

УДК 519.8

**ПРИБЛИЖЕННЫЕ АЛГОРИТМЫ ДЛЯ ВАРИАНТОВ ЗАДАЧИ OPEN SHOP
С УЧЕТОМ РАСХОДА ЭНЕРГИИ¹****Ю. В. Захарова**

Рассматривается задача open shop с учетом расхода энергии. Исследуется вычислительная сложность и предлагаются подходы к решению для различных ее вариантов. Алгоритмы используют двухэтапную схему построения расписаний, где на первом этапе строятся оценки целевой функции и длительностей работ, а затем осуществляется переход от задачи с вариативными скоростями к задаче с фиксированными скоростями и используются методы списочного типа. В результате доказывается NP-трудность в общем случае и предлагаются полиномиальные точные и приближенные алгоритмы для практически значимых частных случаев, когда допускаются или нет прерывания, когда набор скоростей дискретен или непрерывен, когда потребление энергии ограничивается или оптимизируется. Строится модель частично целочисленного выпуклого программирования на основе непрерывного представления времени с использованием точек событий.

Ключевые слова: open shop, расписание, NP-трудность, алгоритм.

Yu. V. Zakharova. Approximation algorithms for Open Shop variations subject to energy consumption.

We consider the open shop scheduling problem subject to speed scaling and energy consumption. The computational complexity is analyzed and approaches to solving various variants of the problem are proposed. The algorithms use a two-stage scheduling scheme. At the first stage, bounds on the objective function and processing times of jobs are constructed. At the second stage, the speed scaling problem is reduced to the classic problem with fixed job speeds, and list-type methods are applied for scheduling. As a result, NP-hardness is proved in the general case, and polynomial-time exact and approximation algorithms are proposed for the practically important special cases when preemptions are allowed or not, when the set of speeds is discrete or continuous, and when energy consumption is bounded or optimized. A model of mixed integer convex programming is constructed based on continuous time representation using the notion of event points.

Keywords: open shop, schedule, NP-hardness, algorithm.

MSC: 90B35, 68M20, 90C59**DOI:** 10.21538/0134-4889-2024-30-4-117-133**Введение**

Цеховая задача составления расписаний имеет три основных варианта: одинаковые маршруты (flow shop), различные маршруты (job shop) и нефиксированные маршруты (open shop; см. [1; 2]). В первых двух случаях для работ заданы жесткие маршруты прохождения по машинам, а в последнем случае работы могут обслуживаться на машинах в произвольном порядке. В настоящей статье рассматривается задача open shop (цеховая задача открытого типа). Известно, что в случае, когда прерывания допускаются, задача полиномиально разрешима (см. [3]). Задача без прерываний с двумя машинами полиномиально разрешима, но является NP-трудной, начиная с трех машин [3]. Также широко известен 2-приближенный алгоритм для случая произвольного числа машин (см. [4]).

Современные производственные и компьютерные системы порождают новые задачи составления расписаний, где в роли машин выступают процессоры и учитывается потребление ресурсов возобновимого и невозобновимого типов. Возобновимые ресурсы приводят к ограничениям в каждый момент времени с последующим возобновлением в следующий момент

¹Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015, <https://rscf.ru/project/22-71-10015/>.

(например, кэш-память). Задача open shop с одним возобновимым ресурсом полиномиально разрешима (алгоритм базируется на решении задачи о максимальном потоке специального вида; см. [5]), а в случае двух таких ресурсов уже является NP-трудной (доказывается с помощью сводимости задачи “Разбиение”).

Невозобновимые ресурсы ограничиваются на некоторый период и распределяются между работами в различные моменты времени. Отметим, что длительности работ могут зависеть от объемов потребления ресурсов. Здесь речь пойдет о невозобновимых ресурсах, потребление которых влияет на длительности работ (см. [6]). Потребление ресурса и длительность работы связаны соотношением

$$p(R) = \left(\frac{Ch}{R} \right)^\kappa,$$

где $0 < \kappa < 1$ — заданная константа, Ch — характеристика объема работы, R — объем ресурса, потребляемый работой. Одним из представителей таких ресурсов является энергия (см. [7]), когда

$$p(R) = \left(\frac{Ch^\alpha}{R} \right)^{\frac{1}{\alpha-1}},$$

где $\alpha > 1$ — характеристика процессора (константа). Исследование вычислительной сложности задачи open shop с длительностями работ, зависящими от потребления ограниченного ресурса, представлено в [8]. Также в указанной статье предложено обобщение алгоритмов, известных для классической задачи open shop на случай наличия общего невозобновимого ресурса.

Задача open shop с учетом расхода энергии будет рассматриваться в следующей постановке. Дано множество работ $J = \{1, \dots, n\}$ и множество процессоров $P = \{1, \dots, m\}$. Каждая работа j состоит из m операций $O_{j1}, \dots, O_{jm}$, причем операция O_{ji} выполняется на процессоре i и имеет объем, равный W_{ji} . В каждый момент времени каждый процессор может выполнять не более одной операции, а также в каждый момент времени не может выполняться более одной операции каждой работы. Операции могут выполняться с разной скоростью. Если операция O_{ji} выполняется со скоростью s_{ji} , то потребление энергии в единицу времени (мгновенная мощность) составляет s_{ji}^α ($\alpha > 1$ — константа). Множество возможных скоростей может быть дискретным или непрерывным. Динамическая составляющая потребления энергии равна суммарному потреблению энергии процессорами во время выполнения операций $\sum_{j \in J} \sum_{i \in P} s_{ji}^\alpha \left(\frac{W_{ji}}{s_{ji}} \right)$. Также имеет место статическая составляющая энергопотребления, равная $F_{\text{stat}} C_{\text{max}}$, где C_{max} — общее время работы системы (длина расписания, makespan), $F_{\text{stat}} \geq 0$ — заданная константа (статическая мощность системы). Учет статической части характерен для систем, в которых допускается управление их выключением, и соответствует $F_{\text{stat}} > 0$. Если статическая составляющая не учитывается при оптимизации, например, когда сервер работает без перерывов или по заданному графику, то это задается через $F_{\text{stat}} = 0$.

Задача исследуется в двух вариантах целевой функции. В первом варианте требуется построить допустимое расписание σ_1 выполнения операций, при котором общее потребление энергии $\left[\sum_{j \in J} \sum_{i \in P} s_{ji}^\alpha \left(\frac{W_{ji}}{s_{ji}} \right) + m C_{\text{max}}(\sigma) F_{\text{stat}} \right]$ не превышает заданной величины E и длина расписания $C_{\text{max}}(\sigma_1)$ минимальна. Во втором варианте необходимо найти допустимое расписание σ_2 , при котором минимизируется общее потребление энергии $E(\sigma_2)$.

Задача исследуется как при допустимости прерывания операций (системы с общей памятью), так и при их запрете (системы с распределенной памятью). Также задача рассматривается в случае многопроцессорных операций (см. [9]), когда каждая работа состоит из m' операций и операция $i = 1, \dots, m'$ задействует некоторое подмножество процессоров $fix_i \subseteq P$, $|fix_i| \geq 1$. Обозначим задачу через $O|(fix_i), (pmtn), (discr), energy| C_{\text{max}}$ в случае, если потребление энергии рассматривается как ограничение, и через $O|(fix_i), (pmtn), (discr)| E$ — если как критерий. В обоих случаях по умолчанию считается, что потребление энергии включает в себя обе составляющие (статическую и динамическую).

Далее представим обзор известных результатов для задач с учетом расхода энергии. Ранее рассматривались задачи составления расписаний в одностадийном варианте при учете расхода энергии (см. [10; 11]). Исследовалась вычислительная сложность; показано, что задача составления расписаний независимых работ с прерываниями полиномиально разрешима, а без прерываний NP-трудна. Предлагались полиномиальные алгоритмы с гарантированной точностью для практически значимых частных случаев на основе различных комбинаторных подходов (декомпозиция; алгоритмы списочного типа; переход к задачам на одном процессоре или к специальным структурам расписаний, а также от задач с прерываниями к задачам без прерываний). Причем учитывалась только динамическая составляющая при расходе энергии. Случай многопроцессорных одностадийных работ исследовался, например, в [12–14]. Здесь предложены различные подходы к построению оценок целевой функции, решению задач с прерываниями (которые уже являются NP-трудными) и алгоритмы списочного типа для задач без прерываний. 2-приближенный алгоритм для задачи open shop с учетом расхода энергии (динамическая составляющая) обсуждался в работе [7]. Также известны статьи, где предложены и экспериментально исследованы эвристические алгоритмы для одностадийных задач (см. [15–17]). Результаты эксперимента показали, что хорошо зарекомендовали себя онлайн алгоритмы списочного типа с локальными улучшениями и алгоритмы, построенные по аналогии с известными жадными стратегиями типа first-fit, next-fit и best-fit для задачи об упаковке в контейнеры.

