



УДК 004.7

О. И. Кутузов, Т. М. Татарникова
Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

И. Н. Дзюбенко
Санкт-Петербургский государственный университет
аэрокосмического приборостроения

Решение одной задачи размещения сенсорных устройств в сетях интернета вещей

Представлен один из возможных вариантов решения задачи размещения сенсорных устройств в коммутационном поле интернета вещей. Предложен алгоритм решения задачи размещения, в основе которого лежит эволюционная теория и адаптивное поведение сенсорных устройств. При программной реализации генетического алгоритма это позволит сенсорным устройствам самоорганизовываться в коммутационном поле интернета вещей. Коммутационное поле является неоднородным участком, на территории которого располагаются естественные препятствия в виде зданий, деревьев и других объектов. Приведено соответствие терминов, заимствованных из генетики, терминам генетического алгоритма, что позволяет адаптировать алгоритм к решению задачи размещения. Все промежуточные и конечные решения должны иметь одинаковую длину кода, что осложняет процесс поиска решения. В статье описаны способы кодирования и декодирования решений. Предложенный алгоритм доведен до программной реализации, в котором результат визуализируется в виде карты территории размещения с препятствиями и размещаемыми сенсорными устройствами с радиусами их действия.

Сеть интернета вещей, сенсорное устройство, задача размещения, NP-трудная задача, моделирование эволюции, генетический алгоритм, целевая функция

Сеть интернета вещей – это современная инфокоммуникация, объединяющая миниатюрные беспроводные узлы узкого назначения (сенсорные устройства – СУ) по беспроводной связи с выходом в глобальные инфокоммуникации. В радиусе действия радиосигнала каждого СУ должно находиться как минимум одно соседнее СУ для возможности передачи данных в глобальное «облако» с целью дальнейшей их обработки и сохранения в рамках задачи, решаемой интернетом вещей. Чем больше «соседей» у каждого из СУ, тем устойчивее интернет вещей. Технологии беспроводной связи в интернете вещей основаны на стандарте IEEE 802.15.4, что гарантирует передачу данных на расстояния до нескольких десятков метров [1], [2].

Структурная схема СУ показана на рис. 1 [3].

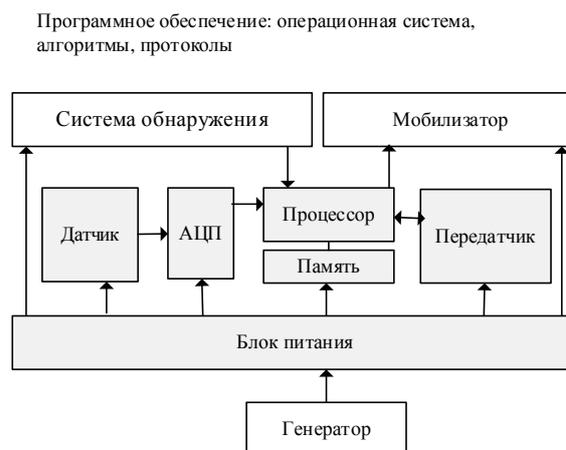


Рис. 1

Аналоговый сигнал датчика, соответствующий измеряемому им явлению, преобразуется в цифровой и подается на обработку процессором.

Процессор и память вместе реализуют частичную обработку данных, например удаление шумов, формирование пакета данных для дальнейшей его передачи в «облако», где обрабатываются и сохраняются данные, поступающие от всех СУ в целях решения задачи.

Передачик является интерфейсом между СУ и сетью [4].

Система обнаружения, которая является элементом мобильных СУ, реализует функцию геолокации СУ, а возможность перемещения СУ реализуется мобилизатором. Для обеспечения бесперебойной работы на заданное время в СУ интегрируется блок питания и генератор. Все перечисленные элементы встраиваются в один корпус [5].

При развертывании работы интернета вещей на заданной территории приходится решать задачу оптимального размещения сенсорных устройств на этой территории.

