Оглавление

Введение

В теории расписаний исследуются вопросы оптимального распределения и упорядочения конечного множества требований, обслуживаемых детерминированными системами с одним или несколькими ресурсами, при различных предположениях относительно характера их обслуживания. Под ресурсами могут пониматься машины, станки, учебные помещения, приборы и т. п., под требованиями — выполняемые работы, программы, школьные занятия, грузоперевозки и многое другое. Для определенности, в дальнейшем ресурсы будем называть машинами, а требования — работами. Если просмотреть многочисленные статьи, обзоры и монографии, посвященные задачам теории расписаний, то можно обнаружить огромное количество различных задач, рассматриваемых в рамках этой теории. Поэтому необходимо описать, какие именно задачи будут рассматриваться в диссертации и какие ограничения на задачи при этом предполагаются.

Первое из ограничений связано с доступностью информации о параметрах задачи. Далее речь пойдет только о детерминированных задачах теории расписаний. В них предполагается, что вся исходная информация об индивидуальной задаче определена и известна заранее.

Остальные ограничения связаны с самими машинами и работами. Обычно каждой работе сопоставляется множество машин, каждая из которых может или должна выполнить эту работу. Если каждая работа может быть выполнена любой машиной, то обслуживающая система называется однооперационной (одностадийной) с одной или несколькими параллельными машинами.

В многооперационных (многостадийных) системах работа состоит из набора операций. В дальнейшем будут рассматриваться следующие типы многооперационных задач. Если каждая работа состоит из последовательных операций и порядок выполнения операций одинаков для всех работ и определяется последовательностью машин, то такая обслуживающая система называется системой поточного типа. В случае, если порядок выполнения операций не фиксирован и может быть различным для каждой работы, то такая обслуживающая система называется системой открытого типа. Подробно с различными постановками и результатами для задач с одной или несколькими параллельными машинами можно ознакомиться в монографии [?], а для задач с многооперационными системами в монографии [?].

Будем предполагать, что в каждый момент времени на любой машине выполняется не более одной работы и в каждый момент времени любая работа обслуживается не более одной машиной.

В зависимости от характера обслуживающей системы процесс выполнения каждой работы или ее операции либо должен протекать непрерывно, либо могут допускаться прерывания с последующим дообслуживанием. В рассматриваемой работе будем предполагать, что прерывания в выполнении операций недопустимы.

Процесс функционирования обслуживающей системы может быть описан путем задания расписания, то есть некоторой совокупности указаний относительно того, какие работы или операции выполняются машинами в каждый момент времени. Например, при сделанных выше предположениях расписание можно рассматривать как совокупность двух массивов чисел, которые указывают каждой операции момент начала ее выполнения и номер машины, на которой она выполняется. Другие различные способы задания расписаний описаны в монографиях [?], [?], [?]. При задании расписания должны соблюдаться все условия и ограничения, вытекающие из постановки рассматриваемой задачи, то есть расписание должно быть допустимым. Следует отметить, что построение допустимого расписания и даже выяснения того факта, существует ли оно вообще, является зачастую нетривиальной задачей.

Если существует несколько допустимых расписаний, то возникает стремление построить наилучшее из них. В связи с этим появляется вопрос оценки качества расписаний. В теории расписаний наибольшее распространение получил следующий способ оценки. Каждое допустимое расписание однозначно определяет вектор моментов завершения выполнения работ. По этому вектору определяется некоторая действительная неубывающая по каждой координате функция и качество расписания оценивается значением этой функции. Расписание, которому соответствует наименьшее значение функции среди всех допустимых расписаний, называется оптимальным. В качестве критериев оптимальности наиболее часто рассматриваются функции, определяющие следующие понятия: длина расписания, сумма моментов завершения работ, взвешенная сумма моментов завершения работ, число запаздывающих работ, максимальное запаздывание работы относительно директивного срока и некоторые другие.

Обычно в задачах теории расписаний предполагается, что время выполнения (длительность) каждой работы или ее операции на каждой машине фиксировано. Однако на практике оно часто зависит от времени начала выполнения. В частности, в [?] указывается, что существует множество факторов, которые могут влиять на длительность работы. Например, если работы проводятся при ухудшении погодных условий или с наступлением темноты, если работа заключается в очистке загрязненной местности или в ремонте работающих механизмов. С помощью таких задач можно моделировать распределение ресурсов для тушения лесных пожаров, процесс борьбы с эпидемиями и многое другое. Изучению таких задач представляет как теоретический, так и практический интерес.

Первый шаг в этом направлении был сделан Мельниковым О.Н. и Шафранским Я.М. [?]. Они рассмотрели задачи, в которых длительность каждой работы J_i в момент времени t задается функцией,

$$p_i = a_i + \Upsilon(t),$$

где $a_i \geq 0$ и $\Upsilon(t) \geq 0$ — функция, определенная для всех $t \geq 0$. В случае одной машины в [?] установлено, что если $\Upsilon(t)$ монотонно возрастает при $t \geq 0$, то перестановка работ по неубыванию величин a_i, минимизирует длину расписания; если $\Upsilon(t)$ — монотонно убывающая дифференцируемая функция и $|\frac{d\Upsilon(t)}{dt}| \leq 1$ при $t \geq 0$, то перестановка работ по невозрастанию величин a_i минимизирует длину расписания. В этой же работе изучались задачи построения оптимального по длине расписания для системы потокового типа (flow shop) на m машинах. Авторы рассмотрели две возможные ситуации. В первом случае длительность операции зависит от времени начала выполнения работы. Во втором случае длительность операции зависит от времени начала выполнения самой операции. В первом случае задача формулировалась следующим образом. Пусть длительность операции i работы J_j

определяется как

$$p_{i,j} = a_{i,j} + \Upsilon(s_j),$$

где $a_{i,j}$ — фиксированные числа, $\min\limits_{i,j} a_{i,j} = a > 0$ для всех $1 \le i \le m-1$, $1 \le j \le n$ и $\max\limits_{i,j} a_{i,j} = A$ для всех $2 \le i \le m$, $1 \le j \le n$, s_j — начало выполнения работы J_j первой машиной, $\Upsilon(t) \ge 0$ — дифференцируемая функция и $|\dfrac{d\Upsilon(t)}{dt}| \ge \lambda > 0$ при $t \ge 0$. В [?] построен полиномиальный алгоритм, находящий точное решение задачи при условии, что число работ $n \ge N$, где $N(a, A, \lambda)$ — конечное число, независящее от функции $\Upsilon(t)$. Во втором случае рассматривалась задача на двух машинах. Длительность операции i работы J_j определяется как $p_{i,j} = a_{i,j} + \Upsilon(s_{i,j})$, где $\Upsilon(t) \ge 0$ — монотонно возрастающая функция при $t \ge 0$, $a_{1,i} \ge a_{2,i} > 0$. Для этой задачи в [?] построен алгоритм с абсолютной оценкой точности.

В [?],[?],[?],[?] рассматривалась задача построения оптимального по длине расписания n работ на одной машине с длительностями, равными

$$p_i = a_i s_i + b_i.$$

В этих работах показано, что оптимальное расписание строится упорядочением работ по неубыванию величин b_i/a_i. Кроме того в [?] рассматривались задачи, когда время выполнения работы J_i описывается полиномом общего вида

$$a_{in}x^n + a_{in+1}x^{n+1} + \ldots + a_{i1}x + a_{i0}.$$

Для этих моделей авторами были предложены эвристические алгоритмы.

В [?] Мошеов Г. рассмотрел задачи на одной машине с различными критериями оптимальности, в которых длительность работы J_i прямо пропорциональна моменту начала ее выполнения и равна

$$p_i = \alpha_i s_i.$$

Автор показал, что если машина выполняет работы без простоя, то длина расписания в этих задачах не зависит от перестановки работ, и, следовательно, любая перестановка является оптимальной. В [?] Мошеов Г. рассмотрел следующие критерии: минимизация максимального запаздывания, минимизация числа запаздывающих работ и минимизация взвешенной суммы времен окончания работ. Для всех критериев в [?] построены полиномиальные алгоритмы нахождения оптимального расписания. Там же отмечено, что полиномиально-разрешимые случаи для задач на одной машине с постоянными длительностями остаются полиномиально-разрешимыми и в рассматриваемом случае.

Дальнейшее обобщение задач с длительностями работ, пропорциональными функции от моментов начала их выполнения, рассматриваются в первой и третьей главах диссертации. Основные результаты для этих задач, полученные автором диссертации, опубликованы в [?], [?], [?].

В [?] Мошеов Г. рассмотрел следующую задачу. Множество из n работ должно быть выполнено на одной машине. Все работы готовы к выполнению в момент времени 0. Длительность работы J_i задается формулой $p_i = 1 + \alpha_i s_i$. Требуется минимизировать суммарное время завершения всех работ. Мошеов Г. показал, что среди оптимальных расписаний существует расписание, обладающее следующими свойствами. Первой выполняется работа с наибольшим α_i. Работа с наименьшим α_i не выполняется ни второй, ни последней. Работы в оптимальной последовательности упорядочены в порядке убывания, если они выполняются до работы с наименьшим α_i, и упорядочены в порядке возрастания, если они выполняются после работы с наименьшим α_i.

В [?] Куннатхур А.С. и Гупта С.К. рассмотрели следующую задачу. Множество из n работ должно быть выполнено на одной машине. Все работы готовы к выполнению в момент времени 0. Для каждой работы J_i определено директивное время на начало выполнения. Длительность каждой работы определяется суммой фиксированной части и некоторой штрафной добавки за нарушение директивного срока. Величина штрафной добавки прямо пропорциональна разности между моментом начала выполнения работы и директивным сроком. Требуется построить расписание минимальной длины. В [?] для решения этой задачи авторы предложили точные экспонециальные алгоритмы, основанные на методе ветвей и границ и динамическом программировании, а также ряд эвристических полиномиальных алгоритмов. В дальнейшем эта задача независимо рассматривалась в [?], [?]. В [?] автором диссертации показано, что эта задача является NP-трудной в сильном смысле. Там же рассмотрены различные подзадачи данной задачи. В частности, устанавлена ее NP-трудность и описан псевдополиномиальный алгоритм для случая, когда все работы имеют общий директивный срок. Результаты этой статьи изложены во второй главе данной диссертации. NP-трудность и псевдополиномиальный алгоритм для задачи с общим директивным сроком также получены и в [?]. Кроме того в [?] рассмотрена задача, в которой рост штрафной добавки ограничен другим директивным сроком также одинаковым для всех работ. Для этой задачи в [?] построен псевдополиномиальный алгоритм. В [?] для задач, рассмотренных в [?], построены полные полиномиальные аппроксимационные схемы.

Модели, в которых длительность работы является убывающей линейной или кусочно-линейной функцией, рассматривались в [?], [?], [?]. В [?] изучалась задача нахождения допустимого относительно директивных сроков расписания, когда длительность работы J_i задается формулой $p_i = -a_i s_i + b_i$, где $0 < b_i < 1$ и $b_i d_i < a_i \leq d_i$. Авторы установили, что задача является NP-полной уже в том случае когда у работ существует только два различных директивных срока. Этот результат влечет NP-трудность и соответствующих оптимизационных задач, в частности, задачи минимизации числа запаздывающих работ. В [?] был построен точный полиномиальный алгоритм решения этой задачи в предположении, что все работы имеют общий директивный срок.

Диссертация посвящена исследованию комбинаторной сложности и построению эффективных алгоритмов для специальных классов параметрических задач теории расписаний, в которых длительности работ зависят от момента начала их выполнения.

Первая глава посвящена исследованию сложности и построению эффективных алгоритмов для задач на одной машине с длительностями работ, пропорциональными функции от времени. Время выполнения работы J_i равно $p_i = \alpha_i \Upsilon(s_i)$, где $\Upsilon(t)$ — некоторая функция от времени, одинаковая для всех работ.

В разделе 1.1 формулируются определения основных понятий комбинаторной сложности задач дискретной оптимизации и задач теории расписаний, приведены эталонные NP-трудные задачи и введены обозначения для рассматриваемых задач теории расписаний.

В разделе 1.2 рассматриваются задачи, в которых функция $\Upsilon(t)$ имеет вид $\Upsilon(t) = at + b$. Устанавлено, что если работы выполняются без простоев, то суммарное время обработки не зависит от последовательности выполнения этих работ. Использование этого факта позволяет построить точные полиномиальные алгоритмы для задач со следующими критериями оптимальности:

- минимизация максимального запаздывания;
- минимизация взвешенного суммарного времени завершения выполнения всех работ;
- минимизация числа запаздывающих работ;
- минимизация максимального временного смещения.

В разделе 1.3 устанавливается полиномиальная разрешимость задачи минимизации длины расписания, когда функция $\Upsilon(t)$ является выпуклой или вогнутой. Далее, рассматриваются задачи с критериями минимизации максимального временного смещения и минимизации взвешенного суммарного времени завершения выполнения всех работ. Для задач с каждым из критериев выделяются полиномиально разрешимые классы подзадач, зависящие от вида функции $\Upsilon(t)$. В частности, устанавливается полиномиальная разрешимость задачи минимизация суммарного времени завершения выполнения всех работ, когда функция является выпуклой.

В разделе 1.4 исследуются задачи минимизации длины расписания, когда длительности работ пропорциональны ступенчатой функции. Для всех рассмотренных в этом разделе задач установлена их NP-трудность и построены "псевдополиномиальные"алгоритмы.

Вторая глава посвящена исследованию сложности и построению эффективных алгоритмов для задач на одной машине с длительностями работ, линейно или кусочно-линейно зависящими от времени.

В разделе 2.1 устанавливается NP-трудность задачи минимизации максимального временного смещения на одной машине с линейными функциями длительности.

В разделе 2.2 рассматриваются задачи минимизации длины расписания, в которых длительность работ определяется суммой фиксированной части и некоторой штрафной добавки. Каждая работа J_I имеет директивный срок d_i на начало выполнения. Штраф равен нулю, если работа начинает выполняться до этого директивного срока, и прямо пропорционален количеству времени, прошедшему с момента наступления директивного срока, то есть длительность работы вычисляется по формуле $p_i = a_i(s_i - d_i)^+ + b_i$. Устанавливается NP-трудность в сильном смысле данной задачи.

В разделах 2.3 и 2.4 изучается задача из предыдущего раздела с условием, что все работы имеют общий директивный срок на начало выполнения ($d_i = D$). В разделе 2.3 строится псевдополиномиальный алгоритм для этой задачи, а в разделе 2.4 устанавливается ее NP-трудность. Там же рассматриваются три подзадачи данной задачи.

Задача ($PD \geq$). Решить задачу с общим директивным сроком при условии, что $b_i/a_i \geq b_j/a_j$ при всех $J_i, J_j \in J$ таких, что $b_i > b_j$ и $a_i > 0$, $a_j > 0$.

Задача ($PD \leq$). Решить задачу с общим директивным сроком при условии, что $b_i/a_i \leq b_j/a_j$ при всех $J_i, J_j \in J$ таких, что $b_i > b_j$ и $a_i > 0$, $a_j > 0$.

Задача PDP. Решить задачу с общим директивным сроком при условии, что существует промежуток времени $[t_0, t_1], t_0 < t_1$, когда машина находится на профилактике, т.е. не обслуживает ни одной работы, и $D \in [t_0, t_1]$.

В разделе 2.3 установлена полиномиальная разрешимость задачи

($PD \geq$) и построены "псевдополиномиальные"алгоритмы для остальных задач. В разделе 2.4 устанавлена NP-трудность двух последних задач.

Третья глава посвящена исследованию сложности и построению эффективных алгоритмов для задач на нескольких параллельных машинах или в многооперационной системе обслуживания.

В разделе 3.1 приводятся основные определения и обозначения для задач теории расписаний при наличии нескольких машин.

В разделе 3.2 формулируется вспомогательная задача 4-ПРОИЗВЕДЕНИЕ. Устанавливается ее принадлежность к классу NP-полных задач. В дальнейшем этот результат используется для доказательства NP-трудности некоторых задач теории расписаний.

В разделе 3.3 изучаются задачи теории расписаний, когда каждая работа может быть выполнена на одной из двух идентичных машин. Рассматриваются критерии минимизации длины расписания и минимизации суммарного времени завершения выполнения всех работ. Для обоих критериев устанавливается NP-трудность построения оптимального расписания.

В разделе 3.4 рассматриваются задачи построения расписания в системах поточного и открытого типа. Во всех задачах длительность операции прямо пропорциональна времени начала ее выполнения. Первой рассматривается задача минимизации длины расписания в системе поточного типа из трех машин. Доказано, что если $P \neq NP$, то для ее решения не существует полиномиального алгоритма, который гарантировано находит расписание длины в фиксированное число раз хуже длины оптимального расписания. Далее в разделе 3.4 рассматриваются следующие задачи:

- минимизация максимального временного смещения в системе поточного типа из двух машин;
- минимизация длины расписания в системе открытого типа из трех машин;
- минимизация максимального временного смещения в системе открытого типа из двух машин.

Показывается NP-трудность рассматриваемых задач.

В разделе 3.5 строятся точные полиномиальные алгоритмы для задач минимизации длины расписания в системах поточного и открытого типа из двух машин.

Основные результаты диссертации опубликованы в работах [3 - 7,] [34 - 37] и докладывались на Втором сибирском конгрессе по прикладной и индустриальной математике (г. Новосибирск, Россия, 1996), на Симпозиуме по исследованию операций 1996 (г. Брауншвейг, Германия, 1996), на Третьем рабочем семинаре по моделям и алгоритмам в планировании и теории расписаний (г.Кембридж, Великобритания, 1997), на Международной конференции проблемы оптимизации и экономические приложения (г. Омск, Россия 1997), на Международном симпозиуме по комбинаторной оптимизации (г. Брюссель, Бельгия, 1998), на семинарах в Институте математики СО РАН.

Непринадлежащие автору результаты будем формулировать в виде "утверждений".

Глава 1

Задачи теории расписаний на одной машине с длительностями работ, пропорциональными произвольной функции

1.1 Основные обозначения и определения

Данный раздел содержит основные определения понятий комбинаторной сложности задач дискретной оптимизации и задач теории расписаний. В конце раздела сформулированы некоторые специальные определения и утверждения, а также приведены эталонные NP-трудные задачи, которые будут использоваться для анализа сложности задач теории расписаний.