В литературе также представлены исследования задач составления расписаний в производственных системах с оптимизацией расхода энергии на различных уровнях (см. [18; 19]). Предложены модели частично целочисленного программирования, конструктивные эвристики и метаэвристики для решения задач цехового типа (см. [20–22]) с учетом специфики структуры решений, когда имеется дискретный набор вариантов выполнения работ с различными длительностями и энергопотреблением. Отметим, что близкие к исследуемой задаче постановки, где рассматриваются два критерия — потребление энергии и длина расписания, можно найти в современных работах (см., например, [18; 20; 22–24]).

В настоящей статье исследуется вычислительная сложность и строятся алгоритмы с гарантированной оценкой точности для различных вариантов задачи open shop с учетом расхода энергии, возникающей в компьютерных системах. Предлагаемые методы решения позволяют обобщить известные результаты для классической задачи open shop без учета ресурсных ограничений с критерием минимизации длины расписания на случай учета расхода энергии и получить новые результаты.

1. Однопроцессорные операции

В этом разделе рассматривается задача составления расписаний с учетом расхода энергии в виде критерия или ограничения для однопроцессорных операций при непрерывном и дискретном наборах скоростей. Выделяются полиномиально разрешимые и NP-трудные частные случаи, предлагается полиномиальный алгоритм с гарантированной оценкой точности.

1.1. Потребление энергии как ограничение. Непрерывное множество скоростей

В этом подразделе установим NP-трудность или полиномиальную разрешимость различных вариантов задачи при условии, что скорость может принимать произвольное значение больше нуля.

Теорема 1. *Задача $O3|energy|C_{\max}$ является NP-трудной.*

Доказательство. В случае $m \geq 3$ задача является NP-трудной. Для доказательства используется подход, представленный в статье [25] и основанный на ограничении дли-

ны расписания и общего потребления ресурса в распознавательном варианте задачи. Строим полиномиальную сводимость задачи “Разбиение” (см. [26]): даны множество элементов $A = \{a_1, a_2, \dots, a_{2n_0}\}$ и вес e_j каждого элемента $a_j \in A$ такие, что

$$\sum_{a_j \in A} e_j = 2B.$$

Требуется определить, может ли множество A быть разбито на подмножества A_1 и A_2 такие, что $\sum_{a_i \in A_1} e_i = \sum_{a_i \in A_2} e_i = B$.

Построим пример задачи open shop с ограничением на расход энергии: $n = 2n_0 + 1$, $m = 3$, $E = 18B$, $W_{ji} = e_j$, $j = 1, \dots, 2n_0$, $W_{2n_0+1,i} = B$ для $i = 1, 2, 3$. Покажем, что расписание σ с $C_{\max}(\sigma) \leq 3B$ существует тогда и только тогда, когда в задаче “Разбиение” ответ “да”.

Отметим, что общий объем работ составляет $\sum_{j=1}^n \sum_{i=1}^3 W_{ji} = 9B$. С учетом выпуклости функции расчета мгновенной мощности, а также заданных ограничений на расход энергии и длину расписания работы в допустимом расписании могут выполняться только с единичной скоростью. Причем как статическая, так и динамическая составляющие ограничены величиной $9B$.

Если в задаче “Разбиение” ответ “да”, т.е. существует разбиение A на два подмножества A_1 и A_2 , то допустимое расписание строится так, что первый процессор выполняет операции согласно последовательности $(2n_0 + 1, \pi_{A_1}, \pi_{A_2})$, второй — $(\pi_{A_1}, 2n_0 + 1, \pi_{A_2})$ и третий — $(\pi_{A_2}, \pi_{A_1}, 2n_0 + 1)$. Перестановка π_{A_i} задает произвольное упорядочение элементов из A_i , $i = 1, 2$.

Если существует допустимое расписание, то ввиду загрузки все процессоры работают без простоев в интервале $(0, 3B]$. Причем работа $2n_0 + 1$ выполняется на одном из процессоров в каждый момент времени, так как длительность всякой ее операции равна B . Значит, на одном из процессоров эта работа выполняется в интервале $(B, 2B]$, а остальные работы разбиваются на две равные по объему части и выполняются на этом процессоре в интервалах $(0, B]$ и $(2B, 3B]$. Эти части и формируют требуемые подмножества в задаче “Разбиение”, гарантируя ответ “да”. $\square$

При отсутствии статической составляющей в оценке потребления энергии ($F_{\text{stat}} = 0$) задача также является NP-трудной (строится аналогичная сводимость, но верхняя граница на потребление энергии полагается равной $9B$).

Теорема 2 (см. [7, теорема 5]). *Для задачи $O|\text{energy}|C_{\max}$ при учете только динамической составляющей потребления энергии ($F_{\text{stat}} = 0$) может быть найдено 2-приближенное решение за полиномиальное время.*

Доказательство. Для построения нижней оценки на целевую функцию будем использовать следующую модель выпуклого программирования (см. [7]). Пусть переменные p_{ji} соответствуют длительностям операций O_{ji} , $j \in J$, $i \in P$.

$$LB \rightarrow \min, \tag{1.1}$$

$$LB \geq \sum_{j=1}^n p_{ji}, \quad i \in P, \tag{1.2}$$

$$LB \geq \sum_{i=1}^m p_{ji}, \quad j \in J, \tag{1.3}$$

$$E \geq \sum_{j=1}^n \sum_{i=1}^m p_{ji} \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha, \tag{1.4}$$

$$p_{ji} \geq 0, \quad j \in J, \quad i \in P. \tag{1.5}$$

Задача (1.1)–(1.5) может быть решена за полиномиальное время (см., например, [27; 28]). Далее применяем алгоритм списочного типа с найденными длительностями работ: как только освобождается процессор $i \in P$, запускаем на нем операцию O_{ji} работы $j \in J$, которая в этот момент не выполняется ни на каком другом процессоре. Гарантированная оценка точности этого алгоритма равна двум по отношению к стандартной нижней оценке на длину расписания $\max\left\{\max_{i \in P} \sum_{j=1}^n p_{ji}; \max_{j \in J} \sum_{i=1}^m p_{ji}\right\}$ (см. [4; 29]). $\square$

Утверждение 1. *Задача $O|pmtn, energy|C_{\max}$ является полиномиально разрешимой.*

Доказательство. Задача с прерываниями полиномиально разрешима для произвольного числа процессоров. Сначала находится оптимальное решение по модели (1.1)–(1.3), (1.5) и

$$E \geq m(LB)F_{\text{stat}} + \sum_{j=1}^n \sum_{i=1}^m p_{ji} \left(\frac{W_{ji}}{p_{ji}}\right)^\alpha. \quad (1.6)$$

Это решение гарантирует получение нижней оценки $LB = \max\left\{\max_{i \in P} \sum_{j=1}^n p_{ji}; \max_{j \in J} \sum_{i=1}^m p_{ji}\right\}$ и соответствующих длительностей работ. Далее для этих длительностей строится оптимальное расписание с прерываниями и значением целевой функции равным LB с помощью известного алгоритма поиска оптимального расписания задачи с прерываниями при заданных длительностях работ (см., например, [2; 3]). $\square$

Теорема 3. *Задача $O2|energy|C_{\max}$ полиномиально разрешима.*

Доказательство. В случае двух процессоров задача без прерываний полиномиально разрешима. Сначала определим нижнюю оценку на целевую функцию и длительности работ по модели (1.1)–(1.3), (1.5)–(1.6) при $m = 2$.

