

Лебедев Олег Борисович – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

Лебедева Екатерина Олеговна – e-mail: lbedevakate@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; аспирант.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +79282897933; the department of computer aided design; professor.

Lebedev Oleg Borisovich – e-mail: lebedev.ob@mail.ru; phone: +79085135512; the department of computer aided design; associate professor.

Lebedeva Ekaterina Olegovna – e-mail: lbedevakate@mail.ru; phone: +79289591426; the department of computer aided design; graduate student.

УДК 004.021

DOI 10.18522/2311-3103-2020-2-157-169

А.А. Могилев, В.М. Курейчик

**МОДИФИЦИРОВАННЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ
ПЛАНИРОВАНИЯ ПРОЕКТОВ, РЕАЛИЗОВАННЫЙ
С ИСПОЛЬЗОВАНИЕМ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ***

Предложена структура модифицированного генетического алгоритма для решения задачи построения расписания проекта с учетом ограниченности ресурсов, реализованного с использованием облачных вычислений, проведен вычислительный эксперимент, в ходе которого было произведено сравнение результатов работы предложенного алгоритма с лучшими из известных, на данный момент, результатами. Исходя из результатов эксперимента был сделан вывод, о том, что предложенный алгоритм может быть использован для планирования работ реальных проектов, так как с его помощью возможно составлять расписания для проектов с количеством работ $n = 90$ за приемлемый промежуток времени. При планировании проектов с количеством работ $n = 30$, $n = 60$, $n = 90$, 120 время выполнения предложенного алгоритма было меньше, чем время выполнения стандартного генетического алгоритма в 2.8, в 4, в 5.5 и 6.8 раз соответственно. В связи с тем, что задача построения расписания проекта с учетом ограниченности ресурсов является NP-трудной, проблема создания новых и модификации существующих методов её решения по-прежнему остается актуальной. Для планирования проектов с большим количеством работ целесообразно использовать облачные вычисления, так как планирование таких проектов может потребовать много времени и вычислительных ресурсов. Использование облачных вычислений позволит сократить время выполнения генетического алгоритма за счет предоставления поставщиком облачного сервиса больших вычислительных ресурсов. В связи с этим, предложенный в данной работе алгоритм отличается от уже имеющихся использованием облачных вычислений для распределения нагрузки между рабочими станциями, на которых одновременно выполняется данный алгоритм. Применение в генетическом алгоритме модифицированных операторов, а также использование облачной инфраструктуры как услуги для реализации генетического алгоритма при решении задачи планирования проектов определяет научную новизну исследования.

Теория расписаний; задача построения расписания для проекта с учетом ограничений на ресурсы; генетический алгоритм.

* Работа выполнена при финансировании гранта РФФИ №18-07-00050.

A.A. Mogilev, V.M. Kureichik

MODIFIED GENETIC PROJECT PLANNING ALGORITHM IMPLEMENTED WITH THE USE OF CLOUD COMPUTING

The paper proposes a structure of a modified genetic algorithm for solving resource constrained project scheduling problem implemented with the use of cloud computing, a computational experiment was conducted, during which the results of the proposed algorithm were compared with the best known, at the moment, results. Based on the results of the experiment, it was concluded that the proposed algorithm can be used to plan the work of real projects, since it is possible to draw up schedules for projects with the number of works $n = 90$ for an acceptable period of time. When planning projects with the number of jobs $n = 30, n = 60, n = 90, 120$, the execution time of the proposed algorithm was less than the execution time of the standard genetic algorithm by 2.8, 4, 5.5 and 6.8 times, respectively. Due to the fact that the task of constructing a project schedule taking into account limited resources is NP-difficult, the problem of creating new and modifying existing methods for solving it remains relevant. For planning projects with a large number of works, it is advisable to use cloud computing, since planning such projects can require a lot of time and computing resources. In this regard, the algorithm proposed in this paper differs from the existing ones by using cloud computing to distribute the load between workstations on which this algorithm is simultaneously running. The use of modified operators in the genetic algorithm, as well as the use of cloud infrastructure as a service for implementing a distributed genetic algorithm, determines the scientific novelty of the study.

Scheduling theory; resource constrained project scheduling problem; genetic algorithm.

Введение. Задача оптимального планирования проекта при наличии ресурсных ограничений известна как задача планирования расписания для проекта с учетом ограниченности ресурсов – *RCPS*P (resource constrained project scheduling problem). Эта задача остается актуальной среди различных исследователей на протяжении трех десятков лет, так как ее решения могут иметь практическое применение во многих отраслях. Различают три основных категории методов решения задачи *RCPS*P [1]: точные методы [2, 3], эвристические методы [4–6] и метаэвристические [7–14]. Предложенный в данной работе генетический алгоритм можно отнести к метаэвристическим методам.