В силу разнообразия сенсорных устройств и их динамических свойств, таких, как изменение статуса СУ (активный или спящий режим), изменение количества СУ и других, при реализации размещения необходимо учитывать эти особенности. Одним из подходов к решению подобных задач являются генетические алгоритмы [6].

Суть и терминология генетического алгоритма. Эффективность генетических алгоритмов (ГА) при решении большинства NP-сложных задач является известным фактом. ГА могут оказаться единственно возможным методом решения задачи, математическая модель которой имеет сложную структуру, а применение стандартных методов типа ветвей и границ, динамического или линейного программирования не приводит к результату.

Генетический алгоритм является своеобразным инструментом решения задачи оптимизации, основанным на феномене естественной эволюции [7].

При естественной эволюции к сложным и меняющимся условиям окружающей среды адаптируются наиболее здоровые особи и соответственно дают самое многочисленное потомство. Степень адаптации, в свою очередь, зависит от набора хромосом конкретной особи, унаследованных от родителей. Это является основным законом выживания сильнейшего – особь не только выживает, но и участвует в формировании следующего поколения. В природе этот закон является определяющим и формирующим основную функцию выживания.

Таким образом, работа ГА заключается в последовательном стягивании пространства поиска до

одной или нескольких точек, которые и будут представлять собой оптимальные решения. Этот процесс происходит под воздействием генетических операторов алгоритма скрещивания и мутации.

Преимуществом ГА является отсутствие требований к виду целевой функции и виду ограничений на значения параметров задачи оптимизации. Это преимущество ГА в сравнении с другими методами оптимизации делает его в некотором роде «универсальным» инструментом решения широкого круга задач оптимизации.

С другой стороны, известным недостатком ГА является отсутствие оценки сходимости алгоритма.

ГА отличаются от других известных алгоритмов оптимизации несколькими базовыми элементами [7]:

- все параметры задачи кодируются в двоичной форме и обрабатываются в таком виде;
- поиск оптимального решения осуществляется сразу из некоторого множества отправных точек;
- при поиске оптимального решения используется только целевая функция, а не ее производные;
- применяются вероятностные операции выбора пространства решений.

При описании ГА используются определения, заимствованные из генетики. Соответствие терминологий генетики и генетического алгоритма приведено в таблице [8].

Термин, заимствованный из генетики	Термин генетического алгоритма
Ген $G_i = \{g_1, \dots, g_j, \dots, g_L\}$	Строка символов $\{a_1, \dots, a_j, \dots, a_L\}$, представленная двоичным кодом или в формате с плавающей запятой, где a_j – бит
Аллель – множество возможных значений гена	«0» или «1»
Хромосома (особь) Ch_j – набор генов $G_i, i = \overline{1, N}$	Промежуточное или конечное решение $Ch_j \in P, j = \overline{1, M}$
Популяция $P = \{Ch_1, Ch_2, \dots, Ch_M\}$	Пространство поиска решений P
Оценка приспособленности хромосомы	Оценка степени близости решения к оптимальному значению целевой функции
Поклоение	Итерация
Функция приспособленности F	Целевая функция задачи оптимизации

Работа ГА является итеративным процессом, условием останова которого может быть один из вариантов: близость полученных решений к заданному значению целевой функции; прекращение изменения значения целевой функции; численность популяции сократилась до заданного порога; число итераций достигло заданного значения и др.

На каждой итерации ГА оценивается приспособленность особей популяции к значению целевой функции и из наиболее приспособленных создается следующая популяция особей, составляющих множество потенциальных решений задачи, т. е. фактически выполняется поиск хромосомы Ch^* , для которой $F(Ch^*) = \max_P F$.

Кодирование и декодирование решений ГА. Предлагаемый ГА позволяет автономно распределяться в коммутационном поле действия интернета вещей с целью максимального покрытия территории.

Предполагается, что коммутационное поле состоит из ячеек, имеющих один из трех статусов: свободное место, естественное препятствие (строения, деревья и т. п.), размещенное СУ.

