

Д.В. Заруба, Э.В. Кулиев, Д.Ю. Запорожец, М.М. Семенова

РАЗРАБОТКА БИОЭВРИСТИК ДЛЯ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДСИСТЕМЫ ПРИНЯТИЯ ЭФФЕКТИВНЫХ РЕШЕНИЙ NP-ТРУДНЫХ И NP-СЛОЖНЫХ КОМБИНАТОРНО-ЛОГИЧЕСКИХ ЗАДАЧ НА ГРАФАХ*

Статья посвящена решению новых актуальных проблем, возникших в условиях современного развития информационных и нанометровых технологий в области проектирования, а также разработке новых инновационных методов, обеспечивающих получение эффективных решений за полиномиальное время. В статье рассматривается проблема решения NP-сложных задач. Приведено описание процедуры измерения сложности задачи. Описаны особенности NP-трудных и NP-сложных комбинаторно-логичеких задач. Приведены основные различия между задачами, а также проблемы, с которыми приходится сталкиваться при решении такого вида задач. Представлена общая схема принятия решений, состоящая из формулировки проблемы; принятия решения; сигнала в автоматических системах и обратной связи. На втором этапе (формирование и выбор вариантов решений) решение основывается на биоинспирированном алгоритме поиска решений задачи коммивояжера. Для решения поставленной задачи был разработан модифицированный биоинспирированный алгоритм, основанный на поведении муравьиной колонии. В отличие от других методов оптимизации, метаэвристические алгоритмы могут находить глобальные оптимальные решения для задач, где существует много локальных решений из-за их случайного характера. Эти причины привели к широкому использованию таких алгоритмов при решении различных задач оптимизации. Биоинспирированные алгоритмы становятся новой революцией в области решений оптимизационных задач. Представлена постановка задачи коммивояжера, а также решение поставленной задачи на основе муравьиного алгоритма. Алгоритмы, такие как генетические алгоритмы и PSO могут быть очень полезными, но они все еще имеют некоторые недостатки в решении проблем мультимодальной оптимизации. Эти алгоритмы способны находить оптимальные решения независимо от физической природы проблемы. В рамках экспериментальных исследований был произведен анализ работы биоинспирированных алгоритмов: алгоритм стаи летучих мышей бактериальный алгоритм и муравьиный алгоритм.

Сложность задачи; NP-задачи; NP-полные задачи на графах; биоинспирированные алгоритмы; муравьиный алгоритм; алгоритм стаи летучих мышей.

D.V. Zaruba, E.V. Kuliev, D.Yu. Zaporozhets, M.M. Semenova

DEVELOPMENT OF BIOHEURISTICS FOR CREATING AN INTELLECTUAL SUBSYSTEM FOR MAKING EFFECTIVE DECISIONS OF NP-HARD AND NP-DIFFICULT COMBINATORIAL-LOGICAL PROBLEMS ON GRAPHS

The article is devoted to the solution of new topical problems that have arisen in the conditions of the modern development of information and nanometer technologies in the field of design, as well as the development of new innovative methods that provide effective solutions in polynomial time. The article deals with the problem of solving NP-hard problems. The description of the procedure for measuring the complexity of the problem is presented the features of NP-hard and NP-difficult combinatorial logic problems are described. The main differences between the tasks are presented, as well as the problems that one has to face when solving this type of task. The general decision-making scheme is presented, consisting of the problem formulation; decision-making; signal in automatic systems and feedback. At the second stage (formation and selection of solutions), the solution is based on a bioinspired algorithm for finding solutions to the traveling salesman problem. To solve this problem, a modified bioinspired algorithm based on the behavior

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-01-00059.

of an ant colony was developed. Unlike other optimization methods, metaheuristic algorithms can find global optimal solutions for problems where there are many local solutions due to their random nature. These reasons have led to the widespread use of such algorithms in solving various optimization problems. Bioinspired algorithms are becoming a new revolution in the field of solving optimization problems. The statement of the traveling salesman problem is presented, as well as the solution of the problem on the basis of the ant algorithm. Algorithms such as genetic algorithms and PSO can be very useful, but they still have some disadvantages in solving multimodal optimization problems. These algorithms can find optimal solutions regardless of the physical nature of the problem. In the framework of experimental studies, the analysis of the work of bioinspired algorithms was carried out: the algorithm of a flock of bats, the bacterial algorithm and the ant algorithm.

Three-dimensional modeling; three-dimensional integration; placement; LSI; VLSI; genetic algorithm; evolution; bioinspired algorithm.

Введение. Современные алгоритмы оптимизации часто вдохновлены природой, как правило, основаны на роевом интеллекте. Данные методы привлекли многих ученых-исследователей, лиц, принимающих решения, и практикующих исследования в качестве мощных интеллектуальных вычислительных методов для решения ряда сложных реальных задач [1]. Эти алгоритмы можно классифицировать на алгоритмы детерминированной и стохастической оптимизации. Детерминированные алгоритмы всегда выдают одинаковые выходные данные для определенных входных данных. Эти алгоритмы часто используются как локальные алгоритмы поиска. В отличие от детерминированных алгоритмов, стохастические алгоритмы имеют случайные компоненты и дают разные выходные данные для конкретных входных данных. Многие метаэвристические алгоритмы реализуют некоторую форму алгоритмов стохастической оптимизации. Следующим этапом развития оптимизационных алгоритмов стали алгоритмы оптимизации роя частиц (PSO). Они были основаны на роящем поведении птиц и рыб, и эта мультиагентная система может иметь роевой или групповой интеллект [1, 2–5].

