

Л.А. Гладков, Н.В. Гладкова, С.А. Громов

ГИБРИДНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ ОПЕРАТИВНОГО ПЛАНИРОВАНИЯ ПРОИЗВОДСТВЕННОГО ПРОЦЕССА*

Рассматривается новый подход к решению задач оперативного планирования производственного процесса. Приведено определение задачи оперативного планирования производства. Показано место задач оперативного планирования в общей теории расписаний. Приведена постановка задачи оперативного планирования для предприятий машиностроения, отмечены особенности решения подобного рода задач на предприятиях машиностроения, заданы ограничения и целевая функция рассматриваемой задачи оптимизации. Приведено описание структуры предложенного гибридного алгоритма оперативного планирования производственного процесса. Разработана методика кодирования решений для использования их в гибридном алгоритме. Описана структура хромосомы, представляющая собой множество заданий за период планирования или оперативный план в целом. Предложены новые модификации генетических операторов, направленные на повышение разнообразия текущей популяции и преодоление локальных оптимумов. Приведены примеры выполнения модифицированных генетических операторов, учитывающих специфику решения задач оперативного планирования. Описана предложенная методика и последовательность расчета значения критерия оценки качества получаемых решений. Введены дополнительные ограничения на область допустимых решений задачи и предложена методика расчета штрафов за нарушение установленных ограничений. Приведена последовательность выполнения операций и структура разработанного алгоритма. Предложена гибридная модель алгоритма на основе интеграции методов генетического поиска и принципов нечеткого управления. Проведены исследования характеристик разработанного алгоритма и определена его временная сложность. Проведены серии вычислительных экспериментов для анализа и сравнения качества получаемых решений, а также определения лучших значений управляющих параметров алгоритма. На основании анализа сделаны выводы о достоинствах и недостатках предложенного алгоритма.

Задачи оперативного планирования производства; генетический алгоритм; нечеткая логика; теория расписаний; оптимизация; гибридный алгоритм.

L.A. Gladkov, N.V. Gladkova, S.A. Gromov

HYBRID ALGORITHM OF SOLVING THE PROBLEMS OF OPERATIONAL PLANNING OF THE PRODUCTION PROCESS

The article considers a new approach to solving the problems of operational planning of the production process. The definition of the task of operational planning of production is given. The place of the tasks of operational planning in the general theory of schedules is shown. The formulation of the task of operational planning for machine-building enterprises is given, specifics of the solution of such tasks at machine-building enterprises are noted, constraints and the objective function of the optimization problem are given. The structure of the proposed hybrid algorithm for operational planning of the production process is described. A method for coding solutions for using them in a hybrid algorithm is developed. The chromosome structure is described, which is a set of tasks for the planning period or the operational plan as a whole. New modifications of genetic operators aimed at increasing the diversity of the current population and overcoming local optima are proposed. Examples of implementation of modified genetic operators that take into account the specifics of solving operational planning problems are given. The proposed methodology and the sequence of calculation of the value of the criterion for assessing the quality of the solutions obtained are described. Additional restrictions were introduced on the area of feasible

* Работа выполнена при финансовой поддержке РФФИ (проект № 16-01-00715).

solutions to the problem and a methodology for calculating fines for violating established restrictions was proposed. The sequence of operations and the structure of the developed algorithm are given. A hybrid model of the algorithm based on integration of methods of genetic search and principles of fuzzy control is proposed. The characteristics of the developed algorithm are studied and its time complexity is determined. A series of computational experiments was conducted to analyze and compare the quality of the solutions obtained, as well as to determine the best values of the control parameters of the algorithm. Based on the analysis, conclusions are drawn about the merits and demerits of the proposed algorithm.

Operational production planning problems; genetic algorithm; fuzzy logic; scheduling theory; optimization; hybrid algorithm.

Введение. Согласно определению международного консорциума по производственным системам Manufacturing Enterprise Solutions Association International (MESA) оперативное (детальное) планирование понимается как процесс составления и расчета производственных расписаний, основанный на приоритетах, атрибутах, характеристиках и способах, связанных со спецификой изделий и технологией производства.

Таким образом, оперативное планирование сводится к задачам теории расписаний (ТР) [1–6], в рамках которых необходимо:

- ◆ назначить исполнителя для каждого задания;
- ◆ упорядочить задания для каждого исполнителя, т.е. найти наилучшую, в плане достижения поставленной цели последовательность их выполнения.

В теории расписаний используются следующие допущения.

1. Подлежащие выполнению работы определены и известны полностью. Предполагается, что все заданные работы должны быть выполнены (разбиение совокупности работ на классы выполняемых и невыполняемых не входит в задачу упорядочения).

