ОРГАНИЗАЦИЯ ОБРАБОТКИ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ СЕТЕВЫХ МОДЕЛЕЙ

А.М. Грузликов, Е.Г. Литуненко, Н.В. Колесов, М.В. Толмачева (Санкт-Петербург)

Введение. Необходимость многоканальной обработки информации возникает на практике в различных прикладных областях. При этом нередко для повышения производительности используется распределенная обработка, удобным инструментом анализа которой являются методы планирования на сетевых моделях ((flow shop)-планирование). При этом разнообразие рассматриваемых задач весьма велико. В настоящей работе обсуждается проблема (flow shop)-планирования, которая традиционно решается при минимизации общего времени выполнения плана, однако ниже в качестве критерия используется среднее по работам время пребывания работы в системе. Оптимальное решение проблемы планирования может быть получено переборными алгоритмами, которые характеризуются экспоненциальной вычислительной сложностью, и в силу этого их применение в целом ряде приложений оказывается невозможным. По этой причине широкое распространение на практике получили субоптимальные алгоритмы [1-8], которые и рассматриваются далее. При этом если в классической постановке модель рассматриваемой системы представляет собой конвейер с линейным графом информационных связей, то в настоящей работе этот граф – ациклический, причем фокусе статьи лежит подход, основанный на использовании так называемых разрешимых классов систем – РКС-алгоритм. Показывается, что базовая идея РКС-алгоритма, предложенного для (flow shop)-планирования с минимизацией общей длительности плана или максимального отклонения от заданных директивных сроков [8], оказывается эффективной и при минимизации среднего по работам времени пребывания работы в системе. Напомним, что это время складывается из двух составляющих - времени ожидания в очереди и времени исполнения

Предварительные сведения и постановка проблемы

Опишем рассматриваемую далее постановку проблемы (flow shop)-планирования. Рассматривается система, которая включает машины, обменивающиеся результатами выполнения операций по их готовности. Предполагается, что рассматриваемое множество операций разбито на независимые группы операций, связанных отношением предшествования (далее работы). В результате планированию подлежат n независимых равноприоритетных работ $t = \{t_j \mid j = \overline{1,n}\}$, обрабатывающих входные требования, находящиеся в очереди. Каждая j-я работа состоит из m операций $t_{j,i}$ длительностью $e_{j,i}$ $i = \overline{1,m}$. Также предполагается, что значения длительностей известно точно. С практической точки зрения это означает, что используются, например, средние значения или верхние границы этих длительностей. Произведенное назначение работ соответствует случаю (flow shop)-системы. В данной работе это означает, что имеется n изоморфизмов $j_j : G_j(S_j,T_j) \circledR\ F(Q,P)$ $j = \overline{1,n}$, где $G_j(S_j,T_j)$ – граф межоперационных связей j-й работы, S_j- множество ребер, T_j – множество вершин (операций), $F(Q,P)$ – граф межмашинных связей, Q – множество ребер, P – множество машин. Все введенные графы являются направленными, ациклическими, содержащими в общем случае не один путь между любыми выделенными вершинами. Графы характеризуются выделенным подмножеством входных вершин и одной выходной вершиной. Далее для рассматриваемой системы будем использовать обозначение $C(F,t)$. Требуется найти наилучший план p вычислений по критерию минимума среднего по работам времени пребывания в системе $\overline{F}(p)$, т.е.

$$J = \min_{p} \overline{F}(p).$$

<u>Основы РКС-алгоритма</u>. Приведем краткое описание теоретических основ предлагаемого и исследуемого ниже алгоритма (flow shop)-планирования - РКС-алгоритма. Этот алгоритм основан на использовании понятия разрешимого класса системы, обсуждаемого ниже. Важным следствием принадлежности системы к разрешимому классу является существование для нее оптимального алгоритма (flow shop)-планирования линейной сложности [8, 9]. В рассматриваемой системе каждая входная машина P_i связана с выходной машиной P_o хотя бы одним путем (последовательностью машин) $p = P_i, P_j, ..., P_o$. Назовем $E(p_j)$ временем выполнения пути p на *j*-й работе. Определим его как сумму времен $e_{j,i}$ выполнения операций *j*-й работы машинами, принадлежащими этому пути. Назовем путь p_j^* критическим для *j*-й работы, если время его выполнения на *j*-й работе является максимальным среди всех остальных путей системы. Очевидно, что для разных работ, выполняемых в одной и той же системе, критические пути могут быть различными.

