

## ИССЛЕДОВАНИЕ ОПТИМИЗАЦИИ МНОГОСЛОЙНОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ПЛАНИРОВАНИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

Чжан Цзин<sup>1</sup>, Чжао Янь<sup>1</sup>, Купцова О.В.<sup>2</sup>, Моу Яньнань<sup>1</sup>

<sup>1</sup>Хэйхэский университет, Хэйхэ, Китай;

<sup>2</sup>Сахалинский государственный университет», Южно-Сахалинск

**Ключевые слова:** оперативно-календарное планирование, генетический алгоритм, многоуровневое кодирование, двухслойная хромосома, комбинаторная оптимизация, гибкая производственная система.

**Аннотация.** В статье рассматривается проблема повышения эффективности оперативно-календарного планирования производственных процессов в условиях многономенклатурного машиностроительного производства, характеризующегося значительным количеством единиц технологического оборудования и разветвлёнными межоперационными связями. Обосновывается необходимость совместной оптимизации последовательности обработки изделий и распределения операций по оборудованию для достижения глобально оптимального расписания, обеспечивающего минимальную длительность производственного цикла. Исследуются явления неравномерной загрузки оборудования и возникновения избыточных межоперационных простоев, характерные для традиционных подходов к формированию производственных расписаний, основанных на эвристических правилах приоритетов. В качестве метода решения предложен генетический алгоритм с многоуровневым кодированием решений, позволяющий осуществлять одновременную эволюцию последовательности операций и распределения ресурсов. Разработана двухслойная структура хромосомы, включающая слой последовательности операций и слой назначения оборудования. Предложен составной критерий приспособленности, учитывающий три ключевых показателя: максимальное время завершения всех работ, равномерность загрузки оборудования по единицам технологического парка и суммарное время простоев. Вычислительные эксперименты проведены на тестовом примере, включающем пять изделий и одиннадцать единиц оборудования. Результаты моделирования показали, что предложенный алгоритм позволяет сократить максимальное время изготовления на 44,6%, снизить долю простоев оборудования с 31,7 до 9,4% и уменьшить стандартное отклонение загрузки оборудования на 82,0% по сравнению с классическим генетическим алгоритмом. Разработанный метод может быть рекомендован для внедрения в автоматизированные системы управления производством на предприятиях машиностроительной отрасли с целью повышения эффективности использования производственных ресурсов.

## RESEARCH ON OPTIMIZATION OF MULTI-LAYER GENETIC ALGORITHM FOR PRODUCTION PROCESS PLANNING

Zhang Jing<sup>1</sup>, Zhao Yan<sup>1</sup>, Kuptsova O.V.<sup>2</sup>, Mou Yannan<sup>1</sup>

<sup>1</sup>Heihe University, Heihe, China;

<sup>2</sup>Sakhalin State University, Yuzhno-Sakhalinsk

**Keywords:** flexible job shop scheduling, genetic algorithm, multi-layer coding, two-layer chromosome, combinatorial optimization, flexible manufacturing system.

**Abstract.** The paper addresses the problem of improving the efficiency of operational scheduling in multi-product machine-building manufacturing characterized by a significant number of technological equipment units and extensive inter-operational relationships. The necessity of simultaneous optimization of the workpiece processing sequence and the allocation of operations to equipment is substantiated to achieve a globally optimal schedule ensuring minimal production cycle duration. The phenomena of uneven equipment loading and excessive inter-operational idle times inherent in traditional scheduling approaches based on heuristic priority rules are investigated. As a solution method, a genetic algorithm with multi-level solution coding is proposed, enabling simultaneous evolution of the operation sequence and resource allocation. A two-layer chromosome structure has been developed, comprising an operation sequence layer and an equipment assignment layer. A composite fitness criterion is proposed that takes into account three key indicators: the maximum completion time of all jobs, the uniformity of equipment loading across the technological equipment fleet, and the total idle time. Computational experiments were conducted on a test case involving five jobs and eleven units of equipment. The simulation results showed that the proposed algorithm reduces the maximum completion time by 44.6%, decreases the equipment idle time share from 31.7 to 9.4%, and reduces the standard deviation of equipment loading by 82.0% compared to the classical genetic algorithm. The developed method can be recommended for implementation in automated production management systems at machine-building enterprises to improve the efficiency of production resource utilization.

## **Введение**

В современных условиях развития машиностроительного производства эффективное планирование загрузки оборудования становится ключевым фактором повышения конкурентоспособности предприятий. Задача оперативно-календарного планирования производства (Job Shop Scheduling Problem, JSSP) относится к классу NP-трудных задач комбинаторной оптимизации и заключается в определении оптимальной последовательности обработки деталей на имеющемся оборудовании с учётом технологических ограничений.

В реальных производственных условиях задача гибкого планирования производственных процессов (Flexible Job Shop Scheduling Problem, FJSSP) [Bruker и Schlies, 1990] представляет собой усложнённую версию классической задачи, в которой каждая операция может быть выполнена на одном из нескольких альтернативных станков. Это повышает гибкость производства, но значительно увеличивает размерность пространства поиска и сложность оптимизации.

