

## ИССЛЕДОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ АГЕНТОМ, ОСНОВАННЫМ НА ЦЕЛИ, С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ

В. ЧЕРНЫШЕВ<sup>1</sup>, Ю. ЧИЖОВ<sup>1</sup>, Г. КУЛЕШОВА<sup>2</sup>, А. БОРИСОВ<sup>2</sup>

<sup>1</sup> *Институт транспорта и связи*  
ул. Ломоносова, 1, Рига, LV-1019, Латвия  
E-mail: neznaju@inbox.lv

<sup>2</sup> *Рижский технический университет*  
ул. Калькю 1, Рига LV – 1658, Латвия  
E-mail: Arkadijs.Borisovs@cs.rtu.lv

Целью работы является исследование свойств алгоритма генетического программирования, а также выполнение сравнительного анализа с нейронной сетью Элмана для задачи управления агентом в клеточном мире. Для решения задачи разработаны два программных продукта: первый реализует алгоритм генетического программирования, второй – нейронную сеть Элмана. Все программные продукты реализованы на языке Object-Pascal в среде Borland Delphi.

В данной работе каждый метод исследован отдельно, затем произведен их сравнительный анализ.

**Ключевые слова:** генетический алгоритм, генетическое программирование, нейронная сеть Элмана, клеточный мир, навигация агента

### 1. Введение

*Клеточный мир.* Клеточный мир представляет собой двумерное прямоугольное поле (рис. 1), поделённое на клетки [3]. Клетка может быть свободна или занята препятствием. Задача агента – обойти заданное препятствие по периметру. Агент имеет восемь сенсоров: N-«север», S-«юг», E-«восток», W-«запад», NW-«северо-запад», NE-«северо-восток», SW-«юго-запад», SE-«юго-восток». Сенсоры позволяют агенту «видеть» препятствия вокруг себя на расстоянии одной клетки. Агент может перемещаться в четырёх направлениях: GN-«на север», GS-«на юг», GW-«на запад», GE-«на восток». Агент не может перемещаться по клеткам с препятствиями (на рисунке агент обозначен буквой «А»).

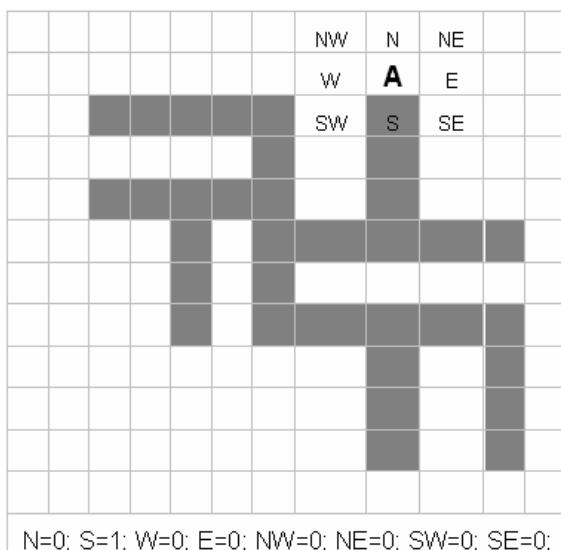


Рис. 1. Пример клеточного мира агента

Идею генетического программирования (ГП) впервые предложил Джон Коца в 1992 году, опираясь на концепцию генетических алгоритмов (ГА). Эта идея заключается в том, что, в отличие от ГА, в ГП все операции производятся не над строками, а над деревьями. При этом используются такие же операторы, как и в ГА: селекция, скрещивание и мутация. Поэтому, чтобы лучше понять алгоритм генетического программирования, необходимо описать генетические алгоритмы.

## 2. Генетические алгоритмы

ГА работают с совокупностью "особей" – популяцией, каждая из которых представляет возможное решение данной проблемы [1,2]. Каждая особь оценивается мерой ее "приспособленности" согласно тому, насколько "хорошо" соответствующее ей решение задачи. В природе это эквивалентно оценке того, насколько эффективен организм в выживании и конкурентной борьбе за ресурсы. Наиболее приспособленные особи получают возможность "воспроизводить" потомство при помощи механизмов скрещивания с другими особями популяции. Это приводит к появлению новых особей, которые сочетают в себе некоторые характеристики, наследуемые ими от родителей. Наименее приспособленные особи с меньшей вероятностью смогут воспроизвести потомков, так что те свойства, которыми они обладали, будут постепенно исчезать из популяции в процессе эволюции. Иногда происходят мутации, или спонтанные изменения в генах.

Преимущество ГА состоит в том, что он находит приблизительные оптимальные решения за относительно короткое время.

В общем случае для работы ГА необходимо сформировать следующие компоненты:

- хромосома (решение рассматриваемой проблемы), состоящая из генов;
- начальная популяция хромосом;
- набор операций селекции, скрещивания и мутации для генерации новых решений из предыдущей популяции;
- целевая функция для оценки приспособленности (fitness) решений;
- критерий (признак) останова.

Таким образом, чтобы применить ГА к задаче, сначала следует выбрать метод кодирования решений в виде строки. Пример хромосомы в ГА: 000110110.

## 3. Особенности генетического программирования

В ГП хромосомами являются программы. Программы представлены в виде деревьев с функциональными (промежуточными) и терминальными (конечными) элементами. Терминальными элементами являются константы, действия и функции без аргументов, функциональными – функции, использующие аргументы.