С использованием нумерации машин вдоль пути p_j выражение для $E(p_j)$ можно записать в виде: $E(p_j) = \sum_{i=1}^{m^*} e_{j,i}$, где m^* – число машин, принадлежащих пути p_j.

Для определения разрешимых классов предварительно введем на множестве машин отношение доминирования «>».

<u>Определение 1</u>. Машина P_q доминирует над машиной P_r ($P_q > P_r$), если $\min_j e_{j,q} \ge \max_j e_{j,r}, (j = \overline{1,n})$.

Общее свойство рассматриваемых далее разрешимых классов систем состоит в следующем: для любой работы, выполняемой в системе, критический путь единственен и проходит по одним и тем же машинам $P_1, P_2, ..., P_{m^*}$. Теперь приведем определяющие свойства для каждого из разрешимых классов.

<u>Определение 2 (класс 1).</u> Множество машин критического пути представляет собой последовательность $P_1 > P_2 > ... > P_{m^*}$, убывающую по отношению доминирования.

<u>Определение 3 (класс 2).</u> Множество машин критического пути представляет собой последовательность $P_1 < P_2 < ... < P_{m^*}$, возрастающую по отношению доминирования.

<u>Определение 4 (класс 3).</u> Множество машин критического пути представляет собой пару соединенных последовательностей:

$$P_1 < P_2 < ... < P_{h^*} > ... > P_{m^*-1} > P_{m^*}, \quad 1 \le h^* \le m^*,$$

первая из которых возрастает, а вторая убывает по отношению доминирования (*h** - номер машины стыковки двух последовательностей).

<u>Определение 5 (класс 4).</u> Множество машин критического пути представляют собой пару соединенных последовательностей

$$P_1 > P_2 > \ldots > P_{h^*} < \ldots < P_{m^*-1} < P_{m^*}, \quad 1 \le h^* \le m^*,$$

первая из которых убывает, а вторая возрастает по отношению доминирования.

Четвертый класс не является в полном смысле разрешимым, поскольку для него неизвестно оптимального алгоритма планирования линейной сложности, а известный алгоритм субоптимален.

Известно [8], что длительности плана p для систем из первого, второго и третьего разрешимых классов определяются выражениями (1), (2) и (3) соответственно:

$$E_1(p) = \sum_{j=1}^{n} e^*_{j,1} + \sum_{i=2}^{m^*} e^*_{n,i}, \tag{1}$$

$$E_2(p) = \sum_{i=1}^{m^*-1} e^*_{1,i} + \sum_{k=1}^{n} e^*_{k,m^*}, \tag{2}$$

$$E_3(p) = \sum_{i=1}^{h^*-1} e^*_{1,i} + \sum_{k=1}^{n} e^*_{k,h^*} + \sum_{i=h^*+1}^{m^*} e^*_{n,i}, \tag{3}$$

где операции, длительности которых отмечены звездочками, выполняются машинами критического пути, n – число работ, m^* – число машин в критическом пути, h^* – номер машины стыковки. Эти выражения можно получить, если записать для некоторой машины критического пути сумму времен работы и простоя. Для первого класса такой машиной будет первая от входа, для второго класса – последняя, для третьего класса – машина стыковки.