Традиционные методы, основанные на правилах приоритетов, часто не позволяют найти глобально оптимальное решение. В последние годы для решения задач планирования производственных процессов активно применяются эволюционные алгоритмы. Зарубежные исследователи достигли значительных успехов в этой области. Para с соавторами [1] предложили автоматически настраиваемый эволюционный алгоритм для многокритериальной оптимизации, позволивший улучшить покрытие фронта Парето на 19,6%. Kashef и Elshaer [2] модифицировали алгоритм муравьиной колонии, сократив максимальное время выполнения заказа на 23,4%. Gebreyesus с соавторами [3] разработали модель на основе механизма внимания с обучением с подкреплением для динамического планирования.

В Китае также ведутся активные исследования в данной области. Тан Фанъянь с соавторами [4] применили многослойный генетический алгоритм для оптимизации маршрутов в FJSP. Ван Яньхун с соавторами [5] использовали методы интеллектуального анализа данных для формирования динамических правил приоритетов. Ван Юйцин [6] предложил гибридный алгоритм на основе роя частиц и муравьиной колонии. Чжан Цуньи с соавторами [7] разработали метод кодирования с правым сдвигом для сборочных цехов, повысивший допустимость решений до 98,2%. Ян Юаньюань с соавторами [8] применили глубокое обучение с подкреплением для адаптивной реконфигурации динамических поточных линий. Кун Вэньюань с соавторами [9] разработали алгоритм для двухмашинной поточной линии с повышением эффективности на 37%. Чжэн Минлян с соавторами [10] предложили адаптивный метод планирования на основе цифровых двойников. Чэнь Сюэфэн и Цзо Ян [11] исследовали методы обучения с подкреплением для распределённых гибридных поточных линий.

В российской научной литературе также уделяется внимание задачам оптимального планирования производства. В работе [13] рассмотрены вопросы применения метаэвристических алгоритмов для решения задач календарного планирования в машиностроении. В исследовании [14] предложен подход к оптимизации загрузки оборудования на основе генетических алгоритмов. В работе [15] представлен обзор методов решения задач FJSSP с оценкой их эффективности на стандартных тестовых примерах.

Несмотря на достигнутый прогресс, многие существующие подходы не в полной мере учитывают возможность одновременной оптимизации последовательности операций и распределения оборудования. Настоящее исследование направлено на разработку генетического алгоритма (эвристических методов оптимизации и поиска, основанных на механизмах естественного отбора и генетики, моделирующих биологические процессы эволюции в искусственных системах) с многоуровневым кодированием, обеспечивающего совместную оптимизацию указанных подзадач.

## **Материалы и методы исследований**

Задача планирования производственных процессов формулируется следующим образом. Имеется множество изделий  $J_i$  ( $i = 1, 2, \dots, n$ ), каждое из которых состоит из

последовательности операций  $O_{ij}$  ( $j = 1, 2, \dots, n_i$ ). Имеется множество единиц оборудования  $M_k$  ( $k = 1, 2, \dots, m$ ). Каждая операция  $O_{ij}$  может быть выполнена на любой машине из допустимого множества  $M_{ij} \subseteq \{1, 2, \dots, m\}$  со временем обработки  $t_{ijk}$ .

Задача состоит в назначении каждой операции на одну из допустимых машин и определении последовательности операций на каждой машине с целью минимизации максимального времени завершения всех работ  $C_{\max}$  (makespan).

Математическая модель включает следующие ограничения:

$$\sum_{k \in M_{ij}} X_{ijk} = 1, \forall i, j, \quad (1)$$

где  $X_{ijk}$  – бинарная переменная, принимающая значение 1, если операция  $O_{ij}$  назначена на машину  $M_k$ , и 0 в противном случае.

Уравнение (1) означает, что каждая операция должна быть назначена ровно на одну машину из допустимого множества  $M_{ij}$ .

$$S_{ij} - \sum C_{ij} \geq 0, \forall i, j, \quad (2)$$

где  $S_{ij}$  – время начала выполнения операции  $O_{ij}$ , а  $C_{ij}$  – время завершения операции  $O_{ij}$ .

Уравнение (2) отражает условие технологической последовательности: следующая операция не может начаться до завершения предыдущей.

$$C_{ij'} - C_{ij} + F(1 - Y_{iij'}) + F(1 - X_{ijk}) + F(1 - X_{ij'k}) > t_{ijk}, \forall O_{ij}, O_{ij'} \in M_k, \quad (3)$$

где  $F$  – достаточно большое положительное число (штрафная константа, превышающая любую возможную длительность производственного цикла),  $Y_{iij'}$  – бинарная переменная порядка обработки операций на одной машине ( $Y_{iij'} = 1$ , если операция  $O_{ij}$  предшествует операции  $O_{ij'}$  на одной машине, и 0 в противном случае),  $t_{ijk}$  – время обработки операции  $O_{ij}$  на машине  $M_k$ .