2. Однозначно определены устройства, выделяемые для выполнения заданных работ.

3. Задана совокупность всех элементарных действий, связанных с выполнением каждой из работ, и ограничений на порядок их выполнения. Известно также, каким образом осуществляются эти действия, и что существует, по крайней мере, по одному устройству, способному выполнить каждое из них.

Постановка задачи. Для предприятий машиностроения имеет место следующая задача оперативного планирования производства, сочетающая в себе различные классы задач ТР.

Известно множество машин (производственных линий) $\{M\}$, $|M| = m$, причем каждая линия характеризуется определенным перечнем параметров, значение которых накладывает дополнительные ограничения при назначении работ. График недоступности производственных линий определяет периоды сервисов или ремонтов.

- ◆ Начало i -го сервиса на j -й линии sb_{ji} ;
- ◆ Окончание i -го сервиса на j -й линии se_{ji} ;
- ◆ Общее количество оснастки RQ ;
- ◆ Матрица времени переналадки линии WT , требуемого при переходе с выпуска i -го задания на j -е, каждый элемент матрицы $wt_{ij} \geq 0$;

Конечное множество заданий $\{N\}$, $|N| = n$, где каждое задание i состоит из одной операции. Задание есть элементарная задача, подлежащая выполнению, оно характеризуется:

- ◆ номером машины m_i , на которую назначено i -е задание, $1 < m_i < m$;
- ◆ длительностью задания t_i ;
- ◆ индивидуальным директивным сроком d_i i -го задания;
- ◆ признаком необходимости использовать оснастку $rq_i \in \{0, 1\}$;

- ♦ матрицей инцидентности заданий и производственных линий R , элемент матрицы r_{ij} принимает значение $r_{ij} \geq 0$, если i -е задание выполняется на j -й линии, и $r_{ij} = \infty$, если i -е задание не выполняется на j -й линии.

Необходимо найти такое разбиение множества заданий N на m непересекающихся подмножеств, что:

1. Распределение заданий по линиям соответствует матрице инцидентности R , $\forall m \in M, i \in N_m \Rightarrow r_{im} > 0$;
2. Для заданий осуществляется выбор линий с наименьшим значением приоритета:

$$\sum_{i \in N} r_{im} \rightarrow \min.$$

3. Для каждого подмножества N_m на горизонте планирования D существует расписание (упорядочение) $\sigma_m: Nm \rightarrow \{0, 1, \dots, D\}$ такое, что:

а) последовательности выполнения заданий на одной линии не повторяются,

$$\forall n_{i+1} \in N_m \setminus n_i \Rightarrow \sigma_m(n_{i+1}) \neq \sigma_m(n_i);$$

б) не нарушается график доступности m -й линии,

$$\forall n_i \in N_m \Rightarrow \begin{cases} \sigma_m(n_i) \notin [sb_{ml}; se_{ml}] \\ \sigma_m(n_i) + t_i \notin [sb_{ml}; se_{ml}] \end{cases};$$

в) не превышает количество загруженных линий одновременно,

$$\forall i \in \{0, 1, \dots, D\}, |\{n_i \in N : \sum_{m=1}^{|M|} \sigma_m(n_i)/i\}| \leq m_{\max}, m_{\max} \leq m;$$

г) выполняются условия назначения переналадок линий,

$$\forall n_{i+1} \in N_m \setminus n_i, \sigma_m(n_i) < \sigma_m(n_{i+1}) \Rightarrow \sigma_m(n_{i+1}) - \sigma_m(n_i) - t_i \geq wt_{n_{i+1}};$$

е) выполняются ограничения по одновременному использованию оснастки,

$$\forall i \in \{0, 1, \dots, D\}: \{\forall n_j \in N:$$

$$[\sigma_m(n_j); \sigma_m(n_j) + t_j] \subset i \wedge r_{qj} = 1\} \leq RQ.$$

Общим критерием для построения итогового расписания является минимизация целевой функции $F \rightarrow \min$, где F понимается как функция штрафа, отражающая суммарное отклонение заданий от индивидуальных директивных сроков:

$$F = \sum_i^N |\sigma(n_i) + t_i - d_i| \rightarrow \min.$$

Описание алгоритма. В данной задаче используем схему кодирования, при которой каждая хромосома представляет собой искомое решение целиком, т.е. один агент (особь) несет в себе закодированную информацию обо всем плане на период планирования. Недостатком данной схемы является большая длина хромосомы. Но при этом, целевая функция (ЦФ) каждой особи отражает общий критерий оптимизации, а, во-вторых, в каждом поколении имеется некоторое множество решений, которое в совокупности с генетическими операторами должно обеспечить более быструю сходимость за счет более разнообразия исходного «генетического материала». Таким образом, структурно хромосома будет представлять собой множество заданий за период планирования или оперативный план в целом. Хромосома содержит число генов: $N_h = Nr_{\max} A_r$, где Nr_{\max} – максимально возможное число заданий за период планирования; A_r – количество варьируемых атрибутов задания. Значение $Nr_{\max} = M$ определяется общим количеством заданий, сформированных по строкам основного графика производства.

