

Применение двухэтапного генетического программирования для построения автомата, управляющего моделью танка в игре Robocode

Соколов Д.О.
научный руководитель: Шалыто А. А.

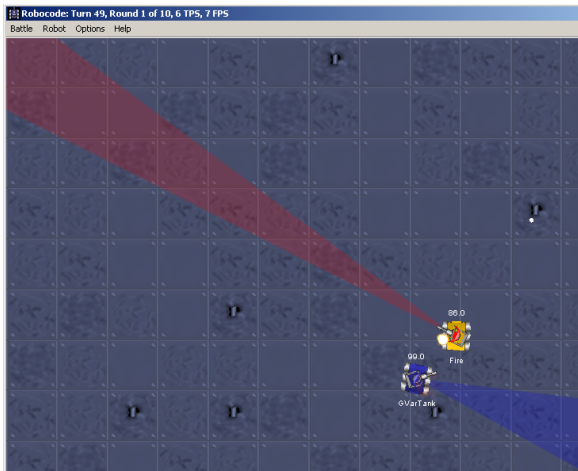
май 2009

Объект управления характеризуется тройкой (X, Z, F) , где:

- X — множество входных воздействий;
- Z — множество выходных воздействий;
- F — функция $X^n \rightarrow Z^m$, которая по заданным входным воздействиям выдает *реакцию* (т.е. множество выходных воздействий) управляемого объекта.

Цель — автоматическое построение по заданным X, Z функции F , удовлетворяющей критериям оптимальности.

Пример соревнования в игре *Robocode*



X — набор из 11 вещественных переменных, определяющих позицию на поле.

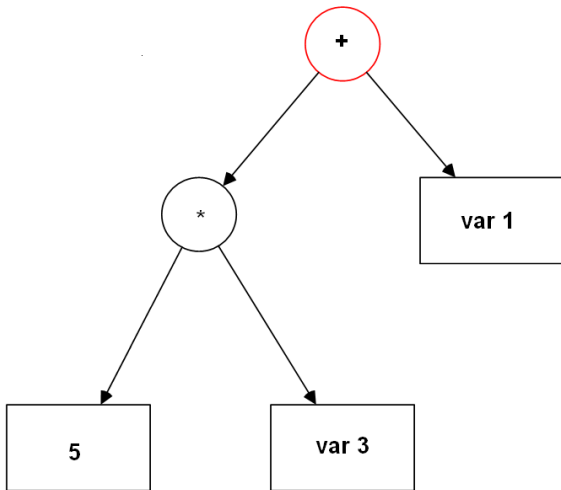
В каждый момент времени модель танка выполняет 4 действия:

- поворот корпуса на заданный угол;
- поворот пушки на заданный угол;
- движение вперед на заданное расстояние;
- выстрел с заданной *силой*.

Осталось только определить параметр передаваемый действиям.
Таким образом $F : R^{11} \rightarrow R^4$

- В качестве управляющей структуры используем конечный автомат.
- В каждом состоянии поместим *обработчик вещественных переменных*. Виды обработчиков:
 - нейронная сеть;
 - четверка деревьев разбора.

Пример дерева разбора



Требование оптимальности к функции F определим, как максимизация функции: $\frac{score}{score+enemy.score}$, где $score$ — число очков, набранное моделью танка в течении соревнования. Для максимизации данной функции используем *островной* генетический алгоритм.

- Оператор отбора — элитизм.
- Оператор *однородного* кроссовера для нейронных сетей.
- Оператор *одноточечного* кроссовера для нейронных сетей.
- Оператор мутации для нейронных сетей и деревьев разбора.
- Оператор кроссовера для деревьев разбора.

Исходная задача разбивается на две подзадачи:

- 1 Построение *обработчиков вещественных переменных* для конкретного состояния.
- 2 Построение графа переходов автомата.

Каждая из подзадач решается на своем этапе генетического алгоритма.

- 1 На первом этапе создается репозиторий, из обработчиков вещественных переменных.
- 2 Из готового репозитория строится конечный автомат. (На данном этапе применяются модифицированные для работы с репозиторием генетические операторы).

Пространство решений (1)

Размер пространства решений для: конечных автоматов с нейронными сетями, конечных автоматов с деревьями разбора без и в случае разбиения задачи на части, можно оценить соответственно:

- $|\Omega_{net}| = (T^V 2^{64E_1})^N$
- $|\Omega_{tree}| = (Z^E)^{V_c} (Z^E)^{4N} N^{2^{V_c}}$
- $|\Omega_{treenew}| = (Z^E)^{V_c} S^4 N^{2^{V_c}}$

Где:

- E_1 — число ребер в нейронной сети.
- V — число нейронов.
- N — число состояний автомата.

- 2^{V_c} — число переходов в автомате.
- S — размер репозитория.
- Z — математическое ожидание числа вершин в дереве разбора.
- T — число типов нейронов.
- E_1 — число типов вершин в дереве разбора.

Очевидно соотношение: $|\Omega_{tree_{new}}| \ll |\Omega_{tree}| \ll |\Omega_{net}|$

- Лучшие результаты, из известных, автоматически построенных, решений данной задачи в соревнованиях на текущий момент.
- Скорость построения на втором этапе в несколько раз быстрее, по сравнению с известными решениями. Составляет примерно 1.5 часа.
- Возможность адаптации предложенных методов для решения множества задач.

- Большие временные затраты на первом этапе алгоритма (порядка 8 суток).
- Невозможность построить автомат против сильных соперников.

- Дальнейшее дробление задачи на части.
- Уменьшение пространства решений за счет анализа главных компонент входных переменных.

Эволюционный процесс (деревья разбора vs. sample.Tracker)