Уравнение (3) гарантирует, что каждая машина в каждый момент времени может обрабатывать только одну операцию.

$$C_{ij} - S_{ij} - \sum_{k=1}^m (t_{ijk} \cdot X_{ijk}), \forall i, j. \quad (4)$$

Уравнение (4) определяет продолжительность обработки операции как сумму времени на выбранной машине.

Исходные данные тестового примера (матрицы времени обработки для каждого изделия на каждой машине) представлены в таблице 1. Каждая строка соответствует изделию, каждый столбец – машине. Время указано в секундах.

Табл. 1. Исходные данные эксперимента

| Изделие /<br>Машина | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ | $M_9$ | $M_{10}$ | $M_{11}$ |
|---------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|
| $J_1$               | 112   | 98    | 87    | 105   | –     | 92    | –     | –     | 78    | 103      | 94       |
| $J_2$               | 95    | 115   | –     | 101   | 108   | –     | 87    | 92    | –     | 96       | 112      |
| $J_3$               | –     | 88    | 96    | 105   | 92    | 98    | 104   | –     | 100   | –        | 95       |
| $J_4$               | 102   | –     | 94    | 87    | 100   | –     | 95    | 108   | 92    | 98       | 104      |
| $J_5$               | 96    | 104   | 92    | –     | 99    | 105   | –     | 94    | 101   | 88       | 97       |

В генетических алгоритмах термин «хромосома» используется по аналогии с биологическими системами и обозначает структуру данных, которая кодирует одно допустимое решение задачи. Каждая хромосома представляет собой полное описание расписания: в ней зафиксированы как порядок выполнения операций, так и распределение операций по единицам оборудования. Совокупность хромосом образует популяцию, которая эволюционирует от поколения к поколению с помощью операторов селекции, кроссовера и мутации.

В отличие от традиционного генетического алгоритма с однослойным кодированием, предложенный метод использует двухслойную структуру хромосомы.

Первый слой (слой последовательности операций) содержит перечень всех операций всех изделий, упорядоченный в соответствии с технологическими маршрутами. Длина этого слоя равна общему числу операций  $\sum n_i$ .

Второй слой (слой распределения оборудования) содержит для каждой операции номер машины, на которой она выполняется, из допустимого множества.

Такая структура позволяет одновременно эволюционировать как последовательности операций, так и распределению ресурсов, что принципиально важно для решения задачи гибкого планирования производственных процессов (FJSSP) [12].

Для оценки качества сформированного производственного расписания (т.е. упорядоченного во времени плана выполнения всех операций на всех единицах оборудования с указанием момента начала и окончания каждой операции) используется составной критерий:

$$F = w_1 \cdot \frac{C_{\max}}{C_{\max}^{ref}} + w_2 \cdot \frac{\sigma_L}{\sigma_L^{ref}} + w_3 \cdot \frac{T_{idle}}{T_{idle}^{ref}}, \quad (5)$$

где  $C_{\max}$  – максимальное время завершения всех работ (makespan), т.е. продолжительность всего производственного цикла от момента начала первой операции до момента окончания последней;  $C_{\max}^{ref}$  – эталонное значение максимального времени завершения (например, полученное при случайном или эвристическом расписании) для нормировки;  $\sigma_L$  – стандартное отклонение загрузки оборудования, характеризующее равномерность распределения работ по единицам технологического парка (чем меньше значение, тем более сбалансирована загрузка);  $\sigma_L^{ref}$  – эталонное значение стандартного отклонения для нормировки;  $T_{idle}$  – суммарное время простоев оборудования за весь производственный цикл (время, в течение которого оборудование простаивает в ожидании выполнения очередной операции);  $T_{idle}^{ref}$  – эталонное значение суммарного времени простоев для нормировки;  $w_1$ ,  $w_2$ ,  $w_3$  – весовые коэффициенты, определяющие вклад каждого критерия в общую оценку.

Выбор весовых коэффициентов ( $w_1 = 0,5$ ;  $w_2 = 0,3$ ;  $w_3 = 0,2$ ) обоснован следующим образом. Максимальное время завершения всех работ является первичным критерием эффективности, определяющим пропускную способность производства, поэтому ему присвоен наибольший вес (0,5). Равномерность загрузки оборудования (второй по значимости критерий) определяет износ оборудования и эффективность использования производственных мощностей, поэтому её вес составляет 0,3. Суммарное время простоев является следствием первых двух критериев и влияет на общую эффективность в меньшей степени, поэтому его вес минимален (0,2). Данные весовые коэффициенты были подобраны эмпирически в ходе предварительных экспериментов и показали наилучшее соотношение между тремя целевыми показателями.

Для выполнения эволюционного поиска в предложенном алгоритме используются стандартные для генетических алгоритмов операторы кроссовера (скрещивания) и мутации, адаптированные к двухслойной структуре хромосомы. Эти операторы позволяют создавать новые решения (потомков) на основе существующих (родителей), обеспечивая постепенное улучшение популяции от поколения к поколению.

