Group 10 Project Report: Policies for Gazebos under multiple constraints

Aadithya Velayutham **Arpit Chandrakar** **Vineeth Veligeti** **Shravan Nermati**

Aditya Dhavala (Leader)
Arizona State University
adhaval1@asu.edu

Abstract

This project tackles the challenge of navigating varied terrains with conflicting objectives in robot control—balancing energy consumption and maximum speed. To address this, an efficient evolutionary learning algorithm is introduced to approximate optimal policies in multi-objective reinforcement learning. Seven purpose-built environments with continuous action spaces serve as benchmarks for evaluating these algorithms.

The primary aim is to find optimal policies for navigating diverse terrains while balancing energy consumption and speed, leveraging existing policies to synthesize a new approach. Genetic mutation within the MuJoCo humanoid explores control signal spaces, making subtle adjustments in joint movements to enhance both speed and energy efficiency. Assessing energy expenditure through manipulated control signals informs the optimization process.

This study contributes an evolutionary algorithm for approximating optimal policies and sheds light on reconciling conflicting objectives in robot navigation. The evaluation within created environments provides insights into these methodologies' efficacy, paving the way for advancements in multi-objective reinforcement learning for robotics.

1 Introduction

The intricacies of navigating diverse terrains while harmonizing conflicting objectives—balancing energy consumption with maximum speed—are pivotal in shaping the future of robotics. This project delves into a critical realm of continuous robot control, addressing a challenge that holds significant implications for autonomous systems operating in various fields, from exploration missions in unfamiliar terrains to optimizing logistical processes in warehouses and manufacturing facilities.

Efficiently reconciling conflicting objectives in robot navigation is indispensable in numerous real-world applications. Autonomous vehicles exploring uncharted territories, delivery drones maneuvering through urban landscapes, and industrial robots optimizing efficiency within complex environments all stand to benefit immensely from strategies that balance energy conservation with optimal speed.

This study introduces an evolutionary learning algorithm tailored for multi-objective reinforcement learning (MO-RL), aiming to find optimal policies that navigate through diverse terrains while accommodating different energy constraints and speed requirements. The core methodology involves designing and implementing purpose-built environments to benchmark and evaluate multi-objective RL algorithms. Within these environments, the genetic mutation strategy is employed within the

control signal space of the MuJoCo humanoid, facilitating subtle yet impactful adjustments in joint movements to enhance both speed and energy efficiency.

The subsequent sections of this paper detail the methodology employed, the formulation of purpose-built environments for evaluation, the genetic mutation approach within the MuJoCo humanoid, and the insights drawn from experimental evaluations. By exploring the intricate interplay between conflicting objectives in robot navigation, this project aims to pave the way for advancements in autonomous systems capable of navigating diverse terrains while optimizing both energy consumption and speed.

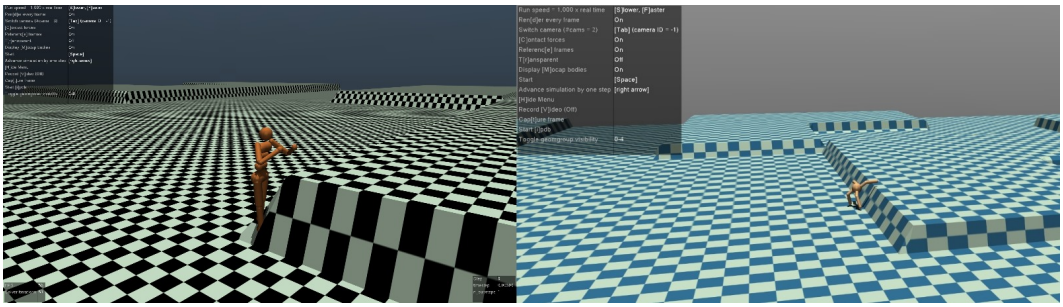

Figure 1: **Gazebos (Humanoid and Ant) navigating through custom generated terrain**