Далее расписание с длиной LB отыскивается, например, по следующему правилу. Обозначим через J_1 работы, для которых $p_{j1} \leq p_{j2}$. Пусть $J_2 = J \setminus J_1$. Идентифицируем работу $k \in J_1$, для которой $p_{k2} \geq \max\{p_{j1}, j \in J_1\}$, и работу $r \in J_2$, для которой $p_{r1} \geq \max\{p_{j2}, j \in J_2\}$. Рассмотрим расписания, задаваемые перестановками $(\sigma'; \pi_{J_1 \setminus \{k\}}; \pi_{J_2 \setminus \{r\}}; \sigma'')$, где $\{\sigma', \sigma''\} = \{k, r\}$ (т. е. работы k, r стоят в начале или в конце расписания) и $\pi_{J'}$ — произвольная перестановка элементов из J' . Тогда одно из расписаний указанного вида будет иметь длину, равную LB (см. [2]). Трудоемкость описанной процедуры равна $O(n)$. $\square$

З а м е ч а н и е. Известны также следующие алгоритмы решения задачи open shop с двумя машинами при фиксированных длительностях операций за время $O(n)$: алгоритм Т. Гонзалеза, С. Сани [3], алгоритм М. Пинедо, Л. Шраге [30], алгоритм де Д. Верра [31], алгоритм А. Сопера [32], алгоритм А. Храмовой [33].

Отметим, что если в задаче учитывается только динамическая составляющая ($F_{\text{stat}} = 0$), то она всегда имеет допустимое решение, а если учитываются обе составляющие (статическая и динамическая), то допустимого решения может не существовать.

Сейчас рассмотрим вариант задачи, в котором скорость должна быть постоянной и одинаковой для всех процессоров. В варианте задачи, когда учитывается только динамическая составляющая ($F_{\text{stat}} = 0$), скорость s находится из уравнения $s^{\alpha-1} \sum_{j=1}^n \sum_{i=1}^m W_{ji} = E$ и длительности операций

$$p_{ji} = \frac{W_{ji}}{s} = W_{ji} \left(\frac{\sum_{j'=1}^n \sum_{i'=1}^m W_{j'i'}}{E} \right)^{\frac{1}{\alpha-1}}, \quad j \in J, \quad i \in P.$$

Далее с помощью описанных ранее подходов при фиксированных длительностях строятся 2-приближенное расписание (задача без прерываний и произвольное число процессоров) или оптимальное решение (двухпроцессорная задача без прерываний или задача с прерываниями).

В варианте задачи, когда учитываются динамическая и статическая составляющие, будем рассматривать только случаи, когда при фиксированных длительностях операций длина оптимального расписания совпадает со стандартной нижней оценкой (двухпроцессорная задача без прерываний или задача с прерываниями). Обозначим

$$LBW := \max \left\{ \max_{i \in P} \sum_{j=1}^n W_{ji}; \max_{j \in J} \sum_{i=1}^m W_{ji} \right\},$$

тогда задача сводится к поиску минимальной скорости среди скоростей $s > 0$, для которых имеет место неравенство $W s^{\alpha-1} + V s^{-1} \leq E$, или

$$W s^{\alpha} \leq E s - V, \quad (1.7)$$

где $W := \sum_{j \in J} \sum_{i \in P} W_{ji}$, $V := (LBW) F_{\text{stat}} m$. Построим касательную к графику функции $f(s) = W s^{\alpha}$ (левая часть неравенства), параллельную прямой $g(s) = E s - V$ (правая часть неравенства). Тогда точка касания соответствует скорости $\bar{s} = \left(\frac{E}{W\alpha} \right)^{\frac{1}{\alpha-1}}$, и уравнение касательной $y(s) = E s - V'$, где $V' = E^{\frac{\alpha}{\alpha-1}} (\alpha W)^{\frac{1}{\alpha-1}} (1 - 1/\alpha)$.

Следовательно, для $s > 0$ неравенство (1.7) не имеет решений при $V > E^{\frac{\alpha}{\alpha-1}} (\alpha W)^{\frac{1}{\alpha-1}} (1 - 1/\alpha)$. В противном случае неравенство (1.7) имеет решения, и единственное минимальное из них может быть найдено с заданной точностью с помощью численных методов (метод дихотомии; см., например, [34]).

1.2. Потребление энергии как ограничение. Дискретный набор скоростей

Рассмотрим случай, когда имеется дискретный набор возможных скоростей и допускаются прерывания выполнения операций.

Теорема 4. *Задача $O|pmtn, discr, energy|C_{\max}$ полиномиально разрешима.*

Доказательство. Покажем, что для указанной задачи оптимальное решение может быть найдено за полиномиальное время. Обозначим через C множество допустимых конфигураций операций на процессорах, которые могут выполняться параллельно. Каждая конфигурация c задается набором операций O_c и скоростью $s_{j,i,c}$ каждой операции $O_{ji} \in O_c$. Тогда мгновенная мощность (потребление энергии в единицу времени) для конфигурации $c \in C$ вычисляется как $E_c = \sum_{O_{ji} \in O_c} (s_{j,i,c})^{\alpha} + F_{\text{stat}} m$ (учитываются статическая и динамическая составляющие).

Пусть переменная x_c отвечает за длительность конфигурации c в расписании. Тогда для поиска оптимального решения можно использовать следующую модель выпуклого программирования:

$$\begin{aligned} \sum_{c \in C} x_c &\rightarrow \min, \\ \sum_{c \in C: O_{ji} \in O_c} \frac{s_{j,i,c} x_c}{W_{ji}} &\geq 1, \quad O_{ji} \in O_c, \\ - \sum_{c \in C} E_c x_c &\geq -E, \\ x_c &\geq 0, \quad c \in C. \end{aligned}$$

Построим двойственную задачу

$$\sum_{O_{ji} \in O_c} \lambda_{ji} - E \mu \rightarrow \max, \quad (1.8)$$

$$\sum_{O_{ji} \in O_c} \frac{s_{j,i,c} \lambda_{ji}}{W_{ji}} - E_c \mu \leq 1, \quad c \in C, \quad (1.9)$$

$$\lambda_{ji} \geq 0, \quad O_{ji} \in O_c, \quad (1.10)$$

$$\mu \geq 0. \quad (1.11)$$

Двойственная задача (1.8)–(1.11) решается методом эллипсоидов с отделяющим оракулом [35], который при заданном решении (λ_{ji}, μ) определяет, является ли решение допустимым, и если ответ отрицательный, то идентифицирует нарушенное ограничение.

Опишем, как работает отделяющий оракул в рассматриваемом случае. Согласно ограничению (1.9) требуется найти максимум суммы по всем конфигурациям. Если он больше 1, то найдено нарушенное ограничение. В противном случае решение допустимо.

Перепишем ограничение (1.9) в виде

$$\sum_{O_{ji} \in O_c} \left(\frac{s_{j,i,c} \lambda_{ji}}{W_{ji}} - (s_{j,i,c})^\alpha \mu \right) \leq F_{\text{stat}} m \mu + 1, \quad c \in C.$$

Для каждой операции O_{ji} скорость v_{ji} , на которой достигается максимум $\frac{s_{j,i,c} \lambda_{ji}}{W_{ji}} - (s_{j,i,c})^\alpha \mu$, отыскивается, как одна из ближайших к величине $\left(\frac{\lambda_{ji}}{W_{ji} \alpha \mu} \right)^{\frac{1}{\alpha-1}}$. Далее для поиска максимума $\sum_{O_{ji} \in O_c} \left(\frac{v_{ji} \lambda_{ji}}{W_{ji}} - (v_{ji})^\alpha \mu \right)$ строим следующий двудольный граф $G = (J, P, E)$. Доли вершин J и P соответствуют работам и процессорам. Ребро $(i, j) \in E$ согласуется с операцией O_{ji} и имеет вес $B_{ji} := \left(\frac{v_{ji} \lambda_{ji}}{W_{ji}} - (v_{ji})^\alpha \mu \right)$. Далее в отделяющем оракуле отыскивается паросочетание (независимое множество ребер) максимального веса за полиномиальное время (см. [36]). Если вес найденного паросочетания больше $F_{\text{stat}} m \mu + 1$, то решение недопустимо, и найдено нарушенное ограничение, иначе решение допустимо.

В результате двойственная задача разрешима за полиномиальное время, а значит, и исходная задача разрешима за полиномиальное время (см. [35]). $\square$

В случае поиска одной дискретной скорости для всей системы можно использовать подход из разд. 1.1 и выбирать одну из ближайших к оптимальной скорости, найденной для непрерывного случая.

1.3. Потребление энергии как критерий. Непрерывное множество скоростей

Покажем, как построить 2-приближенное решение для задачи минимизации расхода энергии, имея 2-приближенный алгоритм для задачи минимизации длины расписания при фиксированных длительностях операций. Обозначим через p_{ji} переменную, отвечающую за длительность операции $O_{ji} \in O$. Пусть переменная LB отвечает за нижнюю оценку на длину расписания.

