An algorithm for finding reliably schedulable plans

Bálint Takács, István Szita and András Lőrincz
Department of Information Systems
Eötvös Loránd University
Budapest, H-1117
E-mail: lorincz@inf.elte.hu

***Abstract*—For interacting agents in time-critical applications, learning whether a subtask can be scheduled reliably is an important issue. The identification of sub-problems of this nature may promote e.g. planning, scheduling and segmenting in Markov decision processes. We define a subtask to be *schedulable* if its execution time has a small variance. We present an algorithm for finding such subtasks.**

I. Motivation

There is an increasing interest in planning in stochastic environments, also called decision-theoretic planning. Since planning is regarded as a mainstay of artificial intelligence, attempts were made to extend classical AI planning methods to handle uncertainty. In parallel with this, operation research has also made important advances recently in solving decision making problems [1]. The earliest work, which tried to unify the two complementary approaches appeared in the seventies [2]. Recently Markov decision processes were proposed as a unifying framework for decision-theoretic planning [3], [4]. The recent success of the closely related reinforcement learning (RL) techniques have encouraged work in planning using Markov decision processes.

For planning one needs goals.[1] Classical AI planning usually determines the goal as a subset of the state space. This definition is extended by Markov decision processes, where an immediate cost (or reward) function is defined and the 'goal' of the decision maker is not to reach a set of states, but to minimize the discounted long-term cumulated cost. In RL, planning is equal to finding 'good' policies, i.e. state-dependent action selection strategies for which the cumulated cost is minimal [5]. In Markov decision processes, we say that a task is *episodic*, when well-defined terminal (goal) states exist and the episode as well as the accumulation of costs stops in finite time — after reaching one of the terminal states. In episodic tasks, planning is often formalized as the task of finding plans for which the probability of achieving the goal states is maximal and/or the expected episode length is minimal.

Plans taken into consideration by a decision making agent may have many evaluation criteria, such as the average cost accumulated during execution, the probability of a successful execution, the expected episode length or any combination of these. We will focus here on a rarely emphasized feature of plans, namely the reliability of the plan execution time. We shall call plans that have a reliable execution time *schedulable*. Schedulability is an important issue in time-sensitive systems. For example, imagine a cooperating multi-agent system, where the agents can interact only if they are close in space (e.g. the agents are the trucks of a transport company). It is impossible to create a global long-term plan if the arrival time of specific agents (trucks) show a great variance. Being accurate may be much more important than having somewhat faster deliveries on the average. Another example is that 'the postman may bring a letter at 8 o'clock'. It might be important to check the mailbox, although the probability of receiving an answer could be low. Missing an appointment may cause long-term effects in the whole system like a small snowball can grow into an avalanche. This is a relevant issue in every system with strongly nonlinear responses, which is the common cause in the engineering practice. In most real-life problems, for long-term planning nearly-deterministic, i.e., schedulable sub-components (sub-tasks) are necessary.

How to measure such kind of reliability? The simplest assumption is that the larger the variance of the duration of a sub-task, the less one can rely on that component when making a plan. In the next section we derive a simple algorithm for calculating the time variance of episodes.

II. Calculating average episode length and variance

A. Assumptions and definitions

Consider an episodic MDP with a finite state space S, a finite action space A and a transition function $P(x, a, y)$, i.e. selecting action a in state x transfers the agent into state y with the probability defined by P. The agent starts from some state x_0, and makes steps according to the nondeterministic policy π, that is, the probability of choosing action a in state x is $\pi(x, a)$, until it reaches a state in the terminal set $Z \subset S$. Let $Z_0 \subset Z$ be the goal set. We may assume that from any terminal state $z \in Z$, the agent is transferred to a hyper-terminal state ω, i.e. $P(z, a, \omega)$=1 for all $z \in Z, a \in A$. This assumption does not modify the time of reaching Z for the first time, but it enables us to simplify our formalism, because every state in Z becomes an absorbing state by this extension, so we do not have to deal with the second-visit third-visit etc. cases.

Denote by $p^\pi(x, y)$ the probability that from state x the agent arrives to state y while following policy π, i.e.