Все определения, связанные с основными понятиями задач дискретной оптимизации, взяты из [?], и уровень формализации соответствует целям настоящего исследования.

Под **задачей** будем понимать некоторый общий вопрос, на который следует дать ответ. Обычно задача содержит несколько **параметров,** или свободных переменных, конкретные значения которых не определены. Задача Π определяется следующей информацией:

(1) общим списком всех ее параметров,

(2) формулировкой тех требований, которым должен удовлетворять **ответ** или, другими словами, решение задачи.

Индивидуальная задача I получается из задачи Π, если всем параметрам задачи Π присвоить конкретные значения.

Комбинаторная оптимизационная задача Π есть либо задача минимизации, либо задача максимизации и состоит из следующих трех частей:

(1) множества D_Π индивидуальных задач;

(2) конечного множества $S_\Pi(I)$ допустимых решений индивидуальной задачи $I \in D_\Pi$;

(3) функции ϕ_Π, сопоставляющей каждой индивидуальной задаче $I \in D_\Pi$ и каждому допустимому решению $\sigma \in S_\Pi(I)$ некоторое положительное число $\phi_\Pi(I, \sigma)$, называемое величиной решения σ.

Если Π задача минимизации, то **оптимальным решением** индивидуальной задачи $I \in D_\Pi$ является такое допустимое решение $\sigma^* \in S_\Pi(I)$, что для всех $\sigma \in S_\Pi(I)$ выполнено неравенство $\phi_\Pi(I, \sigma^*) \leq \phi_\Pi(I, \sigma)$).

Среди комбинаторных оптимизационных задач, частью которых являются и задачи теории расписаний, можно выделить полиномиально разрешимые и NP-трудные. Для каждой полиномиально разрешимой задачи известен по крайней мере один эффективный алгоритм ее решения, то есть алгоритм, трудоемкость которого (число выполняемых элементарных операций, время решения) ограничена сверху некоторым полиномом от длины записи исходной информации задачи, закодированной в бинарном алфавите. Для NP-трудных задач такие алгоритмы неизвестны и скорее всего не существуют. Во всяком случае, если бы существовал полиномиальный алгоритм решения какой-либо одной NP-трудной задачи, то существовали бы полиномиальные алгоритмы для очень многих задач, которые традиционно считаются труднорешаемыми.

Более подробно с вопросами оценки сложности алгоритмов и теорией полиномиальной сводимости комбинаторных задач можно ознакомиться в монографиях [?],[?],[?].

В диссертации используется общепринятая техника доказательства NP-трудности оптимизационных задач. Пусть в конечном множестве S допустимых расписаний требуется отыскать расписание $S^{\star}$, которому соответствует наименьшее значение $F(S)$. Сформулируем соответствующую задачу распознавания: существует ли во множестве S такое расписание S_0, что $F(S_0) \leq y$ для заданного числа y? Очевидно, указанное расписание существует тогда и только тогда, когда $F(S^{\star}) \leq y$. Таким образом, оптимизационнная задача не легче, чем соответствующая ей задача распознавания, и из NP-трудности задачи распознавания следует NP-трудность исходной оптимизационнной задачи. Для доказательства NP-трудности задачи распознавания достаточно осуществить полиномиальное сведение к ней некоторой заведомо NP-трудной задачи.

Пусть Π — задача распознавания, D_{Π} — множество ее индивидуальных задач, Y_{Π} — множество ее индивидуальных задач с ответом "да", I — индивидуальная задача.

Тогда полиномиальная сводимость задачи распознавания Π_1 к задаче Π_2 означает наличие функции $f : D_{\Pi_1} \rightarrow D_{\Pi_2}$, удовлетворяющей двум условиям:

(1) функция f вычисляется полиномиальным алгоритмом и
(2) для всех $I \in D_{\Pi_1}$ выполняется, что $I \in Y_{\Pi_1}$ тогда и только тогда, когда $f(I) \in Y_{\Pi_2}$.

Среди NP-трудных задач можно выделить задачи, для которых существуют псевдополиномиальные алгоритмы их решения, то есть алгоритмы, сложность которых ограничена сверху некоторым полиномом от длины записи исходной информации задачи, закодированной в унарном алфавите (натуральное число k в этом алфавите представляется последовательностью k единиц). Остальные NP-трудные задачи относятся к классу NP-трудных в сильном смысле. Если эталонная задача является NP-трудной в сильном смысле, то нет необходимости строить ее полиномиальное сведение к рассматриваемой задаче распознавания. Достаточно построить псевдополиномиальное сведение и тем самым убедиться в NP-трудности в сильном смысле исходной задачи.

Обозначим функции длины записи индивидуальной задачи I, закодированной в бинарном и унарном алфавите, через $Length(I)$ и $Max(I)$ соответственно. Тогда псевдополиномиальная сводимость задачи распознавания Π_1 к задаче Π_2 означает наличие функции $f : D_{\Pi_1} \to D_{\Pi_2}$, удовлетворяющей следующим условиям:
(1) каждая индивидуальная задача $I \in D_{\Pi_1}$ принадлежит множеству Y_{Π_1} тогда и только тогда, когда $f(I) \in Y_{\Pi_2}$;
(2) функция f может быть вычислена за время, ограниченное полиномом от двух переменных $Length(I)$ и $Max(I)$;
(3) существует такой полином q_1, что для всех $I \in D_{\Pi_1}$ имеет место неравенство $q_1(Length(f(I))) \geq Length(I)$;

(4) существует такой полином q_2 от двух переменных, что для всех $I \in D_{\Pi_1}$ выполнено неравенство $Max(f(I)) \leq q_2(Max(I), Length(I))$.

Приведем список эталонных NP-трудных задач, используемых для доказательства NP-трудности задач теории расписаний, рассматриваемых в диссертации.

Задача СУММА РАЗМЕРОВ.

Даны целые $e_i \geq 1$, $i \in I$, и целое $E \geq 1$. Найти $I' \subseteq I$ такое, что

$$\sum_{i \in I'} e_i = E.$$

Задача ПРОИЗВЕДЕНИЕ РАЗМЕРОВ.

Даны целые $e_i \geq 1$, $i \in I$, и целое $E \geq 1$. Найти $I' \subseteq I$ такое, что

$$\prod_{i \in I'} e_i = E.$$

Задача 3-РАЗБИЕНИЕ.

Даны множество I, состоящее из $3n$ элементов, целое $E \geq 1$ и целые $E/4 \leq e_i \leq E/2$, $i \in I$, такие, что $\sum_{i \in I} e_i = nE$. Найти такое разбиение I на n непересекающихся подмножеств $I_1, I_2, \ldots, I_n$, что для любого l $1 \leq l \leq n$, выполнено соотношение $\sum\limits_{i \in I_l} e_i = E$.

Задача ТРЕХМЕРНОЕ СОЧЕТАНИЕ.

Дано множество $\Phi \subseteq W \times X \times Y$, где W, X и Y — непересекающиеся множества, содержащие одинаковое число элементов q. Верно ли, что Φ содержит трехмерное сочетание, т.е. подмножество $\Phi' \subseteq \Phi$ такое, что $|\Phi'| = q$ и никакие два разных элемента из Φ' не имеют ни одной равной координаты.

Все перечисленные задачи приведены в списке NP-трудных задач в [?]. Задача СУММА РАЗМЕРОВ и задача ПРОИЗВЕДЕНИЕ РАЗ-

МЕРОВ являются NP-трудными, а задача 3-РАЗБИЕНИЕ и задача ТРЕХМЕРНОЕ СОЧЕТАНИЕ являются NP-трудными в сильном смысле.

Поскольку для NP-трудных задач скорее всего не существуют эффективные полиномиальные алгоритмы, представляет интерес построение для них приближенных полиномиальных алгоритмов.

Алгоритм A называется **приближенным алгоритмом** решения задачи минимизации Π, если для любой индивидуальной задачи $I \in D_\Pi$ алгоритм A отыскивает некоторое допустимое решение $\sigma \in S_\Pi(I)$. Определим отношение $R_A(I)$ следующим образом:

$$R_A(I) = \frac{A(I)}{OPT(I)}.$$

Погрешностью приближенного алгоритма A решения задачи Π назовем величину R_A, определяемую следующим образом:

$$R_A = inf\{r \geq 1 | R_A(I) \leq r \text{ для всех } I \in D_\Pi\}.$$

Перейдем к описанию задач теории расписаний. Как уже было упомянуто во введении, в дальнейшем будут рассматриваться задачи, удовлетворяющие следующим условиям.

- Вся исходная информация об индивидуальной задаче определена и известна заранее.
- В каждый момент времени на любой машине выполняется не более одной работы.
- В каждый момент времени любая работа обслуживается не более одной машиной.
- Прерывания в выполнении работ не допустимы.

В двух первых главах данной работы рассматриваются задачи построения расписаний на одной машине.

Пусть на одной машине требуется выполнить множество работ $J = \{J_1, J_2, \ldots, J_n\}$. Обозначим через s_i момент начала выполнения работы J_i. Предполагается, что все работы готовы к выполнению в момент времени $t_0 \geq 0$, а длительность работы J_i равна $p_i = \alpha_i \Upsilon(s_i)$, где $\Upsilon(t)$ — кусочно-непрерывная функция, определенная для всех $t \geq t_0$, $\alpha_i > 0$ — рациональные числа, $1 \leq i \leq n$.

Обозначим через $C_i = s_i + p_i$ — время завершения работы J_i. Для описания целевых функций определим следующие параметры: d_i — директивный срок на завершение выполнения работы J_i, $w_i \geq 0$ — вес работы J_i.

В диссертации рассматриваются задачи со следующими целевыми функциями.

$f = C_{max} = \max\limits_{J_i \in J} C_i$ — время завершения всех работ,

$f = \Sigma C_i = \sum\limits_{J_i \in J} C_i$ — сумма времен завершения всех работ,

$f = \Sigma w_i C_i = \sum\limits_{J_i \in J} w_i C_i$ — взвешенная сумма времен завершения всех работ,

$f = L_{max} = \max\limits_{J_i \in J} (C_i - d_i)$ — максимальное временное смещение,

$f = \Sigma U_i = \sum\limits_{J_i \in J} U_i$ — число запаздывающих работ,

где $U_i = 1$, если $C_i > d_i$, и $U_i = 0$ в противном случае. Требуется найти расписание, когда целевая функция f принимает наименьшее значение.

При обозначении конкретной задачи будет использоваться запись $\alpha|\beta|\gamma$, введенная в [?]. В поле α записывается информация о машинах. Для задач построения расписания на одной машине $\alpha = 1$. В поле β указываются различные характеристики работ. Запись $p_i = \Upsilon_i(s_i)$ в этом поле означает, что длительность выполнения работы J_i равна $p_i = \Upsilon_i(s_i)$; запись $prec$ означает, что между работами есть отношения предшествования; запись $w.i.$ означает, что машина должна выполнять работы без простоев. В третьем поле задается вид целевой функции.

Будем говорить, что индивидуальная задача **хорошо определена**, если выполнены следующие условия:

$$\Upsilon(t_0) > 0, \tag{1.1}$$

$$t_1 + \alpha_i \Upsilon(t_1) < t_2 + \alpha_i \Upsilon(t_2) \text{ при всех } t_2 > t_1 \geq t_0 \text{ и всех } J_i \in J. \tag{1.2}$$

Прокоментируем эти условия подробнее. Первое условие означает, что в начальный момент времени все работы имеют положительную длительность. Из второго условия следует, что если работа начнет выполняться раньше, то и раньше закончится, хотя время, затраченное на ее выполнение, может быть и больше. Заметим, что в этом условии наряду с функцией $\Upsilon(t)$ участвуют и параметры α_i. Поэтому одни индивидуальные задачи для некоторой функции $\Upsilon(t)$ могут быть хорошо определены, а другие с той же функцией — нет. В качестве примера рассмотрим $\Upsilon(t) = 1/t$. Индивидуальная задача с $\Upsilon(t) = 1/t$ и $t_0 = 1$ хорошо определена, если $\alpha_i < 1$ для каждой работы J_i. Таким образом, для некоторых функций условие (??) сужает класс рассматриваемых задач. Если $\Upsilon(t)$ — неубывающая функция, то второе условие выполнено для любого набора параметров α_i.

Лемма 1 *Пусть* $\Upsilon(\tau) \leq 0$ *для некоторого* $\tau > t_0$ *и для любого* $t \in [t_0, \tau)$ *выполняется* $\Upsilon(t) > 0$, *а для индивидуальной задачи с длительностями работ* $p_i = \alpha_i \Upsilon(s_i)$ *выполнены условия (??) и (??). Тогда для любого* $s_i \in [t_0, \tau)$ *время завершения работы* J_i *меньше* τ.

Доказательство. Действительно, $C_i = s_i + \alpha_i \Upsilon(s_i)$. Так как $s_i < \tau$, то по условию 2 $s_i + \alpha_i \Upsilon(s_i) < \tau + \alpha_i \Upsilon(\tau) \leq \tau$. Лемма ?? доказана.

Предположим, что целевая функция не убывает от времени завершения каждой работы. Это условие выполняется для всех рассматриваемых далее задач. Тогда из условий (??) и (??) следует, что в оптимальном расписании машина работает без простоев, и расписание полностью определяется последовательностью π выполнения работ. Такую последовательность будем называть **расписанием**. Более того, из условий (??) и (??) согласно лемме 1 следует, что если для некоторого $\tau > t_0$ выполняется $\Upsilon(\tau) \le 0$, то все работы заканчиваются до момента τ и их длительность строго положительна.

В [?] было введено понятие 1-приоритетно порождающего функционала.

Определение 1 [?] *Пусть $\mathcal{P}$ — множество всех перестановок π из элементов множества J. Функционал $f(\pi)$ на множестве $\mathcal{P}$ называется* **1-приоритетно порождающим**, *если существует функционал $\omega(i)$ такой, что для любых элементов i, l и любых перестановок $\pi^1 = (\pi_1, i, l, \pi_2)$ и $\pi^2 = (\pi_1, l, i, \pi_2)$ из $\mathcal{P}$ выполняются соотношения:*

$$(a) \text{ если } \omega(i) > \omega(l), \text{ то } f(\pi^1) \le f(\pi^2);$$

$$(b) \text{ если } \omega(i) = \omega(l), \text{ то } f(\pi^1) = f(\pi^2).$$

Величина $\omega(i)$ называется **приоритетом** элемента i.

Утверждение 1 [?] *Пусть $\pi \in \mathcal{P}$ и функционал $F(\pi)$ является 1-приоритетно порождающим на множестве $\mathcal{P}$. Тогда $F(\pi)$ принимает минимальное значение на любой перестановке, в которой элементы упорядочены по невозрастанию их приоритетов.*

1.2 Линейные функции

Пусть $\Upsilon(t)$ — произвольная линейная функция, то есть $\Upsilon(t) = at + b$ Тогда время выполнения работы J_i равно

$$p_i = \alpha_i(as_i + b). \tag{1.3}$$

Если $a = 0$, то получаем задачи теории расписаний с фиксированными длительностями. Случай $b = 0$ изучался в [?], [?].

Для рассматриваемых задач условия (??) и (??) принимают следующий вид:

$$at_0 + b > 0, \tag{1.4}$$

$$-\alpha_i a < 1 \text{ при всех } J_i \in J. \tag{1.5}$$

Заметим, что если в рассматриваемой задаче $a < 0$ и выполняются условия (??) и (??), то согласно лемме 1 все работы заканчиваются до момента $-b/a$.

Рассмотрим задачу $1|p_i = \alpha_i(as_i + b)|C_{max}$.

Теорема 1 *Пусть в индивидуальной задаче* $1|p_i = \alpha_i(as_i+b)|C_{max}$ *выполнены неравенства* (??) *и* (??). *Тогда любая перестановка работ дает оптимальное расписание и*

$$C_{max} = \begin{cases} t_0 + b\sum_{i=1}^{n}\alpha_i & , \textit{при} \quad a = 0; \\ (t_0 + b/a)\prod_{i=1}^{n}(\alpha_i a + 1) - b/a & , \textit{при } a \neq 0. \end{cases} \tag{1.6}$$

Доказательство. Покажем, что момент завершения n работ задается формулой (??) и не зависит от порядка их выполнения. Для случая $a = 0$ утверждение очевидно. Для случая $a \neq 0$ доказательство проведем индукцией по числу работ. Рассмотрим произвольную перестановку π из n работ. Без ограничения общности можно считать, что работы занумерованы по порядку от 1 до n. Для $n = 1$ равенство выполнено. Пусть оно выполнено для $n = k - 1$. Тогда имеем

$$C_k = C_{k-1} + p_k = (t + b/a) \prod_{i=1}^{k-1} (\alpha_i a + 1) - b/a + \alpha_k (a s_k + b)$$

$$= (t + b/a) \prod_{i=1}^{k-1} (\alpha_i a + 1) - b/a + \alpha_k a((t + b/a) \prod_{i=1}^{k-1} (\alpha_i a + 1) - b/a)) + \alpha_k b$$

$$= (\alpha_k a + 1)(t + b/a) \prod_{i=1}^{k-1} (\alpha_i a + 1) + \alpha_k b - \alpha_k b - b/a$$

$$= (t + b/a) \prod_{i=1}^{k} (\alpha_i a + 1) - b/a.$$

Теорема **??** доказана.

Результат теоремы **??** позволяет применять классическую технику из теории расписаний для решения задач с пропорциональными линейными длительностями. Действительно, рассмотрим задачу $1|p_i = \alpha_i(as_i + b), prec|f_{max}$, в которой на одной машине требуется выполнить n работ с отношением предшествования между работами. Для каждой работы J_i задана стоимостная функция $f_i(C_i)$. Все функции f_i являются неубывающими и могут быть различны для разных работ. Требуется построить расписание работ, минимизирующее функцию $f_{max} = \max_{J_i \in J} f_i$.

Аналогично классическому случаю применение правила Лолэ [?] позволяет найти оптимальную перестановку, решающую эту задачу.

Правило Лолэ. Пусть $L \subseteq J$ — множество работ, у которых нет последующих, и работа $J_l \in L$ такова, что

$$f_l(C_n) = \min_{i \in L} f_i(C_n),$$

где C_n вычисляется по формуле (**??**). Тогда существует оптимальное расписание, в котором J_l выполняется последней.

Повторив правило Лолэ для оставшихся работ, не более чем за $O(n^2)$ элементарных операций получим оптимальное расписание.