Оптимизация роя частиц (PSO) – это метод эволюционных вычислений, предложенный Кеннеди и Эберхартом в 1995 г. [1], возникшие в результате моделирования хищничества и другого поведения стай птиц и косяков рыб. Решение каждой задачи оптимизации в алгоритме похоже на «частицу» в пространстве поиска. Алгоритм роя частиц случайным образом генерирует начальный рой и придает каждой частице случайную скорость. В процессе оптимизации частицы регулируют скорость и траекторию в соответствии с опытом самих себя и товарищей, так что весь плавание содержит возможность лететь в лучшую область поиска. Благодаря небольшому количеству параметров и простоте реализации, PSO широко используется во многих областях, таких как оптимизация функций, обучение нейронных сетей, управление нечеткими системами, распознавание образов и инженерные приложения. Однако, алгоритм PSO по-прежнему имеет такие проблемы, как преждевременное попадание в локальный оптимум при решении сложных мультимодальных задач. Чтобы улучшить решающую способность оптимизации роя частиц, исследователи предложили такие методы, как корректировка инерционных параметров алгоритма роя частиц, включая динамические политики и адаптивные методы, факторы обучения и социальные факторы [2], стратегию поиска окрестностей для более эффективного изучения окрестностей текущего населения [3], принятие механизма обмена информацией для увеличения разнообразия населения и предотвращения преждевременной конвергенции алгоритмов [4], а также интеграции с другими алгоритмами, такие как комбинация алгоритма оптимизации роя частиц и иммунного алгоритма, генетического алгоритма и алгоритма искусственной пчелиной семьи [5].

В PSO частица изменяет свое положение и векторы скорости, отслеживая свое индивидуальное оптимальное решение, а также глобальное оптимальное решение роя. Такой вид обновления дает возможность глобального поиска PSO на ранней стадии эволюции и способность локальной конвергенции в конце эволюции.

Эволюция роя выглядит следующим образом:

$$v_{i,j}^{G+1} = w_{ps0} v_{i,j}^G + c_1 r_1 (p_{i,j} - x_{i,j}^G) + c_2 r_2 (p_{g,j} - x_{i,j}^G). \quad (1)$$

$$x_{i,j}^{G+1} = x_{i,j}^G + v_{i,j}^{G+1}, \quad (2)$$

где оптимальное положение, которое когда-либо находила i^{th} частица, представлено, как $P_i = [p_{i,1}, p_{i,2}, \dots, p_{i,D_{ps0}}]^T$. Верхний индекс «G» обозначает порождение эволюции, $0 < w_{ps0} < 1$ обозначает вес инерции, который показывает, насколько частица зависит от ее собственной скорости, и $c_1, c_2 \in [0,4]$ обозначает фактор обучения; обозначает влияние индивидуального опыта на i частицу и влияние опыта группы на i частицу. $c_1 i c_2 i r_1, r_2 \in [0,1]$ обозначает равномерно распределенные случайные числа, генерируемые в каждом поколении эволюции.

Позволять $\zeta_1 = c_1 r_1 + c_2 r_2$, а также $\zeta_2 = c_1 r_1 p_{i,j} + c_2 r_2 p_{g,j}$, то имеем

$$x_{i,j}^{G+2} + (\zeta_1 - w_{ps0} - 1) x_{i,j}^{G+1} + w_{ps0} x_{i,j}^G = \zeta_2. \quad (3)$$

Если движение частицы рассматривается как непрерывный процесс, уравнение 3 можно рассматривать как классическое неоднородное дифференциальное уравнение второго порядка без члена скорости.

Все алгоритмы могут работать лишь с конечными объектами. Под конечными объектами понимаются объекты, которые можно «эффективно» закодировать словами в некотором алфавите. Эффективность кодирования означает, что по коду объекта можно вычислить все ключевые характеристики. Например, по коду графа можно вычислить, сколько в нём вершин и соединены ли две данные вершины ребром, по коду матрицы можно вычислить её размер и число в любой ячейке, и т.д. Как правило, кодирование не представляет особых проблем: с обычными строками, описывающими объекты, можно работать алгоритмически. Например, натуральные числа представляются своей двоичной (или десятичной) записью, вектора и матрицы представляются как цепочки записей чисел с разделителями, граф можно представить в виде списка рёбер или в виде матрицы смежности, и т.д. [6]. Во всех примерах считается, что все объекты уже закодированы как слова в некотором алфавите, и не будут отдельно обозначать код объекта. Разумеется, исключением будут бесконечные объекты, такие как действительные числа или геометрические фигуры.

Вычислительная модель называется эффективной (соответственно, предположительно эффективной), если она способна предоставлять решения за полиномиальное время для трудноразрешимых проблем (соответственно, NP-полных проблем) [3]. Термин «предположительно эффективный» относится к тому факту, что, по общему мнению, если P = NP, то каждая NP-полная проблема неразрешима и, следовательно, любая предположительно эффективная модель будет эффективной.