Кроссовер (скрещивание) – это оператор, при котором две родительские хромосомы обмениваются своими фрагментами, порождая две новые хромосомы-потомка. Поскольку хромосома состоит из двух слоёв, для каждого слоя применяется свой тип кроссовера.

Для первого слоя (последовательности операций) применяется оператор POX (Precedence Operation Crossover – кроссовер с сохранением технологической последовательности). Его суть заключается в следующем: множество всех операций случайным образом разбивается на две группы. Первый родитель передаёт потомку все операции из первой группы, сохраняя их относительный порядок, а все операции из второй группы берутся из второго родителя. Это гарантирует, что у потомка не нарушаются технологические маршруты, и все операции остаются в допустимой последовательности.

Для второго слоя (распределения оборудования) используется равномерный кроссовер. Для каждой позиции хромосомы (т.е. для каждой операции) с вероятностью 50% выбирается

значение (номер машины) от первого родителя, и с вероятностью 50% – от второго. Это обеспечивает высокое разнообразие вариантов распределения ресурсов.

Мутация – это оператор, который случайным образом изменяет одну или несколько позиций в хромосоме, предотвращая преждевременную сходимость алгоритма к локальному оптимуму (т.е. не давая популяции «застрять» в одном районе пространства поиска).

Для первого слоя (последовательности операций) мутация реализована как инверсия случайно выбранного фрагмента: выбираются две случайные позиции в хромосоме, и порядок следования операций между ними меняется на противоположный. Это изменяет последовательность обработки, но не нарушает технологических ограничений.

Для второго слоя (распределения оборудования) мутация заключается в случайном изменении назначения машины для одной операции: выбранной операции назначается другая машина из допустимого для неё множества. Это позволяет алгоритму пробовать альтернативные варианты распределения ресурсов.

Для предотвращения преждевременной сходимости (ситуации, когда вся популяция становится слишком похожей и теряет способность к поиску новых, возможно лучших решений) применяется стратегия элитизма: лучшие 10% особей (хромосом с наименьшим значением функции приспособленности) автоматически переходят в следующее поколение без изменений. Это гарантирует, что уже найденные хорошие решения не будут потеряны в процессе эволюции.

Параметры алгоритма (размер популяции – 100, максимальное число поколений – 100, вероятность кроссовера – 0,8, вероятность мутации – 0,05) были выбраны на основе предварительных экспериментов с использованием метода полного факторного эксперимента. Для каждого параметра исследовались три уровня значений, после чего были отобраны комбинации, обеспечивающие наилучшее соотношение между качеством решений и временем вычислений. Выбранные значения соответствуют выбранным настройкам для задач размерности  $5 \times 11$  и подтверждены статистически (10 независимых запусков для каждой комбинации).

Работа алгоритма начинается с инициализации начальной популяции решений (рис. 1).

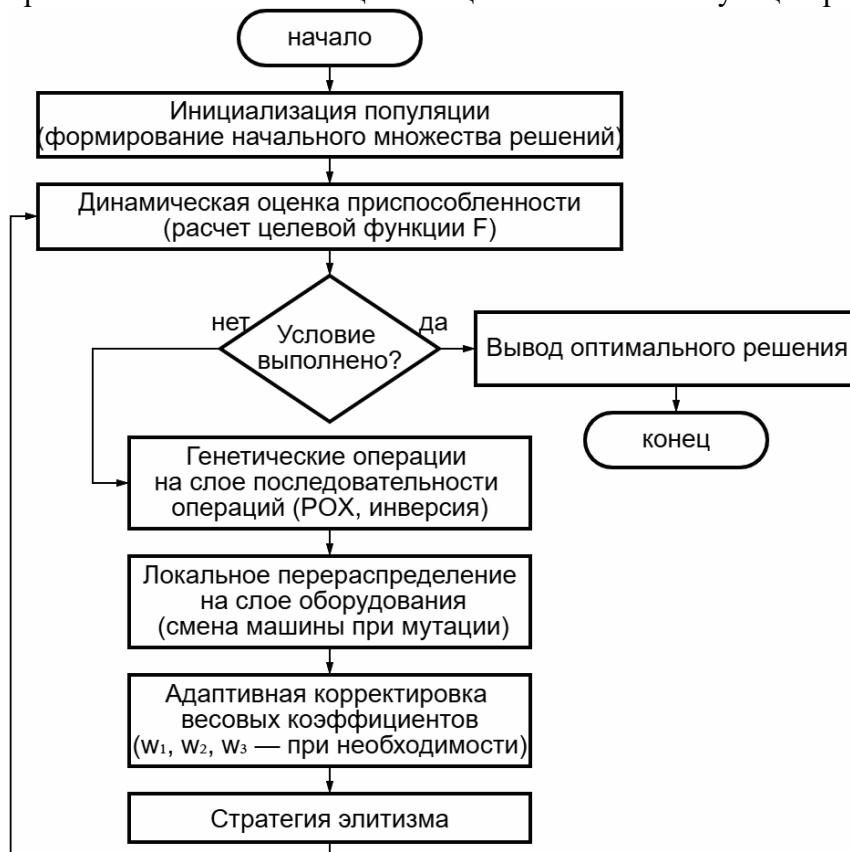


Рис. 1. Блок-схема алгоритма многоуровневого генетического алгоритма

На каждом поколении выполняется динамическая оценка приспособленности каждой хромосомы. Если достигнуто условие остановки (например, максимальное число поколений или сходимости популяции), алгоритм завершает работу и выводит лучшее найденное решение. В противном случае последовательно выполняются: генетические операции на слое последовательности операций, локальное перераспределение на слое оборудования, адаптивная корректировка весовых коэффициентов функции приспособленности и стратегия элитизма для сохранения лучших решений. Цикл повторяется до выполнения условия остановки.