В качестве примера рассмотрим программу, представленную на рис. 2 в виде дерева:

IF (IF (OR (Wall South; NOT (Wall East)); Go West; Go North); Go East; Go North).

Терминальные элементы:

T = {Wall South, Wall East, Go West, Go North, Go East}.

Функциональные элементы:

F = {IF, OR, NOT}.

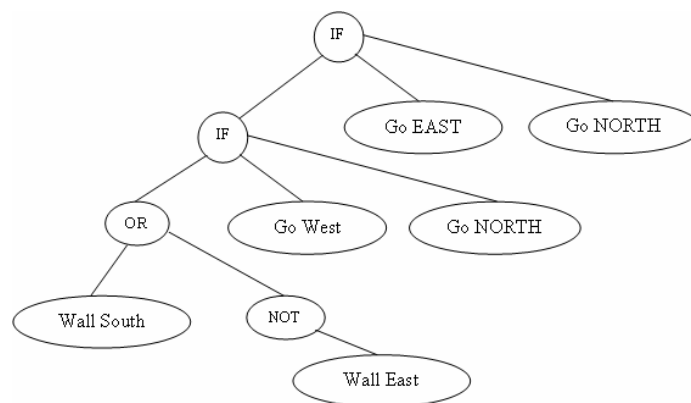


Рис. 2. Древоподобное представление программы

Для того чтобы применить ГП к какой-либо проблеме, помимо компонент ГА, необходимо определить следующие, специфические для ГП компоненты:

- множество терминальных элементов;
- множество функциональных элементов.

Алгоритм работы ГП такой же, как и в ГА: селекция, скрещивание и мутация.

#### 4. Нейронная сеть Элмана

Рекуррентная нейронная сеть Элмана [4] представляет собой синхронную, бинарную двухслойную сеть с обратной связью от выхода к входу скрытого слоя (рис. 3).

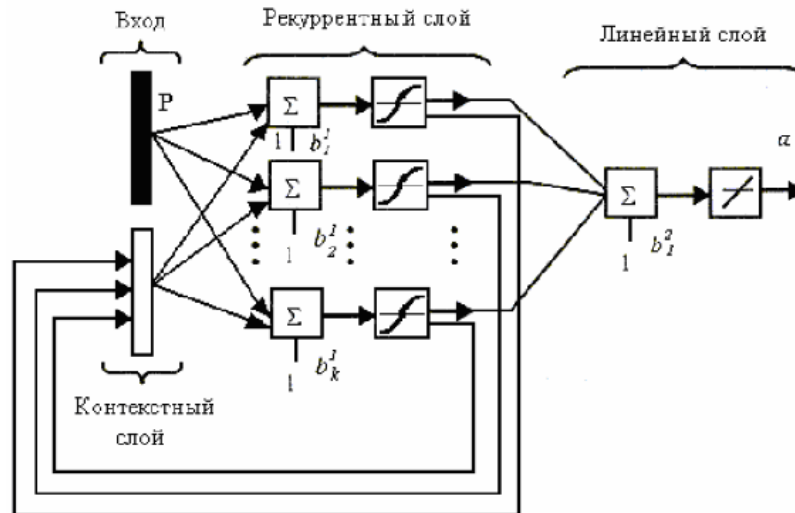


Рис. 3. Структурная схема рекуррентной нейронной сети Элмана

Каждый скрытый нейрон имеет свой аналог в контекстном слое, образующем совместно с внешними входами сети входной слой. В качестве функции активации нейронов скрытого слоя обычно используются сигмоидальные функции. Выходной слой состоит из линейных нейронов, односторонне связанных только с нейронами скрытого слоя. Рекуррентная структура сети Элмана позволяет учитывать непосредственное влияние сигналов в момент  $(n - 1)$  на поведение сети в момент  $n$ .

#### 5. Цель исследования

Основной целью работы является исследование свойств алгоритма генетического программирования, а также сравнительный анализ с нейронной сетью (НС) Элмана для задачи управления агентом в клеточном мире [3, 5]. Для достижения цели необходимо выполнить ряд подзадач, а именно:

- программную реализацию алгоритма ГП;
- программную реализацию нейронной сети Элмана;
- исследование влияния на скорость обучения параметров ГП таких, как размер начальной популяции, глубина деревьев в начальной популяции, вероятность мутации, вероятность скрещивания.

Для исследования применяются оригинальные программные продукты:

- программный продукт, реализующий эволюционный алгоритм ГП, задача которого – изучить влияние количества нейронов в скрытом слое на скорость обучения нейронной сети Элмана;
- программный продукт, реализующий нейронную сеть Элмана.

Основываясь на полученных результатах, проведён сравнительный анализ методов по их скорости решения задачи. В случае ГП – это поиск решения эволюционным методом, в случае нейронной сети Элмана – обучение с целью распознавания пар «состояние – действие».

#### 6. Программная реализация

Программный продукт, реализующий работу алгоритма ГП, позволяет задавать основные параметры алгоритма ГП. Программа также позволяет получать различные данные по каждой итерации как в цифровом, так и в графическом виде (см. рис. 4 и 5).