Пусть фиксированы конкретный алгоритм и конкретная мера его сложности, например время работы. Как из сложности работы на каждом входе получить сложность всего алгоритма? Во-первых, для слов одинаковой длины можно использовать два подхода: подсчитывать сложность в худшем случае, измеряемую как максимум по всем входам данной длины, и сложность в среднем, для которой эти величины так или иначе усредняются. Возможно также изучение сложности в «типичном» случае: на входах, которые обычно встречаются на практике. Но та-

кой подход трудно формализовать. Во-вторых, разных длин тоже бесконечно много, поэтому изучаются асимптотическое поведение сложной меры при длине, стремящейся к бесконечности. У такого подхода есть недостатки, ведь асимптотическое поведение может быть достигнуто при нереальных значениях длины входа. Как правило, такого всё же не случается.

Во многих случаях единицы измерения очевидны из определения: число использованных случайных битов, длина подсказки, число запросов и т. д. В других случаях выбор бинарный, поэтому никаких единиц нет. Например, квантовые вычисления, интерактивность или возможность адаптивных запросов либо доступны, либо нет. А вот к определению времени работы и памяти возможны разные подходы.

Постановка задачи. Переборные задачи, нацеленные на поиск оптимального варианта, называют задачами комбинаторной оптимизации.

Дадим формальную постановку задачи оптимизации. Дано конечное множество X и числовая функция $U(x)$ на этом множестве. Эту функцию называют целевой. Требуется найти такой $x^* \in X$, что $U(x^*)$ будет наименьшим, т.е. $U(x^*) \leq U(x)$ для всех $x \in X$. Вариант постановки задачи, когда требуется найти точки максимума целевой функции, легко сводится к поиску точек минимума функции U [1, 3, 8–10].

Цель оптимизации - найти наилучший приемлемый ответ, учитывая ограничения и потребности проблемы. Для проблемы могут быть разные ответы, которые определяются как целевая функция для их сравнения и выбора оптимального решения. Выбор этой функции зависит от характера проблемы. Например, время или стоимость поездки – одна из общих целей оптимизации транспортной сети. Однако выбор подходящей целевой функции – один из наиболее важных шагов оптимизации. Иногда при оптимизации нескольких целей одновременно такие задачи оптимизации, которые включают несколько целевых функций, называются многокритериальными задачами. Самый простой способ справиться с этими проблемами – сформировать новую целевую функцию в виде линейной композиции основных целевых функций, в котором эффект каждой функции определяется присвоенным ей весом. Каждая задача оптимизации имеет ряд независимых переменных, которые называются проектными переменными, которые представлены вектором x размером n . Целью оптимизации является определение переменных проекта таким образом, чтобы целевая функция была количественной или оптимальной. Различные задачи оптимизации делятся на две категории:

а) Задачи оптимизации без ограничений: в этих задачах цель состоит в том, чтобы сделать целевую функцию наибольшей или наименьшей без каких-либо ограничений на проектные переменные.

б) Задачи оптимизации с ограничениями: оптимизация в большинстве прикладных задач выполняется в соответствии с ограничениями, а в отношении поведения и производительности системы поведенческие ограничения и ограничения в физике и геометрии задачи называются геометрическими или боковыми ограничениями.

Уравнения, представляющие ограничения, могут быть равными или неравными, и в каждом случае метод оптимизации отличается. Однако ограничения определяют приемлемую область дизайна. Задача коммивояжера (TSP) высоко ценится из-за ее различных приложений при тестировании эффективности новых алгоритмов и преобразовании других задач в этот алгоритм. В этой, казалось бы, простой проблеме продавец переезжает со склада и должен вернуться на склад после того, как однажды встретится со всеми покупателями. Задача – максимально сократить пробег. Проблема TSP может рассматриваться с разных точек зрения. Например, эта проблема исследуется для измерения эффективности новых алгоритмов, потому что, хотя эта проблема имеет жесткую структуру в теории сложно-

сти, большинство алгоритмов легко применимы к ней, и, таким образом, эффективность алгоритмов можно исследовать по сравнению с другими алгоритмами. Кроме того, другие проблемы можно превратить в проблемы и решить. Таким образом, шаги для решения некоторых проблем могут быть сокращены с помощью этой процедуры, а с использованием высокоэффективных алгоритмов для решения проблемы TSP основная проблема может быть решена более качественно.

Задача коммивояжёра может быть поставлена как задача оптимизации. Задача коммивояжера (TSP), типичная трудная задача недетерминированного полинома (NP), использовалась во многих инженерных приложениях. Генетические алгоритмы полезны для NP-сложных задач, особенно для задачи коммивояжера.

В качестве множества X достаточно взять S_n (множество перестановок n -элементного множества), а в качестве целевой функции $U(x)$ – длину замкнутой ломаной, проходящей через n заданных точек в порядке, заданной перестановкой $x \in X$.

Для решения задачи поиска точки минимума функции придумано множество методов. Например, для дифференцируемых функций U , определённых на числовом множестве X , как известно, точки минимума (если они есть) следует искать среди критических точек U , то есть таких x , что $U'(x)=0$.

Для решения поставленной задачи был разработан биоинспирированный алгоритм, основанный на поведении муравьиной колонии.

Классом NP называется множество языков L , для которых существует функция $V(x, s)$ с булевыми значениями, вычисляемая за полиномиальное время от длины первого аргумента, такая что:

- ◆ Если $x \in L$, то $\exists s V(x, s) = 1$;
- ◆ Если $x \notin L$, то $\forall s V(x, s) = 0$.

