Санкт-Петербургский государственный университет
Прикладная математика и информатика
Исследование операций и принятие решений в задачах оптимизации, управления и экономики

Поляков Александр Юрьевич

Задачи составления расписаний для одного процессора

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Н. С. Григорьева

Рецензент:
ст. пр. А. С. Зациорский

Санкт-Петербург
2017

Saint Petersburg State University
Applied Mathematics and Computer Science
Operation Research and Decision Making in Optimisation, Control and Economics Problems

Polyakov Aleksandr Yurievich

Single machine scheduling problem

Bachelor's Thesis

Scientific Supervisor:
PhD N. S. Grigoryeva

Reviewer:
Senior Lecturer A. S. Zaciorsky

Saint Petersburg
2017

Оглавление

1. Введение

Рассмотрим задачу составления расписаний на одном процессоре с работами, каждая из которых имеет время выпуска, время обработки и время доставки.

Каждая работа должна начать обрабатываться на процессоре, в момент времени больший или равный времени поступления, и начать процесс доставки сразу же после того, как обработка завершена.

Прерывания во время обработки запрещены.

Требуется составить расписание выполнения работ на одном процессоре с целью минимизировать максимальное время завершение всех работ.

Данная задача явлется NP — трудной, поэтому актуальной задачей является разработка приближенных алгоритмов.

Для начала опишем приближенный алгоритм, представленный Крисом Поттсом в 1980 году, который строит расписание длиной не более 3/2 от длины оптимального расписания, другими словами, этот алгоритм имеет гарантированную оценку точности — $3/2$.

Далее предоставим модифицированный алгоритм Поттса, который строит расписание длиной не более 4/3 от длины оптимального расписания. Также этот алгоритм может обрабатывать ограничения приоритета среди работ.

В заключение приведем результаты тестирования алгоритмов, реализованных на языке "Java" и также некоторые частные случаи заданий, у которых приближенные алгоритмы строят расписания разной длины, но ни один не дает оптимального расписания.

2. Постановка задачи

Пусть n работ должны быть обработаны на одной процессоре. Каждая j-ая работа имеет:

1. время выпуска — r_j , до этого момента работа не может стартовать,

2. время за которое выполняется работа — $p_j > 0$,

3. время доставки — q_j.

Время завершения работы j вычисляется по формуле:

$$C_j = \tau_j + p_j + q_j,$$

где τ_j — время начала обработки работы j.

Требуется составить такое расписание, что:

$$C_{\max} = \max_{j \in 1:n} C_j \longrightarrow \min .$$

Определение: Обратной задачей называется задача, для работ которой времена выпуска меняются местами с временами доставки.

Свойство обратной задачи: $k_1, k_2, \ldots, k_n$ - оптимальный порядок работ для прямой задачи, тогда и только тогда, когда $k_n, k_{n-1}, \ldots, k_1$ оптимальный порядок работ для обратной задачи.

Рассмотрим некоторые сложные случаи нашей задачи:

1. Если все r_j равны между собой.

   Проблема решается "Правилом Джексона": выбирается работа с минимальным временем доставки.

2. Если у нас, время обработки каждой работы равняется единице: $p_j = 1$ и все r_j целые.

   Даная задача решается расширенным правилом Джексона: каждый раз, когда машина свободна, и одна или более работ доступны, выбирается работа с наибольшим времени доставки.

3. Приближенный алгоритм с оценкой точности 3/2

Представим алгоритм Поттса, который генерирует расписание длины не более 3/2 от длины оптимального расписания.

Для начала введем некоторые обозначения. Пусть имеется n работ: $1, 2, \ldots, n$, которые запланированы на одной машине. Каждая j-ая работа имеет время выпуска r_j, до этого момента работа не может стартовать, время за которое выполняется работа $p_j > 0$, и время доставки q_j.

Обозначим за P:

$$P = \sum_{j=1}^{n} p_j.$$

Определим C^* и C_H, как длину оптимального расписания и длину расписания, полученной эвристическим алгоритмом H. Тогда:

$$C^* \geq \max\{P, \max_j r_j, \max_j q_j\}$$

$$C_H = \max_{i \in 1:n}\{\tau_i + p_i + q_i\},$$

где τ_i, время начала обработки j работы.

