Improving the Morphology and Control Policy of Self-reconfiguring Modular Robots in Dynamic Environment (Student Abstract)

Dong Xue, Qiang Lu, and Jian Li
Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing, China
d.hsueh@foxmail.com

Abstract

Ideally, self-reconfiguration modular robots (SMR) can change their morphology and perform actions related to a specific task in any scene. However, most SMRs only adapt to several specific scenes because their morphology and control policies are designed or trained based on these scenes. Once SMRs meet an unknown scene, especially multiply unknown scenes (called dynamic environment), these policies will be useless. Although some of these policies import evolutionary algorithms to enhance the ability of SMR to explore unknown scenes, they are very time-consuming. The reason is that individual fitness depends on the interaction between SMR and these scenes. We propose a two-stage reconfiguration algorithm (TSRA) without any prior knowledge to address the time-consuming problem. In the two stages, the reconfiguration methods use the evolutionary algorithm (GA) to simultaneously generate scene-fitted morphology and actions. The first stage method uses the estimation neural network to evaluate the individual fitness to run faster and can recommend better policies to the second stage. The second stage method obtains this fitness from scenes and updates the neural network to approximate these scenes. Through experiments, TSRA can find better morphology and control policies than the other two canonical algorithms — GA and GEM-RL.

Introduction

Self-reconfiguring modular robots (**SMR**) can change their morphology (body) and make the morphology-fitted behavior control (brain) under different scenes (Rus et al. 2002). This paper focuses on making SMR run in a dynamic environment, which is very close to the above SMR ultimate goal. We first define what is the dynamic environment and formalize the goal of how to let SMR run in this environment. However, the goal is almost impossible to accomplish without any prior scene knowledge in SMR. Because the goal depends on SMR can recognize scenes in the environment, which conflict with the "unknown" scene.

To approximate the goal, we propose a two-stage reconfiguration algorithm (**TSRA**) based on the local scene that the SMR has traveled during the specific period. TSRA first finds a suitable SMR reconfiguration in its experience environment. It then uses the reconfiguration to initialize SMR, let SMR run in the unknown environment, and records the environment's evaluation value. The first stage is experience reconfiguration (ER), while the second is the actual reconfiguration (AR). ER and AR use a genetic algorithm (GA) to find SMR configurations (body shape and behavior). However, the difference between them is that ER utilizes the estimator to evaluate these configurations, while AR imports the real "unknown" environment to evaluate these configurations. Compared with AR, ER runs faster and can evaluate more configurations for SMR because its evaluation does not require interaction with the environment.

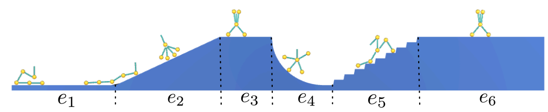


Figure 1: SMR running in a dynamic environment.

Problem Definition

A environment $\boldsymbol{E}$ consists of scenes ($\boldsymbol{e}$), denoted as $E = \{e_1, e_2, ..., e_n\}$ and $e_i \cap e_j = \phi$ as shown in Figure 1. If any two adjacent scenes are different and unknown in the environment, the environment is called a **dynamic environment**. Suppose the time that the robot r_{e_i} passes through the scene e_i is t_i, i.e., $t_i = env(r_{e_i}) = env(\langle\beta(e_i), \zeta(\beta(e_i), e_i)\rangle)$, where $\beta(e_i)$ is the morphology (structure) of the robot at e_i. $\zeta(\beta(e_i), e_i)$ shows the behavior control (actions) of the robot at e_i. The paper aims to let the robot quickly pass through the unknown dynamic environment E, defined as the following formula.

$$\arg\min_{\beta,\zeta} \sum_{i=1}^{n} env(\langle\beta(e_i), \zeta(\beta(e_i), e_i)\rangle) \tag{1}$$

However, in a dynamic environment, the robot can only observe its location scene, not all scenes. For realizing Equation 1, the robot must pass through every scene e_i as quickly as possible. So, Equation 1 is changed into Equation 2

$$\sum_{i=1}^{n} \arg\min_{\pi} env(\pi(e_i)) \tag{2}$$

where $\pi(e_i) = \langle\beta(e_i), \zeta(\beta(e_i), e_i)\rangle$.

