

Применение генетических алгоритмов для оптимизации адаптивной системы управления мобильного робота на параллельном вычислительном комплексе

Л.В. Земских, Е.К. Самаров, А.А.Жданов, В.В. Бабкова

Работа поддержана РФФИ, проекты № 03-01-00323 и № 04-01-08023-офи

Аннотация

В работе на примере имитационной модели мобильного робота рассматриваются возможности оптимизации его системы автономного адаптивного управления. С помощью генетических алгоритмов определяются оптимальные параметры конструкции робота и оптимальные параметры нейроподобной адаптивной системы управления. Рассматриваются схемы реализации используемых алгоритмов на одном компьютере и на вычислительном кластере, позволяющем организовать процесс параллельных вычислений.

Ключевые слова: *автономное адаптивное управление, мобильные роботы, генетические алгоритмы, оптимизация, параллельные вычисления.*

1. Введение. Постановка задачи

Современную деятельность человека трудно представить без использования различных автоматических и автоматизированных устройств. Такие устройства получают все большее распространение во всех сферах человеческой деятельности – от бытовой техники до сложных систем управления производственными процессами.

Как известно, автоматические и автоматизированные устройства состоят из аппаратной части и системы управления. Традиционно системы управления конструировались на основе математических моделей объектов управления и их поведения в среде функционирования. В последние годы все более широкое распространение получают системы управления, чьи принципы функционирования относятся к области работы со знаниями – к искусственному интеллекту (ИИ). К системам такого рода относятся экспертные системы, искусственные нейронные сети, системы с подкрепляющим обучением, системы на основе нечеткой логики, и т.п. К развиваемым в этой области подходам также относится оригинальный подход, называемый методом *автономного адаптивного управления* (ААУ) [1], развиваемый в отделе имитационных систем Института Системного Программирования РАН.

Методология ААУ позволяет создавать адаптивные системы управления для приложений, в которых создание точных математических моделей объекта управления затруднено или невозможно. Используя входящие в состав системы ААУ подсистемы (блок датчиков, исполнительные органы, систему формирования и распознавания образов, базу знаний, систему моделирования эмоций, блок принятия решений), система ААУ накапливает эмпирически найденные знания о свойствах объекта управления и находит способ управлять им. Подсистемы организованы в виде структуры, общей для всех систем ААУ. Подсистемы могут быть реализованы на основе различных подходов, в том числе и на основе сетей специальных нейроподобных элементов [1], разработанных для систем

ААУ (везде ниже, говоря о нейроподобных элементах или о нейронах, мы будем иметь ввиду именно эти модели нейронов, которые существенно отличаются от так называемых формальных нейронов, используемых в традиционных искусственных нейронных сетях). В настоящее время проблемы, связанные с последним вариантом реализации систем ААУ, являются актуальными для дальнейшего развития метода. В частности, при проектировании прикладных систем ААУ на основе сетей нейроподобных элементов возникают задачи подбора параметров нейроподобных элементов, проектирования топологий их сетей и задача оптимизации полученных конструкций.

Настоящая работа посвящена разработке методов синтеза систем ААУ на основе сетей нейроподобных элементов с помощью применения генетических алгоритмов (ГА).

Предложенные в работе методы синтеза могут быть использованы при построении систем ААУ для широкого класса практических приложений.

Одной из основных проблем, связанных с методологией ААУ, являлось отсутствие четких рекомендаций по синтезу и оптимизации параметров систем управления, разрабатываемых для конкретных прикладных задач. Наличие методики синтеза и оптимизации систем ААУ позволит получать более эффективные системы управления.

Системы ААУ являются новым видом нейроподобных систем управления. Поскольку такого рода системы имитируют работу биологических нервных систем, то исследования в этой области опираются не только на знания теории управления, распознавания, статистики и других необходимых разделов математических наук, но и на знания основ нейрофизиологии, генетики, биологии развития. Концепция систем ААУ опирается на представление о структуре и функциях нервной системы, которые логически вынужденно следуют из тех условий, в которых существуют нервные системы. Согласно концепции ААУ, такая система управления должна одновременно решать несколько сложных задач, а именно: задачу автоматической классификации, распознавания, моделирования эмоций, получения, вывода и представления знаний, принятия решений и некоторые другие. К этим свойствам, которые характерны и для системы ААУ, относятся: адаптивность, обучение и управление в одном процессе, многокритериальность управления, независимость от математических моделей объектов управления, универсальность, пригодность для адаптивного управления разнообразными объектами – от технических и технологических процессов до социальных объектов.

Система ААУ состоит из нескольких функциональных подсистем, ответственных за выполнение каждой из перечисленных выше задач. Именно, это следующие подсистемы: блок датчиков, система формирования и распознавания образов, база знаний и система ее формирования, система принятия решения, блок исполнительных органов (актуаторов).

Все основные подсистемы могут быть реализованы на основе сетей специальных нейроподобных элементов [2].

На сегодняшний день одной из основных нерешенных проблем в рамках методологии ААУ является отсутствие формализованной методики синтеза систем управления для прикладных задач. В виду этого обстоятельства, при синтезе прикладных систем ААУ, используются эвристические соображения и простые математические модели.

Синтез системы ААУ представляет собой задачу большой размерности, решить которую аналитически на данном этапе не удастся. Для ее решения требуется знать точную модель взаимосвязи параметров, определяющих систему, а также критерии, характеризующие успешность ее реализации. Такие характеристики можно получить только в результате наблюдений серии экспериментов по моделированию поведения

системы. В виду отсутствия полной математической модели, связывающей параметры, определяющие систему ААУ с критериями качества ее реализации, задачу синтеза необходимо решать методом подбора параметров. Одним из самых перспективных направлений, реализующим метод подбора параметров, являются генетические алгоритмы.

Задачу синтеза систем ААУ можно рассматривать как задачу, состоящую из двух подзадач:

- задачи подбора оптимальных параметров блока датчиков и блока исполнительных органов (актуаторов) в комплексе, и
- задачи подбора оптимальных параметров подсистем, входящих в состав непосредственно системы управления.

Поскольку генетические алгоритмы требуют больших вычислительных ресурсов, было решено использовать параллельные вычисления на кластере.

Предложенные задачи применены к оптимизации программной модели автономного мобильного робота, имеющего целевую функцию – выработка стереотипов поведения при обходе стандартных случайно расставленных препятствий.

