Structural integrity management via hierarchical resource allocation and continuous-control reinforcement learning

Andriotis, Charalampos; Metwally, Ziead

Structural integrity management via hierarchical resource allocation and continuous-control reinforcement learning

Charalampos P. Andriotis
Assistant Professor, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Netherlands

Ziead Metwally
PhD Candidate, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Netherlands

ABSTRACT: Maintenance planning of engineering systems is often posed as a stochastic optimal control problem, aimed at determining a series of discrete interventions that upkeep structural integrity. Advanced algorithmic schemes within the joint framework of Partially Observable Markov Decision Processes (POMDPs) and multi-agent Deep Reinforcement Learning (DRL) have been recently able to approximate well global optima for this complex problem, outperforming existing time- and condition-based decision strategies. Integral to their success is the hypothesis that system components represent individual agents who form cooperative policies to minimize a central life-cycle cost. Thereby, the policy output scales linearly with the number of components, alleviating the curse of dimensionality related to combinatorial choices. State complexity and long-term optimality are handled efficiently via deep learning and POMDP principles, respectively. However, the efficiency of multi-agent coordination can fade as the number of agents increases. To this end, we propose a new formulation: we pose the problem as a continuous-control dynamic resource allocation one, combining hierarchical DRL and mixed-integer programming. Moving from flat decentralized to hierarchical multi-agent decompositions allows us to improve further the policy output scalability. The new Adaptive Knapsack Hierarchical Resource Allocator (AK-HRA) DRL architecture distributes available resources within the system, creating local, independently solvable, multi-choice knapsack optimization problems. By design, AK-HRA allows decision-makers to inscribe known hierarchical structures and local decision rules in their architectures, thereby enhancing control and interpretability over the solution space. The efficacy of the new approach is demonstrated in a multi-component reliability system subject to stochastic deterioration.

1. INTRODUCTION

Structural degradation induced by several stressors and hazards poses a continuous threat to structural safety. In view of an aging and growing built environment, the study of the onset, evolution, and control of deterioration processes, therefore, becomes more and more crucial. This motivated a significant rise of systematic decision analysis and optimization frameworks from the risk and reliability community in past decades, in an effort to mitigate the emerging socioeconomic impacts (Marseguerra et al., 2002; Faber and Stewart, 2003; Straub and Faber, 2005; Frangopol et al., 2012; Papakonstantinou and Shinozuka, 2014; Biondini and Frangopol, 2016; Andriotis and Papakonstantinou, 2019; Morato et al., 2022).

Traditional methodologies for determining maintenance policies for structural systems have largely revolved around informed constructs of engineering judgment, considering time- and condition-based decision rules. These typically seek to optimize intervals between actions and performance thresholds that, when violated, activate corrective or preventive interventions and inspection or monitoring strategies (Marseguerra et al., 2002; Straub and Faber, 2005; Ahmad and

Kamaruddin, 2012; Frangopol et al., 2012). Despite their interpretability merits due to straightforward static optimization statements and respective computational workflows, time- and threshold-based methods can manifest optimality and scalability issues as we increase the complexity of the studied system.

A solution to such limitations comes from the family of stochastic optimal control and (approximate) dynamic programming methods. In (Ellis et al., 1995; Corotis et al., 2005) a Partially Observable Markov Decision Process (POMDP) was proposed to solve nonstationary stochastic inspection and maintenance of deteriorating systems. POMDPs accommodate joint inference and control, by handling noisy observations and actions in a common Bayesian framework. The main burden in incorporating POMDPs had traditionally been the complexity to solve them. To this end, in (Papakonstantinou and Shinozuka, 2014; Memarzadeh et al., 2015; Papakonstantinou et al., 2018), the problem was formulated within point-based value iteration, which allowed POMDPs to scale in inspection and maintenance applications with hundreds of states, actions, and observations. A few years later, in (Andriotis and Papakonstantinou, 2019; 2021), the development of multi-agent actor-critic deep reinforcement learning algorithmic formulations for this POMDP problem, allowed us to trace so far intractable solutions in deteriorating engineering systems with hundreds of components and combinatorically explosive state, action, and observation spaces. Single-agent DRL formulations have also demonstrated important potential when limited intervention actions are involved, employing deep Q-network architectures (Rocchetta et al., 2019; Zhang and Si, 2020).

