

ГЕНЕТИЧЕСКИЙ И МУРАВЬИНЫЙ АЛГОРИТМЫ ДЛЯ ЗАДАЧИ РАЗМЕЩЕНИЯ ШЛЮЗОВ В СЕТИ НАВИГАЦИОННЫХ ЗНАКОВ

GENETIC AND ANT ALGORITHMS FOR GATEWAY LOCATION PROBLEM IN AIDS TO NAVIGATION NETWORK

Представлена реализация алгоритма муравьиной колонии и генетического алгоритма для решения задачи оптимального размещения шлюзов в сети навигационных знаков. Критерием оптимальности является энергетическая эффективность сети. Представлены результаты вычислительных экспериментов по работе рассматриваемых алгоритмов.

Abstract. Implementation of genetic algorithm and ant colony algorithm for gateway optimal location problem in aids to navigation network is proposed. Energy efficiency of network is an optimality criterion. Experimental results for considered algorithms are presented.

Ключевые слова: средства навигационного оборудования, система мониторинга, сенсорная сеть, популяция, хромосома, муравей, феромон.

Key words: aids to navigation, monitoring system, sensor network, population, chromosome, ant, pheromone.

СРЕДСТВА навигационного оборудования (СНО) являются важным элементом системы обеспечения безопасности судоходства. Организация удаленного мониторинга эксплуатационных параметров СНО позволяет своевременно обнаружить или предупредить выход из строя аппаратуры навигационных знаков. Для передачи данных от объектов СНО в центр мониторинга, как правило, применяется сотовая связь или технология AIS, реже — спутниковая связь. На данный момент широкое распространение получила технология сенсорных сетей (СС) на базе реактивного протокола маршрутизации Ad-hoc On-demand Distance Vector (AODV). Ключевыми достоинствами данной технологии являются:

— малое потребление энергии узлами сети за счет возможности применения ретрансляции сообщений, режим «сна» и отсутствие необходимости в постоянном поддержании таблиц маршрутизации в актуальном состоянии;

— устойчивость сети к изменениям топологии в результате удаления, добавления или перемещения узлов, которая достигается за счет возможности каждого узла самостоятельно решать задачу маршрутизации.

Детальное описание технологии СС и протокола AODV можно найти в [1]. В случае размещения центра мониторинга вне радиуса действия передатчиков СС появляется необходимость в дополнении сети шлюзами — устройствами, способными передавать данные на большие расстояния. Таким образом, совместное применение сенсорных сетей и шлюзов является перспективным направлением развития систем удаленного мониторинга СНО.

Энергетическая эффективность сети — это рациональное использование энергетического ресурса узлов, позволяющее увеличить время функционирования всей сети. При размещении шлюзов следует учитывать энергию, затрачиваемую каждым узлом на передачу собственных сообщений и ретрансляцию сообщений от других узлов. В соответствии с протоколом AODV данные передаются по кратчайшему маршруту. Например, на рис. 1, а изображена сеть, в которой при передаче сообщения узлом 8 на шлюз 5 задействован узел 9 в качестве ретранслятора данных. В случае существования нескольких кратчайших маршрутов от узла до нескольких шлюзов следует учитывать нагрузку на всех узлах-ретрансляторах каждого кратчайшего маршрута (рис. 1, б).

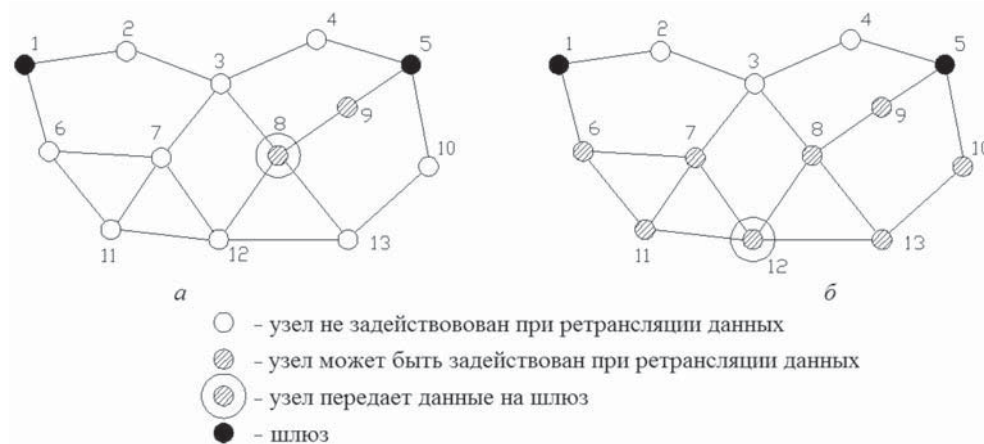


Рис. 1. Энергетическая нагрузка сети при передаче данных узлом.
 а — передача сообщения узлом 8; б — передача сообщения узлом 12

Вычисление максимального количества сообщений $tx_{\max}$, которое может передаваться узлом за один период сбора информации, приведено в работе [2]. Энергетическая эффективность сети будет достигаться, если для каждого узла сети u будет выполняться условие

$$tx_{\max} \geq n_{ret}(u) + 1, \quad (1)$$

где n_{ret} — количество узлов, для которых узел u является ретранслятором.