Второй вход s часто называют сертификатом, а функцию V – верификатором. Таким образом, сертификат удостоверяет, что $x \in L$, а верификатор проверяет правильность сертификата. Для слов из языка подходящий сертификат должен существовать, а для слов не из языка все сертификаты должны отвергаться. Иначе говоря, класс NP – это класс языков, принадлежность к которым можно быстро доказать, а P – класс языков, принадлежность к которым можно быстро выяснить.

Примерами языков из NP служит такая задача:

SAT = { ϕ | ϕ – выполнимая булева формула}. (Выполнимость означает, что формула равна 1 на некотором входе). В данном случае сертификатом будет выполняющий набор, а верификатор проверит, что значение формулы на этом наборе действительно равно 1.

NP-полные задачи на графах. Задачи коммивояжера, хорошо известна как NP-полная задача, вычислительная сложность которой возрастает экспоненциально за счет увеличения количества городов. Небольшой модификацией можно получить NP-полноту и задач о гамильтоновом цикле в ориентированном и неориентированном варианте. Ещё одна известная задача о путях в графе – задача коммивояжера (traveling salesman problem). В этом случае дан не просто граф, а граф с весами, т.е. неотрицательными числами на каждом ребре. Путём коммивояжера называется путь минимального суммарного веса, проходящий хотя бы один раз через каждую вершину. Вершины интерпретируются как города, рёбра – как дороги между городами, веса – как длины дорог. Коммивояжёр должен посетить все города, проехав как можно меньше [11].

При формулировке задачи коммивояжера, в некоторых источниках, требуют, чтобы граф обязательно был полным. Ясно, что это не влияет на суть задачи: все отсутствующие рёбра можно провести, но сопоставить им очень большие веса, так чтобы оптимальный путь через них заведомо не проходил. В рассуждении об NP-полноте можно взять все рёбра, отсутствующие в исходном графе, с весом

больше 1: на вывод это не повлияет. При этом веса рёбер образуют метрику, т.е. для них выполняется неравенство треугольника, соответственно, метрическая задача коммивояжёра также будет NP-полной. Можно показать, что даже евклидова задача коммивояжёра, в которой все вершины лежат в евклидовом пространстве, а расстояния получаются при помощи обычной евклидовой метрики, является NP-полной. Конструкция работает уже в двумерном пространстве, т.е. на декартовой плоскости.

Для решения задач подобного типа были созданы системы поддержки принятия решений (СППР), которые являются системами обработки информации для интерактивной поддержки работы в процессе принятия решений.

Решение задачи коммивояжёра биоинспирированным подходом. Поскольку решение задачи коммивояжёра является NP-полной проблемой, большинство эвристических алгоритмов не обладают высоким качеством для решения задачи коммивояжёра, поэтому многие исследования были направлены на метаэвристику, которая дает хороший тур, если не оптимальный. Поскольку метаэвристические подходы очень эффективны для ухода от локального оптимума, они являются одним из лучших алгоритмов для решения задач комбинаторной оптимизации. Эвристические и метаэвристические алгоритмы могут получить более качественное решение по сравнению с точным алгоритмом. В последнее десятилетие решение задачи коммивояжёра хорошо изучена многими биоинспирированными подходами, такими как генетический алгоритм, оптимизация колоний муравьев, оптимизация роя частиц, поиск с кукушкой и алгоритм светлячков.

В отличие от других методов оптимизации, метаэвристические алгоритмы могут находить глобальные оптимальные решения для задач, где существует много локальных решений из-за их случайного характера. Эти причины привели к широкому использованию таких алгоритмов при решении различных задач оптимизации.

В статье разработан биоинспирированный подход к решению комбинаторно-логических задач на основе метода роевого интеллекта, включающий в себя поведения муравьиной колонии.

Муравьиный алгоритм на основе популяции основан на поведении настоящих муравьев в поисках пищи. Когда муравей ищет пищу в своем гнезде, он формирует маршрут, основанный на количестве феромона на каждом краю, через который он прошел. Каждый муравей будет откладывать феромон на каждом пройденном краю во время поиска, так что уровень феромонов на этих посещенных краях будет увеличен, а уровень феромонов на не посещенных краях будет снижен. Другие муравьи будут искать свои собственные маршруты в соответствии со своим опытом, основанным на уровнях феромонов в краях, которые они выбирают для путешествия. Чем выше уровень феромона на краю, тем более он привлекателен для муравья. Чем больше муравьев проходило по краю, тем выше уровень феромонов на этом краю. Количество осажденных феромонов зависит от качества раствора. В алгоритме оптимизации муравьиной колонии количество отложенного феромона косвенно пропорционально длине тура. Чем меньше длина тура, тем больше феромона, заложенного на пути. Феромон также распадается по пути со временем. Это позволяет менее пройденным путям иметь более низкий след феромонов, поскольку феромоны со временем испаряются. Процедура такого поискового поведения может быть применена для решения задачи коммивояжёра.

Алгоритм муравьиной колонии – это алгоритм случайного поиска. В этом алгоритме решение проблемы абстрагируется в последовательности перехода состояний из начального состояния в целевое состояние в дискретном пространстве состояний. Оптимальное решение задачи соответствует удовлетворению критерия

оптимальной оценки последовательности перехода состояний. Значение интенсивности феромона на пути является основой для перехода муравьев в состояние. После того, как каждый муравей в группе завершает поиск, интенсивность феромона на пути обновляется в соответствии с их соответствующими путями, чтобы завершить групповой поиск [20]. Процесс поиска колонии муравьев продолжает цикл и, наконец, максимизирует силу пути при оптимальной последовательности перемещения, чтобы получить оптимальное решение посредством обмена информацией и взаимного сотрудничества между отдельными муравьями.