Имеем модель выпуклого программирования

$$E_{LB} = \sum_{j=1}^n \sum_{i=1}^m p_{ji} \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha + (LB) F_{\text{stat}} m \rightarrow \min, \quad (1.12)$$

$$LB \geq \sum_{j=1}^n p_{ji}, \quad i \in P, \quad (1.13)$$

$$LB \geq \sum_{i=1}^m p_{ji}, \quad j \in J, \quad (1.14)$$

$$p_{ji} \geq 0, \quad j \in J, \quad i \in P. \quad (1.15)$$

Задача (1.12)–(1.15) может быть решена за полиномиальное время (см., например, [27; 28]). С полученными длительностями операций строим 2-приближенное решение для критерия минимизации длины расписания. Тогда

$$\begin{aligned} \sum_{j=1}^n \sum_{i=1}^m p_{ji} \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha + C_{\max}(p_{ji}) F_{\text{stat}} m &\leq \sum_{j=1}^n \sum_{i=1}^m p_{ji} \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha + 2(LB) F_{\text{stat}} m \\ &\leq 2 \left(\sum_{j=1}^n \sum_{i=1}^m p_{ji} \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha + (LB) F_{\text{stat}} m \right) \leq 2 E_{LB}. \end{aligned}$$

Таким образом, имеет место следующая теорема.

Теорема 5. *Для задачи $O||E$ можно построить 2-приближенное решение за полиномиальное время.*

Отметим, что для задачи с двумя процессорами данный подход позволяет получить оптимальное расписание, так как за полиномиальное время можно всегда построить расписание длины LB .

Используя подход из статьи [25] аналогично доказательству теоремы 1, можно показать, что задача с энергетическим критерием NP-трудна при $m \geq 3$ процессорах.

Рассмотрим случай, когда не для каждой операции индивидуально выбирается скорость, а необходимо задать одну скорость, с которой система будет работать и выполнять операции на всех процессорах. Напомним обозначение $W := \sum_{j \in J} \sum_{i \in P} W_{ji}$ — суммарный объем всех операций. Тогда для расписания σ со скоростью всех операций s_σ потребление энергии составляет $E(\sigma, s_\sigma) = W s_\sigma^{\alpha-1} + m C_{\max}(\sigma) F_{\text{stat}}$. Обозначим оптимальное расписание через σ_{opt} и соответствующее ему потребление энергии — через $E(\sigma_{\text{opt}})$.

Утверждение 2. *Пусть дано расписание σ с единичными скоростями операций для задачи $O||E$. Тогда минимальное потребление энергии для этого расписания достигается при скорости*

$$s_\sigma^E = \left(\frac{m C_{\max} F_{\text{stat}}}{W(\alpha-1)} \right)^{1/\alpha}.$$

Доказательство ведем путем вычисления производной и приравниванием ее к нулю: $\frac{dE}{ds} = \left(W s^{\alpha-1} + m \frac{C_{\max}}{s} F_{\text{stat}} \right)' = W(\alpha-1) s^{\alpha-2} - m \frac{C_{\max}}{s^2} F_{\text{stat}} = 0$.
Тогда

$$s^\alpha = \left(\frac{m C_{\max} F_{\text{stat}}}{W(\alpha-1)} \right). \quad \square$$

При скорости s_σ^E расписание σ будет иметь потребление энергии

$$E(\sigma, s_\sigma^E) = (m F_{\text{stat}})^{\frac{\alpha-1}{\alpha}} W^{\frac{1}{\alpha}} \left((\alpha-1)^{\frac{1-\alpha}{\alpha}} + (\alpha-1)^{\frac{1}{\alpha}} \right) (C_{\max}(\sigma))^{\frac{\alpha-1}{\alpha}}.$$

Пусть σ_2 есть 2-приближенное решение относительно критерия минимизации длины расписания при единичных скоростях операций. Следовательно, можно показать, что

$$C_{\max}(\sigma_2, s_{\sigma_2}^E) \leq C_{\max}(\sigma_2, s_{\sigma^*}^E) \leq 2 C_{\max}(\sigma^*, s_{\sigma^*}^E),$$

где σ^* есть оптимальное расписание относительно критерия минимизации длины расписания при единичных скоростях. В результате имеем

$$\begin{aligned} E(\sigma_2, s_{\sigma_2}^E) &= (m F_{\text{stat}})^{\frac{\alpha-1}{\alpha}} W_{\alpha}^{\frac{1}{\alpha}} \left((\alpha-1)^{\frac{1-\alpha}{\alpha}} + (\alpha-1)^{\frac{1}{\alpha}} \right) (C_{\max}(\sigma_2, s_{\sigma_2}^E))^{\frac{\alpha-1}{\alpha}} \\ &\leq 2^{\frac{\alpha-1}{\alpha}} (m F_{\text{stat}})^{\frac{\alpha-1}{\alpha}} W_{\alpha}^{\frac{1}{\alpha}} \left((\alpha-1)^{\frac{1-\alpha}{\alpha}} + (\alpha-1)^{\frac{1}{\alpha}} \right) (C_{\max}(\sigma^*, s_{\sigma^*}^E))^{\frac{\alpha-1}{\alpha}} \leq 2^{\frac{\alpha-1}{\alpha}} E(\sigma_{\text{opt}}), \end{aligned}$$

т. е. доказана следующая теорема.

Теорема 6. Для задачи $O||E$ может быть найдено $2^{\frac{\alpha-1}{\alpha}}$ -приближенное решение за полиномиальное время при дополнительном условии выбора одной скорости для всех операций.

1.4. Потребление энергии как критерий. Дискретное множество скоростей

Рассмотрим случай, когда имеется дискретный набор возможных скоростей и допускаются прерывания выполнения операций. Тогда оптимальное решение может быть найдено за полиномиальное время.

Теорема 7. Задача $O|pmtn, discr|E$ полиномиально разрешима.

Доказательство аналогично доказательству теоремы 4. Обозначим через C множество допустимых конфигураций операций на процессорах, которые могут выполняться параллельно. Каждая конфигурация c задается набором операций O_c и скоростью $s_{j,i,c}$ каждой операции $O_{ji} \in O_c$. Тогда мгновенная мощность (потребление энергии в единицу времени) для конфигурации $c \in C$ вычисляется как $E_c = \sum_{O_{ji} \in O_c} (s_{j,i,c})^{\alpha} + F_{\text{stat}} m$.

Пусть переменная x_c отвечает за длительность конфигурации c в расписании. Тогда для поиска оптимального решения можно использовать следующую модель выпуклого программирования:

$$\begin{aligned} \sum_{c \in C} E_c x_c &\rightarrow \min, \\ \sum_{c \in C: O_{ji} \in O_c} \frac{s_{j,i,c} x_c}{W_{ji}} &\geq 1, \quad O_{ji} \in O, \\ x_c &\geq 0, \quad c \in C. \end{aligned}$$

Двойственная задача решается методом эллипсоидов с отделяющим оракулом, который, как и в задаче с ограничением на расход энергии (теорема 4), основан на решении задачи поиска паросочетания максимального веса в двудольном графе. $\square$

При дискретном наборе скоростей задача без прерываний является NP-трудной уже при $m = 3$ (считаем, что есть только единичная скорость и сводим задачу “Разбиение”, как в теореме 1; остается минимизировать только длину расписания, так как динамическая составляющая фиксирована при одной допустимой скорости).

Построим приближенный алгоритм для задачи без прерываний с дискретным набором скоростей. Схема представлена в алгоритме 1.

А л г о р и т м 1. Приближенный алгоритм для задачи $O|discr|E$.

- (1) Для каждой операции O_{ji} выбираем скорость s_{ij} (соответственно, длительность $p_{ji} = W_{ji}/s_{ji}$), при которой величина $(p_{ji}(s_{ji})^{\alpha} + p_{ji} F_{\text{stat}})$ минимальна, $j \in J$, $i \in P$. Это достигается на одной из двух ближайших к $\left(\frac{F_{\text{stat}}}{\alpha-1}\right)^{1/\alpha}$ скоростей.