Decentralized multi-agent actor-critic DRL formulations have been proven to be particularly efficient when dealing with multi-component environments, as they allow for linear scaling of system-level decisions with the number of involved components, under the assumption that the latter are controlled by individual agents who try to optimize a common function within the principles of centralized training / decentralized execution (Andriotis and Papakonstantinou, 2019, 2021; Saifullah et al., 2022, Morato et al., 2023). The next big challenge is to scale up in systems with a massive number of components, e.g. in the order of thousands, with limited optimality losses. In such environments, coordination of multiple discrete agents can become hard, whereas the presence complex and noisy spaces calls for exploiting any available prior knowledge of topological and statistical structures of the system at hand, in order to improve the learning and interpretability properties of the employed algorithmic decision-making architectures.

To address this need, we present a novel formulation within hierarchical DRL, a structured framework for intelligent sequential decision-making relying on decomposing complex tasks into multiple, consecutive, simpler subproblems (Kulkarni et al., 2016). In (Zhou et al., 2022), the hierarchies are applied for maintenance planning according to known component importance, whereas in (Botteghi et al., 2022) they are used for autonomous navigation of robots in pipe inspection. Following successful off-policy actor-critic training principles from (Andriotis and Papakonstantinou, 2019, 2021) we herein develop a hierarchical DRL framework based on a new mathematical statement: we bring the problem in a continuous-control dynamic resource allocation form and introduce the Adaptive Knapsack Hierarchical Resource Allocator (AK-HRA) DRL architecture to solve it. AK-HRA consists of hierarchical actors. Upstream actors distribute the resources to be used by downstream actors, thus eventually a structured flow of resources from system-, to subsystem- to component-level is sought. At the ultimate hierarchical layer lies a number of 'leaf' subproblems determining the discrete action selection. These are solved as multi-choice knapsack problems, whose knapsack size is governed by the upstream DRL outputs. Implementation details of AK-HRA are discussed and its performance is assessed in a structural integrity management problem of a degrading system that allows for thorough evaluation.

2. BACKGROUND

In DRL-driven structural integrity management optimization, we aim to determine an optimal mapping from the system belief over deterioration /damage states to the corresponding intervention and information actions throughout a planning horizon. Typically, the agent starts by exploring the environment randomly, far from the optimal region. As the training goes on, the agent accumulates knowledge about the environment by receiving noisy observations, $o_t \in \Omega$, of its state, $s_t \in S$, at different time steps, t, taking actions, $a_t \in A$, and collecting costs, $c_t \in C$. The agent cannot observe s_t accurately, therefore, the formed policy is at best a mapping from the history $h=(a_{0:t-1},o_{0:t})$ to the current action, a_t. When the transition and observation models are known or computable offline, which is not uncommon for deterioration models of engineering systems, we can form a sufficient statistic of h, which is belief, b_t. The mapping from beliefs to actions is generally stochastic and the sought policy is $\pi=\Pr(a_t \mid b_t)$.

Throughout training, π is refined and updated to determine the optimal policy π^* within a region of feasible policies Π_c, defined by the stochastic and/or deterministic constraints of the optimization problem (Andriotis and Papakonstantinou, 2021):

$$\pi^* = \arg\min_{\pi\in\Pi_c} \mathbb{E}\left(\sum_{t=0}^{T} \gamma^t c_t \middle|\ a_t \sim \pi\left(o_{0:t}, a_{0:t-1}\right), b_0 \right) \quad (1)$$

where γ is a positive discount factor lower than 1.0 translating future costs to the current value.