Для четвертого класса в рамках рассматриваемого подхода известно [8] лишь выражение для верхней границы длительности плана:

$$E_4(p) \le \bar{E}_4(p) = \sum_{j=1}^{n} (e^*_{j,h^*} + e^*_{j,h^*+1}) + \sum_{i=1}^{h^*-1} e^*_{1,i} + \sum_{i=h^*+2}^{m^*} e^*_{m,i} \tag{4}$$

Алгоритмы планирования по критерию минимума среднего времени пребывания работы в (flow shop)-системах из разрешимых классов

В настоящем разделе предлагаются оптимальные алгоритмы (flow shop)-планирования при использовании в качестве критерия минимума среднего по работам времени пребывания работы в системе.

Теорема 1. Минимальное значение среднего по работам времени пребывания работы в системе $\bar{F}_1(p)$ для системы из класса 1 достигается в плане p, в котором работы упорядочены по неубыванию длительностей первых операций критического пути, т.е.

$$e^*_{1,1} \le e^*_{2,1} \le \ldots \le e^*_{n,1}. \tag{5}$$

Доказательство. Запишем с использованием (1) выражение для среднего по работам времени пребывания работы в системе $\bar{F}_1(p)$ в случае системы из класса 1:

$$\bar{F}_1(p) = \frac{1}{n} \sum_{k=1}^{n} \left[\sum_{j=1}^{k} e^*_{j,1} + \sum_{i=2}^{m^*} e^*_{k,i} \right] = \frac{1}{n} \left[\sum_{k=1}^{n} \sum_{j=1}^{k} e^*_{j,1} + \sum_{k=1}^{n} \sum_{i=2}^{m^*} e^*_{k,i} \right]. \tag{6}$$

Поскольку цель заключается в поиске плана, минимизирующего полученное выражение, то второе слагаемое в скобках можно отбросить и перейти к минимизации выражения

$$\bar{F}'_1(p) = \frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{k} e^*_{j,1}. \tag{7}$$

Действительно, проанализируем отброшенную двойную сумму. Каждое k-е слагаемое ее внешней суммы представляет собой сумму длительностей всех операций, кроме первой, выполняемых в k-й работе на критическом пути. Поскольку такая сумма длительностей присутствует в двойной сумме для каждой работы, ее значение не зависит от упорядоченности работ, а значит, второе слагаемое можно исключить из (6).

Если далее в выражении (7) развернуть внутреннюю сумму и привести подобные члены, то приходим к выражению:

$$\bar{F}_1'(p) = \frac{1}{n}\sum_{k=1}^{n}\left[(n-k+1)e^*_{k,1}\right]. \tag{8}$$

В этом выражении суммируются почленные произведения двух числовых последовательностей $\{n-k+1\}$ и $\{e^*_{k,1}\}$ $k=\overline{1,n}$, причем первая из них убывающая. Известно [10], что сумма (8) будет иметь минимум, если вторая последовательность будет неубывающей. Отсюда следует, что работы должны упорядочиваться по неубыванию длительностей первых задач критического пути.

Теорема 2. Минимальное значение среднего по работам времени пребывания работы в системе $\bar{F}_2(p)$ для системы из класса 2 достигается в плане p, для которого выполняется:

1) работы упорядочены по неубыванию длительностей последних операций критического пути, т.е.

$$e^*_{1,m^*} \le e^*_{2,m^*} \le \ldots \le e^*_{n,m^*}, \tag{9}$$

2) первая работа плана p удовлетворяет условию

$$j^* = \arg\min_{j} \sum_{i=1}^{m^*-1} e^*_{j,i}. \tag{10}$$

Доказательство. Запишем с использованием (2) выражение для среднего по работам времени пребывания работы в системе $\bar{F}_2(p)$ для системы из класса 2:

$$\bar{F}_2(p) = \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=1}^{m^*-1} e^*_{1,i} + \sum_{j=1}^{k} e^*_{j,m^*}\right] = \frac{1}{n}\left[\sum_{k=1}^{n}\sum_{i=1}^{m^*-1} e^*_{1,i} + \sum_{k=1}^{n}\sum_{j=1}^{k} e^*_{j,m^*}\right]. \tag{11}$$