- (2) При фиксированных длительностях операций p_{ji} строим приближенное расписание σ_A такое, что момент окончания последней операции

$$C_{\max}(\sigma_A) \leq 2 \max \left\{ \max_{i \in P} \sum_{j \in J} p_{ji}; \max_{j \in J} \sum_{i \in P} p_{ji} \right\}.$$

Покажем, что построенное решение σ_A является $2m$ -приближенным, обозначая оптимальное по потреблению энергии расписание через σ_{opt} . По построению имеем

$$\begin{aligned} E(\sigma_A) &\leq \sum_{j \in J} \sum_{i \in P} p_{ji} + 2m F_{\text{stat}} \max \left\{ \max_{i \in P} \sum_{j \in J} p_{ji}; \max_{j \in J} \sum_{i \in P} p_{ji} \right\} \leq 2m \sum_{j \in J} \sum_{i \in P} (p_{ji}(s_{ji})^\alpha + F_{\text{stat}} p_{ji}) \\ &\leq 2m \sum_{j \in J} \sum_{i \in P} (p_{ji}(\sigma_{\text{opt}})(s_{ji}(\sigma_{\text{opt}}))^\alpha + F_{\text{stat}} p_{ji}(\sigma_{\text{opt}})) \leq 2m E(\sigma_{\text{opt}}). \end{aligned}$$

Подчеркнем, что величина $\sum_{j \in J} \sum_{i \in P} (p_{ji}(s_{ji})^\alpha + F_{\text{stat}} p_{ji})$ дает нижнюю границу на потребление энергии, так как для каждой операции выбирается наилучшая скорость на совокупную составляющую потребления энергии при выполнении операции процессором и не учитываются простои в расписании.

Таким образом, верна теорема.

Теорема 8. *Для задачи $O|discr|E$ может быть найдено $2m$ -приближенное решение за полиномиальное время.*

Отметим, что описанный подход работает в случае произвольных функциональных зависимостей между скоростями и длительностями операций, а не только при $p_{ji} = \frac{W_{ji}}{s_{ji}}$, $j \in J$, $i \in P$.

Теперь рассмотрим задачу, в которой необходимо выбрать одну дискретную скорость для всех операций (и соответственно процессоров).

Теорема 9. *Для задачи $O|discr|E$ может быть найдено 2-приближенное решение за полиномиальное время при дополнительном условии выбора одной скорости для всех операций.*

Доказательство. Основная идея доказательства заключается в том, что изначально длительности операций считаются равными их объемам и составляется 2-приближенное решение для критерия минимизации длины расписания. Далее для этого расписания назначается скорость, гарантирующая 2-приближенное решение с точки зрения расхода энергии.

Предположим, что есть допустимое 2-приближенное расписание σ_A при единичных скоростях работ для критерия минимизации длины расписания. Тогда оптимальная скорость для этого расписания вычисляется как экстремум функции $\frac{\sum_{j \in J} \sum_{i \in P} W_{ji}}{s} s^\alpha + m F_{\text{stat}} C_{\max}(\sigma_A)$ и равна

$$s_A^* = \left(\frac{m F_{\text{stat}} C_{\max}(\sigma_A)}{(\alpha - 1) \sum_{j \in J} \sum_{i \in P} W_{ji}} \right)^{1/\alpha}.$$

Поскольку набор скоростей дискретен, то необходимо выбрать лучшую s_{discr}^* из двух скоростей, ближайших к s_A^* . Ясно, что в оптимальном расписании σ^* относительно критерия минимизации длины расписания скорость будет равна максимальной $s_{\max}$ из доступных скоростей. Тогда $C_{\max}(\sigma_A, s_{\max}) \leq 2 C_{\max}(\sigma^*, s_{\max})$.

Как и ранее, будем обозначать через $(\sigma_{\text{opt}}, s_{\text{opt}})$ оптимальные расписание и скорость относительно критерия минимизации расхода энергии. Тогда общее потребление энергии при расписании σ_A и скорости s_{discr}^* характеризуется свойствами:

$$E(\sigma_A, s_{discr}^*) = \sum_{j \in J} \sum_{i \in P} W_{ji} (s_{discr}^*)^{\alpha-1} + m F_{\text{stat}} C_{\max}(\sigma_A, s_{\max}) \frac{s_{\max}}{s_{discr}^*}$$

$$\begin{aligned}
 & \leq \sum_{j \in J} \sum_{i \in P} W_{ji} (s_{\text{opt}})^{\alpha-1} + m F_{\text{stat}} C_{\text{max}} (\sigma_A, s_{\text{max}}) \frac{s_{\text{max}}}{s_{\text{opt}}} \\
 & \leq \sum_{j \in J} \sum_{i \in P} W_{ji} (s_{\text{opt}})^{\alpha-1} + m F_{\text{stat}} 2 C_{\text{max}} (\sigma^*, s_{\text{max}}) \frac{s_{\text{max}}}{s_{\text{opt}}} \\
 & \leq \sum_{j \in J} \sum_{i \in P} W_{ji} (s_{\text{opt}})^{\alpha-1} + m F_{\text{stat}} 2 C_{\text{max}} (\sigma_{\text{opt}}, s_{\text{opt}}) \frac{s_{\text{opt}}}{s_{\text{opt}}} \\
 & \leq 2 \left(\sum_{j \in J} \sum_{i \in P} W_{ji} (s_{\text{opt}})^{\alpha-1} + m F_{\text{stat}} C_{\text{max}} (\sigma_{\text{opt}}, s_{\text{opt}}) \right) \leq 2 E(\sigma_{\text{opt}}, s_{\text{opt}}).
 \end{aligned}$$

Первое неравенство следует из того, что для σ_A скорость s_{discr}^* является наилучшей в отличие от скорости s_{opt} . Второе неравенство гарантируется тем, что σ_A есть 2-приближенное расписание относительно критерия минимизации длины расписания, а третье неравенство — тем, что σ^* есть оптимальное решение со скоростью s_{max} для указанного критерия. $\square$

2. Многопроцессорные операции и ограниченное число стадий

В данном разделе рассматривается задача open shop с ограничением на расход энергии или с энергетическим критерием, когда работа состоит из двух или трех стадий, но операции задействуют при своем выполнении несколько процессоров. Обозначим через fix_i множество процессоров, которые задействует операция $i = 1, 2, 3$. В этом случае будем считать W_{ji} объемом операции Q_{ji} на каждом из используемых процессоров, $i = 1, 2, 3$, $j \in J$.

В случае двух стадий имеют место два случая: стадии совместны ($fix_1 \cap fix_2 = \emptyset$) и стадии несовместны ($fix_1 \cap fix_2 \neq \emptyset$).

Если стадии несовместны, то получаем случай, эквивалентный задаче составления расписаний на одном процессоре, где порядок следования операций может быть любым, а длительности p_{ji} находятся из модели выпуклого программирования с помощью условий Каруша — Куна — Таккера (доказательство аналогично представленному в работе [14]):

$$\sum_{j \in J} (p_{j1} + p_{j2}) \rightarrow \min,$$

$$\sum_{j \in J} |fix_1| p_{j1} \left(\frac{W_{j1}}{p_{j1}} \right)^\alpha + \sum_{j \in J} |fix_2| p_{j2} \left(\frac{W_{j2}}{p_{j2}} \right)^\alpha + \left(\sum_{j \in J} (p_{j1} + p_{j2}) \right) F_{\text{stat}} m \leq E$$

для случая ограничения на расход энергии,

$$\sum_{j \in J} |fix_1| p_{j1} \left(\frac{W_{j1}}{p_{j1}} \right)^\alpha + \sum_{j \in J} |fix_2| p_{j2} \left(\frac{W_{j2}}{p_{j2}} \right)^\alpha + \left(\sum_{j \in J} (p_{j1} + p_{j2}) \right) F_{\text{stat}} m \rightarrow \min$$

для случая энергетического критерия.

Если стадии совместны, то задача сводится к случаю задачи open shop с двумя процессорами, где длительности операций находятся из модели выпуклого программирования

$$LB \rightarrow \min, \tag{2.1}$$

$$LB \geq \sum_{j=1}^n p_{ji}, \quad i = 1, 2, \tag{2.2}$$

$$LB \geq p_{j1} + p_{j2}, \quad j \in J, \tag{2.3}$$

$$E \geq \sum_{j=1}^n \sum_{i=1}^2 |fix_i| \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha p_{ji} + (LB) F_{\text{stat}} m, \tag{2.4}$$

$$p_{ji} \geq 0, \quad j \in J, \quad i = 1, 2, \tag{2.5}$$

для случая ограничения на расход энергии и из модели

$$\sum_{j=1}^n \sum_{i=1}^2 |fix_i| \left(\frac{W_{ji}}{p_{ji}} \right)^\alpha p_{ji} + (LB) F_{\text{stat}} m \rightarrow \min, \quad (2.6)$$

$$LB \geq \sum_{j=1}^n p_{ji}, \quad i = 1, 2, \quad (2.7)$$

$$LB \geq p_{j1} + p_{j2}, \quad j \in J, \quad (2.8)$$

$$p_{ji} \geq 0, \quad j \in J, \quad i = 1, 2, \quad (2.9)$$

для случая минимизации расхода энергии.