After each decision step, belief b_t, which is essentially a probabilistic distribution over the possible states, is computed via Bayesian updates. Following standard POMDP assumptions Eq. (1) can be written as:

$$\pi^* = \arg\min_{\pi\in\Pi_c} V^{\pi}\left(b_0\right) \quad (2)$$

where V^{π} is the value function, representing the cumulative costs until termination of the planning horizon, $t=T$.

In an off-policy actor-critic DRL setting, the policy and the value functions are parametrized by actor and critic neural networks, respectively. For the actor the (policy) gradient is:

$$g_\theta = \mathbb{E}\left[w_t \nabla_\theta \log \pi\left(a_t|b_t,\theta\right) A^{\pi}(b_t,a_t) \right] \quad (3)$$

where θ is the actor neural network weights vector, w_t is an importance sampling weight, and A^{π} is the advantage function. The expectation in Eq. (3) is approximated by randomly sampling a replay buffer (Andriotis and Papakonstantinou, 2019). The advantage function used takes the form:

$$A^{\pi}(b_t,a_t) = c(b_t,a_t) + \gamma V^{\pi}(b_{t+1} \mid \varphi) - V^{\pi}(b_t \mid \varphi) \quad (4)$$

where φ is the critic neural network weights vector. The corresponding gradients are calculated using the mean squared error as a loss function:

$$g_\varphi = \mathbb{E}\left[w_t \nabla_\varphi V^{\pi}(b_t) A^{\pi}(b_t,a_t) \right] \quad (5)$$

In the next section the specific form and parametrization of the policy function for the AK-HRA architecture is presented and discussed.

3. HIERARCHICAL RESOURCE ALLOCATOR

3.1. Problem decomposition & parametrization

Hierarchical reinforcement learning relies on constraining lower-hierarchy decision spaces by higher-hierarchy ones in a nested fashion, thereby creating simpler local policy subspaces (Florensa et al., 2017). Building on this principle, we introduce a novel Adaptive Knapsack Hierarchical Resource Allocator (AK-HRA) DRL architecture to determine optimal intervention policies for engineering systems. AK-HRA DRL seeks to solve a continuous-control resource allocation problem. Structural integrity management is fundamentally a dynamic resource allocation problem and, if formulated as such, the task that DRL agents undertake is to distribute optimally a given resource, such as budget, over time. This allocation is here not done directly from system-to-component, but is supported by a hierarchical structure that mimics the system structure, and leverages known locally optimal responses. Thereby , it distributes resources gradually from

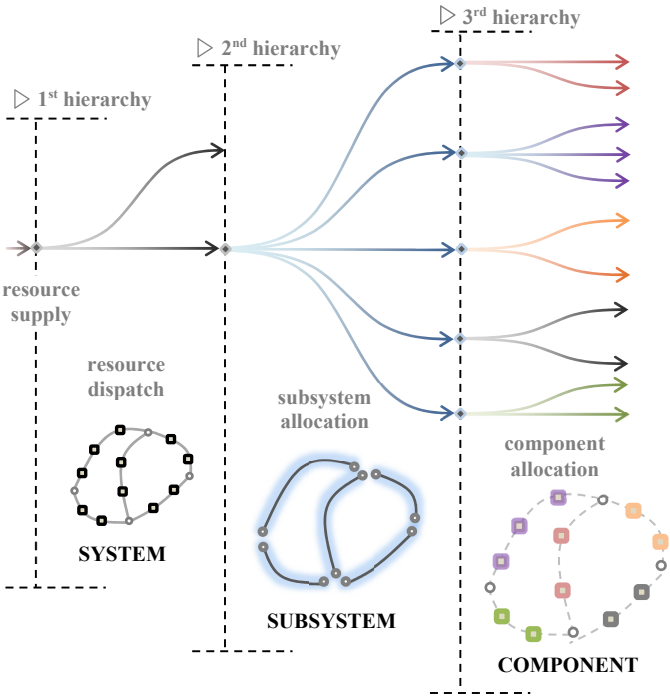

Figure 1: Decision layers for hierarchical resource allocation problem decomposition.

system, to subsystems, up to components level.