Преобразуем первое слагаемое в этом выражении, учитывая, что его внутренняя сумма не зависит от k. Во втором слагаемом развернем внутреннюю сумму и приведем подобные члены. В результате получаем:

$$\bar{F}_2(p) = \frac{1}{n}\left[n\sum_{i=1}^{m^*-1} e^*_{1,i} + \sum_{k=1}^{n}(n-k+1)e^*_{k,m^*}\right] = \sum_{i=1}^{m^*-1} e^*_{1,i} + \frac{1}{n}\sum_{k=1}^{n}(n-k+1)e^*_{k,m^*}.$$

Отсюда следует, что минимизация первого слагаемого определяется условием (10), а минимизация второго слагаемого (по аналогии с доказательством теоремы 1) - условием (9).

Теорема 3. Минимальное значение среднего по работам времени пребывания работы в системе $\bar{F}_3(p)$ для системы из класса 3 достигается в плане p, для которого выполняется:

1) работы упорядочены по неубыванию длительностей операций стыковки критического пути, т.е.

$$e^*_{1,h^*} \le e^*_{2,h^*} \le \ldots \le e^*_{n,h^*}, \tag{12}$$

2) первая работа плана p удовлетворяет условию

$$j^* = \arg\min_j \sum_{i=1}^{h^*-1} e^*_{j,i} . \qquad (13)$$

Доказательство. Запишем с использованием (3) выражение для среднего по работам времени пребывания работы в системе $\overline{F}_3(p)$ для системы из класса 3:

$$\overline{F}_3(p) = \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=1}^{h^*-1} e^*_{1,i} + \sum_{j=1}^{k} e^*_{j,h^*} + \sum_{i=h^*+1}^{m^*} e^*_{n,i}\right] = \frac{1}{n}\left[\sum_{k=1}^{n}\sum_{i=1}^{h^*-1} e^*_{1,i} + \sum_{k=1}^{n}\sum_{j=1}^{k} e^*_{j,h^*} + \sum_{k=1}^{n}\sum_{i=h^*+1}^{m^*} e^*_{k,i}\right] \qquad (14)$$

Как и при анализе систем из первого класса, в этом выражении третье слагаемое в скобках может быть отброшено, как независящее от упорядоченности заданий. В результате можно перейти к минимизации функции:

$$\overline{F}'_3(p) = \frac{1}{n}\left[\sum_{k=1}^{n}\sum_{i=1}^{h^*-1} e^*_{1,i} + \sum_{k=1}^{m}\sum_{j=1}^{k} e^*_{j,h^*}\right].$$

Оставшиеся слагаемые можно преобразовать в соответствии с использованными в предыдущем утверждении приемами. В результате получаем:

$$\overline{F}'_3(p) = \frac{1}{n}\left[\sum_{k=1}^{n}\sum_{i=1}^{h^*-1} e^*_{1,i} + \sum_{k=1}^{n}\sum_{j=1}^{k} e^*_{j,h^*}\right] = \sum_{i=1}^{h^*-1} e^*_{1,i} + \frac{1}{n}\sum_{k=1}^{n}(n-k+1)e^*_{k,h^*}.$$

Отсюда следует, что минимизация первого слагаемого определяется условием (13), а минимизация второго слагаемого – условием (12).

Теорема 4. Минимальное значение оценки среднего по работам времени пребывания работы в системе $\hat{\overline{F}}_4(p)$ для системы из класса 4 достигается в плане p, для которого выполняется:

1) работы упорядочены по неубыванию суммарных длительностей первых и последних операций критического пути, т.е.

$$(e^*_{1,1}+e^*_{1,m^*}) \le (e^*_{2,1}+e^*_{2,m^*}) \le \ldots \le (e^*_{n,1}+e^*_{n,m^*}), \qquad (15)$$