Таким образом, схема размещения шлюзов непосредственно оказывает влияние на энергетическую эффективность сети. Математическая модель задачи оптимального размещения шлюзов в терминологии целочисленного линейного программирования сформулирована в [2].

При ограничении пропускной способности узлов параметром $tx_{\max}$ каждым шлюзом может обслуживаться некоторое подмножество узлов сети. Предлагаемая формулировка позволяет установить соответствие между задачей оптимального размещения шлюзов и известной NP -полной задачи о минимальном покрытии [3]. Различие данных задач заключается в том, что расчетное подмножество узлов, обслуживаемое шлюзом, не является постоянной величиной и может изменяться. Тем не менее каждую схему размещения шлюзов можно проверить на соответствие требованию (1) за полиномиальное время.

Актуальность рассматриваемой темы исследования заключается в том, что существование точного алгоритма с полиномиальным временем работы для решения NP -полных задач маловероятно. Однако существуют эвристические алгоритмы, в число которых входят генетический и муравьиный алгоритмы, способные давать приближенные решения за приемлемое время для подобного рода задач. В современных исследованиях ведется активное изучение производительности эвристических алгоритмов в отношении многих труднорешаемых задач, однако автором не было найдено сведений об их применении для задачи, сформулированной в данной статье.

Представим сеть навигационных знаков в виде графа $G = (V, E)$, где V — множество вершин, характеризующее навигационные знаки ($|V| = n$ — количество вершин, определяемое порядком множества V), E — множество ребер графа, характеризующее наличие радиоканала между СНО. Наличие или отсутствие ребра между вершинами определяется радиусом действия передатчиков сенсорной сети. Обозначим множество шлюзов сети как $V_p \subset V$, $|V_p| = p$. Схему размещения шлюзов будем кодировать в виде n -мерного вектора бинарных элементов x . Каждый элемент вектора принимает значение 0, если вершина является узлом, или 1, если вершина является шлюзом. Минимизируемой целевой функцией рассматриваемой задачи является количество шлюзов в сети $f_{цф}(x) = p$. Для определения степени «перегруженности» узлов при заданном размещении шлюзов будем использовать функцию пригодности:

$$f_{\Phi\P}(x) = \sum_{u \in V'} n_{ret}(u) + 1 - tx_{\max}, \quad V' = \{u \in V / V_p \mid n_{ret}(u) + 1 > tx_{\max}\}. \quad (2)$$

В основе идеи генетического алгоритма (ГА) лежат механизмы эволюционной модели живых организмов, которые в результате естественного отбора приспособляются к окружающей среде. Вектор бинарных элементов $c_i \in C_t$, где $i = \{1 \dots |C_t|\}$, в соответствии с терминологией ГА будем называть хромосомой или особью, элементы вектора — генами, множество особей текущего поколения — популяцией. Далее приведена реализация генетического алгоритма для решения задачи размещения шлюзов:

Шаг 1. Инициализация параметров. Установить поколение $t \leftarrow 1$ и сформировать исходную популяцию C_t .

Шаг 2. Вычислить приспособленность текущего поколения t по значению $f_{\text{ЦФ}}(c_i)$ и $f_{\Phi\P}(c_i)$ $\forall c_i \in C_t$. Применить оператор скрещивания для получения популяции потомков: $C'_t \leftarrow \text{Cross}(C_t)$.

Шаг 3. Применить оператор мутации $\forall c_i \in C'_t$ с вероятностью q : $C'_t \leftarrow \text{Mut}(C'_t, q)$.

Шаг 4. Вычислить приспособленность популяции потомков по значению $f_{\text{ЦФ}}(c_i)$ и $f_{\Phi\P}(c_i)$ $\forall c_i \in C'_t$. Применить оператор рекомбинации: $C_{t+1} \leftarrow \text{Recomb}(C_t, C'_t)$.