Определим расширенное правило Джексона.

Расширенное Правило Джексона. Каждый раз, когда машина свободна и одна или несколько работ доступны для обработки, будем планировать доступную работу с наибольшим временем доставки.

Для любого алгоритма H, предположим, что алгоритм порождает упорядоченные работы $j_1, \ldots, j_n$, и что на работе j_c , целевая функция достигает максимума, т. е.

$$C_H = r_{jc} + t_{jc} + q_{jc}.$$

Пусть есть работа j_m, у которой время выпуска в алгоритме H, либо совпадает с заданным выпуском работы, либо существует простой перед этой работой, и время простоя между обработкой работ j_m и j_c равно нулю. Тогда:

$$C_H = r_{j_m} + \sum_{h=m}^{c} p_{j_h} + q_{j_c}. \tag{1}$$

Определение: Последовательность заданий $j_m, \ldots, j_c$ называется критической последовательностью расписания.

Определение: Работа j_c называется критической работой для критической последовательности.

Когда мы используем расширенное правило Джексона, критические последовательности имеют ряд свойств. Начнем с того, что мы видим, что каждая работа j_h в критической последовательности $j_m, ..., j_c$ должна иметь время выпуска $r_{j_h} \geq r_{j_m}$. Таким образом, если каждая работа j_h в критический последовательности удовлетворяет $q_{j_h} \geq q_{j_c}$ то эвристическое расписание является оптимальным.

Определение: Ближайшая работа j_b к работе j_c в критической последовательности, такая что $q_{j_b} < q_{j_c}$ и $r_{j_b} < r_{j_c}$, называется интерференционной для критической последовательности.

Лемма 1. Пусть $j_m, \ldots, j_c$ является критической последовательностью для расписания, генерируемой расширенным правилом Джексона, алгоритмом H, и предположим, что работа j_b это интерференционная работа (тогда $q_{j_h} \geq q_{j_c}$, для всех $h > b$), тогда:

$$C_H - C^* < p_{j_b};$$

$$C_H - C^* \leq q_{j_c}.$$

Доказательство:

1. Докажем, что $C_H - C^* < p_{j_b}$.

   С учетом работ $j_{b+1}, ..., j_c$ нижняя оценка для T^* получается:

   $$C^* \geq \min\{r_{j_{b+1}}, \ldots, r_{j_c}\} + \sum_{h=b+1}^{c} p_{j_h} + \min\{q_{j_{b+1}}, \ldots, q_{j_c}\} =$$

   $$= \min\{r_{j_{b+1}}, \ldots, r_{j_c}\} + \sum_{h=b+1}^{c} p_{j_h} + \min q_{j_c}.$$

   Вычитание из (1) дает:

   $$C_H - C^* \leq r_{j_m} + \sum_{h=m}^{b} p_{j_h} - \min\{r_{j_{b+1}}, \ldots, r_{j_c}\}.$$

   В процессе применения Алгоритма H, работы $j_{b+1}, \ldots, j_c$ не были доступны для обработки во время $r_{j_m} + \sum_{h=m}^{b-1} p_{j_h}$. Что эквивалентно $r_{j_m} + \sum_{h=m}^{b-1} p_{j_h} < \min\{r_{j_{b+1}}, ..., r_{j_c}\}$.
   Получаем что: $C_H - C^* < p_{j_b}$.

2. Докажем, что $C_H - C^* < q_{j_c}$.

   Нижняя оценка для C^*:

$$C^* \geq \min\{r_{j_m}, \ldots, r_{j_c}\} + \sum_{h=m}^{c} p_{j_h} + \min\{q_{j_m}, \ldots, q_{j_c}\}.$$

Так как мы используем расширенное правило Джексона, то: $r_{j_h} \geq r_{j_m}$ и используя то, что время доставки неотрицательно, получаем: $C^* \geq r_{j_m} + \sum_{h=m}^{c} p_{j_h}$.

Вычитание из (1) дает: $C_H - C^* < q_{j_c}$. Что и требовалось доказать.

4. Алгоритм Поттса

1. Строим расписание с помощью алгоритма, который реализует расширенное правило Джексона.

2. Ищем критическую последовательность.

3. Ищем интерференционную работу j_b.

