Анализируя спектры (рис. 2), можно сделать вывод о применении лазеров с длинами волн излучения порядка 480-520нм, которым соответствует аргоновые лазеры.

Выводы. Предложена аппаратурная реализация метода лазерного деления ранних эмбрионов в животноводстве на основе использования лазера на аргоне, работающего в видимой области спектра. Обосновано использование лазера непрерывного действия, с длиной волны излучения 515 нм или 488 нм, с акустико-оптическим модулятором, формирующий луч лазера в виде импульсов длительностью 1 мс и меньше, с энергией импульсов ≈0,002 Дж, расходимостью 0,9 мрад.

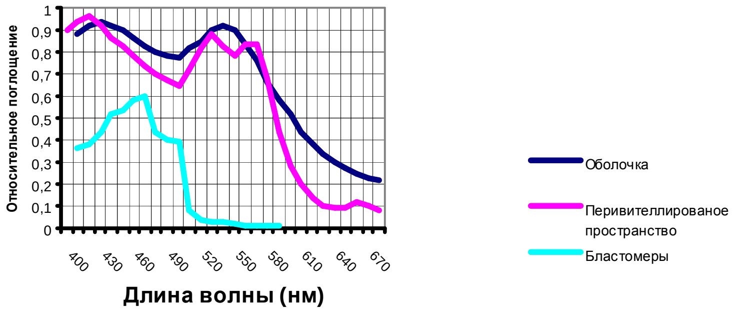


Рис. 2. Спектры поглощения оболочки, перивителлированого пространства, бластомеров, входящие в состав эмбриона

Список литературы: ***1.*** *А. с.* Устройство для регулирования температурного поля: А. с. № 323872 СССР / Е. П. Чубаров, А. М. Суворов, Г. П. Катыс (СССР). – Опубл. в бюл. "Открытия. Изобретения. Пром. образцы. Товарные знаки", 1972. № 1. – с. 212. ***2.*** *А. с.* Устройство для управления процессом электронно–лучевого нагрева: А. с. СССР / Е. П. Чубаров, А. Г. Бутковский, С. А. Важнов (СССР). – А. с. Опубл. в бюл. "Открытия. Изобретения. Пром. образцы. Товарные знаки", 1975. № 31. – с. 170. ***3.*** *Левкін А. В.* Біотехнологічні методи підвищення ефективності тваринництва в період реструктуризації // Проблеми енергозабезпечення та енергозбереження в АПК України. Вісник Харківського державного технічного університету сільського господарства. – Вип. 10, – Харків, 2002. – С. 240 – 242. ***4.*** *Левкин А. В* . Модель и метод решения задачи оптимизации лазерного деления эмбриона // Системи обробки інформації. – Вип. 5 (45) , – Харків: Харківський університет повітряних сил, 2005. – С. 85 – 92. ***5.*** *Левкин А. В* Система лазерного деления ранних эмбрионов в животноводстве // Материалы XX юбилейной Международной научно – практической конференции "Применение лазеров в медицине и биологии". – Ялта, 2003. – С. 136 – 137. ***6.*** *Левкин А. В.* Физиологическое действие лазера // Материалы XXII Международной научно – практической конференции "Применение лазеров в медицине и биологии". – Ялта, 2004. – С. 156 – 157. ***7.*** *Левкин А. В.* Подсистема управления мощностью лазера в процессе деления ранних эмбрионов крупного рогатого скота // Материалы XXV Международной научно – практической конференции "Применение лазеров в медицине и биологии". – Луцк, 2006. – С. 135 – 136. ***8.*** *Левкин А. В. , Мегель Ю. Е.* Моделирование процессов взаимодействия лазерного излучения с микробиологическими объектами // Материалы III Международной научно – технической конференции "Методы представления и обработки случайных сигналов и полей". – Харьков: ХГТУРЭ. 1993. – С. 21 – 22.

Поступила в редколлегию 12.05.08

УДК 519.174.7

В.Ф. ПРОКОПЕНКОВ, ***Ю.Н. КОЖИН***, ***О.Н. МАЛЫХ***, канд. техн. наук

НОВЫЙ ЭВРИСТИЧЕСКИЙ АЛГОРИТМ РАСКРАСКИ ГРАФА

У дискретній математиці відомі різні алгоритми розкрашення графів: точні, перебірні, евристичні. Недоліком точних та перебірних алгоритмів є те, що вони характеризуються неполіноміальною складністю від кількості вершин у графі. Евристичні потребують менших затрат часу, але не гарантують отримання оптимального рішення. В запропонованій статі надається новий евристичний алгоритм розкрашення графу, який має лінійну складність. Якість розкрашення досягається порядком обробки вершин, що вибирається.