Теорема 2 *Пусть в индивидуальной задаче* $1|p_i = \alpha_i(as_i+b), prec|f_{max}$ *выполнены неравенства* (**??**) *и* (**??**). *Тогда построенная по правилу Лолэ перестановка является оптимальной.*

Доказательство. По теореме **??** величина C_n не зависит от последовательности работ и может быть найдена за полиномиальное время. Дальнейшее рассуждение совпадает с доказательством аналогичного результата для задачи с фиксированными длительностями и впервые было опубликовано Лолэ [?]. Обозначим через $f^{\star}_{max}(J)$ значение целевой функции в оптимальном расписании. Тогда выполняются следующие неравенства:

$$f^{\star}_{max}(J) \geq \min_{J_i \in L} f_i(C_n), \tag{1.7}$$

$$f^{\star}_{max}(J) \geq f^{\star}_{max}(J \setminus J_i) \text{ для всех } J_i. \tag{1.8}$$

Пусть работа $J_l \in L$ такова, что $f_l(C_n) = \min_{J_i \in L} f_i(C_n)$. Обозначим через $f_l(J)$ значение оптимальной целевой функции при условии, что J_l выполняется последней. Тогда

$$f_l(J) = \max\{f_l(C_n), f^{\star}_{max}(J \setminus l)\}.$$

Из последнего равенства и неравенств (**??**), (**??**) следует, что $f^{\star}_{max}(J) \geq f_l(J)$, и существует оптимальное расписание, в котором работа J_l выполняется последней. Теорема **??** доказана.

Рассмотрим задачи, в которых целевые функции являются 1-приоритетно-порождающими функционалами.

Теорема 3 *Пусть для индивидуальной задачи* $1|p_i = \alpha_i(as_i + b)|L_{max}$ *выполнены условия* (**??**) *и* (**??**). *Тогда любая перестановка работ по неубыванию величин* d_i *является оптимальной.*

Доказательство. Покажем, что целевая функция L_{max} является 1-приоритетно-порождающим функционалом с приоритетом $\omega_i = -d_i$. Найдем достаточные условия, при которых $L_{max}(\pi_1, J_l, J_k, \pi_2) \leq L_{max}(\pi_1, J_k, J_l, \pi_2)$. Имеем во-первых,

$$L_{max}(\pi_1, J_l, J_k, \pi_2) = \max_{J_i \in J} L_i(\pi_1, J_l, J_k, \pi_2)$$

$$= \max(\max_{J_i \in J \setminus \{J_l\} \setminus \{J_k\}} L_i(\pi_1, J_l, J_k, \pi_2), L_l(\pi_1, J_l, J_k, \pi_2), L_k(\pi_1, J_l, J_k, \pi_2))$$
$$= \max(\max_{J_i \in J \setminus \{J_l\} \setminus \{J_k\}} L_i(\pi_1, J_l, J_k, \pi_2), C(\pi_1, J_l) - d_l, C(\pi_1, J_l, J_k) - d_k).$$

Во-вторых

$$L_{max}(\pi_1, J_k, J_l, \pi_2)$$

$$= \max(\max_{J_i \in J \setminus \{J_l\} \setminus \{J_k\}} L_i(\pi_1, J_l, J_k, \pi_2), C(\pi_1, J_k) - d_k, C(\pi_1, J_k, J_l) - d_l).$$

Так как $C(\pi_1, J_l, J_k) - d_l \geq C(\pi_1, J_l) - d_l$, $C(\pi_1, J_k, J_l) - d_k \geq C(\pi_1, J_k) - d_k$, и по теореме 1 $C(\pi_1, J_l, J_k) = C(\pi_1, J_k, J_l)$, то
$L_{max}(\pi_1, J_l, J_k, \pi_2) \leq L_{max}(\pi_1, J_k, J_l, \pi_2)$ при $d_l \leq d_k$.

Теорема **??** доказана.

Теорема 4 *Пусть для индивидуальной задачи* $1|p_i = \alpha_i(as_i + b)|\Sigma w_i C_i$ *выполнены условия* (**??**) *и* (**??**). *Тогда любая перестановка работ по невозрастанию величин* $w_i(\alpha_i^{-1} + a)$ *является оптимальной.*

Доказательство. Покажем, что целевая функция $f = \Sigma w_i C_i$ является 1-приоритетно порождающим функционалом с приоритетом $\omega_i = w_i(\alpha_i^{-1} + a)$. Найдем достаточные условия, при которых $f(\pi_1, J_l, J_k, \pi_2) \leq$

$f(\pi_1, J_k, J_l, \pi_2)$. Так как $C(\pi_1, J_l, J_k) = C(\pi_1, J_k, J_l)$, то целевые функции отличаются только слагаемыми с весами w_l и w_k. Поэтому имеем

$$w_l C(\pi_1, J_l) + w_k C(\pi_1, J_l, J_k) \leq w_k C(\pi_1, J_k) + w_l C(\pi_1, J_k, J_l). \qquad (1.9)$$

Преобразуем левую и правую части неравенства 8. В результате имеем.

$$w_l C(\pi_1, J_l) + w_k C(\pi_1, J_l, J_k) = w_l(C(\pi_1) + \alpha_l(aC(\pi_1) + b))$$

$$+w_k(C(\pi_1) + \alpha_l(aC(\pi_1) + b) + \alpha_k a(C(\pi_1) + \alpha_l(aC(\pi_1) + b)) + \alpha_k b).$$

$$w_k C(\pi_1, J_k) + w_l C(\pi_1, J_k, J_l) = w_k(C(\pi_1) + \alpha_k(aC(\pi_1) + b))$$

$$+w_l(C(\pi_1) + \alpha_k(aC(\pi_1) + b) + \alpha_l a(C(\pi_1) + \alpha_k(aC(\pi_1) + b)) + \alpha_l b).$$

Подставляя полученные выражения в (**??**), получаем

$$(1 + \alpha_k a) w_k \alpha_l (aC(\pi_1) + b) \leq (1 + \alpha_l a) w_l \alpha_k (aC(\pi_1) + b).$$

Следовательно, если $w_k(\alpha_k^{-1} + a) \leq w_l(\alpha_l^{-1} + a)$, то $f(\pi_1, J_l, J_k, \pi_2) \leq f(\pi_1, J_k, J_l, \pi_2)$.
Теорема **??** доказана.

Рассмотрим задачу о минимизации числа запаздывающих работ $1|p_i = \alpha_i(as_i + b)|\Sigma U_i$. Классическая задача $1||\Sigma U_i$ с тем же критерием оптимальности решается алгоритмом Ходжсона-Мура [**?**] за $O(n \log n)$ операций. Аналогичный алгоритм рассматривался в [**?**] для задачи $1|p_i = \alpha_i s_i|\Sigma U_i$. Обозначим через J^* — множество незапаздывающих работ в оптимальном расписании и без ограничения общности будем считать, что нумерация работ в алгоритме на шаге 5 совпадает с построенной последовательностью.

<u>Алгоритм 1.</u>
Шаг 1. Полагается $J^* := J$.

Шаг 2. Работы упорядочиваются по неубыванию величин d_i.

Шаг 3. Если никакая работа из J^* в последовательности не запаздывает, то стоп. Последовательность оптимальна.

Шаг 4. В полученной последовательности находится первая запаздывающая работа. Пусть J_k является такой работой.

Шаг 5. Находится работа J_i такая, что $\alpha_i = \max_{j=1,\ldots,k} \alpha_j$. Работа J_i ставится последней в J^*. Полагается $J^* = J^* \setminus J$ и осуществляется возврат на шаг 3.

Докажем, что этот алгоритм позволяет находить оптимальное решение задачи $1|p_i = \alpha_i(as_i + b)|\Sigma U_i$. Для удобства найденные алгоритмом на шаге 4 работы, будем называть *критическими.*

Теорема 5 *Пусть в индивидуальной задаче* $1|p_i = \alpha_i(as_i + b)|\Sigma U_i$ *выполнены условия* (**??**) *и* (**??**). *Тогда Алгоритм 1 находит оптимальное расписание.*

Доказательство. Из теоремы **??** следует, что если существует расписание без запаздывающих работ, то оно будет найдено на первом шаге алгоритма. Пусть теперь S^* — оптимальное расписание, Z^* — множество его запаздывающих работ, и $Z^* \neq \emptyset$. Пусть в результате работы алгоритма найдено расписание S с k запаздывающими работами $J_{i_1}, \ldots, J_{i_k}$. Обозначим множество этих работ через Z. По индукции покажем, что каждой работе из множества Z можно инъективно поставить в соответствие работу из множества Z^*.

Для удобства будем считать, что работы занумерованы по неубыванию величин d_i. Пусть J_{r_1} является первой критической работой, найденной на шаге 4, а J_{i_1} является первой работой, выбранной алгоритмом на шаге 5. Обозначим $I_1 = \{J_1, \dots, J_{r_1}\}$. Тогда $C_{r_1} = \sum_{i \in I_1} p_i > d_{r_1}$. Кроме того, $d_i \leq d_{r_1}$ для всех $i = 1, \dots, r_1$. Так как величина C_{r_1} не зависит от последовательности выполнения работ, то среди работ из I_1 есть по крайней мере одна запаздывающая в оптимальном расписании, то есть $Z^* \cap I_1 \neq \infty$. Следовательно, в оптимальном расписании среди работ $J_{j_1}, \dots, J_{j_m}$ есть работы с индексом, не превышающим r_1. Выберем работу согласно следующему правилу. Если $J_{i_1} \in Z^*$, то выберем ее $j_1 = i_1$. В противном выберем произвольную работу $J_{j_1} \in Z^* \cap I_1$. Тогда $\alpha_{j_1} \leq \alpha_{i_1}$ и $j_1 \leq r_1$. Покажем, что для любого $1 \leq q \leq k$ существует инъективное отображение $g : \{i_1, \dots, i_q\} \to Z_*$ такое, что

$$\alpha_{j_p} \leq \alpha_{i_p} \text{ для всех } p \leq q; \tag{1.10}$$

$$j_p \leq r_p, \text{ для всех } p \leq q, \tag{1.11}$$

где J_{r_p} — критическая работа на p-й итерации. Как было показано выше для $q = 1$ такое инъективное отображение существует.

Предположим, что $q < k$ и $J_{i_1}, \dots, J_{i_q}$ — совокупность q первых запаздывающих работ, вычисленных алгоритмом на шаге 4, и $J_{j_1}, \dots, J_{j_q}$ соответствующие им запаздывающие работы из расписания S^* такие, что выполнены условия (??) и (??). Пусть $J_{r_{q+1}}$ — критическая работа, найденная на $(q+1)$-й итерации. Тогда из теоремы 1 следует, что

$$d_{r_{q+1}} < (t + b/a) \prod_{l=1}^{r_{q+1}} (\alpha_l a + 1) / \prod_{l=1}^{q} (\alpha_{i_l} a + 1) - b/a.$$

Из (??) и (??) получаем

$$(t+b/a) \prod_{l=1}^{r_{q+1}} (\alpha_l a+1) / \prod_{l=1}^{q} (\alpha_{i_l} a+1) - b/a \leq (t+b/a) \prod_{l=1}^{r_{q+1}} (\alpha_l a+1) / \prod_{l=1}^{q} (\alpha_{j_l} a+1) - b/a.$$

Следовательно, среди работ Z^* есть работы с номером не более r_{q+1}, отличные от работ $J_{j_1}, \dots, J_{j_q}$. Если $J_{i_{q+1}} \in Z^* \setminus \{J_{j_1}, \dots, J_{j_q}\}$, то берем

$j_{q+1} = i_{q+1}$. В противном случае выберем произвольную работу из $Z^* \cap I_{q+1} \setminus \{J_{j_1}, \dots, J_{j_q}\}$ в качестве j_{q+1}. Покажем, что $\alpha_{j_{q+1}} \leq \alpha_{i_{q+1}}$.

Действительно, $J_{j_{q+1}}$ не может совпадать ни с какой работой J_{i_p}, $p \leq q$, так как в этом случае $J_{i_p} \in Z^* \setminus \{J_{j_1}, \dots, J_{j_q}\} \subset Z^* \setminus \{J_{j_1}, \dots, J_{j_{p-1}}\}$, и согласно приведенному выше правилу выбора работы J_p, мы должны были выбрать $j_p = i_p$. Следовательно, $J_{j_{q+1}} \in I_{q+1} \setminus \{J_{j_1}, \dots, J_{j_q}\}$, и по выбору i_q на шаге 5 алгоритма 1 имеем $\alpha_{j_{q+1}} \leq \alpha_{i_{q+1}}$, что завершает доказательство индукционного шага. В итоге получили, что число элементов в множествах Z и Z^* совпадает, и алгоритм 1 находит оптимальное расписание. Теорема ?? доказана.

1.3 Выпуклые и вогнутые функции

Рассмотрим задачи с нелинейными функциями длительности, для которых удается найти оптимальную перестановку за полиномиальное от числа работ время.

Пусть $\Upsilon(t)$ — выпуклая функция, удовлетворяющая условиям (??) и (??). Примерами таких функций для $t_0 = 0$ являются: $\Upsilon(t) = \exp(t)$, $\Upsilon(t) = t^2+1$, $\Upsilon(t) = 1/(t+1)$, $\Upsilon(t) = |t-1|+1$. Как было отмечено ранее, в первых двух примерах условия (??) и (??) выполняются для любых α_i, а для $\Upsilon(t) = |t-1|+1$ и $\Upsilon(t) = 1/(t+1)$ задача $1|p_i = \alpha_i \Upsilon(s_i)|F$ хорошо определена, если $\alpha_i < 1$ для каждой работы J_i.

Рассмотрим задачу $1|p_i = \alpha_i \Upsilon(s_i)|L_{max}$. Пусть параметры задачи удовлетворяют следующему ограничению

$$d_i \leq d_l \text{ для всех } J_i, J_l \text{ таких, что } \alpha_i < \alpha_l. \tag{1.12}$$

Теорема 6 *Пусть $\Upsilon(t)$ — выпуклая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i \Upsilon(s_i)|L_{max}$ удовлетворяет условиям (??),*

(??) *и* (??). *Тогда любая перестановка работ по неубыванию величин* $\alpha_i + d_i$ *является оптимальной.*

Доказательство. Покажем, что целевая функция L_{max} является 1-приоритетно пораждающим функционалом с приоритетом $\omega_i = -\alpha_i - d_i$. Рассмотрим два расписания $\pi^1 = (\pi_1, J_i, J_l, \pi_2)$ и $\pi^2 = (\pi_1, J_l, J_i, \pi_2)$, отличающиеся друг от друга только положением двух работ J_l и J_i. Предположим, что

$$\alpha_l + d_l > \alpha_i + d_i. \tag{1.13}$$

Тогда для работ J_l и J_i выполняется, что $\alpha_l \geq \alpha_i$ и $d_l \geq d_i$. Сначала докажем первое неравенство. Предположим обратное, пусть $\alpha_l < \alpha_i$. Тогда из (??) следует, что $d_i \leq d_l$ и, складывая два последних неравенства получаем противоречие.

Неравенство $d_l \geq d_i$ в случае $\alpha_l > \alpha_i$ следует из (??), а в случае $\alpha_l = \alpha_i$ следует из (??).

Покажем, что из неравенства $\alpha_l \geq \alpha_i$ следует, что

$$C_{max}(\pi_1, J_l, J_i) \geq C_{max}(\pi_1, J_i, J_l). \tag{1.14}$$

Обозначим через $\bar{t}$ время завершения последней работы из перестановки π_1. Тогда имеем

$$C_{max}(\pi_1, J_l, J_i) = \bar{t} + \alpha_l \Upsilon(\bar{t}) + \alpha_i \Upsilon(\bar{t} + \alpha_l \Upsilon(\bar{t})), \tag{1.15}$$

$$C_{max}(\pi_1, J_i, J_l) = \bar{t} + \alpha_i \Upsilon(\bar{t}) + \alpha_l \Upsilon(\bar{t} + \alpha_i \Upsilon(\bar{t})). \tag{1.16}$$

Рассмотрим линейную функцию

$$R(t) = t\frac{\Upsilon(\bar{t} + \alpha_l \Upsilon(\bar{t})) - \Upsilon(\bar{t})}{\alpha_l \Upsilon(\bar{t})} + \frac{\Upsilon(\bar{t})(\bar{t} + \alpha_l \Upsilon(\bar{t})) - \bar{t}\Upsilon(\bar{t} + \alpha_l \Upsilon(\bar{t}))}{\alpha_l \Upsilon(\bar{t})}.$$

Так как

$$R(\bar{t}) = \Upsilon(\bar{t}), R(\bar{t} + \alpha_l R(\bar{t})) = \Upsilon(\bar{t} + \alpha_l \Upsilon(\bar{t})), \tag{1.17}$$

то подставив (??) в (??), получаем

$$C_{max}(\pi_1, J_l, J_i) = \bar{t} + \alpha_l R(\bar{t}) + \alpha_i R(\bar{t} + \alpha_l R(\bar{t})). \tag{1.18}$$

Из линейности функции $R(t)$, согласно теореме (??) следует равенство

$$\bar{t} + \alpha_l R(\bar{t}) + \alpha_i R(\bar{t} + \alpha_l R(\bar{t})) = \bar{t} + \alpha_i R(\bar{t}) + \alpha_l R(\bar{t} + \alpha_i R(\bar{t})). \tag{1.19}$$

Тогда из (??) и (??) имеем

$$C_{max}(\pi_1, J_l, J_i) = \bar{t} + \alpha_i R(\bar{t}) + \alpha_l R(\bar{t} + \alpha_i R(\bar{t})). \tag{1.20}$$

Из (??) и предположения $\alpha_l > \alpha_i$ следует, что $\bar{t} < \bar{t} + \alpha_i R(\bar{t}) = \bar{t} + \alpha_i \Upsilon(\bar{t}) < \bar{t} + \alpha_l \Upsilon(\bar{t})$. Тогда из выпуклости функции $\Upsilon(t)$ вытекает неравенство $R(\bar{t}+\alpha_i R(\bar{t})) \geq \Upsilon(\bar{t}+\alpha_i \Upsilon(\bar{t}))$. Сравнив выражения в равенствах (??) и (??), получим $C_{max}(\pi_1, J_l, J_i) \geq C_{max}(\pi_1, J_i, J_l)$.