4. Если нашли интерференционную работу, то $r_{j_b} := r_{j_c}$, далее возвращаемся на первый шаг.

5. Если интерференционной работа отсутствует в критической последовательности, то расписание оптимально, конец алгоритма.

6. Повторяем, пока существуют интерференционные работы или n итераций не будут выполнены.

Будем называть этот алгоритм A.

5. Теорема об оценке точности алгоритма Поттса

Приведенный выше алгоритм генерирует расписание C_H с гарантированной оценкой точности: $\frac{C_H}{C^*} < 3/2$.

Доказательство Есть два случая, которые необходимо учитывать. Зафиксируем некоторое оптимальное расписание.

Случай 1. Все ограничения, введенные поочередно согласуются с оптимальным расписанием. Если алгоритм останавливается в определенный момент, потому что нет интерференционных работ, то расписание является оптимальным. В противном случае, алгоритм продолжается в течение n итераций. В этом случае может быть не более одной работы u с $p_u > C^*/2 = \frac{\sum p_i}{2}$. Если u интерференционная работа, то по Лемме 1: $C_H - C^* < p_u$, получаем, что расписание генерируется в алгоритме с длинной $\geq (3/2)C^*$.

Но u может быть интерференционной только $n-1$ раз (последняя работа не может быть интерференционной). Таким образом, некоторые другие работы $v \neq u$ с $p_v \leq C^*/2$ будут интерференционными, и по лемме 1 получаем, что $C_H < (3/2)C^*$.

Случай 2. Пусть в какой-то момент вводится некорректное ограничение приоритета. Рассмотрим первый случай, когда вводится ограничение приоритета, которое нарушает фиксированное оптимальное расписание. Предположим, что $p_{j_b} \geq C^*/2$ и $q_{j_c} \geq C^*/2$, иначе, как было доказано в первом случае: $C_H < (3/2)C^*$.

Так как работа j_b предшествует работе j_c в оптимальном расписании, то:

$$C^* \geq r_{j_b} + p_{j_b} + p_{j_c} + q_{j_c} \geq r_{j_b} + C^*/2 + p_{j_c} + C^*/2 > C^*,$$

получаем противоречее.

Следовательно, либо $p_{j_b} < C^*/2$ или $q_{j_c} < C^*/2$ и получаем, что расписание удовлетворяет $C_H < (3/2)C^*$ по лемме 1.

6. Приближенный алгоритм с оценкой точности 4/3

Рассмотрим случай, когда существуют ограничения предшествования среди работ.

Если существуют ограничения предшествования среди работ, то в начале алгоритма сделаем следующие преобразования:

$$\begin{gathered}\text{Если } r_j < r_i, \text{ то } r_j := r_i \\ \text{Если } q_i < q_j + p_j, \text{ то } q_i := q_j + p_j\end{gathered} \tag{2}$$

Убедимся, что при этих изменениях, расширенное правило Джексона автоматически обеспечивает соблюдение ограничения порядка работ. Обратим внимание на то, что если i должна предшествовать j, тот факт, что $r_i \leq r_j$, гарантирует, что работа j не станет доступной раньше работы i, и тот факт, что $q_i \geq q_j + p_j > q_j$, гарантирует, что если обе работы i и j будут свободны, правильно Джексона выберет первой работу i.

Модифицируем алгоритм Поттса следующим образом: каждый раз, когда находится интерференционная работа b , то $r_b = r_c$, и если работа b должна предшествовать работе j, то $r_j := r_b (= r_c)$.

Будем называть этот модифицированный алгоритм алгоритмом A'.

Модификация алгоритма Поттса основана на выполнении алгоритма A для прямой и обратной задачи. Данный алгоритм дает $4/3$ — приближение. Тем не менее, ситуация, в которой две очень длинные, несравнимые работы существуют, должны рассматриваться отдельно. Ниже приведен модифицированный алгоритм Поттса.

