

КОНСТАНТИН НИКОЛАЕВИЧ СЕНИЧЕВ

аспирант кафедры прикладной математики и кибернетики математического факультета, программист ООО «Опти-Софт», программист Центра ПетрГУ-Метсо Систем Автоматизации (ЦПМСА), Петрозаводский государственный университет (Петрозаводск, Российская Федерация)
csenichev@gmail.com

ЗАДАЧА ПОИСКА ОПТИМАЛЬНОГО РАСПИСАНИЯ ОДНОГО СТАНКА С ДИРЕКТИВНЫМИ СРОКАМИ РАБОТ*

Представлена задача оптимизации работы одного станка, которая может возникать при расчете оптимального плана работы бумагоделательной машины, гофроагрегата, фанерного пресса или других производственных агрегатов, последовательно и неразрывно выполняющих определенные работы. В случае преждевременного начала работы или ее завершения позже установленного директивного срока на расписание налагается штраф в заданных размерах, пропорциональных величине нарушения срока. Требуется минимизировать сумму штрафных санкций за выполнение всех работ. Также рассмотрены вариации задачи в условиях минимизации общего количества нарушенных директивных сроков, одинаковых значений верхних и нижних штрафных санкций, отсутствия ранних сроков начала выполнения работ. Для решения задачи использованы как базовые метаэвристические алгоритмы, так и предложенный автором комбинированный алгоритм, использующий набор метаэвристик для последовательного улучшения найденного решения. При этом для построения окрестности текущего решения используются операторы инверсии, сдвига в начало, сдвига в конец и сдвига в начало и в конец. В результате комбинированный алгоритм демонстрирует прирост качества найденного решения без изменения порядка вычислительной сложности.

Ключевые слова: директивные сроки, оптимальное расписание, метаэвристические методы, комбинированный алгоритм

Задача поиска оптимального расписания одного станка с директивными сроками работ возникает при расчете оптимального плана работы гофроагрегата, бумагоделательной машины (БДМ), фанерного пресса и других производственных агрегатов, последовательно и неразрывно выполняющих определенные работы.

В литературе достаточно широко представлены методы решения задачи поиска оптимального расписания работ одного станка в случае отсутствующих директивных сроков начала работ, основанные на применении эвристического подхода [7]. Однако изложенные подходы не обеспечивают эффективное решение задачи поиска оптимального расписания одного станка с заданными директивными сроками работ, востребованной на реальном производстве.

Метаэвристическим называют общий подход к решению широкого класса задач поиска или оптимизации, чаще всего не имеющий строгого обоснования, но дающий хорошие практические результаты [3], [4]. Метаэвристики основаны на определенных наблюдениях, не связанных с математикой, или являются попыткой приспособить рассуждения здравого смысла для решения определенных задач оптимизации. Данная работа описывает применение предложенного автором комбинированного алгоритма решения задачи поиска оптимального расписания одного станка с директивными сроками работ. Комбинирован-

ный метод опирается на использование таких базовых метаэвристических методов, как метод локальной оптимизации, генетический алгоритм, меметический алгоритм, метод отжига и метод табуированного поиска [2], [5], [9]. Также в работе приведены результаты численных экспериментов, подтверждающие эффективность авторского решения.

ОПИСАНИЕ ПРОБЛЕМЫ И МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Имеется один станок (в данном случае гофроагрегат), который в течение заданного временного промежутка должен последовательно, одну за другой, но в произвольном порядке выполнить множество работ, составляющих множество N . Каждая из этих работ должна выполняться в течение определенного времени. В случае начала работы ранее или ее завершения позже установленных (директивных) сроков на расписание накладывается штраф в заданных размерах, пропорциональных величине нарушения срока.

Требуется составить расписание выполнения работ $j \in N$, каждая из которых характеризуется следующими параметрами:

t_j – ее продолжительность;

a_j, b_j – ранний срок начала и поздний срок завершения;

c_j, \bar{c}_j – штраф за нарушение раннего срока начала и позднего срока завершения, пропорци-

ональный величине выхода из заданного интервала.

Целевой функцией z задачи является сумма штрафных санкций за выполнение всех работ, которую требуется минимизировать.

Решение задачи определяется $p = (p_1, p_2, p_3, \dots, p_n)$ – перестановкой очередности работ или неизвестными $x = (x_1, x_2, x_3, \dots, x_n)$ – связанными с ней моментами их запуска.