2) первое задание плана p удовлетворяет условию

$$j^* = \arg\min_j \sum_{i=h^*+1}^{m^*-1} e^*_{j,i} . \qquad (16)$$

Доказательство. Запишем с использованием (4) выражение для оценки среднего по работам времени пребывания работы в системе из класса 4:

$$\hat{\overline{F}}_4(\mathrm{p}) = \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{j=1}^{k}(e^*_{j,1}+e^*_{j,m^*}) + \sum_{i=h^*+1}^{n^*-1} e^*_{1,i} + \sum_{i=2}^{h^*} e^*_{k,i}\right] =$$

$$= \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{j=1}^{k}(e^*_{j,1}+e^*_{j,m^*})\right] + \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=h^*+1}^{n^*-1} e^*_{1,i}\right] + \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=2}^{h^*} e^*_{k,i}\right] \qquad (17)$$

Как и при анализе систем из первого класса, в этом выражении третье слагаемое может быть отброшено, как независящее от упорядоченности работ. В результате можно перейти к минимизации функции:

$$\hat{\overline{F}}'_4(\mathrm{p}) = \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{j=1}^{k}(e^*_{j,1}+e^*_{j,m^*})\right] + \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=h^*+1}^{m^*-1} e^*_{1,i}\right].$$

Оставшиеся слагаемые можно преобразовать в соответствии с использованными ранее приемами. В результате получаем:

$$\hat{\hat{F}}'_4(\pi)=\frac{1}{n}\sum_{k=1}^{n}\left[\sum_{j=1}^{k}(e^*_{j,1}+e^*_{j,m^*})\right]+\frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=h^*+1}^{n^*-1}e^*_{1,i}\right]=\frac{1}{n}\sum_{k=1}^{n}(n-k+1)(e^*_{k,1}+e^*_{j,m^*})+\sum_{i=h^*+1}^{m^*-1}e^*_{1,i}.$$

Отсюда следует, что минимизация первого слагаемого определяется условием (16), а минимизация второго слагаемого – условием (17). Если в конкретном случае условия в теоремах 2, 3 или 4 противоречат друг другу, то лучший из вариантов может быть определен перебором.

Обсудим асимптотическую сложность описанных выше алгоритмов. Для алгоритма первого класса базовой процедурой является сортировка, сложность которой можно оценить как $O(n\log n)$. Для остальных алгоритмов без учета перебора вариантов добавляется процедура выбора минимального линейной сложности. Перебор в данном случае также имеет линейную сложность. В результате итоговую сложность последних трех алгоритмов можно оценить как $O(n\log n)$.

Алгоритм планирования по критерию минимума среднего времени пребывания работы в (flow shop)-системе общего вида.

Очевидно, что на практике для (flow shop)-системы общего вида, описанной в постановке задачи, условия ее принадлежности к тому или иному разрешимому классу чаще всего не выполняются. В частности, могут не совпадать критические пути разных работ, может нарушаться отношение доминирования между машинами, а если оно и выполняется, то может не быть упорядоченных по этому отношению цепочек машин, характерных для разрешимых классов. В результате исчезают гарантии оптимальности описанных выше алгоритмов. В связи с этим предлагается пусть приближенный, но справедливый для любой из рассматриваемых систем рекурсивный алгоритм планирования, выполняемый за число шагов, не большее, чем число заданий. На каждом шаге рекурсии определяется некоторый аналог критического пути, называемый псевдокритическим. Далее используется алгоритм планирования (теоремы 1 – 4), соответствующий тому разрешимому классу, к которому наиболее близка рассматриваемая на данном шаге система $C'=\{P,t'\}$, где t' – множество неразмещенных работ ($t'\subseteq t$).Соответствующий класс определяется на основе геометрического классификационного правила [8], предполагающего на каждом j-м шаге аппроксимацию параболой $y_j(i)$ зависимости средней длительности $\bar{e}_{i,j}$ исполняемых стадий от номера узла i и формирование оценки d_j точности аппроксимации (достоверности классификации), имеющей следующий вид:

$$d_j=\sum_{i=1}^{n}[y_j(i)-\bar{e}_{i,j}]^2.$$

Algorithm 1 РКС-алгоритм

Вход: P - процессоры системы, $\tau' = \{\overline{1,n}\}$ - множество неразмещенных заданий
Выход: π - план заданий