Задача *RCPS*P является NP-трудной даже без дополнительных ограничений. Лучший из известных точных алгоритмов Брукера за приемлемое время может решать примеры размерности не больше шестидесяти работ [15]. В связи с этим, для достижения близких к оптимальным результатов решения задачи за приемлемое время в данной работе предлагается структура генетического алгоритма решения задачи *RCPS*P, основанная на использовании облачных вычислений. Использование параллельной структуры позволит достигать близких к оптимальным результатов за счет распределения альтернативных решений между тремя независимыми рабочими станциями.

Использование облачных вычислений позволит сократить время выполнения генетического алгоритма за счет предоставления поставщиком облачного сервиса больших вычислительных ресурсов.

В рамках концепции облачных систем принято выделять три модели обслуживания: облачное программное обеспечение как услуга (*Cloud Software as a Service, SaaS*), облачная платформа как услуга (*Cloud Platform as a Service, PaaS*) и облачная инфраструктура как услуга (*Cloud Infrastructure as a Service, IaaS*) [16].

Для апробации предложенного в данной работе алгоритма была выбрана модель «облачная инфраструктура как услуга». При использовании данной модели обслуживания потребителям предоставляются по подписке фундаментальные информационно-технологические ресурсы – виртуальные серверы с заданной вычислительной мощностью, операционной системой (чаще всего – предустановленной провайдером из шаблона) и доступом к сети.

Принципиальным отличием предложенного генетического алгоритма от уже имеющихся решений является использование распределенной структуры для выполнения в «облаке», а также использование модифицированных генетических операторов таких как параметризованный кроссинговер, мутация на основе дихотомии и транслокация. Проведенный вычислительный эксперимент показал, что использование модифицированных генетических операторов и облачных вычислений позволяет достигать близких к оптимальным результатов при планировании проектов с количеством работ $n = 90$ за приемлемый промежуток времени.

Постановка задачи построения расписания для проекта с учетом ограниченности ресурсов. В задаче построения расписания для проекта необходимо построить оптимальное расписание проекта (выполнения работ проекта) с учетом сетевого графика (отношений предшествования между работами) и с учетом необходимых/доступных ресурсов, при котором будет оптимизирована некоторая целевая функция. Самая популярная целевая функция – общее время выполнения проекта (*makespan* или C_{\max}) [15].

Постановка задачи построения расписания для проекта имеет следующий вид [15]. Дано множество работ (требований) $N = \{1, \dots, n\}$ и R возобновляемых ресурсов $r = 1, \dots, R$. В каждый момент времени t доступно Q_r единиц ресурса r . Заданы продолжительности обслуживания $d_i \geq 0$ для каждого требования $i = 1, \dots, n$. Во время обслуживания требования i требуется $q_{ir} \leq Q_r$ единиц ресурса $r = 1, \dots, R$. После завершения обслуживания требования, освобожденные ресурсы в полном объеме могут быть мгновенно назначены на обслуживание других требований.

Между некоторыми парами требований заданы ограничения предшествования: $i \rightarrow j$ означает, что обслуживание требования j начинается не раньше окончания обслуживания требования i . Обслуживание требований начинается в момент времени $t = 0$. Прерывание при обслуживании требований запрещено. Необходимо определить моменты времени начала обслуживания требований S_i , $i = 1, \dots, n$, так, чтобы минимизировать время выполнения всего проекта, т.е. минимизировать значение

$$C_{\max} = \max\{C_i\}, i = 1, \dots, n, \quad (1)$$

где $C_i = S_i + d_i$. При этом должны быть соблюдены следующие ограничения:

1) в каждый момент времени $t \in [0, C_{\max}]$ должно выполняться

$$\sum_{i=1}^n q_{ir} \phi_i(t) \leq Q_r, r = 1, \dots, R, \quad (2)$$

где $\phi_i(t) = 1$, если требование i обслуживается в момент времени t и $\phi_i(t) = 0$, в противном случае. То есть требования в процессе своего обслуживания должны быть полностью обеспечены ресурсами;

2) не нарушаются отношения предшествования между требованиями, т.е.

$$S_i + d_i \leq S_j, \text{ если } i \rightarrow j \text{ для } i, j \in N.$$

На рис. 1 представлен пример сетевого графа проекта. Вершинами графа являются работы проекта, над вершинами изображены продолжительность выполнения соответствующих работ d_i и необходимое количество ресурсов для выполнения работы r_i . Количество возобновляемых ресурсов проекта $R = 4$. На рис. 2 представлена диаграмма возможного расписания данного проекта, по вертикали указано количество ресурсов, по горизонтали – время выполнения работ.