Математическая модель алгоритма муравьиной колонии подробно описан следующим образом: Во-первых, а муравьи случайным образом размещены на b городов, каждый путь между городами имеет феромон инициализации, и каждый муравей имеет государственную последовательность записи таблицы. Он используется для записи городов, по которым ходили муравьи. Затем каждый муравей совершает переход между состояниями случайным образом в соответствии с вероятностью перехода состояния, и каждому переходу разрешен переход только из текущего состояния в соседнее состояние, так что муравей k переходит из состояния. Вероятность перехода состояния в соседнее состояние определяется как

$$\zeta_{ij}^k(t) = \begin{cases} \frac{[\delta_{ij}(t)]^\alpha \cdot [\epsilon_{ij}(t)]^\beta}{\sum_{\vartheta_i \in Q_u} [\delta_{ij}(t)]^\alpha \cdot [\epsilon_{ij}(t)]^\beta}, & \vartheta_i \in Q_u \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Со временем информация, оставленная на пути, постепенно исчезнет. Мы используем параметр, чтобы указать степень улетучивания информации. Через n раз муравей завершает цикл. В настоящее время объем информации по каждому пути равен

$$\delta_{ij}(t+n) = \rho \cdot \delta_{ij}(t) + \sum_{k=1}^m \Delta \delta_{ij}^k. \quad (5)$$

Среди них обозначает город, который муравью k разрешено посетить на следующем шаге. Поскольку система муравьиной колонии имеет функцию памяти, она может записывать городские узлы, по которым муравей k ходил раньше, и набор будет динамически корректироваться с развитием муравьев. представляет интенсивность феромона на пути от города i к городу j , δ – коэффициент расстояния от города i до города j , α – важность феромона на пути, β – важность эвристического фактора, а также α и β больше нуля.

В общем, алгоритмы оптимизации колоний муравьев (АСО) используют искусственных агентов конечного размера с определенными характеристиками, которые коллективно ищут качественные решения рассматриваемой проблемы. Начиная с начального состояния, выбранного в соответствии с некоторыми зависящими от случая критериями, каждый муравей строит решение, подобное хромосоме в генетическом алгоритме. Создавая собственное решение, каждый муравей собирает информацию о своих действиях и использует эту информацию, чтобы изменить представление о проблеме, которое видят другие муравьи [15–20]. Во внутренних состояниях муравья хранится информация о прошлом поведении муравья, которую можно использовать для вычисления качества / ценности сгенерированного решения.

Также муравьи используют политику вероятностных решений, чтобы направить свои поиски в наиболее интересные области поискового пространства. Уровень стохастичности политики и сила обновлений в феромонном следе определяют баланс между исследованием новых точек в пространстве состояний и использованием накопленных знаний

Экспериментальные исследования. В рамках экспериментальных исследований был произведен анализ работы алгоритмов стаи летучих мышей, бактериальный алгоритм и муравьиный алгоритм, используемые при решении NP-сложных задач на графах. В качестве задачи была выбрана классическая оптимизационная задача – Задача Коммивояжера. В качестве тестовых графов была сгенерирована серия графов от 30 до 120 вершин с шагом 10 с известным глобальным значением ЦФ. В качестве ЦФ рассматривается длина гамильтонова цикла для графа. Цель оптимизации является минимизация значения ЦФ. В ходе экспериментальных исследований были исследованы такие характеристики алгоритмов, как скорость сходимости алгоритма в зависимости от количества вершин в графе, а также оптимальность полученных решений в зависимости от количества вершин в графе. Под оптимальностью будем понимать отношение полученного значения ЦФ к глобальному значению ЦФ.

Был произведен запуск программного продукта с количеством итераций – 600.

На рис. 1 представлен график зависимости номера итерации, на которой алгоритм сошелся, т. е. после данной итерации значение ЦФ более не изменялось, от размера входных данных.

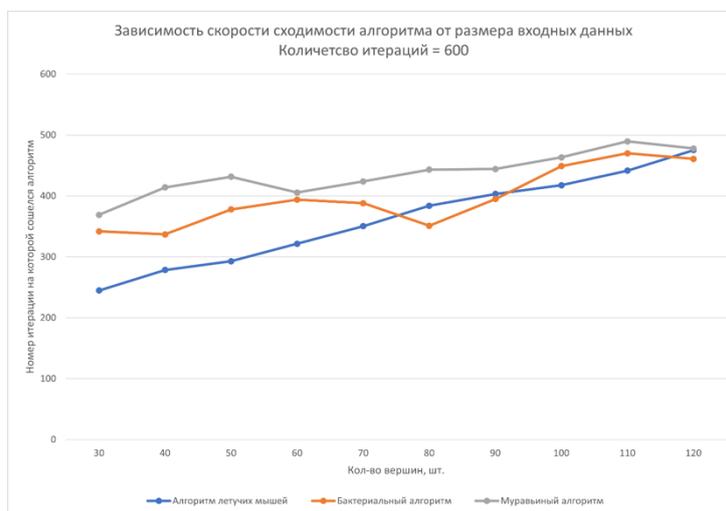


Рис. 1. Зависимость номера итерации от размера входных данных

Как видно из графика, различие в скорости сходимости алгоритмов становится не существенным для графов с количеством вершин более 90. Для графов с количеством вершин менее 90 в среднем наиболее быстрым в данном исследовании является алгоритм летучих мышей. Тем не менее качество получаемых решений, полученных с помощью алгоритма летучих мышей, резко ухудшается, с ростом количества вершин в графе. На рис. 2 представлена зависимость качества получаемых решений от количества вершин в графе. Из рисунка видно, что качество решений, полученных с помощью муравьиного алгоритма достаточно высокого и менее чувствительно к росту количества входных данных. Это объясняется хорошим балансом между диверсификацией поиска на начальных этапах и интенсификацией на заключительных этапах выполнения алгоритма.

