

УДК 519.16

Е.А. Кумагина, Е.А. Неймарк

**РАЗРАБОТКА ГИБРИДНОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ УПОРЯДОЧЕНИЯ**Национальный исследовательский  
Нижегородский государственный университет им. Н.И. Лобачевского

Рассмотрена  $NP$ -трудная задача построения оптимального по быстродействию расписания выполнения работ в системе конвейерного типа. Представлены точный, эвристический и гибридный подходы к ее решению. Подход к решению задачи оптимизации работы в системе конвейерного типа основан на методе ветвления и связывания с целью уменьшения затрат на вычислительные потери без потери оптимального решения. Рассмотрен гибридный подход, основанный на методе ветвей и границ, а также использующий генетический алгоритм для формирования начального решения и получения верхней оценки.

В ходе вычислительного эксперимента оценивается сокращение числа рассматриваемых вершин в дереве решений метода ветвей и границ. Показано, что гибридные подходы уменьшают количество вершин дерева решений метода ветвей и границ. Приведены результаты, показывающие высокую точность приближенных решений и эффективность предложенного гибридного алгоритма.

*Ключевые слова:* метод ветвей и границ, эволюционно-генетический алгоритм, гибридный алгоритм, задача упорядочения.

**Введение**

В теории расписаний основной фокус исследования сосредоточен на вопросах оптимального распределения и упорядочения конечного числа требований, которые обслуживаются детерминированными (входные параметры известны заранее) или стохастическими (значения параметров носят вероятностный характер) системами с одним или несколькими приборами. Задачи теории расписания конвейерного типа относятся к классу задач упорядочения множества работ на множестве станков, которые имеют не только большую теоретическую ценность, но и широкое практическое применение во многих инженерных и управленческих задачах. Ресурсы конвейерной системы состоят из набора обслуживающих станков (машин, процессоров, приборов). В зависимости от особенностей задачи они являются либо идентичными, либо одинаковыми только по функциональным возможностям, но разными по быстродействию, либо различными как по возможностям, так и по быстродействию. Для решения задачи конвейерного типа необходимо найти последовательность выполнения работ, при которой значение критерия будет оптимальным. Число допустимых решений этой задачи  $n$ . Конвейерная задача может формулироваться как задача минимизации общего времени выполнения всех работ на всех станках.

Использование эвристических алгоритмов для решения таких трудоемких задач зарекомендовало себя достаточно хорошо: при помощи эвристик (в частности, эволюционно-генетического алгоритма) можно получить решения приемлемого качества при незначительных вычислительных и временных затратах [1-3]. При работе с задачей упорядочения эволюционно-генетический алгоритм (ЭГА) может использоваться как самостоятельный алгоритм для улучшения ранее полученных решений (например, при помощи жадных алгоритмов), для поиска улучшенных начальных точек ветвления или для оценки верхней границы в методе ветвей и границ. При этом необходимо принимать во внимание, что применение эвристик не гарантирует получения оптимальных решений. В данной работе дающий точное решение эволюционно-генетический алгоритм применяется для ускорения работы метода ветвей и границ.

### Содержательная постановка задачи

Рассмотрим одну из задач теории расписаний – задачу упорядочения работ в системе конвейерного типа. Она формулируется следующим образом: работы множества  $N=\{1,2,\dots,n\}$  подлежат выполнению. Для выполнения работ предоставляются станки множества  $M=\{1,2,\dots,m\}$ . Работа считается выполненной, если она прошла выполнение последовательно на  $m$ -станках, начиная с первого. Для каждой работы известно время ее выполнения на каждой машине. Технологические условия выполнения работ состоят в следующем:

- на одной машине одновременно может выполняться только одна работа;
- процесс выполнения работы протекает непрерывно;
- выполнение работы на следующем станке может начаться не раньше, чем завершится ее выполнение на предыдущем станке;
- все работы доступны сразу;
- порядок выполнения работ на станках одинаковый.

Требуется построить оптимальное (минимальное) по быстродействию расписание.

