The Cyber-Physical System Approach Towards Artificial General Intelligence: The Problem of Verification

Zoltán Tősér and András Lőrincz(✉)

Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
lorincz@inf.elte.hu

Abstract. Cyber-Physical Systems have many components including physical ones with heavy demands on workflow management; a real-time problem. Furthermore, the complexity of the system involves some degree of stochasticity, due to interactions with the environment. We argue that the factored version of the event-learning framework (ELF) being able to exploit robust controllers (RCs) can meet the requirements. We discuss the factored ELF (fELF) as the interplay between episodic and procedural memories, two key components of AGI. Our illustration concerns a fELF with RCs and is a mockup of an explosive device removal task. We argue that (i) the fELF limits the exponent of the state space and provides solutions in polynomial time, (ii) RCs decrease the number of variables and thus decrease the said exponent further, while the solution stays ε-optimal, (iii) solutions can be checked/verified by the execution being linear in the number of states visited, and (iv) communication can be restricted to instructions between subcomponents of an AGI system.

1 Introduction

Cyber-physical systems (CPSs) are in the forefront of algorithmic, software, and hardware developments. They are goal oriented. In the typical setting they are distributed, have physical components, and can include e.g., sensory, computational and robotic units. Given their complexity, testing may become the bottleneck, especially for safety- and time-critical applications. In case of any unexpected event or anomaly in the behavior, fast workflow management may become a necessity and might involve changes of the plan and thus communication of new subtasks, new roles, and new methods of communication, among other things. We say that a simple instruction or a more complex subtask make sense in a given context, if the responsible actors can execute them given the information provided. Successful completion of an instruction or a subtask verifies a portion of a larger plan. The larger the plan and the more complex the system, the more serious anomalies may occur. In turn, stochastic formulation is required.

We shall put forth the factored event-learning framework (fELF), a special form of reinforcement learning (RL), that has polynomial time learning characteristics and the maximal number of concurrent and dependent factors limits

J. Bieger (Ed.): AGI 2015, LNAI 9205, pp. 373–383, 2015.
DOI: 10.1007/978-3-319-21365-1_38

the exponent of the state space (Sect. 2). We illustrate fELF via a toy mockup explosive device (ED) removal task (Sect. 3). Up to the number of variables, the solution is '*hard to find*'. In the discussion section (Sect. 4) we will argue that this problem is '*easy to verify*' by following the steps in time as prescribed by the *solution*. Such solutions are worth to communicate. We conjecture that IQ tests are of similar nature. Conclusions will be drawn in Sect. 5.

2 Theoretical Background

We propose the MDP framework for CPSs. We utilize the generalized MDP (gMDP) formulation. Its ε-gMDP extension concerns ε-precise quantities and can exploit robust controllers if they meet the ε-precise condition. We review the event-learning framework (ELF) [8,16] that breaks tasks into subtasks, can admit ε-precise robust controllers and can hide some of the variables. An ELF extended with robust controllers is an ε-gMDP. The factored formulation of MDP gives rise to polynomial time optimization. Taken together, a factored generalized ELF with a robust controller is an ε-gMDP with polynomial time optimization. Execution requires the communication of instructions to the subcomponents making verification linear in time for deterministic systems.

2.1 Markov Decision Processes

A (finite) MDP [10] is defined by the tuple $\langle X, A, R, P\rangle$. X and A denote the finite set of states and actions, respectively. $P : X \times A \times X \to [0, 1]$ is the transition function, the probability of arriving at state y after executing action a in state x. $R : X \times A \times X \to \mathbb{R}$ is the reward function: $R(x, a, y)$ is the immediate reward for transition (x, a, y).

Decision making aims at finding the optimal behavior subject to some optimality criterion, e.g., to infinite-horizon expected discounted total reward, when we want to find a policy $\pi : X \times A \to [0, 1]$ that maximizes the expected value of $\sum_{t=0}^{\infty} \gamma^t r_t$, where r_t is the immediate reward in time step t and $0 \leq \gamma < 1$ is the discount factor.

