

ГЕНЕРАЦИЯ КОНЕЧНЫХ АВТОМАТОВ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ О ПОИСКЕ ЦЕЛИ СЕНСОРНЫМ АГЕНТОМ В ОБЛАСТИ С ПРЕПЯТСТВИЯМИ

А. А. Соколов

студент кафедры компьютерных технологий;
ansokolmail@gmail.com

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Аннотация: В данной работе рассматривается возможность применения генетического алгоритма для генерации конечных автоматов, которые можно использовать для решения простейшего варианта задачи навигации.

Введение

Автоматное программирование — это парадигма программирования, в которой поведение программы может быть представлено в виде системы управляющих конечных автоматов [5]. Автоматный подход применяется в событийно-ориентированных приложениях. Множество ответственных технических систем являются событийно-ориентированными, а управление в них является автоматным.

Задачи навигации возникают в различных отраслях современной науки. Так, на практике часто возникают задачи о перемещении робота из одной точки в другую в различном пространстве конфигураций — от простейшего случая передвижения робота на плоскости среди препятствий до поиска оптимальной стратегии перемещения в пространстве робота-манипулятора со множеством степеней свободы.

Среди всего множества задач навигации обычно рассматривают некоторые крайние случаи, например, обладание полной информацией о местности, или, наоборот, отсутствие возможности запоминать что-либо, кроме нескольких чисел. В реальности эти случаи комбинируются, и системе управления роботом необходимо использовать доступную ему информацию наиболее эффективным способом, одновременно корректно обрабатывая нештатные ситуации. Сложность требований, предъявляемых к системе управления мобильным роботом, очень высока. Использование автоматного программирования, по крайней мере, в некоторых частях системы управления позволит уменьшить число ошибок, что приведет к повышению надежности системы. По этой причине необходимо исследовать возможность реализации алгоритмов, решающих задачи навигации, автоматными методами. Принимая

в расчет сложность этих алгоритмов, авторы данного исследования считают целесообразным исследовать автоматическое построение автоматов, решающих некоторые задачи навигации.

Рассмотрим следующую задачу навигации. Имеется некоторая двумерная область с препятствиями. Препятствия могут быть произвольной формы, однако никакие два препятствия не должны иметь общих точек. Кроме того, любой круг конечного радиуса пересекает конечное число препятствий, и любое препятствие покрывается кругом конечного радиуса.

В области, удовлетворяющей описанным требованиям, дана точка — *цель*. Агент в начальный момент времени находится в произвольной точке области, не принадлежащей ни одному препятствию. Задача агента заключается в том, чтобы добраться до цели или, если она недостижима, сообщить об этом. При этом агент знает свои координаты и координаты цели. Агент не знает, где именно находятся препятствия, но может определить, если ли препятствия в непосредственной с ним близости. Агент обладает $O(1)$ дополнительной памяти, по этой причине он не может запоминать уже посещенную часть области, но может хранить некоторый заранее определенный объем информации (например, координаты некоторого числа точек).

Известны алгоритмы, решающие данную задачу. К ним относятся алгоритмы семейства *Bug*. Общая идея этих алгоритмов такова: пока агент может двигаться по направлению к цели, он это делает. Если агент находит препятствие, которое не дает ему двигаться, он начинает обход этого препятствия. Обход прекращается и продолжается движение к цели, когда выполнено какое-либо условие, специфичное для конкретного алгоритма. В процессе этого обхода алгоритм также может прийти к выводу, что цель недостижима.

Первые алгоритмы семейства *Bug* (*Bug1*, *Bug2*) были впервые описаны в работе [1]. Известными алгоритмами этого семейства также являются *Distant Bug* и *Tangent Bug* [2–4].

Постановка задачи

В данной работе исследуется возможность автоматического нахождения автоматной реализации алгоритмов, решающих поставленную задачу. Однако, при использовании автоматов более естественной является постановка задачи с конечным числом дискретных входных воздействий, поэтому авторы работы полагают, что для упрощения работы имеет смысл рассмотреть вариацию данной задачи, при которой возможно описать такие входные воздействия.

Областью, на которой действует агент, будем считать бесконечное клетчатое поле. Некоторые клетки этой области заняты препятствиями, причем таких клеток конечное число. Цель находится в одной из клеток, не занятых препятствием. Агент также занимает одну клетку и может перемещаться только в клетки, смежные по стороне с текущей.

Агент управляется конечным автоматом — входные воздействия для этого автомата формируются на основе состояния агента и наблюдаемого им состояния среды, выходные воздействия автомата управляют движением агента. Цель работы — вырастить конечный автомат с помощью генетического алгоритма [6], который бы управлял агентом таким образом, чтобы, если цель достижима из начального положения агента, агент за конечное число шагов достиг бы этой цели и, находясь там, сообщил бы о завершении работы. Если же цель недостижима, то агент за конечное время должен это определить и завершить работу, сообщив о том, что цель недостижима.

Объект управления

Агент, являющийся объектом управления, находится на поле, занимая одну клетку. Помимо координат на поле, у агента также есть ориентация — в каждый момент времени он направлен в одну из сторон («север», «запад», «юг», «восток»). Внешне наблюдаемые действия агента ограничиваются следующими:

- перейти «вперед» (в ту сторону, в которую он направлен) на соседнюю клетку;
- повернуться по часовой стрелке;
- повернуться против часовой стрелки;
- закончить работу при достижении цели;
- закончить работу при выяснении, что цель недостижима.