Итак, в полученной хромосоме имеется M групп генов, каждая из которых полностью определяет соответствующее задание, т.е. первой группе соответствует первое задание, второй – второе, и т.д. Значение гена определяет значение соответствующего атрибута.

Для атрибутов «длительность задания» значением гена является количество часов, на которое задание загружает выбранную линию.

Для атрибутов «номер производственной линии» значением гена является порядковый номер ненулевого элемента в соответствующей строке матрицы инцидентности продуктов и производственных линий R .

Для атрибутов «очередность задания» значение гена определяет номер часа, в который начинается производственное задание.

Следует отметить, что идентификатор продукта для заданий отдельно не кодируется, они определяются жестко, т.е. каждое задание четко связано с определенным продуктом. Такой способ кодирования, при котором полезную информацию несут в себе не только значения генов, но и их позиции в хромосоме, минимизирует длину хромосомы, что, в свою очередь, влечет за собой уменьшение пространства поиска. Следствием этого является уменьшение времени сходимости (в смысле числа поколений, которое нужно обработать для достижения сходимости) и времени обработки одного поколения. Также еще хотелось бы отметить некоторые особенности данной схемы представления особи:

- ◆ нулевое значение длительности задания или даты начала задания интерпретируется как отсутствие задания – таким образом, в процессе эволюции ГА имеет возможность сойтись к оптимальному количеству заявок, при условии, что $M > \text{Mort}$;
- ◆ величины M являются искусственными ограничениями, накладываемыми на пространство поиска в данном представлении решения.

Непосредственно кодировать хромосому предлагается в двоичном виде (рис. 1).

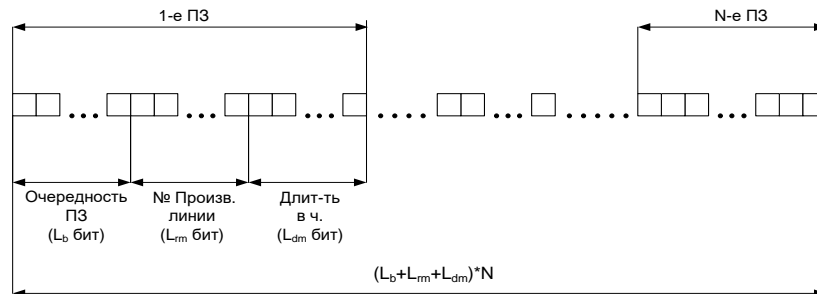


Рис. 1. Схема кодирования хромосомы

Длина хромосомы есть:

$$L_h = (L_b + L_{rm} + L_{dm})N,$$

где L_b – число бит, необходимое для кодирования любого момента начала задания в течение периода планирования с выбранной точностью (для наших целей с точностью до часа);

L_{rm} – число бит, необходимое для кодирования альтернативной линии, для продукта m -го задания. Значение этого гена определяет порядковой номер ненулевого элемента в соответствующей строке матрицы инцидентности R ;

L_{dm} – число бит, необходимое для кодирования длительности m -го задания (для наших целей с часовой кратностью).

Такая схема кодирования представляется достаточно гибкой, так как в процессе работы алгоритма могут варьироваться даты начала, номера линий и длительности заданий.

Как правило, специфика решаемой задачи закладывается в штрафные функции, используемые при расчете ЦФ особи. Помимо использования штрафных функций предлагается модифицировать логику базовых операторов генетического поиска. Идея модификации сводится к применению правил, которыми руководствуются предметные эксперты в процессе построения расписания. Суть этих правил состоит в направленной корректировке отдельных параметров производственных заданий, с целью разрешения коллизий, возникающих при нарушении ограничений, обусловленных спецификой задачи. Модификация предполагает изменить базовые операторы мутации и кроссинговера.