A standard way to find an optimal policy is to estimate the optimal value function $V^* : X \to \mathbb{R}$, which gives the value (the expected cumulated discounted reward with the given starting state) of each state. From this, the optimal policy is the 'greedy' policy with respect to the optimal value function, i.e., the following Bellman equation:

$$V^*(x) = \max_a \sum_y P(x, a, y) \left(R(x, a, y) + \gamma V^*(y)\right), \quad \text{for all } x \in X. \tag{1}$$

2.2 Generalized MDP (gMDP) and ε-gMDPs

Operations $\sum_y P(x, a, y) \ldots$ and $\max_a \ldots$ can be extended, e.g., with risk considerations. Joint formalism for the different Bellman equiations has been constructed in [13]: a generalized MDP is defined by the tuple $\langle X, A, R, \bigoplus, \bigotimes\rangle$,

where X, A, R are defined as above; $\bigoplus : (X \times A \times X \to \mathbb{R}) \to (X \times A \to \mathbb{R})$ is an 'expected value-type' operator and $\bigotimes : (X \times A \to \mathbb{R}) \to (X \to \mathbb{R})$ is a 'maximization-type' operator. We want to find the value function V^*, where

$$V^*(x) = \bigotimes\bigoplus(R(x, a, y) + \gamma V^*(y)), \quad \text{for all } x \in X.$$

or in short form $V^* = \bigotimes\bigoplus(R + \gamma V^*)$. The optimal value function can be interpreted as the total reward received by an agent behaving optimally in a non-deterministic environment. The operator $\bigoplus$ describes the effect of the environment. The operator $\bigotimes$ describes the action-selection of an optimal agent. When $0 \leq \gamma < 1$, and both $\bigoplus$ and $\bigotimes$ are non-expansions, the optimal solution V^* of the equations exists and it is unique.

Generalized ε-MDP (ε-gMDP) assumes a prescribed $\varepsilon > 0$ and is defined by the tuple $\langle X, A, R, \{\bigoplus_t\}, \{\bigotimes_t\}\rangle$, with $\bigoplus_t : (X \times A \times X \to \mathbb{R}) \to (X \times A \to \mathbb{R})$ and $\bigotimes_t : (X \times A \to \mathbb{R}) \to (X \to \mathbb{R})$, $t = 1, 2, 3, \ldots$, if there exists a generalized MDP $\langle X, A, R, \bigoplus, \bigotimes\rangle$ such that $\limsup_{t\to\infty} \|\bigotimes_t \bigoplus_t - \bigotimes\bigoplus\| \leq \varepsilon$. ε-MDPs have been first introduced in [7].

2.3 The Event-Learning Framework (ELF)

Event learning turns the MDP into a hierarchical problem via the *event-value function* $E : X \times X \to \mathbb{R}$ [16]. Pairs of states (x, y) and (x, y^d) are called *events* and *desired events*, respectively: for a given initial state x, y^d denotes the desired next state. The formalism remains the same, but any event can be seen as an *MDP subtask*: the $e_d = (x, y^d)$ state sequence can be a *subtask* to be optimized. $E(x, y^d)$ is the value of trying to get from actual state x to next state y^d. Note that state y reached could differ from desired state y^d.

2.4 Robust Controller

Assume that a state space X and a velocity field $v^d : X \to \dot{X}$ are given. At time t, the system is in state x_t with velocity v_t. We are looking for a control action that modifies the actual velocity to $v^d(x_t)$ with maximum probability:

$$u_t(x_t, v_t^d) = \Phi(x_t, v_t^d),$$

$\Phi(x_t, v_t^d)$ is called the inverse dynamics and it can be approximated. Under certain conditions, one can bound the tracking error to the desired level (see [16] and the references therein).

If time is discrete, like here, then prescribing the desired velocity v^d is equivalent to prescribing the desired successor state y^d. The controller can be directly inserted into an ELF by setting $\pi_t^A(x_t, y_t^d, a) = 1$ if $a = u_t(x_t, y_t^d)$ and 0 otherwise (Fig. 1).