[1]Note that 'planning' in the field of reinforcement learning is sometimes used in the meaning 'off-line optimization of decision making' ([1]). However, here planning is used in its ordinary meaning, i.e., as a synonym for considering different trajectories for scheduling or to achieve a goal.

$$p^\pi(x,y) = \sum_{a\in A} \pi(x,a)P(x,a,y). \quad (1)$$

The visited states in a start-goal episode are noted by $\{x_0,\dots,x_K\}$ assuming the episode takes K steps (note that K is also a random variable). If an $x \to y$ transition takes $\tau_{xy} \in \mathcal{N}_0$ time,[2] then the completion time of an episode is $T = \sum_{i=0}^{K-1} \tau_{x_i x_{i+1}}$ where $T \in \mathcal{N}_0$. Naturally, if every transition takes 1 unit of time then $K \equiv T$.

We would like to find the answer to the following three questions:

1) What is the probability that the agent ends up in a *goal state*, i.e. in Z_0?
2) What is the average time needed for reaching a goal state?
3) What is its variance?

B. The probability of success

An episode may be either successful (the agent ends up in Z_0) or unsuccessful (the agent ends up in $Z\backslash Z_0$). Denote by $s(x)$ the probability that starting from x, the agent will be successful, i.e. $s(x) = \Pr(x_K \in Z_0|x_0 = x)$. Clearly,

$$s(x) = \begin{cases} 1 & \text{if } x \in Z_0, \\ 0 & \text{if } x \in Z\backslash Z_0, \\ \sum_{y\in S} p^\pi(x,y)s(y) & \text{if } x \in S\backslash Z. \end{cases} \quad (2)$$

C. The probability of success in exactly T time

Let $q(T|x)$ denote the probability of reaching a state in Z_0 exactly at time T, assuming the agent started from state x at time 0 (the duration of reaching Z_0 is exactly T). That is, $q(T|x) = \Pr(T, x_K \in Z_0|x_0 = x)$ for every $T \geq 0$. Then,

$$q(0|x) = \begin{cases} 1 & \text{if } x \in Z_0, \\ 0 & \text{if } x \in S\backslash Z_0, \end{cases}$$

and the following simple recursion holds:

$$q(T|x) = \begin{cases} \sum_{y\in S} p^\pi(x,y)q(T-\tau_{xy}|y) & \text{if } x \in S\backslash Z \\ 0 & \text{if } x \in Z \end{cases}$$

for $T \geq 1$ and $q(T|x) = 0$ for every $T < 0$.

For the sake of simplicity, we assume that from any non-terminal state the agent reaches a terminal state in finite time with probability one, i.e.

$$\{\Pr(x_K \notin Z_0|x_0 = x), q(0|x), q(1|x), q(2|x), \dots\}$$

is a probability distribution. It is easy to see that $\sum_{T=0}^{\infty} q(T|x) = s(x)$.

[2]The time spent by a transition could depend also on the applied action, which constrains one to work with function $\tau(x,a,y)$ instead of τ_{xy}. The emerging equations are similar and are omitted here for the sake of simplicity. The only difference is that the simplified notation of Eq. (1) can not be used in the more general case.

D. The average episode length

Using of the above recursion, similar recursion can be derived for the expected number of time steps needed to reach a goal state from x (denoted by $A(x)$):

$$\begin{aligned} A(x) &= \mathbf{E}(T|x_K \in Z_0, x_0 = x) = \\ &= \sum_{T=0}^{\infty} T\cdot \Pr(T|x_K \in Z_0, x_0 = x) = \\ &= \sum_{T=0}^{\infty} T\cdot \frac{\Pr(T, x_K \in Z_0|x_0 = x)}{\Pr(x_K \in Z_0|x_0 = x)} = \\ &= \frac{1}{s(x)} \sum_{T=0}^{\infty} T\cdot q(T|x). \end{aligned}$$

In particular, if $x \in Z$, then $A(x) = 0$. For $x \in S\backslash Z$, $A(x)$ can be expressed as

$$\begin{aligned} A(x) &= \frac{1}{s(x)} \sum_{T=0}^{\infty} T\cdot q(T|x) = \\ &= \frac{1}{s(x)} \sum_{T=0}^{\infty} T \sum_{y\in S} p^\pi(x,y) q(T-\tau_{xy}|y) = \\ &= \frac{1}{s(x)} \sum_{y\in S} p^\pi(x,y)\cdot \\ &\quad \cdot \sum_{T=0}^{\infty} \left[(T-\tau_{xy})\cdot q(T-\tau_{xy}|y) + \tau_{xy} q(T-\tau_{xy}|y)\right] = \\ &= \frac{1}{s(x)} \sum_{y\in S} p^\pi(x,y)\cdot \\ &\quad \cdot \left(\sum_{T=0}^{\infty} T\cdot q(T|y) + \tau_{xy}\sum_{T=0}^{\infty} q(T|y)\right) = \\ &= \frac{1}{s(x)} \sum_{y\in S} p^\pi(x,y)s(y)\left[A(y) + \tau_{xy}\right]. \end{aligned} \quad (3)$$

