УДК 681.3

Э.И. Ватутин, канд. техн. наук, доцент, кафедра вычислительной техники, ЮЗГУ (e-mail: evatutin@rambler.ru)
В.С. Титов, докт. техн. наук, профессор, зав. кафедрой вычислительной техники, ЮЗГУ (e-mail: titov-kstu@rambler.ru)

ОСОБЕННОСТИ МЕТАОПТИМИЗАЦИИ АЛГОРИТМА ПЧЕЛИНОЙ КОЛОНИИ В ЗАДАЧЕ ПОИСКА КРАТЧАЙШЕГО ПУТИ В ГРАФЕ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЙ НА ПЛОТНОСТЬ ГРАФА

Приведено подробное описание стратегии использования метода пчелиной колонии в задаче поиска кратчайшего пути в графе в качестве тестовой задачи с возможностью сопоставления качества решений с оптимальными решениями, получаемыми с использованием алгоритма Дейкстры. Метод базируется на использовании агентов двух типов: пчел-разведчиков, выполняющих случайную разведку пространства параметров, и рабочих пчел, уточняющих данные пчел-разведчиков в пределах заданного радиуса разведки. Приведены результаты метаоптимизации, рекомендуемые значения настроечных параметров (вероятности использования различных модифицирующих операций, соотношение рабочих пчел и пчел-разведчиков, радиус разведки, число итераций, объем колонии) и сопоставление качества решений с известными эвристическими методами решения задач дискретной комбинаторной оптимизации (методы случайного и взвешенного случайного перебора, метод муравьиной колонии, метод имитации отжига, генетический (эволюционный) метод) по критериям средневыборочного качества решений и вероятности получения оптимального решения для выборок случайных графов объемом 1000 тестовых примеров. Показано, что метод пчелиной колонии требует различных оптимальных значений настроечных параметров в зависимости от размерности решаемой задачи (числа вершин графа N) и плотности графа d, что является недостатком и вынуждает выполнять метаоптимизацию непосредственно перед поиском решений, что, в свою очередь, повышает необходимые вычислительные затраты как минимум на 1–2 порядка. Показано, что даже после тонкой настройки параметров метод пчелиной колонии в рассматриваемой задаче зачастую уступает таким методам, как взвешенный случайный перебор, метод муравьиной колонии и генетический (эволюционный) метод.

Ключевые слова: поиск путей в графе, дискретная комбинаторная оптимизация, эвристические методы, метод пчелиной колонии, метаоптимизация.

***

Существует достаточно большое количество практически важных задач из области дискретной комбинаторной оптимизации и различных смежных разделов математики, таких как исследование операций, теория графов, теория расписаний и пр., которые относятся к классу *NP* и не допускают отыскание оптимальных за приемлемое время для случаев практически важной размерности [1]. Для их решения на практике с успехом применяется ряд методов, именуемых эвристическими, которые не гарантируют получение оптимальных решений, однако обеспечивают получение суб- или квазиоптимальных решений неплохого качества с приемлемыми затратами вычислительного времени. К ним относятся жадные подходы [2–9], вариации методов полного перебора с различными ограничениями (по стратегии ветвей и границ [10], с ограничением на глубину анализа дерева комбинаторного перебора [11] или на число комбинаторных возвратов [12]), методы случайного перебора [13–14] и их модификации [15], генетические и эволюционные методы [16–20], метод имитации отжига [21–22].

Относительно новым направлением в исследованиях, зародившимся в 1990-х – 2000-х годах в работах М. Дориго (M. Dorigo), Д. Карабога (D. Karaboga), Д. Фам (D. Pham) и др. является построение так называемых биоинспирированных методов, в основу которых положены подмеченные у природы принципы организации мультиагентных систем, к которым большинство ученых относят принципы, положенные в основу функционирования муравьиной [23–25] и пчелиной колонии [26–27]. Они представляют собой коллектив достаточно простых агентов (муравьев и пчел соответственно в природе или их электронных аналогов на компьютере), каждый из которых в отдельности является достаточно примитивным и действует по установленным сравнительно простым правилам, однако в совокупности в составе соответствующей колонии они способны решать нетривиальные задачи, в природе обычно связанные с нахождением кратчайших путей к подходящему источнику пищи.

При решении оптимизационных задач в рамках данных подходов количество или доступность пищи заменяется некоторой целевой функцией $f\left(X\right)$, для которой необходимо отыскание экстремума (обычно $f\left(X\right)\to\min$) в соответствии с заданными ограничениями в рамках решаемой задачи путем отыскания соответствующего вектора значений параметров $X^{*}=\arg\min\limits_{X\in\Re} f\left(X\right)$, где $\Re$ – множество допустимых решений, такого что $f^{*}=f\left(X^{*}\right)\to\min$. В данной работе рассмотрена возможность и особенности применения алгоритма пчелиной колонии в задаче поиска кратчайших путей в графе в соотнесении с качеством решений, получаемых другими эвристическими методами, перечень которых кратко рассмотрен выше. Данная задача достаточно просто формулируется в терминах теории графов и может быть решена за полиномиальное время $O\left(N^{2}\right)$ с использованием широко известного алгоритма Дейкстры [28], что позволяет достаточно простое отыскание как ее оптимального, так и субоптимальных решений, что, в свою очередь, позволяет оценить потенциал различных эвристических методов и сформулировать рекомендации об их преимущественной применимости на практике в указанных условиях использования.