В частности для оператора кроссинговера применяется специализированное правило выбора аллелей при реализации скрещивания хромосом. Для визуализации модифицированной логики представлен упрощенный пример, в рамках которого рассматриваются 5 работ, 3 альтернативные производственные линии. Выбраны 2 хромосомы, представляющие собой различные решения, а именно варианты назначения и последовательность исполнения работ производственными линиями. Каждый прямоугольник помечен индексом соответствующей работы. Прямоугольники расположены по горизонтали вдоль линий, определяющих ту или иную производственную линию. Таким образом, получаем вариацию диаграммы Ганта, зачастую используемую для визуализации расписаний (рис. 2).



Рис. 2. Варианты производственного расписания, построенные с помощью модифицированного оператора кроссинговера

Для каждой производственной линии выбираем работы, по одной из каждой особи. Для каждой пары случайным образом определяем будущую принадлежность соответствующему потомку (рис. 3). При этом выполняется проверка на дублирование работ в одном и том же решении. На рис. 4 представлен пример перераспределения работ в процессе скрещивания.

Итог выполнения оператора кроссинговера представлен на рис. 4. Получены новые решения, при этом исключено образование заведомо некорректных или нарушающих ограничения очередности заданий на линиях решений.

Линия 1	Родитель 1	Родитель 2		Потомок 1	Потомок 2
Пара 1	3	6	->	6	3
Пара 2	4	1	->	4	1
Пара 3	NULL	2	->	NULL	2
Линия 2	Родитель 1	Родитель 2		Потомок 1	Потомок 2
Пара 1	1	4	->	1	4
Пара 2	NULL	5	->	5	NULL
Линия 3	Родитель 1	Родитель 2		Потомок 1	Потомок 2
Пара 1	5	3	->	3	5
Пара 2	2	NULL	->	2	NULL
Пара 3	6	NULL	->	NULL	6

Рис. 3. Перераспределение работ в результате использования модифицированного оператора кроссинговера



Рис. 4. Результат перераспределения работ при использовании модифицированного кроссинговера

Таким образом, в логику базового генетического оператора добавляется фактор предопределенности, исключающий потенциальные решения, т.е. потомков, которые заведомо не удовлетворяют ограничениям.

- ◆ Оператор мутации. В нашей задаче будем использовать многоточечный ОМ. Здесь в каждой хромосоме случайным образом выбирается N_{OM} пар различных генов, между которыми и будет происходить обмен значениями. Значение N_{OM} определяется из выражения: $N_{OM} = \alpha_{OM}L$, где $\alpha_{OM} \in (0; 0,5)$ – параметр, определяющий долю пар генов, участвующих в мутации, от общей длины хромосомы.

Полученное значение N_{OM} округляется до ближайшего большего целого. Значение α_{OM} рекомендуется выбирать из диапазона: $[0,01; 0,03]$.

- ◆ Оператор миграции, используется для того, чтобы не допустить преждевременной сходимости алгоритма (попадание в локальный оптимум). Предполагается наличие двух параллельно эволюционирующих популяций:

$$PR^1 = \{pr_1^1, \dots, pr_i^1, \dots, pr_{N_{pr}}^1\} \text{ и } PR^2 = \{pr_1^2, \dots, pr_i^2, \dots, pr_{N_{pr}}^2\}.$$

На определенном этапе определенное количество особей из первой популяции N_M переходит во вторую популяцию, а из второй популяции такое же количество особей переходит в первую популяцию. Таким образом, размеры обеих популяций не изменяются. Отбор особей из каждой популяции осуществляется по принципу «либо лучшие, либо худшие». В этом случае из одной популяции отбирается N_M особей с наилучшим значением ЦФ, а из другой – с наихудшим. Количество особей, участ-

вующих в миграции задается величиной N_M , значение которой можно определить из следующего соотношения: $N_M = \alpha_M N_{pr}$, где $\alpha_M \in (0; 1)$ – параметр, определяющий долю агентов (особей), участвующих в миграции хромосомы от общего размера популяции. Полученное значение N_M округляется до ближайшего большего целого. Значение α_M рекомендуется выбирать из диапазона: $[0,1; 0,3]$.

Расчет целевой функции состоит из следующих частей:

- ◆ расчет критерия F ;
- ◆ расчет корректирующих поправок, учитывающих ограничения;
- ◆ расчет ЦФ на основе рассчитанных значений F и корректирующих поправок.

Критерий F определяется функцией количества заданий, нарушающих директивные требования за весь период планирования.

Смысл поправок применительно к ГА, связанный с наложением некоторого «штрафа» на особь, предполагает понижение значения ее ЦФ в случае, если решение, которое представляет собой конкретная особь, выходит за границы области допустимых решений. Важно отметить, что ограничения, задающие ОДР, относятся либо к конкретному производственному заданию, либо ко всему расписанию в целом. Условно их можно разделить на две группы: штрафы по заданиям и штрафы по всему расписанию. Ограничения, которые относятся к группе штрафов по всему расписанию:

- ◆ ограничения использования технологической оснастки;
- ◆ ограничение количества работающих линий в сутки;
- ◆ жесткие ограничения по использованию линий с наименьшим значением приоритета.