При наличии одинаковых СУ задача их размещения сводится к очевидному детерминированному решению, типа задачи упаковки шаров. Однако будем считать, что СУ неоднородны, т. е. имеют разный радиус действия и функционально предназначены для сбора и передачи разной информации. Для решения задачи предложен следующий генетический алгоритм.

Коммутационное поле для размещения СУ представим как матрицу A со следующими значениями элементов:

- ноль – пустое (незанятое) место на территории;
- единица – естественное препятствие;
- $2 \dots N + 1$ – установленное СУ, где N – номер СУ.

На рис. 2 показано соответствие представления области размещения СУ (a) в виде матрицы (b).

При реализации ГА возникает необходимость разбить матрицу A на 2 множества: множество B , в котором исключены ячейки с единичным статусом (содержащие препятствия), и вспомогательное множество C , помогающее в дальнейшем восстановить из B оригинальную матрицу A :

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 \end{pmatrix} \Rightarrow \begin{matrix} C = [22 \ 23 \ 24 \ 25 \ 32 \ \dots] \\ B = \overbrace{[0 \ 0 \ \dots \ 0 \ 2]}^{N_i} \end{matrix}$$

Множества B и C получают следующим образом: выполняется поэлементный обход матрицы A , в результате которого для каждой ячейки, не являющейся препятствием, во множество C записываются индексы этой ячейки, а во множество B записывается числовое значение ячейки (ноль или номер СУ). Таким образом, анализируя множество B можно определить пустые позиции, где еще возможно разместить СУ. Если обозначить размер множества B как N , то для кодирования индекса СУ в этом множестве необходимо количество бит $N_b = \log_2 N$.

Собственно, закодированный индекс СУ в виде двоичной строки будет представлять собой один ген хромосомы, а набор из записанных подряд индексов СУ в битовом коде, указывающих расположение СУ во множестве B , будет представлять собой хромосому.

Из-за правила кодирования хромосом ГА, устанавливающим одинаковый размер всех хромосом, приходится за одну итерацию алгоритма рассматривать только постоянное количество СУ.

Оценка целевой функция F реализована следующим образом: гены из проверяемой хромосомы записываются как номер СУ во множество B , и восстанавливается картина размещения СУ в виде матрицы A . На основе таблицы характеристик сенсор-

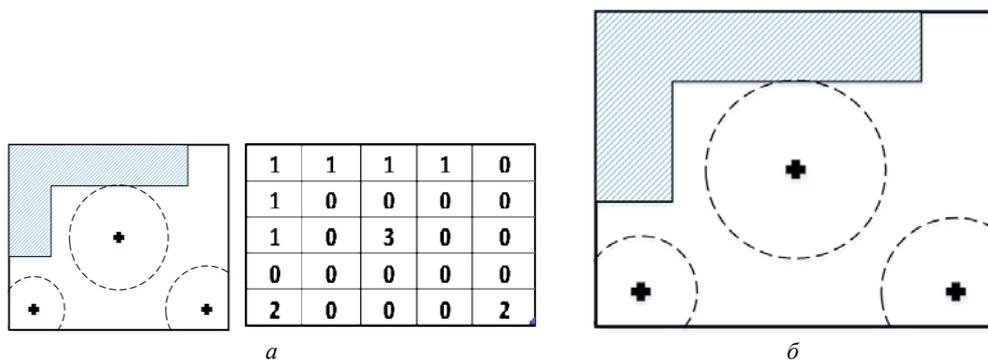


Рис. 2

ных устройств определяются их зоны покрытия. Затем высчитывается суммарная площадь покрытия, которая и является значением целевой функции.