2. Описание конструкции робота

Мобильный робот имеет круглый в плане корпус, реверсивный движитель и рулевое управление, обеспечивающие возможность передвижения робота по плоскости вдоль заранее определенного относительно корпуса направления с постоянной скоростью вперед и назад, а также осуществления при движении вперед и назад поворотов направо и налево по дуге окружности постоянного радиуса R (рис.1).

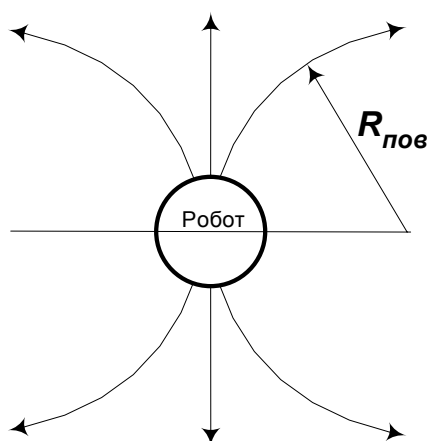


Рис.1. Перемещения, которые может совершать мобильный робот.

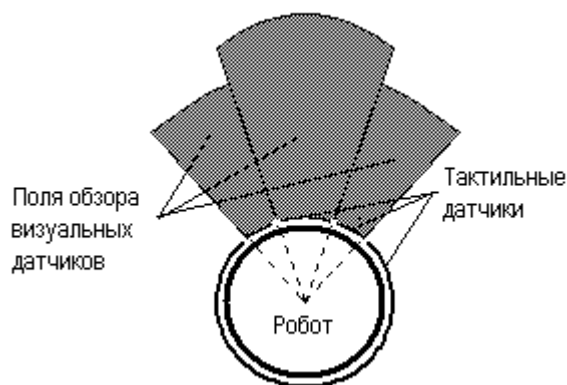


Рис.2. Система датчиков робота.

Робот имеет семь датчиков: три «визуальных» и четыре тактильных (Рис. 2.).

Область обнаружения каждого визуального датчика является частью кругового сектора с центром, совпадающим с центром корпуса робота, и задается угловым раскрытием, а также максимальной и минимальной длиной обнаружения. Ось симметрии области обнаружения центрального визуального датчика совпадает с направлением движения «вперед».

Области обнаружения визуальных датчиков делят окружность основания робота на четыре части, каждая из которых является зоной чувствительности тактильного датчика.

Определяющими характеристиками конструкции робота являются три параметра (скорость робота v , радиус поворота R , угловое раскрытие центрального сектора α), а остальные параметры выражаются через них. Минимальная длина обнаружения каждого визуального датчика совпадает с радиусом корпуса робота r . Внешний радиус области обнаружения каждого визуального датчика совпадает с максимальным расстоянием, на которое смещается робот за один такт движения в соответствующем направлении (рис.3).

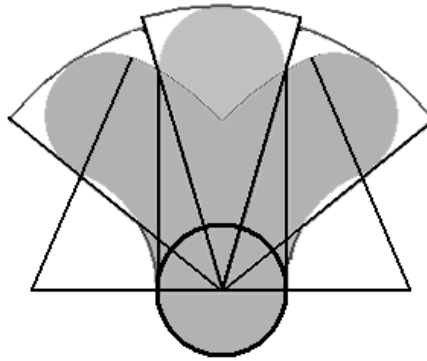


Рис. 3. Относительные размеры полей обзора визуальных датчиков.

Если обозначить внешние радиусы областей обнаружения центрального, правого и левого визуальных датчиков через R_c , R_r , R_l , а угловые раскрытия боковых секторов через α_r , α_l соответственно, то эти параметры будут удовлетворять следующим соотношениям:

$$R_c = v + r,$$

$$R_r = R_l = 2R \sin \frac{v}{2R} + r,$$

$$\alpha_r = \alpha_l = \begin{cases} \frac{v}{2R} + \arcsin \frac{r}{2R \sin \frac{v}{2R}} - \frac{\alpha}{2}, & \text{при } r < 2R \sin \frac{v}{2R}, \\ \pi - \frac{\alpha}{2}, & \text{при } r \geq 2R \sin \frac{v}{2R}. \end{cases}$$

3. Оптимизация конструкции робота

Задача оптимизации конструкции робота состоит в том, чтобы подобрать определяющие параметры (скорость робота v , радиус поворота R , угловое раскрытие центрального сектора α) так, чтобы минимизировать количество ложных срабатываний визуальных датчиков.

Для решения задачи оптимизации конструкции робота используется генетический алгоритм. С этой целью формируется популяция роботов с некоторыми допустимыми значениями параметров (v , R , α). Область допустимых значений параметров задается неравенствами

$$r \leq R,$$

$$0 < v \leq \min\left(v_0, \pi \frac{R}{2}\right),$$

$$0 < \alpha \leq \begin{cases} 2 \min\left(\frac{\pi}{2}, \frac{v}{2R} + \arcsin \frac{r}{2R \sin \frac{v}{2R}}\right) & \text{при } r < 2R \sin \frac{v}{2R}, \\ \pi & \text{при } r \geq 2R \sin \frac{v}{2R}, \end{cases}$$

где через v_0 обозначена максимально возможная скорость движения робота.

Для каждого робота из популяции вычисляется функция приспособленности его конструкции. С помощью функции приспособленности производится отбор роботов, обладающих наилучшими качествами, для производства следующего поколения роботов на основе операций скрещивания и мутации, которые описываются ниже.

Функция приспособленности конструкции робота - это число, характеризующее степень «правдивости» информации о положении препятствий, поступающей в систему управления роботом от визуальных датчиков. Под «правдивостью» понимается согласованность информации об обнаружении препятствия с возможностями системы управления и исполнителей правильно реагировать на эту информацию. Так, например, в условиях «полной правдивости» информации, если левый визуальный датчик в своей зоне видимости не зафиксировал препятствие, то выполнение исполнителем команды системы управления «вперед-налево» не должно приводить к столкновению.

3.1. Вычисление функции приспособленности конструкции робота

В рассматриваемой модели все препятствия выпуклы, поэтому при столкновении в контакте с роботом участвует только одна точка из границы препятствия. Кроме того, поскольку визуальные датчики лишь фиксируют наличие препятствия в зоне обзора, то для определения оптимальных параметров конструкции робота препятствия можно моделировать точками. В случае, когда препятствие попадает сразу в несколько зон обнаружения датчиков, его можно рассматривать как суперпозицию точечных препятствий, расположенных в зоне видимости соответствующих датчиков.