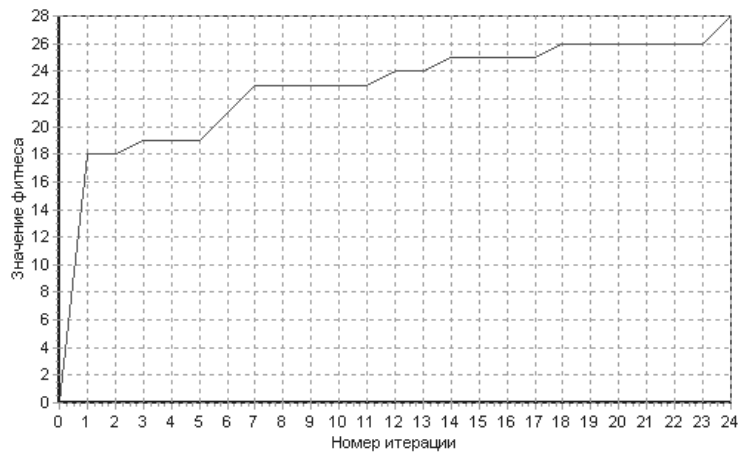


Рис. 4. Размер фитнеса в зависимости от номера итерации (методом ГП)

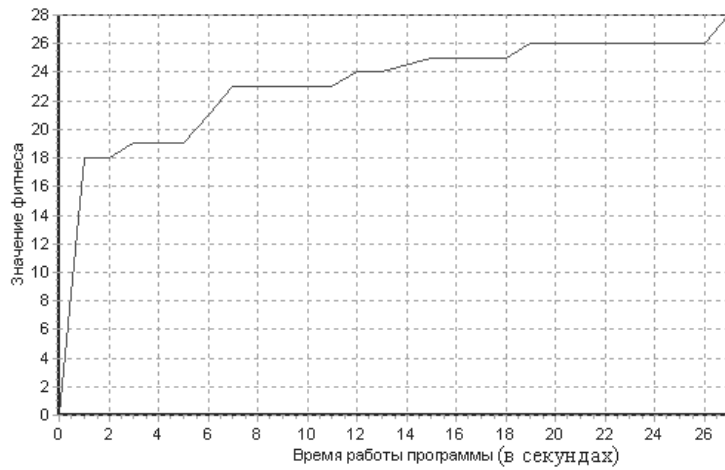


Рис. 5. Размер фитнеса в зависимости от времени работы программы (методом ГП)

Программный продукт, реализующий нейронную сеть, позволяет задавать основные параметры нейронной сети Элмана, а также получать различные данные по каждой итерации как в цифровом, так и в графическом виде (рис. 6 и 7).

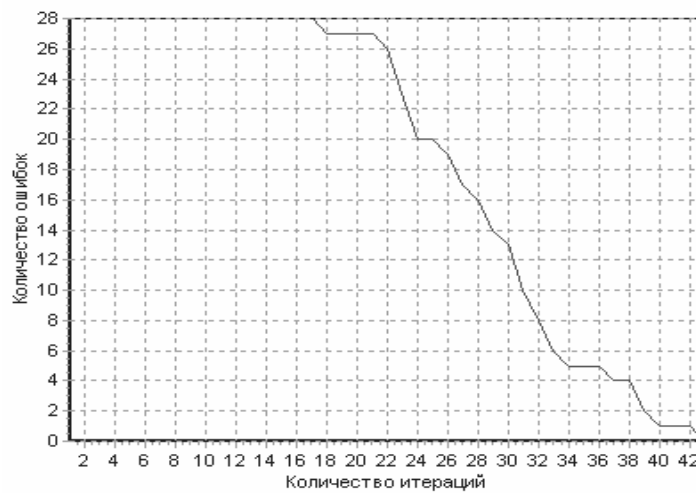


Рис. 6. Количество ошибок в зависимости от количества итераций (методом НС)

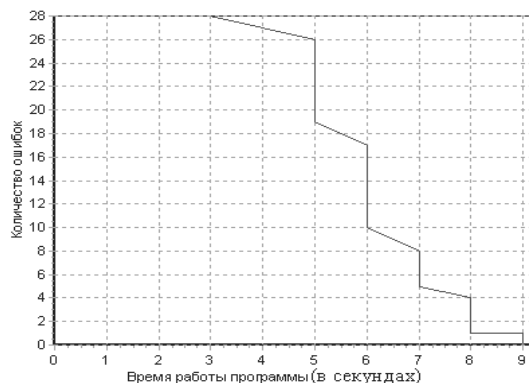


Рис. 7. Количество ошибок в зависимости от времени работы программы (методом НС)

### 7. Полученные результаты

*Клеточные миры.* На рисунках 8, 9, 10 и 11 приведены клеточные миры, в которых производились эксперименты с различными параметрами обоих методов. Первые три мира являются марковскими, а четвёртый мир является немарковским. В последнем присутствуют элементы, образующие такие состояния агента, которые однозначно не определяют дальнейшего направления его движения, другими словами, требуется по-разному реагировать на одни и те же состояния. Примером немарковости может служить элемент «коридора», в котором агенту, согласно решению, необходимо пройти вначале в одном направлении, а затем – в обратном. Так как эксперименты показали, что алгоритм ГП не может обучаться в немарковских средах, то из экспериментов с ГП четвёртый мир был исключён. Время работы программ измеряется в секундах.