Площадь покрытия определяется следующим образом. Для хромосомы Ch_i , которая представляет собой определенное промежуточное решение, генерируется двумерный булевский массив X , равный размеру коммутационного поля. Назовем его массивом покрытия, который будет иметь аналогичную индексацию. Далее осуществляется последовательный обход по сенсорным устройствам: для каждого СУ в зависимости от радиуса его действия определяются 4 границы в разные стороны ячейки (если полученная граница выходит за пределы матрицы, то ее заменяет крайняя ячейка матрицы). В соответствии с полученными границами совершается обход уже квадратной области и выполняется проверка ячейки на соответствие условию

$$(x - a)^2 + (y - b)^2 \leq R^2, \quad (1)$$

где a, b – координаты расположения СУ в коммутационном поле; x, y – координаты ячейки коммутационного поля; R – радиус действия СУ.

Если для ячейки условие (1) выполняется, то в двумерный булевский массив X с аналогичным индексом записывается «1» – true. Таким образом, создаются «круги покрытия» вокруг СУ, как показано на рис. 3.

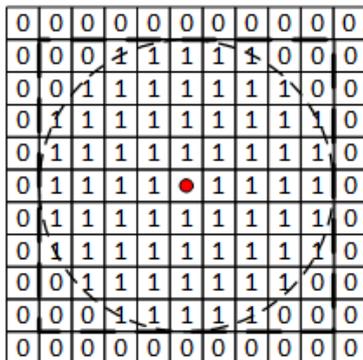


Рис. 3

Тогда значением целевой функции F является сумма элементов массива X , обеспечивающих покрытие размером $N \times N$:

$$F(Ch_i) = \sum_{i=0}^N x_i, \quad x_i \in (0; 1).$$

На каждой итерации выполняется оценка функции приспособленности хромосом и сохраняется лучшее значение. В последующих итерациях лучший кандидат рассматриваемого поколения сравнивается с текущим значением функции

приспособленности, т. е. с лучшим на данный момент решением. Если выигрывает это решение, то оно заменяет худшую хромосому в своем поколении, а если выигрывает лучший кандидат, то он становится новым значением целевой функции.

Функция отбора реализована следующим образом: для каждой хромосомы определяется вероятность p_i отбора хромосомы в родительский пул:

$$p_i = \frac{F(Ch_i)}{\sum_{i=0}^N F(Ch_i)},$$

где $F(Ch_i)$ – значение функции приспособленности i -й хромосомы.

Поскольку $\sum_{i=1}^N p_i = 1$, то процедура отбора

может быть представлена как результат поворота колеса рулетки, где выбранная хромосома относится к выпавшему сектору этого колеса. Таким образом, хромосомы, имеющие более высокую оценку p_i , с большей вероятностью попадают в родительский пул и становятся претендентами на получение потомства.

Скрещивание и мутация в алгоритме реализованы стандартными методами: между случайным образом взятыми парами хромосом, прошедших отбор в родительский пул, производится одноточечное скрещивание. Точка скрещивания выбирается в соответствии с вероятностью скрещивания p_{cr} ($p_{cr} \geq 0.5$). Затем к хромосомам применяется оператор мутации – с вероятностью $p_{mt} = 0.1$ значение гена в хромосоме меняется на противоположное.

В реализации ГА предполагается проверка генов на запрещенные значения – каждый ген в хромосомах полученной популяции проверяется на возможную ошибку. При наличии в гене ошибки он будет заменен на случайно выбранный ген из другой, проверенной хромосомы.

В предлагаемом ГА условие выхода выполняется при достижении одного из двух ограничений:

1. Прошло максимальное число итераций. Оптимальным решением задачи размещения будет считаться лучшая хромосома.

2. Покрытие заданной территории выполнено и найденное решение размещения СУ на этой территории невозможно улучшить.

Таким образом, на каждой последующей итерации будет появляться новое решение, улучшенное с точки зрения оптимального размещения СУ в рассматриваемом коммутационном поле. На рис. 4 показано решение (оптимизация) первоначального варианта со случайным размещением СУ.