7. Модифицированный алгоритм Поттса

Модифицированный алгоритм Потта:

1. Проверяем, существуют ли работы u и v с $p_u, p_v > P/3$,
2. Если таких работ нет, то применяем алгоритмы A для прямой и обратной задачи.
   - 2.1. Сравниваем полученные расписания.
   - 2.2. Возвращаем лучшее расписание.
   - 2.3. Конец алгоритма.
3. Если существуют две длинные работы u и v с $p_u,\ p_v > P/3$, и известно, что одна работа должна начаться раньше другой, то:
   - 3.1. Для работ u и v применим соответствующие преобразования, как показано в формуле (2)
   - 3.2. Применяем алгоритм A' $2n-1$ раз.
   - 3.3. Конец алгоритма.
4. Если существуют две длинные работы u и v с $p_u,\ p_v > P/3$ и, отсутствуют ограничения предшествования для работ u и v, то:
   - 4.1. Вводим ограничение $u < v$ и применим алгоритм A' $2n-1$ раз.
   - 4.2. Вводим ограничение $u > v$ и применим алгоритм A' $2n-1$ раз.
   - 4.3. Возвращаем лучшее расписание.
   - 4.4. Конец алгоритма.
5. Конец Алгоритма.

8. Результаты

Приближенные алгоритмы реализовались на языке "Java". Среда разработки — IntelliJ IDEA 2017.1. Все полученные результаты записываются в текстовый документ.

Тест 1

Входные данные:

1) Количество сгенерированных тестов — 10000.
2) Количество работ, установленных в расписании — 5.
3) Диапазон времени выпуска, обработки и доставки работы — 20.

Количество заданий, где $C^* = C_{\text{Джек.}}$	Количество заданий, где $C^* = C_{\text{Поттса}}$	Количество заданий, где $C^* = C_{\text{Мод. Поттса}}$	Количество заданий, где $C^* \neq C_{\text{Джек.}} \neq$ $\neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$
8344	9805	9994	1

Количество заданий, где $\frac{C_{\text{Джек.}}}{C^*} > 1.8$	Количество заданий, где $\frac{C_{\text{Поттса}}}{C^*} > 1.4$	Количество заданий, где $\frac{C_{\text{Мод.Поттса}}}{C^*} > 1.2$
0	0	0

Пример задания, где $C^* \neq C_{\text{Джек.}} \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$:

№	1	2	3	4	5
r_j	0	2	3	12	10
p_j	7	2	20	5	6
q_j	10	7	13	19	6

1. Длина расписания, полученная расширенным правилом Джексона: $C_{\text{Джек.}} = 51$.

   Расписание, полученное расширенным правилом Джексона:

1	3	4	2	5

2. Длина расписания, полученная алгоритмом Поттса: $C_{\text{Поттса}} = 50$.

   Расписание, полученное алгоритмом Поттса:

1	2	4	3	5

3. Длина расписания, полученная модифицированным алгоритмом Поттса: $C_{\text{Мод.Поттса}} = 49$.

   Расписание, полученное модифицированным алгоритмом Поттса:

3	4	1	2	5

4. Длина оптимального расписания: $C^* = 48$.

Тест 2

Входные данные:

1) Количество сгенерированных тестов —10000.

2) Количество работ, установленных в расписании — 5.

3) Диапазон времени выпуска, обработки и доставки работы — 50.

Количество заданий, где $C^* = C_{\text{Джек.}}$	Количество заданий, где $C^* = C_{\text{Поттса}}$	Количество заданий, где $C^* = C_{\text{Мод. Поттса}}$	Количество заданий, где $C^* \neq C_{\text{Джек.}} \neq \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$
7789	9702	9984	3

Количество заданий, где $\frac{C_{\text{Джек.}}}{C^*} > 1.8$	Количество заданий, где $\frac{C_{\text{Поттса}}}{C^*} > 1.4$	Количество заданий, где $\frac{C_{\text{Мод.Поттса}}}{C^*} > 1.2$
0	0	0

Пример задания, где $C^* \neq C_{\text{Джек.}} \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$:

№	1	2	3	4	5
r_j	0	11	42	18	22
p_j	41	13	45	50	11
q_j	4	4	50	46	36

1. Длина расписания, полученная расширенным правилом Джексона: $C_{\text{Джек.}} = 186$.

   Расписание, полученное расширенным правилом Джексона:

1	4	3	5	2

2. Длина расписания, полученная алгоритмом Поттса: $C_{\text{Поттса}} = 184$.

   Расписание, полученное алгоритмом Поттса:

1	3	4	5	2

3. Длина расписания, полученная модифицированным алгоритмом Поттса: $C_{\text{Мод.Поттса}} = 182$.