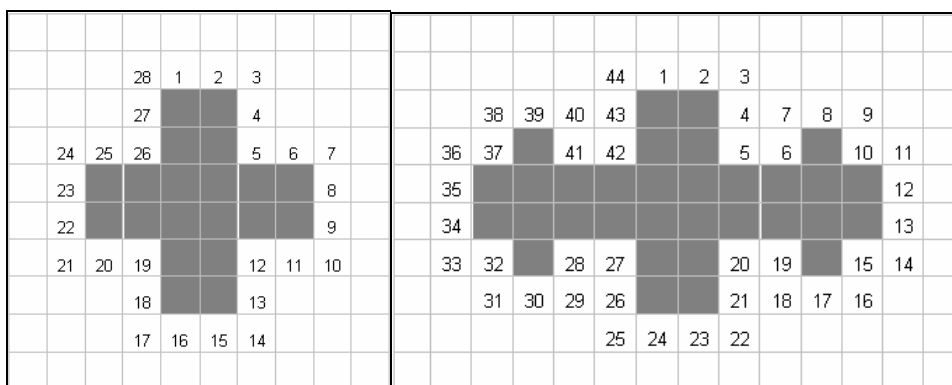


Рис. 8. Первый и второй клеточный мир

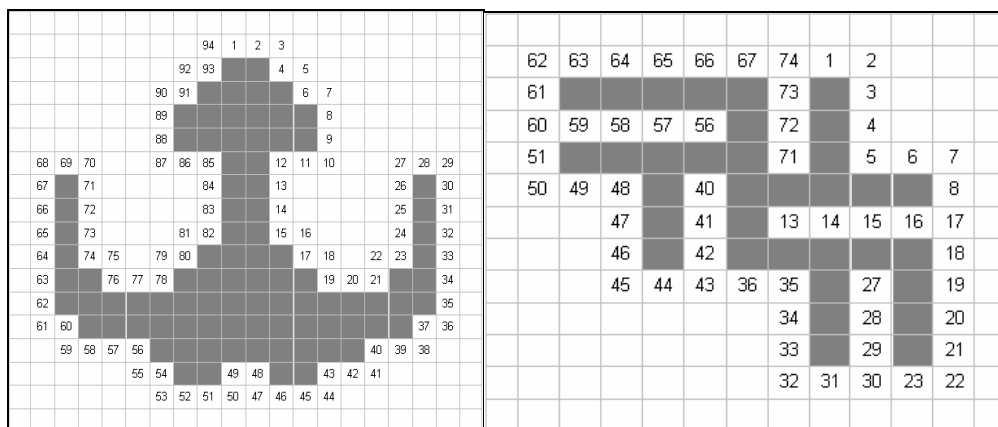


Рис. 9. Третий и четвёртый клеточные миры

Влияние размера популяции на скорость обучения алгоритма ГП. На рис. 10 и 11 приведены графики, объединяющие в себе результаты экспериментов с размером популяции для трёх клеточных миров.

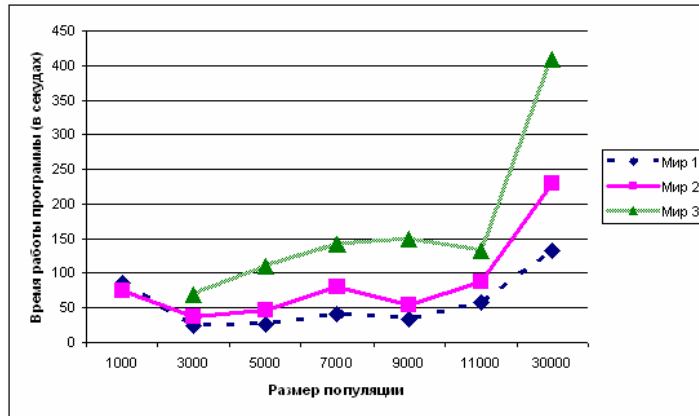


Рис. 10. Время работы программы ГП в зависимости от размера популяции

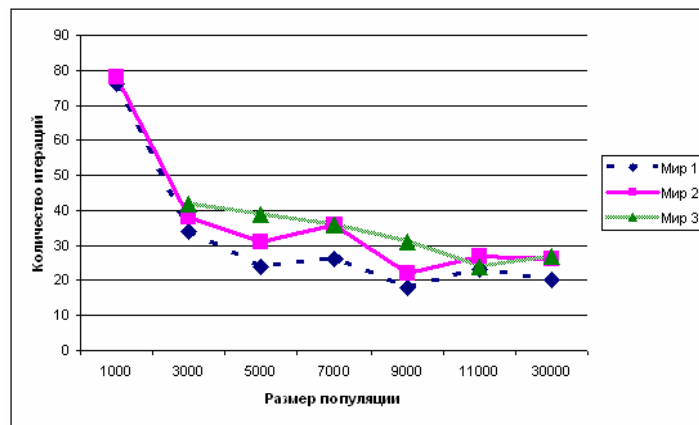


Рис. 11. Количество итераций ГП в зависимости от размера популяции

Из графиков видно, что слишком маленький размер популяции увеличивает время работы алгоритма ГП. Увеличивать размер популяции имеет смысл только в двух случаях, когда не удаётся найти оптимальное решение или когда необходимо снизить количество итераций.

Влияние начальной глубины деревьев на скорость обучения алгоритма ГП. На рисунках 12 и 13 приведены графики, объединяющие в себе результаты экспериментов с начальной глубиной деревьев для трёх клеточных миров.