Штрафы за нарушение этих ограничений рассчитывается явным образом в виде поправки на значение критерия. Для расчета поправки используются соответствующие коэффициенты:

Коэффициент превышения использования технологической оснастки:

$$\alpha_{st} = \frac{\sum_{t=1}^T \frac{bs(t) - bs_{lim} + |bs(t) - bs_{lim}|}{2}}{bs_{lim}},$$

где $bs(t)$ – количество используемой оснастки в t -й момент; bs_{lim} – общее количество технологической оснастки RQ ; T – горизонт планирования.

Коэффициент превышения количества работающих линий в сутки:

$$\alpha_{bq} = \frac{\sum_{t=1}^T \frac{bq(t) - bq_{lim} + |bq(t) - bq_{lim}|}{2}}{bq_{lim}},$$

где $bq(t)$ – количество используемой оснастки в t -й момент; bq_{lim} – предельное количество одновременно работающих линий m_{max} ; T – горизонт планирования.

Коэффициент, учитывающий использование линий с наименьшим значением приоритета $\alpha_{M_r} = \sum_{t=0}^T \frac{M_r(t)}{M(t)}$, где $M_r(t)$ – количество заданий в t -й день, кото-

рые назначены на производственную линию с приоритетом, отличным от наименьшего приоритета среди альтернативных линий соответствующего задания согласно матрице инцидентности R . $M(t)$ – общее количество заданий в t -й день с ненулевой длительностью.

Соответственно рассчитываем поправки по трем составляющим:

$$\Delta_{st} = \tilde{F} * \alpha_{st}; \quad \Delta_{bq} = \tilde{F} * \alpha_{bq}; \quad \Delta_{M_r} = \tilde{F} * (1 - \alpha_{M_r}).$$

Таким образом, значение $\Phi = \tilde{F} - \Delta_{st} - \Delta_{bq} - \Delta_{M_r}$ и представляет собой то значение ЦФ, которое используется при работе операторов ГА.

Следующим этапом работы ГА является проверка на выполнение условий окончания работы алгоритма. В качестве таких условий могут выступать:

- ◆ достижение заданного количества поколений работы алгоритма;
- ◆ достижение заданного времени работы алгоритма;
- ◆ сохранение относительного изменения среднего значения ЦФ по популяции в течение определенного количества поколений заданного значения.

Также общим условием для всех вышеприведенных является получение решения, которое удовлетворяет ограничениям, имеющимся в постановке задачи оптимизации. Из этих трех основных критериев наиболее целесообразно использовать последний, так как в отсутствие информации об оптимальном значении он наиболее адекватно отражает факт сходимости алгоритма.

Гибридная модель поиска решений. В последние годы большой интерес вызывают интегрированные и гибридные модели, включающие генетические алгоритмы. Здесь можно выделить подходы «внешней» гибридизации, например, построение гибридов генетических алгоритмов и эволюционной стратегии и нейросетевых метамоделей [7–11] и «внутренней» гибридизации, когда в контексте эволюционного проектирования объединяются алгоритмы на базе различных эволюционных моделей, например, модели Дарвина и модели Ламарка [11–14].

В работе предлагается вариант «внешней» гибридизации. Здесь гибридный нечеткий генетический алгоритм сочетает в подходы нечеткой логики и генетического поиска в рамках единого оптимизационного процесса. Структурная схема этого алгоритма представлена на рис. 5.