Значение функции приспособленности рассматриваемого робота вычисляется интегрированием по методу Монте-Карло функции, характеризующей согласованность работы датчиков и исполнителей системы управления движением робота.

Зона интегрирования по методу Монте-Карло изображена на рис.3 и представляет собой объединение областей обнаружения визуальных датчиков и следов корпуса робота при движениях «вперед», «вперед-направо» и «вперед-налево». Заметим, что существуют наборы допустимых значений определяющих параметров, и это показано на рис.3, при которых имеются части следов корпуса робота, выходящие за пределы зон обнаружения визуальных датчиков.

Для реализации метода Монте-Карло в зону интегрирования случайным образом вбрасываются точечные препятствия. Затем с одним и тем же положением препятствия поочередно рассматриваются три варианта движения робота «вперед», «вперед-направо» и «вперед-налево».

Для каждого случая вбрасывания точечного препятствия и каждого варианта движения робота вычисляется значение следующей штрафной функции, характеризующей согласованность работы датчиков и исполнителей робота (Таблица 1).

Таблица 1.

	Есть столкновение	Нет столкновений
Препятствие не зарегистрировано визуальными датчиками		
«вперед»	-10	+1
«вперед-направо»	-10	+1
«вперед-налево»	-10	+1
Препятствие зарегистрировано центральным визуальным датчиком		
«вперед»	+1	-1
«вперед-направо»	-1	+1
«вперед-налево»	-1	+1
Препятствие зарегистрировано правым визуальным датчиком		
«вперед»	-1	+1
«вперед-направо»	+1	-1
«вперед-налево»	-1	+1
Препятствие зарегистрировано левым визуальным датчиком		
«вперед»	-1	+1
«вперед-направо»	-1	+1
«вперед-налево»	+1	-1

После этого все три оценки суммируются.

В итоге каждому конкретному случайно выбранному точечному препятствию (x_i, y_i) ставится в соответствие целое число $f(x_i, y_i)$, заключенное в пределах $[-30, +3]$.

Такая процедура вбрасывания точечных препятствий и подсчета значений штрафной функции проводится N (≈ 25000) раз, затем результаты усредняются, а полученное число умножается на площадь S области интегрирования (рис.3). Это и дает значение функции приспособленности конструкции рассматриваемого робота.

Реализация процесса вбрасывания точечных препятствий производится с помощью генерирования двумерной псевдослучайной последовательности Соболя.

Погрешность интегрирования по методу Монте-Карло вычисляется по формуле

$$S \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}, \text{ где } \langle f \rangle = \frac{1}{N} \sum_{i=1}^N f(x_i, y_i), \quad \langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^N f^2(x_i, y_i).$$

3.2. Формирование родительских пар

Отбор роботов в родительские пары происходит случайно, в соответствии с вычисляемой для каждого робота «вероятностью выбора».

«Вероятности выбора» в популяции роботов определяются с помощью следующего алгоритма.

Так как функция приспособленности некоторых роботов популяции может принимать отрицательные значения, то производится сдвиг функции приспособленности в положительную область значений. С этой целью просматривается вся популяция и регистрируется робот с наихудшей приспособленностью f_{\min} . Если число $f_{\min} \leq 0$, то значение функции приспособленности каждого робота f_i увеличивается на величину $|f_{\min}| + 1$.

Вычисляется среднее значение функции приспособленности всей популяции \tilde{f} .

Для каждого робота определяется параметр p_i , равный отношению его приспособленности к суммарной приспособленности всей популяции.

Для тех роботов, у которых значение функции приспособленности $f_i < \tilde{f}$, «вероятность выбора» принимается равной $0,1 p_i$. Параллельно подсчитывается количество M_1 таких роботов и вычисляется сумма $\tilde{p} = \sum 0,9 p_i$.

Для тех роботов, у которых значение функции приспособленности $f_i \geq \tilde{f}$, «вероятность выбора» принимается равной $p_i + \frac{\tilde{p}}{M - M_1}$, где через M обозначена численность всей популяции.

После этого в соответствии с «вероятностями выбора» формируются родительские пары. Каждая пара составляется из двух различных роботов. Допускается вхождение одного и того же робота в несколько родительских пар.

Если численность популяции равна M , то формируется $M - 1$ родительская пара. Каждая родительская пара производит одного потомка.

Робот с наилучшим значением функции приспособленности добавляется к следующему поколению автоматически.

Таким образом, при переходе от поколения к поколению численность популяции M не изменяется.

3.3. Оператор скрещивания

Поскольку конструкция робота характеризуется тремя определяющими параметрами (v, R, α) , то этот набор параметров можно рассматривать как координаты вектора в 3-мерном пространстве.

Пусть \vec{x} и \vec{y} - 3-мерные векторы, соответствующие двум различным роботам-родителям. Тогда робот-потомок строится так:

$$\vec{x}_{new} = \vec{x} + t(\vec{y} - \vec{x}),$$

где параметр $t \left(t \in \left[-\frac{1}{4}, \frac{5}{4} \right] \right)$ генерируется датчиком случайных чисел с равномерным распределением.

Если определяющие параметры робота-потомка не входят в область допустимых значений параметров, то операция скрещивания этих роботов-родителей производится заново.

3.4. Оператор мутации

Каждый робот из популяции с вероятностью p_{mut} может подвергнуться мутации. В результате мутации роботу присваиваются новые значения определяющих параметров, выбранные случайным способом из области допустимых значений.

Новые определяющие параметры робота, подвергшегося мутации, выбираются, исходя из нормального закона распределения $N(\mu, \sigma^2)$, у которого μ равно исходному значению параметра, а $\sigma^2 = 2\rho$, где ρ - расстояние в 3-мерном евклидовом пространстве между параметрами робота, подвергшегося мутации, и робота из поколения роботов-родителей с наилучшим значением функции приспособленности.

Если новые значения определяющих параметров не попадают в область допустимых значений, то операция случайного выбора новых определяющих параметров проводится заново.

Таким образом, общая схема оптимизации с помощью генетического алгоритма представлена на рисунке 4.



Рис. 4. Общая схема оптимизации с помощью генетического алгоритма.