Теперь перейдем к рассмотрению случая задачи open shop с тремя стадиями при многопроцессорных операциях. Если все три стадии несовместны ($fix_1 \cap fix_2 \neq \emptyset$, $fix_1 \cap fix_3 \neq \emptyset$, $fix_2 \cap fix_3 \neq \emptyset$), то задача сводится к однопроцессорному случаю, как и при двух несовместных стадиях из ранее представленных результатов.

Если одна из стадий не совместна с двумя другими (например, $fix_2 \cap fix_1 \neq \emptyset$, $fix_2 \cap fix_3 \neq \emptyset$, $fix_1 \cap fix_3 = \emptyset$), то операции второй стадии можно поставить в любом порядке в начале расписания, а для двух других стадий составить расписание, как для случая двух процессоров. При этом длительности операций могут быть найдены с помощью моделей выпуклого программирования, аналогичных моделям (2.1)–(2.5) и (2.6)–(2.9).

Если две стадии не совместны друг с другом, а еще одна стадия совместна со всеми (например, $fix_1 \cap fix_2 \neq \emptyset$, $fix_1 \cap fix_3 = \emptyset$, $fix_2 \cap fix_3 = \emptyset$), то операции двух первых стадий можно объединить в одну и для двух новых стадий (первая совместно со второй и третья) найти решение, как для случая двух процессоров. В случае двухпроцессорной задачи open shop всякое расписание с прерываниями можно превратить в расписание без прерываний без увеличения его длины (см. [3]), поэтому возможным прерыванием между двумя первыми стадиями можно пренебречь. Модели выпуклого программирования для поиска длительностей операций и нижней границы на длину расписания имеют вид, аналогичный моделям (2.1)–(2.5) и (2.6)–(2.9), но с учетом особенностей комбинации стадий.

3. Модель выпуклого программирования

В настоящем разделе будет представлена модель частично целочисленного выпуклого программирования для общего случая, когда операции могут задействовать заданное подмножество процессоров и прерывания операций запрещены. Обозначим через $O = \{o_1, \dots, o_k\}$ множество операций всех работ; пусть $P = \{1, \dots, m\}$, как и ранее, обозначает множество процессоров, а $J = \{1, \dots, n\}$ — множество работ. Через fix_o будем обозначать множество процессоров, используемых операцией $o \in O$, а через W_o — объем данной операции на каждом из задействуемых процессоров. Пусть $F_l = \{o \in O : l \in fix_o\}$ есть подмножество операций, использующих процессор $l \in P$, а O_j — множество операций работы $j \in J$.

При построении модели будем использовать понятие точки событий, соответствующей в расписании подмножеству операций, выполняемых одновременно, с указанием моментов их начала, окончания и скорости. Множество точек событий обозначим через $K = \{1, \dots, k_{\max}\}$, где $k_{\max} \leq k$.

Введем переменные: $x_{oi} = \begin{cases} 1, & \text{если операция } o \text{ назначена в точку событий } i, \\ 0 & \text{иначе;} \end{cases}$

$T_{oi}^f \geq 0$ — момент окончания операции o в точке событий i ; $T_{oi}^s \geq 0$ — момент начала операции o в точке событий i ; $s_o \geq 0$ — скорость операции o , $o \in O$, $i \in K$.

Обозначим через H достаточно большую константу.

Тогда модель может быть записана следующим образом:

$$C_{\max} \rightarrow \min$$

$$C_{\max} \geq T_{oi}^f, \quad o \in O, \quad i \in K, \quad (3.1)$$

$$\sum_{o \in F_l} x_{oi} \leq 1, \quad l \in P, \quad i \in K, \quad (3.2)$$

$$\sum_{i \in K} x_{oi} = 1, \quad o \in O, \quad (3.3)$$

$$\sum_{o \in O_j} x_{oi} \leq 1, \quad j \in J, \quad i \in K, \quad (3.4)$$

$$T_{oi}^f \geq T_{oi}^s, \quad o \in O, \quad i \in K, \quad (3.5)$$

$$T_{oi}^s \geq T_{o'i'}^f - H(2 - x_{oi} - x_{o'i'}), \quad l \in P, \quad o, o' \in F_l, \quad i, i' \in K: i' < i, \quad i \neq 1, \quad (3.6)$$

$$T_{oi}^f - T_{oi}^s \leq x_{oi}H, \quad o \in O, \quad i \in K, \quad (3.7)$$

$$s_o \sum_{i \in K} (T_{oi}^f - T_{oi}^s) \geq W_o, \quad o \in O, \quad (3.8)$$

$$\sum_{o \in O} |fix_o| W_o (s_o)^{\alpha-1} + C_{\max} F_{\text{stat}} m \leq E, \quad (3.9)$$

$$x_{oi} \in \{0, 1\}, \quad T_{oi}^s \geq 0, \quad T_{oi}^f \geq 0, \quad s_o \geq 0, \quad o \in O, \quad i \in K.$$

В случае минимизации расхода энергии критерий записывается формулой

$$C_{\max} F_{\text{stat}} m + \sum_{o \in O} |fix_o| W_o (s_o)^{\alpha-1} \rightarrow \min .$$

Ограничения (3.1) задают оценки снизу на длину расписания. Неравенство (3.2) гарантирует, что на каждом процессоре в каждой точке событий выполняется не более одной операции. Ограничение (3.3) соответствует назначению каждой операции ровно в одну точку событий. Неравенство (3.4) задает условие, согласно которому в каждой точке событий по каждой работе выполняется не более одной операции. Ограничения (3.5)–(3.6) задают монотонность времени в точках событий и между ними. Ограничение (3.7) говорит о том, что если операция не назначена в точку событий, то ее длительность выполнения равна нулю в этой точке событий. Ограничение (3.8) согласует длительности, скорости и объемы операций. Ограничение (3.9) задает границу на общее потребление энергии.

Данная модель может использоваться в пакетах прикладных программ решения задач частично целочисленного выпуклого программирования для поиска верхних и нижних оценок на целевую функцию.

Заключение

В работе представлен анализ вычислительной сложности различных вариантов задачи open shop с учетом расхода энергии в виде критерия или ограничения. Благодаря двухэтапному подходу удалось получить новые результаты с учетом свойств расписаний с гарантированной оценкой точности для критерия минимизации длины расписания. Исследованы свойства решений с учетом статической и динамической составляющих в потреблении энергии. В результате построены полиномиальные точные алгоритмы и алгоритмы с гарантированной точностью для различных случаев задачи open shop. Представляет интерес для дальнейших исследований обобщение результатов на другие многопроцессорные компьютерные системы с различными типами работ.