Указанная задача поиска кратчайшего пути в графе формулируется следующем образом: в заданном графе $G=\left\langle A,V\right\rangle$, где $A=\left\{a_1,a_2,...,a_N\right\}$ – множество вершин, $V=\left\{v_1,v_2,...,v_M\right\}\subseteq A\times A$ – множество дуг, взвешенных значением длины $l\left(v_i\right)=l\left(a_{i_{нач}},a_{i_{кон}}\right)$, $N=\left|A\right|$ – число вершин, $M=\left|V\right|$ – число дуг, $d=\dfrac{M}{N\left(N-1\right)}$ – плотность графа, необходимо найти кратчайший путь

$P^* = \left[a_{i_1} = a_{нач}, a_{i_2}, a_{i_3,} ..., a_{i_Q} = a_{кон}\right]$ между заданной парой вершин $\{a_{нач}, a_{кон}\} \in A$, именуемых начальной и конечной, такой что $L^* = L\left(P^*\right) = \sum_{j=1}^{Q\left(P^*\right)-1} l\left(a_{i_j}, a_{i_{j+1}}\right) \to \min$.

Основной целью работы пчелиной колонии в природе является разведка окружающего улей пространства на предмет обнаружения мест расположения максимального количества нектара с последующим его сбором. Для этого в составе колонии существуют различные типы пчел: пчелы-разведчики и рабочие пчелы-фуражиры (в реальном улье существуют и другие типы пчел, такие как трутни или матка, однако в контексте поставленной задачи они не находят применения и поэтому не рассматриваются). Разведчики ведут исследование окружающего улей пространства и сообщают информацию о перспективных местах, в которых было обнаружено наибольшее количество нектара (для обмена информацией в биологическом улье существует специальный хитроумный механизм, именуемый танцем пчелы). Далее по наиболее перспективным из указанных разведчиками направлениям производится вылет рабочих пчел, которые занимаются сбором нектара, попутно проводя уточнение информации разведчиков о количестве нектара в некоторой окрестности от указанной разведчиком области. Между указанными типа пчел в улье поддерживается определенное соотношение (например, пять рабочих пчел на одного разведчика), оптимальное значение которого по-видимому было найдено в ходе эволюции, что позволяет организацию эффективной работы улья в целом (см. рис. 1).

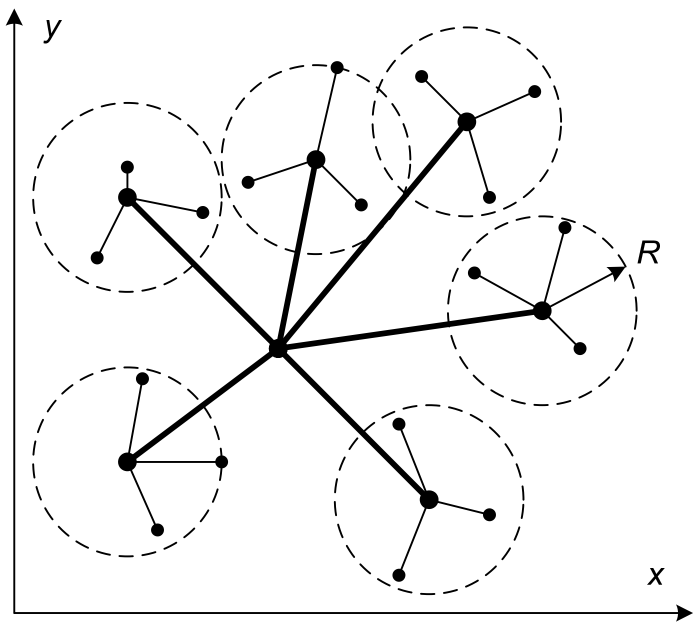

Рис. 1. Схематичное изображение поведения пчелиной колонии на двумерной плоскости. Колония представлена разведчиками (траектории движения от условного начального положения колонии показаны жирными

линиями) и рабочими пчелами-фуражирами (траектории движения показаны тонкими линиями от мест, указанных разведчиками) в соотношении 1:3, радиус действия пчел-фуражиров R показан штриховкой