E. The variance of episode length

The second moment $B(x)$ of the episode length for each $x \in S$ can be derived in a similar manner:

$$B(x) = \mathbf{E}(T^2|x_K \in Z_0, x_0 = x) = \frac{1}{s(x)} \sum_{T=0}^{\infty} T^2 \cdot q(T|x).$$

If x is a terminal state, then $B(x) = 0$. Otherwise,

$$
\begin{aligned}
B(x) &= \frac{1}{s(x)} \sum_{T=0}^{\infty} T^2 \cdot q(T|x) = \\
&= \frac{1}{s(x)} \sum_{T=0}^{\infty} T^2 \sum_{y \in S} p^{\pi}(x,y) q(T-\tau_{xy}|y) = \\
&= \frac{1}{s(x)} \sum_{y \in S} p^{\pi}(x,y) \cdot \\
&\quad \cdot \sum_{T=0}^{\infty} \big[(T-\tau_{xy})^2 \cdot q(T-\tau_{xy}|y) + \\
&\quad +2 \cdot \tau_{xy} \cdot T \cdot q(T-\tau_{xy}|y) - \tau_{xy}^2 q(T-\tau_{xy}|y) \big] = \\
&= \frac{1}{s(x)} \sum_{y \in S} p^{\pi}(x,y) \cdot \\
&\quad \cdot \sum_{T=0}^{\infty} \big[(T-\tau_{xy})^2 \cdot q(T-\tau_{xy}|y) + 2 \cdot \tau_{xy} \cdot \\
&\quad \cdot [(T-\tau_{xy}) \cdot q(T-\tau_{xy}|y) + \tau_{xy} \cdot q(T-\tau_{xy}|y)] - \\
&\quad -\tau_{xy}^2 q(T-\tau_{xy}|y) \big] = \\
&= \frac{1}{s(x)} \sum_{y \in S} p^{\pi}(x,y) \cdot \\
&\quad \cdot \left(\sum_{T=0}^{\infty} T^2 \cdot q(T|y) + 2 \cdot \tau_{xy} \sum_{T=0}^{\infty} T \cdot q(T|y) + \right. \\
&\quad \left. +\tau_{xy}^2 \sum_{T=0}^{\infty} q(T|y) \right) = \\
&= \frac{1}{s(x)} \sum_{y \in S} p^{\pi}(x,y) s(y) \cdot \\
&\quad \cdot \left[B(y) + 2 \cdot \tau_{xy} \cdot A(y) + \tau_{xy}^2 \right] .
\end{aligned} \tag{4}
$$

The quantities s, A and B each satisfy fixed-point equation, so they can be effectively approximated by dynamic programming, which is essentially turning the equations into iterations. Note also that the combined vector $(s; A; B)$ also satisfies a fixed-point equation, which means that they can be computed in a single loop. So the algorithm summarized below gives the answers to the questions at the beginning of this section:

```
Initialize $s$, $A$, $B$ with 0 for every $x$
REPEAT
    Update $s(x)$ using Equation 2
    Update $A(x)$ using Equation 3
    Update $B(x)$ using Equation 4,
UNTIL convergence
```

This iteration, like all fixed-point iterations, converges to a unique solution if and only if the update equations form a contraction mapping. In our case, this will surely be true if either (a) all episodes end in at most $K < \infty$ steps or (b) all episodes end at each time step at least with probability p for some fixed $p > 0$.

Summarizing our results, the above method gives (i) the probability of success, (ii) the time an episode takes on average, and (iii) the variance of the episode time from all possible starting states for a prescribed terminal state set.