3.6. Организация процесса параллельных вычислений на вычислительном кластере

Реализация генетического алгоритма на одном компьютере требует больших затрат машинного времени. Генетические алгоритмы в общем случае хорошо поддаются распараллеливанию. Использование вычислительного кластера (кластер Института системного программирования РАН 16 AMD Athlon XP 1500+, Myrinet 2000, N=8 узлов, P=29 Gflops [3]) позволило значительно сократить расход времени, организовав процесс

параллельных вычислений. В данном примере оба алгоритма оптимизации распараллеливались идентично. Программа писалась на языке Java +MPI (версия Java 1.4.2_04) с использованием среды разработки параллельных программ ParJava. Среда ParJava предоставляет пользователю реализацию интерфейса MPI на языке Java, а также набор инструментальных средств [4], получающих и анализирующих динамические свойства разрабатываемой программы, помогая тем самым добиваться требуемого уровня эффективности работы. Также, для сравнения, генетические алгоритмы были реализованы на C+MPI. Результаты показали схожее ускорение программ, что видно из графика.

При написании параллельной версии использовалась так называемая «островная модель» генетических алгоритмов. Островная модель характеризуется тем, что вся популяция делится на несколько подпопуляций, каждая из которых размещается на своем процессоре, которые выполняют генетические операции, а результатами обмениваются посредством миграции. Каждая из этих относительно небольших подпопуляций вычисляется независимо на своем процессоре и время от времени обменивается особями с другими подпопуляциями. На каждом острове подпопуляция подвергается в точности таким же генетическим операциям на таком же генотипе, что и в последовательном алгоритме, и, в результате, отыскивается лучший индивид во всей популяции [5].

Соответственно, можно выделить две группы операций:

- локальные операции, выполняемые на каждом процессоре независимо, которые включают в себя вычисление значений функций приспособленности для каждого робота, выбор индивидуумов, операцию скрещивания, мутацию и формирование родительских пар;
- обмен роботами между процессорами, предполагающий коммуникацию и синхронизацию.

На рисунке 5 приведена общая схема вычислений, где операции из первой группы выделена двойными рамками. Начальная популяция роботов делится по возможности на равные части и в процессе вычислений раз в итерацию цикла производился обмен роботами каждой группы для определения лучшего робота всей популяции и последующего формирования новой популяции. Благодаря небольшому количеству обменов мы получили хорошее ускорение (рис. 6).

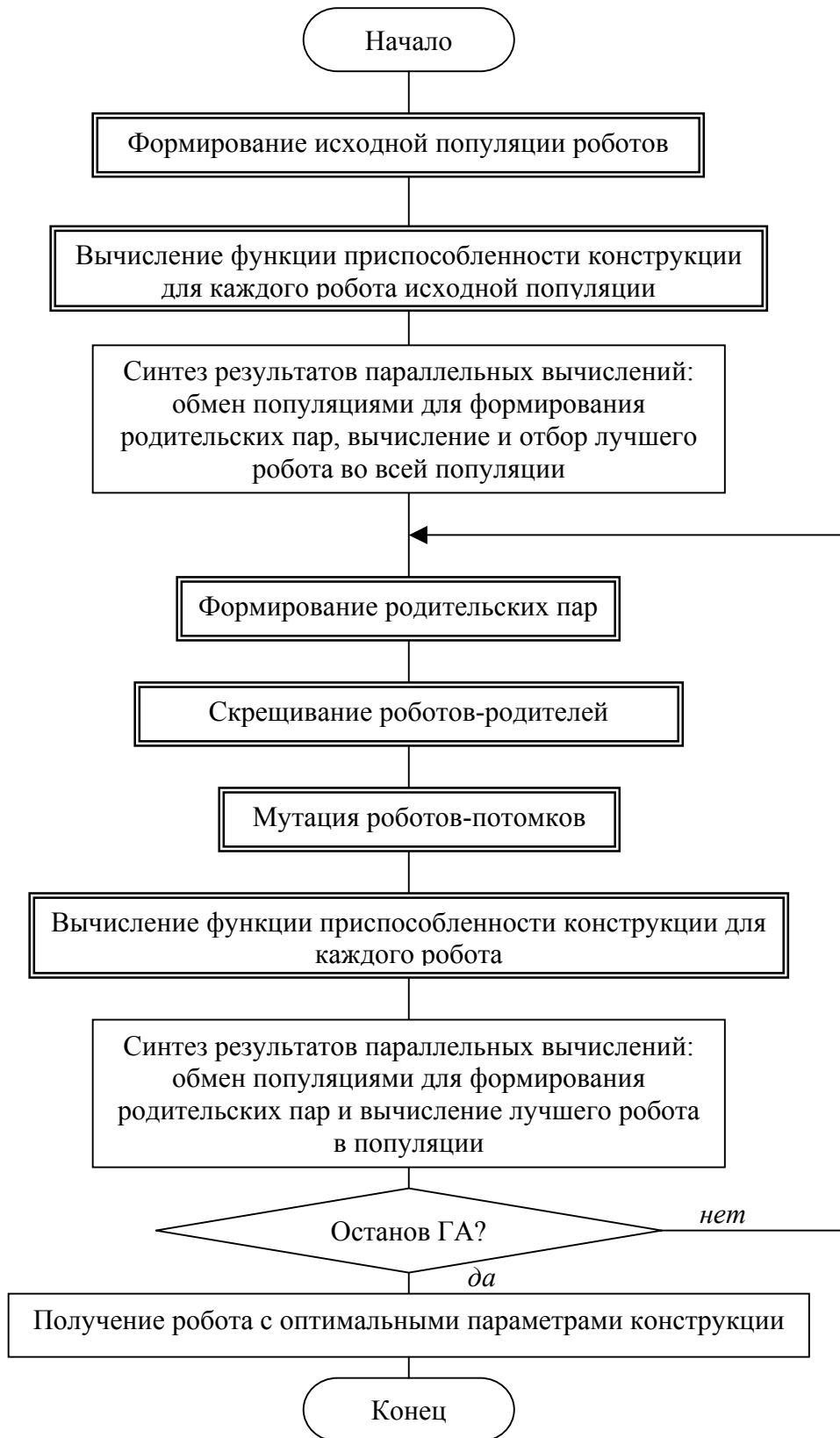


Рис. 5. Общая схема параллельных вычислений.