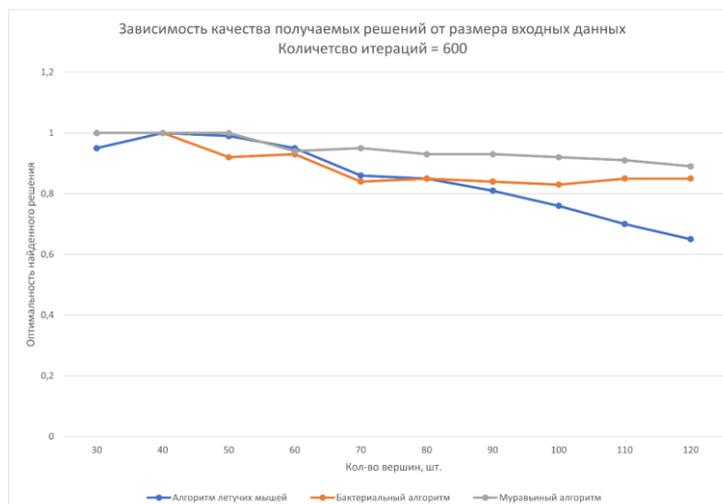


Рис. 2. Зависимость качества полученных решений от размера входных данных

Подводя итог, можно сказать, что биоинспирированные алгоритмы являются одними из самых мощных алгоритмов для оптимизации, которые будут иметь большое влияние на будущее поколение вычислений. Особое место они уже нашли при решении NP-полных и NP-сложных задач.

Заключение. В статье разработана биоэвристика для создания интеллектуальной подсистемы принятия эффективных решений NP-трудных и NP-сложных комбинаторно-логических задач на графах. В процессе выполнения был разработан модифицированный биоинспирированный алгоритм для решения NP- трудных и NP-сложных комбинаторно-логических задач на графах.

Приведено описание процедуры измерения сложности задачи. Описаны особенности NP- трудных и NP-сложных комбинаторно-логических задач. Приведены основные различия между задачами, а также проблемы, с которыми приходится сталкиваться при решении такого вида задач. Представлена общая схема принятия решений, состоящая из формулировки проблемы; принятие решения; сигнала в автоматических системах и обратной связи. На втором этапе (формирование и выбор вариантов решений) решение основывается на биоинспирированном алгоритме поиска решений задачи коммивояжера.

В статье разработан биоинспирированный подход к решению комбинаторно-логических задач на основе метода роевого интеллекта, включающий в себя поведения муравьиной колонии.

Эффективность биоинспирированных методов и алгоритмов определяется временем их работы, которое растет при росте объемов обрабатываемых данных, также растет при росте сложности решаемых для этих данных задач, но уменьшается при использовании параллельных или распределенных вычислительных систем.