СПИСОК ЛИТЕРАТУРЫ

1. **Kubiak W.** A book of open shop scheduling. Cham: Springer Internat. Publ., 2022. 290 p. doi: 10.1007/978-3-030-91025-9
2. **Танаев В.С., Сотсков Ю.Н., Струсович В.А.** Теория расписаний: Многостадийные системы. М.: Наука, 1989. 327 с.
3. **Gonzalez T., Sahni S.** Open shop scheduling to minimize finish time // J. ACM. 1976. Vol. 23, no. 4. P. 665–679. doi: 10.1145/321978.321985
4. **Barany I., Fiala T.** Nearly optimum solution of multimachine scheduling problems (in Hungarian) // Szigma Matematika Kozgazdasagi Folyoirat. 1982. Vol. 15. P. 177–191.
5. **Jurisch B., Kubiak W.** Two-machine open shops with renewable resources // Oper. Res. 1997. Vol. 45, no. 4. P. 544–552. doi: 10.1287/opre.45.4.544
6. **Shabtay D., Kaspi M.** Parallel machine scheduling with a convex resource consumption function // European J. Oper. Res. 2006. Vol. 173, no. 1. P. 92–107. doi: 10.1016/j.ejor.2004.12.008
7. **Bampis E., Letsios D., Lucarelli G.** A note on multiprocessor speed scaling with precedence constraints // SPAA '14: Proc. 26th ACM Symposium on Parallelism in algorithms and architectures. 2014. P. 138–142. doi: 10.1145/2612669.2612672
8. **Shabtay D., Kaspi M.** Minimizing the makespan in open-shop scheduling problems with a convex resource consumption function // Naval Research Logistics. 2006. Vol. 53, no. 3. C. 204–216. doi: 10.1002/nav.20133
9. **Adak Z., Arioglu Akan M.O., Bulkan S.** Multiprocessor open shop problem: literature review and future directions // J. Comb. Optim. 2020. Vol. 40. P. 547–569. doi: 10.1007/s10878-020-00591-3
10. **Gerards M.E.T., Hurink J.L., Holzspies P.K.F.** A survey of offline algorithms for energy minimization under deadline constraints // J. Scheduling. 2016. Vol. 19. P. 3–19. doi: 10.1007/s10951-015-0463-8
11. **Li D., Wu J.** Energy-aware scheduling on multiprocessor platforms. NY: Springer, 2012. 59 p. (Ser. Springer Science and Business Media). doi: 10.1007/978-1-4614-5224-9
12. **Kononov A., Kovalenko Y.** Approximation algorithms for energy-efficient scheduling of parallel jobs // J. Scheduling. 2020. Vol. 23, no. 6. P. 693–709. doi: 10.1007/s10951-020-00653-8
13. **Kononov A., Zakharova Y.** Speed scaling scheduling of multiprocessor jobs with energy constraint and makespan criterion // J. Glob. Optim. 2022. Vol. 83. P. 539–564. doi: 10.1007/s10898-021-01115-x
14. **Kononov A., Zakharova Y.** Speed scaling scheduling of multiprocessor jobs with energy constraint and total completion time criterion // Inter. J. Artif. Intel. 2023. Vol. 21, no. 2. P. 109–129.
15. **Li K.** Energy efficient scheduling of parallel tasks on multiprocessor computers // J. Supercomput. 2020. Vol. 60. P. 223–247. doi: 10.1007/s11227-010-0416-0
16. **Zakharova Yu., Sakhno M.** Complexity and heuristic algorithms for speed scaling scheduling of parallel jobs with energy constraint // J. Comput. Appl. Math. 2024. Art. 116254. Опубликована онлайн: 6.10.2024. doi: 10.1016/j.cam.2024.116254
17. **Kong F., Guan N., Deng Q., Yi W.** Energy-efficient scheduling for parallel real-time tasks based on level-packing // Proc. 2011 ACM Symposium on Applied Computing. 2011. P. 635–640. doi: 10.1145/1982185.1982326
18. **Gao M., He K., Li L., Wang Q., Liu C.** A review on energy consumption, energy efficiency and energy saving of metal forming processes from different hierarchies // Processes. 2019. Vol. 7, no. 6. P. 357–381. doi: 10.3390/pr7060357
19. **Mouzon G., Yildirim M.B., Twomey J.** Operational methods for minimization of energy consumption of manufacturing equipment // Inter. J. Prod. Res. 2007. Vol. 45, no. 18–19. P. 4247–4271. doi: 10.1080/00207540701450013
20. **Mansouri S.A., Aktas E., Besikci U.** Green scheduling of a two-machine flow-shop: trade-off between makespan and energy consumption // European J. Oper. Res. 2016. Vol. 248, № 3. P. 772–788. doi: 10.1016/j.ejor.2015.08.064
21. **He L.J., Cao Y.L., Li W.F., Cao J.J., Zhong L.C.** Optimization of energy-efficient open shop scheduling with an adaptive multi-objective differential evolution algorithm // Appl. Soft Comput. 2022. Vol. 118. Art. 108459. doi: 10.1016/j.asoc.2022.108459
22. **Salido M.A., Escamilla J., Giret A., Barber F.** A genetic algorithm for energy-efficiency in job-shop scheduling // The Inter. J. Adv. Manufact. Techn. 2016. Vol. 85. P. 1303–1314. doi: 10.1007/s00170-015-7987

23. **Stewart R., Raith A., Sinnen O.** Optimising makespan and energy consumption in task scheduling for parallel systems // *Comp. Oper. Res.* 2023. Vol. 154. Art. 106212. doi: 10.1016/j.cor.2023.106212
24. **Sathya Sofia A., GaneshKumar P.** Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II // *J. Netw. Syst. Manage.* 2018. Vol. 26. P. 463–485. doi:10.1007/s10922-017-9425-0
25. **Kononov A., Kovalenko Y.** On speed scaling scheduling of parallel jobs with preemption // *Proc. Internat. Conf. on Discrete Optimization and Operations Research*. Cham: Springer, 2016. Vol. 9869. P. 309–321. (Ser. Lecture Notes in Computer Science). doi: 10.1007/978-3-319-44914-2_25
26. **Garey M., Johnson D.** Computers and intractability. A Guide to the theory of NP-completeness. San Francisco, CA: W.H. Freeman and Company, 1979. 339 p.
27. **Nesterov Y.** Lectures on convex optimization. Cham: Springer, 2018. 603 p. doi: 10.1007/978-3-319-91578-4
28. **Boyd S., Vandenberghe L.** Convex Optimization. Cambridge: Cambridge Univ. Press, 2009. 714 p.
29. **Shmoys D.B., Stein C., Wein J.** Improved approximation algorithms for shop scheduling problems // *SIAM J. Comp.* 1994. Vol. 23, no. 3. P. 617–632. doi: 10.1137/S009753979222676X
30. **Dempster A.H., Lenstra J.K., Rinnooy Kan A.H.G.** Deterministic and stochastic scheduling // *Proceedings of the NATO Advanced Study Institute Series*. Dordrecht: Springer. 1982. Vol. 84. P. 181–96. doi: 10.1007/978-94-009-7801-0_9
31. **de Werra D.** Graph-theoretical models for preemptive scheduling // *Advances in Project Scheduling*. Amsterdam: Elsevier, 1989. P. 171–185. doi: 10.1016/B978-0-444-87358-3.50012-2
32. **Soper A.J.** A cyclical search for the two machine flow shop and open shop to minimise finishing time // *J. Sched.* Vol. 18, no. 3. P. 311–314. doi: 10.1007/s10951-013-0356-7
33. **Khramova A.P., Chernykh I.** A new algorithm for the two-machine open shop and the polynomial solvability of a scheduling problem with routing // *J. Sched.* 2021. Vol. 24, no. 4. P. 405–412. doi: 10.1007/s10951-021-00694-7
34. **Kunz K.S.** Numerical analysis. NY: McGraw-Hill, 1957. 381 p.
35. **Grotschel M., Lovasz L., Schrijver A.** Geometric algorithms and combinatorial optimizations. Berlin: Springer, 1993. 362 p. doi: 10.1007/978-3-642-78240-4
36. **Edmonds J.** Maximum matching and a polyhedron with 0,1-vertices // *J. Res. Not. Bur. Standords.* 1965. Vol. 698, no. 1–2. P. 125–130. doi: 10.6028/JRES.069B.013

Поступила 6.10.2024

После доработки 20.10.2024

Принята к публикации 28.10.2024

Захарова Юлия Викторовна

канд. физ.-мат. наук

старший науч. сотрудник

Институт математики им. С. Л. Соболева СО РАН, Омский филиал;

Омский государственный университет им. Ф. М. Достоевского

г. Омск

e-mail: yzakharova@ofim.oscsbras.ru

REFERENCES

1. Kubiak W. *A book of open shop scheduling*. Cham, Springer Internat. Publ., 2022, 290 p. doi: 10.1007/978-3-030-91025-9
2. Tanaev V.S., Sotskov Y.N., Strusevich V.A. *Scheduling theory, multi-stage systems*. Springer Dordrecht, 1994. 406 p. Original Russian text was published in Tanaev V. S., Sotskov Y. N., Strusevich V. A. *Teoriya raspisaniy: mnogostadiynnye sistemy*, Moscow, Nauka Publ., 1989, 327 p. ISBN: 5-02-013984-X.
3. Gonzalez T., Sahni S. Open shop scheduling to minimize finish time. *J. ACM*, 1976, vol. 23, no. 4, pp. 665–679. doi: 10.1145/321978.321985
4. Bárány I., Fiala T. Nearly optimum solution of multimachine scheduling problems (in Hungarian). *Sigma Math. Közgazdasági Folyóirat*, 1982, vol. 15, pp. 177–191.
5. Jurisch B., Kubiak W. Two-machine open shops with renewable resources. *Oper. Res.*, 1997, vol. 45, no. 4, pp. 544–552. doi: 10.1287/opre.45.4.544