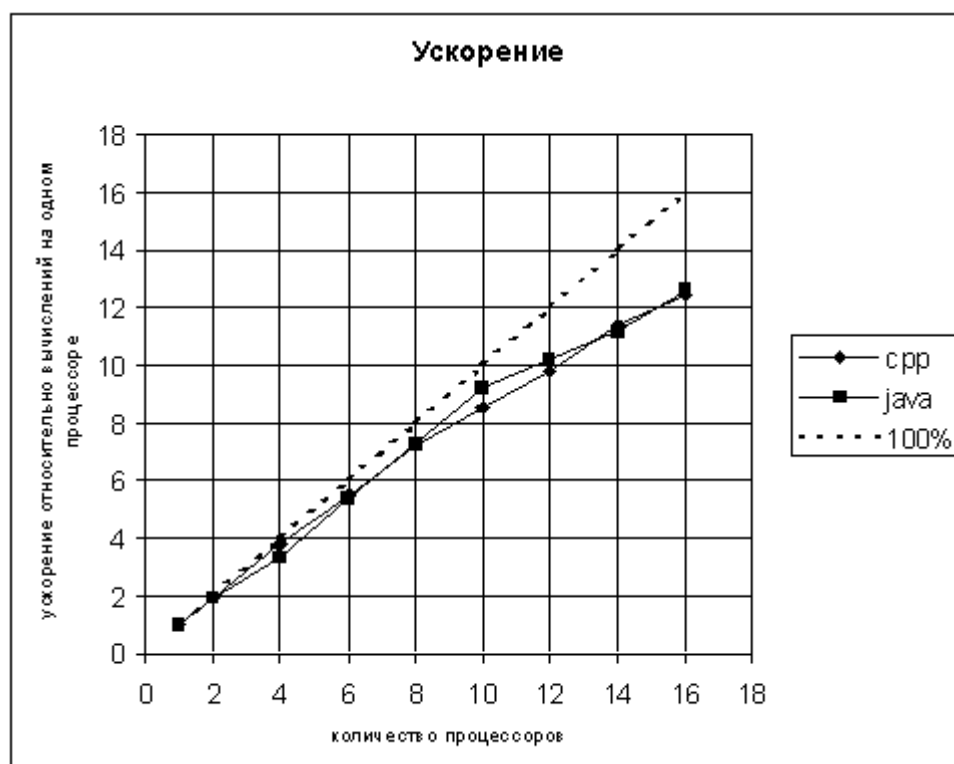


Рис. 6. Ускорение вычислений при использовании вычислительного кластера.

3.7. Результаты расчета оптимальных параметров конструкции робота

Для контролирования процесса сходимости текущие данные по каждой популяции роботов, полученные в ходе работы генетического алгоритма, выводятся в *dat*-файлы. В этих файлах отображаются значения определяющих параметров конструкции каждого робота из популяции (скорость робота v , радиус поворота R , угловое раскрытие центрального сектора α), а также значение функции приспособленности конструкции робота и относительная погрешность ее вычисления в процентах.

Кроме того, заполняется сводный *dat*-файл по всем популяциям, в который заносятся значения определяющих параметров конструкции «лучших» роботов, т.е. роботов, обладающих наивысшим в популяции значением функции приспособленности, и средние значения функций приспособленности роботов каждой популяции.

В результате обработки 500 популяций получились следующие оптимальные параметры конструкции робота:

- скорость робота $v=66.151229$,
- радиус поворота $R=44.158181$,
- угловое раскрытие центрального сектора $\alpha =0.481133$,

где за единицы измерения длины и времени приняты, соответственно, радиус корпуса робота и один такт движения.

4. Моделирование системы управления мобильным роботом на основе принципа автономного адаптивного управления

4.1. Состояния блока датчиков

Как было отмечено выше, система управления мобильным роботом получает информацию от 7 датчиков (трех визуальных и четырех тактильных). Каждый из датчиков может находиться в двух состояниях: «1» и «0». Состояние «1» свидетельствует о том, что препятствие зафиксировано, состояние «0» показывает, что препятствия нет.

Таким образом, в каждый момент времени состояние блока датчиков робота определяется строкой длины 7, состоящей из нулей и единиц. Общее число всевозможных состояний равно 128.

Заметим, что в рассматриваемой модели робота реализуются не все 128 состояний, а всего лишь не более 28.

Действительно, во-первых, в силу выпуклости корпуса робота и препятствий невозможно одновременное срабатывание двух и более тактильных датчиков, во-вторых, срабатывание каждого из передних тактильных датчиков влечет за собой и срабатывание соответствующего визуального датчика.

Если обозначить символами tr, tc, tl, tb состояния правого, центрального, левого и заднего тактильных датчиков соответственно, а символами vr, vc, vl - состояния правого, центрального и левого визуальных датчиков, то весь возможный набор состояний блока датчиков описывается Таблицей 2.

Таблица 2.

№	tc	tl	tr	tb	vc	vl	vr
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
2	0	0	0	0	0	1	0
3	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0
5	0	0	0	0	1	0	1
6	0	0	0	0	1	1	0
7	0	0	0	0	1	1	1
8	0	0	0	1	0	0	0
9	0	0	0	1	0	0	1
10	0	0	0	1	0	1	0
11	0	0	0	1	0	1	1
12	0	0	0	1	1	0	0
13	0	0	0	1	1	0	1
14	0	0	0	1	1	1	0
15	0	0	0	1	1	1	1
16	1	0	0	0	1	0	0
17	1	0	0	0	1	0	1
18	1	0	0	0	1	1	0
19	1	0	0	0	1	1	1
20	0	1	0	0	0	1	0
21	0	1	0	0	0	1	1

22	0	1	0	0	1	1	0
23	0	1	0	0	1	1	1
24	0	0	1	0	0	0	1
25	0	0	1	0	1	0	1
26	0	0	1	0	0	1	1
27	0	0	1	0	1	1	1

4.2. Автоматическое формирование и расширение нейросети

Для управления движением робота в лабиринте формируется и затем автоматически надстраивается специальная нейросеть, которая строится по следующему принципу.

Каждый нейроноподобный элемент (нейрон) имеет четыре характеристики: номер слоя, свой порядковый номер в слое, тип и список исходящих связей (порядковые номера нейронов следующего слоя, порожденных данным нейроном).

Тип нейрона определяется состоянием блока датчиков. Под “типом” понимаем следующее. Соединение нейроноподобного элемента с датчиками организуется таким способом, чтобы с пары клемм – прямой и инверсной, одного бинарного датчика к одному нейрону могла идти только одна связь. При таком построении, один нейроноподобный элемент может иметь только N входящих связей от датчиков. Организуя соединения нейроноподобных элементов с датчиками таким способом, можно получить 2^N различных типов соединений отдельного нейроноподобного элемента с датчиками – по одному на каждую возможную комбинацию бинарных сигналов с датчиков.

Нулевой слой нейросети создается до начала движения робота и содержит 28 элементов. Остальные слои нейросети заполняются нейронами по мере их возникновения при надстраивании нейросети на каждом такте движения робота.