   Расписание, полученное модифицированным алгоритмом Поттса:

4	5	3	1	2

4. Длина оптимального расписания: $C^* = 175$.

Тест 3

Входные данные:

1) Количество сгенерированных тестов — 10000.

2) Количество работ, установленных в расписании — 5.

3) Диапазон времени выпуска, обработки и доставки работы — 75.

Количество заданий, где $C^* = C_{\text{Джек.}}$	Количество заданий, где $C^* = C_{\text{Поттса}}$	Количество заданий, где $C^* = C_{\text{Мод. Поттса}}$	Количество заданий, где $C^* \neq C_{\text{Джек.}} \neq \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$
7609	9693	9981	4

Количество заданий, где $\frac{C_{\text{Джек.}}}{C^*} > 1.8$	Количество заданий, где $\frac{C_{\text{Поттса}}}{C^*} > 1.4$	Количество заданий, где $\frac{C_{\text{Мод.Поттса}}}{C^*} > 1.2$
0	0	0

Пример задания, где $C^* \neq C_{\text{Джек.}} \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$:

№	1	2	3	4	5
r_j	0	46	41	61	4
p_j	44	23	37	66	56
q_j	2	41	5	75	43

1. Длина расписания, полученная расширенным правилом Джексона: $C_{\text{Джек.}} = 241$.

   Расписание, полученное расширенным правилом Джексона:

1	5	4	2	3

2. Длина расписания, полученная алгоритмом Поттса: $C_{\text{Поттса}} = 234$.

   Расписание, полученное алгоритмом Поттса:

1	2	4	5	3

3. Длина расписания, полученная модифицированным алгоритмом Поттса: $C_{\text{Мод.Поттса}} = 233$.

   Расписание, полученное модифицированным алгоритмом Поттса:

5	4	3	2	1

4. Длина оптимального расписания: $C^* = 232$.

Тест 4

Входные данные:

1) Количество сгенерированных тестов — 10000.

2) Количество работ, установленных в расписании — 8.

3) Диапазон времени выпуска, обработки и доставки работы — 50.

Количество заданий, где $C^* = C_{\text{Джек.}}$	Количество заданий, где $C^* = C_{\text{Поттса}}$	Количество заданий, где $C^* = C_{\text{Мод. Поттса}}$	Количество заданий, где $C^* \neq C_{\text{Джек.}} \neq \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$
9136	9953	9997	0

Количество заданий, где $\frac{C_{\text{Джек.}}}{C^*} > 1.8$	Количество заданий, где $\frac{C_{\text{Поттса}}}{C^*} > 1.4$	Количество заданий, где $\frac{C_{\text{Мод.Поттса}}}{C^*} > 1.2$
0	0	0

Тест 5

Входные данные:

1) Количество сгенерированных тестов — 10000.

2) Количество работ, установленных в расписании — 8.

3) Диапазон времени выпуска, обработки и доставки работы — 75.

Количество заданий, где $C^* = C_{\text{Джек.}}$	Количество заданий, где $C^* = C_{\text{Поттса}}$	Количество заданий, где $C^* = C_{\text{Мод. Поттса}}$	Количество заданий, где $C^* \neq C_{\text{Джек.}} \neq \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$
9073	9928	9990	2

Количество заданий, где $\frac{C_{\text{Джек.}}}{C^*} > 1.8$	Количество заданий, где $\frac{C_{\text{Поттса}}}{C^*} > 1.4$	Количество заданий, где $\frac{C_{\text{Мод.Поттса}}}{C^*} > 1.2$
0	0	0

Пример задания, где $C^* \neq C_{\text{Джек.}} \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$:

№	1	2	3	4	5	6	7	8
r_j	0	18	38	8	12	20	37	60
p_j	7	19	7	38	61	8	2	26
q_j	8	23	63	13	42	32	22	65

1. Длина расписания, полученная расширенным правилом Джексона: $C_{\text{Джек.}} = 205$.

   Расписание, полученное расширенным правилом Джексона:

1	4	3	5	8	6	2	7

2. Длина расписания, полученная алгоритмом Поттса: $C_{\text{Поттса}} = 191$.