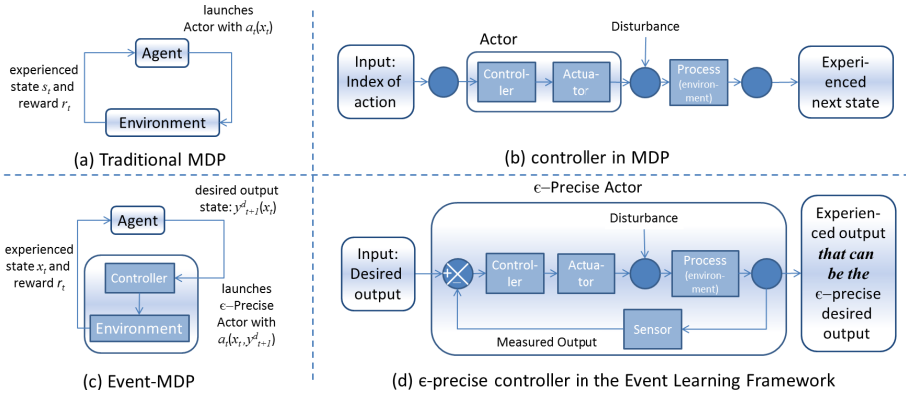

Fig. 1. MDP models. (a): MDP, (b): One step. Input: index of the action, output: experienced next state, (c): ELF, (d): One step. Input: desired output and the output can be the ε-precise version of the desired output.

2.5 Event-Learning with Robust Controller Belongs to the ε-gMDP Family

In the generalized ε-MDP, X denotes the set of states and the action corresponds to selecting a new desired state; the set of actions A is also equal to X. Reward function R is $R(x, y^d, y)$ and it gives the reward for arriving at y from x, when the desired state was y^d. Now, $(\bigotimes_t E)(x) = \max_{y^d} E(x, y^d)$, independently of t, and $(\bigoplus_t E)(x, y^d) = \sum_y p_t(y|x, y^d)E(x, y^d, y)$, where $p_t(y|x, y^d) = \sum_u \pi^A_t(x, y^d, u)P(x, u, y)$. Finally, we define the operators $\bigoplus$ and $\bigotimes$ as $(\bigotimes E)(x) = \max_{y^d} E(x, y^d)$ and $(\bigoplus E)(x, y^d) = \sum_y \sum_u \pi^A(x, y^d, u)P(x, u, y)E(x, y^d, y)$. In turn, if robust controllers are introduced into an ELF, then we still have an ε-gMDP problem with errors that can be bounded.

2.6 Factored Markov Decision Processes (fMDPs)

In CPS, naïve tabular representation of the transition probabilities requires a state space exponential in the number of variables. However, ongoing processes typically exclude other ones and a much smaller number of variables may be sufficient at any given time instant. Let $\mathbf{X}$ be the Cartesian product of m smaller state spaces (corresponding to individual variables), i.e., $\mathbf{X} = X_1 \times X_2 \times \ldots \times X_m$. Each X_i has size $|X_i| = n_i$ and the size of the state space is $N = |\mathbf{X}| = \prod_{i=1}^m n_i$.

In this case, the next-step value of a state variable depends only on a few other variables, so the full transition probability can be obtained as the product of several simpler factors. Formally, for any subset of variable indices $Z \subseteq \{1, 2, \ldots, m\}$, $\mathbf{X}[Z]$ denotes $\underset{i \in Z}{\times} X_i$ and for any $\mathbf{x} \in \mathbf{X}$, $\mathbf{x}[Z]$ denotes the value of the variables with indices in Z. Below, we shall use the shorthand $\mathbf{x}$ for the sake of simplicity. FMDPs were first introduced in [3].

2.7 Polynomial Time Learning

An fMDP with a factored optimistic initialization model (fOIM) – defined below – has a polynomial per-step computational complexity. FOIM gets ε-close to the value function of factored value iteration (which could be suboptimal) in polynomial time [15]:

Theorem 1 (fOIM). *Suppose that an agent is following factored value iteration in an unknown fMDP, where all reward components fall into the interval* $[0, R_{\max}]$*, there are* m *state factors, and all probability- and reward-factors depend on at most* m_f *factors. Let* $E^{\times}(\mathbf{x}_t, \mathbf{y}_t^d)$ *denote the value function of the approximate value iteration exploiting function approximations. Let* $N_f = n^{m_f}$ *and let* $\varepsilon > 0$ *and* $\delta > 0$*. If we set*

$$R_E = c \cdot \frac{mR_{\max}^2}{(1-\gamma)^4 \varepsilon} \left[\log \frac{mN_f|A|}{(1-\gamma)\varepsilon\delta} \right] ,$$

as the initial values of the MDP, then the number of time steps when the agent makes non-near-optimal moves, i.e., when $E^{fOIM}(\mathbf{x}_t, \mathbf{y}_t^d) < E^{\times}(\mathbf{x}_t, \mathbf{y}_t^d) - \varepsilon$*, is bounded by*

$$O\left(\frac{R_{\max}^2 m^4 N_f |A|}{\varepsilon^4 (1-\gamma)^4} \log^3 \frac{1}{\delta} \log^2 \frac{mN_f|A|}{\varepsilon} \right)$$

with probability at least $1 - \delta$*.*

3 Illustrative Experiment and the CPS Connection

For the sake of a fELF illustration we show an experiment with a WheelPhone (WP) and with a Lego NXT, both equipped with Android phones, image processing, QR code reading (not detailed here), and work sharing on a mockup explosive device (ED) removal task. This illustration gives us the opportunity to explain the concept of events, event hierarchy, desired states, episodes, robust controllers and procedures, cost and risk sensitive decision making, meta-level communication and finally, *the problem of verification.*

The illustration is by no means at the level of true cyber-physical systems, although it is a high-risk analogue of a smart factory shop-floor task [11] and has the relevant issues, such as work sharing, path planning, and execution time. The goal of the robots is to find explosive devices and transport them to a given safe location. The terrain contains several obstacles, which may be pushed aside to give way to the ED-carrying robot – but this takes time. Robots used the fELF method for decision making.

The two robots have different capabilities, their control precisions also differ and they share the work. One robot has chances to remove the ED, the other can clear the terrain. We used different number of obstacles and starting points and estimated the distributions of the execution times of the subtasks and their success rates. Each subtask is a desired event given by the actual state and the desired state of the event may become the experienced state later. Desired states include: 'ED found', 'obstacles found', 'path planned', 'first obstacle probably

cleared', 'ED collected', 'terrain cleared', 'track is left', 'ED removed', among a few others. Some tasks are concurrent. The low-complexity RL task in [14] is similar and thus direct policy optimization is also possible, in case if the Markov property is questionable. FMDP description is like in [6]: transitions are limited to the possible ones.

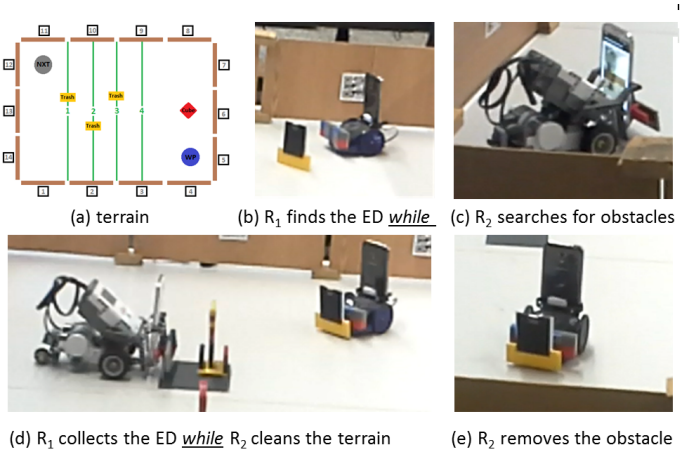

Fig. 2. Experimental arrangement with subtasks. Some of them, including subfigure (d), can be concurrent.

Explosion time has a distribution. The fELF makes decisions at discrete time steps according to the time elapsed, the subtasks executed, the ongoing subtasks, and the time-discretized distributions.

3.1 Results

According to the results (Fig. 3), there are three typical groups in the time variable: execution time is shorter than 2 min, it is longer than 2 min 20 sec and it is between these two values. We used these values for the discretization of the execution time.