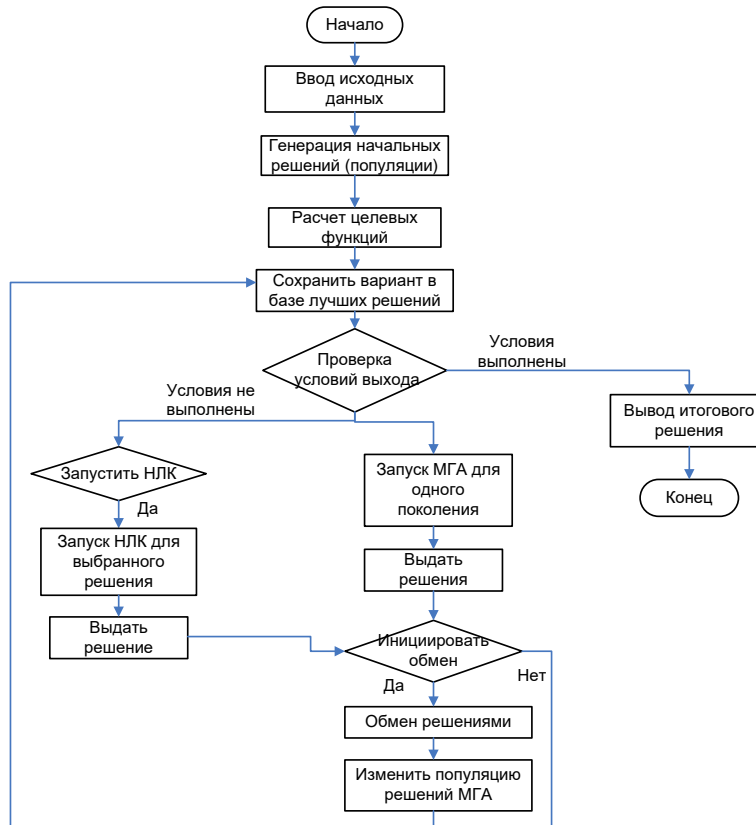


Рис. 5. Структурная схема гибридного алгоритма

Основная идея гибридизации заключается в возможности использования математического аппарата теории нечетких систем для кодирования, подбора оптимальных параметров генетических алгоритмов, значений вероятности генетических операторов, выбора функции пригодности и критерия останова, создания нечетких генетических операторов. Предлагаемый алгоритм можно применять в условиях ведения параллельных вычислений на соответствующих вычислительных ресурсах. Справедливо отметить, что архитектура современных процессоров, даже для персональных компьютеров, является многоядерной, что в свою очередь предоставляет возможность ведения параллельных и распределенных вычислений [15–18]. Таким образом, гибридный алгоритм возможен вариант, при котором осуществляется одновременный расчет решений в НГА, а также обмен полученными решениями в рамках одной расчетной итерации гибридного алгоритма.

Для повышения качества результатов генетического поиска решается задача включения в контур эволюции экспертной информации путем построения нечеткого контроллера, регулирующего значения параметров процесса эволюции.

В качестве входных используются следующие параметры [19]:

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)}; \quad e_2(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{worst}(t) - f_{best}(t)};$$

$$e_3(t) = \frac{f_{best}(t) - f_{best}(t-1)}{f_{best}(t)}; \quad e_4(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{ave}(t)},$$

где t – временной шаг, $f_{best}(t)$ – лучшее значение ЦФ на итерации t , $f_{best}(t-1)$ – лучшее значение ЦФ на итерации $(t-1)$, $f_{worst}(t)$ – худшее значение, ЦФ на итерации t , $f_{ave}(t)$ – среднее значение ЦФ на итерации t , $f_{ave}(t-1)$ – среднее значение ЦФ на итерации $(t-1)$ [20].

На выходе получаем вероятности применения операторов кроссинговера, мутации, а также оператора миграции.

Вычислительные эксперименты. В ходе проведения вычислительных экспериментов стояла задача определения и анализа зависимости скорости сходимости и качества решений разработанных алгоритмов для различных случаев их использования.

Необходимо установить характер зависимости размерности задачи (значения количества работ N , количества линий M), и временных затрат на поиск решения. Под сходимостью понимается достижение такого значения целевой функции D , при котором за последующие δ циклов работы алгоритма изменение ΔD составит не более $\Delta\%$ от предыдущего значения.

$$\frac{|D_{i-\delta} - D_i|}{D_{i-\delta}} * 100\% \leq \Delta,$$

где $D_{i-\delta}$ – значение целевой функции на $(i-\delta)$ -м шаге работы алгоритма; D_i – значение целевой функции на i -м шаге работы алгоритма; δ – количество шагов работы, на котором рассчитывается относительное изменение целевой функции; i – текущий шаг работы алгоритма; Δ – пороговое значение относительного изменения целевой функции.

Для проводимых в данной работе исследований в качестве значений параметров критерия сходимости были взяты следующие величины: $\delta = 10$; $\Delta = 1\%$.

Исследование проводилось по двенадцати точкам, каждая точка эксперимента, для которых строится производственное расписание, определяется вектором исходных данных, координатами которого являются: общее количество заданий N ,

общее количество линий M , количество спаренных линий $LD = \sum_{j_1=1}^M \sum_{j_2=(M-j_1-1)}^M ld_{j_1 j_2}$,

горизонт планирования D в часах, количество оснастки RQ.

Ниже представлена таблица исходных данных (табл. 1).

Для всех точек эксперимента приняты общие исходные данные:

- ◆ длительности работ выбраны произвольно из заданного интервала $t_i \in [t_{min}, t_{max}]$;
- ◆ учитывается один сервисный интервал, заданный для спаренной линии.