III. Demonstration

We generated a simple toy problem to demonstrate the algorithm. Imagine a river, which flows rapidly from west to east. There is a port on the left bank of the river. There is also a ship on the river, initially positioned somewhere on the river. The ship has to reach the port, otherwise it enters a terrible and huge waterfall, situated at the east end of the river. Painfully, the water has very strong vortices and currents, therefore it is almost impossible to control the movement of the ship. Our model may assume in advance that in the next time step the currents will almost surely drive the ship towards the waterfall, but we don't know exactly in which direction. A small chance exists that the vortices drive the ship backwards, too. Of course, at a given instant, and if not at the east end of the river, the ship can't be very far from its previous position. Details of this toy problem are depicted in Fig. 1. Almost every path in this state space (except for the ones near the banks) has a low probability to happen. Direct planning of particular trajectories is almost meaningless. But, as it can be seen on Figs. 3 and 4, scheduling in the sense described in Section II, is still meaningful. Here the episode length can be arbitrarily long, but with probability converging to zero: $\lim_{k\to\infty} \Pr(K = k) \to 0$. An average trajectory takes about as much steps as long the river (50 steps), so we did 100 iterations to be sure about reaching equilibrium. Figures 2, 3 and 4 plot the resulting quantities $s(x)$, $A(x)$ and the square root of $B(x)$ respectively. The figures show that there is a very low chance to reach the port when the ship starts from the right of it, as expected. However, because we calculate the time needed for *successful* episodes, the average episode times needed are almost the same as when the ship starts in the leftmost positions, but the reliability of these times – measured by the variance or the standard deviation (Fig. 4) – are higher in the first case. If the harbor master knows the ship's starting position, he knows when to look for the ship on the horizon according to Fig. 3, and he can decide from Fig. 4 how long should he wait until he can almost surely guess that the ship is lost. Being schedulable is a property independent from being easy, as it is illustrated by this simple problem.

IV. Discussion

Examining the probabilistic properties of the cumulated cost beyond its average value is not new in the literature. The attempts concentrated mainly on penalizing the variance of the accumulated cost [6], [7], [8], [9], [10], [11]. Algorithms for calculating the variance directly have also been published. Kemeny and Snell ([12]) developed a formula for the second moment of first-passage times in Markov chains. Formulae are derived in [13] for the second moment of the accumulated total reward of Markov chains with rewards. In [14] Sobel presented a general method for calculating the arbitrary moments of the cumulated discounted cost. This method is similar to ours. The algorithm introduced here emphasizes the possibility of

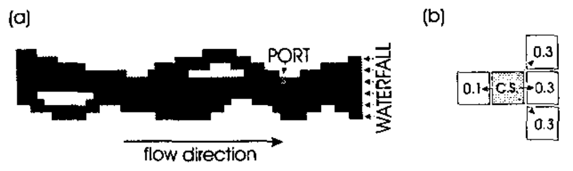

Fig. 1. **Description of the toy problem.**
Subfigure (a) shows the 'river' flowing from the left to the right. The river is constructed on a 50 by 10 square grid. The 'port' is marked with a red square. The river empties into a waterfall on the right side. If the ship reaches any states of the last column, the trial has an unsuccessful end. Subfigure (b) shows the predefined state transitions. If not obstructed by the river's bank, there is an equal probability (0.3) of transferring from the current state (noted by C.S.) into one of the three neighboring states in the next right column. The diagonal steps take 2 time units, while the forward step takes 1 time unit. Simultaneously, there is also a low probability (0.1) of moving backwards to the left column, which move takes 5 time units. The banks of the river and the islands limit the number of available transitions. In this cases, the probability of the unavailable transition(s) is equally distributed between the remaining ones.

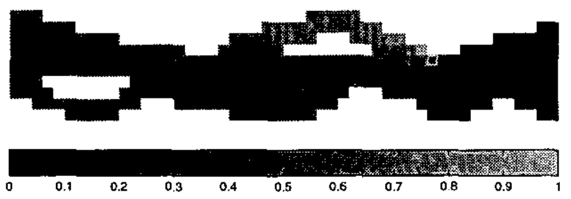

Fig. 2. **Probability of a successful episode.**
The figure plots the probability of successful arrivals from any given state ($s(x)$) (computed by the recursive formula (2). The graded blue color of a point of the river provides the probability of arriving to the port. Color coding is provided at the bottom of the figure. The dark colors on the right hand side of the port indicate that chances of successful arrivals from that part of the river are low.

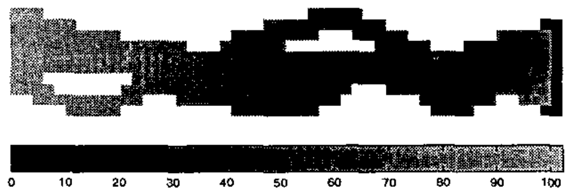

Fig. 3. **Average episode lengths.**
Average durations ($A(x)$) of successful episodes are indicated by graded blue color for every state (x).