The concept forming the basis of the AK-HRA decompositions is illustrated in Fig. 1, where, without loss of generality, the resources are streamed into three levels (system, subsystem, and component). Each rhombus represents a distribution node, that controls the flow of resources streamed down to lower hierarchical levels. Accordingly, the optimal policy, π, is the one that optimizes the division of resources as we traverse the hierarchical structure from left to right. At the core of this architecture lies the hypothesis that, however complex a structural management policy is, it can be decomposed into simple resource allocation decisions. Synthesizing back these decision primitives, we can capture global policies of increased sophistication.

The distribution branches are modeled as Dirichlet distributions (Gammelli et al., 2021; Tian et. al, 2022), which provide a consistent form for continuous allocation applications:

$$\pi_d\left(y_{d,t}|b_{d,t}\right)=\frac{1}{B\left(\alpha_{d,t}\right)}\prod_{i=1}^{K_d}y_{t,i}^{\alpha_{d,t,i}-1} \qquad (6)$$

$$\sum_{i=1}^{K_d}y_{d,t,i}=1, y_{d,t,i}\geq 0, \alpha_{d,t,i}>0$$

where $d=1,2,\ldots,D$ is the distribution node index; $y_{d,t}$ is the K_d-dimensional output of the Dirichlet; $\alpha_{d,t}$ is the vector of concentration parameters; and B is a multivariate Beta function, expressed as a function of the Gamma function, Γ:

$$B\left(\alpha_{d,t}\right)=\frac{\prod_{i=1}^{K_d}\Gamma\left(\alpha_{d,t,i}\right)}{\Gamma\left(\sum_{i=1}^{K_d}\alpha_{d,t,i}\right)} \qquad (7)$$

Eqs. (6),(7) are used for each one of the rhombus nodes of Fig. 1. As such, consecutive decision layers are designed to allocate resources to the entire system, subsystems, and components in a nested sequential manner. The joint policy, $\pi=\pi_{HRA}$, for N decision hierarchies can be expressed as:

$$\pi_{HRA}=\prod_{i=1}^{N}\prod_{d=1}^{D_i}\pi_d(y_{d,t}^i\,|\,b_{d,t}^i, y_t^{1:i-1}, \theta_i) \qquad (8)$$

Substituting Eq. (8) in Eq. (3), we obtain the gradients of the actors controlling the different decision hierarchies. In DRL terms, the actors learn the mapping between their belief and the Dirichlet distributions concentration parameters. Finally, the developed formulation seamlessly incorporates constraints in the same fashion as in (Andriotis and Papakonstantinou, 2021).

3.2. Multi-choice knapsack subproblems

The nested hierarchical decomposition of the original problem shown in Fig. 1, results in a number of simple and parallelizable 'leaf' subproblems at the lowest-level hierarchy. To prevent generation of branches in systems with large number of components, the stopping depth of the hierarchical structure is herein chosen to be before the component level is reached, i.e. at the 2nd hierarchy. Then, we form the resource allocation among subsystem components as a multi-choice knapsack optimization problem. This allows us to further reduce DRL parametrization and outputs, thus making training more efficient.

As such, the task of the AK-HRA agents is to define the sizes of the knapsack problems in an adaptive manner at every decision step. The multi-choice knapsack objective for the leaf problems is to maximize the cumulative risk improvement given a set of discrete maintenance actions and a DRL-prescribed budget constraint.

Overall, the final architecture constructs independent and parallelizable knapsack problems by passing the system resources across two hierarchies. The first hierarchy is responsible for deciding the budget to spend for the whole system. Conditioned on the system belief, the fraction of the budget to be spent is decided (the rest is set aside as shown by the unconnected branch of the 1st hierarchy in Fig. 1). This decision is passed on to the input layer of the 2nd hierarchy, which distributes the budget among the different subsystems. Finally, the 3rd hierarchical allocation is conducted by solving a knapsack problem for each subsystem, based on the local budget. The solution to this problem is approximated by risk reduction per unit cost ranking. Based on the finally chosen actions, the costs, advantages, and gradients are computed to update π and V^{π}.