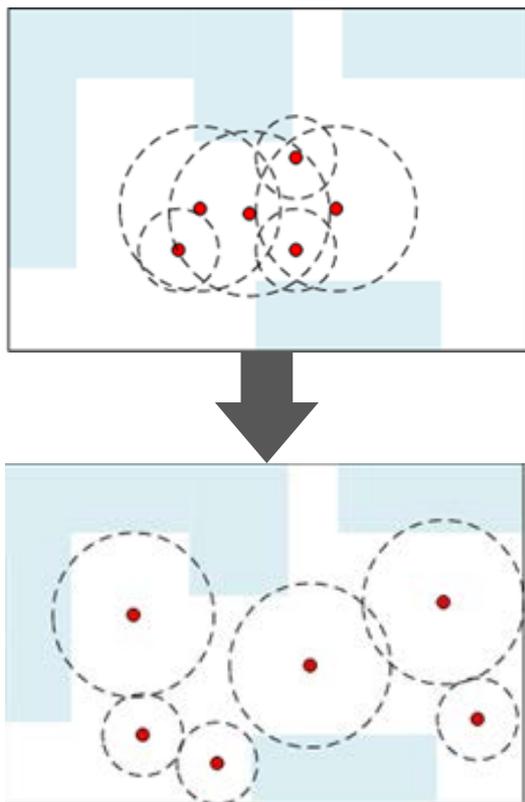


Рис. 4

Предлагаемый ГА размещения возможно реализовать и для трехмерного пространства. В этом случае для описания коммутационного поля массивы A и X будут трехмерными. Для определения, находится ли ячейка внутри радиуса действия СУ при создании массива X , используется уравнение сферы:

$$(x - a)^2 + (y - b)^2 + (z - c)^2 \leq R^2,$$

где a, b, c – координаты СУ в коммутационном поле; x, y, z – координаты ячейки коммутационного поля; R – радиус действия СУ.

Размерность множеств B и C для трехмерного пространства не изменится.

Особенности программной реализации ГА.

Генетический алгоритм размещения СУ в коммутационном поле действия интернета вещей реализован на языке C++. В данной версии языка класс Grid позволяет описать территорию и класс GA, осуществляющий работу алгоритма. Для реализации генетических операторов скрещивания и мутации были задействованы включенные в стандарт C++11 инструменты – классы `uniform_int_distribution` и `uniform_real_distribution`, значения вероятностей для которых генерируются датчиком случайных положительных чисел `mt19937` (алгоритм «Вихрь Мерсенна»).

Класс Grid поддерживает различные размеры коммутационного поля, позволяет указать препятствия, а также установить некий начальный вариант расстановки СУ.

На вход ГА подаются следующие параметры:

- существующая карта первоначального размещения СУ с границами (Grid);
- список сенсорных устройств, включающий номер СУ и количество;
- таблица радиусов действия СУ в виде `map<номер СУ, R>`;
- размер популяции M ;
- вероятность скрещивания p_{cr} ;
- вероятность мутации хромосомы p_{mt} ;
- условие останова алгоритма.

Для визуализации результата работы ГА разработан графический интерфейс с использованием открытой библиотеки wxWidgets. Программа реализована в среде разработки Code:Blocks с использованием плагина wxSmith для написания графического интерфейса. В визуализации отмечаются радиусы действия всех СУ, что позволяет визуально оценить качество решения.

Для эксперимента были выбраны следующие параметры: коммутационное поле размером 100×100 ячеек с произвольно расставленными препятствиями; 7 датчиков с $R = 15$ м; 3 датчика с $R = 30$ м; $M = 12$; $p_{cr} = \geq 0.5$; $p_{mt} = 0.1$; условие останова – количество итерации 100 000.

В итоге было найдено решение, занимающее 98 % плоскости (рис. 5).