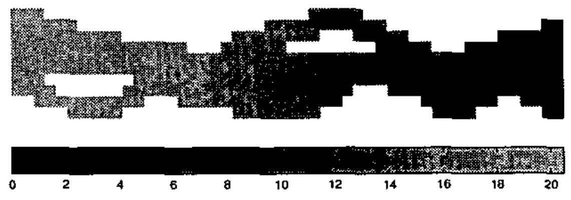

Fig. 4. **Standard deviation of episode length.**
The standard deviation of the duration of successful episodes ($\sqrt{B(x)}$) are indicated by graded blue color for every state (x).

using the average duration of the episode, or higher moments of the duration of an episode as costs to transfer the problem of temporal planning to the domain of reinforcement learning, as it will be discussed shortly. By introducing the probability of success, a novel property of the formulae is that they can be used to compute the appropriate properties for successful or unsuccessful episodes.

A. Reinforcement Learning Methods

The algorithm does not employ the immediate costs (rewards) defined by the original MDP. Nevertheless, the algorithm is closely related to such cost-based approaches: If the transition time is viewed as cost, then calculating the average time of the execution of the episode can be seen as calculating the average sum of the (time-)costs in the non-discounted case. In this sense, quantities τ_{xy} determine the corresponding cost function. Indeed, we can recognize Eq. 4 as the well-known Bellman-equality [15] for the case of non-discounted costs.

Therefore, similarly to the standard reinforcement learning problem, we do not need to solve the DP equations directly, but we may update the s, A and B values in different ways explored by theoretical works in the field of RL (for a historical review, see, e.g., [16] and references therein). Asynchronous and Monte Carlo methods, as well as sampling methods analogous to Q-learning or SARSA can be of use here. The advantage of RL methods as compared to DP methods can be seen in problems that are too large for direct computations. Moreover, RL methods can be used by direct interaction with the system (called on-line learning). This solution is the only option when no model is available. On the other hand, if a DP model is available and sampling methods with the real system are cumbersome, then DP has the advantage of being an off-line method, not limited by constraints on interactions with a real system.

Another advantage of formulating temporal planning within the framework of RL is that costs on time (i.e., planning) and real costs of the episode can be unified into a single cost function. Moreover, the multi-criteria paradigm [17], [18] can be applied to consider multiple cost functions in parallel. In particular, cost functions with mean-variance tradeoffs [6], [8], [9], [10], [11] may penalize solutions with high variance. One has the option of solving the MDP for the original cost function and also for the time as the 'cost' and then combine the two aspects.

B. Planning as a Segmentation Tool

As it has been mentioned before, in our terminology planning is an off-line method, which serves to minimize the frequency of on-line decision making. Planning can be the tool to segment decision making problems into subproblems.

In practice, the major problem of decision-theoretic planning is the intractably large space state. Handling complex, real-life situations is impossible without grouping states into larger sets (possibly structured in hierarchical fashion), thus allowing for crude discretization, partial observation and assigning *subgoals* for complex tasks. In most real-life problems, the environment is only partially observable, therefore we need to solve partially observable Markov decision processes instead of classical MDPs. Partial observability may come from the limited perceptual capabilities of the agent: the agent may possess insufficient resources to observe all variables of its environment, the agent may be localized not having information about remote objects in space, the agent may be slow to observe all temporal changes happening in the environment. If the optimization problem is partially observable, then solving this problem is intractable in general [19]. Every available tool should be used to ease the learning problem. The most promising methods are those, which are only partly limited by the number of interactions with the real system, that is, the on-line methods extended by off-line methods. Off-line computations are preferred when they are relatively inexpensive. By finding schedulable plans we may be able to segment the current problem into subproblems. One can use the variance calculated by Eq. (4) to measure which states are reliable subgoals. Then, variance of the duration of the episode serves as auxiliary information for distinguishing states which can be later assigned as subgoals. The advantages of subgoals and concept formation using subgoals have been in the focus of recent research interest (see, e.g., [20], [21], [22], [23], [24] and references therein).

These concepts may gain applications in a variety of areas: basically everywhere, where RL and DP need to be extended by scheduling. A few representative examples are as follows: job-shop scheduling [25], elevator optimization [26], robotic soccer [27] or context focused Internet crawling making use of RL [28].

ACKNOWLEDGEMENTS

This work was supported by the Hungarian National Science Foundation (Grant OTKA 32487) and by EOARD (Contract No. FA8655-03-1-3036). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory.

REFERENCES

[1] R. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *SIGART Bulletin*, vol. 2, pp. 160–163, ACM Press 1991.

[2] J. A. Feldman and R. F. Sproull, "Decision theory and artificial intelligence II: The hungry monkey," *Cognitive Science*, vol. 1, pp. 158–192, 1977.