Далее имеем

$$L_{max}(\pi_1, J_l, J_i) =$$

$$\max\{L_{max}(\pi_1); C_{max}(\pi_1, J_l) - d_l; C_{max}(\pi_1, J_l, J_i) - d_i\}; \tag{1.21}$$

$$L_{max}(\pi_1, J_i, J_l) =$$

$$\max\{L_{max}(\pi_1); C_{max}(\pi_1, J_i) - d_i; C_{max}(\pi_1, J_i, J_l) - d_l\}. \tag{1.22}$$

Из (??) и неравенства $d_l \geq d_i$ получаем

$$C_{max}(\pi_1, J_l, J_i) - d_i \geq C_{max}(\pi_1, J_i, J_l) - d_l. \tag{1.23}$$

Кроме того,

$$C_{max}(\pi_1, J_l, J_i) - d_i \geq C_{max}(\pi_1, J_i) - d_i. \tag{1.24}$$

Из (??), (??), (??), (??) следует $L_{max}(\pi_1, J_l, J_i) \geq L_{max}(\pi_1, J_i, J_l)$ и отсюда и из (??), получаем $L_{max}(\pi^1) \geq L_{max}(\pi^2)$.

Осталось показать, что если $\alpha_l + d_l = \alpha_i + d_i$, то $L_{max}(\pi^1) = L_{max}(\pi^2)$. Покажем, что из равенства $\alpha_l + d_l = \alpha_i + d_i$ следует, что $\alpha_l = \alpha_i$ и, следовательно, $d_l = d_i$. Действительно, пусть $\alpha_l > \alpha_i$, тогда из (??) $d_l \geq d_i$ и $\alpha_l + d_l > \alpha_i + d_i$. Получаем противоречие. Следовательно, параметры работ J_i и J_l совпадают и $L_{max}(\pi^1) = L_{max}(\pi^2)$.

Согласно определению 1 целевая функция L_{max} является 1-приоритетно порождающим функционалом с приоритетом $\omega_i = -\alpha_i - d_i$, и по утверждению 1 любая перестановка, в которой работы расположены по неубыванию величин $\alpha_i + d_i$, является оптимальной. Теорема ?? доказана.

Следствие 1 *Пусть $\Upsilon(t)$ — выпуклая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ удовлетворяет условиям (??) и (??). Тогда любая перестановка работ по неубыванию величин α_i, является оптимальной.*

Рассмотрим задачу $1|p_i = \alpha_i \Upsilon(s_i)|\Sigma w_i C_i$.

Теорема 7 *Пусть $\Upsilon(t)$ — выпуклая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i \Upsilon(s_i)|\Sigma w_i C_i$ удовлетворяет условиям (??) и (??). Тогда*

a) если $\Upsilon(t) \geq 0$ для всех t, $\lim\limits_{t \to \infty} \dfrac{d\Upsilon(t)}{dt} = \infty$ и $w_i \geq w_l$ для всех J_i, J_l таких, что $\alpha_i < \alpha_l$, то любая перестановка работ по неубыванию величин $\alpha_i - w_i$ задает оптимальное расписание;

b) если $\Upsilon(t) \geq 0$ для всех t, $\lim\limits_{t \to \infty} \dfrac{d\Upsilon(t)}{dt} = H$ и

$$w_i(\alpha_i^{-1} + H) \geq w_l(\alpha_l^{-1} + H) \tag{1.25}$$

для всех J_i, J_l таких, что $\alpha_i < \alpha_l$, то любая перестановка работ по неубыванию величин $\alpha_i - w_i(\alpha_i^{-1} + H)$ задает оптимальное расписание;

c) если существует $\tau > t_0$ такое, что $\Upsilon(\tau) = 0$ и для любого $t \in [t_0, \tau)$ выполнено $\Upsilon(t) > 0$, $\lim\limits_{t \to \tau - 0} \dfrac{d\Upsilon(t)}{dt} = H$ и $w_i(\alpha_i^{-1} + H) \geq w_l(\alpha_l^{-1} + H)$ для всех J_i, J_l таких, что $\alpha_i < \alpha_l$, то любая перестановка работ по неубыванию величин $\alpha_i - w_i(\alpha_i^{-1} + H)$ задает оптимальное расписание.

Доказательство проведем для случая b).

Рассмотрим два расписания $\pi^1 = (\pi_1, J_i, J_l, \pi_2)$ и $\pi^2 = (\pi_1, J_l, J_i, \pi_2)$, отличающиеся друг от друга только положением двух работ J_l и J_i. Предположим, что

$$\alpha_l - w_l(\alpha_l^{-1} + H) > \alpha_i - w_i(\alpha_i^{-1} + H). \tag{1.26}$$

Тогда для работ J_l и J_i выполняется, что $\alpha_l \geq \alpha_i$ и $w_l(\alpha_l^{-1} + H) \leq w_i(\alpha_i^{-1} + H)$. Сначала, докажем первое неравенство. Предположим обратное, пусть $\alpha_l < \alpha_i$. Тогда из (??) $w_l(\alpha_l^{-1} + H) \leq w_i(\alpha_i^{-1} + H)$ и, вычитая последнее неравенство из предыдущего получаем противоречие.

Неравенство $w_l(\alpha_l^{-1} + H) \leq w_i(\alpha_i^{-1} + H)$ в случае $\alpha_l > \alpha_i$ следует из (??), а в случае $\alpha_l = \alpha_i$ следует из (??).

Обозначим через $\bar{t}$ время завершения работ из перестановки π_1. Тогда

$$f(\pi_1, J_l, J_i) = f(\pi_1) + w_l\bar{t} + w_l\alpha_l\Upsilon(\bar{t}) + w_i\bar{t} + w_i\alpha_l\Upsilon(\bar{t}) + w_i\alpha_i\Upsilon(\bar{t} + \alpha_l\Upsilon(\bar{t})),$$

$$f(\pi_1, J_i, J_l) = f(\pi_1) + w_i\bar{t} + w_i\alpha_i\Upsilon(\bar{t}) + + w_l\bar{t} + w_l\alpha_i\Upsilon(\bar{t}) + w_l\alpha_l\Upsilon(\bar{t} + \alpha_i\Upsilon(\bar{t})).$$

Далее имеем

$$f(\pi_1, J_l, J_i) - f(\pi_1, J_i, J_l) = w_l\alpha_l\Upsilon(\bar{t}) + w_i\alpha_l\Upsilon(\bar{t}) + w_i\alpha_i\Upsilon(\bar{t} + \alpha_l\Upsilon(\bar{t}))$$
$$- w_i\alpha_i\Upsilon(\bar{t}) - w_l\alpha_i\Upsilon(\bar{t}) - w_l\alpha_l\Upsilon(\bar{t} + \alpha_i\Upsilon(\bar{t})). \tag{1.27}$$

Рассмотрим линейную функцию $R(t) = at + b$, определенную в предыдущей теореме. Подставив (??) в (??), получаем

$$f(\pi_1, J_l, J_i) - f(\pi_1, J_i, J_l) = w_l\alpha_l R(\bar{t}) + w_i\alpha_l R(\bar{t}) + w_i\alpha_i R(\bar{t} + \alpha_l R(\bar{t}))$$

$$-w_i\alpha_i R(\bar{t}) - w_l\alpha_i R(\bar{t}) - w_l\alpha_l\Upsilon(\bar{t} + \alpha_i\Upsilon(\bar{t})).$$

Из выпуклости функции $\Upsilon(t)$ следует, что

$$f(\pi_1, J_l, J_i) - f(\pi_1, J_i, J_l) \geq w_l\alpha_l R(\bar{t}) + w_i\alpha_l R(\bar{t}) + w_i\alpha_i R(\bar{t} + \alpha_l R(\bar{t}))$$

$$-w_i\alpha_i R(\bar{t}) - w_l\alpha_i R(\bar{t}) - w_l\alpha_l R(\bar{t} + \alpha_i R(\bar{t})).$$

Заменив $R(t)$ на $at + b$, получаем

$$f(\pi_1, J_l, J_i) - f(\pi_1, J_i, J_l) \geq (at + b)\alpha_l\alpha_i(w_i(\alpha_i^{-1} + a) - w_l(\alpha_l^{-1} + a)).$$

Заметим, что из (**??**) $(\alpha_i^{-1} + a) \geq 0$ для любой работы J_i. Тогда если $w_i \geq w_l$, то $f(\pi_1, J_l, J_i) - f(\pi_1, J_i, J_l) \geq 0$. Если $w_i < w_l$, то из выпуклости функции $\Upsilon(t)$ следует, что $a \leq H$ и $f(\pi_1, J_l, J_i) - f(\pi_1, J_i, J_l) \geq (at + b)\alpha_l\alpha_i(w_i(\alpha_i^{-1} + H) - w_l(\alpha_l^{-1} + H)) \geq 0$. Итак $f(\pi_1, J_l, J_i) \geq f(\pi_1, J_i, J_l)$. Кроме того из $\alpha_l \geq \alpha_i$ согласно следствию (**??**) имеем, что $C_{max}(\pi_1, J_l, J_i) \geq C_{max}(\pi_1, J_i, J_l)$. Поэтому $f(\pi^1) \geq f(\pi^2)$.

Предположим, что $\alpha_l - w_l(\alpha_l^{-1} + H) = \alpha_i - w_i(\alpha_i^{-1} + H)$. Тогда из (**??**) следует, что $\alpha_l = \alpha_i$, $w_l = w_i$ и, следовательно, $f(\pi^1) = f(\pi^2)$. По теореме (**??**) любая перестановка, в которой работы расположены по неубыванию величин $\alpha_i - w_i(\alpha_i^{-1} + H)$, является оптимальной. Случай b) доказан. Доказательство случаев a),c) проводится аналогично.

Теорема **??** доказана.

Следствие 2 *Пусть $\Upsilon(t)$ — выпуклая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i\Upsilon(s_i)|\Sigma C_i$ удовлетворяет условиям* (**??**) *и* (**??**). *Тогда любая перестановка работ по неубыванию величин α_i является оптимальной.*

Проиллюстрируем результат теоремы **??** на примерах.

Пример 1

Пусть $\Upsilon(t) = y^t$, $y > 1$ и параметры задачи $1|p_i = \alpha_i \Upsilon(s_i)|w_i C_i$ удовлетворяют следующему ограничению:

$$w_i \geq w_l \text{ для всех } J_i \text{ и } J_l \text{ таких, что } \alpha_i < \alpha_l.$$

Тогда любая перестановка работ по неубыванию величин $\alpha_i - w_i$ задает оптимальное расписание.

Пример 2

Пусть $\Upsilon(t) = t^y$, $y > 1$, $t_0 > 0$ и параметры задачи $1|p_i = \alpha_i \Upsilon(s_i)|w_i C_i$ удовлетворяют следующему ограничению:

$$w_i \geq w_l \text{ для всех } J_i \text{ и } J_l \text{ таких, что } \alpha_i < \alpha_l.$$

Тогда любая перестановка работ по неубыванию величин $\alpha_i - w_i$ задает оптимальное расписание.

Пример 3

Пусть $\Upsilon(t) = t^{-y}$, $y > 0$, $t_0 > 0$ и параметры задачи $1|p_i = \alpha_i \Upsilon(s_i)|w_i C_i$ удовлетворяют следующему ограничению:

$$w_i \alpha_i^{-1} \geq w_l \alpha_l^{-1} \text{ для всех } J_i \text{ и } J_l \text{ таких, что } \alpha_i < \alpha_l.$$

Тогда любая перестановка работ по неубыванию величин $\alpha_i - w_i \alpha_i^{-1}$ задает оптимальное расписание.

В заключение параграфа рассмотрим случай, когда $\Upsilon(t)$ является вогнутой функцией при $t \geq t_0$. Примерами таких функций являются функция $\Upsilon(t) = \sqrt{t}$, $\Upsilon(t) = \log t$. Проведя рассуждения, аналогичные случаю, когда $\Upsilon(t)$ является выпуклой функцией, получаем следующие результаты.

Рассмотрим задачу $1|p_i = \alpha_i\Upsilon(s_i)|L_{max}$. Пусть параметры задачи удовлетворяют следующему ограничению:

$$d_i \geq d_l \text{ для всех } J_i \text{ и } J_l \text{ из } J \text{ таких, что } \alpha_i < \alpha_l. \tag{1.28}$$

Теорема 8 *Пусть $\Upsilon(t)$ — вогнутая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i\Upsilon(s_i)|L_{max}$ удовлетворяет условиям (??), (??) и (??). Тогда любая перестановка работ по невозрастанию величин $\alpha_i - d_i$ является оптимальной.*

Доказательство. Поскольку условия (??) являются аналогом условий (??) в теореме 7, то проводя аналогичные рассуждения, получаем утверждение теоремы.

Следствие 3 *Пусть $\Upsilon(t)$ — вогнутая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i\Upsilon(s_i)|C_{max}$ удовлетворяет условиям (??) и (??). Тогда любая перестановка работы по невозрастанию величин α_i является оптимальной.*

Теорема 9 *Пусть $\Upsilon(t)$ — вогнутая функция при $t \geq t_0$, а индивидуальная задача $1|p_i = \alpha_i\Upsilon(s_i)|\Sigma w_iC_i$ удовлетворяет условиям (??) и (??). Тогда*
a) если $\Upsilon(t) \geq 0$ для всех t, $\lim\limits_{t\to\infty} \dfrac{d\Upsilon(t)}{dt} = H$ и $w_i(\alpha_i^{-1}+H) \leq w_l(\alpha_l^{-1}+H)$ для всех J_i, J_l таких, что $\alpha_i < \alpha_l$, то любая перестановка работ по невозрастанию величин $\alpha_i + w_i(\alpha_i^{-1} + H)$ задает оптимальное расписание;
b) если существует $\tau > t_0$ такой, что $\Upsilon(\tau) = 0$ и для любого $t \in [t_0, \tau)$ выполнено $\Upsilon(t) > 0$, $\lim\limits_{t\to\tau-0} \dfrac{d\Upsilon(t)}{dt} = H$ и $w_i(\alpha_i^{-1} + H) \leq w_l(\alpha_l^{-1} + H)$ для всех J_i, J_l таких, что $\alpha_i < \alpha_l$, то любая перестановка работ по невозрастанию величин $\alpha_i + w_i(\alpha_i^{-1} + H)$ задает оптимальное расписание.

Доказательство аналогично доказательству теоремы ??.

Рассмотрим несколько примеров, иллюстрирующих теорему.

Пример 4

Пусть $\Upsilon(t) = \sqrt{t}$ и параметры задачи $1|p_i = \alpha_i \Upsilon(s_i)|w_i C_i$ удовлетворяют следующему ограничению:

$$w_i \alpha_i^{-1} \leq w_l \alpha_l^{-1} \text{ для всех } J_i \text{ и } J_l \text{ таких, что } \alpha_i < \alpha_l.$$

Тогда любая перестановка работ по невозрастанию величин $\alpha_i + w_i \alpha_i^{-1}$ задает оптимальное расписание.

Пример 5

Пусть $\Upsilon(t) = t + \ln t$ и параметры задачи $1|p_i = \alpha_i \Upsilon(s_i)|w_i C_i$ удовлетворяют следующему ограничению:

$$w_i(\alpha_i^{-1} + 1) \leq w_l(\alpha_l^{-1} + 1) \text{ для всех } J_i \text{ и } J_l \text{ таких, что } \alpha_i < \alpha_l.$$

Тогда любая перестановка работ по невозрастанию величин $\alpha_i + w_i(\alpha_i^{-1} + 1)$ задает оптимальное расписание.

Пример 6

Пусть $\Upsilon(t) = 100 - t^2$ и параметры задачи $1|p_i = \alpha_i \Upsilon(s_i)|w_i C_i$ удовлетворяют следующим ограничениям:

$$\alpha_i < \frac{1}{20} \text{ для всех } J_i \in J,$$

$$w_i(\alpha_i^{-1} - 20) \leq w_l(\alpha_l^{-1} - 20) \text{ для всех } J_i \text{ и } J_l \text{ таких, что } \alpha_i < \alpha_l.$$

Тогда любая перестановка работ по невозрастанию величин $\alpha_i + w_i(\alpha_i^{-1} - 20)$ задает оптимальное расписание.

1.4 Ступенчатые функции

Рассмотрим задачу $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ с функциями длительности вида

$$\Upsilon(t) = \begin{cases} a_1 & \text{при} \quad t \leq b, \\ a_1 + a_2 & \text{при} \quad t > b, \end{cases} \tag{1.29}$$

где a_1, a_2, b — положительные рациональные числа.

Для удобства вычислений предположим, что все работы готовы к выполнению в момент времени $t_0 = 0$. Заметим, что если $\sum\limits_{i=1}^{n} \alpha_i a_1 \leq b$, то задача становится тривиальной. Поэтому пусть $\sum\limits_{i=1}^{n} \alpha_i a_1 > b$. Тогда

$$C_{max} = \sum_{i=1}^{n} \alpha_i a_1 + \sum_{i \in \{i | s_i > b\}} \alpha_i a_2. \tag{1.30}$$

Пусть $J_m \in J$ — работа с наибольшим α_i.

Лемма 2 *Для задачи* $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ *существует оптимальное расписание, в котором работа* J_m *начинает выполняться не позднее момента времени* b *и среди работ множества* $\{J_i \mid s_i \leq b\}$ *выполняется последней.*

Доказательство. Предположим, что в оптимальном расписании π работа J_m начинает выполняться после момента времени b. Пусть $J_k \in \{J_i \mid s_i \leq b\}$ и выполняется последней из работ этого множества. Рассмотрим перестановку π^*, в которой на месте работы J_m стоит работа J_k и наоборот. Заметим, что множества работ, которые выполняются после момента времени b, в расписаниях π и π^* отличаются только работами J_k и J_m соответственно. Обозначив пересечение этих множеств через J', получаем

$$C_{max}(\pi) = \sum_{i=1}^{n} \alpha_i a_1 + \sum_{i \in J'} \alpha_i a_2 + \alpha_m a_2 \geq \sum_{i=1}^{n} \alpha_i a_1 + \sum_{i \in J'} \alpha_i a_2 + \alpha_k a_2 = C_{max}(\pi^*).$$

Следовательно, расписание π^* является оптимальным.