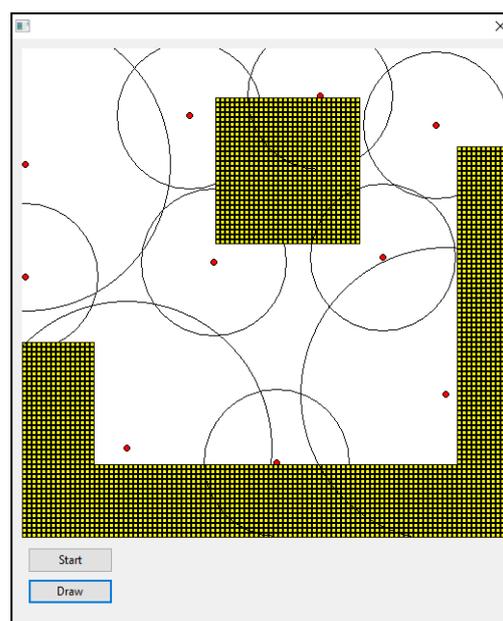


Рис. 5

Подобные эксперименты проводились для различных форм препятствий и разного количества разнородных СУ. Результат работы алгоритма размещения (покрытия заданной площади) близок к 100 %, худшим при 100 различных экспериментах оказалось решение 97.4 % покрытия площади.

Предложен генетический алгоритм для самоорганизованного размещения сенсорных устройств в

заданном коммутационном поле. Генетический алгоритм решения задачи размещения доведен до программной реализации. Эксперименты, проводимые для различных вариантов коммутационного поля и разных радиусов действия сенсорных устройств, показали результаты, близкие к 100 % покрытия территории.

СПИСОК ЛИТЕРАТУРЫ

1. Интернет вещей / А. В. Росляков, С. В. Ваняшин, А. Ю. Гребешков, М. Ю. Самсонов. Самара: Изд-во ПГУТИ, 2014. 340 с.

2. Bonomi F. Fog computing and its role in the internet of things // Proc. of the first edition of the MCC workshop on Mobile cloud computing, USA, 2012. P. 13–16.

3. Гольдштейн Б. С., Кучерявый А. Е. Сети связи пост-NGN. СПб.: БХВ-Петербург, 2014. 160 с.

4. Татарникова Т. М. Структурный синтез центра сопряжения корпоративных сетей // Информационно-управляющие системы. 2015. № 3. С. 92–98.

5. Daniel K. The Silent Intelligence: The Internet of Things. Publisher: DND Ventures LLC, 2013. 156 p.

6. Кутузов О. И., Татарникова Т. М. Подход к оптимизации структуры межсетевого устройства с привлечением генетических алгоритмов // Изв. СПбГЭТУ «ЛЭТИ». 2006. № 11. С. 61–67.

7. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия–Телеком, 2013. 384 с.

8. Кутузов О. И., Татарникова Т. М. Моделирование систем и сетей телекоммуникаций. СПб.: Изд-во РГГМУ, 2012. 136 с.

O. I. Kutuzov, T. M. Tatarnikova
Saint Petersburg Electrotechnical University «LETI»

I. N. Dzubenko
Saint Petersburg State University of Aerospace Instrumentation

SOLUTION OF A LOCATING SENSOR DEVICES PROBLEM IN NETWORKS INTERNET OF THINGS

Presents one of the possible solutions to the problem of placing sensor devices in the switching field of the Internet of things. An algorithm for solving the allocation problem based on evolutionary theory and adaptive behavior of sensory devices is proposed. When implementing a genetic algorithm in the form of software, it will allow sensory devices to self-organize in the switching field of the Internet of things. The commutation field is an inhomogeneous site, in the territory of which natural obstacles in the form of buildings, trees and other objects are located. The correspondence of terms borrowed from genetics to terms of the genetic algorithm is resulted, which allows adapting the algorithm to the solution of the allocation problem. All intermediate and final solutions must have the same code length, which complicates the process of finding a solution. The article describes methods of encoding and decoding solutions. The proposed algorithm is brought to the software implementation, in which the result is visualized as a map of the location area with obstacles and placed sensor devices with radii of their action.

Internet of Things, a sensor device, location problem, NP-hard task, modeling of evolution, genetic algorithm, the objective function