По причинам того, что столкновения с препятствиями «запрещены», для анализа окружающей его среды агент должен уметь непосредственно узнавать, есть ли препятствия в некоторых клетках в его непосредственной окрестности. В данной работе агент способен узнать, есть ли препятствие непосредственно перед ним.

Перечислим все величины, хранящиеся в агенте (включая константные величины и величины, получаемые «от датчиков»):

- абсцисса и ордината цели;
- абсцисса, ордината и ориентация текущего положения агента;
- абсцисса, ордината и ориентация сохраненного положения агента;
- есть ли препятствие в клетке, находящейся непосредственно перед агентом.

Первый, второй и четвертый набор величин в совокупности образуют наблюдаемое агентом состояние среды. Третий же набор образует вычислительное состояние агента.

Так же в работе рассматривается задача с расширенным набором данных, которые агент может дополнительно получить от датчиков следующие величины:

- есть ли препятствие в клетке, находящейся слева от агента;
- есть ли препятствие в клетке находящейся справа от агента.

Система управления

В данной работе система управления агентом представлена конечным автоматом (автоматом Мили). Входные воздействия автомата, непосредственно получающиеся из условий задачи, приведены в табл. 1. Авторы работы также рассмотрели расширенный набор, дополнительно включающий следующие входные воздействия (табл. 2).

Т а б л и ц а 1

Входные воздействия

Метка	Название в коде	Описание
x1	CAN_MOVE_FORWARD	Отсутствует ли перед агентом препятствие
x2	IS_MOVE_FORWARD_COOL	Приблизится ли агент к цели, если сделает шаг вперед
x3	IS_AT_FINISH	Находится ли агент на финише
x4	IS_AT_SAVED	Находится ли агент в сохраненной точке
x5	IS_BETTER_THAN_SAVED	Ближе ли текущая клетка к финишу, чем сохраненная

Т а б л и ц а 2

Расширение набора входных воздействий

Метка	Название в коде	Описание
x6	CAN_MOVE_LEFT	Отсутствует ли слева от агента препятствие
x7	IS_MOVE_LEFT_COOL	Приблизится ли агент к цели, если перейдет на клетку соседнюю слева
x8	CAN_MOVE_RIGHT	Отсутствует ли справа от агента препятствие
x9	IS_MOVE_RIGHT_COOL	Приблизится ли агент к цели, если перейдет на клетку соседнюю справа

Выходные воздействия автомата приведены в табл. 3.

Т а б л и ц а 3

Выходные воздействия

Метка	Название в коде	Описание
z1	DO_NOTHING	Ничего не делать
z2	MOVE_FORWARD	Сделать шаг «вперед»
z3	ROTATE_POSITIVE	Повернуться по часовой стрелке
z4	ROTATE_NEGATIVE	Повернуться против часовой стрелки
z5	SAVE_POSITION	Сохранить текущее расположение
z6	REPORT_REACHED	Выдать сообщение о прибытии на финиш
z7	REPORT_UNREACHABLE	Цель недостижима

В качестве представления состояний и переходов автомата используются деревья решений [7]. В листьях деревьев решений находятся пары из выходного действия и номера состояния, в которое нужно осуществить переход. Внутренние узлы соответствуют входным воздействиям, у каждого узла две потомка, один для случая, если соответствующее узлу входное воздействие имеет значение true, другой для значения false.

Использовался островной генетический алгоритм — это такой параллельный генетический алгоритм, в котором популяция разбивается на несколько подпопуляций. Для каждой из них параллельно запускается генетический алгоритм. Каждые несколько поколений процессы обмениваются между собой лучшими особями.

Результаты

Для первого набора предикатов произведено 800 запусков генетического алгоритма. В каждом запуске был найден корректный автомат, решающий задачу. Минимальное число поколений, за которое был найден автомат, равнялось 63, максимальное — 82 598. Среднее время вычисления составило 3.75 часа, причем минимальное время вычисления равнялось 1.5 минутам, а максимальное — 47 часам. Для расширенного набора предикатов было проведено 20 запусков. В каждом из запусков также был найден корректный автомат, решающий задачу. Число поколений, необходимых для этого, варьировалось от 4996 до 290 983. Время работы каждого запуска принимало значение от 12 до 80 часов.

Заключение

В ходе выполнения работы было показано, что возможно получать алгоритмы для решения задач поиска пути, реализованные в виде управляющего конечного автомата, с помощью генетического алгоритма. Получены автоматы, которые корректно решают поставленную задачу на любом заданном поле. Таким образом, генетический алгоритм был успешно применен для данной задачи.

Л и т е р а т у р а

1. *Lumelsky V. J., Stepanov A. A.* Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2: 403–430, 1987.
2. *Lumelsky V. J., Skewis T.* Incorporating range sensing in the robot navigation function. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5): 1058–1068, 1990.
3. *Kamon I, Rimon E, Rivlin E.* A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots. CIS — Center of Intelligent Systems 9517, Computer Science Dept., Technion, Israel, 1995.

4. *Liu Y. H., Arimoto S.* Path planning using a tangent graph for mobile robots among polygonal and curved obstacles. *International Journal of Robotic Research*, 11 (4): 376–382, 1992.
 5. *Поликарпова Н. И., Шалыто А. А.* Автоматное программирование. СПб.: Питер, 2010.
 6. *Holland John H.* *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor. 1975
 7. *Данилов В. Р.* Технология генетического программирования для генерации автоматов управления со сложным поведением. СПбГУ ИТМО, 2007. Бакалаврская работа. http://is.ifmo.ru/papers/danilov_bachelor
-