В дискретной математике известны разные алгоритмы раскраски графов: точные, переборные, эвристические. Недостатком точных и переборных алгоритмов является то, что они характеризуются неполиномиальной сложностью от количества вершин в графе. Эвристические требуют меньших затрат времени, но не гарантируют получения оптимального решения. В предлагаемой статье описывается новый эвристический алгоритм раскраски графа, обладающий линейной сложностью. Качество окраски достигается выбираемым порядком обработки вершин.

The exact, searching and heuristic algorithms are known in discrete mathematics. Non polynomial computational complexity in relation to graph nodes quantity is the imperfection of exact and searching algorithms. The heuristic algorithms are need smallest time, but optimal solution isn’t guaranteed. This article presents new graph paint heuristic algorithm with linear computational complexity. The paint quality is achieved by selected nodes processing order.

Введение. Многие вопросы логического проектирования сводятся к задаче о нахождении минимальной раскраски вершин графа или ее модификациям. Обычно они формулируются как задачи о разбиении данного множества на минимальное число совместимых в том или ином смысле подмножеств.

Задачи, где условие попарной совместимости элементов подмножества необходимо и достаточно для их совместимости в совокупности, сводятся к нахождению минимальной раскраски графа, вершины которого соответствуют элементам данного множества, а ребра — парам несовместимых элементов этого множества.

Для нахождения решения задачи раскраски графа большой размерности можно применять точные или приближенные методы ее решения. Точные методы гарантируют получение оптимального решения. Приближенные методы не всегда обеспечивают наилучший результат. И те и другие характеризуются экспоненциальной временной сложностью [1,2].

Как решение проблемы для задач большой размерности предлагается идти по пути разработки новых однократных эвристические алгоритмов, что и делается в этой статье.

Постановка задачи. Пусть задан граф $G=(V,E)$, в котором $V=\{v_n \mid n=\overline{1,N}\}$ - множество вершин графа, $E=\left\{e_{ij} \middle| i,j\in\overline{1,I}, i\neq j\right\}$ - множество дуг графа. Пусть k - натуральное число, тогда функция $f:V\to\{1,..,k\}$ называется раскраской графа.

При этом раскраска считается правильной, если для любой пары вершин $v_i, v_j\in V$ таких, что $e_{ij}\in E$ справедливо, что $f(v_i)\neq f(v_j)$.

Необходимо определить такую функцию f, чтобы $k\to\min=c(G)$, где $c(G)$ - хроматическое число графа.

Метод решения. Предлагаемый метод состоит не в явном окрашивании вершин, а формировании набора множеств независимых вершин исходного графа.

Две вершины $v_i, v_j\in V$ называются независимыми, если не существует дуги $e_{ij}\in E$.

Множества независимых вершин графа формируются так, что для очередной обрабатываемой вершины реализуется попытка разместить эту вершину в одном из уже сформированных множеств. И, если это не представляется возможным, то формируется новое множество.

Количество k сформированных в результате непустых множеств составит результирующее число красок раскраски, а вершины, вошедшие в одно множество, будут окрашены в один из k цветов.

Из рассмотренного метода понятно, что количество таких множеств зависит от порядка обработки вершин.

Структура данных. Каждую вершину исходного графа $v_i\in V$ будем представлять парой $v_i=<i,links_i>$, где: i - это номер вершины, а $links_i=\{j \mid j\in\overline{1,N}, e_{ij}\in E\}$ - множество номеров вершин $v_j\in V$ графа G, с которыми v_i в графе связана дугой. Тогда граф G представим как $G=\{v_n \mid n=\overline{1,N}\}=\{<n,links_n> \mid n=\overline{1,N}\}$.

Если $color\_counter$ - это количество используемых красок для окраски графа G, тогда решение задачи можно представить как

$S=\{set_c \mid c=\overline{0,color\_counter-1}\}$, где set_c - это множество номеров вершин графа G, окрашиваемых в цвет c.

Алгоритм раскраски.

П.1. Положить $color\_counter=0$ - количество сформированных множеств независимых вершин графа G.

П.2. Выполнить цикл по переменной $i=\overline{1,N}$ с шагом 1, в котором выполнить обработку каждой вершины графа $G=\{<i,links_i> \mid i=\overline{1,N}\}$:

2.1. Определить номер c множества set_c независимых вершин графа G, в которое необходимо добавить текущую обрабатываемую вершину v_i, т.е. выполнить цикл по переменной $c=\overline{0,color\_counter-1}$ с шагом 1:

2.1.1. Положить $connected=0$ флаг наличия связей в графе между вершиной v_i и вершинами-элементами множества set_c.

2.1.2. Перебрать все вершины $v_j\in set_c$. Если между v_i и $v_j\in set_c$ существует дуга e_{ij} в графе G, то выполнить: $connected=connected+1$.