Нейроны нулевого слоя соответствуют конкретным состояниям блока датчиков робота.

Нейроны 1-го слоя содержат информацию сразу о двух последовательных состояниях блока датчиков. Соответственно, нейроны $(n-1)$ -го слоя содержат информацию об n последовательных состояниях блока датчиков.

На каждом такте движения робота составляется список сработавших нейронов. Все эти нейроны, независимо от номера слоя, имеют один и тот же тип, определяемый состоянием блока датчиков робота в текущий момент времени. При получении системой управления робота очередной информации о состоянии блока датчиков список сработавших нейронов обновляется. Каждый из сработавших на предыдущем такте нейронов надстраивается в нейрон, тип которого отвечает текущему состоянию блока датчиков, если он не был надстроен в нейрон этого типа ранее.

Надстраивание нейросети происходит так: в следующий слой добавляется новый нейрон, тип которого соответствует текущему состоянию блока датчиков, а его порядковый номер добавляется в список исходящих связей породившего его нейрона предыдущего слоя.

Расширение нейросети осуществляется до тех пор, пока число нейронов в сети не достигнет предельного значения.

В зависимости от решаемой задачи пользователь программы самостоятельно устанавливает предельные значения числа нейронов и числа слоев нейросети.

4.3. Априорная вероятностная база знаний мобильного робота

Мобильный робот в рассматриваемом здесь варианте обладает априорной базой знаний, представляющей собой 2-мерный массив с 28 строками и 6 столбцами. Строки соответствуют всем возможным типам нейронов, а столбцы – всем возможным вариантам движений робота («вперед-направо», «вперед», «вперед-налево», «назад-направо», «назад», «назад-налево»). Элементами каждой строки массива являются априорные вероятности принятия системой управления роботом решений о выборе одного из 6-ти вариантов движения при срабатывании соответствующего типа нейрона.

Поскольку при срабатывании передних тактильных датчиков движения робота «вперед-направо», «вперед» и «вперед-налево» физически невозможны, то в строках априорной базы знаний, соответствующих нейронам типов от 16 до 27 включительно (см. Табл.2), первые три элемента равны нулю.

Априорная вероятностная база знаний является основой для построения и обновления динамической базы знаний, которая играет важную роль в системе автономного адаптивного управления движением робота.

4.4. Динамическая база знаний и принципы принятия решений

В отличие от неизменяемой априорной базы знаний динамическая база знаний мобильного робота постоянно расширяется и обновляется во время его движения.

До начала движения робота динамическая база знаний совпадает с априорной базой знаний.

На каждом такте движения робота при надстраивании нейросети и появлении в ней нового нейрона происходит добавление в динамическую базу знаний новой строки, совпадающей со строкой априорной базы знаний, соответствующей типу появившегося нейрона.

Каждому нейрону нейросети отвечает строка в динамической базе знаний.

На каждом такте движения система управления фиксирует сработавшие нейроны (нейроны с единичным выходным сигналом) с номерами i_1, i_2, \dots, i_k и вычисляет вектор принятия решений

$$\vec{P}_{\text{реш}} = \frac{1}{k} \sum_{m=1}^k \vec{P}_{i_m},$$

где $\vec{P}_{i_1}, \dots, \vec{P}_{i_k}$ - строки динамической базы знаний, соответствующие сработавшим нейронам. Компоненты вектора $\vec{P}_{\text{реш}}$ используются в качестве вероятностей для принятия системой ААУ решения о выборе одного из шести возможных вариантов дальнейшего движения робота.

4.5. Подсистема моделирования эмоций системы ААУ

Система ААУ имеет подсистему моделирования эмоций, которая на каждом такте движения робота дает качественную оценку $S(t)$ текущего состояния объекта

управления. В данной версии системы управления мы использовали очень простую подсистему эмоций, в которой качественные оценки заданы изначально и не изменяются автоматически в дальнейшем. Оценка $S(t)$ выставляется по показаниям датчиков в соответствии с Таблицей 3.

Таблица 3.

Столкновение с препятствием в текущем состоянии	Препятствие на предыдущем такте движения	Оценка $S(t)$
произошло	не зафиксировано ни одним визуальным датчиком	- 4
произошло	зафиксировано 1 визуальным датчиком	- 3
произошло	зафиксировано 2 визуальными датчиками	- 2
произошло	зафиксировано 3 визуальными датчиками	- 1
не произошло	зафиксировано 3 визуальными датчиками	+ 1
не произошло	зафиксировано 2 визуальными датчиками	+ 2
не произошло	зафиксировано 1 визуальным датчиком	+ 3
не произошло	не зафиксировано ни одним визуальным датчиком	+ 4

Согласно методологии ААУ, качественная оценка текущего состояния блока датчиков $S(t)$ является определяющим фактором при построении динамической базы знаний и оценке эффективности работы системы управления мобильным роботом в целом.

4.6. Обновление динамической базы знаний

Обновление динамической базы знаний происходит по следующей схеме. На основании оценки $S(t)$ текущего состояния блока датчиков, которую выдает подсистема моделирования эмоций на каждом такте движения робота, вычисляется поправка

$$\Delta p = \frac{S(t) - S(t-1)}{8}.$$

Если избранный системой управления вариант движения привел к улучшению качественной оценки $S(t)$, то поправка Δp положительна. В случае ухудшения качественной оценки $S(t)$ поправка Δp отрицательна.

Для внесения изменений в динамическую базу знаний в ней сначала выбираются строки, соответствующие нейронам, сработавшим на предыдущем такте. Затем каждая из этих строк

$$\vec{P}_m = (p_{m1}, \dots, p_{mk}, \dots, p_{m6})$$

преобразуется по следующим правилам.

По строке \vec{P}_m формируется строка $\vec{\tilde{P}}_m$, у которой все компоненты за исключением компоненты \tilde{p}_{mk} , отвечающей последней выполненной команде системы управления, совпадают с соответствующими компонентами строки \vec{P}_m .

Компонента \tilde{p}_{mk} вычисляется по формуле

$$\tilde{p}_{mk} = \begin{cases} p_{mk} + \Delta p, & \text{если } \Delta p > 0, \\ p_{mk} (1 + \Delta p), & \text{если } \Delta p \leq 0. \end{cases}$$

Исходная строка динамической базы знаний \vec{P}_m заменяется на строку

$$\frac{\vec{\tilde{P}}_m}{\sum_{i=1}^6 \tilde{p}_{mi}}.$$

Кроме обновления существующих строк на каждом такте работы системы управления динамическая база знаний пополняется новыми строками, которые соответствуют нейронам, образовавшимся при проращивании нейросети.