### Математическая модель

*Исходные параметры.*

$N=\{1, 2, \dots, n\}$  – множество работ;

$M=\{1, 2, \dots, m\}$  – множество станков;

$T= \|t_{ij}\|_{m \times n}$ , – матрица времен выполнения работ на станках, где  $t_{ij}$  – время выполнения работы  $j$  на станке  $i$ ,  $i=\overline{1,m}$ ,  $j=\overline{1,n}$ .

*Варьируемые параметры модели.*

Обозначим через  $\pi=(\pi_1, \pi_2, \dots, \pi_n)$  последовательность выполнения работ, где  $\pi_j$  – номер по порядку выполнения работы  $j$ ,  $j=\overline{1,n}$ .

Обозначим через  $Y= \|y_{ij}\|_{m \times n}$  матрицу времен окончания выполнения, где  $y_{ij}$  – время окончания выполнения  $j$ -ой работы на станке  $i$ ,  $i=\overline{1,m}$ ,  $j=\overline{1,n}$ .

*Ограничения математической модели.*

Выполнение работы на станке может начаться не раньше, чем завершится выполнение любой работы, выполняющейся раньше ее на этом станке (1):

$$y_{ij} \geq y_{ik} + t_{ik}, \quad (1)$$

если  $\pi_k < \pi_j$ ,  $i=\overline{1,m}$ ,  $j=\overline{1,n}$ ,  $k=\overline{1,n}$ .

Выполнение работы на очередном (в порядке обслуживания) станке начинается после завершения ее выполнения на предыдущем станке (2):

$$y_{ij} \geq y_{i-1,j} + t_{i-1,j}, \quad (2)$$

если  $i=\overline{2,m}$ ,  $j=\overline{1,n}$ .

Естественные ограничения на введенные переменные (3):

$$\pi_j \neq \pi_k, \quad j \neq k, \quad j=\overline{1,n}, \quad k=\overline{1,n}, \quad \pi_j \in \{1,2,\dots,n\}, \quad y_{ij} \geq 0, \quad i=\overline{1,m}, \quad j=\overline{1,n} \quad (3)$$

### Постановка оптимизационной задачи

Обозначим через  $F(Y, \pi)$  момент окончания выполнения последней работы на последнем станке.

Задача нахождения оптимального по быстродействию расписания будет заключаться в определении среди множества допустимых решений ( $Y$  и  $\pi$  удовлетворяют ограничениям (1)-(3)) такого, которое доставляет минимальное значение функционалу  $F$  (4):

$$F(Y, \pi) = \max_{mj} \rightarrow \min. \quad (4)$$

Для конвейерных систем вообще получено много результатов [1-6], расписания для них составляются гораздо легче, чем для систем произвольной структуры. Однако эта простота относительна и включает в себе много проблематичных аспектов.  $NP$ -трудность конвейерной задачи показана в [7]. Поскольку она относится к классу  $NP$ -трудных задач, поиск точного решения возможен только для задач небольшой размерности, и разработка эвристических методов является актуальной.

### Процедуры метода ветвей и границ для конвейерной задачи

Для решения поставленной задачи предлагается методология ветвей и границ. Чтобы сконструировать такой метод решения задачи, необходимо разработать его индивидуальные и универсальные процедуры. К индивидуальным методам, зависящим от конкретной задачи, относятся процедуры получения верхних и нижних оценок, а также процедуры ветвления. К универсальным процедурам относятся процедура отсева неперспективных направлений и процедура останова.

При построении дерева решений будем именовать вершины следующим образом:  $v = (v_1, v_2, \dots, v_k, v_{k+1}, \dots, v_n)$ , где  $k$  – уровень дерева,  $v_k \in N$ ,  $v_k \neq v_i$   $k = \overline{1, n}$ ,  $i = \overline{1, n}$ . При этом для первых  $k$ -работ порядок выполнения фиксируется и определяется путем в вершину, а для работ, начиная с номера  $k+1$  порядок выполнения не зафиксирован. На первую роль претендуют  $n$ -работы. Это  $n$ -ветвей, выходящих из корня. Для каждой ветви рассматривается вторая работа из оставшихся при фиксированной первой, и т.д.

*Стратегия ветвления.*

Для выбора очередной вершины для ветвления выбирается вершина с наименьшей нижней оценкой.