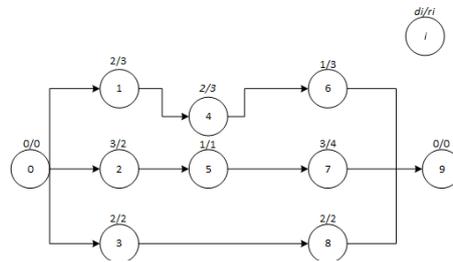


Рис. 1. Сетевой граф проекта

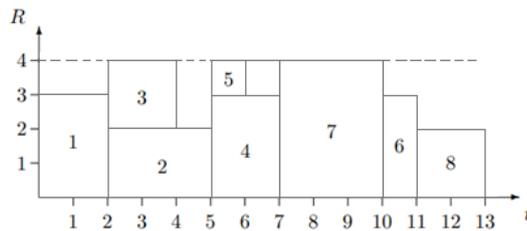


Рис. 2. Пример расписания для проекта

Модифицированный генетический алгоритм. Генетический алгоритм выполняется на четырех вычислительных машинах (Мастер, ПК1, ПК2 и ПК3), расположенных в «облаке» и состоит из следующих этапов:

- ◆ «Мастер» генерирует начальную популяцию хромосом, разделяет ее на три части и передает каждую из частей ПК1, ПК2 и ПК3;
- ◆ На ПК1, ПК2 и ПК3 выполняется модифицированный генетический алгоритм, представленный на рис. 4;
- ◆ Мастер выбирает наилучшее из трех решений в соответствии с целевой функцией.

Схема выполнения генетического алгоритма представлена на рис. 3.

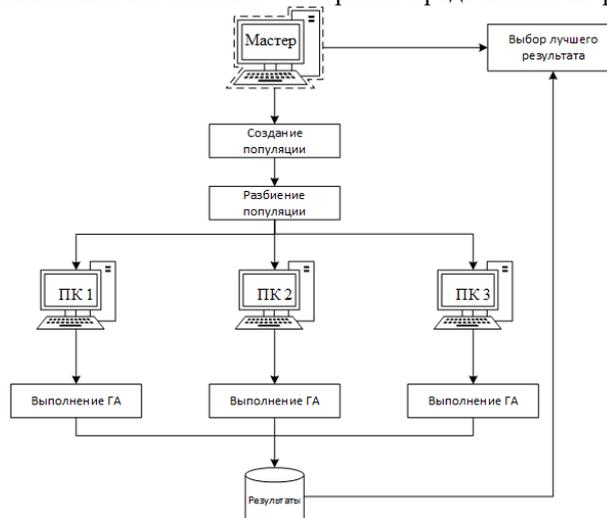


Рис. 3. Схема выполнения генетического алгоритма

Структурная схема генетического алгоритма для решения *RCPSP*-задачи, выполняемого на рабочих станциях ПК1, ПК2 и ПК3 представлена на рис. 4. Временная сложность данного алгоритма зависит от размера начальной популяции N и длины набора альтернативных решений L и определяется формулой:

$$T = O(NL). \quad (3)$$



Рис. 4. Структурная схема генетического алгоритма

Генерация начальной популяции. Процесс выполнения любого генетического алгоритма начинается с генерации набора альтернативных решений (хромосом) [17], который называется популяцией. Альтернативное решение характеризуется набором параметров, которые называют генами. В альтернативном решении ген соответствует элементу, концептуальному узлу, связи и т.п. [18]

При генерации начальной популяции необходимо учитывать два типа ограничений: ограничения предшествования (работа не может быть начата, прежде чем закончатся все её предшественники) и ресурсные ограничения (число ресурсов, выделенных на работы проекта в единицу времени, не может превышать максимальное доступное число ресурсов).

Популяция в предложенном генетическом алгоритме представляет собой набор альтернативных решений (рис. 5), каждый ген которых содержит: t_j – время начала работы, d_j – продолжительность работы, r_j – необходимое число ресурсов, где j – номер работы проекта. Таким образом решение C представляет собой потенциально возможное решение и состоит из последовательности работ:

$$C = \{j_1, j_2, j_3, \dots, j_n\}. \quad (4)$$

j_1	j_2	j_3	...	j_n
$t_{j_1}, d_{j_1}, r_{j_1}$	$t_{j_2}, d_{j_2}, r_{j_2}$	$t_{j_3}, d_{j_3}, r_{j_3}$...	$t_{j_4}, d_{j_4}, r_{j_4}$

Рис. 5. Представление альтернативных решений

Селекция. После генерации начальной популяции необходимо выбрать пары хромосом для скрещивания. Выбор пар хромосом производится следующим образом. В качестве родителей берутся сперва наборы хромосом с наибольшим значением функции приспособленности, далее два следующих по значению и так далее пока в исходной популяции есть хромосомы, не прошедшие процесс скрещивания.

Качество популяции в генетических алгоритмах принято оценивать с помощью функции приспособленности. Так как задача построения расписания для проекта предполагает нахождение решения с минимально возможным временем выполнения проекта, функция приспособленности выбрана обратно пропорциональной сумме времени выполнения всех работ проекта [19]:

$$F(x) = \frac{1}{\max\{t_i + d_i | i=1,2,\dots,n\}} \quad (5)$$

Вероятность выживания особи зависит от значения данной функции.