Отметим здесь, что имеются и другие способы построения баз знаний для систем ААУ в нейроноподобном варианте их реализации. Например, нами разработан способ построения базы знаний не на вероятностной основе, который обладает своими особенностями; этот способ подробно описан в [6]. Еще один вариант построения базы знаний для робота описан в [2].

4.7. Оценка эффективности работы системы ААУ

Для оценки эффективности функционирования системы ААУ в целом используется характеристика U - среднее значение качественных оценок состояния блока датчиков $S(t)$, которые на каждом такте движения робота выдаются подсистемой моделирования эмоций.

Поскольку в начальный период работы система ААУ больше обучается, чем управляет, то расчет характеристики U эффективности работы системы ААУ проводится по наблюдениям, зафиксированным в течение последней трети периода ее функционирования, когда акты обучения уже очень редки. Чем удачнее реализация системы управления, тем выше средняя оценка.

Так как система управления использует в своей работе случайные переменные, то характеристика, полученная в результате наблюдения одного эксперимента по оценке эффективности работы системы, не является точной. Для получения более точной

характеристики проводится серия экспериментов и на их основе вычисляется усредненная характеристика. Идеальной считается такая система ААУ, характеристика U которой является максимально возможной, то есть $U = 4$.

4.8. Схема проведения эксперимента по оценке эффективности работы системы ААУ при движении робота в лабиринте

Для того, чтобы проверить, насколько хорошо функционирует система управления движением мобильного робота в лабиринте, ставится эксперимент, который проводится по следующей схеме (рис. 7).



Рис. 7. Алгоритм оценки эффективности работы системы ААУ при движении робота в лабиринте.

5. Оптимизация системы управления мобильным роботом

Задача оптимизации системы управления мобильным роботом состоит в подборе для робота такой априорной вероятностной базы знаний, которая обеспечивает наиболее эффективное функционирование его системы ААУ.

Для решения задачи оптимизации системы управления мобильным роботом используется генетический алгоритм. Генотипом индивида является априорная база знаний. Отбор роботов с наилучшими качествами для формирования следующего поколения производится с помощью функции приспособленности.

Функция приспособленности системы управления роботом – это число, характеризующее степень близости показателя U эффективности работы системы ААУ данного робота к показателю эффективности работы идеальной системы ААУ. Здесь под “идеальной” системой ААУ понимается такая система, которая в пространственной системе оптимизируемых признаков имеет вполне определенные, но на практике недостижимые значения. Однако эти значения можно использовать как точки отсчета для относительного сравнения всех остальных вариантов.

Для реализации генетического алгоритма сначала создается исходная популяция роботов с различными априорными базами знаний, которые выбираются случайно. Затем вычисляются вероятности попадания роботов в родительские пары («вероятности выбора»). «Вероятность выбора» конкретного индивида зависит от значения функции приспособленности его системы управления, а также от значений функций приспособленности систем управления роботов всей популяции. После скрещивания каждый робот-потомок с некоторой вероятностью подвергается мутации. В итоге образуется новое поколение роботов. Эта процедура повторяется до тех пор, пока не будет выполнен критерий останова генетического алгоритма (число популяций устанавливается пользователем программы).

5.1. Схема расчета функции приспособленности системы управления

Значение функции приспособленности системы ААУ зависит от оценки эффективности ее работы и определяется формулой

$$F = \frac{4}{8 - U}.$$

Процесс определения показателя U эффективности работы системы управления роботом по серии экспериментов и вычисление функции приспособленности проводится по следующей схеме (рис. 8).



Рис. 8. Схема расчетов для определения показателя эффективности системы управления.

Следует отметить, что при проведении серии экспериментов с одним и тем же роботом каждый последующий эксперимент развивает динамическую базу знаний и нейросеть, которые сформировались в предыдущих экспериментах.

5.2. Формирование родительских пар

Отбор роботов в родительские пары происходит случайно, в соответствии с вычисляемыми для роботов «вероятностями выбора».

«Вероятности выбора» определяются с помощью следующего алгоритма.

1. Вычисляется среднее значение функции приспособленности всей популяции \tilde{F} .
2. Для каждого робота определяется параметр p_i , равный отношению его приспособленности к суммарной приспособленности всей популяции.
3. Для тех роботов, у которых значение функции приспособленности $F_i < \tilde{F}$, «вероятность выбора» принимается равной $0,1 p_i$. Параллельно подсчитывается количество M_1 таких роботов и вычисляется сумма $\tilde{p} = \sum 0,9 p_i$.
4. Для тех роботов, у которых значение функции приспособленности $F_i \geq \tilde{F}$, «вероятность выбора» принимается равной $p_i + \frac{\tilde{p}}{M - M_1}$, где через M обозначена численность всей популяции.

После этого формируются родительские пары. Каждая пара составляется из двух различных роботов. Допускается вхождение одного и того же робота в несколько родительских пар. Если численность популяции равна M , то формируется $M-1$ родительская пара. Каждая родительская пара производит одного потомка. Робот с наилучшим значением функции приспособленности добавляется к следующему поколению автоматически.

5.3. Скрещивание роботов

Генотип каждого робота определяется априорной базой знаний, то есть матрицей $A = (a_{ij})$ типа $[28 \times 6]$, элементы которой удовлетворяют следующим соотношениям:

$$0 \leq a_{ij} \leq 1 \text{ для всех } 0 \leq i \leq 27, \quad 0 \leq j \leq 5;$$

$$\sum_{j=0}^5 a_{ij} = 1 \text{ для всех } 0 \leq i \leq 27;$$

$$a_{ij} = 0 \text{ для всех } 16 \leq i \leq 27, \quad j = 0, 1, 2.$$

Пусть A и B - матрицы генотипов двух роботов-родителей. В результате операции скрещивания получается робот-потомок, генотип C которого задается формулой

$$C = A + t(B - A),$$

где параметр t ($t \in [0, 1]$) генерируется датчиком случайных чисел с равномерным распределением.

Заметим, что полученная таким способом матрица C при всех значениях параметра $t \in [0, 1]$ удовлетворяет условиям 1,2,3 и действительно может рассматриваться в качестве генотипа мобильного робота.