*Верхняя оценка.*

В качестве верхней оценки выбирается значение критерия на перестановке  $\pi = (\pi_1, \pi_2, \dots, \pi_k, \pi_{k+1}, \dots, \pi_n)$ , где работы  $\pi_1, \pi_2, \dots, \pi_k$  определены путем в вершину (фиксированы), а оставшиеся  $\pi_{k+1}, \dots, \pi_n$  выполняются в лексикографическом порядке.

*Нижняя оценка.*

В качестве нижней оценки для фиксированных работ  $\pi_1, \pi_2, \dots, \pi_k$  строится расписание, находится момент окончания выполнения работы  $\pi_k$ , и к этому результату добавляется сумма времён на последнем станке для оставшихся (нефиксированных) работ  $\pi_{k+1}, \dots, \pi_n$ .

*Отсев.*

На каждом шаге метода происходит удаление вершин, нижняя оценка которых больше, чем самая меньшая найденная верхняя граница (рекорд).

*Остановка.*

Остановка метода происходит, когда в дереве решений останется одна вершина, у которой совпадают верхняя и нижняя оценки.

В ходе расчетов сохраняется рекорд: перестановка, на которой достигается минимальная верхняя оценка.

Использование метода ветвей и границ для решения задачи упорядочения работ в конвейерной системе позволяет найти точное решение, однако этот процесс требует больших вычислительных и временных затрат. При данном подходе к решению задачи одной из наиболее критичных операций является получение значения верхней границы, поскольку уменьшение верхней оценки позволит сократить размер дерева решений. Для поиска верхней оценки предлагается использовать эволюционно-генетический алгоритм.

### Процедуры генетического алгоритма для конвейерной задачи

Для разработки эволюционно-генетического алгоритма решения конкретной задачи необходимо определить кодировку решений, сопоставить критерий и функцию приспособ-

ленности, определить процедуры мутации, скрещивания и отбора [8]. Работая одновременно с несколькими допустимыми решениями, ЭГА позволяет найти удовлетворительное решение аналитически неразрешимых проблем, зачастую довольно близкое к оптимальному. Алгоритм в ходе работы моделирует процессы популяционной генетики (мутация, скрещивание, отбор и выживание сильнейших). Настройка его эффективности осуществляется за счет последовательного подбора и комбинирования параметров алгоритма [9].

#### *Кодирование решений.*

Одним из важных этапов формирования среды для работы ЭГА является выбор способа кодирования решений. Решением конвейерной задачи является перестановка  $n$  элементов, соответствующих номерам работ, поэтому целесообразно использовать  $n$ -арное кодирование [8]. Каждая позиция в кодирующей строке может принимать одно из значений в пределах от 1 до  $n$ . Помимо кодировки, для представления особи необходимо задать ее приспособленность.

#### *Функция приспособленности.*

Функция приспособленности в ЭГА играет роль целевой функции, она должна давать оценку кодировки особи для выяснения ее пригодности (полезности). В данной работе функция приспособленности вычисляет по перестановке(кодировке) общее время, требуемое для отработки этой перестановки на всех станках. В классической постановке ЭГА решается задача на максимум, то есть ищется особь, обладающая наибольшей приспособленностью. В рассматриваемом случае нас интересует особь с наименьшей приспособленностью, т.е., решается задача на минимум.

#### *Формирование начальной популяции.*

Одним из параметров ЭГА является размер популяции, который не меняется в ходе работы всего алгоритма. В данной работе численность популяции совпадает с количеством работ и равно  $n$ . Вторым важным вопросом, связанным с популяцией, является проблема генерации начальной популяции [9]. Основным требованием к начальной популяции является разнообразие базовых особей. В данной работе применяется следующий способ создания начальной популяции. Первая базовая особь – это лексикографическое упорядочение всех номеров работ. Каждая следующая особь получается путем проведения  $n-1$  транспозиции соседних элементов. Таким образом, будет построено  $n$  перестановок, наиболее удаленных друг от друга в метрике, предложенной в [2].