Кроссинговер. Для построения потомков хромосом был использован оператор параметризованного равномерного кроссинговера [20]. Рассмотрим реализацию параметризованного равномерного кроссинговера на примере из рис. 2. Пусть даны две родительские хромосомы, выбранные случайным образом из всей популяции. Данные хромосомы представляют собой последовательности работ проекта $P_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ и $P_2 = \{3, 1, 2, 5, 4, 8, 7, 6\}$. Для каждого нового гена в хромосоме потомке P_3 подбрасывается абстрактная «монета» для того, чтобы определить хромосому родителя, из которой будет браться ген для хромосомы потомка. Если выпадает «орел», то берется ген из хромосомы P_1 , если «решка» - из P_2 . Если в хромосоме потомке уже имеется выбранная работа, то выбирается ген из другого родителя. В таблице 1 представлен результат выполнения данного оператора, с вероятностью выпадения «орла» $P = 0.6$.

Таблица 1

Результат выполнения оператора кроссинговера

P_1	1	2	3	4	5	6	7	8
P_2	3	1	2	5	4	8	7	6
Случайное число	0.25	0.5	0.7	0.85	0.3	0.9	0.1	0.35
Вероятность	< 0.6	< 0.6	> 0.6	> 0.6	< 0.6	> 0.6	< 0.6	< 0.6
Повтор	-	-	+	-	+	-	-	+
P_3	1	2	3	5	4	8	7	6

Мутация. В качестве оператора мутации был выбран оператор мутации на основе дихотомии [21]. Его нечеткий алгоритм имеет вид:

1. Задать родительскую хромосому длины L .
2. Разделить хромосому пополам (при нечетном размере в любую часть берется большее число).
3. Определить точку мутации метода дихотомии через точку разрыва.
4. По правилам построения одноточечного оператора мутации получить новую хромосому-потомок.
5. Каждую половину хромосомы-потомка снова разделить на две части и процесс расчета продолжить по исходной схеме до тех пор, пока не будет получено заданное число хромосом-потомков.
6. Конец работы алгоритма.

Приведем пример реализации оператора мутации дихотомии. Пусть задана родительская хромосома, представляющая собой последовательность работ $P1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$, в соответствии с рис. 1. Необходимо выполнить три итерации алгоритма и получить три хромосомы-потомка: $P2, P3, P4$. Согласно алгоритму, получим:

$$\begin{array}{l}
 P1: 1\ 2\ 3\ 4\ | 5\ 6\ 7\ 8 \\
 \hline
 P2: 1\ 2\ 3\ 5\ | 4\ 6\ 7\ 8' \\
 \hline
 P2: 1\ 2\ | 3\ 5\ 4\ 6\ | 7\ 8 \\
 \hline
 P3: 1\ 3\ | 2\ 5\ 4\ 7\ | 6\ 8' \\
 P3: 1\ | 3\ 2\ 5\ 4\ 7\ 6\ | 8 \\
 \hline
 P4: 3\ | 1\ 2\ 5\ 4\ 7\ 8\ | 6'
 \end{array}$$

Применение данного оператора позволяет минимизировать количество потомков, не удовлетворяющих функции пригодности.

Транслокация. На последнем этапе генетического алгоритма выполняется оператор транслокации [22]. Данный оператор представляет собой языковую конструкцию, позволяющую на основе скрещивания и инвертирования из пары родительских хромосом создавать две хромосомы потомков. Алгоритм выполнения оператора транслокации имеет следующий вид:

1. Из заданной популяции случайным образом выбрать две альтернативных хромосомы;
2. Случайным образом выбрать точку разрыва в заданных хромосомах.
3. Выполнить скрещивание для получения хромосом-потомков.
4. В хромосомах-потомках выполнить инверсию правой части до точки разрыва;
5. Конец работы алгоритма.

Рассмотрим пример выполнения оператора транслокации. Пусть для транслокации задана последовательность работ $P_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ и $P_2 = \{3, 1, 2, 5, 4, 8, 6, 7\}$. Необходимо получить две хромосомы-потомка P'_1 и P'_2 на основе транслокации.

$$\begin{array}{l}
 P_1: 1\ 2\ 3\ 4\ 5\ | 6\ 7\ 8 \\
 P_2: 3\ 1\ 2\ 5\ 4\ | 8\ 6\ 7 \\
 \hline
 P'_1: 1\ 2\ 3\ 4\ 5\ 7\ 6\ 8' \\
 P'_2: 3\ 1\ 2\ 5\ 4\ 8\ 7\ 6
 \end{array}$$

Использование приведенных выше модифицированных генетических операторов позволит обеспечить высокую выживаемость лучших альтернативных решений и получать близкие к оптимальным результаты за отведенное время.

Выбор облачного сервиса. Риски использования облачных вычислений. Для предприятий, выполняющих планирование своих проектов при помощи облачных вычислений важно оценить риски использования различных облачных сервисов.

С точки зрения клиента облачных сервисов принято выделять следующие типы рисков:

- ◆ риски, связанные с неправомерным доступом к данным;
- ◆ риск, связанный с потерей данных;
- ◆ риски, связанные с доступностью данных.