Таблица 1

Данные для оценки сходимости алгоритмов

№	N	M/LD	D	RQ
1	5	3/2	72	1
2	10	3/2	96	1
3	20	6/2	96	2
4	50	12/4	120	3
5	70	16/6	120	3
6	100	24/8	120	4
7	150	26/8	144	5
8	200	30/10	168	6
9	250	34/10	192	6
10	300	40/12	240	7
11	400	50/16	240	8
12	500	54/18	240	8

Заключение. Как видно из графиков, время сходимости нарастает линейно с увеличением размерности задачи. Также следует отметить более быструю сходимость адаптивного поиска по отношению к модифицированному генетическому алгоритму. Полученные результаты подтвердили общее предположение о целесообразности использования приближенного алгоритма для получения приемлемого (квазиоптимального) решения. Характер зависимостей отражает почти линейную зависимость времени счета от размерности задачи, таким образом можно предполагать полиномиальную временную сложность предложенных алгоритмов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Задачи и алгоритмы. – М.: Изд-во МГУ, 2011. – 224 с.
2. Conway R.M., Maxwell W.L., Miller L.W. Theory of Scheduling. – 2nd ed. – Dover Publications, Mineola NY, 2004.
3. Высочин С.В. Принципы построения систем для расчета производственных расписаний // САПР и графика. – 2008. – № 9. – С. 57-59.
4. Pinedo M. Scheduling: Theory, Algorithms and Systems. – 3rd ed. – Springer Verlag, New York, 2008.
5. Leung J.Y.T. Handbook of Scheduling // Boca Raton, Florida: Chapman & Hall/CRC, 2004.
6. Танаев В.С. Введение в теорию расписаний. – М.: Наука, 1975. – 256 с.
7. Michael A., Takagi H. Dynamic control of genetic algorithms using fuzzy logic techniques // Proceedings of the Fifth International Conference on Genetic Algorithms. Morgan Kaufmann. – 1993. – P. 76-83.
8. Lee M.A., Takagi H. Integrating design stages of fuzzy systems using genetic algorithms // Proceedings of the 2nd IEEE International Conference on Fuzzy System. – 1993. – P. 612-617
9. Herrera F., Lozano M. Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions // J. Soft Computing. – Springer-Verlag, 2003. – P. 545-562

10. Kennedy J., Eberhart R.C. Particle swarm optimization // Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995. – P. 1942-1948.
11. Hayder M., Tony H., Naz E.I., Hybrid Algorithm for the Optimization of Training Convolutional Neural Network // International Journal of Advanced Computer Science and Applications. – 2015. – Vol. 6, No. 10. – P. 79-85.
12. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2010.
13. Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов П.В. Биоинспирированные методы в оптимизации. – М.: Физматлит, 2009. – 384 с.
14. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003.
15. Кныш Д.С., Курейчик В.М. Параллельные генетические алгоритмы: Проблемы, обзор и состояние // Известия РАН. Теория и системы управления. – 2010. – № 4. – С. 72-82.
16. Гладков Л.А., Гладкова Н.В. Особенности использования нечетких генетических алгоритмов для решения задач оптимизации и управления // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 130-136.
17. King R.T.F.A., Radha B., Rughooputh H.C.S. A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration // Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, 2004. – P. 577-582
18. Zhongyang X., Zhang Y., Zhang L., Niu S. A parallel classification algorithm based on hybrid genetic algorithm // Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, 2006. – P. 3237-3240
19. Gladkov L., Gladkova N., Leiba S. Manufacturing scheduling problem based on fuzzy genetic algorithm // In: Proceeding of IEEE East-West Design & Test Symposium – (EWDTS'2014). – Kiev, Ukraine, 2014. – P. 209-212.
20. Gladkov L.A., Gladkova N.V., Leiba S.N. Manufacturing Scheduling Problem Based on Fuzzy Genetic Algorithm. // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2014). Kiev, Ukraine, September 26–29, 2014. – P. 209-213.

