# Adaptive Multi-Fitness Learning for Robust Coordination

Connor Yates
Collaborative Robotics and Intelligent Systems Institute
Oregon State University
Corvallis, Oregon
yatesco@oregonstate.edu

Ayhan Alp Aydeniz
Collaborative Robotics and Intelligent Systems Institute
Oregon State University
Corvallis, Oregon
aydeniza@oregonstate.edu

Kagan Tumer
Collaborative Robotics and Intelligent Systems Institute
Oregon State University
Corvallis, Oregon
kagan.tumer@oregonstate.edu

## ABSTRACT

Many long term robot exploration domains have sparse fitness functions that make it hard for agents to learn and adapt. This work introduces Adaptive Multi-Fitness Learning (A-MFL), which augments the structure of Multi-Fitness Learning (MFL) [7] by injecting new behaviors into the agents as the environment changes. A-MFL not only improves system performance in dynamic environments, but also avoids undesirable, unforeseen side-effects of new behaviors. On a multi-robot coordination problem, A-MFL provides up to 90% improvement over MFL and 100% over a one-step evolutionary approach.

## CCS CONCEPTS

• **Computer systems organization** → **Evolutionary robotics**; • **Computing methodologies** → **Evolutionary robotics**; Value iteration; Agent / discrete models;

## KEYWORDS

Adaptive Multi-Fitness Learning; Fitness Structures for Learning

**ACM Reference Format:**
Connor Yates, Ayhan Alp Aydeniz, and Kagan Tumer. 2021. Adaptive Multi-Fitness Learning for Robust Coordination. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459563

## 1 INTRODUCTION

Coordinating multiple robots in long-term missions is challenging because the changing environmental dynamics are not known *a priori*. Current state-of-the-art algorithms and methods such as evolutionary robotics [2], deep learning methods [6], reward shaping methods for multiagent cooperation, and multi-task learning [3–5] are not designed with long-term adaptation in multiagent systems in mind. Recently, MFL has found success by separating execution of low-level tasks from the solution to the overall mission [7]. While promising, MFL is like other state-of-the-art methods as it cannot adapt to dynamic environments during a deployment. We introduce A-MFL to address adaptation during deployment by integrating new behaviors to handle environmental disturbances without expensive relearning of the agents' controllers.

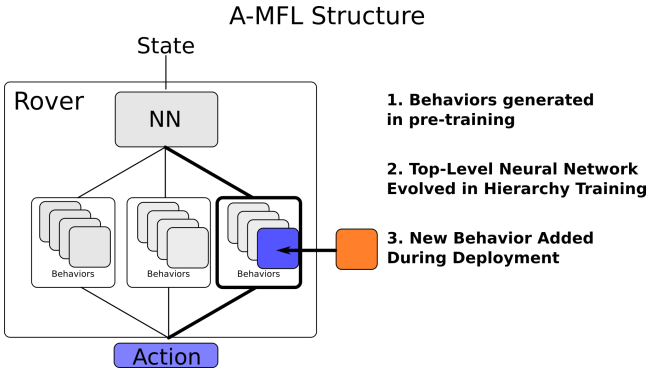


**Figure 1: Illustration of the components of A-MFL and the three phases they operate under: pre-training, hierarchy training, and deployment adaptation.**

The main contribution of this paper is to inject new behaviors into trained agents to enable them to adapt to unexpected environmental changes, and to integrate those new behaviors via value iteration over to support a high-level policy that tracks the overall mission goal.

## 2 ADAPTIVE MULTI-FITNESS LEARNING

An *adaptive* learning structure enables agents to change their actions when they encounter a situation they currently do not know how to solve. This is distinct from *autonomous* adaptation, where the agents are deciding when and how to adapt to new situations. A-MFL is a learning structure that provides adaptivity to agents trained with evolutionary algorithms with minimal re-training.

The basis for adaptive learning in this work is *behaviors*, which define A-MFL's tiered structure and value-iteration populations. Adaptive Multi-Fitness Learning (A-MFL) is able to adapt to the changing environment by extending the behavior-focused hierarchy of MFL. This work formalizes behaviors as the pair of the policy and the fitness used to train the policy. Identifying and exploiting the differences between behaviors is the main challenge facing agent teams.

**Adaptive Multi-Fitness Learning (A-MFL)** learning uses a two-tier hierarchy to take actions, where at the top level a neural network evolves to optimize the sparse global fitness, and low level actions are generated as a result of learned behavior policies.


*GECCO '21 Companion, July 10–14, 2021, Lille, France*

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3459563

As A-MFL adapts agents while deployed, three different phases are used throughout the life-cycle of the robot: **pre-training**, **hierarchy training**, and **deployment adaptation** (Figure 1).