Таким образом, поставщик облачных услуг должен обеспечивать следующие условия для минимизации рисков:

- ◆ Конфиденциальность и безопасность данных. Поставщик облачных услуг должен обеспечивать шифрование данных, а также автоматически создавать резервные копии данных.
- ◆ Управление доступом. Поставщик облачных услуг должен предоставлять возможность аутентификации пользователей и управления доступом.
- ◆ Планирование ресурсов. Поставщик облачных услуг должен обеспечивать контроль над планированием вычислительных ресурсов.

Вероятность возникновения рисков в облачной среде зачастую связана с определенными событиями, вызывающими изменения в работе облачной среды, и определяется следующей формулой [23]:

$$(r|E) = \frac{P(E|r)}{P(E)} * P(r), \quad (6)$$

где E – событие, $P(E)$ – вероятность наступления события, $P(r)$ – вероятность возникновения риска r до события E , $P(r|E)$ – вероятность возникновения риска r во время события E , $P(E|r)$ – вероятность того, что событие E приведет к возникновению риска r .

В качестве поставщика облачных услуг был выбран *Amazon Web Services* (AWS), так как AWS имеет стратегический бизнес-план, который включает идентификацию рисков и внедрение мер контроля для снижения или управления рисками.

Amazon Web Services обеспечивает защиту глобальной инфраструктуры, которая управляет всеми услугами, предлагаемыми в облаке AWS. Эта инфраструктура включает в себя оборудование, программное обеспечение, сети и средства, работающие с сервисами AWS.

Результаты вычислений. Генетический алгоритм был разработан с использованием языка программирования C++. Для эксперимента были арендованы четыре виртуальные машины *t3.2xlarge* (CPU: 8 и RAM: 32 Gb). Набор данных для апробации генетического алгоритма был взят из открытой библиотеки *PSPLIB datasets*. Для задачи *RCPSP* в данной библиотеке содержатся примеры разной степени сложности как для точных методов, так и для приближенных методов [24]. В библиотеке имеется по 480 примеров для количества работ $n = 30, 60, 90$. При апробации генетического алгоритма были использованы наборы данных *j301_1 – j301_10*, содержащие по 30 работ, *j601_1 – j601_10*, содержащие по 60 работ, *j901_1 – j901_10* и *j1201_1 – j1201_10*, содержащие по 90 работ. Результаты вычислений и сравнение их с лучшими значениями, по версии PSPLIB, представлены на рис. 6–9. На данных графиках по горизонтали указаны номера проектов, по вертикали – время выполнения проекта (меньше – лучше).

В табл. 2 представлено сравнение времени выполнения работы в секундах генетического алгоритма, запущенного в облаке и генетического алгоритма, запущенного на одной рабочей станции с процессором *AMD Ryzen 7 2700x* 3.7 ГГц и 16 Гб оперативной памяти.



Рис. 6. Результаты вычислений при количестве работ $n = 30$



Рис. 7. Результаты вычислений при количестве работ $n = 60$



Рис. 8. Результаты вычислений при количестве работ $n = 90$

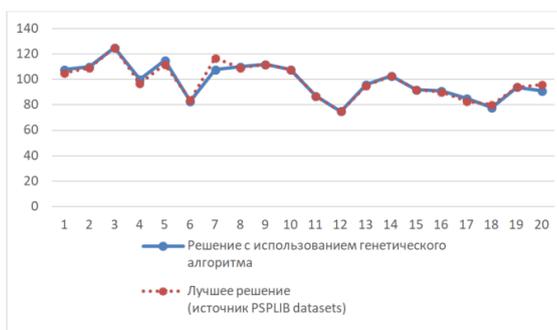


Рис. 9. Результаты вычислений при количестве работ $n = 120$

Таблица 2

Сравнение времени выполнения алгоритмов в секундах

Количество работ	МГА на основе облачных вычислений	МГА
30	0.65	1.85
60	1.1	4.4
90	1.45	7.92
120	2.1	14.2

Из анализа таблиц и графиков видно, что использование предложенного модифицированного генетического алгоритма при планировании расписания для проекта может гарантировать получение близких к оптимальным результатов за меньшее время чем модифицированный генетический алгоритм, не использующий облачных вычислений. Однако, использование предложенного алгоритма имеет смысл при общем количестве работ в проекте $n > 30$, так как время выполнения алгоритма при таком количестве работ существенно не отличается от времени выполнения модифицированного генетического алгоритма, выполненного на одной рабочей станции.

Заключение. В работе был представлен модифицированный генетический алгоритм для решения актуальной задачи построения расписания проекта с учетом ограниченности ресурсов.

Принципиальным отличием предложенного генетического алгоритма от уже имеющихся решений является использования модифицированных генетических операторов, а также использование облачных вычислений при планировании работ проектов. Данными факторами также характеризуется научная новизна исследования.

Применение модифицированных генетических операторов дает возможность оптимизировать генетический алгоритм по времени. Использование данных операторов позволяет получать близкие к оптимальным результаты за отведенное время.

Применение облачных вычислений позволяет выполнять планирование проектов с большим объемом необходимых для выполнения работ со значительно меньшей вычислительной стоимостью, в сравнении с алгоритмами, выполненными на одной рабочей станции.