Шаг 5. Если $\forall c_i \in C_{t+1}$, суммарные значения целевой функции и функции пригодности $f_{\Sigma}(c_i)$ равны, то вывод любого $c_i \in C_{t+1}$, для которого выполняется условие $f_{\Phi\P}(c_i) = 0$ и Конец. Иначе $t \leftarrow t + 1$ и переход на Шаг 2.

Рассмотрим данный алгоритм подробнее. Для формирования исходной популяции вычисляется значение минимально возможного количества шлюзов $p_{\min}$ в сети по формуле из [2]. Каждая хромосома исходной популяции создается путем случайного размещения $p_{\min}$ шлюзов в сети. Функция *Cross* обеспечивает создание хромосом потомков путем обмена генетического материала родительских хромосом. В данной реализации алгоритма для получения двух потомков используются две родительские хромосомы, а также применяется одноточечный оператор скрещивания, где точка разреза хромосом определяется случайным образом. Например, если для родительских хромосом $\{111111\}$, $\{000000\}$ определяем точку разреза между третьим и четвертым генами, то получаем следующих потомков: $\{111000\}$, $\{000111\}$. Выбор особей, участвующих в формировании потомства, производится с использованием модели «колеса рулетки», в которой каждой хромосоме выделяется сектор, по размеру пропорциональный значению ее приспособленности. Поскольку оценка хромосом производится по значениям $f_{\text{ЦФ}}(c_i)$ и $f_{\Phi\P}(c_i)$ при $i = \{1 \dots |C_t|\}$, целесообразно использовать отдельное «колесо рулетки» для каждой функции. Родитель 1 определяется по количеству шлюзов, а родитель 2 — по значению функции пригодности (2). Подобный подход обеспечивает разнообразие генетического материала при скрещивании, что уменьшает вероятность преждевременной сходимости алгоритма в локальных оптимумах. Функция *Mut* выполняет частичное преобразование хромосомы случайным образом с вероятностью q . В данном алгоритме происходит случайный выбор позиции гена, подвергаемого мутации, и инвертируется его значение. Также если для хромосомы i выполняется неравенство $f_{\text{ЦФ}}(c_i) < p_{\min}$, то происходит ее корректировка путем случайного добавления $[p_{\min} - f_{\text{ЦФ}}(c_i)]$ шлюзов. Функция *Recomb* выполняет формирование новой популяции из числа хромосом текущей популяции и хромосом потомков. В данном алгоритме применяется элитный отбор. Для каждой хромосомы из C_t и C'_t вычисляется суммарное значение $f_{\Sigma}(c_i) = f_{\text{ЦФ}}(c_i) + f_{\Phi\P}(c_i)$. В новое поколение $t + 1$ переходят $|C_t|$ хромосом с минимальным значением $f_{\Sigma}(c_i)$. Подобный подход позволяет сохранить заданный размер популяции и не потерять накопленный в процессе эволюции генетический материал в случае неудачного применения операторов скрещивания и мутации. В качестве критерия останова выбрано условие, при котором в поколении $t + 1$ все хромосомы имеют одинаковые значения $f_{\Sigma}(c_i)$, то есть алгоритм не может улучшить получаемые решения. Подробное описание генетических операторов и методики их применения можно найти в [4].

Концепция алгоритма муравьиной колонии (АМК) основана на поведении живых муравьев. При движении от гнезда к источнику пищи муравей выделяет специальное вещество — феромон.

Как правило, каждый муравей делает выбор в пользу пути, на котором концентрация феромона более высокая, чем на других возможных направлениях. Чем короче путь, тем больше муравьев успевают по этому пути пройти и соответственно оставить больше феромона. В результате со временем все муравьи начинают передвигаться по кратчайшему пути. Поиск оптимального размещения шлюзов построен на идее непрямого обмена информацией о количестве искусственного феромона на узлах между муравьями. «Путь», пройденный муравьем i , следует рассматривать как вектор бинарных элементов a_i , характеризующий схему размещения шлюзов в сети, где V_p^i — множество шлюзов, выбранных муравьем i , $|V_p^i| = p_i$. Далее приведена реализация АМК:

Шаг 1. Инициализация параметров. Установить счетчик итераций $t \leftarrow 1$ и исходное количество феромона на узлах $\tau_v(t) \leftarrow 1, \forall v \in V$. Каждый узел сети $v \in V$ сделать отправной точкой пути a_i для одного муравья i : $V_p^i \leftarrow \{v\}$. Для каждого муравья $i = \{1 \dots n\}$ установить количество шлюзов $p_i \leftarrow 1$.

Шаг 2. Для каждого муравья i , для пути которого выполняется условие $f_{\text{ФП}}(a_i) > 0$ или $p_i = 0$, выполнить один шаг на вершину $v \in V / V_p^i$: $a_i \leftarrow \text{AntStep}(a_i)$; $p_i \leftarrow p_i + 1$.