The size of the state space in the fMDP depends on the number of factored variables at decision points. This number can be decreased if controllers are precise. For example, the NXT robot is sufficiently precise and direction uncertainties are neglected. NXT can clear away obstacles with certain probabilities, but it remains uncertain if it succeeded to move an obstacle out of the way of the WP robot or not. The motion of the WP robot is straight, but its direction is somewhat imprecise. We left it like this and that made uncertain the success of each obstacle clear-away subtask. Uncertainties measured experimentally and the computed direction uncertainties are used in decision making. The number of obstacles is randomly chosen from 2, 3 and 4 and are placed quasi-randomly over the terrain.

Nr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Time (min:sec)	2:16	2:19	2:25	2:14	2:21	2:33	2:28	1:57	2:15	2:14	1:49	2:19	2:18	2:06	2:12
Outcome	S	S	S	S	S	S	S	F	F	S	F	S	S	S	S
Trash location	1,2,4	1,3,4	1,2,3	1,2	1,2,3	1,3,4	1,2,3	1,2,4	1,3,4	1,2,4	2,3,4	2,3,4	1,3,4	1,4	2,3
NXT start position	12	11	14	12	11	14	12	14	14	1	12	12	11	12	1
WheelPhone start position	5	6	8	4	6	4	7	4	4	4	4	4	7	6	8

Fig. 3. Examples for estimating distributions. Green S: success. Red F: failure. For start positions, see Fig. 2(a). Failures in order: obstacle 1 is not cleared away, NXT-WP crashed, obstacles 3 and 4 are not cleared away.

3.2 Outlook to General Cyber-Physical Systems

Components of a real CPS task are similar to a large extent. Tools, computers, robots — that take part in the task hierarchy — all have capabilities that can be characterized by the complexity of the subtasks they can execute, the belonging success rates, and execution time distribution, for example. Subtasks may be sequential or concurrent according to causal relationships and urgency. Spatio-temporal dependencies of the processes in a complex CPS constrain possible state–desired state pairs of fMDP events. Depending on the type of the ask, e.g., if it is a smart factory, or an emergency situation [9], stochastic environmental disturbances may occur with different probabilities and they may require frequent real-time workflow management. Decision making about the changes of the workflow may not take considerable time even for complex systems and the lowering of the number of variables is highly desired due to the exponential dependence of the state space on those. This is a crucial problem and robust controllers can help in saving time, since such controllers support module construction that may span longer time intervals. For example, the controller of the NXT is more precise than that of the WP and the number of states that may occur and the required frequency of decision making is smaller for the NXT robot than for the WP one. Note that control precision could be increased for the WP robot using its high quality camera. The image processing, however, may increase energy consumption, the need for recharging, and thus the execution time. Plans and workflow management depend on the actual ED and the related risk and cost considerations.

4 Discussion: The Problem of Verification

In the ED removal problem we used higher order concepts (factors) for decision making. Such concepts include ‘explosive’, ‘device’, ‘time’, and alike, instead of raw visual, acoustic, and motor information. Furthermore, we could neglect some of these factors in the description of the situation if those factors were not relevant at that time of decision making. Such simplifications suit factored RL. The problem of forming higher order concepts — that fits the task to be solved and decreases the state space of decision making — falls outside of our considerations.

Problem solving is combinatorial in terms of the selection of the relevant factors, the order of actions to be executed, and the selection of the agent that should execute the action. If a decision is made then it should be communicated to the partners and they must *make sense* of the messages by verifying that the attempt towards the execution of the sub-task is feasible. This procedure of making sense is typical: intelligence proves the solution by means of verification. In general, intelligent verification is a pro-active mental step that exploits an approximate an sufficiently detailed model of the world. Evolution also verifies, but in a different way: evolution finds solutions by their success rates and without any mental model. We take a closer look to the issue of verification below. We note that model based verification is not part of our illustration, but the mockup itself or its computer model can serve as tools for such verification.

4.1 Verification in the Context of Intelligence

There are at least three types of knowledge transfer:

Supervised training. concerns the agreement about concepts (or categories) and can serve meta-level communication after the training phase.

Observations. are important pieces for decision making. However, the world is typically partially observed and distributed observation by many agents can help in solving the problems in due course, e.g., in the case of danger. This knowledge transfer happens at the meta level.