Предположим, что в оптимальном расписании π работа J_m начинает выполняться до момента времени b и существуют работы из $\{J_i \mid s_i \leq b\}$, которые выполняются после нее. Пусть $J_k \in \{J_i \mid s_i \leq b\}$ и выполняется последней из работ этого множества. Рассмотрим перестановку π^*, в которой на месте работы J_m стоит работа J_k и наоборот. Так как $\alpha_m a_2 \geq \alpha_k a_2$, то момент начала каждой работы из $\{J_i \mid s_i \leq b\}$ кроме работы J_m, не увеличится. Обозначим через t время завершения работ из $\{J_i \mid s_i \leq b\}$ в расписании π. Тогда

$$b \geq s_k(\pi) = t - \alpha_k a_1 \geq t - \alpha_m a_1 = s_m(\pi^*).$$

Следовательно, $C_{max}(\pi) = C_{max}(\pi^*)$ и расписание π^* является оптимальным. Лемма **??** доказана.

Теорема 10 *Пусть $\Upsilon(t)$ — функция, заданная равенством (??). Тогда задача $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ является NP-трудной.*

Доказательство. Сведем к задаче $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ эталонную NP-трудную задачу СУММА РАЗМЕРОВ. Пусть $|I| = n$ и $y = \max\limits_{i \in I} e_i$, $Y = \sum\limits_{i \in I} e_i$. Положим $|J| = n+1$, $\alpha_0 = \frac{yb}{a_1 E}$ и $\alpha_i = \frac{e_i b}{a_1 E}$ для всех $1 \leq i \leq n$. Покажем, что в этой задаче расписание π, в котором $C_{max}(\pi) \leq \frac{b(Y+y)}{E} + \frac{a_2 b(Y-E)}{a_1 E}$, существует тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что сумма размеров его элементов равна E. По лемме (??) в оптимальном расписании имеем $J_0 \in \{J_i \mid s_i \leq b\}$ и $s_0 \geq s_i$ для всех $J_i \in \{J_i \mid s_i \leq b\}$. Пусть в расписании π работа J_0 начинает выполняться в момент времени $b_0 \leq b$. Тогда имеем

$$b \geq b_0 = \sum_{i \in \{i|s_i \leq b\} \setminus 0} p_i = a_1 \sum_{i \in \{i|s_i \leq b\} \setminus 0} \alpha_i = \sum_{i \in \{i|s_i \leq b\} \setminus 0} \frac{e_i b}{E}.$$

Следовательно,

$$\sum_{i\in\{i|s_i\le b\}\setminus 0} e_i \le E. \tag{1.31}$$

Вычислим длину расписания. Имеем

$$C_{max} = \sum_{i=0}^{n} \alpha_i a_1 + \sum_{i\in\{i|s_i>b\}} \alpha_i a_2 = \sum_{i=0}^{n} \frac{e_i b}{a_1 E} a_1 + \sum_{i\in\{i|s_i>b\}} \frac{e_i b}{a_1 E} a_2 =$$

$$\frac{b(Y+y)}{E} + \frac{a_2 b}{a_1 E} \sum_{i\in\{i|s_i\le b\}\setminus 0} e_i = \frac{b(Y+y)}{E} + \frac{a_2 bY}{a_1 E} - \frac{a_2 b}{a_1 E} \sum_{i\in\{i|s_i\le b\}\setminus 0} e_i. \tag{1.32}$$

Из (??) и (??) получаем, что $C_{max}(\pi) \le \frac{b(Y+y)}{E} + \frac{a_2 b(Y-E)}{a_1 E}$ тогда и только тогда, когда

$$\sum_{i\in\{i|s_i\le b\}\setminus 0} e_i = E.$$

Теорема ?? доказана.

Пусть a_1, α_i, b — целые положительные числа. Опишем алгоритм, который точно решает задачу $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ за $O(bn)$ элементарных операций.

Алгоритм 2.

Шаг 1. Находится работа $J_m \in J$ такая, что $\alpha_m = \max_{J_i\in J} \alpha_i$.

Шаг 2. Решается следующая оптимизационная задача.

$$\text{Найти } J' \subseteq J \setminus \{J_m\},$$

$$\text{которое максимизирует величину} \quad \sum_{i\in\{i|J_i\in J'\}} \alpha_i \tag{1.33}$$

$$\text{при условии, что} \quad \sum_{i\in\{i|J_i\in J'\}} a_1\alpha_i \le b. \tag{1.34}$$

Шаг 3. Строится следующая перестановка π: сначала в произвольном порядке выполняются работы множества J', затем работа J_m и наконец в произвольном порядке выполняются оставшиеся работы.

Теорема 11 *Пусть $\Upsilon(t)$ задается равенством (??). Тогда Алгоритм 2 точно решает задачу $1|p_i = \alpha_i\Upsilon(s_i)|C_{max}$ не более чем за $O(bn)$ элементарных операций.*

Доказательство. Из (??) следует, что целевая функция принимает минимальное значение в том случае, когда сумма $\sum\limits_{i\in\{i|s_i>b\}} \alpha_i$ принимает минимальное значение. Так как

$$\sum_{i\in\{i|s_i>b\}} \alpha_i = \sum_{J_i\in J} \alpha_i - \sum_{i\in\{i|s_i\le b\}} \alpha_i,$$

то для построения оптимального расписания требуется найти такое множество индексов работ $\{i \mid s_i \le b\}$, чтобы сумма $\sum\limits_{i\in\{i|s_i\le b\}} \alpha_i$ была максимальной. По лемме 2 имеем $J_m \in \{J_i \mid s_i \le b\}$ и работа J_m выполняется последней среди таких работ. Следовательно, $\sum\limits_{i\in\{i|s_i\le b\}\setminus m} \alpha_i \le b$, и поэтому, решив задачу (??) — (??) , найдем искомое множество. Задача (??) — (??) — известная задача о рюкзаке — может быть решена алгоритмом динамического программирования за $O(bn)$ элементарных операций. Трудоемкость второго шага и определяет трудоемкость всего алгоритма. Теорема ?? доказана.

Рассмотрим функцию $\Upsilon(t)$ следующего вида:

$$\Upsilon(t) = \begin{cases} a_1 + a_2, & \text{если } \ t < b, \\ a_1, & \text{если } \ t \ge b \end{cases} \tag{1.35}$$

где a_1, a_2, b — положительные числа.

Заметим, что задача $1|p_i = \alpha_i\Upsilon(s_i)|C_{max}$ не может быть хорошо определена, так как для любого набора α_i условие (??) не выполняется. Поэтому введем дополнительное ограничение. Пусть, начиная с момента времени 0, машина работает без простоев. Обозначим эту задачу $1|p_i = \alpha_i \Upsilon(s_i), w.i.|C_{max}$. Тогда имеем:

$$C_{max} = \sum_{i=1}^{n} \alpha_i a_1 + \sum_{i \in \{i|s_i<b\}} \alpha_i a_2.$$

Теорема 12 *Пусть функция $\Upsilon(t)$ задана равенством (??). Тогда задача $1|p_i = \alpha_i \Upsilon(s_i), w.i.|C_{max}$ является NP-трудной.*

Доказательство. Построим полиномиальное сведение задачи СУММА РАЗМЕРОВ к задаче $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$. Положим, $\alpha_i = \frac{e_i b}{(a_1+a_2)E}$ для всех i, $1 \leq i \leq n$. Покажем, что в этой задаче расписание π, в котором $C_{max}(\pi) \leq \frac{ba_1 Y}{(a_1+a_2)E} + \frac{a_2 b}{a_1+a_2}$, существует тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что сумма размеров его элементов равна E. Так как по условиям задачи машина с момента времени 0 работает без простоев, то выполнение последней работы из множества $\{i \mid s_i < b\}$ заканчивается не ранее момента времени b. Следовательно,

$$b \leq \sum_{i \in \{i|s_i<b\}} p_i = \sum_{i \in \{i|s_i<b\}} \alpha_i (a_1 + a_2) = \sum_{i \in \{i|s_i<b\}} \frac{e_i b}{E},$$

и

$$\sum_{i \in \{i|s_i<b\}} e_i \geq E. \tag{1.36}$$

Далее имеем

$$C_{max} = \sum_{i=1}^{n} \alpha_i a_1 + \sum_{i \in \{i|s_i<b\}} \alpha_i a_2 = \sum_{i=1}^{n} \frac{e_i b}{(a_1+a_2)E} a_1 + \sum_{i \in \{i|s_i<b\}} \frac{e_i b}{a_1 E} a_2$$

$$= \frac{bYa_1}{(a_1+a_2)E} + \frac{a_2 b}{(a_1+a_2)E} \sum_{i \in \{i|s_i<b\}} y_i \tag{1.37}$$

Из (??) и (??) получаем, что $C_{max}(\pi) \leq \frac{ba_1 Y}{(a_1+a_2)E} + \frac{a_2 b}{a_1+a_2}$ тогда и только тогда, когда

$$\sum_{i \in \{i|s_i<b\}} y_i = E.$$

Теорема 13 доказана.

Пусть a_1, a_2, α_i, b — целые положительные числа. Опишем алгоритм, который точно решает задачу $1|p_i = \alpha_i \Upsilon(s_i), w.i.|C_{max}$ за $O(bn)$ элементарных операций.

Алгоритм 3.

1. Решается следующая оптимизационная задача.

$$\text{Найти } J' \subseteq J, \text{ которое минимизирует величину } \sum_{J_i \in J'} \alpha_i$$

$$\text{при условии, что } \sum_{i \in \{J_i \in J'} (a_1 + a_2)\alpha_i \geq b.$$

2.Строится следующая перестановка π: сначала в произвольном порядке выполняются работы множества J' и затем в произвольном порядке оставшиеся работы.

Теорема 13 *Пусть функция $\Upsilon(t)$ задана равенством (??). Тогда алгоритм 3 позволяет находить точное решение задачи $1|p_i = \alpha_i \Upsilon(s_i), w.i.|C_{max}$ не более чем за $O(bn)$ элементарных операций.*

Доказательство аналогично доказательству теоремы ??.

В заключение параграфа рассмотрим задачу $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$, в которой функция $\Upsilon(t)$ имеет вид (??) и простои на машине допускаются.

Пусть порядок запуска работ на выполнение фиксирован. Тогда длина оптимального расписания относительно этого порядка находится следующим образом. Для каждой работы J_i вычисляются величины s_i и C_i, при условии, что работы выполняются без простоев. Если момент завершения последней работы меньше b, то расписание оптимально. В противном случае находится работа J_k такая, что $s_k < b \leq C_k$.

Тогда если $s_k + \alpha_k a_2 < b$, то в оптимальном расписании относительно заданного порядка работа J_k начинает выполняться в момент времени s_k в противном случае — в момент времени b.

Таким образом, как и в рассмотренных выше задачах, достаточно найти оптимальную перестановку π для нахождения оптимального расписания. Пусть $\pi = (\pi_1, \ldots, \pi_n)$ — некоторая перестановка, $k(\pi) = \max\{k \mid (a_1 + a_2) \sum_{i=1}^{k-1} \alpha_{\pi_i} < b\}$, $J^0(\pi) = \{\pi_i \mid i = 1, \ldots, k(\pi)\}$. Тогда длина кратчайшего расписания S_π, задаваемого перестановкой работ π определяется по формуле:

$$C_{max}(S_\pi) \sum_{i=1}^{n} \alpha_i a_1 + \sum_{i \in J^0(\pi)} \alpha_i a_2 + \min\{0; b + a_1 \alpha_{k(\pi)} - \sum_{i \in J^0(\pi)} \alpha_i (a_1 + a_2)\}.$$

Теорема 14 *Если функция $\Upsilon(t)$ задана равенством (??), то задача $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ является NP-трудной.*

Доказательство Сведем задачу СУММА РАЗМЕРОВ к задаче $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$. Положим $\alpha_i = \frac{y_i b}{(a_1+a_2)E}$ для всех i, $1 \le i \le n$. Проведя рассуждения, аналогичные рассуждениям в теоремах ?? и 14, получим, что в задаче $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ расписание π, в котором $C_{max}(\pi) \le \frac{b a_1 Y}{(a_1+a_2)E} + \frac{a_2 b}{a_1+a_2}$, существует тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что сумма размеров его элементов равна E. Теорема ?? доказана.

Пусть a_1, a_2, α_i и b — целые положительные числа. Опишем алгоритм, который точно решает задачу $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ за $O(bn)$ элементарных операций.

Алгоритм 4.

1. Решается следующая оптимизационная задача.

$$\text{Найти } J' \subseteq J, \text{ которое минимизирует величину } \sum_{J_i \in J'} \alpha_i \tag{1.38}$$

$$\text{при условии, что } \sum_{J_i \in J'} (a_1 + a_2)\alpha_i \geq b. \tag{1.39}$$

2. Строится перестановка π_1: сначала в произвольном порядке выполняются работы множества J' и затем в произвольном порядке выполняются оставшиеся работы.

3. Решается следующая оптимизационная задача.

$$\text{Найти } J' \subseteq J, \text{ которое максимизирует величину } \sum_{J_i \in J'} \alpha_i \tag{1.40}$$

$$\text{при условии, что } \sum_{J_i \in J'} (a_1 + a_2)\alpha_i \leq b. \tag{1.41}$$

4. Строится расписание π_2: сначала в произвольном порядке выполняются работы множества J' а затем с момента времени b в произвольном порядке выполняются оставшиеся работы.

5. Вычисляются значения функционалов $C_{max}(\pi_1)$ и $C_{max}(\pi_2)$. Перестановка, на которой значение функционала меньше, берется в качестве решения.

Теорема 15 *Пусть функция* $\Upsilon(t)$ *задана равенством* (??) . *Тогда алгоритм 3 точно решает задачу* $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ *за* $O(bn)$ *элементарных операций.*

Доказательство. Как было отмечено ранее, в оптимальном решении задачи $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ либо машина работает без простоев, либо машина заканчивает выполнять работы множества $J^0 \setminus \{k(\pi)\}$ не позднее момента времени b и затем с момента времени b выполняет оставшиеся работы. На шагах 1 и 2 находится оптимальная перестановка, когда машина работает без простоев. На шагах 3 и 4 находится оптимальная перестановка, когда моменту b предшествует простой. Полученные решения сравниваются и из них выбирается оптимальное.

Трудоемкость алгоритма определяется трудоемкостью решения задачи (??), (??) и задачи (??), (??) и равна $O(bn)$. Теорема ?? доказана.

Отметим, что в формулировках теорем 10, 12 и 14 не указано являются ли числа a_1, a_2, b частью входа задачи $1|p_i = \alpha_i \Upsilon(s_i)|C_{max}$ или нет. Однако доказательства теорем справедливы в обоих случаях.

Глава 2

Задачи теории расписаний на одной машине с длительностями, линейно или кусочно-линейно зависящими от времени

2.1 Задача минимизации максимального запаздывания с линейными функциями длительностей

В этой главе мы рассмотрим задачи на одной машине, в которых длительность работы J_i равна $p_i = \Upsilon_i(s_i)$. Пусть $\Upsilon_i(t) = a_i t + b_i$. Тогда, как было отмечено во введении, в задаче $1|p_i = a_i s_i + b_i|C_{max}$ перестановка, в которой работы упорядочены по неубыванию величин b_i/a_i является оптимальной. Рассмотрим задачу $1|p_i = a_i s_i + b_i|L_{max}$. Если $a_i = 0$ или $b_i = 0$ для всех работ, то перестановка, в которой работы упорядочены по неубыванию их директивных сроков является оптимальной. Первый случай рассмотрел Джексон [?], а второй — Мошеов [?].

Теорема 16 *Пусть все работы поступают на выполнение в момент $t_0 > 0$. Тогда задача $1|p_i = a_i s_i + b_i|L_{max}$ является NP-трудной уже* *в том случае, когда $a_i = 0$ для всех работ кроме одной и $d_i = D$ для всех работ с $a_i = 0$.*

Доказательство. Сведем к задаче $1|p_i = a_i s_i + b_i|L_{max}$ эталонную задачу ПРОИЗВЕДЕНИЕ РАЗМЕРОВ, сформулированную в параграфе 1.1. Пусть $E_0 = \prod_{i=1}^{n} e_i$. Определим подзадачу задачи $1|p_i = a_i s_i + b_i|L_{max}$ с $n+1$-й работой и $p_0 = 1$, $d_0 = t_0 E + 1$, $p_i = (e_i - 1)s_i$, $d_i = E_0(t_0 E + 1)/t_0 E$, для всех $1 \le i \le n$. Покажем, что в сформулированной задаче расписание S, при котором $L_{max} \le 0$ существует тогда и только тогда, когда существует подмножество I такое, что произведение размеров его элементов равно E.

1.Пусть существует подмножество I такое, что $\prod_{i \in I} e_i = E$. Тогда расписание S строится следующим образом. Сначала в произвольном порядке выполняются работы с номерами из множества I, затем работа J_0, и затем оставшиеся работы. В построенном расписании работы из множества I закончат выполнение в момент $C(I) = t_0 \prod_{i \in I} e_i = t_0 E$, работа J_0, соответственно, — в $C_0 = t_0 E + 1$, и последняя из оставшихся работ в момент $C(J) = E_0(t_0 E + 1)/t_0 E$. Следовательно, все работы будут выполнены до наступления директивного срока и $L_{max} = 0$.

2.Пусть существует расписание S такое, что $L_{max} \le 0$. Тогда в нем работа J_0 начинает выполняться в момент $t \le t_0 E$. Предположим, что $t < t_0 E$. Тогда выполнение всех работ заканчивается в момент $C(J) \le E_0(t+1)/t > E_0(t_0 E + 1)/t_0 E$, что противоречит предположению $L_{max} \le 0$. Следовательно, $t = t_0 E$. Если в процессе выполнения работ машина в какой-то момент простаивает, то опять получаем $C(J) > E_0(t_0 E + 1)/t_0 E$, и приходим к противоречию. Следовательно, в промежутке $[t_0, t_0 E]$ машина работает без простоев. Обозначим множество работ, выполняемых в этом промежутке, через I. Тогда

$t_0 E = t_0 \prod_{i\in I} e_i$ и подмножество I является искомым. Теорема ?? доказана.

2.2 Задача минимизации длины расписания с кусочно-линейными функциями длительностей

Рассмотрим задачу $1|p_i = a_i(s_i - d_i)^+ + b_i)|C_{max}$.