Инициализация переменных:
$\pi = \{\}$
$C' = P, \tau'$
for $k = 1$ to n **do**
  Вычисление средней длительности выполнения задач для каждого процессора: $\check{e}_i = \frac{1}{|\tau'|}\sum_{j\in\tau'} \overline{e}_{j,i}$, где $i = \overline{1,n}$
  Вычисление критического пути p^* на $\{P\}$ с использованием средней длительности выполнения задач $\check{e} = \{\check{e}_i | i = \overline{1,n}\}$
  Определение класса системы и характеристики достоверности: $CL_k, \delta_k = GetClass(p^*, \check{e})$.
  if $k > 1$ **then**
    if $\delta_k > \delta_{k-1}$ **then**
      Упорядочить все оставшиеся задания в соответстсвии с классом CL_{k-1}: $\pi.append(GetJobs(CL_{k-1}, C'))$
      return π
    end if
  end if
  Определить текущее задание: $\tau_k = GetJobs(CL_{k-1}, C'))[0]$
  Добавить задание в конец плана: $\pi.append(\tau_k)$
  Исключить задание τ_k из множества τ'
  $C' = \{P, \tau'\}$
end for
return π

Рис. 1. Иллюстрация алгоритма планирования РКС

При этом выбранная работа занимает первую позицию из интервала свободных позиций формируемого плана. Заметим, что размещение *j*-й работы на *l*-й позиции плана означает, что на каждой машине системы, располагающейся как на псевдокритическом пути, так и за его пределами, операция из *j*-й работы будет находиться на *l*-й позиции. После размещения эта работа исключается из исходного множества, и осуществляется переход к следующей итерации, реализуемой уже для оставшегося множества работ на множестве свободных позиций плана. В случае, если достигнутая на k-м шаге достоверность классификации окажется меньше значения достоверности на предыдущем шаге, то планирование завершается с сохранением упорядоченности $(k - 1)$-го шага. В результате алгоритм последовательно размещает в плане все рассматриваемые работы в направлении от начала плана к его концу.

На рис. 1, 2, 3 приведен псевдокод РКС-алгоритма и его составляющих.

Algorithm 2 GetClass - правило классификации и определение её достоверности

Вход: p^* - критический путь, $\check{e}$ - средняя длительность
Выход: CL - класс, δ - достоверность класса

Аппроксимация параболой зависимости $\check{e}$ от номера процессора вдоль критического пути p^* с использованием метода наименьших квадратов:$y = x^2 * a + x * b + c$.
Вычисление достоверности класса: $\delta = \frac{1}{|p^*|}\sum_{i\in p^*} (y(i) - \check{e})^2$
if $a > 0$ **then**
  if $(-b/2a) < 1$ **then**
    $CL = 1$ (Первый класс)
  else if $(-b/2a) > |p^*|$ **then**
    $CL = 2$ (Второй класс)
  else
    $CL = 3$ (Третий класс)
  end if
else
  if $(-b/2a) < 1$ **then**
    $CL = 2$ (Второй класс)
  else if $(-b/2a) > |p^*|$ **then**
    $CL = 1$ (Первый класс)
  else
    $CL = 4$ (Четвертый класс)
  end if
end if
return CL, δ

Рис. 2. Алгоритм определения правила классификации и её достоверности

Algorithm 3 GetJobs - формирование плана в соответствии с классом системы

Вход: CL - класс, C - система

Выход: π - упорядочивание заданий в соответствии с классом CL

1: π - упорядочивание заданий системы C в соответствии с требованиями оптимального алгоритма соответствующего класса CL (утверждения 1-4).

2: **return** π

Рис. 3. Алгоритм формирования плана в соответствии с классом системы

Результаты моделирования.