Получены и исследованы теоретические оценки трудоемкости и эффективности разработанного биоинспирированного алгоритма для решения комбинаторно-логических задач на графах. Разработанный модифицированный биоинспирированный алгоритм точнее показывает значение целевой функции при большем количестве элементов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П.* Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2010. – 364 с.
2. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 474 с.
3. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Теория эволюционных вычислений. – М.: Физматлит, 2012. – 260 с.
4. *Штовба Д.С.* Муравьиные алгоритмы: теория и применение // Математика в приложениях. – 2004. – С. 70-75.
5. *Борознов В.О.* Исследование решения задачи коммивояжера // Вестник Астраханского государственного технического университета. Управление, вычислительная техника и информатика. – 2009. – С. 147-151.
6. *Курейчик В.М., Курейчик В.В.* Генетический алгоритм разбиения графа // Известия Российской академии наук. Теория и системы управления. – 1999. – № 4. – С. 79-87.
7. *Курейчик В.М., Лебедев Б.К., Лебедев О.К.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006. – 272 с. – ISBN 5-9221-0749-6.
8. *Курейчик В.В., Курейчик В.М., Гладков Л.А., Сороколетов П.В.* Биоинспирированные методы в оптимизации. – М.: Физматлит, 2009. – 384 с.
9. *Курейчик В.В., Заруба Д.В., Запорожец Д.Ю.* Биоинспирированный алгоритм компоновки блоков ЭВА на основе модифицированной раскраски графа // Известия ЮФУ. Технические науки. – 2015. – № 4 (165). – С. 6-14.
10. *Курейчик В.В., Курейчик Вл.Вл., Бова В.В.* Биоинспирированный поиск в задачах конструкторского проектирования и оптимизации // Информационные технологии в науке, образовании и управлении / под ред. проф. Е. Л. Глориозова. – 2015. – С. 427-432.
11. *Курейчик В.В.* Гибридный подход к решению комбинаторно-логических задач на графах // Тр. Международного научно-технического конгресса "Интеллектуальные системы и информационные технологии - 2019" ("ИС & ИТ-2019", "IS&IT19"). Научное издание: в 2-х т. – 2019. – С. 36-41.
12. *Bova V.V., Lezhebokov A.A., Gladkov L.A.* Problem-oriented algorithms of solutions search based on the methods of swarm intelligence // World Applied Sciences Journal. – 2013. – Vol. 27 (9). – P. 1201-1205.
13. *Kureichik V., Kureichik V., Bova V.* Placement of VLSI fragments based on a multilayered approach // Advances in Intelligent Systems and Computing. – 2016. – Vol. 464. – P. 181-190.
14. *Kureichik V.V., Zaruba D.V.* The bioinspired algorithm of electronic computing equipment schemes elements placement // Advances in Intelligent Systems and Computing. – 2015. – Vol. 347. – P. 51-58.
15. *Гладков Л.А., Гладкова Н.В., Гордиенко В.Н.* Модифицированный генетический алгоритм решения задачи компоновки блоков ЭВА // Информатика, вычислительная техника и инженерное образование. – 2015. – № 4 (24). – С. 18-27.
16. *Курейчик В.В., Курейчик Вл.Вл.* Биоинспирированный алгоритм разбиения схем при проектировании СБИС // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 23-29.
17. *Курейчик В.В., Заруба Д.В.* Двухуровневый алгоритм разбиения графа на части // Известия ЮФУ. Технические науки. – 2019. – № 2 (204). – С. 6-15.
18. *Курейчик В.М., Логунова Ю.А.* Об алгоритмах решения задачи коммивояжера в сети интернет // Вестник Рязанского государственного радиотехнического университета. – 2019. – № 68. – С. 37-43.
19. *Кулиев Э.В., Кравченко Ю.А., Логинов О.А., Запорожец Д.Ю.* Метод интеллектуального принятия эффективных решений на основе биоинспирированного подхода // Известия Кабардино-Балкарского научного центра РАН. – 2017. – № 6-2 (80). – С. 162-169.
20. *Zaporozhets D., Zaruba D., Kuliev E.* Ant algorithm for determining of critical CONNECTIONS IN VLSI // Proceedings of 2018 IEEE East-West Design and Test Symposium, EWDTTS 2018. Electronic publication. – 2018. – P. 8524709.

REFERENCES

1. *Norenkov I.P.* Osnovy avtomatizirovannogo proektirovaniya [Fundamentals of computer-aided design], Moscow: Izd-vo MGTU im. N.E. Baumana, 2010, 364 p.
2. *Karpenko A.P.* Sovremennyye algoritmy poiskovoi optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: a textbook]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014, 474 p.
3. *Kureichik V.V., Kureichik V.M., Rodzin S.I.* Teoriya evolyutsionnykh vychislenii [The theory of evolutionary calculations]. Moscow: Fizmatlit, 2012, 260 p.
4. *Shtovba D.S.* Murav'inye algoritmy: teoriya i primeneniye [Ant algorithms: theory and application], *Matematika v prilozheniyakh* [Mathematics in applications], 2004, pp. 70-75.
5. *Boroznov V.O.* Issledovanie resheniya zadachi kommivoyazhera [Research of the solution of the traveling salesman problem], *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of the Astrakhan State Technical University. Management, computer engineering and computer science], 2009, pp. 147-151.
6. *Kureichik V.M., Kureichik V.V.* Geneticheskii algoritm razbieniya grafa [Genetic algorithm of graph partitioning], *Izvestiya Rossiiskoi akademii nauk. Teoriya i sistemy upravleniya* [Izvestiya Rossiyskoy akademii nauk. Theory and control systems], 1999, No. 4, pp. 79-87.
7. *Kureichik V.M., Lebedev B.K., Lebedev O.K.* Poiskovaya adaptatsiya: teoriya i praktika [Search adaptation: theory and practice]. Moscow: Fizmatlit, 2006, 272 p. ISBN 5-9221-0749-6.
8. *Kureichik V.V., Kureichik V.M., Gladkov L.A., Sorokoletov P.V.* Bioinspirirovannyye metody v optimizatsii [Bioinspired methods in optimization]. Moscow: Fizmatlit, 2009, 384 p.
9. *Kureichik V.V., Zaruba D.V., Zaporozhets D.Yu.* Bioinspirirovannyy algoritm komponovki blokov EVA na osnove modifitsirovannoi raskraski grafa [Bioinspired algorithm for the layout of EVA blocks based on a modified graph coloring], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 4 (165), pp. 6-14.
10. *Kureichik V.V., Kureichik V.V., Bova V.V.* Bioinspirirovannyy poisk v zadachakh konstruktorskogo proektirovaniya i optimizatsii [Bioinspired search in the problems of design design and optimization], *Informatsionnye tekhnologii v nauke, obrazovanii i upravlenii* [Information technologies in science, education and management], ed. by prof. E. L. Glorizova, 2015, pp. 427-432.
11. *Kureichik V.V.* Gibridnyi podkhod k resheniyu kombinatorno-logicheskikh zadach na grafakh [A hybrid approach to solving combinatorial-logical problems on graphs], *Tr. Mezhdunarodnogo nauchno-tekhnicheskogo kongressa "Intellektual'nye sistemy i informatsionnye tekhnologii - 2019" ("IS & IT-2019", "IS&IT19")*. Nauchnoe izdanie: v 2-kh t. [Proceedings of the International Scientific and Technical Congress "Intelligent Systems and Information Technologies-2019" ("IS & IT-2019", "IS&IT19")]. Scientific publication: in 2 vol., 2019, pp. 36-41.
12. *Bova V.V., Lezhebokov A.A., Gladkov L.A.* Problem-oriented algorithms of solutions search based on the methods of swarm intelligence, *World Applied Sciences Journal*, 2013, Vol. 27 (9), pp. 1201-1205.
13. *Kureichik V., Kureichik V., Bova V.* Placement of VLSI fragments based on a multilayered approach, *Advances in Intelligent Systems and Computing*, 2016, Vol. 464, pp. 181-190.
14. *Kureichik V.V., Zaruba D.V.* The bioinspired algorithm of electronic computing equipment schemes elements placement, *Advances in Intelligent Systems and Computing*, 2015, Vol. 347, pp. 51-58.
15. *Gladkov L.A., Gladkova N.V., Gordienko V.N.* Modifitsirovannyy geneticheskii algoritm resheniya zadachi komponovki blokov EVA [Modified genetic algorithm for solving the problem of EVA block layout], *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Informatics, computer engineering and engineering education], 2015, No. 4 (24), pp. 18-27.
16. *Kureichik V.V., Kureichik V.V.* Bioinspirirovannyy algoritm razbieniya skhem pri proektirovanii SBIS [Bioinspired algorithm for splitting circuits in VLSI design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 23-29.
17. *Kureichik V.V., Zaruba D.V.* Dvukhurovnevyy algoritm razbieniya grafa na chasti [A two-level algorithm for splitting a graph into parts], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2019, No. 2 (204), pp. 6-15.
18. *Kureichik V.M., Logunova Yu.A.* Ob algoritmakh resheniya zadachi kommivoyazhera v seti internet [About algorithms for solving the traveling salesman problem on the Internet], *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta* [Bulletin of the Ryazan State Radio Engineering University], 2019, No. 68, pp. 37-43.

