

УДК 004.89

МНОГОАГЕНТНЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧ  
НЕИНФОРМИРОВАННОГО ПОИСКА

И.А. Бессмертный, К.А. Булыгин

Рассматривается подход к задаче поиска на дереве решений, основанный на использовании множества агентов, обменивающихся информацией и обладающих роевым интеллектом. В отличие от классических методов поиска, данный подход обеспечивает быстрое получение удовлетворительных решений и может использоваться в задачах с большим числом участников движения при условии доступа к коллективной базе данных для получения известных маршрутов и сохранения накопленного опыта.

**Ключевые слова:** задача поиска, агенты, роевой интеллект.

## Введение

Существует обширный класс задач, решение которых сводится к поиску на дереве решений. К ним относятся транспортно-логистические задачи, планирование, упаковка, раскрой, маршрутизация в коммуникационных сетях и др. Самыми простыми являются методы неинформированного поиска, не требующие дополнительной информации для выбора путей на дереве решений. Эти методы имеют экспоненциально возрастающую сложность, поэтому получить оптимальное решение в приемлемые сроки не всегда возможно [1].

Между тем в повседневной жизни нам постоянно приходится решать подобные задачи, и помогает в этом опыт, как собственный, так и заимствованный. В последнее время в решении многих задач активно используется подход, называемый роевым интеллектом (*swarm intelligence*) [2], заключающийся в использовании многочисленных агентов, функционирующих по очень простому алгоритму и обменивающихся между собой информацией. Результативность работы роя агентов может быть существенно выше, чем одного агента, работающего по сложному алгоритму. Целью данного исследования является оценка возможности применения мультиагентного подхода и роевого интеллекта к неинформированному поиску.

## Метод случайных блужданий

Вероятность  $p$  достижения цели при однократном случайном перемещении по дереву поиска на величину средней глубины дерева вычисляется следующим образом:

$$p = 1/b^d,$$

где  $b$  – коэффициент ветвления дерева поиска;  $d$  – средняя глубина дерева поиска. Среднее число попыток  $A$  до первого достижения цели при равномерном распределении вероятностей ветвления в каждой  $i$ -ой ветви дерева

$$A = \sum_{i=1}^{b^d} p(1-p)^{i-1}i = \sum_{i=1}^{b^d} \frac{1}{b^d} \left(1 - \frac{1}{b^d}\right)^{i-1} i. \quad (1)$$

Пусть имеется транспортная сеть размерностью  $N \times N$ , изображенная на рис. 1, состоящая из узлов и дуг, где узлы соответствуют перекресткам, а дуги – дорогам.

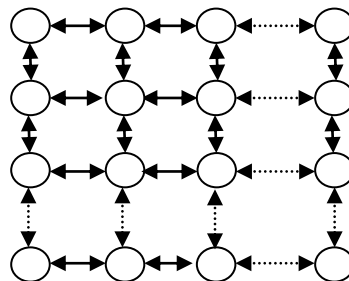


Рис. 1. Топология сети

Участник движения (агент) совершает поездки из одного узла в другой. Нахождение оптимального маршрута требует знания взаимного положения узлов и дорожной обстановки. В противном случае размерность задачи поиска не позволит решить ее в разумные сроки. Длительность поиска методом сначала в ширину определяется формулой

$$T = tb^d, \quad (2)$$

где  $t$  – время развертывания одного узла;  $b$  – коэффициент ветвления;  $d$  – глубина поиска (средняя длина маршрута). Для вычисления времени  $T$  по (2) необходимо располагать значениями  $t$ ,  $b$  и  $d$ . Средняя протяженность оптимального маршрута (глубина поиска) для данной топологии сети равна  $2N/3$ , а коэффициент ветвления  $b = 2$  при  $N = 2$  и  $b \rightarrow 3$  при  $N \rightarrow \infty$ . Величину  $t$  можно определить только экспериментальным путем. Однако подобрать  $t$  для (1) не удалось, а для результатов натуральных экспериментов была найдена аппроксимирующая функция

$$T_N = tb^{2N}, \tag{3}$$

где время развертывания одного узла  $t = 0,15 \cdot 10^{-5}$  подобрано для конкретного компьютера, на котором проводились измерения. Подстановка в (2) значения размерности сети  $N = 10$  даст среднее время поиска около 1,5 часов, что неприемлемо для большинства применений.