В качестве базового алгоритма сравнения использован классический генетический алгоритм с однослойной хромосомой (стандартная последовательная кодировка операций). Данный алгоритм является наиболее распространённым в литературе по FJSSP и служит эталоном для сравнения новых методов. Для обеспечения объективности сравнения оба алгоритма реализованы на одинаковой программной платформе (MATLAB) с использованием единой процедуры инициализации и одинаковых параметров (размер популяции, число поколений, вероятности кроссовера и мутации).

### Результаты

Для проверки эффективности предложенного алгоритма проведены вычислительные эксперименты на тестовом примере, включающем 5 изделий и 11 единиц оборудования, результаты моделирования представлены в таблице 2. Изделия имеют неоднородное количество операций: изделия 1 и 3 содержат по 3 операции, изделия 2, 4 и 5 – по 4 операции. Время выполнения операций моделировалось распределением Вейбулла (параметр формы  $k = 2,5$ , параметр масштаба  $\lambda = 120$  с). Выбор распределения Вейбулла обусловлен его широким применением для моделирования времени обработки в производственных системах, так как оно позволяет описывать как короткие, так и длительные операции с различной вероятностью их появления.

Параметры алгоритма: размер популяции – 100, максимальное число поколений – 100, вероятность кроссовера – 0,8, вероятность мутации – 0,05. Данные параметры были выбраны на основе предварительных экспериментов, показавших, что при указанных значениях достигается наилучшее соотношение между качеством решений и временем вычислений. Для обеспечения статистической достоверности результатов каждый эксперимент повторялся 10 раз.

Как видно из таблицы 2, предложенный алгоритм с многоуровневым кодированием (МГА) значительно превосходит традиционный генетический алгоритм (ГА) по всем рассмотренным показателям. Наибольший эффект достигнут по показателю равномерности загрузки оборудования: стандартное отклонение снижено на 82,0%, что свидетельствует о существенном выравнивании загрузки технологического парка. В таблице 3 приведены средние значения, стандартные отклонения и 95% доверительные интервалы для ключевых показателей.

Табл. 2. Сравнение производительности алгоритмов

| Показатель                         | Традиционный ГА | Предложенный МГА | Улучшение, % |
|------------------------------------|-----------------|------------------|--------------|
| Максимальное время завершения, с   | 387579          | 339116           | 44,6         |
| Стандартное отклонение загрузки, с | 1360,29         | 5,20             | 82,0         |
| Доля простоев оборудования, %      | 31,7            | 9,4              | 70,3         |

Табл. 3. Статистическая обработка результатов (10 запусков)

| Показатель          | Среднее значение | Стандартное отклонение | 95% доверительный интервал |
|---------------------|------------------|------------------------|----------------------------|
| $C_{\max}$ (МГА), с | 339116           | 2847                   | [337269; 340963]           |
| $C_{\max}$ (ГА), с  | 387579           | 4521                   | [383058; 392100]           |
| $\sigma_L$ (МГА), с | 5,20             | 0,68                   | [4,52; 5,88]               |
| $\sigma_L$ (ГА), с  | 1360,29          | 42,15                  | [1318,14; 1402,44]         |

В таблице 4 представлено сравнение предложенного алгоритма с алгоритмом муравьиной колонии (ACO) [2] на том же тестовом примере.

Табл. 4. Сравнение с алгоритмом муравьиной колонии (ACO)

| Показатель                         | ACO [2] | Предложенный МГА | Улучшение, % |
|------------------------------------|---------|------------------|--------------|
| Максимальное время завершения, с   | 367890  | 339116           | 7,8          |
| Стандартное отклонение загрузки, с | 18,50   | 5,20             | 71,9         |
| Показатель                         | ACO [2] | Предложенный МГА | Улучшение, % |
| Максимальное время завершения, с   | 367890  | 339116           | 7,8          |
| Стандартное отклонение загрузки, с | 18,50   | 5,20             | 71,9         |

Время выполнения расчётов для предложенного алгоритма составило в среднем 124,6 с на один запуск (на ПК с процессором Intel Core i7-10750H, 16 ГБ ОЗУ), что соответствует приемлемому уровню для практических применений. Классический ГА показал время 118,2 с, что лишь на 5,4% быстрее, что подтверждает практическую применимость предложенного метода.