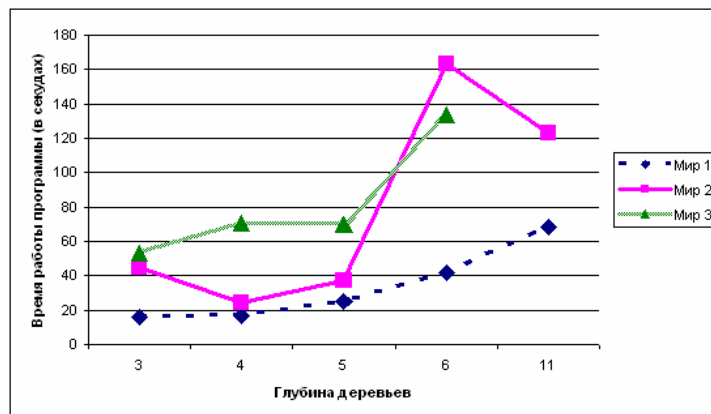


Рис. 12. Время работы программы ГП в зависимости от начальной глубины деревьев

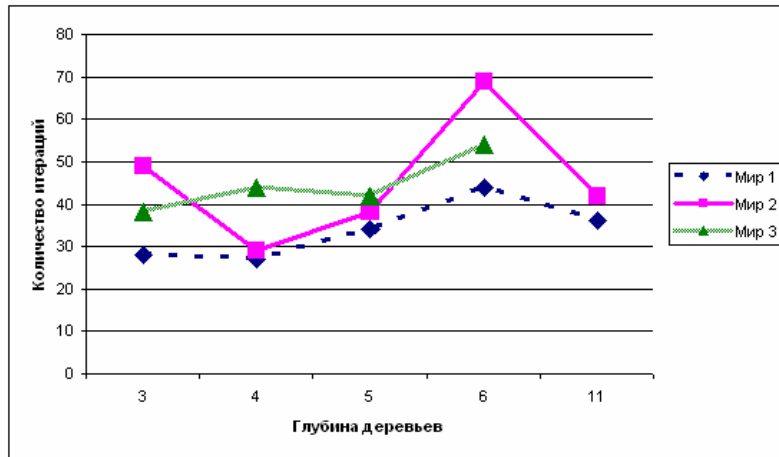


Рис. 13. Количество итераций ГП в зависимости от начальной глубины деревьев

Из графиков видно, что лучше всего использовать минимально допустимую глубину дерева.  
 Влияние вероятности мутации на скорость обучения алгоритма ГП. На рис. 14 и 15 приведены графики, объединяющие в себе результаты экспериментов с вероятностью мутации для трёх клеточных миров.

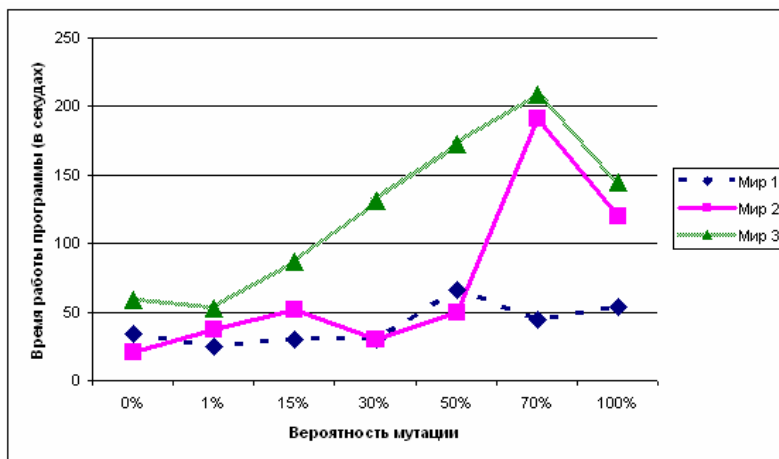


Рис. 14. Время работы программы ГП в зависимости от вероятности мутации

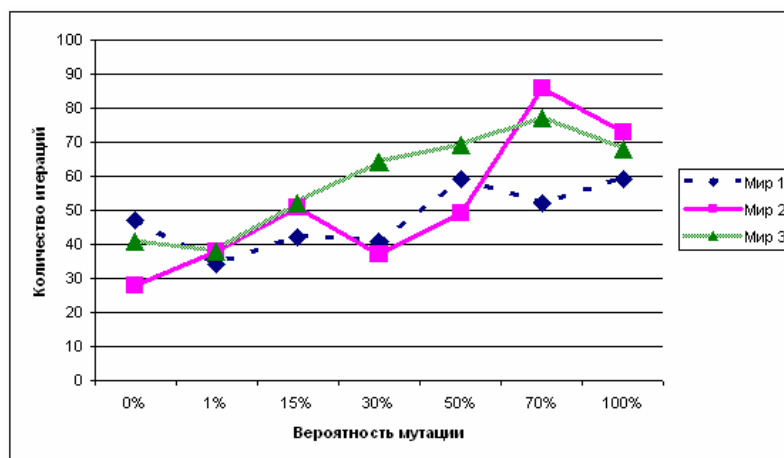


Рис. 15. Количество итераций в зависимости от вероятности мутации

Из графиков видно, что оптимально использовать значение вероятности мутации 0–15%, дальнейшее повышение вероятности мутации ведёт к увеличению времени работы алгоритма ГП.