19. Kuliev E.V., Kravchenko Yu.A., Loginov O.A., Zaporozhets D.Yu. Metod intellektual'nogo prinyatiya effektivnykh reshenii na osnove bioinspirirovannogo podkhoda [The method of intellectual making effective decisions based on a bioinspired approach], *Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN* [Izvestiya Kabardino-Balkar Scientific Center of the Russian Academy of Sciences], 2017, No. 6-2 (80), pp. 162-169.
20. Zaporozhets D., Zaruba D., Kuliev E. Ant algorithm for determining of critical CONNECTIONS IN VLSI, *Proceedings of 2018 IEEE East-West Design and Test Symposium, EWDTS 2018. Electronic publication*, 2018, pp. 8524709.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

Заруба Дарья Викторовна – Южный федеральный университет; e-mail: dzaruba@sfedu.ru; г. Таганрог, Россия; тел.: 88634371651; ассистент.

Кулиев Эльмар Валерьевич – e-mail: ekuliev@sfedu.ru; доцент.

Запорожец Дмитрий Юрьевич – e-mail: zaporozhets@sfedu.ru; доцент.

Семенова Марина Максимовна – e-mail: bogomolova@sfedu.ru; студент.

Zaruba Darya Viktorovna – Southern Federal University; e-mail: dzaruba@sfedu.ru; Taganrog, Russia; phone: +78634371651; assistant.

Kuliev Elmar Valerievich – e-mail: ekuliev@sfedu.ru; associate professor.

Zaporozhets Dmitry Yurievich – e-mail: zaporozhets@sfedu.ru; associate professor.

Semenova Marina Maximovna – e-mail: bogomolova@sfedu.ru; student.

УДК 004.891.2

DOI 10.18522/2311-3103-2021-4-128-144

А.Н. Целых, В.С. Васильев, Л.А. Целых

АЛГОРИТМ ЭФФЕКТИВНЫХ УПРАВЛЕНИЙ В НЕСТОХАСТИЧЕСКИХ ПРИЧИННЫХ МОДЕЛЯХ В ОТСУТСТВИИ НАБЛЮДАЕМЫХ ПЕРЕМЕННЫХ ДЛЯ СИСТЕМ ПРИНЯТИЯ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ*

Рассматривается проблема репликации процесса принятия человеком управленческих решений в условиях неопределенности и неполноты исходных данных. Лицо, принимающее решение, опирается на свою систему взглядов, в которую входит общее видение системы, относительно которой принимается решение. Система представлена в виде причинной модели, созданной на основе ментальных представлений человека. Эти модели представляют собой направленные графы, на дугах которых причинность выражена в виде меток, которые имеют знак, определяющий направление изменений состояния системы. Вершины этого направленного графа представляют собой концепты высокого уровня абстракции. Такой граф моделирует функционирование реальной системы. Таким образом, мы исследуем проблему предсказания и управления действиями человека на основе нестохастических причинных моделей в отсутствие наблюдаемых переменных для использования в системах поддержки принятия решений и экспертных системах. Принятие решений рассматривается с точки зрения выбора объектов приложения управленческих воздействий – факторов модели. В настоящем исследовании мы показываем, что применение предложенного алгоритма может облегчить принятие решений относительно выбора управляющих воздействий, которые поддерживают достижение тактических и стратегических целей лица, принимающего решения. Следует отметить, что алгоритм реализует автоматизированный подбор параметра регуляризации, что делает доступным разработку и применение пред-

* Работа выполнена при поддержке гранта Российского Фонда фундаментальных исследований № 19-01-00109.