In **pre-training**, numerous discrete behaviors are trained independent of one-another. In **hierarchy training** the top level policy is trained via neuroevolution to select between behavior pools. By modeling the population as a multi-armed bandit, a single behavior is picked from the population via $\epsilon$-greedy selection on the behavior values and the selected behavior is used to physically execute low-level actions. The global fitness updates the value of each behavior, with $G = 1$ increasing the value and $G = 0$ decreasing it.

During **deployment** a single behavior population has new behaviors added and value-iteration is resumed. Adaptation to the environment happens *without needing to re-train the top level policy, and without impacting the selected behaviors in other populations.*

## 3 EXPERIMENTS AND RESULTS

A-MFL is tested on a modified version of the Continuous Rover Problem [1]. The goal for the team of rovers is to observe point of interest (POI) scattered around a two-dimensional plane.

Two modifications are made from this domain. First, POI are heterogeneous and must be observed in a specific order (Equation 1). Second, some POI will become "sticky" and change how rovers move around the environment. The fitness in Equation 1 measures if the POI have been observed in the correct order, where $I_A$ is a Boolean function reporting if the team observed a type A POI, $t_A$ is the time at which the team observed a type A POI, and the others as follows. Critically, the observation order is not known to the rover team when they enter the world.

$$G = I_A \cdot I_B \cdot I_C \cdot (t_A < t_B < t_C) \tag{1}$$

A-MFL is compared against a neural-evolved controller and MFL. To equalize the knowledge given to A-MFL and MFL, every behavior in A-MFL's populations is an independent selection for MFL.

Figure 2 shows the first requirement for our rover team; A-MFL is able to learn the general solution to the sequential observation problem in the same way as MFL when both are presented with the same behaviors. In this situation, both MFL and A-MFL have access to every behavior; the earlier convergence of A-MFL comes from the bundling of behaviors by similarity.

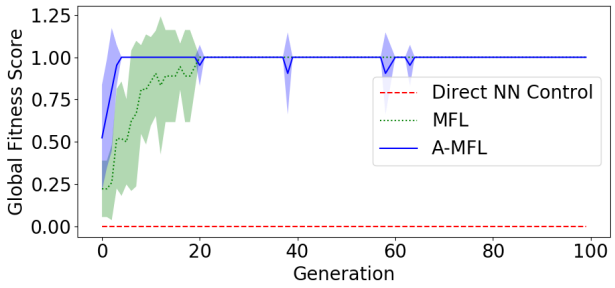


**Figure 2: Contrasting performance of direct control of the rover, providing every single behavior policy as an independent selection for MFL, and A-MFL.**

Next, agents were trained on the standard environment with the global fitness (Equation 1). Then agents are deployed into an environment where Type A POI are "sticky" and will stop agents if they move at a speed less than two if they move within 3 units of the POI. New behaviors that can move through the sticky area are injected into A-MFL at epoch five, A-MFL quickly integrates these policies and increases the score to 1 (Figure 3).

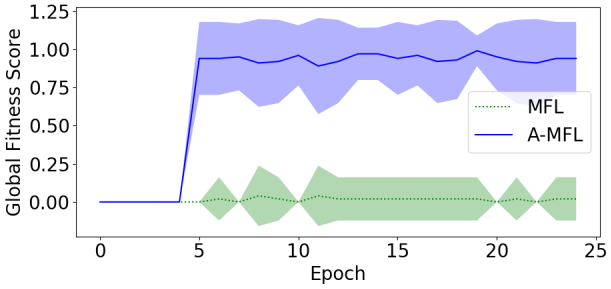


**Figure 3: MFL and A-MFL performance on the sticky domain. When new behaviors are added to deal with the sticky POI, A-MFL quickly resumes achieving high scores while MFL cannot incorporate the new behaviors.**

## 4 DISCUSSION

This paper introduces Adaptive Multi-Fitness Learning, a learning structure for multiagent teams which learns to select behaviors grouped by similarity. By grouping behaviors by similarity, a general solution to the problem can be learned by the agent team using whichever behaviors it learns to use. Then during deployment, similar behaviors can be selectively changed to adapt to unforeseen changes in the environment.

## ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under grant No. IIS-1815886 and the Air Force Office of Scientific Research under grant No. FA9550-19-1-0195.

## REFERENCES

[1] Adrian K. Agogino and Kagan Tumer. 2008. Analyzing and visualizing multiagent rewards in dynamic and stochastic domains. *Autonomous Agents and Multi-Agent Systems* 17, 2 (Oct. 2008), 320–338. https://doi.org/10.1007/s10458-008-9046-9

[2] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. 2014. Flop and roll: Learning robust goal-directed locomotion for a tensegrity robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2236–2243.

[3] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1–10.

[4] Michael L Seltzer and Jasha Droppo. 2013. Multi-task learning in deep neural networks for improved phoneme recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 6965–6969.

[5] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*. 527–538.

[6] Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. 2017. Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*. 5392–5402.

[7] Connor Yates, Reid Christopher, and Kagan Tumer. 2020. Multi-fitness learning for behavior-driven cooperation. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 453–461.