*Влияние вероятности скрещивания на скорость обучения алгоритма ГП.* На рис. 16 и 17 приведены графики, объединяющие в себе результаты экспериментов с вероятностью скрещивания для трёх клеточных миров.

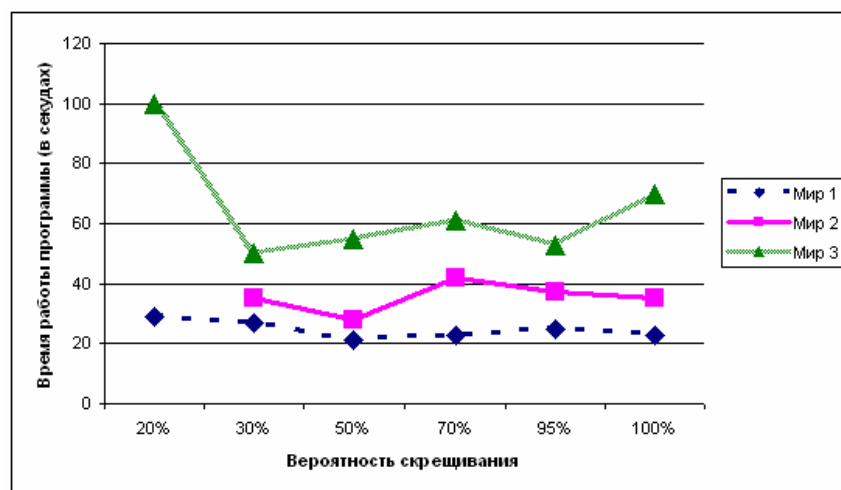


Рис. 16. Время работы программы ГП в зависимости от вероятности скрещивания

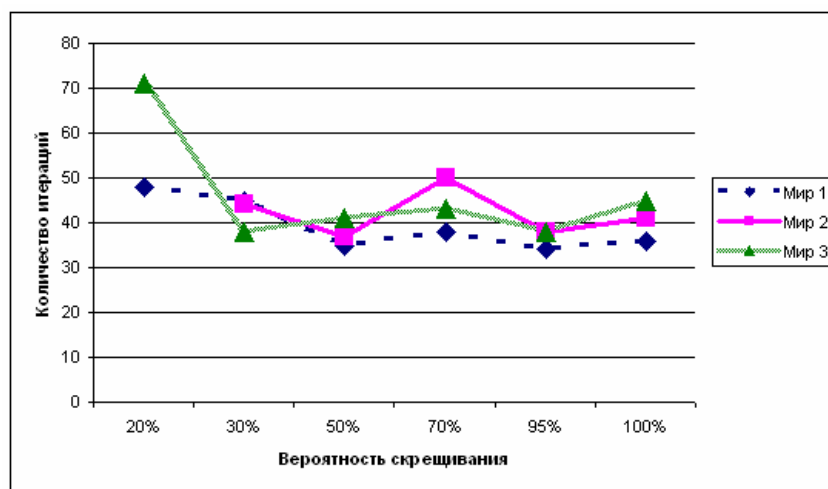


Рис. 17. Количество итераций ГП в зависимости от вероятности скрещивания

Из графиков видно, что время работы программы и количество итераций не очень зависят от вероятности скрещивания. Такие закономерности выявить не удалось: программа одинаково быстро обучалась и с низкой и с высокой вероятностью скрещивания.

*Влияние количества нейронов в скрытом слое на скорость обучения нейронной сети Элмана.* Так как нейронная сеть Элмана имеет «память предыдущих состояний», то она смогла обучаться и в немарковских средах, поэтому в экспериментах участвовал четвёртый клеточный мир. Также были проведены дополнительные эксперименты с пятым клеточным миром, чтобы выяснить, насколько эффективно нейронная сеть Элмана действует в более сложном мире. На рис. 18 приведён пятый клеточный мир. Пятый клеточный мир является немарковским.

На рис. 19 и 20 приведены графики, объединяющие в себе результаты экспериментов с нейронной сетью Элмана для пяти клеточных миров.