Для исследования эффективности РКС-алгоритма при минимизации среднего времени пребывания работы во (flow shop)-системе был осуществлен модельный эксперимент, основанный на случайной генерации примеров. При этом использовалась случайная генерация графов работ и длительностей составляющих их операций.

В целях увеличения достоверности результата было сгенерировано порядка 350000 примеров, представленных тремя группами, а именно, работами, имеющими графы в виде цепочек, деревьев и ациклических графов. Число работ в примерах варьировалось от 10 до 50, при числе машин, равном 20. Длительности операций, измеряемые в условных единицах, формировались как случайные равномерно распределенные величины из интервала [200 - 1000]. Для каждого примера составлялся план на основе предложенного алгоритма. Для вычисления длительностей плана использовалась алгебра max-plus. Далее для получения оценки эффективности алгоритма эту длительность следовало бы сопоставить с минимальной, полученной в результате полного перебора вариантов плана. Однако известно, что при большом числе работ поиск оптимального плана путем перебора становится нереализуемым за приемлемое время. По этой причине в модельном эксперименте использовались оценки Тэйларда [11] и их обобщение на случай ациклических графов. Для всех исследованных трех групп примеров средний проигрыш оптимальному результату оказался порядка 8 %, при этом наихудшие варианты соответствовали уровню 20 %.

Заключение

В настоящей работе были рассмотрены вопросы планирования в (flow shop)-системах. При планировании в качестве критерия был использован минимум среднего времени пребывания работы в системе. Особенностью предложенного РКС-алгоритма является простота. Моделирование продемонстрировала достаточную эффективность предложенного алгоритма.

Дополнительно был исследован вопрос (flow shop)-планирования в условиях блокировок машин. Установлено, что с точки зрения эффективности поставляемых планов РКС-алгоритм занимает промежуточное положение между NEH и CDS, но по быстродействию РКС-алгоритм превосходит алгоритмы NEH и CDS.

Работа поддержана Российским фондом фундаментальных исследований (проект № 19-08-00052).

Литература

1. **Brucker P.** Scheduling Algorithms – Springer, 2007.
2. Minimization of Makespan in Flow Shop Scheduling Using Heuristics // International Conference on Mechanical, Industrial and Energy Engineering 2014, Khulna, Bangladesh
3. **Nawaz M, Enscore Jr. EE, Ham I.** A Heuristic Algorithm for the m-Machine, n-Job Flow-shop Sequencing Problem // Omega – International Journal of Management Science, 1983. № 11: 91-95.

4. **Bocewicz G.** Declarative approach to cyclic steady state space refinement: periodic process scheduling / G. Bocewicz, Z. A. Banaszak // Int. J. Adv. Manuf. Technol. 2013. 67. p.137-155.
5. **Korytkowski P.** Ant colony optimization for job shop scheduling using multi-attribute dispatching rules / P. Korytkowski, S. Rymaszewski, T. Wiśniewski // Int. J. Adv. Manuf. Technol. 2013. 67. p.231–241.
6. **Liu J.W.S.** Real-Time Systems, Prentice Hall, Englewood Cliffs, NJ, 2000. 600p.
7. **Cottet F., Kaiser J., Mammeri Z.** Scheduling in Real-Time Systems. John Wiley & Sons Ltd. 2002.
8. **Грузликов А.М., Колесов Н.В., Скородумов Ю.М., Толмачева М.В.** Планирование заданий в распределенных системах реального времени // ТиСУ, № 2, 2017.
9. **Wang J-B, Xia Z-Q**. Flow shop scheduling with deteriorating jobs under dominating machines. Omega, 34, (2006): p.p. 327-336.
10. **Харди Г.Г., Литтльвуд Дж. Е., Полиа Г.** Неравенства. М.: Изд-во иностранной лит., 1948. 456 с.
11. **Taillard E.** Benchmarks for Basic Scheduling Problems. Europ. J. Operational Research. 1993. V.64. №2. P.278-285.