4. APPLICATION

The partially observable deteriorating system introduced in (Andriotis and Papakonstantinou, 2021) and shown in Fig. 2 is adopted here to evaluate the developed architecture. The aim is to determine a near-optimal maintenance schedule over a 30-year planning horizon. The features adapted and added for the current set-up are:

- The expected risk, corresponding to network closure due to component failures, should not exceed a probability of 3% in 30 years.
- Three maintenance actions are available: do nothing, partial repair, and replace. Do nothing has no effect on deterioration, partial repair improves component's state by one, and replace restarts the deterioration process. Inspections are conducted every year.
- A maintenance crew mobilization cost is added on top of individual maintenance costs every time at least one non-do-nothing action is taken system-wide (15% of replacing a component).

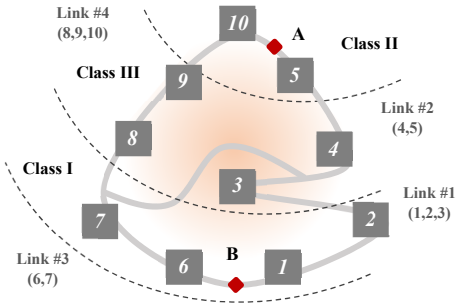

Figure 2: 10-component system with three classes of deterioration severity.

In this application three decision hierarchies are considered, one at the entire network level (system), one at the link level (subsystem), and one at the component level. The actor networks consist of two DRL levels, each including two fully connected hidden layers with 100 ReLU activation functions. No weights are shared among different hierarchical layers. The first and second hierarchical layers output a 2- and 4-dimensional softplus outputs, respectively, allocating a gradually decreasing budget supply throughout the training. The critic network has two fully connected layers with 150 ReLU activations each, and outputs a 1-demensioal linear output, which is a surrogate of the 30-year expected maintenance cost, i.e. the value function. Both networks use the Adam optimizer to update functions, with learning rates adjusted from 1E-5 to 1E-6, and 1E-4 to

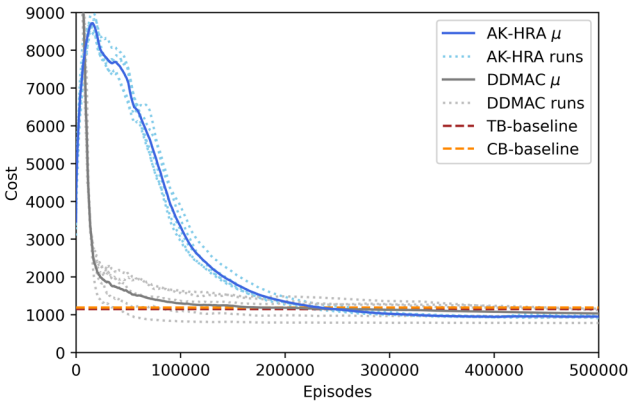

Figure 3: Training performance of AK-HRA algorithm and comparison with baseline policies.

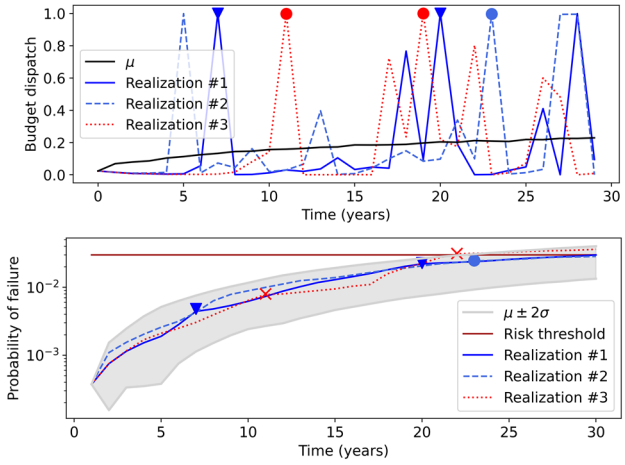

Figure 4: Policy features at the system-level: Budget dispatch (top); PoF evolution (bottom).

