Максименко Андрій Олександрович
доктор філософії, старший викладач
Національний університет «Одеська політехніка», Україна

ПЕРЕНЕСЕННЯ ОПТИМІЗОВАНИХ ПАРАМЕТРІВ НЕЧІТКОГО РЕГУЛЯТОРА РОЮ МІЖ СЦЕНАРІЯМИ СЕРЕДОВИЩА

Постановка проблеми. Налаштування нечіткого регулятора рою гетерогенних мобільних платформ генетичним алгоритмом обчислювально дороге: кожне оцінювання пристосованості потребує серії повних імітаційних прогонів місії, а весь цикл оптимізації — годин машинного часу. При цьому оптимізація прив'язана до конкретного сценарію, тому відкритим лишається питання: отриманий регулятор є загальною стратегією керування чи переобученим під навчальну карту розв'язком.

Аналіз досліджень і публікацій. Поєднання нечіткої логіки з еволюційною оптимізацією для мобільних роботів розглянуто в [1], огляд [3] систематизує підходи до ройової робототехніки. Узагальнювальну здатність навчених політик опрацьовано в навчанні з підкріпленням: у [2] показано втрату якості агента на згенерованих наново середовищах, у [4] — досвід перенесення політик між симуляцією та реальними платформами. Для нечітких регуляторів рою з імовірнісною метрикою виконання множини цілей кількісної оцінки перенесення не наведено.

Мета роботи — кількісно оцінити, яка частка виграшу від генетичної оптимізації нечіткого регулятора рою зберігається при перенесенні знайдених параметрів на інший, суттєво складніший сценарій середовища.

Виклад основного матеріалу. Навчальний сценарій A містить сім цілей і рій із двадцяти платформ; цільовий сценарій B — вісімнадцять цілей, двадцять вісім зон інтересу, три коридори загроз, що перекривають прямі шляхи до кластерів цілей, і рій із тридцяти платформ чотирьох типів. Вектор налаштування містить 189 параметрів: функції належності вхідних змінних та індекси вихідних термів тактичних і стратегічних правил.

Порівнюються три набори параметрів (рис. 1): базовий, сформований експертно без навчання; перенесений, отриманий оптимізацією на сценарії A; власний, отриманий оптимізацією на самому сценарії B. Усі три оцінено на єдиному стенді сценарію B за тридцятьма прогонами Монте-Карло.

Рис. 1. **Структурна схема експерименту з перенесення параметрів**

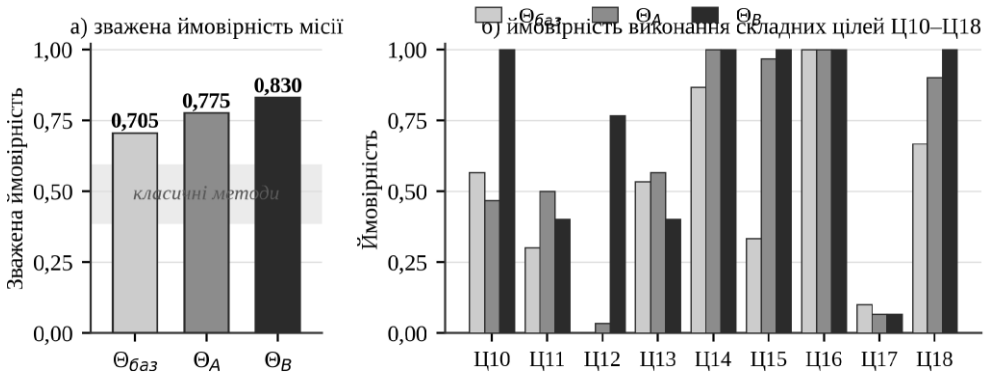

Рис. 2. **Результати оцінювання трьох наборів параметрів на сценарії В**

Базовий набір забезпечує зважену ймовірність виконання місії 0,705, перенесений — 0,775, власний — 0,830 (рис. 2, а). Отже, перенесення дає 0,071 із 0,125 повного приросту, тобто зберігає 56,6 % виграшу, доступного за оптимізації в цільовому середовищі; усі три конфігурації лишаються істотно вище смуги класичних методів, реалізованих на тому самому стенді.

Структура виграшу неоднорідна. Базовий набір залишає одну ціль із нульовою ймовірністю виконання, тоді як обидва оптимізовані набори не мають жодної такої: складова «усунення повних відмов» переноситься повністю, а «пікова якість» — лише частково (цілей зі стовідсотковою ймовірністю 10, 11 та 14).

По-цільовий розподіл (рис. 2, б) показує, що дев’ять із вісімнадцяти цілей виконуються з одиничною ймовірністю за будь-якої конфігурації; усі відмінності зосереджені у восьми складних цілях за коридорами загроз. Перенесений набір дає приріст на Ц15 (з 0,333 до 0,967), Ц18 (з 0,667 до 0,900) і Ц11, але поступається базовому на Ц10. Власний набір «розв’язує» Ц10 та Ц12, підвищуючи ймовірність останньої з 0,033 до 0,767, проте поступається

перенесеному на Ц11 і Ц13. Жодна конфігурація не домінує за всіма цілями одночасно — це компроміс у розподілі обмеженого ресурсу рою. Обчислювальної ціни на етапі виконання перенесення не має: час тридцяти прогонів для нечітких конфігурацій становив 13,7–15,4 с проти 19,7–32,0 с для класичних алгоритмів.

Висновки. Перенесений набір зберігає 56,6 % приросту зваженої ймовірності виконання місії, доступного за оптимізації в цільовому середовищі. Складові виграшу переносяться нерівномірно: усунення цілей із нульовою ймовірністю — повністю, приріст кількості повністю виконуваних цілей — частково. Оптимізація змінює результат лише для складної підмножини цілей за коридорами загроз, і жодна конфігурація не є домінантною, що вказує на перерозподіл ресурсу рою як основний механізм дії налаштованих правил. Практичний наслідок: за браку ресурсу на повторну оптимізацію доцільно застосовувати раніше знайдений набір — він усуває повні відмови й дає понад половину потенційного приросту. Подальші дослідження спрямовано на формування навчальної множини з кількох сценаріїв.

Список використаних джерел:

1. Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., & Bouzouia, B. (2017). Optimal path planning and real-time obstacle avoidance using adaptive fuzzy logic and genetic algorithms. Robotics and Autonomous Systems, 89, 95–109.
2. Cobbe, K., Hesse, C., Hilton, J., & Schulman, J. (2020). Leveraging procedural generation to benchmark reinforcement learning. Proceedings of the 37th International Conference on Machine Learning, 119, 2048–2056.
3. Nguyen, L. V., & Nguyen, H. L. (2024). Swarm intelligence-based multi-robotics: A comprehensive review. AppliedMath, 4(4), 1192–1210.
4. Zhao, W., Queralta, J. P., & Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: A survey. IEEE Symposium Series on Computational Intelligence (SSCI), 737–744.