#### *Основной цикл ЭГА.*

Основной цикл ЭГА состоит в последовательном применении операторов кроссовера, мутации и селекции. В качестве кроссовера применяется оператор РМХ [9]. Данный вид кроссовера требует выбора двух родительских особей, которые выбираются из популяции случайно (панмиксия). Оператор *мутации* производит с указанной вероятностью обмен местами двух случайных генов (элементов) кодировки (перестановки). В качестве оператора селекции используется Бета-турнир. Оператор селекции производит отбор особей из текущей популяции. Для этого из всех полученных в данной популяции особей выбирается случайно  $\beta$  особей (где  $\beta$  – параметр алгоритма), лучшая по приспособленности особь копируется в следующее поколение, отбор и копирование происходит в цикле. Количество циклов совпадает с численностью популяции. Остановка производится по окончании указанного количества поколений, что также является параметром алгоритма.

## **Гибридный алгоритм**

Метод ветвей и границ позволяет получить точное решение, но процесс его нахождения требует весьма больших затрат памяти и времени. ЭГА позволяет найти приближенное решение, но качество алгоритма надо доказывать экспериментально. Целесообразно сочетать сильные стороны обоих подходов. Предполагается, что гибридный подход поможет увеличить эффективность поиска оптимального решения в методе верей и границ за счет умень-

шения времени и количества вычислений. Применение ЭГА возможно, как на этапе формирования начального решения для метода ветвей и границ, так и для вычисления верхней оценки.

В приведенном выше описании метода ветвей и границ в качестве верхней оценки выбиралось значение критерия на перестановке  $\pi=(\pi_1, \pi_2, \dots, \pi_k, \pi_{k+1}, \dots, \pi_n)$ , где порядок выполнения первых  $k$  работ определялся путем в вершину, а оставшиеся работы выполнялись в лексикографическом порядке. Для сужения области перебора требуется сделать верхнюю оценку в вершинах как можно меньшей. Предлагается использовать «ускоренную» версию ЭГА для поиска решения для незафиксированных работ. В этом случае количество работ у нас сокращено до  $k < n$ , тем самым сокращается длина кодировки и размер популяции, что повлияет на общее быстроедействие алгоритма.

### Вычислительный эксперимент

Для выяснения эффективности разработанного гибридного подхода был проведен вычислительный эксперимент. Был программно реализован генератор задач, задающий данные для конвейерных задач с заданным числом станков в диапазоне [5, 20] и числом работ в диапазоне [10, 50] со случайными целочисленными значениями времен выполнения из диапазона [0, 100]. Каждая задача решалась последовательно всеми разработанными алгоритмами:

- 1) методом ветвей и границ с лексикографически упорядоченной начальной перестановкой;
- 2) эволюционно-генетическим алгоритмом;
- 3) гибридным алгоритмом № 1 (с применением ЭГА для формирования начального решения для метода ветвей и границ);
- 4) гибридным алгоритмом № 2 (с применением ЭГА для вычисления верхних оценок в методе ветвей и границ и лексикографическим начальным упорядочением);
- 5) гибридным алгоритмом № 3 (с применением ЭГА для формирования начального решения и верхних оценок метода ветвей и границ).

Решения, полученные методом ветвей и границ с лексикографически упорядоченной начальной перестановкой, принимались в качестве контрольных. Число вершин ветвления, оцениваемых в дереве решения при работе метода ветвей и границ, использовалось для оценки сокращения числа просмотренных вершин при работе гибридных алгоритмов. Для каждого запуска ЭГА высчитывалось относительное отклонение от оптимума, полученного методом ветвей и границ:

$$\text{относительное отклонение от оптимума} = \frac{F_{\text{opt}} - F_{\text{ЭГА}}}{F_{\text{opt}}} \cdot 100\% .$$

Настройки ЭГА:

- шагов 100;
- размер популяции  $n$ ;
- бета-турнир на 3 особи;
- вероятность мутации 1 %.

В результате применения ЭГА среднее относительное отклонение приближенных решений, составило 2,86 %. Процент запусков, в которых получено точное решение составил 95 %. Для гибридных алгоритмов высчитывалось среднее сокращение числа вершин дерева решений.

В табл. 1 приведены результаты вычислительного эксперимента.