6. Shabtay D., Kaspi M. Parallel machine scheduling with a convex resource consumption function. *European J. Oper. Res.*, 2006, vol. 173, no. 1, pp. 92–107. doi: 10.1016/j.ejor.2004.12.008
7. Bampis E., Letsios D., Lucarelli G. A note on multiprocessor speed scaling with precedence constraints. In: *Proc. 26th ACM Symposium on Parallelism in algorithms and architectures (SPAA'14)*, 2014, pp. 138–142. doi: 10.1145/2612669.2612672
8. Shabtay D., Kaspi M. Minimizing the makespan in open-shop scheduling problems with a convex resource consumption function. *Naval Research Logistics*, 2006, vol. 53, no. 3, pp. 204–216. doi: 10.1002/nav.20133
9. Adak Z., Arioglu Akan M.Ö., Bulkan S. Multiprocessor open shop problem: literature review and future directions. *J. Comb. Optim.*, 2020, vol. 40, no. 1, pp. 547–569. doi: 10.1007/s10878-020-00591-3
10. Gerards M.E.T., Hurink J.L., Holzenspies P.K.F. A survey of offline algorithms for energy minimization under deadline constraints. *J. Scheduling*, 2016, vol. 19, pp. 3–19. doi: 10.1007/s10951-015-0463-8
11. Li D., Wu J. *Energy-aware scheduling on multiprocessor platforms*. Ser.: Springer Science and Business Media, NY, Springer, 2012, 59 p. doi: 10.1007/978-1-4614-5224-9
12. Kononov A., Kovalenko Y. Approximation algorithms for energy-efficient scheduling of parallel jobs. *J. Scheduling*, 2020, vol. 23, no. 6, pp. 693–709. doi: 10.1007/s10951-020-00653-8
13. Kononov A., Zakharova Y. Speed scaling scheduling of multiprocessor jobs with energy constraint and makespan criterion. *J. Glob. Optim.*, 2022, vol. 83, pp. 539–564. doi: 10.1007/s10898-021-01115-x
14. Kononov A., Zakharova Y. Speed scaling scheduling of multiprocessor jobs with energy constraint and total completion time criterion. *Inter. J. Artif. Intel.*, 2023, vol. 21, no. 2, pp. 109–129.
15. Li K. Energy efficient scheduling of parallel tasks on multiprocessor computers. *J. Supercomput.*, 2020, vol. 60, pp. 223–247. doi: 10.1007/s11227-010-0416-0
16. Zakharova Yu., Sakhno M. Complexity and heuristic algorithms for speed scaling scheduling of parallel jobs with energy constraint. *J. Comput. Appl. Math.*, 2024, art. 116254. Published online: September 16, 2024. doi: 10.1016/j.cam.2024.116254
17. Kong F., Guan N., Deng Q., Yi W. Energy-efficient scheduling for parallel real-time tasks based on level-packing. In: *Proc. 2011 ACM Symposium on Applied Computing (SAC'11)*, 2011, pp. 635–640. doi: 10.1145/1982185.1982326
18. Gao M., He K., Li L., Wang Q., Liu C. A review on energy consumption, energy efficiency and energy saving of metal forming processes from different hierarchies. *Processes*, 2019, vol. 7, no. 6, pp. 357–381. doi: 10.3390/pr7060357
19. Mouzon G., Yildirim M.B., Twomey J. Operational methods for minimization of energy consumption of manufacturing equipment. *Inter. J. Prod. Res.*, 2007, vol. 45, no. 18–19, pp. 4247–4271. doi: 10.1080/00207540701450013
20. Mansouri S.A., Aktas E., Besikci U. Green scheduling of a two-machine flow-shop: trade-off between makespan and energy consumption. *European J. Oper. Res.*, 2016, vol. 248, no. 3, pp. 772–788. doi: 10.1016/j.ejor.2015.08.064
21. He L.J., Cao Y.L., Li W.F., Cao J.J., Zhong L.C. Optimization of energy-efficient open shop scheduling with an adaptive multi-objective differential evolution algorithm. *Appl. Soft Comput.*, 2022, vol. 118, art. 108459. doi: 10.1016/j.asoc.2022.108459
22. Salido M.A., Escamilla J., Giret A., Barber F. A genetic algorithm for energy-efficiency in job-shop scheduling. *The Inter. J. Adv. Manufact. Techn.*, 2016, vol. 85, pp. 1303–1314. doi: 10.1007/s00170-015-7987-0
23. Stewart R., Raith A., Sinnen O. Optimising makespan and energy consumption in task scheduling for parallel systems. *Comp. Oper. Res.*, 2023, vol. 154, art. 106212. doi: 10.1016/j.cor.2023.106212
24. Sathya Sofia A., GaneshKumar P. Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II. *J. Netw. Syst. Manage.*, 2018, vol. 26, pp. 463–485. doi: 10.1007/s10922-017-9425-0
25. Kononov A., Kovalenko Y. On speed scaling scheduling of parallel jobs with preemption. In: *Proc. Inter. Conf. "Discrete optimization and operations research" (DOOR' 2016)*. Lecture notes in computer science. Cham, Springer, 2016, vol. 9869, pp. 309–321. doi: 10.1007/978-3-319-44914-2_25
26. Garey M.R., Johnson D.S. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco, CA, W.H. Freeman and co., 1979, 338 p. ISBN: 0-7167-1045-5. Translated to Russian under the title *Vychislitel'nye mashiny i trudnoreshaemye zadachi*, Moscow, Mir Publ., 1982, 416 p.
27. Nesterov Yu. *Lectures on convex optimization*. Cham, Springer, 2018, vol. 137, 603 p. doi: 10.1007/978-3-319-91578-4
28. Boyd S., Vandenberghe L. *Convex optimization*. UK, Cambridge; NY, Cambridge Univ. Press, 2004, 716 p.

29. Shmoys D.B., Stein C., Wein J. Improved approximation algorithms for shop scheduling problems. *SIAM J. Comp.*, 1994, vol. 23, no. 3, pp. 617–632. doi: 10.1137/S009753979222676X
30. Dempster A.H., Lenstra J.K., Rinnooy Kan A.H.G. Deterministic and stochastic scheduling. In: *Proc. of the NATO Advanced Study Institute Series*, Dordrecht, Springer, 1982, vol. 84, pp. 181–196. doi: 10.1007/978-94-009-7801-0_9
31. de Werra D. Graph-theoretical models for preemptive scheduling. In book: *Advances in project scheduling*. Amsterdam, Elsevier, 1989, pp. 171–185. doi: 10.1016/B978-0-444-87358-3.50012-2
32. Soper A.J. A cyclical search for the two machine flow shop and open shop to minimise finishing time. *J. Sched.*, vol. 18, no. 3, pp. 311–314. doi: 10.1007/s10951-013-0356-7
33. Khramova A.P., Chernykh I. A new algorithm for the two-machine open shop and the polynomial solvability of a scheduling problem with routing. *J. Sched.*, 2021, vol. 24, no. 4, pp. 405–412. doi: 10.1007/s10951-021-00694-7
34. Kunz K.S. *Numerical analysis*. NY, McGraw-Hill, 1957, 381 p.
35. Grötschel M., Lovász L., Schrijver A. *Geometric algorithms and combinatorial optimizations*. Ser.: Algorithms and combinatorics (AC, vol. 2). Berlin, Heidelberg, Springer, 1993, 362 p. doi: 10.1007/978-3-642-78240-4
36. Edmonds J. Maximum matching and a polyhedron with O, l -vertices. *J. Res. Nat. Bur. Stand.*, 1965, vol. 698, no. 1–2, pp. 125–130. doi: 10.6028/JRES.069B.013

Received October 6, 2024

Revised October 20, 2024

Accepted October 28, 2024

Funding Agency: The research was supported by the Russian Science Foundation (project no. 22-71-10015, <https://rscf.ru/en/project/22-71-10015/>).

Yulia Victorovna Zakharova, Cand. Sci. (Phys.-Math.), Sobolev Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences, Omsk Department, Omsk, 644099 Russia; Dostoevsky Omsk State University, Omsk, 644077 Russia, e-mail: yzakharova@ofim.oscsbras.ru.

Cite this article as: Yu. V. Zakharova. Approximation algorithms for Open Shop variations subject to energy consumption. *Trudy Instituta Matematiki i Mekhaniki UrO RAN*, 2024, vol. 30, no. 4, pp. 117–133.