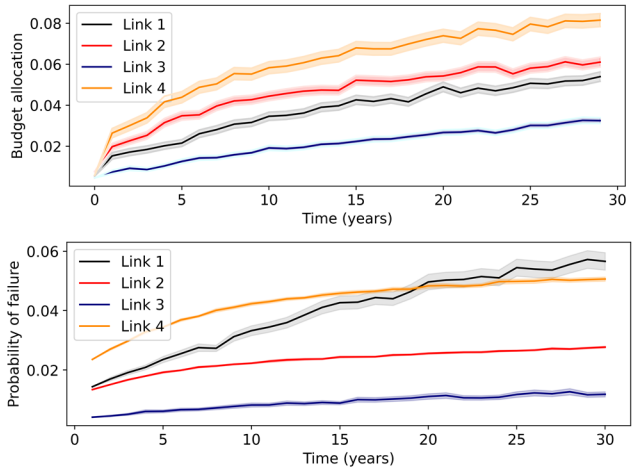

Figure 5: Policy features at the subsystems-level: Budget allocation (top); PoF evolution (bottom).

1E-5 for the actor and critic, respectively.

For evaluating performance, the developed AK-HRA architecture is compared to condition- and time-based optimized heuristics, as well as the Decentralized Multi-agent Actor Critic (DDMAC) baseline. Multiple DRL instances are trained, each of which for 500K episodes and the best 5 are reported for each type. The total maintenance policy costs during training are presented in Fig. 3. It is shown that AK-HRA surpasses the time- and condition-based baselines after ~250K training life-cycle realizations, eventually reaching policies improved by ~25%, comparing the expected cost of the policy retrieved from the converged agents, under 10K realizations. The 30-year risk constraint was observed to be stably met from early training stages (~50K episodes). AK-HRA compares well with the flat decentralized DDMAC architecture. Observing the mean training curves, AK-HRA is ~8% better, whereas the best AK-HRA reaches ~110-75% of the expected life-cycle cost of the best and worst DDMAC, respectively, under 10K realizations. Importantly, this is achieved with 6 outputs, instead of 30 in the decentralized case.

To visualize the dynamics of the trained policy on a temporal scale, Fig. 4 shows the risk evolution over time during the planning horizon for three realizations accompanied by the corresponding dispatched budget (as a ratio of the total supply), as predicted by the first hierarchical Dirichlet layer. Naturally, at the beginning of the life-cycle, components are let to deteriorate with less control actions. Maintenance interventions slow down the risk evolution, as observed at the marked points of Fig. 4, and correspond to spikes in the dispatched budget. We observe that the dispatched budget remains mostly relatively low (i.e., lower than ~0.1), which corresponds to none or minor interventions, except for sparser spikes, which indicate moments of major intervention needs, involving simultaneous maintenance across the system. The preference to rare simultaneous actions is explained by the agents attempt to eliminate unnecessary mobilization costs. Naturally, the spikes have visible effects in the slope of the risk evolution curves throughout the planning horizon. Overall, the devised adaptive intervention schedule manages to keep the expected risk within the prescribed Probability of Failure (PoF) threshold over the planning horizon.

Focusing on the subsystem hierarchical layers, it is observed that the Dirichlet-allocated budget and the respective PoF dynamics depend on the mixture of deterioration classes and number of components of the subsystem (i.e., link). For instance, link #4 has two components that belong to the severe deterioration class, resulting in higher allocated resources and PoF, as seen in Fig. 5 (incl. the 95% confidence intervals). The subsystem allocation hierarchy essentially allows the algorithm to weigh the relative importance of a link to the budget (simpler than assigning actions). As