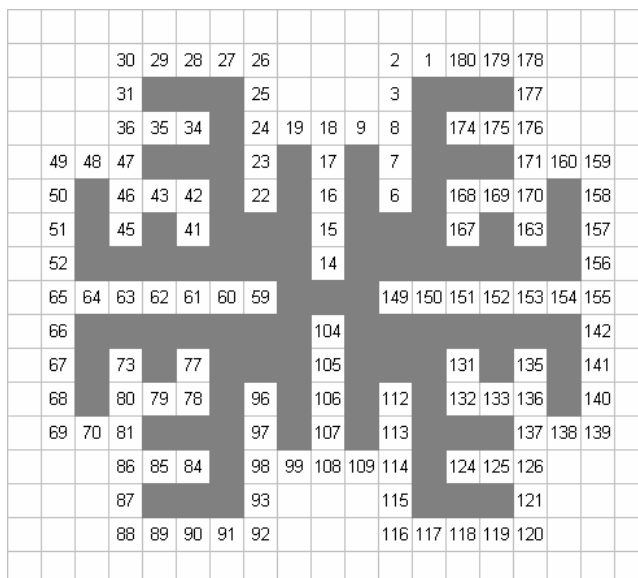


Рис. 18. Пятый клеточный мир

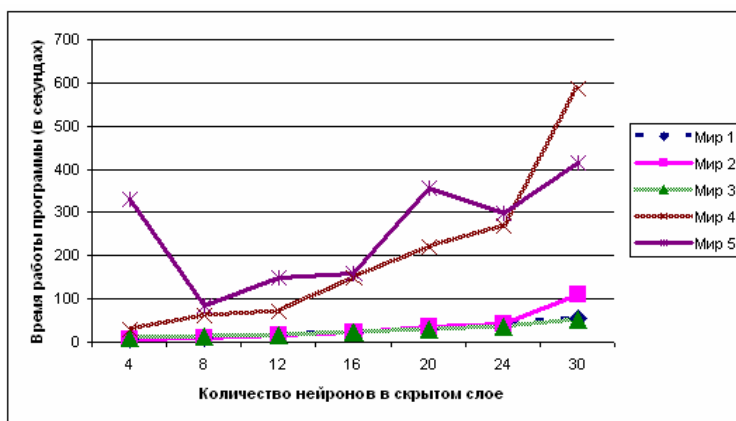


Рис. 19. Время работы программы НС в зависимости от количества нейронов в скрытом слое

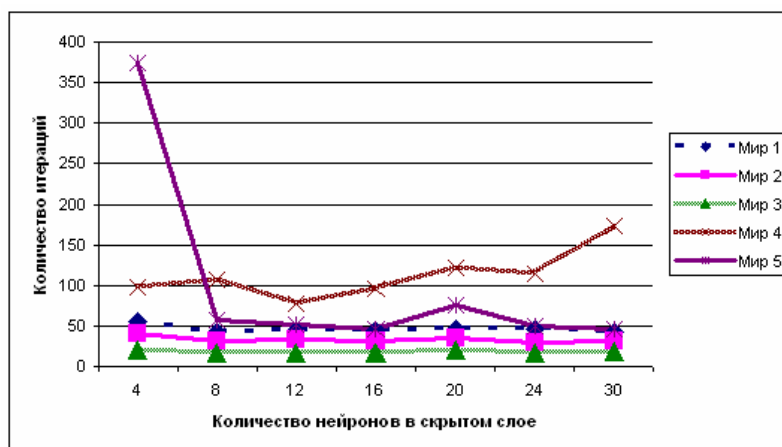


Рис. 20. Количество итераций НС в зависимости от количества нейронов в скрытом слое

Из графиков видно, что для первых трёх миров результаты примерно одинаковы. Все эти миры обладают свойством марковости. Результаты четвёртого мира отличаются, так как данный мир обладает свойством немарковости. Из графиков видно, что время обучения в пятом клеточном мире сравнимо со временем обучения в четвёртом клеточном мире, при этом количество итераций меньше и сравнимо с марковскими средами. Это говорит о том, что сеть Элмана успешно справляется с обучением в более сложных мирах.

Сравнительный анализ алгоритма ГП и нейронной сети Элмана. На рис. 21 и 22 изображены результаты сравнительного анализа методов.

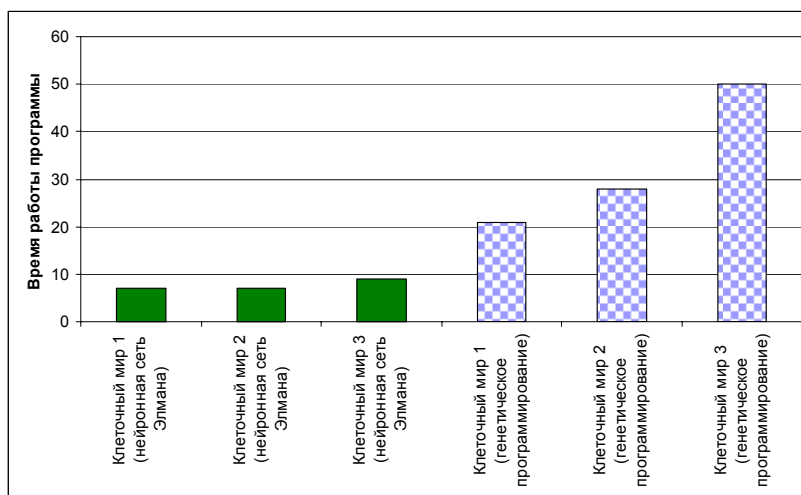


Рис. 21. Время работы программы в зависимости от клеточного мира

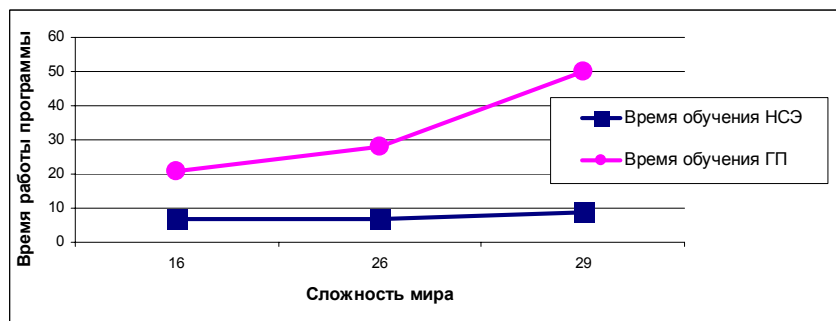


Рис. 22. Время работы программ в зависимости от сложности мира

Из графиков видно, что алгоритм ГП проигрывает нейронной сети Элмана по времени обучения в клеточных мирах, обладающих свойством марковости. Сеть Элмана показала значительно лучшие результаты и затрачивает значительно меньше времени для нахождения решения.