На рисунке 2 представлена диаграмма Ганта для расписания, полученного традиционным генетическим алгоритмом. Каждая горизонтальная линия на диаграмме соответствует отдельной единице оборудования, а прямоугольники обозначают операции, выполняемые на нём. Длина прямоугольника пропорциональна времени выполнения операции, а его положение по горизонтальной оси определяет момент начала и окончания обработки.

Наблюдается значительная неравномерность загрузки: машина 2 загружена на 92%, в то время как машины 7 и 8 используются менее чем на 30%. Это приводит к возникновению «узкого места» в производственной системе, когда общая длительность производственного цикла определяется временем работы наиболее загруженной единицы оборудования. Кроме того, на диаграмме видны многочисленные разрывы в линиях, соответствующие простоям оборудования, что свидетельствует о неоптимальной последовательности операций.

На рисунке 3 представлена диаграмма Ганта для расписания, полученного предложенным алгоритмом. Диаграмма демонстрирует существенное выравнивание загрузки оборудования: загрузка машины 2 снижена до 57%, высокочатотные операции перераспределены на машины 1, 4 и другие единицы оборудования. Пространство между операциями на каждой линии значительно сократилось, что указывает на уменьшение межоперационных простоев. Общая длина диаграммы (максимальное время завершения) уменьшилась.

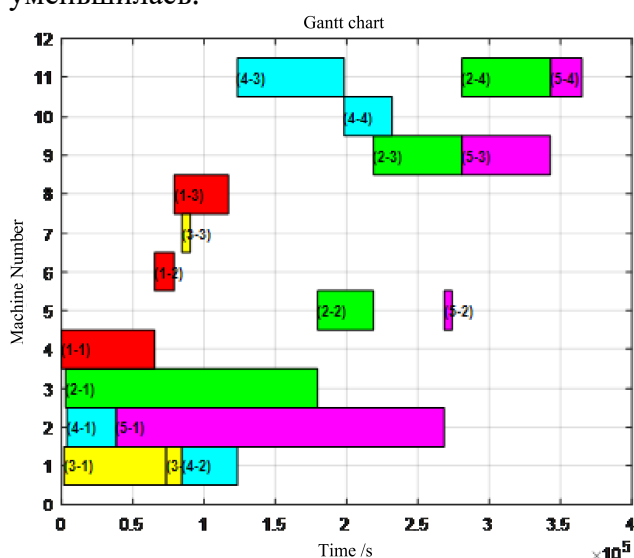


Рис. 2. Диаграмма Ганта до оптимизации

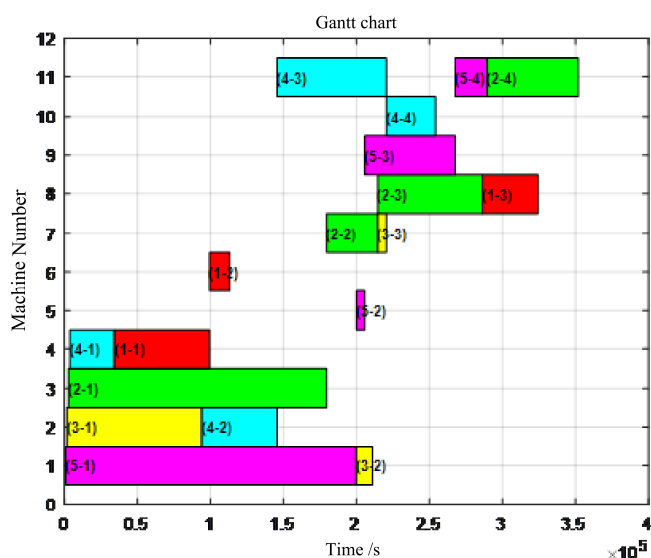


Рис. 3. Диаграмма Ганта после оптимизации

На рисунке (рис. 4) представлены кривые сходимости обоих алгоритмов. Кривая сходимости показывает, как изменяется значение функции приспособленности ( $F$ ) в зависимости от номера поколения. Чем быстрее кривая достигает низкого значения, тем эффективнее работает алгоритм. Предложенный МГА демонстрирует более стабильную и быструю сходимость к лучшему решению, при этом значение функции приспособленности снижается с  $4,85 \cdot 10^9$  до  $3,75 \cdot 10^9$ , причём основное улучшение происходит в первые 40-50 поколений, после чего наступает плавная стабилизация. Преждевременной сходимости не наблюдается: во всех 10 независимых запусках не зафиксировано случаев стагнации в течение 10 поколений, что свидетельствует о его устойчивости и способности продолжать поиск даже на поздних этапах эволюции.