Solutions to problems. include concept forming, procedures, tricks, quizzes, mathematical proofs that exploit the formed concepts, among other things. Many of these procedures (problems) are hard to find (solve), but the verification of the solution can be easy.

Out of the ten broad abilities underpinning the g factor of intelligence [4], only fluid intelligence is connected to the third item, i.e., to concept forming, solving problems, and reasoning abilities. The other nine features of intelligence include reading and writing abilities, quantitative reasoning abilities, speed of decision making and alike. They are of high importance, but we believe that — from point of view of AGI and cyber-physical systems — they are either *solved* or can be solved by available technologies since efficient algorithms can reach superhuman performance if sufficiently large training samples are made available to them [12].

Fluid intelligence seems to differ: it shows up in two steps. One step is concept formation and the other one is solving the problem by means of those new concepts. These two processes are interlinked. The solution can be checked by means of verification using the formed concepts. One may say that if concept formation and problem solving are the core problems of general intelligence then model based verification is the tool for the appreciation of the solution.

4.2 'Verification' is the Goal of Intelligent Communication

There are four categories according to the complexity of solving problems and the complexity of the verification of the solution since both can be 'hard', or

'easy'. Tasks can be hard or easy if they scale exponentially or polynomially with the number of variables, respectively. Out of the four cases, problems belonging to the hard to solve, but easy to verify category are particularly worth to communicate. Such solutions can provide large savings in time and efforts for teammates.

4.3 Interplay Between Procedural and Episodic Memories

Our example has both procedural and episodic components. Any event is an episode and it can be saved in episodic memory for data mining, anomaly detection, model construction, and for learning to predict and control the event. The method of dealing with an ongoing event is the procedure. It is made of actions and sub-events. The 'ED removal story' is an 'ED removal event' brought off by the 'ED removal procedure'. This event may be concurrent with other events and it is probably embedded into a larger one. The event, as described here is independent from the other ongoing concurrent events, which in principle, could disturb it. However, such disturbance is also an event and it is limited in space and time. New concepts, new sensors and additional control tools can be introduced to overcome disturbances of the events provided that the details of the event are knowable, time is available and if the related costs and savings justify the effort.

From the point of view of a larger system, 'ED removal' could be one of its capabilities. Capabilities, i.e., the number of different events that can be invoked by the agent, correspond to desired states in a fELF and they make the variables of decision making. The number of events that can be invoked in a given state enters exponent of state space. The size of the state space can be decreased by learning and optimizing new capabilities made of smaller ones. The number of variables can be decreased by introducing robust controllers. For example, the measurement of the weight of the load can be neglected by adding a robust controller to increase the range of the capability, see. e.g. the example presented in [16]. Communication towards the decision making unit can be limited to the experienced state after execution of a sub-task and to an instruction towards the unit that has the capability to execute the next step. Such instruction contains the desired state and possibly (some of) the steps towards the desired state, i.e., (part of) the 'solution' .

In turn, a fELF with robust controllers efficiently decreases both the number of variables and the data to be communicated. From the point of view of verification, deterministic solutions are easy to verify if a model of the environment is available. For stochastic problems, stochasticity indicates limited knowledge about a knowable universe and may call for further exploration and learning. If more knowledge cannot be acquired in due course or if the collection of such information is costly, then solutions and verifications may require high costs since risks can be overestimated. Model based experimental methods of risk estimation are in the focus of ongoing research [1].

5 Conclusions

We have used an illustrative CPS mockup experiment in the factored event learning framework (fELF). The problem involved recognition, planning, decision making, work sharing, and risk estimation. We included distributions of execution times and success rates either via computational estimations or by measuring those experimentally.

We have argued that a fELF with a robust controller decreases combinatorial explosion. From the point of view of deterministic CPS problems, verification is polynomial *in the number of states* [2]. If we can afford non-tight bounds and additional resources, then experimental verification can be fast, if a model of the environment is available [1].

It has been noted that the problem of verification is alleviated by subtask construction provided that the subtasks can be executed with high fidelity. Robust controllers suit such demands and can save task execution even in the case of environmental disturbances. Any subtask can be viewed as a fELF problem and as such, it can be the subject of optimization. In the same vein, optimized fELF solutions can be embedded into larger tasks. In turn, fELF makes a partially ordered hierarchical RL in a natural fashion.