## 8. Анализ результатов

В процессе экспериментирования были решены следующие задачи:

- исследовано влияние параметров алгоритма генетического программирования на скорость обучения алгоритма;
- исследовано влияние количества нейронов в скрытом слое на скорость обучения нейронной сети Элмана;
- проведены дополнительные эксперименты с нейронной сетью Элмана;
- проведён сравнительный анализ алгоритма генетического программирования и нейронной сети Элмана.

Исследование различных параметров алгоритма генетического программирования позволило сделать следующие выводы.

1. **Исследование размера популяции.** Было выявлено, что слишком маленький размер популяции увеличивает время работы алгоритма ГП. Для клеточных миров, используемых при экспериментах, оптимальный размер популяции составил 3000–5000 особей. Увеличивать размер популяции имеет смысл только в двух случаях: если не удаётся обучить алгоритм клеточному миру и если есть необходимость снизить количество итераций.
2. **Исследование максимальной глубины деревьев в начальной популяции.** Проведённые эксперименты показали, что лучше всего использовать минимально возможную глубину дерева, при которой возможно обучить алгоритм ГП. Для алгоритма, использованного в данной работе с применением правил построения деревьев, минимальная и оптимальная максимальная начальная глубина деревьев в популяции составила 3–5 узлов. Дальнейшее увеличение начальной глубины деревьев способствует увеличению времени обучения программы. При глубине дерева в 11 узлов уже не удалось обучить алгоритм ГП второму клеточному миру, а при количестве узлов свыше 11 – и остальным мирам. Это происходит из-за того, что в результате скрещивания и мутации глубина деревьев очень сильно возрастает, что способствует быстрому истощению ресурсов компьютера и остановке программы.
3. **Исследование вероятности мутации.** В ходе экспериментов было выявлено, что оптимальное значение вероятности мутации составляет 0–15%. Дальнейшее повышение вероятности мутации ведёт к увеличению времени работы алгоритма ГП. Для клеточных миров из обучающей выборки оптимальное значение вероятности мутации составило 1%.
4. **Исследование вероятности скрещивания.** Было выявлено, что при вероятности скрещивания ниже 30% не удаётся обучить алгоритм ГП всем мирам. Поэтому использовать вероятность скрещивания ниже 30% не рекомендуется. Для клеточных миров из обучающей выборки оптимальное значение вероятности скрещивания составило 50%.  
Также был выявлен недостаток алгоритма ГП: он не может быть применён в немарковских средах.

В первых трёх мирах (из обучающей выборки) увеличение нейронов в скрытом слое практически не влияет на скорость обучения нейронной сети Элмана. Это обусловлено тем, что эти среды являются марковскими. Результат в четвёртом клеточном мире отличается, так как эта среда немарковская.

При увеличении количества нейронов в скрытом слое время работы программы возрастает. Время работы достигает минимума при количестве нейронов в скрытом слое, равном 4; свыше этого количество итераций начинает возрастать. Это подтвердили и дополнительные эксперименты с более сложным миром при количестве нейронов в скрытом слое выше 8. Различное количество нейронов в скрытом слое, при котором было достигнуто минимальное время, в четвёртом и пятом мирах отличается, так как сложность пятого мира более чем в два раза выше. Дополнительные эксперименты с пятым клеточным миром показали также, что время обучения в этом мире сравнимо со временем обучения в четвёртом клеточном мире, при этом затрачивалось меньше итераций. Это свидетельствует о том, что нейронная сеть Элмана хорошо справляется даже со сложными клеточными мирами, а то, что время обучения достигло минимума при использовании восьми нейронов в скрытом слое, говорит о том, что повышать количество нейронов в скрытом слое необходимо только при усложнении препятствия.

Сравнительный анализ алгоритма генетического программирования и нейронной сети показал, что алгоритм ГП проигрывает нейронной сети Элмана в скорости, а также не может быть применён в немарковских средах. В свою очередь, нейронная сеть Элмана способна обучаться в таких мирах, а с учётом превосходства во времени обучения такая сеть становится более предпочтительной для решения задачи обхода агентом препятствия по периметру в клеточном мире.

### Литература

- [1] Goldberg D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison – Wesley Pub. Company. 412 p.

- [2] *Handbook of Genetic Algorithms* (1991) Ed. by L. Davis. Van Nostrand Reinhold, New York, 332–349.
- [3] Nilson N. J. (1998) *Artificial Intelligence: A New Synthesis.*: Morgan Kaufmann Publishers, Inc., San Mateo, California 513 p.
- [4] Оссовский С. (2002) *Нейронные сети для обработки информации*. Пер. с польского И. Д. Рудинского. М.: Финансы и статистика, Москва. 344 с.
- [5] Ziemke T. (1999) Remembering how to behave: recurrent neural networks for adaptive robot behavior. In: L. Medsker & L. Jain (eds.) *Recurrent Neural Networks: Design and applications*. Boca Raton: CRC Press, 355–389.

*Received on the 12<sup>th</sup> of August 2006*