Проще всего построить математическую модель задачи в терминах $x = (x_1, x_2, x_3, \dots, x_n)$ моментов запуска работ. Действительно, в таком случае целевая функция имеет вид

$$z(x) = \sum_{j \in N} (\bar{c}_j \max(x_j + t_j - b_j, 0) + \underline{c}_j \max(a_j - x_j, 0)) \rightarrow \min. \quad (1)$$

Возможное перекрытие сроков выполнения работ устраняется наложением следующих дополнительных условий:

$$x_i + t_i \leq x_j \text{ или } x_j + t_j \leq x_i, \quad i \neq j \in N. \quad (2)$$

Задачу можно записать в терминах перестановки, сократив при этом количество ограничений (2).

Последовательность выполнения работ определим перестановкой $p = (p_1, p_2, p_3, \dots, p_n)$, где каждый элемент p_i содержит порядковый номер работы, которая будет обработана i -й по списку.

Зависимость момента запуска работ от перестановки можно записать следующим образом:

$$x_{p_1} = 0, \quad x_{p_i} = x_{p_{i-1}} + t_{p_{i-1}}, \quad i = 2..N.$$

Очевидно, что выполнение работ производится без простоев оборудования между обработкой последовательных работ.

ВАРИАНТЫ ЗАДАЧИ

Представляет интерес возможность использования метаэвристик для решения данной задачи, а также ряда ее следующих вариантов.

а) Случай минимизации общего количества нарушенных директивных сроков

$$z(x) = \sum_{j \in N} (\bar{c}_j \text{sign}(x_j + t_j - b_j) + \underline{c}_j \text{sign}(a_j - x_j)) \rightarrow \min.$$

б) Случай одинаковых значений верхних и нижних штрафных санкций

$$\underline{c}_j = \underline{c}, \quad \bar{c}_j = \bar{c}, \quad j \in N$$

в линейном и дискретном случаях.

$$z(x) = \sum_{j \in N} (\underline{c} \max(x_j + t_j - b_j, 0) + \bar{c} \max(a_j - x_j, 0)) \rightarrow \min.$$

Случай отсутствия ранних сроков начала выполнения работ $a_j = 0, \quad c_j = \bar{c}_j, \quad \forall j \in N$:

$$z(x) = \sum_{j \in N} c_j \max(x_j + t_j - b_j, 0) \rightarrow \min.$$

Отметим, что задача хорошо изучена и полиномиально разрешима в случае $a_j = b_j = 0, \quad c_j = \bar{c}_j, \quad \forall j \in N$ – классическая задача одного станка

$$z(x) = \sum_{j \in N} c_j (x_j + t_j) \rightarrow \min.$$

Комбинированный алгоритм

Опишем комбинированный алгоритм (обозначим его *СМВ*), позволяющий использовать преимущества различных метаэвристик для нахождения наилучшего решения.

Введем следующие обозначения:

Ω – множество допустимых решений.

$k = 1, 2, 3 \dots$ – номер текущей итерации.

z_{pek} – рекордное значение целевой функции, найденное в процессе вычислений; $x_k^* \in \Omega$ – план, на котором достигается $z_{pek} = f(x_k^*)$.

Далее, пусть M – набор базовых метаэвристических алгоритмов, включающий метод локальной оптимизации *LO*, генетический алгоритм *GA* и меметический алгоритм *MA*, метод отжига *SA* и метод табуированного поиска *TS*. Особенности программной реализации данных алгоритмов изложены в работе [2].

Тогда комбинированный алгоритм *СМВ* можно записать следующим образом:

Шаг 0 (Start)

01. Установить $z_{pek} = \infty$ и $k = 1$.

02. Построить начальное решение $x_0^* \in \Omega$ с использованием эвристики или случайного выбора.

Шаг k (Estimate)

k1. Найти решение $x_k^* \in \Omega$ методом M_k , используя $x_{k-1}^* \in \Omega$ в качестве начального решения.

k2. Установить $z_{pek} = f(x_k^*)$.

k3. Если $k < |M|$, то положить $k = k + 1$.

Иначе, закончить работу.

Finish.

Важно отметить, что время выполнения каждого алгоритма $M_k, k = 1..|M|$ может

- быть зафиксировано;
- определяться допустимым установленным временем счета;
- отслеживаться по динамике изменения рекорда.

Окрестность решения

Рассмотрим следующие способы построения окрестности перестановки.

а) Инверсия

Оператором инверсии двух работ, расположенных на позициях $i \neq j \in N$ перестановки $p = (p_1, p_2, p_3, \dots, p_n)$, будем называть обмен элементов p_i и p_j .