We note that time critical cyber-physical systems require easy to verify solutions. Such solutions are of high importance for interacting intelligences, since they offer combinatorial gains for teammates. Furthermore, communication can be limited to meta-level instructions about the states to be reached and meta-level information about the states that have been reached upon the execution of the instructions. CPS verification assumes approximately non-interacting sub-events that can run concurrently or may follow each other.

We conclude that CPS tasks concern fluid intelligence and — for large distributed systems — model based real-time verification is required and the time of verification is critical. Finding and learning potentially concurrent, but barely interacting, i.e., *independently and robustly executable* sub-tasks derived from the task space itself offer both exponential gains in the state space and flexibility in multi-tasking. Evolution demonstrates the feasibility of such constructs [5] and engineered solutions may follow similar routes. However, from the point of view of artificial general intelligence this is an unsolved problem. This problem is closely related to task oriented episodic and procedural memories and it deserves further investigations.

Acknowledgments. Thanks are due to Richárd Bellon, Dávid Hornyák, Mike Olasz, and Róbert Rill for running the experiments. Research was supported by the European Union and co-financed by the European Social Fund (grant no. TÁMOP 4.2.1./B-09/1/KMR-2010-0003) and by the EIT ICTLabs grant on *CPS for Smart Factories*.

References

1. Altmeyer, S., Cucu-Grosjean, L., Davis, R.I.: Static probabilistic timing analysis for real-time systems using random replacement caches. Real-Time Systems **51**(1), 77–123 (2015)
2. Angluin, D.: A note on the number of queries needed to identify regular languages. Information and Control **51**(1), 76–87 (1981)
3. Boutilier, C., Dearden, R., Goldszmidt, M., et al.: Exploiting structure in policy construction. IJCAI **14**, 1104–1113 (1995)
4. Carroll, J.B.: The higher-stratum structure of cognitive abilities. In: The Scientific Study of General Intelligence, ch., pp. 5–21. Pergamon (2003)
5. Graziano, M.: The organization of behavioral repertoire in motor cortex. Annu. Rev. Neurosci. **29**, 105–134 (2006)
6. Gyenes, V., Bontovics, Á., Lőrincz, A.: Factored temporal difference learning in the New Ties environment. Acta Cybern. **18**(4), 651–668 (2008)
7. Kalmár, Z., Szepesvári, C., Lőrincz, A.: Module-based reinforcement learning: Experiments with a real robot. Machine Learning **31**, 55–85 (1998)
8. Lőrincz, A., Pólik, I., Szita, I.: Event-learning and robust policy heuristics. Cognitive Systems Research **4**(4), 319–337 (2003)
9. Orlosky, J., Toyama, T., Sonntag, D., Sárkány, A., Lőrincz, A.: On-body multi-input indoor localization for dynamic emergency scenarios. In: IEEE Int. Conf. on Pervasive Comp. Comm. Workshop, pp. 320–325. IEEE (2014)
10. Puterman, M.: Markov decision processes. John Wiley & Sons, New York (1994)
11. Ribeiro, L., Rocha, A., Veiga, A., Barata, J.: Collaborative routing of products using a self-organizing mechatronic agent framework - a simulation study. Comp. Ind. **68**, 27–39 (2015)
12. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61**, 85–117 (2015)
13. Szepesvári, C., Littman, M.L.: Generalized Markov decision processes. In: Proceedings of International Conference of Machine Learning 1996, Bari (1996)
14. Szita, I., Lőrincz, A.: Learning to play using low-complexity rule-based policies. J. Artif. Int. Res. **30**, 659–684 (2007)
15. Szita, I., Lőrincz, A.: Optimistic initialization and greediness lead to polynomial time learning in factored MDPs. In: Int. Conf. Mach. Learn., pp. 1001–1008. Omnipress (2009)
16. Szita, I., Takács, B., Lőrincz, A.: Epsilon-MDPs. J. Mach. Learn. Res. **3**, 145–174 (2003)