Указанные социальные роли агентов могут быть использованы при решении оптимизационных задач, что и было предложено в работах [26–27]. При этом при решении задач дискретной комбинаторной оптимизации роль разведчика трансформируется в проведение разведки пространства допустимых решений $\Re$, с выбором одной из его точек $X_i \in \Re$, в которой целевая функция имеет некоторое значение $f(X_i)$. Данный выбор в простейшем случае может быть реализован с использованием метода случайного перебора, хотя возможны и более сложные стратегии [29]. С учетом невозможности хранения информации о всех путях (в указанной задаче $|\Re| \simeq O(N!)$) в электронном аналоге улья сохраняется информация о V лучших местах $S = \{X_1, X_2, ..., X_V\} \subseteq \Re$, которые характеризуются наилучшими значениями целевой функции $f(X_1) \leq f(X_2) \leq ... \leq f(X_V)$ (что в достаточной степени схоже с ограничением на объем популяции при использовании генетических подходов [16], однако в данном случае величина V скорее характеризует информационную емкость «памяти» улья, нежели чем непосредственное число пчел). Рабочая пчела берет в качестве направления движения одно из значений $X_i \in S$, $i = \overline{1,V}$, при этом в ходе своей работы ей необходимо осуществление разведки «вокруг» начальной точки X_i. Для этого необходимо каким-либо образом модифицировать начальное решение X_i, для чего можно использовать элементарные модифицирующие операции [22], применение каждой из которых приводит к однократному элементарному изменению решения $X_i \to X_i'$, а минимальное число подобных операций $d(X_i, X_j)$, необходимое для превращения решения X_i в X_j, можно рассматривать в качестве некоторой метрики, специфичной для данной задачи и являющейся аналогом известных расстояний Хэмминга [30] или Левенштейна [31], применяемых при сопоставлении строк, или их обобщение для задачи изоморфизма графов [32]. В простейшем случае, следуя работе [22], в качестве подобных операций могут быть применены операции добавления случайной вершины в случайную позицию текущего пути (сокр. ins), удаление случайной вершины из текущего пути (сокр. del), замена случайной вершины в текущем пути (сокр. ch) и перестановка пары случайно выбранных вершин пути (сокр. swp). Число R подобных модифицирующих операций, последовательно применяемых к решению X_i с целью получения решения X_j, не превосходит значения рассмотренной выше метрики $d(X_i, X_j)$ и может характеризоваться как радиус окрестности разведки. Результатом работы пчелы-фуражира является решение X_j и соответствующая ему оценка целевой функции $f\left(X_j\right)$. В случае, если хотя бы одно решение $X_k \in S$, $k=\overline{1,V}$ среди найденных ранее V решений уступает по качеству вновь найденному, «старое» решение заменяется на новое при их неизменном общем количестве в электронной модели улья: $S^{(t)} = S^{(t-1)} \setminus \left\{X_k\right\} \cup \left\{X_j\right\}$, где t – номер итерации. Комбинируя работу разведчиков и рабочих пчел в течение C итераций следует ожидать монотонного увеличения качества решений в составе отобранного подмножества. Выбор наилучшего среди них можно считать результатом работы алгоритма. При такой постановке работа алгоритма пчелиной колонии очень похожа на мультистарт-алгоритмы, используемые в задачах непрерывной оптимизации.

Более детальное и формализованное описание стратегии отыскания субоптимального решения в рассматриваемой задаче с использованием рассмотренного подхода может быть представлено в виде следующего алгоритма.

1. (инициализация) Положить номер текущей итерации $t=1$. Задать соотношение числа рабочих пчел N_W на одного разведчика. Задать вероятности применения элементарных модифицирующих операций $p_i, \sum_i p_i = 1, i=\overline{1,N_p}$. Задать радиус разведки R и число итераций C и объем колонии V.
2. (начальная разведка) Сформировать выборку $S=\left\{X_1, X_2, ..., X_V\right\}$ из V случайных решений, оценить их качество $f\left(X_i\right)$, положить $f^* = \min f\left(X_i\right)$, $X^* = \arg\min f\left(X_i\right)$, $i=\overline{1,V}$.
3. (вылет рабочих пчел) Положить $i=1$.
   3.1. Случайным образом выбрать решение $X_{curr} \in S$.
   3.2. (модификация решения) Положить $j=1$.
      3.2.1. Случайным образом выбрать модифицирующую операцию пропорционально вероятностям $p_k, k=\overline{1,N_p}$.
      3.2.2. Произвести модификацию решения X_{curr} с использованием выбранной модифицирующей операции p_k.
      3.2.3. Положить $j := j+1$.
      3.2.4. Если $j < R$, перейти к п. 3.2.1.
   3.3. Оценить качество решения $f\left(X_{curr}\right)$ после выполнения R модификаций.
   3.4. Если $f\left(X_{curr}\right) < \max\limits_{k=\overline{1,V}} f\left(X_k\right)$, исключить из S решение $X_{bad} = \arg\max\limits_{k=\overline{1,V}} f\left(X_k\right)$, поместив на его место решение X_{curr} после R модификаций: $S := S \setminus \left\{X_{bad}\right\} \cup \left\{X_{curr}\right\}$.

4. Положить $i := i+1$.
5. Если $i < N_W$, перейти к п. 3.1.
6. (вылет разведчика) Сформировать случайное решение X_{curr}, оценить его качество $f\left(X_{curr}\right)$.
7. Если $f\left(X_{curr}\right) < \max\limits_{k=\overline{1,V}} f\left(X_k\right)$, исключить из S решение $X_{bad} = \arg\max\limits_{k=\overline{1,V}} f\left(X_k\right)$, поместив на его место решение X_{curr}: $S := S \setminus \left\{X_{bad}\right\} \cup \left\{X_{curr}\right\}$.
8. Положить $t := t+1$.
9. Если $t\left(N_W+1\right) < C$, перейти к п. 3.
10. Вернуть в качестве результата $X_{best} = \arg\min\limits_{k=\overline{1,V}} f\left(X_k\right)$.
11. Конец алгоритма.