Определим окрестность Ω_p^{inv} перестановки $p = (p_1, p_2, p_3, \dots, p_n)$ как множество всех перестановок, полученных инверсией пары элементов p_i, p_j исходной перестановки $p, i \neq j \in N$.

б) Сдвиг в начало

Оператор сдвига вперед переставляет работу с позиции i в начало перестановки, то есть переводит перестановку

$$p = (p_1, p_2, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_{j-1}, p_j, p_{j+1}, \dots, p_n)$$

в перестановку

$$p' = (p_i, p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_{j-1}, p_j, p_{j+1}, \dots, p_n).$$

Множество перестановок, полученных применением оператора сдвига в начало к перестановке

$p = (p_1, p_2, p_3, \dots, p_n)$ для всех значений индекса $i = 1 \dots N$ будем называть окрестностью $\Omega_p^{forward}$.

с) Сдвиг в конец

Сдвиг назад перемещает работу с позиции j на последнюю позицию, то есть перестановку

$$p = (p_1, p_2, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_{j-1}, p_j, p_{j+1}, \dots, p_n)$$

преобразует в перестановку

$$p' = (p_1, p_2, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_{j-1}, p_{j+1}, \dots, p_n, p_j).$$

Окрестность Ω_p^{back} определим как множество перестановок, которые можно получить из перестановки $p = (p_1, p_2, p_3, \dots, p_n)$ с помощью оператора сдвига назад.

д) Сдвиг в начало и в конец

Данный оператор является комбинацией предыдущих двух. Элементы перестановки p_i, p_j , расположенные на позициях $i \neq j \in N$, перемещаются в начало и в конец перестановки соответственно. Таким образом, исходная перестановка

$$p = (p_1, p_2, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_{j-1}, p_j, p_{j+1}, \dots, p_n)$$

преобразуется в перестановку

$$p' = (p_i, p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_{j-1}, p_{j+1}, \dots, p_n, p_j).$$

Окрестностью $\Omega_p^{forback}$ назовем множество всех перестановок, полученных из перестановки p методом сдвига пары элементов в начало и в конец.

Тестовые данные

В сети Интернет имеется библиотека тестовых данных, расположенная в общем доступе на ресурсе [11]. Всего представлено 300 наборов входных параметров для $n = 40, 50$ и 100 , по 100 тестов на каждую размерность.

Длительности работ $p_j, j \in N$ сгенерированы согласно равномерному распределению на интервале $[0, 100]$, штрафы $\underline{c}_j, \bar{c}_j$ – на интервале $[0, 10]$.

Сроки выполнения работ a_j, b_j сгенерированы в зависимости от комбинации параметров $R, T \in \{0.2, 0.4, 0.6, 0.8, 1\}$ в интервале

$$[P(1 - T - R/2), P(1 - T + R/2)],$$

где $P = \sum_{j=1}^n p_j$.

РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

Для проведения тестов использована система следующей конфигурации: Intel Core i7–2630QM @ 2.00 GHz, 4 GB RAM. Программные модули реализованы на языке C# и выполняются в среде Microsoft.NET.

Критериями останова базовых алгоритмов являются превышение заданного времени выполнения, равного 180 секундам, или невозможность улучшить текущее решение. В случае комбинированного алгоритма CMB максимальное время выполнения делится на количество элементов множества M вложенных алгоритмов.

В табл. 1 представлены результаты вычислительного эксперимента решения задачи поиска расписания работ одного станка с директивными сроками при тестовых данных для $n = 40$ работ с использованием окрестностей перестановок $\Omega_p^{inv}, \Omega_p^{forward}, \Omega_p^{back}$ и $\Omega_p^{forback}$.

Табл. 2. содержит данные тестовых вычислений для случая $n = 50$ работ с использованием того же набора окрестностей.

Таблица 1
Среднее отклонение от оптимального решения (%) и среднее время счета (с) при $n = 40$

Метод	Среднее отклонение				Среднее время счета			
	Ω_p^{inv}	$\Omega_p^{forward}$	Ω_p^{back}	$\Omega_p^{forback}$	Ω_p^{inv}	$\Omega_p^{forward}$	Ω_p^{back}	$\Omega_p^{forback}$
LO	3,0	4,4	4,6	2,4	15,4	14,9	16,6	17,5
GA	4,9	2,6	2,2	2,9	31,5	28,3	29,0	27,2
MA	4,4	2,7	2,1	2,8	32,1	30,8	28,7	29,5
SA	3,9	4,0	4,3	3,0	22,7	18,5	20,2	19,4
TS	4,9	4,4	4,2	4,6	20,0	24,3	22,1	21,3
CMB	2,5	2,1	2,9	1,7	22,8	25,3	23,3	21,0

