA: MIT Press.

Weiß, G. 1993b. Learning to coordinate actions in multi-agent systems. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, 311–316. San Mateo, CA: Morgan Kaufmann.

Wilson, S.W. 1985. Knowledge growth in an artificial animal. In Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 16–23. Pittsburgh, PA: Erlbaum.

Wilson, S.W. 1987. Hierarchical credit allocation in a classifier system. In L. Davis ed. *Genetic algorithms and simulated annealing*, 104–115. Los Altos, CA: Morgan Kaufmann.

Wilson, S.W., and Goldberg, D.E. 1989. A critical review of classifier systems. In Proceedings of the Third International Conference on Genetic Algorithms, 244–255. Fairfax, VA: Morgan Kaufmann.