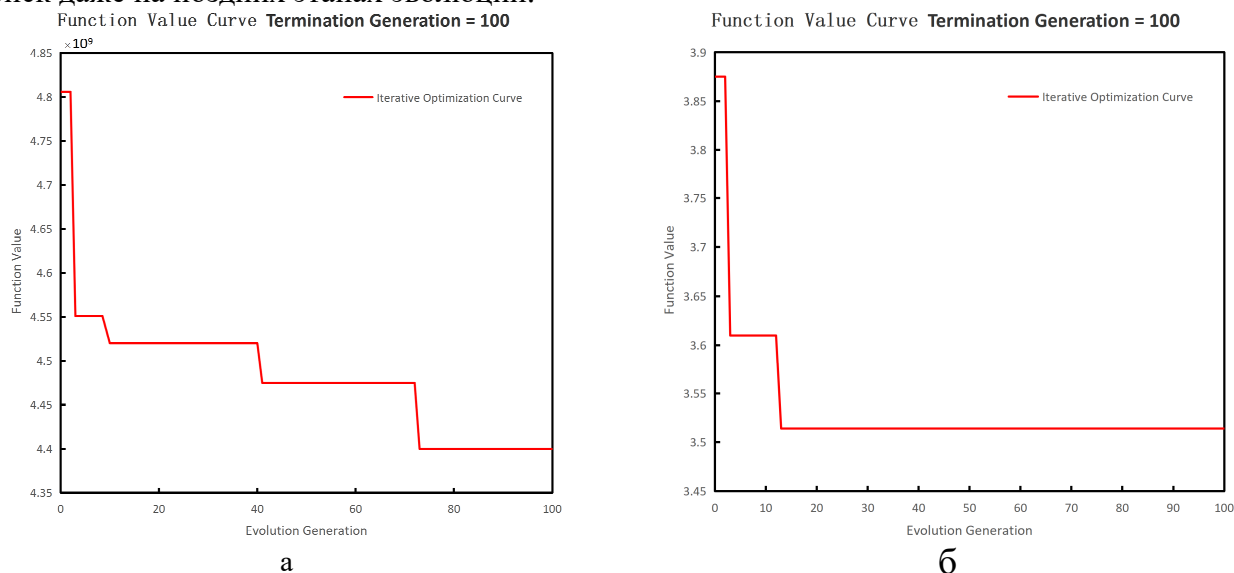


Рис. 4. Кривые сходимости алгоритмов: а – перед оптимизацией, б – после оптимизации

### Заключение

В работе предложен генетический алгоритм с многоуровневым кодированием для решения задачи оперативно-календарного планирования производственных процессов в условиях гибкого производства (FJSSP).

Разработана двухслойная структура хромосомы, включающая слой последовательности операций и слой распределения оборудования. Такая структура обеспечивает совместную эволюцию обеих составляющих решения, что позволяет преодолеть ограничения традиционных подходов, рассматривающих эти задачи по отдельности.

Предложен составной критерий приспособленности, учитывающий три ключевых показателя качества расписания: максимальное время завершения всех работ (продолжительность производственного цикла), равномерность загрузки оборудования и суммарное время простоев. Использование трёх критериев позволяет одновременно улучшать различные аспекты производственного процесса, что особенно важно для современных многономенклатурных производств.

Экспериментально подтверждено, что предложенный алгоритм сокращает максимальное время изготовления на 44,6%, снижает долю простоев оборудования с 31,7 до 9,4%, а стандартное отклонение загрузки оборудования – на 82,0% по сравнению с классическим генетическим алгоритмом. Полученные результаты подтверждаются статистической обработкой данных (10 независимых запусков) и могут считаться достоверными.

Предложенный алгоритм демонстрирует высокую устойчивость к преждевременной сходимости, что подтверждается отсутствием случаев стагнации в течение 10 поколений во всех проведённых экспериментах.

Перспективы дальнейших исследований связаны с адаптацией алгоритма к динамическим условиям производства (учёт отказов оборудования, изменение номенклатуры заказов, срочные заказы), использованием методов параллельных вычислений для повышения производительности, а также внедрением механизмов обучения с подкреплением для автоматической настройки параметров алгоритма.

### **Благодарности и финансирование**

Исследование выполнено при поддержке Фонда естественных наук провинции Хэйлунцзян (проект № LH2023A016) и стартового научного гранта Хэйхэского университета (проект № 2020-KYQDJJ-HX01). Работа также выполнялась в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема «Лаборатория мониторинга и прогнозирования чрезвычайных ситуаций с применением беспилотных авиационных систем», номер государственного учёта ЕГИСУ НИОКТР 1025091500035-5, шифр FEFF-2026-0005).

### **Список литературы**

1-12. См. *References*.

13. Юсупова Н.И., Сметанина О.Н., Ахтариев А.А. Применение эвристических методов для решения задач оперативно-календарного планирования в машиностроении // Технологии нефти и газа. – 2008. – №3(56). – С. 55-61.
14. Сергеев А.И., Гуньков С.А. Проектирование конфигурации оборудования производственного участка на основе генетического алгоритма // Онтология проектирования. – 2026. – Т. 16, №2(60). – С. 314-326. – DOI: 10.18287/2223-9537-2026-16-2-314-326.
15. Трыков А.В. Анализ возможностей применения методов машинного обучения для производственного планирования // Естественно-гуманитарные исследования. – 2024. – №5(55). – С. 637-642.