В качестве разведчиков, которые в природе производят вылеты в случайных направлениях равновероятно, был взят метод случайного перебора с возвратами [1]. Комбинаторные возвраты повышают его эффективности в области графов малой плотности, что должно положительно сказаться на качестве работы разведчиков и алгоритма пчелиной колонии в целом.

Эффективность применения приведенного алгоритма на практике зависит от ряда настроечных параметров $\left(p_1, p_2, ..., p_{N_p}, N_W, R, C, V\right)$, значения которых могут быть определены в ходе метаоптимизации. Для этого, следуя работам [11, 15, 16, 22, 24, 25], выполнялось формирование выборки $\Lambda = \left\{G_1, G_2, ..., G_K\right\}$ из K случайных графов с заданным числом вершин N и заданной плотностью d, для каждой выборки производилось нахождение решений и усреднение их качества с получением средневыборочной длины путей $\overline{L}$, которую необходимо минимизировать, а также расчетом вероятности получения оптимального решения p_{opt}.

Следуя указанной выше серии работ, число итераций C будем выбирать равным 1000 с целью сопоставления итерационных эвристических методов в равных условиях при одинаковом числе итераций (на практике это соответствует ограниченному объему времени, отведенному на поиск решения). Объем выборки тестовых примеров $\Lambda = \left\{G_1, G_2, ..., G_{N_\Lambda}\right\}$, используемых в вычислительных экспериментах для усреднения получаемых результатов, выбран $N_\Lambda = \left|\Lambda\right| = 1000$, что является компромиссом между временем выполнения вычислительных экспериментов (что обычно составляет несколько часов на один эксперимент при однопоточной программной реализации) и качеством получаемых зависимостей (дисперсией средневыборочных оценок и, соответственно, шириной доверительных интервалов).

Для выборки графов с $N = 10$ и $d = 0{,}9$ была проведена метаоптимизация, результаты которой представлены на рис. 2, в результате чего были выбраны следующие значения настроечных параметров алгоритма: $p_{ins}^* = 0{,}9$, $p_{del}^* = 0{,}4$, $p_{ch}^* = 0{,}5$, $p_{swp}^* = 0$, $N_W^* = 3$, $R^* = 1$, $V^* = 200$.

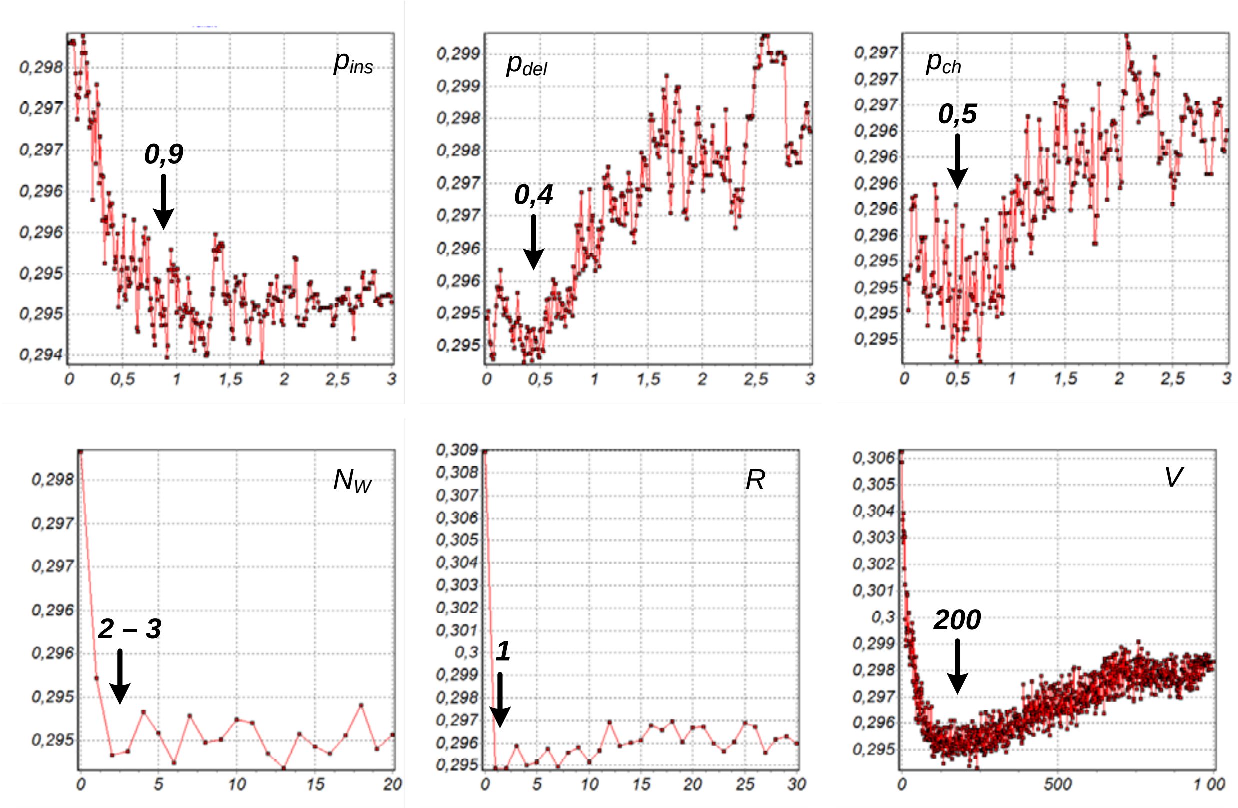

Рис. 2. Результаты зависимости средней длины кратчайших путей $\overline{L}$ от значений настроечных параметров алгоритма при $N = 10$ и $d = 0{,}9$

С использованием данных значений настроечных параметров и разработанного расчетного модуля [33] был проведен ряд вычислительных экспериментов для выборок графов с разным числом вершин и разной плотностью, результаты которых приведены в табл. 1.