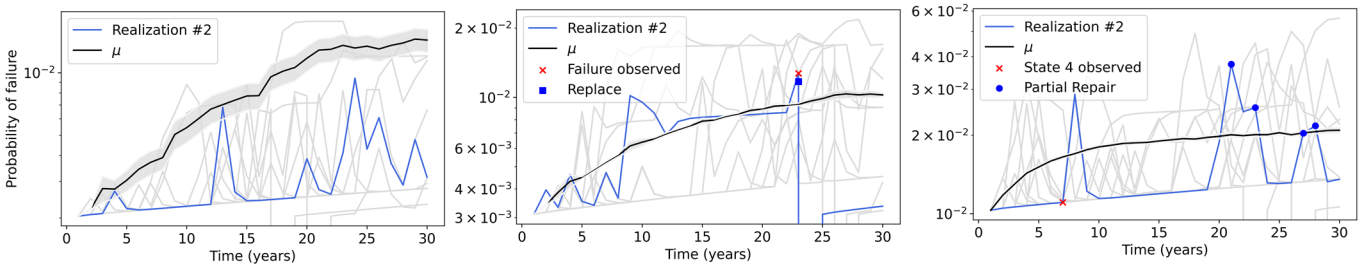

Figure 6: Policy realizations and probabilities of failure (incl. 95% confidence intervals) for low-, medium-, and high-severity class deteriorating components: #2 (left), #10 (middle), and #8 (right).

seen in Fig. 5, from high to low importance, the links' order is #4,2,1,3. An interesting dynamic is observed for links #1,4. Link #1 is let to degrade faster than its counterparts. This link consists of 3 components, one high- and two low-severity ones, and is the safest link of the least reliable path from A to B (links #1,4). Thus, for as long as link #4 PoF is higher, there is little incentive to upkeep link #1.

Fig. 6 focuses on the component level, where the actual maintenance actions are chosen, highlighting an indicative realization and corresponding statistics of component PoF (following realization #2 from Fig. 4). Three characteristic components of each class are shown. In realization time, fluctuations come from the noisy observations, which trigger changes in prior beliefs. For instance, state 4 is observed for component 8 (severest deterioration class) at step 7, resulting in a major increase of the perceived risk. However, subsequent observations, updating the belief further, confirm that the component is deteriorating at a steady rate and no intervention is, therefore, imposed. Partial repair actions are taken at later stages to reduce increased deterioration. On the contrary, no intervention is needed for component 2 (low-severity class) due to its relatively low probability of failure over the planning horizon. Similarly, component 10 (medium deterioration class) is maintained only due to failure. Overall, the schedule remains robust to jumps in risk estimates, and the agents tend to postpone maintenance interventions to leverage discounted costs and new state observations.

5. CONCLUSIONS

We introduce a new formulation for structural integrity management optimization, articulating the problem as a continuous-control dynamic resource allocation one. To facilitate allocation from the global system-level to the local component-level, we devise hierarchical decision layers leveraging prior knowledge of the system structure. To solve the resulting problem, we propose a novel Adaptive Knapsack Hierarchical Resource Allocator (AK-HRA) DRL architecture, which consists of hierarchical actors that control the flow of resources across the different decision layers. Upstream actors dispatch and distribute the resources to be used by downstream ones, thus streamlining resources from system to subsystem to component level in a principled manner. By traversing the formed hierarchies, the global decision is shaped along the way, whereas the problem is ultimately broken down into a number of independent and parallelizable multi-choice knapsack problems that determine the actions for each component. AK-HRA DRL is implemented in a maintenance planning problem of a deteriorating system under risk constraints, where it is shown to outperform traditional baselines, and to reach competitive alternative optima to state-of-the-art multi-agent formulations based on flat decentralization. Further, the new formulation allows us to considerably reduce the policy output, while also bringing interpretability advances.

ACKNOWLEDGEMENTS

This material is based upon work supported by the TU Delft AI Labs Program.

6. REFERENCES

Ahmad, R., and Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers and Industrial Engineering,* 63(1), 135-149.

Andriotis, C. P., and Papakonstantinou, K. G. (2019). Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety*, 191.

Andriotis, C. P., and Papakonstantinou, K. G. (2021). Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering & System Safety*, 212.