2 Background

In the realm of continuous robot control, finding optimal policies to address conflicting objectives (Pareto-optimal) is a persistent challenge. To bridge this gap, our work introduces an efficient evolutionary learning algorithm for continuous robot control, extending state-of-the-art reinforcement learning approaches. Our method approximates the Pareto set, efficiently discovering individual policies on the Pareto front and constructing a dense set of high-quality Pareto-optimal solutions through comprehensive analysis and interpolation. Seven purpose-built multi-objective RL environments with continuous action spaces serve as benchmarks for our approach. Additionally, we contribute a tool that autonomously constructs 3D landscapes from 2D images or laser range finder data, integrating seamlessly into the Gazebo simulator. This tool has been rigorously tested using simultaneous localization and mapping (SLAM) algorithms in environments of varying complexity, demonstrating efficient mission completion with acceptable real-time factor (RTF) benchmarks. In the domain of traversing terrains, RoboGrammar, employing graph-based representations and graph grammar, efficiently expresses diverse robot assemblies through a compact set of rules. The framework's unique Graph Heuristic Search method navigates complex design spaces, optimizing robot structures for challenging terrains. RoboGrammar's success in generating nontrivial, optimized robots tailored for diverse terrains highlights its potential in revolutionizing automated robot design [1-3].

3 Data and Results

In this project, our objective is to assess the performance of various environments offered by OpenAI's Gymnasium, focusing specifically on the humanoid and ant scenarios. Our aim is to evaluate the efficacy of diverse reinforcement learning algorithms while concurrently engaging in multi-objective optimization, specifically balancing speed and energy consumption. The experimentation involves training the bot within a customized terrain featuring elevations and obstacles, providing a more nuanced understanding of the algorithms' adaptability and efficiency in dynamic environments.

3.1 Terrain Generation

The terrain generation process involves transforming a grayscale image into a realistic 3D virtual environment through a two-step method. First, the grayscale image is converted into a heightmap, creating a 2D array where each element signifies the terrain's height. Grayscale values determine the heights, with black as the minimum and white as the maximum. Subsequently, the heightmap

is converted into a 3D solid model in STL format using LIBTRIX. This model, composed of small triangles representing the surface, allows for the creation of a detailed virtual terrain. A Z-scale coefficient can be applied to adjust the height of the model, offering flexibility in terrain customization[2].

3.2 Algorithms Used

3.2.1 Soft Actor Critic (SAC)

Soft Actor-Critic (SAC) is a model-free reinforcement learning algorithm designed for continuous action spaces. It addresses exploration challenges by incorporating entropy regularization, encouraging diverse actions. SAC is off-policy, making efficient use of collected data, and employs a dual-critic architecture for enhanced stability and robustness in learning policies for continuous tasks.

In the pursuit of optimizing agent behavior, Soft Actor-Critic (SAC) was adapted to incorporate a custom reward function: rewards + 10 * total speed - total energy * (0.1/6). This custom function effectively integrated speed and energy objectives into the learning process. The RL-SAC algorithm underwent training for N = 3000 iterations, with each iteration comprising time steps (T) set to 1000.

Both Humanoid and Ant models were subjected to training using identical parameters, ensuring a consistent evaluation framework. However, noteworthy limitations surfaced during training, particularly in edge cases where the Humanoid exhibited a tendency to fall, and the Ant displayed instances of flipping.

To mitigate these limitations and enhance the models' robustness, the environment's step and reward functions underwent customization. These adjustments were aimed at encouraging model survival in challenging scenarios. The tailored approach successfully addressed the identified limitations, allowing for more resilient and adaptive training outcomes in both Humanoid and Ant models.

3.2.2 Non-dominated Sorting Genetic Algorithm (NSGA II)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a powerful evolutionary optimization algorithm designed for multi-objective problems. Using non-dominated sorting and crowding distance calculations, NSGA-II efficiently generates a diverse set of non-dominated solutions, offering trade-offs for conflicting objectives. This makes it particularly effective in providing well-balanced solutions for complex optimization challenges.

In pursuit of optimizing agent behavior, the Genetic Algorithm (GA) was implemented with a population size (P) of 500 for Humanoid (17 control signals) and 100 for Ant (8 control signals). The NSGA-II algorithm was employed for multi-objective optimization over N generations (N = 300), utilizing key parameters: K = 100, cxpb = 0.7, mutpb = 0.3. Control signals underwent evaluation using a custom function that considered both speed and energy objectives.

Despite the algorithm's success in effectively optimizing speed and energy, the GA outputs revealed unconventional behaviors. Notably, the Humanoid exhibited movements resembling crawling and dancing rather than walking, while the Ant displayed erratic jumping patterns. These observations underscore the need for further refinement in the optimization process to align the algorithm's outputs more closely with the desired behaviors.