Таблица 1. Результаты вычислительных экспериментов: средняя длина путей $\overline{L}$ и вероятность получения оптимального решения p_{opt}, полученные различными эвристическими методами, для выборок случайных графов различной размерности и плотности. Здесь O – оптимальное решение (алгоритм Дейкстры); AC – алгоритм муравьиной колонии; RM – случайный перебор; WRM – взвешенный случайный перебор; BC – алгоритм пчелиной колонии; SA – метод имитации отжига; GA – генетический алгоритм; ACR, RMR и WRMR – модификации перечисленных ранее методов с поддержкой комбинаторных возвратов [12]. Знаками «+» и «–» отмечены соответственно лучшие и худшие решения.

Метод	$N=10$ $d=0{,}5$	$N=10$ $d=0{,}9$	$N=100$ $d=0{,}5$	$N=100$ $d=0{,}9$
O	$\bar{L}=0{,}4804$ $p_{opt}=1{,}0$	$\bar{L}=0{,}2928$ $p_{opt}=1{,}0$	$\bar{L}=0{,}0967$ $p_{opt}=1{,}0$	$\bar{L}=0{,}0557$ $p_{opt}=1{,}0$
ACR	$\bar{L}=0{,}4804$ $p_{opt}=1{,}0$ +	$\bar{L}=0{,}2928$ $p_{opt}=0{,}999$ +	$\bar{L}=0{,}0991$ $p_{opt}=0{,}856$ +	$\bar{L}=0{,}0572$ $p_{opt}=0{,}848$ +
AC	$\bar{L}=0{,}4804$ $p_{opt}=1{,}0$ +	$\bar{L}=0{,}2928$ $p_{opt}=0{,}999$ +	$\bar{L}=0{,}0990$ $p_{opt}=0{,}856$ +	$\bar{L}=0{,}0573$ $p_{opt}=0{,}847$ +
WRMR	$\bar{L}=0{,}4808$ $p_{opt}=0{,}998$	$\bar{L}=0{,}2938$ $p_{opt}=0{,}992$	$\bar{L}=0{,}1132$ $p_{opt}=0{,}587$	$\bar{L}=0{,}0711$ $p_{opt}=0{,}519$
WRM	$\bar{L}=0{,}4757$ $p_{opt}=0{,}836$ –	$\bar{L}=0{,}2939$ $p_{opt}=0{,}987$	$\bar{L}=0{,}1130$ $p_{opt}=0{,}584$	$\bar{L}=0{,}0707$ $p_{opt}=0{,}519$
RMR	$\bar{L}=0{,}4814$ $p_{opt}=0{,}993$	$\bar{L}=0{,}2982$ $p_{opt}=0{,}931$	$\bar{L}=0{,}3208$ $p_{opt}=0{,}097$ –	$\bar{L}=0{,}2871$ $p_{opt}=0{,}073$ –
RM	$\bar{L}=0{,}4813$ $p_{opt}=0{,}855$ –	$\bar{L}=0{,}2985$ $p_{opt}=0{,}918$ –	$\bar{L}=0{,}3219$ $p_{opt}=0{,}097$ –	$\bar{L}=0{,}2872$ $p_{opt}=0{,}073$ –
SA	$\bar{L}=0{,}4894$ $p_{opt}=0{,}925$	$\bar{L}=0{,}2958$ $p_{opt}=0{,}969$	$\bar{L}=0{,}2461$ $p_{opt}=0{,}157$	$\bar{L}=0{,}1250$ $p_{opt}=0{,}178$
GA	$\bar{L}=0{,}4813$ $p_{opt}=0{,}990$	$\bar{L}=0{,}2954$ $p_{opt}=0{,}963$	$\bar{L}=0{,}2221$ $p_{opt}=0{,}161$	$\bar{L}=0{,}1366$ $p_{opt}=0{,}152$
BC	$\bar{L}=0{,}4810$ $p_{opt}=0{,}993$	$\bar{L}=0{,}2949$ $p_{opt}=0{,}966$	$\bar{L}=0{,}3449$ $p_{opt}=0{,}103$ –	$\bar{L}=0{,}2099$ $p_{opt}=0{,}096$ –

Как уже было отмечено ранее [24, 25], высокой эффективностью характеризуются решения, получаемые с использованием алгоритма муравьиной колонии. Сопоставимыми с ними по качеству являются решения, получаемые с использованием взвешенного случайного перебора (с поддержкой комбинаторных возвратов при рассмотрении графов малой плотности). Рассматриваемый в данной работе алгоритм пчелиной колонии по качеству решений сопоставим с аналогами при решении задач малой размерности, однако с ее ростом качество решений существенно падает, практически приближая метод к случайному перебору (средняя длина путей почти в 4 раза хуже оптимальной). С целью разобраться в сложившейся ситуации метаоптимизация была проведена повторно для другой точки пространства параметров $(N;d)$, ее результаты для $N=100$ и $d=0{,}5$ приведены на рис. 3.