[3] C. Boutilier, T. Dean, and S. Hanks, "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, pp. 1–99, 1999.

[4] T. Dean, L. Kaelbling, J. Kirman, and A. Nicholson, "Planning under time constraints in stochastic domains," *Artificial Intelligence*, vol. 76, no. 1-2, pp. 35–74, 1995.

[5] M. L. Littman, J. Goldsmith, and M. Mundhenk, "The computational complexity of probabilistic planning," *Journal of Artificial Intelligence Research*, vol. 9, pp. 1–38, 1998.

[6] J. A. Filar, L. C. M. Kallenberg, and H. M. Lee, "Variance-penalised markov decision processes," *Math Oper. Res.*, vol. 14, pp. 147–161, 1989.

[7] J. Greffenstette, C. Ramsey, and A. Schultz, "Learning sequential decision rules using simulation models and competition," *Machine Learning*, vol. 5, pp. 355–381, 1990.

[8] Y. Huang and L. C. M. Kallenberg, "On finding optimal policies for markov decision chains: a unifying framework for mean-variance-tradeoffs," *Math Oper. Res.*, vol. 19, pp. 434–448, 1994.

[9] D. J. White, "Computational approaches to variance penalised markov decision processes," *OR Spektrum*, vol. 14, pp. 79–83, 1992.

[10] ——, "A mathematical programming approach to a problem in variance penalised markov decision processes," *OR Spektrum*, vol. 15, pp. 225–230, 1994.

[11] E. J. Collins, "Finite-horizon variance penalised markov decision processes," *OR Spektrum*, vol. 19, pp. 35–39, 1997.

[12] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Van Nostrand, New York, 1960.

[13] L. K. Platzman, *Mimeographed Lecture Notes for IOE 315*. Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, 1978.

[14] M. J. Sobel, "The variance of discounted Markov decision processes," *Journal of Applied Probability*, vol. 19, pp. 794–802, 1982.

[15] R. Bellman, *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[16] R. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.

[17] L. G. Mitten, "Composition principles for synthesis of optimum multi-stage processes," *Operations Research*, vol. 12, pp. 610–619, 1964.

[18] Z. Gábor, Zs. Kalmár, and Cs. Szepesvári, "Multi-criteria reinforcement learning," ser. Proceedings of the Fifteenth International Conference on Machine Learning, 1998.

[19] M. L. Littman, "Algorithms for sequential decision making," Ph.D. dissertation, Department of Computer Science, Brown University, February 1996.

[20] M. Ring, "Incremental development of complex behaviors through automatic construction of sensory-motor hierarchies," ser. Proceedings of the Eighth International Conference on Machine Learning. San Mateo, CA: Morgan Kaufmann, 1991.

[21] J. Schmidhuber, *Artificial Neural Networks, Eds.: T. Kohonen, K. Mkisara, O. Simula, and J. Kangas*. Elsevier Science Publishers, 1991, ch. Learning to generate sub-goals for action sequences, pp. 967–972.

[22] M. Wiering and J. Schmidhuber, "HQ-learning: Discovering markovian subgoals for non-markovian reinforcement learning," Technical Report IDSIA-95-96, 1996. [Online]. Available: http://www.idsia.ch/~juergen/subgoals.html

[23] R. Sun and C. Sessions, "Self-segmentation of sequences: automatic formation of hierarchies of sequential behaviors," *IEEE Transactions on Systems, Man, and Cybernetics: Part B Cybernetics*, vol. 30, pp. 403–418, 2000.

[24] A. McGovern, "Autonomous discovery of temporal abstractions from interaction with an environment," PhD Thesis, University of Massachusetts, Amherst, MA, 2002.

[25] W. Zhang and T. G. Dietterich, "A reinforcement learning approach to job-shop scheduling," in *Proceedings of the International Joint Conference on Artificial Intellience*, 1995.

[26] R. Crites and A. Barto, "Improving elevator performance using reinforcement learning," ser. Advances in Neural Information Processing Systems, M. C. M. D. S. Touretzky and M. E. Hasselmo, Eds., vol. 8. San Mateo, CA: Morgan and Kaufmann, 1996, pp. 1017–1023.

[27] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The robot world cup initiative," in *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, W. L. Johnson and B. Hayes-Roth, Eds. New York: ACM Press, 5–8, 1997, pp. 340–347.

[28] I. Kókai and A. Lőrincz, "Fast adapting value estimation based hybrid architecture for searching the world-wide web," *Applied Soft Computing*, vol. 2, pp. 11–23, 2002.