Таблица 2
Среднее отклонение от оптимального решения (%) и среднее время счета (с) при $n = 50$

Метод	Среднее отклонение				Среднее время счета			
	Ω_p^{inv}	$\Omega_p^{forward}$	Ω_p^{back}	$\Omega_p^{forback}$	Ω_p^{inv}	$\Omega_p^{forward}$	$\Omega_p^{forback}$	$\Omega_p^{forback}$
LO	3,1	4,5	4,0	2,8	21,6	21,1	22,9	20,7
GA	3,4	3,0	2,9	2,6	36,9	35,8	40,1	36,5
MA	3,9	2,4	5,9	2,3	43,2	43,4	39,7	41,5
SA	4,7	5,8	4,7	3,4	37,5	35,1	34,7	35,6
TS	4,6	5,0	5,9	4,0	41,8	43,8	43,1	40,9
CMB	2,3	2,2	3,4	2,1	35,9	33,8	35,2	33,4

Таблица 3

Среднее отклонение от оптимального решения (%) и среднее время счета (с) при $n = 100$

Метод	Среднее отклонение				Среднее время счета			
	Ω_p^{inv}	$\Omega_p^{forward}$	Ω_p^{back}	$\Omega_p^{forback}$	Ω_p^{inv}	$\Omega_p^{forward}$	$\Omega_p^{forward}$	$\Omega_p^{forback}$
LO	4,9	4,6	5,7	4,8	81,7	86,4	79,5	79,7
GA	3,8	4,1	4,8	3,7	103,2	105,4	108,5	103,9
MA	3,5	3,7	4,2	3,3	125,3	116,3	129,6	121,3
SA	3,8	5,3	5,7	4,3	100,5	93,9	102,0	97,6
TS	3,4	3,7	4,7	3,1	97,2	101,5	105,7	101,4
CMB	3,1	3,2	4,0	2,2	87,8	90,2	103,8	95,4

Результаты вычислений на тестовых данных для размерности $n = 100$ работ показаны в табл. 3.

Отметим, что в каждом из представленных случаев наименьшее среднее отклонение от оптимального решения достигается с использованием комбинированного алгоритма *CMB* и окрестности $\Omega_p^{forback}$. Среднее время выполнения комбинированного алгоритма незначительно уступает времени работы метода локальной оптимизации *LO*, что, однако, нивелируется большим ухудшением качества найденного решения, полученного последним методом, при увеличении размерности задачи.

ЗАКЛЮЧЕНИЕ

Задача поиска оптимального расписания одного станка с директивными сроками работ ха-

рактерна для достаточно широкого класса производственных процессов. Однако существующие методы ее решения, основанные на применении эвристического подхода, не являются достаточными для эффективного использования в реальных условиях.

Предложенный автором комбинированный алгоритм решения задачи поиска оптимального расписания работы одного станка с директивными сроками демонстрирует значительное улучшение качества найденного решения без изменения порядка вычислительной сложности. Наиболее успешные результаты достигаются в сочетании с оператором сдвига в начало и в конец для построения окрестности текущего решения.

* Работа выполнена при поддержке Программы стратегического развития ПетрГУ в рамках реализации комплекса мероприятий по развитию научно-исследовательской деятельности на 2012–2016 гг.

СПИСОК ЛИТЕРАТУРЫ

1. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учеб. пособие. 2-е изд. М.: Физматлит, 2006. 320 с.
2. Кузнецов В. А., Сеничев К. Н. Метаэвристики для решения комбинаторных задач: Учеб. пособие. Петрозаводск: Изд-во ПетрГУ, 2012. 40 с.
3. Кузнецов В. А., Сеничев К. Н. Использование метаэвристик для решения задач дискретного программирования // Вестник Череповецкого государственного университета (технические науки). 2012. № 4. С. 27–31.
4. Кузнецов В. А., Сеничев К. Н. Состав и структура метаэвристик для решения комбинаторных задач // Новые информационные технологии в ЦБП и энергетике: Материалы X Междунар. конф. Петрозаводск: Изд-во ПетрГУ, 2012. С. 34–35.
5. Мак Коннелл Дж. Основы современных алгоритмов. М.: Техносфера, 2004. 368 с.
6. Хачатуров В. Р., Веселовский В. Е., Злотов А. В. и др. Комбинаторные методы и алгоритмы решения задач дискретной оптимизации большой размерности. М.: Наука, 2000. 354 с.
7. Ballestin F., Leus R. Meta-heuristics for stable scheduling on a single machine // Computer and Operation Research. 2008. № 7. P. 2175–2192.
8. Blum C. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison // ACM Computing Surveys. 2003. № 3. P. 268–308.
9. Glover F., Kochenberger G. A. Handbook of Metaheuristics. Springer Science & Business Media, 2003. 556 p.
10. Mautor T. Intensification Neighborhoods for Local Search Methods. Essays and Surveys in Metaheuristics. Operation Research Computer Science. Kluwer Acad. Publ., 2001. P. 493–508.
11. OR-Library. Weighted tardiness. Available at: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html>