Шаг 3. Если остались муравьи, для которых выполняется неравенство $f_{\text{ФП}}(a_i) > 0$, то осуществить переход на Шаг 2. Иначе, обновить информацию о количестве феромонов на узлах: $\tau_v(t+1) \leftarrow \text{PherUpdate}(\tau_v(t)) \forall v \in V$.

Шаг 4. Если все значения p_i при $i = \{1 \dots n\}$ равны, то вывод лучшего значения a_p , полученного муравьями за все итерации, и Конец. Иначе, установить всех муравьев на исходные позиции $a_i \leftarrow 0$; $p_i \leftarrow 0$; $V_p^i \leftarrow \emptyset$ при $i = \{1 \dots n\}$, выполнить $t \leftarrow t + 1$ и перейти на Шаг 2.

Рассмотрим алгоритм более подробно. При инициализации переменных установка начального количества феромона $\tau_v(t)$ на узлах необходима для корректного выполнения функции AntStep при $t = 1$. Маршрут муравья считается пройденным, если выполняется условие $f_{\text{ФП}}(a_i) = 0$. Один шаг муравья выполняет функция AntStep , которая модифицирует вектор a_i преобразованием узла сети в шлюз. Вероятность преобразования q_v^i для каждого узла вычисляется по формуле

$$q_v^i = \frac{[\tau_v(t)]^\alpha \left[\frac{1}{f_{\text{ФП}}(a_{i,v}) + 1} \right]^\beta}{\sum_{u \in V / V_p^i} [\tau_u(t)]^\alpha \left[\frac{1}{f_{\text{ФП}}(a_{i,u}) + 1} \right]^\beta}, \quad v \in V / V_p^i, \quad (3)$$

где $f_{\text{ФП}}(a_{i,v})$ — функция пригодности для вектора a_i в случае выбора вершины v в качестве шлюза; α, β — константы, определяющие «вес» соответствующих параметров.

Для выбора преобразуемого узла применяется модель «колеса рулетки», где размеры секторов пропорциональны значениям q_v^i каждой вершины. После выбора узла v выполняются следующие операции: $f_{\text{ФП}}(a_i) \leftarrow f_{\text{ФП}}(a_{i,v})$; $V_p^i \leftarrow V_p^i \cup \{v\}$. Функция PherUpdate изменяет информацию о концентрации феромона на вершинах по формуле

$$\tau_v(t+1) = \psi \tau_v(t) + \sum_{i \in I_v} \frac{Q}{p_i}, \quad v \in V, \quad (4)$$

где I_v — множество муравьев i , для которых $v \in V_p^i$;

ψ — коэффициент испарения феромона;

Q — количество феромона, откладываемого муравьем при прохождении одного пути от начала до конца.

В процессе выполнения алгоритма узлы, на которых количество феромона близко к нулю, с наименьшей вероятностью будут включены в путь муравьев. Со временем всем муравьям будет требоваться одинаковое количество шагов по наиболее обогащенным феромоном узлам для прохождения своего маршрута, что будет сигнализировать о схождении алгоритма. Данное условие выбрано в качестве критерия останова. Различные варианты модификаций алгоритма можно найти в [5].

Для проведения вычислительного эксперимента была разработана программа в среде Borland C++ Builder 6. Эксперимент проводился на ПК с процессором Intel Atom с тактовой частотой 1,67 ГГц. Серия запусков алгоритмов проводилась на связных графах, сгенерированных случайным образом. Также был разработан точный алгоритм на основе метода последовательного перебора (ПП), который вычисляет $f_{\text{ФП}}(x)$ для всех возможных вариантов размещения p шлюзов, начиная с $p = 2$. Инкремент значения p выполняется до тех пор, пока не найдется вариант с $f_{\text{ФП}}(x) = 0$. Для каждого значения параметров n и tx_{max} было выполнено 100 запусков ГА и АМК. В табл. 1 приведено минимальное время t_{min} , потраченное каждым алгоритмом на получение оптимального решения по значению $f_{\text{ЦФ}}(x)$, а также $\epsilon_{\text{опт}}$ — доля оптимальных из общего числа решений, полученных алгоритмом.