   Расписание, полученное алгоритмом Поттса:

1	4	3	6	8	5	2	7

3. Длина расписания, полученная модифицированным алгоритмом Поттса: $C_{\text{Мод.Поттса}} = 188$.

   Расписание, полученное модифицированным алгоритмом Поттса:

5	2	8	3	6	7	4	1

4. Длина оптимального расписания: $C^* = 186$.

Тест 6

Входные данные:

1) Количество сгенерированных тестов — 10000.

2) Количество работ, установленных в расписании — 8.

3) Диапазон времени выпуска, обработки и доставки работы — 95.

Количество заданий, где $C^* = C_{\text{Джек.}}$	Количество заданий, где $C^* = C_{\text{Поттса}}$	Количество заданий, где $C^* = C_{\text{Мод. Поттса}}$	Количество заданий, где $C^* \neq C_{\text{Джек.}} \neq \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$
8984	9920	9993	2

Количество заданий, где $\frac{C_{\text{Джек.}}}{C^*} > 1.8$	Количество заданий, где $\frac{C_{\text{Поттса}}}{C^*} > 1.4$	Количество заданий, где $\frac{C_{\text{Мод.Поттса}}}{C^*} > 1.2$
0	0	0

Пример задания, где $C^* \neq C_{\text{Джек.}} \neq C_{\text{Поттса}} \neq C_{\text{Мод. Поттса}}$:

№	1	2	3	4	5	6	7	8
r_j	0	5	62	12	61	82	94	9
p_j	42	39	51	11	31	8	20	15
q_j	35	39	83	51	6	68	94	19

1. Длина расписания, полученная расширенным правилом Джексона: $C_{\text{Джек.}} = 257$.

   Расписание, полученное расширенным правилом Джексона:

1	4	2	3	7	6	8	5

2. Длина расписания, полученная алгоритмом Поттса: $C_{\text{Поттса}} = 248$.

   Расписание, полученное алгоритмом Поттса:

1	4	2	7	3	6	8	5

3. Длина расписания, полученная модифицированным алгоритмом Поттса: $C_{\text{Мод.Поттса}} = 243$.

   Расписание, полученное модифицированным алгоритмом Поттса:

1	3	6	7	2	4	8	5

4. Длина оптимального расписания: $C^* = 230$.

Вывод, исходя из полученных результатов:

1. При увеличении диапазона времени выпуска, обработки и доставки работ количество оптимальных расписаний, построенных с помощью расширенного правила Джексона, алгоритм Поттса, уменьшается.

2. Если в задании небольшое количество работ, то при увеличении диапазона времени выпуска, обработки и доставки работ, количество расписаний, у которых алгоритмы строят разную длину расписаний, но не дают оптимальное расписание, увеличивается.

9. Заключение

В ходе работы были изучены приближенные алгоритмы составления расписаний на одном процессоре. Для реализации приближенных алгоритмов написана программа, в которой присутствует:

1. Генератор данных с возможностью выбора количества работ в расписании и диапазона их времени выпуска, обработки и доставки.
2. Возможность ввода данных с клавиатуры.
3. Все входные данные и результаты записываются в текстовый файл.

Данную программу можно скачать по ссылке:

`https://yadi.sk/d/pstEP6aB3JRUdY/2017`.

Был проведен вычислительный эксперимент, который позволил найти некоторые частные случаи заданий, когда приближенные алгоритмы строят разной длины расписания, но ни один не дает оптимального расписания.

Список литературы

1. Leslie, A. H. Operations Research Center / B. S. David, A. H. Leslie // *Jackson's Rule for One-Machine Scheduling*: Making a Good Heuristic Better. —1988.—P. 1-10.
2. Potts, C. N. (1980) Analysis of a Heuristic for One Machine Sequencing with Release Dates and Delivery Times, Operations Research, Vol. 28, No. 6, 1436-41.
3. Garey, M. R., D. S. Johnson, B. B. Simons, and R. E. Tarjan. (1981) Scheduling unit-time tasks with arbitrary release times and deadlines, SIAM J. Comput., Vol. 10, No. 2, 256-69.
4. E. Chinos, Production & Manufacturing / N. Vakhania, E. Chinos // Adjusting scheduling model with release and due dates in production planning. —.2017.—P. 1-8.