Biondini, F., and Frangopol, D. M. (2016). Life-Cycle Performance of Deteriorating Structural Systems under Uncertainty: Review. *Journal of Structural Engineering*, 142(9).

Botteghi, N., Grefte, L., Poel, M., Sirmacek, B., Brune, C., Dertien, E., and Stramigioli, S. (2022). Towards autonomous pipeline inspection with hierarchical reinforcement learning. In *Robot Intelligence Technology & Applications*, 259-271.

Corotis, R. B., Hugh Ellis, J., and Jiang, M. (2005). Modeling of risk-based inspection, maintenance and life-cycle cost with partially observable Markov decision processes. *Structure and Infrastructure Engineering*, 1(1), 75-84.

Ellis, H., Jiang, M., and Corotis, R. B. (1995). Inspection, maintenance, and repair with partial observability. *Journal of Infrastructure Systems*, 1(2), 92-99

Faber, M. H., and Stewart, M. G. (2003). Risk assessment for civil engineering facilities: critical overview and discussion. *Reliability engineering & system safety*, *80*(2), 173-184.

Florensa, C., Duan, Y., & Abbeel, P. (2017). Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint* arXiv:1704.03012.

Frangopol, D. M., Saydam, D., & Kim, S. (2012). Maintenance, management, life-cycle design and performance of structures and infrastructures: a brief review. *Structure and infrastructure engineering*, 8(1), 1-25.

Gammelli D, Yang K, Harrison J, Rodrigues F, Pereira FC, Pavone M. (2021). Graph neural network reinforcement learning for autonomous mobility-on-demand systems. In *60th IEEE Conference on Decision and Control* (CDC), 2996-3003.

Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., and Tenenbaum, J. B. (2016). Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *Advances in Neural Information Processing Systems*, 29.

Marseguerra, M., Zio, E., and Podofillini, L. (2002). Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation. *Reliability Engineering & System Safety*, *77*(2), 151-165.

Memarzadeh, M., Pozzi, M., and Zico Kolter, J. (2015). Optimal planning and learning in uncertain environments for the management of wind farms. *Journal of Computing in Civil Engineering*, 29(5), 04014076.

Morato, P. G., Papakonstantinou, K. G., Andriotis, C. P., Nielsen, J. S., and Rigo, P. (2022). Optimal inspection and maintenance planning for deteriorating structural components through dynamic Bayesian networks and Markov decision processes. *Structural Safety*, 94.

Morato, P. G., Andriotis, C. P., Papakonstantinou, K. G., and Rigo, P. (2023). Inference and dynamic decision-making for deteriorating systems with probabilistic dependencies through Bayesian networks and deep reinforcement learning. *Reliability Engineering & System Safety*, 109144.

Papakonstantinou, K. G., Andriotis, C. P., and Shinozuka, M. (2018). POMDP and MOMDP solutions for structural life-cycle cost minimization under partial and mixed observability. *Structure and Infrastructure Engineering*, 14(7), 869-882.

Papakonstantinou, K. G., and Shinozuka, M. (2014). Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II:P OMDP implementation. *Reliab. Engineering & System Safety*,130,214-224.

Rocchetta, R., Bellani, L., Compare, M., Zio, E., and Patelli, E. (2019). A reinforcement learning framework for optimal operation and maintenance of power grids. Applied Energy, 241, 291-301.

Straub, D., and Faber, M. H. (2005). Risk based inspection planning for structural systems. *Structural Safety*, *27*(4), 335-355.

Tian Y, Han M, Kulkarni C, Fink O. (2022) A prescriptive Dirichlet power allocation policy with deep reinforcement learning. Reliability Engineering & System Safety. 1;224:108529.

Zhang, N. L., and Si, W. J. (2020). Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering and System Safety*, 203.

Zhou, Y. F., Li, B. C., and Lin, T. R. (2022). Maintenance optimisation of multicomponent systems using hierarchical coordinated reinforcement learning. *Reliability Engineering and System Safety*, 217.