В дальнейшем назовем эту задачу задачей P. Заметим, что при оптимальном расписании машина будет работать без простоев. С учетом сказанного целевая функция может быть записана в виде $C_{max} = \sum_{i\in I} p_i$, а расписание будет полностью определяться последовательностью π запуска работ на выполнение. Далее расписанием для задачи P будем называть перестановку π, указывающую порядок запуска работ на выполнение. Через $C(\pi)$ будем обозначать длину расписания, заданного перестановкой π. Перестановку π, которая минимизирует величину $C(\pi) = \sum_{i\in I} p_i(\pi)$, будем называть оптимальной.

Утверждение 2 [?] *Пусть имеется такое расписание для первых k работ из множества J, $\sum_{i=1}^{k} p_i = t$, что $d_j \leq t$ для всех $j > k$. Тогда оптимальная последовательность оставшихся работ получается упорядочением их по возрастанию величин $(b_i - d_i a_i)/a_i$.*

Рассмотрим известную задачу $1||\Sigma w_i L_i$ минимизации суммарного взвешенного запаздывания на одной машине.

Задача $1||\Sigma w_i L_i$

При заданных целочисленных параметрах $\lambda_i \geq 0, \nu_i \geq 0, \delta_i \geq 0, J_i \in J$,найти величины β_i, удовлетворяющие условиям

$$\xi_i = max\{0, \beta_i + \lambda_i - \delta_i\},$$

$$\beta_i + \lambda_i \leq \beta_j, \text{ если } \beta_i < \beta_j,\ J_i \in J;$$

$$\lambda_i \lambda_j = 0, \text{ если } \beta_i = \beta_j,\ j \in I; J_i \in J;$$

$$\beta_i \geq 0$$

и минимизирующие величину $L = \sum\limits_{i \in I} \nu_i \xi_i$.

Параметры $\lambda_i, \delta_i, \nu_i$ обозначают длительность выполнения, директивный срок и штрафной вес работы J_i соответственно , β_i — переменная, обозначающая время начала выполнения работы J_i, ξ_i — величина запаздывания работы J_i. В [?] показано, что задача $1||\Sigma w_i L_i$ является NP-трудной в сильном смысле. Заметим, что для задачи $1||\Sigma w_i L_i$ расписание определяется перестановкой π, указывающей порядок запуска работ на выполнение.

По входу задачи $1||\Sigma w_i L_i$ при некотором $\varepsilon > 0$ получим вход задачи P_ϵ.Обозначим $\nu = \max\limits_{i \in I} \nu_i$, $\lambda = \max\limits_{J_i \in J} \lambda_i$ и положим

$$b_i = \lambda_i, J_i \in J,$$

$$d_i = \delta_i - \lambda_i, J_i \in J,$$

$$a_i = \nu_i \varepsilon, J_i \in J, \text{ где } \varepsilon > 0.$$

Пусть $\Lambda = \sum\limits_{J_i \in J} \lambda_i$ — длина расписания в задаче $1||\Sigma w_i L_i$, $r_i = s_i - d_i$. Так как $\delta_i \geq 0$, то $\xi_i \leq \Lambda$ для всех $J_i \in J$.

Пусть π — некоторая перестановка работ множества J,а $S_\pi(P_\epsilon) = \{s_i | J_i \in J\}$ и $S_\pi(1||\Sigma w_i L_i) = \{\beta_i | J_i \in J\}$ — расписания, определяемые перестановкой π в задачах P_ϵ и $1||\Sigma w_i L_i$ соответственно. Множества запаздывающих работ в расписаниях $S_\pi(P_\epsilon)$ и $S_\pi(1||\Sigma w_i L_i)$ соответственно обозначим через $J_\epsilon(\pi) = \{J_i \in J | s_i \geq d_i\}$ и $J(\pi) = \{J_i \in J | \beta_i + \lambda_i \geq \delta_i\}$. Пусть $\{J_{j_1}, ..., J_{j_k}\}$ — работы множества $J_\epsilon(\pi)$, перечисленные в порядке их выполнения.Тогда величина $\rho_{j_q} = \sum\limits_{i=1}^{q} r_{j_i} a_{j_i}$

задает увеличение времени окончания работы J_{j_q} в расписании $S_\pi(P_\epsilon)$ по сравнению с расписанием $S_\pi(1||\Sigma w_i L_i)$.

Лемма 3 *Для любой работы $J_{j_q} \in J_\epsilon(\pi)$ справедлива оценка*

$$\rho_{j_q} \le \Lambda((1+\nu\varepsilon)^q - 1). \qquad (2.1)$$

Доказательство. Из определения величин r_j, d_j, ρ_j, ξ_j для каждого $j, 1 \le j \le k$ имеем

$$r_{j_i} = s_{j_i} - d_{j_i} = \beta_{j_i} + \rho_{j_{i-1}} + \lambda_{j_i} - \delta_{j_i} = \xi_{j_i} + \rho_{j_{i-1}}, \qquad (2.2)$$

где $\rho_{j_0} = 0$. Оценку (**??**) для работ $J_{j_q} \in J_\epsilon(\pi)$ докажем индукцией по $q, 0 \le q \le k$. При $q = 0$ оценка (**??**) справедлива, так как $\rho_{j_0} = 0$. Пусть оценка (**??**) справедлива для $q = i-1$. Тогда при $q = i$ из соотношения (**??**) получаем

$$\rho_{j_i} = \rho_{j_{i-1}} + r_{j_i} a_{j_i} \le \rho_{j_{i-1}} + (\Lambda + \rho_{j_{i-1}})\nu\varepsilon$$

$$= \rho_{j_{i-1}}(1+\nu\varepsilon) + \Lambda\nu\varepsilon \le \Lambda((1+\nu\varepsilon)^i - (1+\nu\varepsilon) + \nu\varepsilon) = \Lambda((1+\nu\varepsilon)^i - 1).$$

Лемма **??** доказана.

Лемма 4 *Существует $\varepsilon_0 > 0$ такое, что при любом $\varepsilon < \varepsilon_0$ для любой перестановки π имеет место равенство $J(\pi) = J_\varepsilon(\pi)$.*

Доказательство. Поскольку длительности работ в задаче P_ϵ не меньше длительностей соответствующих работ в задаче $1||\Sigma w_i L_i$, то всякая работа, запаздывающая в расписании $S_\pi(1||\Sigma w_i L_i)$, является таковой и для расписания $S_\pi(P_\epsilon)$ и справедливо $J(\pi) \subseteq J_\epsilon(\pi)$. Докажем, что при достаточно малом ε имеет место обратное включение, т.е. $j_q \in J(\pi)$ для любой работы $j_q \in J_\epsilon(\pi)$. Искомое свойство вытекает из (??), неравенства $r_{j_i} \geq 0$ при $j_i \in J_\epsilon(\pi)$, целочисленности величин ξ_i и неравенств

$$\rho_{j_{i-1}} < 1, 1 \leq i \leq k, \tag{2.3}$$

вытекающих из (??) при малых ε. В качестве такого ε достаточно положить $\varepsilon < \varepsilon_0 = 1/\nu((1/\Lambda + 1)^{1/n-1} - 1)$.

Лемма **??** доказана.

Теорема 17 *Задача P является NP - трудной в сильном смысле.*

Доказательство. Для доказательства теоремы достаточно показать псевдополиномиальное сведение к задаче P какой-либо эталонной NP -трудной в сильном смысле задачи. В качестве последней выбрана задача $1||\Sigma w_i L_i$. Покажем, что решение задачи $1||\Sigma w_i L_i$ сводится к решению задачи P_ϵ. Более того, оптимальная перестановка π^* задачи P_ϵ является оптимальной перестановкой работ и для задачи SL, если ε достаточно мало.

Итак, пусть π^* -оптимальная перестановка работ в задаче P_ϵ при некотором $\varepsilon \geq 0$. Предположим, что она не оптимальна для задачи SL. Так как функционал $L(\pi)$ задачи $1||\Sigma w_i L_i$ принимает целые значения, то из нашего предположения следует неравенство

$$L(\pi^*) - L(\overline{\pi}) \geq 1, \tag{2.4}$$

где $\overline{\pi}$- оптимальная перестановка для задачи $1||\Sigma w_i L_i$. Докажем,что из (??) следует неравенство

$$C(\pi^*) - C(\overline{\pi}) > 0, \tag{2.5}$$

противоречащее оптимальности π^* для задачи P_ϵ и доказывающее оптимальность π^* для задачи $1||\Sigma w_i L_i$. Поскольку лемма **??** справедлива при $\varepsilon < \varepsilon_0$, пользуясь леммой ?? и (??), для любой перестановки π и любого $j_i \in J_\epsilon(\pi)$ получаем

$$r_{j_i} = \xi_{j_i} + \rho_{j_{i-1}}.$$

Согласно определению функционала C для любой перестановки π

$$\begin{aligned} C(\pi) &= \sum_{J_i \in J} p_i = \sum_{J_i \in J} (b_i + a_i(s_i - d_i)^+) \\ &= \sum_{J_i \in J} b_i + \sum_{J_i \in J_\epsilon(\pi)} a_i(s_i - d_i) = \sum_{J_i \in J} b_i + \sum_{J_{j_i} \in J_\epsilon(\pi)} a_{j_i} r_{j_i} \\ &= \sum_{J_i \in J} b_i + \sum_{J_{j_i} \in J_\epsilon(\pi)} \nu_{j_i} \varepsilon(\xi_{j_i} + \rho_{j_{i-1}}) = \sum_{J_i \in J} b_i + \varepsilon \sum_{J_{j_i} \in J_\epsilon(\pi)} \nu_{j_i}(\xi_{j_i} + \rho_{j_{i-1}}). \end{aligned}$$

Учитывая неравенство (??) и определение функционала L получаем

$$\begin{aligned} C(\pi^*) - C(\overline{\pi}) &= \varepsilon\Big(\sum_{J_j \in J(\pi^*)} \nu_j \xi_j - \sum_{J_j \in J(\overline{\pi})} \nu_j \xi_j\Big) + \varepsilon\Big(\sum_{J_{j_i} \in J(\pi^*)} \nu_{j_i} \rho_{j_{i-1}} - \sum_{J_{j_i} \in J(\overline{\pi})} \nu_{j_i} \rho_{j_{i-1}}\Big) \\ &\geq \varepsilon\Big(L(\pi^*) - L(\overline{\pi}) - \sum_{J_{j_i} \in J(\overline{\pi})} \nu_{j_i} \rho_{j_{i-1}}\Big) \geq \varepsilon - \varepsilon\nu \sum_{J_{j_i} \in J(\overline{\pi})} \rho_{j_{i-1}}. \end{aligned} \tag{2.6}$$

Оценим сверху величину $\varepsilon\nu \sum_{J_{j_i} \in J(\overline{\pi})} \rho_{j_{i-1}}$. Из (??) для любой перестановки π имеем

$$\begin{aligned} \varepsilon\nu \sum_{J_{j_i} \in J(\overline{\pi})} \rho_{j_{i-1}} &\leq \varepsilon\nu\Lambda\Big(\sum_{q=1}^{n} (1 + \nu\epsilon)^q - n\Big) \leq \varepsilon\nu\Lambda\Big(\frac{(1+\nu\varepsilon)^{n+1} - 1 - \nu\varepsilon}{\nu\varepsilon} - n\Big) \\ &= \Lambda((1+\nu\varepsilon)^{n+1} - 1 - \nu\varepsilon(n+1)) = \Lambda((1+\nu\varepsilon)^{n+1} - 1 - \nu\varepsilon - n\nu\varepsilon) \\ &= \Lambda\Big(\sum_{i=0}^{n+1} \begin{pmatrix} i \\ n+1 \end{pmatrix} (\nu\varepsilon)^i - 1 - \nu\varepsilon - n\nu\varepsilon\Big) = \Lambda \sum_{i=2}^{n+1} \begin{pmatrix} i \\ n+1 \end{pmatrix} (\nu\varepsilon)^i \\ &< \Lambda \sum_{i=2}^{n+1} \frac{(n+1)^i (\nu\varepsilon)^i}{i!} = \Lambda(\nu\varepsilon(n+1))^2 \sum_{i=2}^{n+1} \frac{(n+1)^{i-2}(\nu\varepsilon)^{i-2}}{i!}. \end{aligned}$$

Возьмем рациональное ε

$$\varepsilon = \frac{1}{\Lambda\nu^2(n+1)^2}.$$

Заметим, что $\varepsilon < \varepsilon_0$, а длина записи ε не превосходит $log_2 \Lambda \nu^2 (n+1)^2$ и ограничена полиномом от длины исходной информации задачи SL. Тогда имеем

$$\Lambda(\nu\varepsilon(n+1))^2 \sum_{i=2}^{n+1} \frac{(n+1)^{i-2}(\nu\varepsilon)^{i-2}}{i!} \le \varepsilon \sum_{i=2}^{n+1} \frac{1}{i!} \le (e-2)\varepsilon < 0.8\varepsilon.$$

Отсюда и из (**??**) следует (**??**). Теорема **??** доказана.

Рассмотрим задачу $1|p_i = a(s_i - d_i)^+ + b_i|C_{max}$.

Теорема 18 *Задача* $1|p_i = a(s_i - d_i)^+ + b_i|C_{max}$ *является* NP*-трудной.*

Рассмотрим известную задачу $1||\Sigma L_i$ минимизации суммарного запаздывания. В [?] доказана NP-трудность этой задачи. Повторяя доказательство теоремы **??**, убеждаемся в том, что задача минимизации суммарного запаздывания сводится к задаче $1|p_i = a(s_i - d_i)^+ + b_i|C_{max}$..

2.3 Полиномиальные и псевдополиномиальные алгоритмы для задач с общим директивным сроком

В дальнейшем будем предполагать, что директивные сроки для всех работ одинаковы и рассмотрим следующие задачи.

Задача PD. Решить задачу P при условии, что имеет место равенство $d_i = D$ для всех $J_i \in J$

Задача ($PD \ge$). Решить задачу PD при условии, что $b_i/a_i \ge b_j/a_j$ при всех $J_i, J_j \in J$ таких, что $b_i > b_j$ и $a_i > 0$, $a_j > 0$.

Задача ($PD \le$). Решить задачу PD при условии, что $b_i/a_i \le b_j/a_j$ при всех $J_i, J_j \in J$ таких, что $b_i > b_j$ и $a_i > 0$, $a_j > 0$.

Задача PDP. Решить задачу PD при условии, что существует промежуток времени $[t_0, t_1], t_0 < t_1$, когда машина находится на профилактике, т.е. не обслуживает ни одной работы, и $D \in [t_0, t_1]$.

В этом параграфе установим полиномиальную разрешимость задачи $(PD \geq)$ и построим "псевдополиномиальные" алгоритмы для остальных задач. Отметим, что согласно утверждению 2 справедливо следующее утверждение: если $D \leq 0$, то оптимальная последовательность работ в задаче PD и, следовательно, во всех ее частных случаях получается упорядочением работ по возрастанию величин b_i/a_i.
Пусть множество работ J занумеровано следующим образом:

$$\begin{aligned} &1.\text{Если } b_i < b_j, \text{ то } i < j. \\ &2.\text{Если } b_i = b_j \text{ и } b_i/a_i < b_j/a_j, \text{ то } i < j. \end{aligned} \tag{2.7}$$

Лемма 5 *Перестановка π, указывающая порядок запуска работ на выполнение в соответствии с порядком из* (2.7)*, задает оптимальное расписание.*

Доказательство проведем от противного. Пусть π — оптимальное расписание, в котором существуют работы J_i и J_j такие, что $j < i$,и работа J_i выполняется раньше работы J_j. Без ограничения общности можно считать, что работы J_i и J_j выполняются одна за другой. Если это не так, то всегда можно найти работы J_{i_1} и J_{j_1}, которые выполняются аналогичным образом и $j_1 < i_1$.

Если $D \leq s_i < s_j$, то работы J_i и J_j принадлежат множеству Q работ, не упорядоченных к моменту времени D. Работы множества Q упорядочим согласно порядку из (2.7). Тогда по определению задачи $(PD \geq)$ эти работы упорядочатся по возрастанию величин b_i/a_i, что согласно утверждению 2 влечет оптимальность расписания.

Пусть $t_0 = s_i < D < s_j$. Обозначим через $\bar{\pi}$ расписание, получающееся из π перестановкой работ J_i и J_j. Пусть $t_{ij} = p_i(\pi) + p_j(\pi)$ и $t_{ji} = p_j(\bar{\pi}) + p_i(\bar{\pi})$ Тогда

$$t_{ij} = b_i + b_j + (t_0 + b_i - D)a_j,$$

$$t_{ji} = b_i + b_j + (t_0 + b_j - D)^+ a_i.$$

Докажем, что $t_{ij} \geq t_{ji}$. Если $(t_0 + b_j - D)^+ = 0$, то $t_{ij} \geq t_{ji}$ выполнено. Если же $(t_0 + b_j - D)^+ > 0$, то имеем

$$t_{ij} - t_{ji} = (t_0 + b_i - D)a_j - (t_0 + b_j - D)a_i$$

$$= a_i a_j (b_i/a_i - (D - t_0)/a_i - b_j/a_j + (D - t_0)/a_j)$$

$$= a_i a_j (b_i/a_i(1 - (D - t_0)/b_i) - b_j/a_j(1 - (D - t_0)/b_j)) \geq 0$$

Следовательно, если в перестановке π переставить местами работы J_i и J_j,то длина расписания не увеличится. Наконец, если $s_i < s_j \leq D$, то от перемены последовательности выполнения работ J_i и J_j в перестановке π длина расписания не изменится. В каждом из разобранных случаев перестановка работ J_i и J_j не ухудшает значения функционала C. Отсюда следует существование оптимального расписания π, в котором работы выполняются по возрастанию величин b_i. Лемма **??** доказана.

Теорема 19 *Для решения задачи* $(PD \geq)$ *существует алгоритм трудоемкости* $O(n \log n)$.

Справедливость теоремы непосредственно вытекает из леммы **??**.

Сформулируем без доказательства три леммы.