2.1.3. Если $connected=0$, перейти к п.2.2.

2.2. Если $c=color\_counter$, т.е. списка с номером c еще не существует (v_i не может быть добавлена ни в одно из существующих множеств), то создать множество set_c и увеличить переменную $color\_counter=color\_counter+1$.

2.3. Добавить вершину-элемент v_i в список set_c.

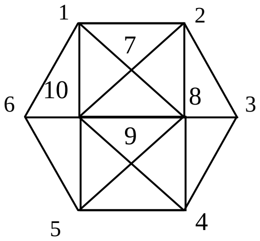
П.3. Конец.

Тестирование алгоритма. Тестирование алгоритма проведено на примере графа G={<1,{}>, <2,{1}>, <3,{2}>, <4,{3}>, <5,{4}>, <6,{1, 5}>, <7,{1, 2}>, <8,{2,3,4,7}>, <9,{4,5,8}>, <10,{1,5,6,7,8,9}>} (см. рисунок).

Алгоритм применялся для четырех исходных порядков вершин указанного графа:

- в порядке исходной нумерации вершин;

- в порядке, обратном нумерации вершин;
- в порядке возрастания количества связей вершин;
- в порядке убывания количества связей вершин.



Результаты испытания алгоритма представлены в таблице.

№	порядок обработки	Решение	K
1	совпадающий с нумерацией вершин	S={{1,3,5,},{2,4,6},{7,9},{8},{10}}	5
2	обратный нумерации вершин	S={{10,4,2,},{9,7,6,3},{8,5,1}}	3
3	по возрастанию числа связей вершины	S={{3,6,7,9},{1,4},{2,5},{8},{10}}	5
4	по убыванию числа связей вершины	S={{8,1,5,},{10,2,4},{7,9,3,6}}	3

Отметим, что первые два правила по существу характеризуют случайный порядок обработки. Из двух последних правил, лучший результат получен при обработке вершин в порядке убывания количества их связей с другими вершинами в графе. Этот результат является оптимальным.

Заключение. При удачно выбранном порядке обработки вершин графа рассмотренный алгоритм позволяет найти оптимальное решение. Для данного примера оптимум достигается при обработке вершин в порядке убывания их инцидентностей.

Также, необходимо отметить, что при обработке очередной вершины графа возможна вариантность ее добавления в одно из сформированных множеств независимых вершин.

Эта неоднозначность в приведенном алгоритме разрешается добавлением вершины в первое из сформированных множеств независимых вершин графа, в которое эта вершина может быть добавлена. В общем случае эта вариантность является источником разных вариантов решения задачи, которая может быть исследована.

Список литературы: ***1.*** *Кристофидес Н.* Теория графов. Алгоритмический подход.– М.: Мир, 1978.- 432 с. ***2.*** *Новиков Ф.А.* Дискретная математика для программистов.– С.-Пб.: Питер, 2004.- 364 с.

Поступила в редколлегию 09.01.08

УДК 519.174.7

О. Н. МАЛЫХ, канд. техн. наук,
Ю. Д. ОГИЕНКО, студент НТУ «ХПИ»

МЕТРИЗАЦИЯ ПРОСТРАНСТВА РЕШЕНИЙ ЗАДАЧИ РАСКРАСКИ ГРАФА

У статті пропонується здійснити пошук хроматичного числа графа методом випадкового пошуку з локальною оптимізацією. Для цього пропонується провести метрізацію простіра рішень задачі розфарбовування графа. Так само розглядаються шляхи рішення даної задачі і проблеми що виникають в процесі рішення.

В статье предлагается осуществить поиск хроматического числа графа методом случайного поиска с локальной оптимизацией. Для этого предлагается произвести метризацию пространства решений задачи раскраски графа. Так же рассматриваются пути решения данной задачи и проблемы, возникающие в процессе решения.

The article it is suggested to carry out the search of chromatic number of count the method of random search with peep-hole optimization. For this purpose it is suggested to apply metrics on spaces of decisions of task of coloration of count. The ways of decision of this task and problem are similarly examined arising up in the process of decision.

Введение. Экстремальная постановка задачи раскраски графов звучит следующим образом. Дан граф, вершины которого необходимо раскрасить различными красками так, чтобы ни какие две смежные вершине не были окрашены в один и тот же цвет. Иными словами можно сказать, что все вершины графа необходимо разбить на минимальное количество подмножеств. Решение считается допустимым, если никакие две смежные вершины не принадлежат одному подмножеству.

Критерием качества решения является количество подмножеств, на которые разделены вершины графа, т.е. количество используемых красок. Минимально возможное количество красок называется хроматическим числом графа.

Следует отметить, что решение не зависит от способа нумерации красок. В большинстве методов раскраски графа качество решения зависит от порядка просмотра вершин.