Эффективность алгоритма была подтверждена экспериментом, в результате которого было установлено, что результаты вычислений, полученных с использованием предложенного генетического близки к лучшим результатам опубликованных на портале *PSPLIB datasets*, результаты вычисления некоторого количества проектов превосходят результаты, опубликованные на данном портале, а время выполнения алгоритма значительно превосходит генетический алгоритм, реализованный без применения облачных вычислений. При планировании проектов с количеством работ $n = 30$, $n = 60$, $n = 90$, $n = 120$ время выполнения предложенного алгоритма было меньше, чем время выполнения стандартного генетического алгоритма [25] в 2.8, в 4, в 5.5 и 6.8 раз соответственно. Таким образом, предложенный алгоритм может быть использован при планировании реальных проектов с большим количеством работ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Abdolshah M.* A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions/ M. Abdolshah // International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies. – 2014. – No. 1. – P. 253-286.
2. *Shirzadeh Chaleshtarti A., Shadrokh S., & Fathi Y.* Branch and bound algorithms for resource constrained project scheduling problem subject to nonrenewable resources with prescheduled procurement // Mathematical Problems in Engineering. – 2014.
3. *Azizoglu M., Çetinkaya F.C., & Pamir S.K.* LP relaxation-based solution algorithms for the multi-mode project scheduling with a non-renewable resource // European Journal of Industrial Engineering. – 2015. – No. 9 (4). – P. 450-469.
4. *Daniel Morillo-Torres, Luis Fernando Moreno-Velásquez & Francisco Javier Díaz-Serna.* A branch and bound hybrid algorithm with four deterministic heuristics for the resource constrained project scheduling problem (RCPSP). – 2014.
5. *Zhenyuan L., & Hongwei W.* Heuristic algorithm for RCPSP with the objective of minimizing activities' cost // Journal of Systems Engineering and Electronics. – 2006. – Vol. 17 (1). – P. 96-102.
6. *Zhu X., Ruiz R., Li S., & Li X.* An effective heuristic for project scheduling with resource availability cost // European Journal of Operational Research. – 2017. – Vol. 257 (3). – P. 746-762.

7. *Gutierrez-Franco E.* A Genetic Algorithm for the Resource Constrained Project Scheduling Problem (RCPSP) // *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*. – 2014. – No. 1. – P. 13-22.
8. *Yuan X., Liu J., & Wimmers M.O.* A multi-agent genetic algorithm with variable neighborhood search for resource investment project scheduling problems // In 2015 IEEE Congress on Evolutionary Computation (CEC). – 2015, May. – P. 23-30. IEEE.
9. *Vanucci S.C., Carrano E.G., Bicalho R., & Takahashi R.H.* A modified NSGA-II for the Multiobjective Multi-mode Resource-Constrained Project Scheduling Problem // In 2012 IEEE Congress on Evolutionary Computation. – 2012, June. – P. 1-7. IEEE.
10. *Vartouni A.M., & Khanli L.M.* A hybrid genetic algorithm and fuzzy set applied to multimode resource-constrained project scheduling problem // *Journal of Intelligent & Fuzzy Systems*. – 2014. – Vol. 26 (3). – P. 1103-1112.
11. *Xiao J., Wu Z., Hong X.X., Tang J.C., & Tang Y.* Integration of electromagnetism with multi-objective evolutionary algorithms for RCPSP // *European Journal of Operational Research*. – 2016. – Vol. 251 (1). – P. 22-35.
12. *Sebt M.H., Afshar M.R., & Alipouri Y.* Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem // *Engineering Optimization*. – 2017. – Vol. 49 (3). – P. 513-530.
13. *Gomes H.C., das Neves F.D.A., & Souza M.J.F.* Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations // *Computers & Operations Research*. – 2014. – Vol. 44. – P. 92-104.
14. *Kadam S.U., & Kadam N.S.* Solving resource-constrained project scheduling problem by genetic algorithm // In *Business and Information Management (ICBIM), 2014 2nd International Conference on*. – 2014, January. – P. 159-164. IEEE.
15. *Лазарев А.А., Гафаров Е.Р.* Теория расписаний. Задачи и алгоритмы: учеб. пособие. – М.: Московский государственный университет им. М.В. Ломоносова, 2011. – 222 с.
16. *Вильчинский Н.* От хранения данных к управлению информацией. – Питер, 2016. – 544 с.
17. *Гладков Л.А., Зищенко Л.А., Курейчик В.В. и др.* Методы генетического поиска. – Таганрог: Изд-во ТРТУ, 2002.
18. *Курейчик В.В., Сороколетов П.В.* Концептуальная модель представления решений в генетических алгоритмах // *Известия ЮФУ. Технические науки*. – 2008. – № 9 (86). – С. 7-12.
19. *Шевляков А.О., Матвеев М.Г.* Решение RCPSP при нечетких трудозатратах выполнения операций // *Вестник ВГУ*. – 2015. – № 4. – С. 119-123.
20. *Gonçalves José & Resende Mauricio.* A multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem // *J. Comb. Optim.* – 2011. – Vol. 22. – P. 180-201. – DOI 10.1007/s10878-009-9282-1.
21. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы: учеб, пособие. – 2-е изд. – М.: Физматлит. 2006. – 320 с.
22. *Kureichik V.V., Kureichik V.M.* Genetic search-based control // *Automation and Remote Control*. – 2001. – Vol. 62, No. 10. – P. 1698-1710.
23. *Samad J., Loke S.W. and Reed K.* Quantitative Risk Analysis for Mobile Cloud Computing: A Preliminary Approach and a Health Application Case Study // 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, VIC, 2013. – P. 1378-1385. – Doi: 10.1109/TrustCom.2013.166.
24. *Habibi, Farhad & Barzinpour, Farnaz & Sadjadi, Seyed.* Resource-constrained project scheduling problem: review of past and recent developments // *Journal of Project Management*. – 2018. – No. 3. – P. 55-88. 10.5267/j.jpm.2018.1.005.
25. *Могилев А.А.* Обзор методов решения задач теории расписаний // *Информатика, вычислительная техника и инженерное образование*. – 2019. – Т. 37, № 2.