Лемма 6 *Пусть π - оптимальное расписание в задаче PD, в котором выполнение работы i заканчивается до наступления директивного срока D. Тогда существует такое оптимальное расписание π^*, в котором работа i выполняется первой.*

Лемма 7 *Пусть π - оптимальное расписание в задаче PD, для работы i выполнено неравенство $s_i(\pi) \geq D$ и $b_i/a_i \geq b_j/a_j$ для любой работы $j \in I$ такой, что $s_j(\pi) \geq D$. Тогда существует такое оптимальное расписание π^* , в котором работа i выполняется последней.*

Лемма 8 *Пусть в оптимальном расписании π задачи PD для работы i справедливо неравенство $s_i \leq D$ и $b_i \leq b_j$ для всех j таких,что $s_j < s_i$. Тогда существует такое оптимальное расписание π^*, в котором работа i выполняется первой.*

Заметим что утверждения лемм **??** и **??** очевидны, а справедливость леммы **??** непосредственно вытекает из утверждения **??**. Перейдем к построению алгоритмов.

Пусть множество работ I занумеровано следующим образом:

$$
\begin{aligned}
&1.\text{Если } b_i/a_i < b_j/a_j, \text{ то } i < j.\\
&2.\text{Если } b_i/a_i = b_j/a_j \text{ и } b_i > b_j, \text{ то } i < j.
\end{aligned}
\tag{2.8}
$$

Ниже в этом параграфе будем предполагать, что работы множества I занумерованы в порядке,заданном в (**??**). Через $\langle k, d, l_0, l_1 \rangle$ обозначим частную задачу PDP, в которой множество работ состоит из первых k работ множества I, общий директивный срок равен d, а $[l_0, l_1]$ — интервал профилактики. Пусть $\langle n, D, t_0, t_1 \rangle$ - исходная задача PDP, $0 \leq t_0 \leq D \leq t_1$.Рассмотрим двухпараметрическое семейство частных задач PDP : $\{\langle k, \Delta \rangle | k = 1, ..., n; \Delta = 0, ...t_0\}$, где $\langle k, \Delta \rangle = \langle k, D -$

$\Delta, t_0 - \Delta, t_1 - \Delta\rangle$. Через $C^*_{max}(k, \Delta)$ обозначим оптимальное значение целевой функции C_{max} в задаче $\langle k, \Delta\rangle$.Для простоты выкладок будем предполагать, что функция $C^*_{max}(k, \Delta)$ определена для всех значений параметров k и Δ, причем $C^*_{max}(k, \Delta) = \infty$ для $k \notin [1, n]$ или $\Delta \notin [0, t_0]$.

Лемма 9 *Для задачи PDP справедливы следующие рекуррентные соотношения динамического программирования:*

$$C^*_{max}(1, \Delta) = \begin{cases} b_1 & \text{при } b_1 \leq t_0 - \Delta \\ t_1 - \Delta + b_1 + (t_1 - D)a_1 & \text{при } b_1 > t_0 - \Delta, \end{cases} \tag{2.9}$$

$$C^*_{max}(k, \Delta) = \min\{C_1, C_2\}, \tag{2.10}$$

где

$$C_1 = b_k + C^*_{max}(k - 1, \Delta + b_k),$$

$$C_2 = C'_{max} + b_k + (C'_{max} - D + \Delta)a_k,$$

$$C'_{max} = \max\{t_1 - \Delta, C^*_{max}(k - 1, \Delta)\}.$$

Доказательство. Соотношение (??) позволяет вычислить длину оптимального расписания, когда требуется выполнить одну работу. Для обоснования справедливости соотношения (??) предположим, что значения всех величин из правой части выражения (??) уже известны. Тогда по леммам ?? и ?? для работы k существует оптимальное расписание, в котором эта работа выполняется либо первой до интервала профилактики, либо последней после этого интервала. В первом случае оптимальное расписание остальных работ получается решением задачи $\langle k - 1, \Delta + b_k\rangle$, и, следовательно, $C^*_{max}(k, \Delta) = C_1$. Во втором случае работа k начинает выполняться в момент времени $t_1 - \Delta$,если предшествующие ей работы завершились до момента времени $t_0 - \Delta$, или в момент времени $C^*_{max}(k - 1, \Delta)$ — в противном случае, и тогда

$C^*_{max}(k,\Delta) = C_2$. Лемма **??** доказана.

Пусть $D^+ = \max\{0, D\}$.

Теорема 20 *Задача PDP с целочисленными исходными данными, n работами и директивным сроком D разрешима с трудоемкостью $O(nD^+ + n\log n)$.*

Доказательство. Действительно,из формул (??),(??) следует, что для получения оптимального расписания задачи $\langle n, 0\rangle$ достаточно рассмотреть $n(t_0+1)$ задач $\langle k, \Delta\rangle$. Кроме того, для получения исходной нумерации работ требуется $O(n\log n)$ операций. Следовательно, решение задачи PDP находится за $O(nD^+ + n\log n)$ операций. Теорема **??** доказана.

Через $\langle k, d\rangle'$ обозначим частную задачу PD, в которой множество работ состоит из первых k работ множества I и общий директивный срок равен d. Пусть $\langle n, D\rangle'$ - исходная задача PD. Через $C^*_{max}(k,d)$ обозначим оптимальное значение целевой функции C_{max} в задаче $\langle k, d\rangle$.Для простоты выкладок будем предполагать, что функция $C^*_{max}(k,d)$ определена для всех значений параметров k и d, причем $C^*_{max}(k,d) = \infty$, если $k \notin [1,n]$ или $d \notin [0,D]$.

Лемма 10 *Для задачи PD справедливы следующие рекуррентные соотношения динамического программирования :*

$$C^*_{max}(1,d) = b_1; \tag{2.11}$$

$$C^*_{max}(k,d) = \min \begin{cases} C^*_{max}(k-1, d-b_k) + b_k \\ C^*_{max}(k-1,d) + b_k + a_k(C^*_{max}(k-1,d) - d)^+ \\ \min\limits_{i\in(0,b_k)} \max \begin{cases} d+i \\ C^*_{max}(k-1,d,d-b_k+i,d+i) \end{cases} \end{cases}, \quad (2.12)$$

где $C^*_{max}(k-1,d,d-b_k+i,d+i)$ *является оптимальным решением частной задачи* PDP $\langle k-1,d,d-b_k+i,d+i\rangle$.

Доказательство. Соотношение (??) позволяет вычислить длину оптимального расписания, когда требуется выполнить одну работу. Предположим, что значения всех величин из правой части выражения (??) уже известны. Тогда,если работа k в оптимальном расписании выполняется до директивного срока, то по лемме **??** эта работа выполняется первой, и оптимальное расписание остальных работ получается решением задачи $\langle k-1, d-b_k\rangle'$, и, следовательно, $C^*_{max}(k,d) = C^*_{max}(k-1,d-b_k)+b_k$. Если работа k в оптимальном расписании выполняется после директивного срока, то по лемме **??** она выполняется последней, и тогда $C^*_{max}(k,d) = C^*_{max}(k-1,d)+b_k+a_k(C^*_{max}(k-1,d)-d)^+$. Наконец,если работа k в оптимальном расписании начинает выполняться до директивного срока,а заканчивает выполняться после него,то оптимальное расписание остальных работ получается решением задачи $\langle k-1,d,d-b_k+i,d+i\rangle$ для некоторого $i\in(0,b_k)$, и, следовательно, $C^*_{max}(k,d) = \min\limits_{i\in(0,b_k)} \max\{d+i, C^*_{max}(k-1,d,d-b_k+i,d+i)\}$ Лемма **??** доказана.

Теорема 21 *Задача* PD *с целочисленными исходными данными,* n *работами и директивным сроком* D *разрешима с трудоемкостью* $O(nD^+p^2+n\log n)$, *где* $p = \max\limits_{J_i\in J} b_i$.

Доказательство. Заметим, что при фиксированных значениях d и k имеется не более p^2 различных задач $\langle k-1, d, d-b_k+i, d+i\rangle$. Поэтому из (??),(??) следует, что для получения оптимального расписания задачи $\langle n, D\rangle'$ достаточно решить не более nD^+p^2 задач $\langle k-1, d, d-b_k+i, d+i\rangle$. и не более nD^+ задач $\langle k, d\rangle'$. Кроме того, для получения исходной нумерации работ надо выполнить $O(n\log n)$ операций. Следовательно, решение задачи PD находится за $O(nD^+p^2 + n\log n)$ операций. Теорема **??** доказана.

Лемма 11 *Для задачи* $(PD \leq)$*справедливы следующие рекуррентные соотношения динамического программирования:*

$$C^*_{max}(1, d) = b_1; \tag{2.13}$$

$$C^*_{max}(k, d) = \min\begin{cases} C^*_{max}(k-1, d-b_k) + b_k \\ C^*_{max}(k-1, d) + b_k + a_k(C^*_{max}(k-1, d) - d)^+ \end{cases}. \tag{2.14}$$

Доказательство. Соотношение (??) позволяет вычислить длину оптимального расписания, когда требуется выполнить одну работу. Для обоснования справедливости соотношения (??) предположим, что значения всех величин из правой части этого выражения уже известны.Заметим,что так как работы занумерованы в порядке, указанном (??), то для задачи $(PD \leq)$ эти работы занумерованы по убыванию b_i. Тогда если работа k в оптимальном расписание начинает выполняться до директивного срока, то по лемме **??** она выполняется первой, и оптимальное расписании остальных работ получается решением задачи $\langle k-1, d-b_k\rangle'$, и, следовательно, $C^*_{max}(k, d) = C^*_{max}(k-1, d-b_k) + b_k$. Если работа k в оптимальном расписании выполняется после директивного срока, то по лемме **??** она выполняется последней, и тогда $C^*_{max}(k, d) = C^*_{max}(k-1, d) + b_k + a_k(C^*_{max}(k-1, d) - d)^+$. Лемма **??** доказана.

Теорема 22 *Задача* $(PD \leq)$ *с целочисленными исходными данными, n работами и директивным сроком D разрешима с трудоемкостью* $O(nD^+ + n\log n)$.

Доказательство. Действительно, из (??),(??) следует, что для получения оптимального расписания задачи $\langle n, D\rangle'$ достаточно решить nD^+ задач $\langle k, d\rangle'$. Кроме того, для получения исходной нумерации работ требуется выполнить $O(n\log n)$ операций. Следовательно, решение задачи $(PD \leq)$ находится за $O(nD^+ + n\log n)$ операций. Теорема ?? доказана.

2.4 Сложность задач с общим директивным сроком

В этом параграфе установим NP-трудность задач $PD, PDP, (PD \leq)$. Рассмотрим известную задачу Сумма размеров, сформулированную в параграфе 1.1.

По входу задачи СУММА РАЗМЕРОВ получим вход задачи PD. Обозначим $e = \max\limits_{i\in I} e_i, E_0 = \sum\limits_{i\in I} e_i$ и положим

$$b_0 = 1;$$

$$a_0 = \mu^{-1};$$

$$b_i = \mu e_i, i \in I;$$

$$a_i = \varepsilon e_i, i \in I;$$

$$D = \mu E,$$

где $\varepsilon = e^{-2}n^{-2}$, $\mu = \varepsilon(28E_0 n)^{-1}$.

Для доказательства NP-трудности задачи PD нам потребуется следующая лемма.

Лемма 12 *Пусть* $K = \{J_1, \dots, J_k\}$ *— список нерасписанных работ к моменту времени* $t > D$, *и пусть начиная с момента* t *работы выполняются последовательно по порядку номеров в списке* K. *Тогда при любом* $l = 1, \dots, k$

$$C_l = D + (T - D)\prod_{i=1}^{l}(1 + a_i) + \sum_{i=1}^{l}(b_i \prod_{j=i+1}^{l}(1 + a_j)). \tag{2.15}$$

Доказательство. Из определения величин $\overline{t_i}, p_i$ и условия леммы для каждого $i = 1, \dots, k$ имеем

$$C_i = C_{i-1} + b_i + (C_{i-1} - D)a_i. \tag{2.16}$$

Утверждение леммы докажем индукцией по q $1 \leq q \leq k$. При $q = 1$ имеем $C_1 = D + (T - D)(1 + a_1) + b_1 = T + b_1 + (T - D)a_1$ и равенство (??) выполняется. Пусть для $q = l - 1$ равенство (??) справедливо. Тогда из (??) при $q = l$ получаем

$$\begin{aligned}
C_l &= C_{l-1} + b_l + (C_{l-1} - D)a_l \\
&= D + (T - D)\prod_{i=1}^{l-1}(1 + a_i) + \sum_{i=1}^{l-1}(b_i \prod_{j=i+1}^{l-1}(1 + a_j)) + b_l \\
&+(D + (T - D)\prod_{i=1}^{l-1}(1 + a_i) + \sum_{i=1}^{l-1}(b_i \prod_{j=i+1}^{l-1}(1 + a_j)) - D)a_l \\
&= D + (T - D)\prod_{i=1}^{l-1}(1 + a_i) + a_l(T - D)\prod_{i=1}^{l-1}(1 + a_i) \\
&+ \sum_{i=1}^{l-1}(b_i \prod_{j=i+1}^{l-1}(1 + a_j)) + a_l \sum_{i=1}^{l-1}(b_i \prod_{j=i+1}^{l-1}(1 + a_j)) + b_l \\
&= D + (T - D)(\prod_{i=1}^{l-1}(1 + a_i))(1 + a_l) + (\sum_{i=1}^{l-1}(b_i \prod_{j=i+1}^{l-1}(1 + a_j)))(1 + a_l) + b_l \\
&= D + (T - D)\prod_{i=1}^{l}(1 + a_i) + \sum_{i=1}^{l}(b_i \prod_{j=i+1}^{l}(1 + a_j)).
\end{aligned}$$

Лемма **??** доказана.

Теорема 23 *Задача PD является NP-трудной.*

Доказательство. Для доказательства утверждения теоремы достаточно показать полиномиальное сведение к задаче PD какой-либо эталонной NP -трудной задачи. В качестве последней выберем задачу СУММА РАЗМЕРОВ, а преобразование входа задачи СУММА РАЗМЕРОВ ко входу задачи PD показано в начале этого параграфа. Покажем, что решение задачи СУММА РАЗМЕРОВ сводится к решению задачи PD.

Пусть π — оптимальная перестановка в задаче PD и пусть Q — множество работ, выполняющихся без нарушения директивного срока. Рассмотрим два случая. Пусть работа J_0 принадлежит Q. Так как $b_0 \geq b_i$ для любой работы $J_i \in J$, можно считать, что среди работ из Q работа J_0 выполняется последней. Если в расписании, заданном перестановкой π, работа J_0 начинает обслуживаться в момент времени μE, то множество работ $Q \setminus \{J_0\}$ является решением задачи СУММА РАЗМЕРОВ. Предположим, что работа J_0 в оптимальной перестановке π начинает обслуживаться в момент времени T, $T < \mu E$. Покажем, что тогда не существует $J \subseteq I$ такого, что $\sum\limits_{i \in J} e_i = E$. Предположим противное, т.е. пусть такое подмножество J существует. Построим перестановку π^* следующим образом. Пусть первыми выполняются работы из множества J, затем — работа J_0, и, наконец, оставшиеся работы в произвольном порядке. Будем считать, что работы в перестановке π занумерованы в порядке $j_0, j_1, \dots, j_n$, а в перестановке π^* — в порядке $k_0, k_1, \dots, k_n$. Работы множества J будут завершены в момент времени μE, так как по предположению $\sum\limits_{i \in J} b_i = \sum\limits_{i \in J} \mu e_i = \mu E$. По лемме ?? получаем

$$C_{max}(\pi) = D + (T+1-D)\prod_{j_i\in I\setminus Q}(1+a_{j_i}) + \sum_{j_i\in I\setminus Q} b_{j_i}(\prod_{l=i}^{n}(1+a_{j_l})),$$

$$C_{max}(\pi^*) = D + \prod_{k_i\in I\setminus J}(1+a_{k_i}) + \sum_{k_i\in I\setminus J} b_{k_i}(\prod_{l=i}^{n}(1+a_{k_l})).$$

Следовательно,

$$C_{max}(\pi) - C_{max}(\pi^*) = (T+1-D)\prod_{j_i\in I\setminus Q}(1+a_{j_i}) + \sum_{j_i\in I\setminus Q} b_{j_i}(\prod_{l=i}^{n}(1+a_{j_l}))$$

$$-\prod_{k_i\in I\setminus J}(1+a_{k_i}) - \sum_{k_i\in I\setminus J} b_{k_i}(\prod_{l=i}^{n}(1+a_{k_l}))$$

$$\geq (1-(D-T))\prod_{j_i\in I\setminus Q}(1+a_{j_i}) - \prod_{k_i\in I\setminus J}(1+a_{k_i}) - \sum_{k_i\in I\setminus J} b_{k_i}(\prod_{l=i}^{n}(1+a_{k_l}))$$

$$\geq (1-(D-T))\prod_{j_i\in I\setminus Q}(1+\varepsilon e_{j_i}) - \prod_{k_i\in I\setminus J}(1+\varepsilon e_{k_i}) - \sum_{k_i\in I\setminus J} \mu e_{k_i}(\prod_{l=i}^{n}(1+\varepsilon e_{k_l}))$$

$$\geq 1-(D-T) + (1-(D-T))\sum_{j_i\in I\setminus Q}\varepsilon e_{j_i} - 1 - \sum_{k_i\in I\setminus J}\varepsilon e_{j_i}$$

$$-\sum_{i=2}^{n}\begin{pmatrix} n \\ i \end{pmatrix}(\varepsilon e)^i - \sum_{k_i\in I\setminus J}\mu e_{k_i}(\prod_{l=i}^{n}(1+\varepsilon e_{k_l})). \tag{2.17}$$

Оценим сверху $\sum_{i=2}^{n}\begin{pmatrix} n \\ i \end{pmatrix}(\varepsilon e)^i$.

$$\sum_{i=2}^{n}\begin{pmatrix} n \\ i \end{pmatrix}(\varepsilon e)^i \leq \varepsilon\sum_{i=2}^{n}\frac{n^i e^i \varepsilon^{i-1}}{i!} \leq \varepsilon\sum_{i=2}^{n}\frac{n^{1-i}e^{1-i}}{i!} < 0.8\varepsilon \tag{2.18}$$

Оценим сверху $\sum_{k_i\in I\setminus J}\mu e_{k_i}(\prod_{l=i}^{n}(1+\varepsilon e_{k_l}))$. С учетом (??) получаем

$$\sum_{k_i\in I\setminus J}\mu a_{k_i}(\prod_{l=i}^{n}(1+\varepsilon e_{k_l})) \leq \varepsilon\sum_{k_i\in I\setminus J}\frac{1}{28n}(1+\sum_{i=1}^{n}\varepsilon e_i + 0.8\varepsilon)$$

$$\leq \frac{1}{28}\varepsilon(1+\frac{1}{en}+\frac{0.8}{e^2n^2}) \leq 0.1\varepsilon. \tag{2.19}$$

Из (??), используя (??),(??) имеем

$$C_{max}(\pi) - C_{max}(\pi^*) \geq 1 - (D-T) + (1-(D-T)) \sum_{j_i \in I \setminus Q} \varepsilon e_{j_i} - 1 - \sum_{k_i \in I \setminus J} \varepsilon e_{j_i} - 0.9\varepsilon.$$

Так как $D - T \leq D = \mu B = \dfrac{\varepsilon}{28n}$, а $\sum_{j_i \in I \setminus Q} e_{j_i} - \sum_{k_i \in I \setminus J} e_{j_i} \geq 1$ получаем

$$C_{max}(\pi) - C_{max}(\pi^*) \geq 0.1\varepsilon - \frac{\varepsilon}{28n} - \frac{\varepsilon}{28en^2} > 0,$$

что противоречит оптимальности π. Пусть теперь π оптимальная перестановка такая, что работа J_0 не принадлежит Q. Так как e_i — целые, работа J_0 начинает выполняться не раньше момента времени $\mu E + \mu$, и

$$C_{max}(\pi) \geq \mu E + \mu + 1 + \mu a_0 > 2.$$

Пусть π^* перестановка в которой работа J_0 выполняется первой, а затем — оставшиеся работы в произвольном порядке. Тогда по лемме **??**, используя (??), (??), получаем

$$C_{max}(\pi^*) = D + (1-D) \prod_{i=1}^{n} (1+a_i) + \sum_{i=1}^{n} (b_i \prod_{j=i+1}^{n} (1+a_j)) \leq D + 1 + \sum_{i=1}^{n} \varepsilon e_i + 0.9\varepsilon < 2,$$

что противоречит оптимальности π. Теорема **??** доказана.