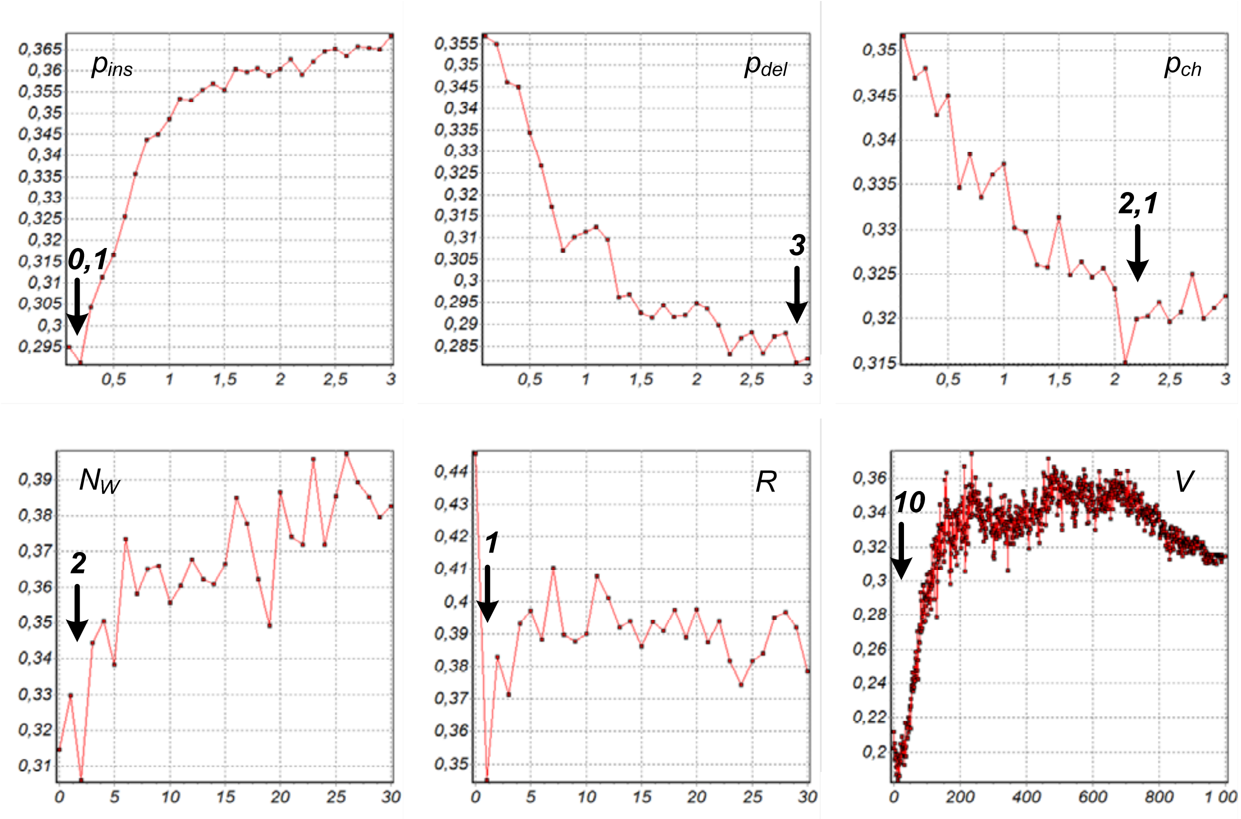

Рис. 3. Результаты зависимости средней длины кратчайших путей $\overline{L}$ от значений настроечных параметров алгоритма при $N = 100$ и $d = 0{,}5$

Для других пар значений $(N; d)$ также была проведена метаоптимизация, результаты которой в виде графиков ввиду ограниченного объема статьи не приводятся, однако основной вывод, который можно сделать по ее результатам, заключается в том, что в каждом конкретном случае выбранные значения параметров сильно отличаются друг от друга. Для наглядности оптимальные значения параметров для ряда выбранных сочетаний параметров сведены в табл. 2.