REFERENCES

1. *Abdolshah M.* A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions/ M. Abdolshah, *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, 2014, No. 1, pp. 253-286.
2. *Shirzadeh Chaleshtarti A., Shadrokh S., & Fathi Y.* Branch and bound algorithms for resource constrained project scheduling problem subject to nonrenewable resources with prescheduled procurement, *Mathematical Problems in Engineering*, 2014.

3. Azizoglu M., Çetinkaya F.C., & Pamir S.K. LP relaxation-based solution algorithms for the multi-mode project scheduling with a non-renewable resource, *European Journal of Industrial Engineering*, 2015, No. 9 (4), pp. 450-469.
4. Daniel Morillo-Torres, Luis Fernando Moreno-Velásquez & Francisco Javier Díaz-Serna. A branch and bound hybrid algorithm with four deterministic heuristics for the resource constrained project scheduling problem (RCPSP). – 2014.
5. Zhenyuan L., & Hongwei W. Heuristic algorithm for RCPSP with the objective of minimizing activities' cost, *Journal of Systems Engineering and Electronics*, 2006, Vol. 17 (1), pp. 96-102.
6. Zhu X., Ruiz R., Li S., & Li X. An effective heuristic for project scheduling with resource availability cost, *European Journal of Operational Research*, 2017, Vol. 257 (3), pp. 746-762.
7. Gutierrez-Franco E. A Genetic Algorithm for the Resource Constrained Project Scheduling Problem (RCPSP), *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, 2014, No. 1, pp. 13-22.
8. Yuan X., Liu J., & Wimmers M.O. A multi-agent genetic algorithm with variable neighborhood search for resource investment project scheduling problems, *In 2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, May, pp. 23-30. IEEE.
9. Vanucci S.C., Carrano E.G., Bicalho R., & Takahashi R.H. A modified NSGA-II for the Multiobjective Multi-mode Resource-Constrained Project Scheduling Problem, *In 2012 IEEE Congress on Evolutionary Computation*, 2012, June, pp. 1-7. IEEE.
10. Vartoumi A.M., & Khanli L.M. A hybrid genetic algorithm and fuzzy set applied to multimode resource-constrained project scheduling problem, *Journal of Intelligent & Fuzzy Systems*, 2014, Vol. 26 (3), pp. 1103-1112.
11. Xiao J., Wu Z., Hong X.X., Tang J.C., & Tang Y. Integration of electromagnetism with multi-objective evolutionary algorithms for RCPSP, *European Journal of Operational Research*, 2016, Vol. 251 (1), pp. 22-35.
12. Sebt M.H., Afshar M.R., & Alipouri Y. Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem, *Engineering Optimization*, 2017, Vol. 49 (3), pp. 513-530.
13. Gomes H.C., das Neves F.D.A., & Souza M.J.F. Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations, *Computers & Operations Research*, 2014, Vol. 44, pp. 92-104.
14. Kadam S.U., & Kadam N.S. Solving resource-constrained project scheduling problem by genetic algorithm, *In Business and Information Management (ICBIM), 2014 2nd International Conference on*, 2014, January, pp. 159-164. IEEE.
15. Lazarev A.A., Gafarov E.R. Teoriya raspisaniy. Zadachi i algoritmy: ucheb. posobie [he scheduling theory. Problems and algorithms: textbook]. Moscow: Moskovskiy gosudarstvennyy universitet im. M.V. Lomonosova, 2011, 222 p.
16. Vil'chinskiy N. Ot khraneniya dannykh k upravleniyu informatsiy [From data storage to information management]. Piter, 2016, 544 p.
17. Gladkov L.A., Zinchenko L.A., Kureychik V.V. *i dr.* Metody geneticheskogo poiska [Genetic search methods]. Taganrog: Izd-vo TRTU, 2002.
18. Kureychik V.V., Sorokoletov P.V. Kontseptual'naya model' predstavleniya resheniy v geneticheskikh algoritmakh [A conceptual model for representing solutions in genetic algorithms], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 9 (86), pp. 7-12.
19. Shevlyakov A.O., Matveev M.G. Reshenie RCPSP pri nechetkikh trudozatratakh vypolneniya operatsiy [RcpSP solution for fuzzy labor costs of performing operations], *Vestnik VGU* [Vestnik VSU], 2015, No. 4, pp. 119-123.
20. Gonçalves José & Resende Mauricio. A multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem, *J. Comb. Optim.*, 2011, Vol. 22, pp. 180-201. Doi 10.1007/s10878-009-9282-1.
21. Gladkov L.A., Kureychik V.V., Kureychik V.M. Geneticheskie algoritmy: ucheb. posobie [Genetic algorithms: a textbook]. 2nd ed. Moscow: Fizmatlit. 2006, 320 p.
22. Kureichik V.V., Kureichik V.M. Genetic search-based control, *Automation and Remote Control*, 2001, Vol. 62, No. 10, pp. 1698-1710.
23. Samad J., Loke S.W. and Reed K. Quantitative Risk Analysis for Mobile Cloud Computing: A Preliminary Approach and a Health Application Case Study, *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, VIC, 2013*, pp. 1378-1385. Doi: 10.1109/TrustCom.2013.166.