Среднее число развертываемых вершин для метода случайных блужданий  $V_r$  будет равно

$$V_r = dA = d \sum_{i=1}^{b^d} \frac{1}{b^d} \left(1 - \frac{1}{b^d}\right)^{i-1} i.$$

На рис. 2 приведено сравнение числа развертываемых вершин при поиске сначала в ширину и случайном поиске.



Рис. 2. Сравнение поиска в ширину и случайного поиска

Как и ожидалось, случайные блуждания приводят к существенно большему числу развертываемых вершин до первого достижения цели, чем при поиске методом перебора, что делает такой подход совершенно бесперспективным.

Если агент, двигаясь по сети, не имеет цели, но при случайных блужданиях сохраняет в памяти все маршруты, то через некоторое время сеть окажется полностью покрытой маршрутами, соединяющими каждую точку с каждой. На рис. 3 показаны расчетные зависимости средней длины маршрута при случайном поиске, а также сложности поиска в ширину (числа развертываемых вершин) от размерности сети.

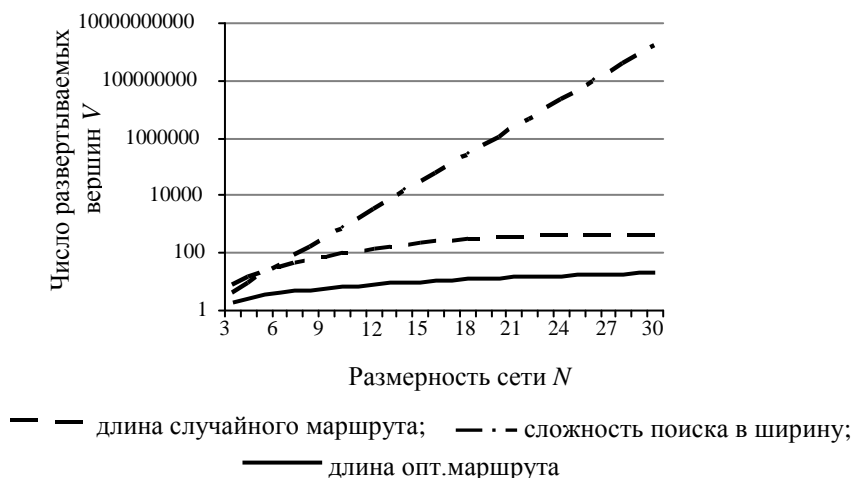


Рис. 3. Сравнение поиска в ширину и метода случайных блужданий

Приведенные здесь данные показывают, что сложность покрытия всей сети маршрутами методом случайного поиска на 3–7 порядков ниже, чем при поиске одного маршрута путем перебора, но средняя длина маршрута больше кратчайшей длины приблизительно на порядок. Очевидно, что такой результат нельзя признать удовлетворительным.

Если при случайных блужданиях агент вправе на каждом шаге выбирать любое направление, то возможны заведомо неэффективные маршруты. Для устранения таких неэффективных маршрутов интеллект агентов можно усилить, добавив два варианта запоминания исходного направления и запретив при случайных блужданиях движение, противоположное первоначально выбранному (тактика «ни шагу назад»).

Вариант «а» предполагает, что агент размещается в произвольно выбранном узле сети и случайно выбирает одно из четырех направлений. При последующих блужданиях разрешаются все направления, кроме противоположного первоначальному. Предположительно, такой алгоритм будет плохо прокладывать диагональные маршруты, хотя при такой топологии сети движение по диагонали зигзагами не более эффективно, чем перемещение по катетам. Заметим, что в реальной транспортной задаче постоянные повороты лишь замедляют движение. При варианте «б» агент первоначально выбирает одно из восьми направлений, четыре ортогональных и четыре диагональных. Если на первом шаге выбрана диагональ, то при последующих блужданиях разрешены только два направления, горизонтальное и вертикальное, в сторону данной диагонали.