4 Conclusion

Upon analyzing the results, it becomes evident that the Ant outperforms the Humanoid in navigating complex terrains more efficiently. The Ant exhibits a notably lower speed/energy trade-off, implying a superior optimization of energy resources during traversal. This advantageous characteristic enables the Ant to explore a larger portion of the terrain map compared to the Humanoid. The Ant's ability to navigate complex landscapes with ups and downs showcases its adaptability and efficiency, positioning it as a more optimized choice for scenarios with diverse terrains.

4.1 Key Observations:

- Lower Speed/Energy Trade-off: The Ant demonstrates a more optimized use of energy resources, achieving a balance between speed and efficiency. This characteristic is particularly valuable in scenarios with challenging terrains.
- Exploration Capability: The Ant explores a more extensive part of the map compared to the Humanoid. This suggests a higher degree of adaptability and effectiveness in maneuvering through complex landscapes.

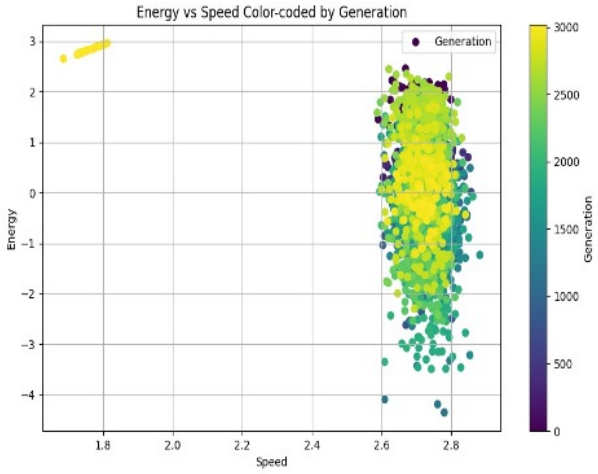

Figure 2: **Speed vs Energy Optimization for Humanoid**

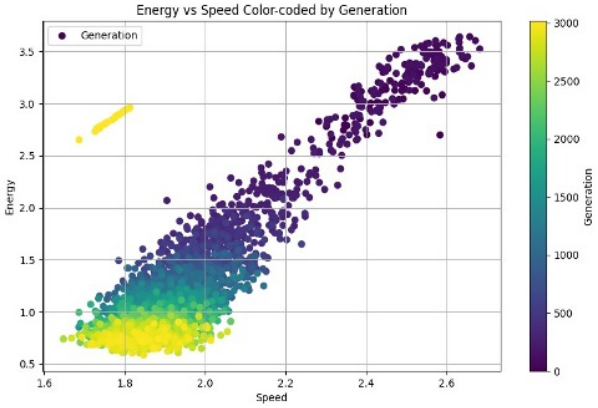

Figure 3: **Speed vs Energy Optimization for Ant**

4.2 Implications for Future Work:

The observed performance differences between the Ant and Humanoid agents in complex terrains lay the groundwork for future investigations. Further fine-tuning and exploration of algorithmic enhancements for the Humanoid may be warranted to bridge the performance gap and enhance its adaptability in challenging environments. In conclusion, the Ant's superior performance in complex terrains underscores its potential as a more optimized choice for scenarios requiring efficient traversal across diverse and challenging landscapes.

References

[1] Xu, Jie & Tian, Yunsheng & Ma, Pingchuan & Rus, Daniela & Sueda, Shinjiro& Matusik, Wojciech. (2020). Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control.

[2] Abbyasov, Bulat & Lavrenov, Roman & Zakiev, Aufar & Yakovlev, Konstantin & Svinin, Mikhail & Magid, Evgeni. (2020). Automatic tool for Gazebo world construction: from a grayscale image to a 3D solid model. 10.1109/ICRA40945.2020.9196621.

[3] Zhao, Allan & Xu, Jie & Konaković-Luković, Mina & Hughes, Josephine & Spielberg, Andrew & Rus, Daniela & Matusik, Wojciech. (2020). RoboGrammar: graph grammar for terrain-optimized robot design. ACM Transactions on Graphics. 39. 1-16. 10.1145/3414685.3417831.

Contributions

Table 1: Individual Contributions

Name	Work	Percentage of Work
Arpit Chandrakar	Generated Custom Terrain	24%
Aadithya Velayutham	Implemented Algorithms	24%
Vineeth Veligeti	Explored Multi-Objective Optimization	24%
Shravan Nermati	Progress Presentation	4%
Aditya Dhavala	Trained Robots and Rendered Videos	24%

Code Repository

The link for the code repository is mentioned below.

`https://drive.google.com/drive/folders/1gkNMlO8diKeJl4k3ifV5kFoLgs-YhzUx?usp=drive_link`