REFERENCES

1. Lazarev A.A., Gafarov E.R. Teoriya raspisaniy. Zadachi i algoritmy [The scheduling theory. Problems and algorithms]. Moscow: Izd-vo MGU, 2011, 224 p.
2. Conway R.M., Maxwell W.L., Miller L.W. Theory of Scheduling. 2nd ed. Dover Publications, Mineola NY, 2004.
3. Vysochin S.V. Printsipy postroeniya sistem dlya rascheta proizvodstvennykh raspisaniy [Principles of building systems to calculate production schedules], SAPR i grafika [CAD and graphics], 2008, No. 9, pp. 57-59.
4. Pinedo M. Scheduling: Theory, Algorithms and Systems. 3rd ed. Springer Verlag, New York, 2008.
5. Leung J.Y.T. Handbook of Scheduling, Boca Raton, Florida: Chapman & Hall/CRC, 2004.
6. Tanaev V.S. Vvedenie v teoriyu raspisaniy [Introduction to the theory of schedules]. Moscow: Nauka, 1975, 256 p.
7. Michael A., Takagi H. Dynamic control of genetic algorithms using fuzzy logic techniques, Proceedings of the Fifth International Conference on Genetic Algorithms. Morgan Kaufmann, 1993, pp. 76-83.
8. Lee M.A., Takagi H. Integrating design stages of fuzzy systems using genetic algorithms, Proceedings of the 2nd IEEE International Conference on Fuzzy System, 1993, pp. 612-617
9. Herrera F., Lozano M. Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions, J. Soft Computing. Springer-Verlag, 2003, pp. 545-562
10. Kennedy J., Eberhart R.C. Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942-1948.
11. Hayder M., Tony H., Naz E.I., Hybrid Algorithm for the Optimization of Training Convolutional Neural Network, International Journal of Advanced Computer Science and Applications, 2015, Vol. 6, No. 10, pp. 79-85.
12. Gladkov L.A., Kureychik V.V., Kureychik V.M. Geneticheskie algoritmy [Genetic algorithms]. Moscow: Fizmatlit, 2010.

13. Gladkov L.A., Kureychik V.V., Kureychik V.M., Sorokoletov P.V. Bioinspirovannyye metody v optimizatsii [Bioinspired methods in optimization]. Moscow: Fizmatlit, 2009, 384 p.
14. Emel'yanov V.V., Kureychik V.V., Kureychik V.M. Teoriya i praktika evolyutsionnogo modelirovaniya [Theory and practice of evolutionary modeling]. Moscow: Fizmatlit, 2003.
15. Knysh D.S., Kureychik V.M. Parallelnyye geneticheskiye algoritmy: Problemy, obzor i sostoyaniye [Parallel genetic algorithms: Problem overview and state], *Izvestiya RAN. Teoriya i sistemy upravleniya* [Journal of Computer and Systems Sciences International], 2010, No. 4, pp. 72-82.
16. Gladkov L.A., Gladkova N.V. Osobennosti ispol'zovaniya nechetkikh geneticheskikh algoritmov dlya resheniya zadach optimizatsii i upravleniya [Features of use of fuzzy genetic algorithms for the decision of problems of optimisation and control], *Izvestiya YuFU. Tekhnicheskiye nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 4 (93), pp. 130-136.
17. King R.T.F.A., Radha B., Rughooputh H.C.S. A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration, *Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, 2004*, pp. 577-582
18. Zhongyang X., Zhang Y., Zhang L., Niu S. A parallel classification algorithm based on hybrid genetic algorithm, *Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, 2006*, pp. 3237-3240
19. Gladkov L., Gladkova N., Leiba S. Manufacturing scheduling problem based on fuzzy genetic algorithm, *In: Proceeding of IEEE East-West Design & Test Symposium – (EWDTS'2014). Kiev, Ukraine, 2014*, pp. 209-212.
20. Gladkov L.A., Gladkova N.V., Leiba S.N. Manufacturing Scheduling Problem Based on Fuzzy Genetic Algorithm, *Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2014). Kiev, Ukraine, September 26–29, 2014*, pp. 209-213.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Гатчин.

Гладков Леонид Анатольевич – Южный федеральный университет; e-mail: leo@tgn.sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371625; кафедра систем автоматизированного проектирования; доцент.

Гладкова Надежда Викторовна – e-mail: leo_gladkov@mail.ru; тел.: 88634393260; кафедра систем автоматизированного проектирования; старший преподаватель.

Громов Сергей Алексеевич – Московский государственный технический университет им. Баумана; e-mail: sergey.a.gromov@gmail.com; г. Москва, 2-я Бауманская ул., д. 5, стр. 1; аспирант.

Gladkov Leonid Anatol'evich – Southern Federal University; e-mail: leo@tgn.sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371625; the department of computer aided design; associate professor.

Gladkova Nadezhda Viktorovna – e-mail: leo_gladkov@mail.ru; phone: +78634393260; the department of computer aided design; senior teacher.

Gromov Sergey Alexandrovich – Moscow State Technical University; e-mail: sergey.a.gromov@gmail.com; 5, 2-ya Baumanskaya street, build. 1, Moscow, Russia; postgraduate student.