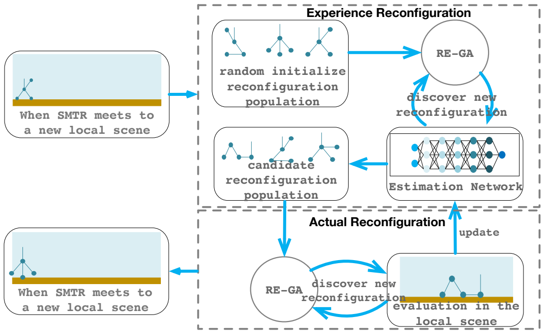
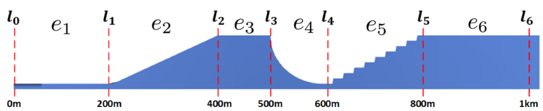
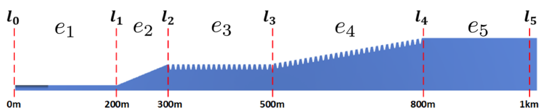


Figure 2: Two Stage Reconfiguration Algorithm – TSRA.

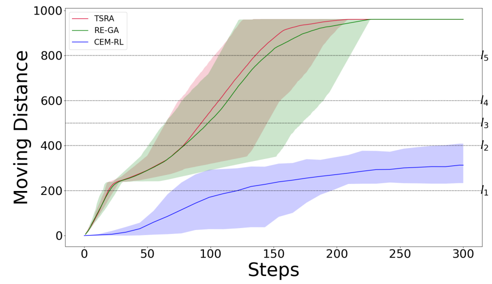


(a) Dynamic Environment 1

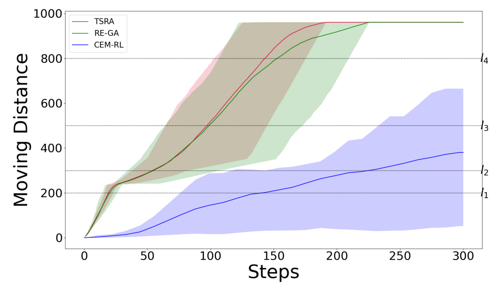


(b) Dynamic Environment 2

Figure 3: Experimental Environments.



(a) Experimental Results in Dynamic Environment 1



(b) Experimental Results in Dynamic Environment 2

Figure 4: Experimental Results.

Two Stage Reconfiguration Algorithm–TSRA

The algorithm contains two stages of SMR reconfiguration, as shown in Figure 2. The first stage is called experience reconfiguration (**ER**), which is used to find simulated SMR morphology and behavior with GA and the estimator. The second stage is called actual reconfiguration (**AR**), which discovers the scene-fitted morphology and behavior with GA. AR updates the estimator according to feedback from the real scene. At last, AR will recommend the best morphologies for SMR in the current scene.

Experiments

To evaluate the proposed algorithm TSRA, we created two simulated dynamic environments in Unity3D (Juliani et al. 2018) and ran SMR in them with the ml-agent toolkit, as shown in Figure 3. Each of the two environments contains four different scenes. Except for the algorithm TSRA, the other two algorithms RE-GA and CEM-RL [1] were introduced into these experiments.

After each of the three algorithms – TSRA, RE-GA, and GEM-RL has been run ten times, their results are shown in Figure 4. TSRA and RE-GA both can find better morphologies and behaviors, making SMR walk through the two dynamic environments at about 170 steps. CEM-RL cannot discover morphology and behavior, which makes SMR walk through them. Compared with RE-GA, TSRA can discover morphology and behavior, which lets SMR pass through the two environments faster. Because TSRA uses the estimator in the ER stage, making TSRA evaluate the solution to morphology and behavior quickly.

[1]https://github.com/apourchot/CEM-RL

Conclusion

This paper firstly defines the dynamic environment and formally describes the objective function that allows self-configure robots (SMR) to walk through the dynamic environment as quickly as possible. The two-stage reconfiguration algorithm (TSRA) is then proposed to solve the objective function. Owing to the estimator and the encoding that consists of morphology and a sequence of actions, TSRA can make SMR change their morphology to fit an unknown scene and do the correct actions to pass through the scene quickly without any experience. Experiments show that TSRA is more efficient than the other two algorithms – RE-GA and CEM-RL.

Acknowledgments

This work was supported by China National Key Research Project (no. 2019YFC0312003), National Natural Science Foundation of China (Grant no. 61402532),

References

Juliani, A.; Berges, V.; Vckay, E.; Gao, Y.; Henry, H.; Mattar, M.; and Lange, D. 2018. Unity: A General Platform for Intelligent Agents. *CoRR* abs/1809.02627. URL http://arxiv.org/abs/1809.02627.

Rus, D.; Butler, Z.; Kotay, K.; and Vona, M. 2002. Self-reconfiguring robots. *Communications of the ACM* 45(3): 39–45.