Таблица 2. Оптимальные значения настроечных параметров, полученные в ходе метаоптимизации, для различных пар значений $(N; d)$

Условия использования	p^*_{ins}	p^*_{del}	p^*_{ch}	N^*_W	R^*	V^*
$N = 10$ $d = 0{,}9$	0,7–3	0–0,5	0–1	3	1–2	200
$N = 10$ $d = 0{,}5$	0–3	0–3	0–3	2	1–2	600–1000
$N = 50$ $d = 0{,}9$	0–0,5	1–3	3	4–5	1	30
$N = 50$ $d = 0{,}5$	0,1–0,2	1–3	2–3	1–2	1–2	10–30
$N = 50$ $d = 0{,}1$	2,5–3	0,7–1,5	1,5–2,5	0	1–2	900–1000
$N = 100$ $d = 0{,}9$	0–0,1	3	2,5–3	2	1	10
$N = 100$ $d = 0{,}5$	0–0,1	3	2–3	2	1	10

$N = 100$ $d = 0{,}1$	1,5–2	1–3	0,5–1	0	1–2	900–1000

Выбор неоптимальных значений настроечных параметров может привести к ухудшению качества получаемых решений от нескольких десятков процентов, до 2–3 раз. В особенности это касается вероятностей добавления p_{ins} и удаления p_{del} вершин при модификации текущего пути и объема колонии *V*, которые имеют существенно различные оптимальные значения для различных условий применения. С целью наглядной демонстрации того, что качество решений может быть существенно улучшено путем тонкой настройки значений параметров алгоритма пчелиной колонии, в табл. 3 приведены результаты вычислительного эксперимента для $N = 100$ и $d = 0{,}5$ при различных значениях настроечных параметров.

Таблица 3. Пример зависимости качества получаемых решений от значений настроечных параметров алгоритма

Значения настроечных параметров	$\overline{L}$	p_{opt}
$p_{ins}^* = 0{,}9,\ p_{del}^* = 0{,}4,\ p_{ch}^* = 0{,}5,\ N_W^* = 3,\ R^* = 1,\ V^* = 200$	0,3449	0,103
$p_{ins}^* = 0{,}9,\ p_{del}^* = 0{,}4,\ p_{ch}^* = 2{,}5,\ N_W^* = 2,\ R^* = 1,\ V^* = 10$	0,1912	0,219

Данная особенность алгоритма пчелиной колонии является в достаточной степени неудобной и не позволяет выработать единые рекомендации по компромиссным значениям настроечных параметров, которые были бы неплохими для всех условий применения методов, в отличие от ряда других итерационных эвристических методов [15, 16, 22, 24, 25]. Слабая (в пределах нескольких процентов результирующего качества решений) зависимость от значений параметров наблюдается лишь для генетического алгоритма применительно к решению указанной задачи. Данная особенность вынуждает выполнение метаоптимизации непосредственно перед синтезом решений в указанных условиях использования, что является достаточно ресурсоемкой задачей (общие затраты машинного времени при этом возрастают как минимум на 1–2 порядка). Однако даже при наличии тонкой настройки результаты алгоритма пчелиной колонии существенно уступают результатам алгоритма муравьиной колонии для графов с большим числом вершин и малой плотностью.

Также результаты вычислительных экспериментов позволяют сделать ряд дополнительных выводов. Так оптимальное значение радиуса разведки *R* лежит в пределах 1–2 модификаций от начального решения (чаще всего $R^* = 1$), что является неочевидным эмпирическим следствием экспериментов. При его увеличении теоретически должно увеличиться разнообразие анализируемых путей, однако на практике это выливается в существенное (местами до 50%) ухудшение качества результирующих решений. При этом полное отсутствие модификаций $(R=0)$ в абсолютной большинстве случаев приводит к существенному ухудшению качества решений, что демонстрирует необходимость наличия в колонии не только разведчиков, но и рабочих пчел (при наличии только разведчиков алгоритм пчелиной колонии превращается в случайный перебор). Однако из этого правила есть исключение, которое проявляется в условиях использования алгоритма для графов малой плотности, в которых число корректных решений мало и модифицирующие операции зачастую переводят исходное корректное решение в некорректное из-за появления в нем запрещенных переходов между вершинами графа. При этом работа пчел-фуражиров является неэффективной (на нее впустую расходуются итерации работы алгоритма, число C которых ограничено по условиям эксперимента) и их рациональнее заменить разведчиками, положив $R=0$ или $N_W=0$, что, как уже было отмечено выше, фактически превращает алгоритм пчелиной колонии в случайный перебор.

Список литературы

1. Ватутин Э.И., Титов В.С., Емельянов С.Г. Основы дискретной комбинаторной оптимизации. М.: Инфра-М, 2016. 271 с.
2. Баранов С.И., Журавина Л.Н., Песчанский В.А. Обобщенный метод декомпозиции граф-схем алгоритмов // А и ВТ. 1982. № 5. С. 43–51.
3. Ватутин Э.И. Библиотека функций построения разбиений методом С.И. Баранова с жадным последовательным формированием блоков // Свидетельство о государственной регистрации программы для ЭВМ № 2010612902 от 28.04.10.
4. Ватутин Э.И., Леонов М.Е. Использование смежной окрестности при жадном последовательном формировании блоков разбиения граф-схем параллельных алгоритмов // Известия высших учебных заведений. Приборостроение. 2013. Т. 56. № 6. С. 30–35.
5. Ватутин Э.И., Титов В.С. Библиотека функций для построения разбиений с использованием смежной жадной стратегии и последовательным формированием блоков // Свидетельство о государственной регистрации программы для ЭВМ № 2013619395 от 03.10.13.
6. Ватутин Э.И., Романченко А.С., Титов В.С. Исследование влияния порядка рассмотрения пар на качество расписаний при использовании жадного подхода // Известия Юго-Западного государственного университета. 2013. № 1 (46). С. 58–64.
7. Ватутин Э.И., Бобынцев Д.О., Романченко А.С. Исследование влияния частичного упорядочивания пар и локального улучшения окрестности пары на качество расписаний при использовании жадного подхода // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2014. № 1. С. 8–16.