Ключевым достоинством ГА по сравнению с АМК можно назвать более высокую скорость получения решений. Однако в большинстве случаев АМК вырабатывает больше оптимальных решений, о чем свидетельствуют значения $\epsilon_{\text{опт}}$ алгоритмов. С ростом количества узлов в сети общая скорость получения решений в АМК падает быстрее, чем в ГА. Подобное явление можно объяснить тем, что для модификации значения $f_{\text{ФП}}(a_i)$ на одном шаге одного муравья требуется выполнить $n - p_i$ вычислений значения $f_{\text{ФП}}(a_{i,v})$, то есть трудоемкость каждого шага муравья растет пропорционально значению n . В генетическом алгоритме для модификации $f_{\text{ФП}}(c_i)$ требуется выполнить два вычисления $f_{\text{ФП}}(c_i)$: для хромосомы текущей популяции C_i и хромосомы потомка из C'_i .

Таблица 1

Результаты вычислительного эксперимента для ГА и АМК

n	tx_{max}	$f_{\text{ЦФ}}(x)$, усл. ед.	ПП	ГА		АМК	
			t_{min} , с	$\epsilon_{\text{опт}}$, %	t_{min} , с	$\epsilon_{\text{опт}}$, %	t_{min} , с
40	2	7	2481,165	33	0,593	34	8,424
40	3	6	106,268	97	0,765	99	5,179
40	4	4	1,279	100	0,468	99	1,903
60	3	7	12292,630	40	1,575	52	34,008
60	4	5	489,266	15	1,467	12	27,565
60	5	4	7,941	47	0,780	73	13,603
80	5	5	1820,142	16	1,388	38	52,573
80	6	4	181,538	28	2,012	64	43,945
80	7	4	45,911	69	1,591	94	34,804
100	10	5	1823,574	76	2,355	99	109,123
100	11	4	325,184	25	2,481	38	110,027
100	12	4	70,825	59	2,325	100	72,119

Помимо способов реализации генетических операторов, на производительность ГА влияют такие параметры, как вероятность мутации q и размер популяции $|C_t|$. Высокая вероятность мутации может не позволить «сильным» хромосомам занять доминирующее положение в популяции, и алгоритм может не сойтись, а низкая вероятность приведет к сокращению проверяемой области пространства решений и преждевременной сходимости алгоритма. Большой размер популяции снижает общую скорость работы алгоритма, а малый размер сокращает проверяемую область пространства решений. Результаты экспериментов получены при следующих значениях параметров: $q = 0,6$; $|C_t| = 150$.

В алгоритме муравьиной колонии значения параметров α , β , Q , ψ подбираются экспериментальным путем, что является одним из недостатков алгоритма. При $\alpha = 0$ алгоритм становится

«жадным», а при $\beta = 0$ муравьи начинают перемещаться «вслепую». Высокая скорость испарения и малый объем откладываемого муравьями феромона приводят к потере решений, найденных колонией. Низкая скорость испарения и большой объем откладываемого муравьями феромона приводят к преждевременной сходимости алгоритма в локальных оптимумах. Таким образом, крайне важно найти баланс между данными параметрами. Результаты экспериментов получены при следующих значениях параметров: $\alpha = 2$; $\beta = 1$; $Q = 2$; $\psi = 0,6$.

В результате проведенного исследования можно сделать следующие выводы. Разработанные алгоритмы позволяют получать оптимальные решения задачи размещения шлюзов за приемлемое время, что свидетельствует о перспективности их изучения и применения. Также определены значения параметров алгоритмов, прямым образом влияющих на их производительность в данной реализации. Применение разработанных алгоритмов позволит рационализировать расход энергетического ресурса каждого навигационного знака в сети и, следовательно, повысить эффективность использования технологии сенсорных сетей в системах мониторинга СНО. Представленная работа ограничена рассмотрением лишь основных принципов ГА и АМК. В дальнейшем предстоит изучение влияния различных эволюционных моделей и модификаций генетических операторов на производительность ГА. Также следует рассмотреть влияние на производительность АМК различных дополнительных механизмов поиска, например элитарной или ранговой системы муравьев.

Список литературы

1. *Dargie W. Fundamentals of wireless sensor networks: theory and practice* / W. Dargie, C. Poellabauer. — Singapore: Markono, 2010. — 311 p.
2. *Шейкин Т. Ю.* Применение технологии сенсорных сетей в системе мониторинга судового навигационного оборудования / Т. Ю. Шейкин // Журнал Университета водных коммуникаций. — 2013. — Вып. 3 (19).
3. *Гэри М.* Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. — М.: Мир, 1982. — 416 с.
4. *Гладков Л. А.* Генетические алгоритмы / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик; под ред. В. М. Курейчика. — 2-е изд., испр. и доп. — М.: Физматлит, 2006. — 320 с.
5. *Dorigo M. Ant Colony Optimization* / M. Dorigo, T. Stutzle; Massachusetts Institute of Technology, 2004. — 321 p.