Рис. 4 показывает результаты компьютерного моделирования полностью случайных блужданий, а также с использованием двух вариантов алгоритма «ни шагу назад». Для сравнения показано также расчетное время поиска в ширину.

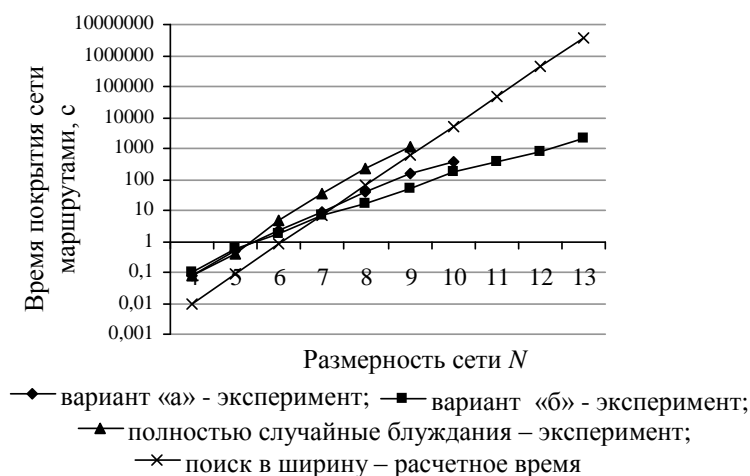


Рис. 4. Время покрытия сети

Результаты экспериментов показывают, что при полностью случайных блужданиях, начиная с размерности сети  $8 \times 8$ , сеть полностью покрывается маршрутами быстрее, чем поиск в ширину находит один маршрут, а тактика «ни шагу назад» дает ускорение покрытия сети еще на порядок.

**Исследование сочетания метода случайных блужданий и роевого интеллекта**

Предположим, что имеется множество агентов (участников движения), которые совершают поездки с заданными целями и запоминают результаты в общей базе данных. Если маршрут, соединяющий две точки, уже имеется в базе, он прокладывается заново, и если длительность нового маршрута меньше, чем уже запомненного, то данные в базе обновляются, т.е. происходит обучение роя агентов. Таким образом, после множества поездок всех агентов роя будут найдены кратчайшие маршруты для всех пар начальных и конечных точек.

Как и в предыдущих экспериментах, вместо нахождения одного маршрута ставится задача найти все возможные маршруты в сети. Если между текущей позицией агента и целью уже есть проложенный маршрут, он присоединяется к текущему маршруту. Если между стартовой и финишной точкой есть промежуточная точка, к которой уже есть маршруты от старта и финиша, то маршрут строится из двух уже проложенных отрезков. Исследование данного подхода проводилось на компьютерной модели, написанной на языке SWI-Prolog. Зависимость времени покрытия сети  $T$  от размерности  $N$  представлена на рис. 5.

Полученные результаты показывают, что агенты, обменивающиеся опытом, покрывают всю сеть маршрутами существенно быстрее, чем это делает один агент. Поскольку каждый найденный маршрут сохраняется в базе маршрутов, эта база довольно быстро становится полной, а дальнейшие перемещения агентов только сокращают длины маршрутов. Таким образом, рой агентов достаточно быстро дает все решения, пусть даже и не идеальные, а дальнейшие блуждания сокращают длины маршрутов.

Для количественной оценки качества покрытия сети введем понятие оптимальность как отношение минимальной суммарной длины маршрутов к их фактической суммарной длине.

Рис. 6 демонстрирует скорость сходимости процесса поиска решений, полученную в результате экспериментов на модели. Графики на рис. 6 показывают, что средняя протяженность маршрута при-

ближается к оптимальной по асимптоте. Таким образом, здесь действует закон сохранения: выигрыш времени получен за счет потери качества. Получить покрытие сети кратчайшими маршрутами можно лишь после длительного поиска.