Таблица 1

## Результаты вычислительного эксперимента

| Гибридный алгоритм | Сокращение числа вершин дерева решений, % |
|--------------------|---|
| № 1                | 15,1                                      |
| № 2                | 7,3                                       |
| № 3                | 22,7                                      |

По результатам вычислительного эксперимента видно, что разработанный ЭГА дает в среднем высокую точность решений, и при этом точное решение получено в 95 % запусков. Поскольку получаемое при помощи ЭГА начальное решение близко к точному, применение этого решения в качестве начального в методе ветвей и границ позволило сократить число вершин в дереве решений на 15,1 %. Применение ЭГА для подсчета верхней оценки в методе ветвей и границ при лексикографическом начальном упорядочении дает незначительное сокращение дерева решений. Гибридный алгоритм, в котором ЭГА применяется как для генерации начального решения, так и для получения верхних оценок, оказался самым эффективным по используемой в ходе решения памяти, но вместе с тем – и самым затратным по времени.

## Заключение

В работе рассмотрена *NP*-трудная задача построения оптимального по быстродействию расписания выполнения работ в системе конвейерного типа. Представлены точный, эвристический и гибридный подходы к ее решению. Приведены результаты вычислительного эксперимента, показавшие высокую точность приближенных решений и эффективность предложенного гибридного алгоритма.

## Библиографический список

1. Прилуцкий, М.Х. Задача упорядочения работ как задача о назначениях / М.Х. Прилуцкий, Е.А. Кумагина // Вестник Нижегородского университета им. Н.И. Лобачевского. – 1999. – № 2. – С. 270-275.
2. Кумагина, Е.А. Метаэвристический алгоритм решения задач упорядочения / Е.А. Кумагина, Д.В. Бужков // Интеллектуальные информационные системы: труды Всероссийской конференции с международным участием. – Воронеж: Воронежский государственный технический университет. – 2017. – Ч.2. – С. 40-44.
3. Неймарк, Е.А. Задача определения оптимального порядка выполнения операций технического контроля для микроэлектронного производства / Е.А. Неймарк, Д.В. Седаков // Системы управления и информационные технологии. – 2015. – Т. 59. – № 1. – С. 100-104.
4. Танаев, В.С. Теория расписаний. Многостадийные системы / В.С. Танаев, Ю.Н. Сотсков, В.А. Струсевич. – М.: Наука, 1989. – 328 с.
5. Прилуцкий, М.Х. Построение оптимальных по быстродействию расписаний в канонических системах «конвейер-сеть» / М.Х. Прилуцкий, В.С. Власов // Информационные технологии. – 2011. – № 3. – С. 26-31.
6. Прилуцкий, М.Х. Метод ветвей и границ решения задачи многоресурсного сетевого планирования / М.Х. Прилуцкий, Е.А. Кумагина // Системы управления и информационные технологии. – 2014. – №2(56). – С. 48-51.
7. Гэри, Н. Вычислительные машины и труднорешаемые задачи / Н. Гэри, Д. Джонсон. – М.: Мир, 1992.
8. Неймарк, Е.А. Решение нестационарной задачи о ранце при помощи генетического алгоритма // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2006. – № 3. – С. 133-138.
9. Неймарк, Е.А. Улучшение качества начальной популяции эволюционно-генетического алгоритма для задачи коммивояжера // Вестник Волжской государственной академии водного транспорта. – 2017. – № 50. – С. 69-73.

Дата поступления  
в редакцию: 25.06.2019

**Е.А. Kumagina, Е.А. Neumark**

## **A HYBRID ALGORITHM FOR PERMUTATION FLOWSHOP SCHEDULING**

Lobachevsky state university of Nizhni Novgorod

**Purpose:** the article discusses an approach to solving the task of streamlining work in a conveyor-type system based on the branch and bound method in order to reduce computational cost without losing the optimal solution.

**Approach:** the article presents a hybrid approach of solving the tasks of streamlining work in a conveyor-type system based on the branch and bound method and using a genetic algorithm to form initial permutations and obtain upper estimates.

**Findings:** during the computational experiment the reduction of the number of vertices of decision tree of the branch and bound method is evaluated. It is shown: hybrid approaches reduce the number of vertices of decision tree of the branch and bound method.

**Originality/value:** all results are new.

*Keywords:* branch and bound method, genetic algorithm, hybrid algorithm, permutation problem.