24. *Habibi, Farhad & Barzinpour, Farnaz & Sadjadi, Seyed.* Resource-constrained project scheduling problem: review of past and recent developments, *Journal of Project Management*, 2018, No. 3, pp. 55-88. 10.5267/j.jp.m.2018.1.005.
25. *Mogilev A.A.* Obzor metodov resheniya zadach teorii raspisaniy [Review of methods for solving problems in the theory of schedules], *Informatika, vychislitel'naya tekhnika i inzhenerное obrazovanie* [Informatics, computer engineering and engineering education], 2019, Vol. 37, No. 2.

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

Могилев Алексей Алексеевич – Южный федеральный университет; e-mail: alexmogilev08@mail.ru; 347928, г. Таганрог, Некрасовский, 44; тел.: 89897258619; кафедра систем автоматизированного проектирования; аспирант.

Курейчик Виктор Михайлович – e-mail: vmkureychik@sfedu.ru; тел.: 88634311487; кафедра системы автоматизированного проектирования; д.т.н.; профессор.

Mogilev Alexey Alexeevich – Southern Federal University; e-mail: alexmogilev08@mail.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +79897258619; the department of computer aided design; postgraduate student.

Kureichik Viktor Mikhaylovich – e-mail: vmkureychik@sfedu.ru; phone: +78634311487; the department of computer aided design; dr. of eng. sc.; professor.

УДК 004.896

DOI 10.18522/2311-3103-2020-2-169-179

Э.В. Кулнев, А.А. Лежебоков, Семенова М.М., В.А. Семенов

ПОДХОД К КОДИРОВАНИЮ РЕШЕНИЙ В ЭВОЛЮЦИОННЫХ МЕТОДАХ ДЛЯ СОЗДАНИЯ ИНСТРУМЕНТАЛЬНОЙ ПЛАТФОРМЫ ПРОЕКТИРОВАНИЯ*

Рассмотрены актуальные вопросы и проведен анализ проблемы трехмерной интеграции и трехмерного моделирования, возникающей на этапе конструкторского проектирования в ходе решения задачи оптимального планирования компонентов больших и сверхбольших интегральных схем и корпусных устройств электронной вычислительной аппаратуры. Представлены и достаточно детально описаны основные преимущества применения принципов трехмерной интеграции, позволяющие эффективно организовывать производство персонализированной электроники, оптимально планировать конфигурацию больших и сверхбольших интегральных схем с учетом тепловых и энергетических характеристик. В ходе выполнения исследований авторами разработан подход к кодированию решений на основе интеллектуального механизма, который характеризуется наличием встроенных средств контроля допустимых решений. Одним из таких средств, экспериментально доказавших свою эффективность, является встроенный механизм «смертельных мутаций», учитывающий статусы генов и заранее заданные ограничения на итоговую конфигурацию корпуса проектируемого устройства. В работе предложен ряд общих подходов и конкретных алгоритмов решения задачи планирования, основывающихся на результатах исследований авторского коллектива и современных подходах к решению NP-полных задач. Важнейшим практически значимым результатом исследований обозначенной проблемы является разработанная программно-инструментальная платформа проектирования на современном кроссплатформенном языке программирования Java. Выбранная технология разработки позволяет использовать все основные достоинства современных многоядерных и многопроцессорных архитектур, по использованию программной многопоточности для реализации параллельных схем решения комбинаторных задач. Программно-инструментальная платформа обладает дружественным интерфейсом, что позволяет эффективно управлять процессом решения задачи планирования компонентов

* Работа выполнена при финансовой поддержке РФФИ, проекты № 18-01-00041 и № 18-07-00055.