### **References**

1. Para J., Ser J., Nebro A.J. Solving a Multi-objective Job Shop Scheduling Problem with an Automatically Configured Evolutionary Algorithm // Swarm and Evolutionary Computation. 2023, vol. 79, p. 101295. DOI: 10.1016/j.swevo.2023.101295.
2. Kashef S., Elshaer R. New ant colony algorithm for solving partial job shop scheduling problem // Journal of Advanced Manufacturing Systems. 2023, vol. 22, no. 4, pp. 789-810. DOI: 10.1142/S021968672350028X.
3. Gebreyesus G., Fellek G., Farid A., Fujimura S., Yoshie O. Gated-Attention Model with Reinforcement Learning for Solving Dynamic Job Shop Scheduling Problem // IEEE Transactions on Electrical and Electronic Engineering. 2023, vol. 18, no. 12, pp. 1924-1932. DOI: 10.1002/tee.23915.
4. Tang Fangyan, Gan Shenglin, Lu Nanfang. Optimization of FJSP process routes based on multi-layer coding genetic algorithm // New Technology and New Products of China. 2023, no. 22, pp. 1-4.
5. Wang Yanhong, Zhao Yejian, Liu Wenxin. Application of data mining algorithms in job shop scheduling // Computer Integrated Manufacturing Systems. 2024, vol. 30, no. 2, pp. 520-536. DOI: 10.13196/j.cims.2024.02.008.
6. Wang Yuqin. Flexible shop scheduling algorithm based on hybrid particle swarm and ant colony algorithm // Electronic Design Engineering. 2023, vol. 31, no. 17, pp. 65-69. DOI: 10.14022/j.issn1674-6236.2023.17.014.
7. Zhang Congyi, Li Xinyu, Xia Kai, Zhang Chunjiang, Wang Sihan. Assembly job shop scheduling algorithm based on right-shift repair coding method // Computer Integrated Manufacturing Systems. 2024, vol. 30, no. 10, pp. 3514-3524. DOI: 10.13196/j.cims.2024.10.012.
8. Yang Yuanyuan, Hu Rong, Qian Bin, Zhang Changsheng, Jin Huaiping. Deep reinforcement learning algorithm for solving dynamic flow shop real-time scheduling // Control Theory and Applications. 2024, vol. 41, no. 6, pp. 1047-1055. DOI: 10.7641/CTA.2023.20928.
9. Kong Wenyuan, Xu Zhijie, Tang Bin. Flow shop scheduling algorithm and program implementation on two machines // Computer Programming and Application. 2024, vol. 15, no. 9, pp. 33-35.
10. Zheng Mingliang, Wang Taiyong, Tian Songling, Zhang Lei, Ju Chengjie, Xiong Xi. Research on adaptive scheduling method based on digital twin workshop // Proceedings of the Conference on Intelligent Manufacturing and Mechanical Dynamics. 2023.
11. Chen Xuefeng, Zuo Yang. Research on distributed hybrid flow shop scheduling algorithm based on learning mechanisms // Journal of Henan Institute of Technology. 2024, vol. 32, no. 4, pp. 25-31. DOI: 10.3969/j.issn.2095-9734.2024.04.005.
12. Li Changyun, Li Tingyu, Wang Zhibing, Gu Pengfei, Lin Duo. Improved genetic algorithm based on multi-objective optimization for solving flexible job shop scheduling problem // Manufacturing Technology and Machine Tools. 2023, no. 5, pp. 173-178. DOI: 10.19287/j.mtmt.1005-2402.2023.05.025.
13. Yusupova N.I., Smetanina O.N., Akhtariev A.A. Application of heuristic methods for solving operational scheduling problems in mechanical engineering // Oil and Gas Technologies. 2008, no. 3(56), pp. 55-61.

14. Sergeev A.I., Gun'kov S.A. Designing equipment configuration for a production site based on a genetic algorithm // Ontology of Designing. 2026, vol. 16, no. 2(60), pp. 314-326. DOI: 10.18287/2223-9537-2026-16-2-314-326.
15. Trykov A.V. Analysis of the possibilities of applying machine learning methods for production planning // Natural and Humanitarian Research. 2024, no. 5(55), pp. 637-642.

*Сведения об авторах:*

*Information about authors:*

|                                                                                                            |                                                                                                                   |
|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Чжан Цзин</b> – преподаватель                                                                           | <b>Zhang Jing</b> – lecturer                                                                                      |
| <b>Чжао Янь</b> – преподаватель                                                                            | <b>Zhao Yan</b> – lecturer                                                                                        |
| <b>Купцова Олеся Витальевна</b> – кандидат технических наук, доцент кафедры безопасности жизнедеятельности | <b>Olesya V. Kuptsova</b> – candidate of technical sciences, associate professor of the Department of life safety |
| <b>Моу Яньнань</b> – преподаватель                                                                         | <b>Mou Yannan</b> – lecturer                                                                                      |
| Korsuncevaolesy@gmail.com                                                                                  |                                                                                                                   |

Получена 25.06.2026