5.4. Мутация роботов

Каждый робот из популяции с вероятностью p_{mut} может подвергнуться мутации, в результате которой его априорная база знаний $A = (a_{ij})$ заменяется на $\tilde{A} = (\tilde{a}_{ij})$ по следующим правилам.

$$\text{Для } 16 \leq i \leq 27, \quad j = 0, 1, 2 \text{ элементы } \tilde{a}_{ij} = a_{ij} = 0.$$

Все остальные элементы \tilde{a}_{ij} выбираются исходя из нормального закона распределения $N(\mu, \sigma^2)$ с $\mu = a_{ij}$ и

$$\sigma^2 = \sqrt{\sum_{i=0}^{27} \sum_{j=0}^5 (a_{ij} - b_{ij})^2},$$

где $B = (b_{ij})$ - априорная база знаний робота из поколения роботов-родителей с наилучшим значением функции приспособленности.

Все отрицательные элементы матрицы \tilde{A} заменяются на нули.

Каждая ненулевая строка матрицы \tilde{A} нормируется, то есть ее элементы \tilde{a}_{ij} заменяются на

$$\frac{\tilde{a}_{ij}}{\sum_{j=0}^5 \tilde{a}_{ij}}.$$

Каждая нулевая строка матрицы \tilde{A} заменяется на соответствующую строку исходной матрицы A .

Полученная в результате этих операций матрица \tilde{A} удовлетворяет условиям 1-3 пункта 5.3 и используется в качестве генотипа робота-мутанта.

5.5. Результаты расчета оптимальных параметров системы ААУ

Для контролирования процесса сходимости текущие данные по каждой популяции роботов, полученные в ходе работы генетического алгоритма, выводятся в *dat*-файлы. В этих файлах фиксируются значения функции приспособленности системы ААУ, числа нейронов в нейросети и априорная база знаний системы ААУ каждого робота из популяции.

Помимо этого, заполняется сводный *dat*-файл по всем популяциям, в который заносятся наивысшие и средние значения функций приспособленности систем ААУ роботов каждой популяции. В этот же файл выводится и окончательный результат работы генетического алгоритма – оптимальная априорная база знаний системы ААУ.

В результате обработки 100 популяций получилась следующая оптимальная априорная база знаний (Таблица 4).

Таблица 4.

Тип нейрона	«вперед-направо»	«вперед»	«вперед-налево»	«назад-направо»	«назад»	«назад-налево»
0	0.141622	0.183516	0.142791	0.181005	0.192140	0.158926
1	0.108586	0.066873	0.199556	0.278332	0.083482	0.263171
2	0.209672	0.223982	0.197452	0.126108	0.092866	0.149920
3	0.110389	0.115509	0.314849	0.091610	0.205607	0.162035
4	0.193046	0.316465	0.101252	0.119948	0.082261	0.187028
5	0.217434	0.188244	0.126212	0.202834	0.049810	0.215465
6	0.152947	0.213423	0.165188	0.186781	0.081349	0.200312
7	0.279198	0.224800	0.095258	0.103027	0.172960	0.124758
8	0.170527	0.081985	0.171960	0.139070	0.224069	0.212390
9	0.154633	0.180394	0.294573	0.140286	0.028957	0.201157
10	0.082708	0.161329	0.137279	0.211248	0.203411	0.204025
11	0.191846	0.339691	0.116274	0.122461	0.176902	0.052825
12	0.148198	0.108315	0.233745	0.175919	0.121909	0.211915
13	0.205268	0.137347	0.104957	0.079871	0.282393	0.190165
14	0.227979	0.139938	0.119611	0.250579	0.104509	0.157384
15	0.141065	0.118774	0.240665	0.189698	0.167526	0.142273
16	0.000000	0.000000	0.000000	0.353457	0.389900	0.256642

17	0.000000	0.000000	0.000000	0.193515	0.330058	0.476427
18	0.000000	0.000000	0.000000	0.198943	0.171011	0.630045
19	0.000000	0.000000	0.000000	0.177748	0.385197	0.437054
20	0.000000	0.000000	0.000000	0.272198	0.220643	0.507159
21	0.000000	0.000000	0.000000	0.258159	0.406276	0.335565
22	0.000000	0.000000	0.000000	0.243869	0.184708	0.571423
23	0.000000	0.000000	0.000000	0.230332	0.299344	0.470325
24	0.000000	0.000000	0.000000	0.242498	0.396957	0.360544
25	0.000000	0.000000	0.000000	0.279066	0.272603	0.448331
26	0.000000	0.000000	0.000000	0.399899	0.309248	0.290852
27	0.000000	0.000000	0.000000	0.489620	0.258690	0.251690

6. Заключение

Основные результаты данной работы состоят в том, что были предложены и разработаны способы оптимизации с помощью генетических алгоритмов главных подсистем объекта управления, снабженного системой ААУ, с применением параллельного вычислительного кластера. Оптимизации подверглись параметры блока датчиков, исполнительных устройств, а также вероятностная база знаний управляющей системы.

Блок датчиков и исполнительные устройства оптимизировались как одно целое, при этом варьировались такие параметры, как скорость, радиус поворота и угловое раскрытие центрального сектора. Минимизировалось число ложных срабатываний визуальных датчиков. Оптимизация системы управления мобильным роботом состояла в подборе наиболее эффективной вероятностной базы знаний. Обе задачи были решены с помощью генетических алгоритмов.

Были найдены подходы к реализации оптимизационных программ в параллельном варианте на вычислительном кластере, что позволяет существенно снизить временные затраты на расчеты.

В целом результаты данной работы существенно развивают методику построения прикладных систем управления на основе метода ААУ.

Литература

1. А.А. Жданов, Метод автономного адаптивного управления // Известия Академии Наук. Теория и системы управления, 1999, № 5, с. 127-134.
2. А.А. Жданов, М.В. Крыжановский, Н.Б. Преображенский, Бионическая интеллектуальная автономная адаптивная система управления мобильным роботом // Мехатроника, 2004, №1, С. 21-30 и №2, С.17-22 (часть 2).
3. <http://www.ispras.ru/groups/ctt/ispran.html>
4. Victor Ivannikov, Serguei Gaissaryan, Arutyun Avetisyan, Vartan Padaryan. Improving properties of a parallel program in ParJava Environment // The 10th EuroPVM/MPI conference. LNCS 2840. Sept. 2003, Venice. pp. 491-494.
5. Rajkumar Buyya "High Performance Cluster Computing", Prentice Hall PTR, vol.2, pp 331-349.

6. Л.В. Земских, Возможности оптимизации системы автономного адаптивного управления с помощью генетических алгоритмов. Препринт №3 Института системного программирования РАН, Москва, 2004.