8. Ватутин Э.И., Бобынцев Д.О., Романченко А.С. Программа для жадного построения расписаний учебных занятий вуза // Свидетельство о государственной регистрации программы для ЭВМ № 2013618554 от 11.09.13.
9. Ватутин Э.И., Бобынцев Д.О., Романченко А.С. Метод жадного построения расписаний занятий вуза со случайным порядком рассмотрения учебных групп и улучшением окрестности текущей пары // Свидетельство о государственной регистрации программы для ЭВМ № 2013619101 от 25.09.13.
10. Land A.H., Doig A.G. An Automatic Method of Solving Discrete Programming Problems // Econometrica. Vol. 28. 1960. pp. 497–520. DOI: 10.2307/1910129.
11. Ватутин Э.И., Мартынов И.А., Титов В.С. Анализ результатов использования метода перебора с ограничением глубины в задаче поиска кратчайшего пути в графе // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов (МППОС’15). Барнаул, 2015. С. 120–128.
12. Ватутин Э.И., Мартынов И.А., Титов В.С. Способ обхода тупиков при решении задач дискретной оптимизации с ограничениями // Перспективные информационные технологии (ПИТ-2014). Самара: изд-во Самарского научного центра РАН. С. 313–317.
13. Ватутин Э.И., Колясников Д.В., Мартынов И.А., Титов В.С. Метод случайного перебора в задаче построения разбиений граф-схем параллельных алгоритмов // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов. Барнаул: Барнаул, 2014. С. 115–125.
14. Ватутин Э.И., Титов В.С. Библиотека функций построения разбиений граф-схем параллельных алгоритмов методом случайного перебора // Свидетельство об официальной государственной регистрации программы для ЭВМ № 2015618917 от 20.08.15.
15. Ватутин Э.И., Дремов Е.Н., Мартынов И.А., Титов В.С. Метод взвешенного случайного перебора для решения задач дискретной комбинаторной оптимизации // Известия ВолГТУ. Серия: Электроника, измерительная техника, радиотехника и связь. № 10 (137). Вып. 9. 2014. с. 59–64.
16. Ватутин Э.И., Титов В.С. Исследование особенностей применения генетического алгоритма в задаче поиска кратчайшего пути в графе при наличии ограничений на плотность графа // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов (МППОС’16). Барнаул, 2016. Принята к опубликованию
17. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Харьков: Основа, 1997 г. 112 с.
18. Галкина В.А. Дискретная математика: комбинаторная оптимизация на графах. М.: Гелиос АРВ, 2003. 232 с.

19. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М.: ФИЗМАТЛИТ, 2006. 320 с.
20. Панченко Т.В. Генетические алгоритмы / под ред. Ю.Ю. Тарасевича. Астрахань: Издательский дом "Астраханский университет", 2007. 87 с.
21. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by Simulated Annealing // Science. Vol. 220 No. 4598. 1983. pp. 671–680. DOI: 10.1126/science.220.4598.671.
22. Ватутин Э.И., Титов В.С. Параметрическая оптимизация алгоритма имитации отжига на примере решения задачи поиска кратчайшего пути в графе // Вестник Череповецкого государственного университета. № 6 (67). 2015. С. 13–16.
23. Dorigo M. Optimization, Learning and Natural Algorithms // PhD thesis. Politecnico di Milano, Italie, 1992.
24. Ватутин Э.И., Титов В.С. Анализ результатов применения алгоритма муравьиной колонии в задаче поиска пути в графе при наличии ограничений // Известия Южного федерального университета. Технические науки. 2014. № 12 (161). С. 111–120.
25. Ватутин Э.И., Титов В.С. Об одном подходе к использованию алгоритма муравьиной колонии при решении задач дискретной комбинаторной оптимизации // Интеллектуальные и информационные системы (Интеллект 2015). Тула, 2015. С. 8–13.
26. Karaboga D.D. An Idea Based On Honey Bee Swarm for Numerical Optimization // Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
27. Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M. The Bees Algorithm // Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
28. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. V. 1 (1959), PP. 269–271.
29. Кажаров А.А. Разработка и исследование роевых алгоритмов для решения транспортно-логистических задач // Дисс... к.т.н. по специальности 05.13.01. Таганрог, 2013. 165 с.
30. Hamming R.W. Error detecting and error correcting codes // Bell System Technical Journal. Vol. 29. 1950. pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.
31. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады академий наук СССР, 1965. Т. 163. Вып. 4. С. 845–848.
32. Ватутин Э.И. Эвристический подход к распознаванию изоморфизма графов // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание – 2015). Курск, 2015. С. 80–83.
33. Ватутин Э.И., Валяев С.Ю., Дремов Е.Н., Мартынов И.А., Титов В.С. Расчетный модуль для тестирования комбинаторных оптимизационных алгоритмов в задаче поиска кратчайшего пути в графе с использованием

добровольных распределенных вычислений // Свидетельство о государственной регистрации программы для ЭВМ № 2014619797 от 22.09.14.