Senichev K. N., Opti-Soft Ltd, Petrozavodsk State University (Petrozavodsk, Russian Federation)

SEARCH OF OPTIMAL MASTER SCHEDULE FOR SINGLE MACHINE OPTIMIZATION

The article is concerned with the single machine optimization problem that is typical for scheduling production processes with a number of consecutive and non-preemptive operations. A paper machine, a plywood press or a corrugator execution plan construction can be considered as an example. The processing work should start at a particular release time and should be completed no later than the relative due date. Otherwise, a proportional to the violation fine is imposed. The objective of the problem is to find

a processing order of the works that minimize total weighted tardiness. Moreover, the study covers the case of the total number of violated due dates minimization and an option with the works available at zero start time. The solution of the problem is based on metaheuristic algorithms that provide an opportunity to obtain sufficiently good result. Additionally, the author proposes a new combined algorithm that exploits basic metaheuristics in order to consistently improve the current plan. Inversion operators, shift forward, shift back and shift forward, and back operators are used to employ obtained solutions in practice. As a result, the combined algorithm provides an important solution in quality improvement comparable to the level of computation complexity in mentioned earlier metaheuristics.

Key words: relative due dates, optimal schedule, metaheuristic algorithms, combined methods

REFERENCES

1. Gladkov L. A., Kureychik V. V., Kureychik V. M. *Geneticheskie algoritmy* [Genetic Algorithms: A Tutorial]. Moscow, Fizmatlit Publ., 2006. 320 p.
2. Kuznetsov V. A., Senichev K. N. *Metaevristiki dlya resheniya kombinatornykh zadach* [Metaheuristics for Combinatorial Problems: A Tutorial]. Petrozavodsk, PetrSU Publ., 2012. 40 p.
3. Kuznetsov V. A., Senichev K. N. The Use of Metaheuristics for Solving Discrete Programming Problems [Ispol'zovanie metaevristik dlya resheniya zadach diskretnogo programmirovaniya]. *Vestnik Cherepovetskogo gosudarstvennogo universiteta (tekhnicheskie nauki)*. 2012. № 4. P. 27–31.
4. Kuznetsov V. A., Senichev K. N. Composition and Structure of Metaheuristics for Combinatorial Problems [Sostav i struktura metaevristik dlya resheniya kombinatornykh zadach]. *Novye informatsionnye tekhnologii v TsBP i energetike: Materialy X Mezhdunarodnoy konferentsii*. Petrozavodsk, PetrSU Publ., 2012. P. 34–35.
5. MakKonnell Dzh. *Osnovy sovremennykh algoritmov* [Fundamentals of Modern Algorithms]. Moscow, Tekhnosfera Publ., 2004. 368 p.
6. Khachaturov V. R., Veselovskiy V. E., Zlotov A. V. i dr. *Kombinatornye metody i algoritmy resheniya zadach diskretnoy optimizatsii bol'shoy razmernosti* [Combinatorial Methods and Algorithms for Solving Discrete Optimization Problems of Large Dimensions]. Moscow, Nauka Publ., 2000. 354 p.
7. Ballestin F., Leus R. Meta-heuristics for stable scheduling on a single machine // *Computer and Operation Research*. 2008. № 7. P. 2175–2192.
8. Blum C. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison // *ACM Computing Surveys*. 2003. № 3. P. 268–308.
9. Glover F., Kochenberger G. A. *Handbook of Metaheuristics*. Springer Science & Business Media, 2003. 556 p.
10. Mautor T. Intensification Neighborhoods for Local Search Methods. *Essays and Surveys in Metaheuristics*. Operation Research Computer Science. Kluwer Acad. Publ., 2001. P. 493–508.
11. OR-Library. Weighted tardiness. Available at: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html>

Поступила в редакцию 20.11.2014