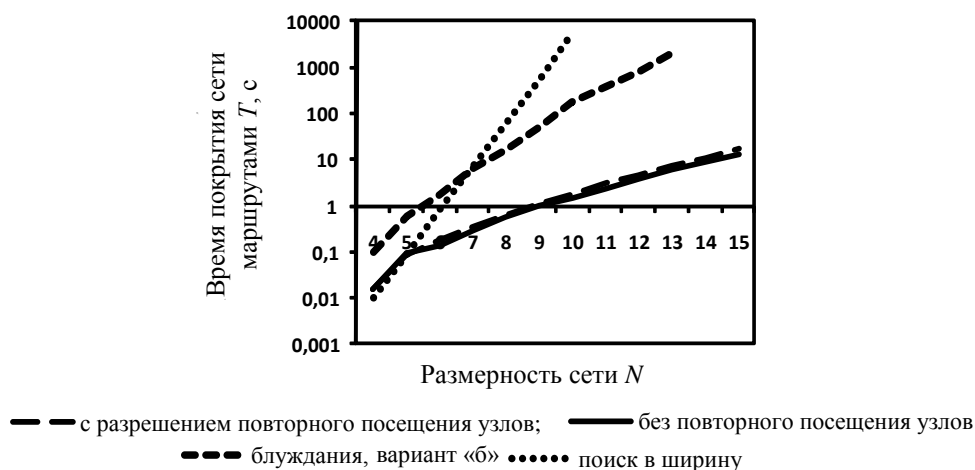


Рис. 5. Зависимость времени покрытия  $T$  от размерности сети  $N$

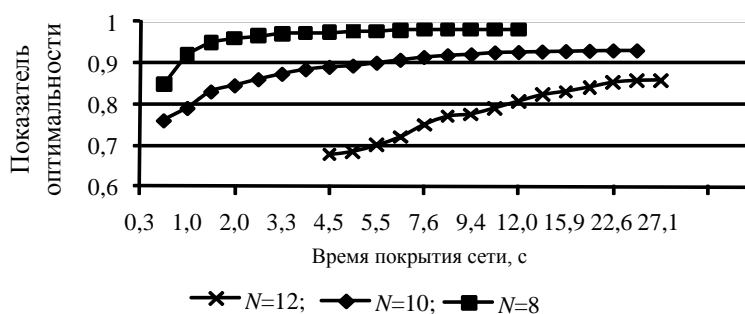


Рис. 6. Сходимость процесса поиска решений

### Заключение

Результаты исследования роевого интеллекта применительно к поиску маршрутов в транспортной сети показывают, что метод случайных блужданий при накоплении опыта показывает удовлетворительные результаты, причем не требует эвристик. В отличие от других методов (либо решение оптимальное, либо его нет вовсе) [3, 4] качество решения здесь является функцией от длительности обучения агентов. Примечательно, что сеть полностью покрывается маршрутами существенно быстрее, чем классические алгоритмы находят единственный маршрут. Скорость нахождения решений зависит не от индивидуального интеллекта агентов, а, главным образом, от способности агентов обмениваться опытом. Таким образом, эффект роевого интеллекта заключается в доступности накопленного чужого опыта.

### Литература

1. Рассел С., Норвиг П. Искусственный интеллект: Современный подход. – 2-е изд. Пер. с англ. – М.: Вильямс, 2006. – 1408 с.
2. Engelbrecht A.P. Fundamentals of Computational Swarm Intelligence. – John Wiley & Sons, Chichester, UK, 2005.
3. Бессмертный И.А. Теоретико-множественный подход к логическому выводу в базах знаний // Научно-технический вестник СПбГУ ИТМО. – 2010. – № 2(66). – С. 43–48.
4. Бессмертный И.А. Быстрый логический вывод в среде программирования Visual Prolog // Научно-технический вестник СПбГУ ИТМО. – 2010. – № 3(67). – С. 50–56.

**Бессмертный Игорь Александрович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, igor\_bessmertny@hotmail.com

**Булыгин Кирилл Александрович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, kirill.bulygin@gmail.com