Теорема 24 *Задача* $(PD \leq)$ *является* NP*-трудной.*

Справедливость теоремы **??** вытекает из того, что параметры задачи PD, построенной в ходе доказательства теоремы **??**, удовлетворяют следующему условию: если $b_i > b_j$, то $b_i/a_i \leq b_j/a_j$ для любых $J_i, J_j \in J$.

Теорема 25 *Задача* PDP *является* NP*-трудной.*

Действительно, положив в задаче PD профилактику машины в промежутке $[1 + \mu E, 1 + \mu E + \mu]$ и, повторяя рассуждения из доказательства теоремы **??**, получаем требуемое.

Глава 3

Задачи теории расписаний на параллельных машинах и в многооперационных системах с линейно растущими длительностями

3.1 Основные обозначения и определения

Пусть задано множество работ $J = \{J_1, J_2, \ldots, J_n\}$. Каждая работа $J_i \in J$ состоит из последовательности операций $O_{i,1}, O_{i,2}, \ldots O_{i,k(i)}$. Обозначим через $s_{i,m}$ — момент начала выполнения операции $O_{i,m}$. Тогда длительность этой операции равна $p_{i,m} = \alpha_{i,m}\Upsilon(s_{i,m})$. Все работы поступают на выполнение в момент времени $t = 1$. Прерывания при выполнении работ не допускаются. Операции выполняются на M машинах. В этой главе рассмотрим следующие задачи теории расписаний.

Задача P (система с параллельными машинами). Каждая работа $i \in I$ состоит из одной операции, которая может выполняться на любой машине

Задача F (система потокового типа). Каждая работа $J_i \in J$ состоит из M последовательных операций. Операция $O_{i,m}$ выполняется на машине m. Порядок выполнения операций одинаков для всех работ и определяется последовательностью машин $1, 2, \ldots, M$.

Задача O (система открытого типа). Каждая работа $J_i \in J$ состоит из M операций. Операция $O_{i,m}$ выполняется на машине m. Порядок выполнения операций произвольный.

Как и впервых двух главах для обозначения конкретной задачи для удобства будет использоваться запись $\alpha|\beta|\gamma$. В отличие от одномашинных задач в первом поле α связанном с ограничениями на машины будут стоять два символа α_1 и α_2, каждое из которых обозначает следующее:
если $\alpha_1 = P$, то работы выполняются в системе с параллельными машинами;
если $\alpha_1 = F$, то работы выполняются в системе потокового типа;
если $\alpha_1 = O$, то работы выполняются в системе открытого типа.
Если α_2 является натуральным числом, тогда число машин равно α_2. Если $\alpha_2 = \circ$, то число машин произвольно. Значения записей в полях β и γ остаются прежними.

3.2 Вспомогательный результат

В этом параграфе рассмотрим вспомогательную задачу, которую в дальнейшем используем для доказательства NP-трудности задач теории расписаний, изучаемых в настоящей статье.

Задача 4-ПРОИЗВЕДЕНИЕ.
Заданы множество I, состоящее из $4n$ элементов, граница $E \in Q^+$ и

"размеры" $e_i \in Q^+, e_i \geq 1$ всех элементов, $i \in I$, причем $\sqrt[5]{E} < e_i < \sqrt[3]{E}$ и $\prod\limits_{i \in I} e_i = E^n$. Можно ли I разбить на n непересекающихся подмножеств $I_1, I_2, \ldots, I_n$ таких, что $\prod_{i \in I_j} e_i = E$, $1 \leq j \leq n$.

Отметим, что указанные ограничения на размеры элементов приводят к тому, что каждое множество I_j должно содержать ровно четыре элемента из множества I.

Теорема 26 *Задача 4-ПРОИЗВЕДЕНИЕ является NP-полной в сильном смысле.*

Доказательство. Поскольку за полиномиальное время легко проверить, что заданное разбиение множества I удовлетворяет необходимым требованиям, то легко видеть, что задача 4-ПРОИЗВЕДЕНИЕ лежит в классе NP. Убедимся в том, что данная задача является NP-полной в сильном смысле. Для этого сведем следующую NP-полную в сильном смысле задачу ТРЕХМЕРНОЕ СОЧЕТАНИЕ к задаче 4-ПРОИЗВЕДЕНИЕ.

Пусть $W = \{w_1, w_2, \ldots, w_q\}$, $X = \{x_1, x_2, \ldots, x_q\}$, $Y = \{y_1, y_2, \ldots, y_q\}$ и $\Phi \subseteq W \times X \times Y$, определяют произвольную индивидуальную задачу из задачи ТРЕХМЕРНОЕ СОЧЕТАНИЕ. Не ограничивая общности, можно считать, что $|\Phi| \geq q$. Построим множество I из $4|\Phi|$ элементов. Каждому элементу $g \in W \cup X \cup Y$ будет соответствовать $N(g)$ элементов из множества A, где $N(g)$ — число троек из Φ, в которые входит g. Элементы, соответствующие фиксированному $g \in W \cup X \cup Y$, обозначим через $g[1], g[2], \ldots, g[N(g)]$. Размеры этих элементов определим следующим образом:

$$
\begin{aligned}
&\eta(w_i[1]) = \varphi_Q^{10}\varphi_i, && 1 \le i \le q;\\
&\eta(w_i[l]) = \varphi_Q^{11}\varphi_i, && 1 \le i \le q, \quad 2 \le l \le N(w_i);\\
&\eta(x_j[1]) = \varphi_Q^{10}\varphi_{q+j}, && 1 \le j \le q;\\
&\eta(x_j[l]) = \varphi_Q^{11}\varphi_{q+j}, && 1 \le j \le q, \quad 2 \le l \le N(x_j);\\
&\eta(y_k[1]) = \varphi_Q^{10}\varphi_{2q+k}, && 1 \le k \le q;\\
&\eta(y_k[l]) = \varphi_Q^{8}\varphi_{2q+k}, && 1 \le k \le q, \quad 2 \le l \le N(y_k),
\end{aligned}
$$

где φ_k есть k-ое простое число, а $Q = 3q+1$. Единственный элемент, соответствующий каждой тройке $\phi_l = (w_i, x_j, y_k) \in Z$, обозначается через u_l, а его размер $\eta(u_l)$ зависит от индексов элементов тройки следующим образом, $\eta(u_l) = \frac{\varphi_Q^{12}}{\varphi_i\varphi_j\varphi_k}$. В качестве границы возьмем $E = \varphi_Q^{42}$. Из неравенства Чебышева для распрелеления простых чисел [?] следует, что $k < \varphi_k < 10k\ln k/9$, выбранные размеры элементов удовлетворяют неравенству $\sqrt[5]{E} < \eta(a) < \sqrt[3]{E}$. Размер максимального элемента не превосходит $c_1q^{42}\ln q + c_2$ и ограничен полиномом от общего числа элементов задачи ТРЕХМЕРНОЕ СОЧЕТАНИЕ.

Для доказательства сильной NP-полноты задачи 4-ПРОИЗВЕДЕНИЕ достаточно показать, что искомое 4-разбиение существует тогда и только тогда, когда Φ содержит трехмерное сочетание.

1. Предположим сначала, что имеется трехмерное сочетание $\Phi' \subseteq \Phi$. Соответствующее 4-разбиение получается из $|\Phi|$ подмножеств, каждое из которых содержит 4 элемента — один вида u_l, один вида w_i, один вида x_j, один вида y_k, где $(w_i, x_j, y_k) = \phi_l \in \Phi$. Если $\phi_l \in \Phi'$, то u_l группируем с элементами $w_i[1], x_j[1], y_k[1]$. Если $\phi_l \notin \Phi'$, то u_l группируем с элементами $w_i[r], x_j[r], y_k[r], r \neq 1$. Заметим, что в полученном разбиении произведение элементов в каждой четверке равно $E = \varphi_Q^{42}$.

2. Предположим, что задано 4-разбиение нужного вида. Рассмо-

трим любое 4-множество из этого 4-разбиения. Размер любого элемента $g \in W \cup X \cup Y$ имеет вид $\eta(g) = \phi_Q^r \phi_g$, где r — целое, а φ_g — простое число не равное φ_Q. Так как $E = \varphi_Q^{42}$, то в 4-разбиение входит элемент u_l. Размер любого элемента u_l имеет вид $\eta(u_l) = \frac{\varphi_Q^{12}}{\varphi_i \phi_j \varphi_k}$, где $1 \leq i \leq q$, $q+1 \leq j \leq 2q$, $2q+1 \leq i \leq 3q$. Поэтому в 4-разбиение входит по одному элементу, соответствующему каждому из составляющих тройки, причем все три элемента либо имеют вид $w_i[1], x_j[1], y_k[1]$, либо $w_i[r], x_j[r], y_k[r], r \neq 1$. Пусть w_i, x_j, y_k обозначают соответствующие элементы множеств W, X и Y, а $\phi_l = (w_{i'}, x_{j'}, y_{k'})$ обозначает соответствующую тройку из Φ. Из простоты чисел φ_i, φ_j, φ_k, $\varphi_{i'}$, $\varphi_{j'}$, $\varphi_{k'}$ следует, что $i = i'$, $j = j'$, $k = k'$. Таким образом, тройки вида $w_i[1], x_j[1], y_k[1]$, входящие в заданное 4-разбиение, содержатся в Φ и образуют искомое трехмерное сочетание. Теорема **??** доказана.

3.3 Параллельные машины. NP-трудность

В этом параграфе покажем NP-трудность задач построения расписаний на параллельных машинах с длительностями работ $p_i = \alpha_i s_i$ и с критериями C_{max} и ΣC_i.

Рассмотрим задачу $P2|p_i = \alpha_i s_i|C_{max}$. Сформулируем несколько вспомогательных утверждений.

Утверждение 3 *[?] Пусть с момента времени t на некоторой машине без простоя выполняется n работ. Тогда время C_n завершения выполнения всех работ равно*

$$C_n = t \prod_{i=1}^{n} (\alpha_i + 1) \tag{3.1}$$

и не зависит от порядка выполнения работ.

Утверждение является частным случаем теоремы ??.

Лемма 13 *Пусть n операций в расписании s выполняются на m машинах, причем машина k, $1 \le k \le m$, работает без простоев в промежутке времени $[T_k, \bar{T}_k]$. Тогда*

$$\prod_{i=1}^{n}(\alpha_i + 1) = \frac{\prod_{k=1}^{m}\bar{T}_k}{\prod_{k=1}^{m}T_k}. \tag{3.2}$$

Справедливость леммы ?? непосредственно вытекает из утверждения ??.

Теорема 27 *Задача $P2|p_i = \alpha_i s_i|C_{max}$ является NP-трудной.*

Доказательство. Эталонную NP-трудную задачу ПРОИЗВЕДЕНИЕ РАЗМЕРОВ сведем к задаче $P2|p_i = \alpha_i s_i|C_{max}$. Пусть $E_0 = \prod_{i\in I} e_i$. Положим $M = n + 2$, $\alpha_{n+1} = 2\frac{E_0}{E} - 1$, $\alpha_{n+2} = 2E - 1$, $\alpha_i = e_i - 1$, где $i \in I$. Покажем, что в построенной задаче расписание s, на котором $C_{max}(s) \le 2E_0$, существует тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что произведение размеров его элементов равно E.

1. Пусть такое подмножество существует. Тогда расписание s строится следующим образом. На первой машине выполняются работа J_{n+1} и работы с номерами из множества I' в произвольном порядке, а на второй — остальные работы. Тогда

$$\bar{T}_1 = (\alpha_{n+1} + 1)\prod_{i\in I'}(\alpha_i + 1) = 2\frac{E_0}{E}\prod_{i\in I'} e_i = 2E_0,$$

$$\bar{T}_2 = (\alpha_{n+2} + 1)\prod_{i\in I\setminus I'}(\alpha_i + 1) = 2E\prod_{i\in I\setminus I'} e_i = 2E_0.$$

значим через I_1 множество работ, выполняемых на этой машине. Следовательно, $C(s) = \max\{\bar{T}_1, \bar{T}_2\} \le 2E_0$, и расписание s искомое.

2. Пусть существует расписание s такое, что $C_{max}(s) \leq 2E_0$. В расписании s работы $n+1$ и $n+2$ выполняются на разных машинах. Действительно, если это не так, то $C_{max}(s) \geq (\alpha_{n+1}+1)(\alpha_{n+2}+1) = 2\frac{E_0}{E}2E = 4E_0 > 2E_0$, и получаем противоречие. Пусть для определенности работа $n+1$ в расписании s выполняется на первой машине. Обозначим через I_1 множество номеров работ, выполняемых на этой машине. Положим $I' = I_1 \setminus \{n+1\}$. Так как $C_{max}(s) = \max\{T_1, T_2\} \leq 2E_0$, и $\prod_{i\in I}(\alpha_i+1) = 4E_0^2$, то из **??** следует, что в промежутке времени $[1, 2E_0]$ все машины работают без простоев. Тогда $2E_0 = \bar{T}_1 = 2\dfrac{E_0}{E}\prod\limits_{i\in I'} e_i$. Следовательно, $\prod\limits_{i\in I'} e_i = E$, и множество I' искомое. Теорема **??** доказана.

Рассмотрим задачу $P2|p_i = \alpha_i s_i|\Sigma C_i$.

Утверждение 4 *[?] Пусть с момента времени t на некоторой машине без простоя обслуживается подряд k работ. Тогда*

$$\sum_{i=1}^{k} C_i = t\sum_{i=1}^{k}\prod_{j=1}^{i}(\alpha_j+1). \tag{3.3}$$

Справедливость утверждения следует из определения длительности работ $p_j = \alpha_j t_j$.

Лемма 14 *Пусть $a_i \geq 2$ для всех $1 \leq i \leq n, n \geq 2$. Тогда $\sum\limits_{k=1}^{n-1}\prod\limits_{i=1}^{k} a_i \leq \prod\limits_{i=1}^{n} a_i$.*

Доказательство проводится по индукции. При $n = 2$ утверждение справедливо, так как $a_1 \leq a_1 a_2$. Пусть для $n = l-1$ неравенство выполнено. Тогда

$$\sum_{k=1}^{l-1}\prod_{i=1}^{k} a_i = \prod_{i=1}^{l-1} a_i + \sum_{k=1}^{l-2}\prod_{i=1}^{k} a_i \leq 2\prod_{i=1}^{l-1} a_i \leq \prod_{i=1}^{l} a_i.$$

Лемма **??** доказана.

Утверждение 5 *[?] Пусть множество работ J упорядочено в порядке возрастания величин α_i. Тогда данный порядок задает оптимальное расписание для задачи $1|p_i = \alpha_i s_i|\Sigma C_i$.*

Справедливость этого утверждения следует из утверждения **??** или теоремы **??**.

Теорема 28 *Задача $P2|p_i = \alpha_i s_i|\Sigma C_i$ является NP-трудной.*

Доказательство. Задачу ПРОИЗВЕДЕНИЕ РАЗМЕРОВ сведем к задаче $P2|p_i = \alpha_i s_i|\Sigma C$. Положим $M = n+4$, и $\alpha_{n+1} = 2\frac{E_0}{E} - 1$, $\alpha_{n+2} = 2E-1$ $\alpha_{n+3} = \alpha_{n+4} = 6E_0 - 1$, $\alpha_i = e_i - 1, i \in I$. Покажем, что в этой задаче расписание s, при котором $\Sigma C(s) \leq y = 24E_0^2 + 8E_0$, существует тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что произведение размеров его элементов равно E.

1. Пусть такое подмножество существует. Тогда расписание s строится следующим образом. На первой машине выполняются работы J_{n+1}, J_{n+3} и работы множества I' , а на второй — остальные работы. Каждая машина выполняет работы без простоев в порядке возрастания величин α_i. Тогда работы J_{n+3}, J_{n+4} начинают выполняться в моменты времени:

$$t_{n+3} = (\alpha_{n+1} + 1) \prod_{i \in I'} (\alpha_i + 1) = 2\frac{E_0}{E} \prod_{i \in I'} e_i = 2E_0,$$

$$t_{n+4} = (\alpha_{n+2} + 1) \prod_{i \in I \setminus I'} (\alpha_i + 1) = 2E \prod_{i \in I \setminus I'} e_i = 2E_0.$$

Обозначим через σ_i суммарное время выполнения работ на машине i. Пусть $|I'| = k$. Из **??** получаем

$$\sigma_1 = \sum_{i=1}^{k} C_i + C_{n+1} + C_{n+3} = \sum_{i=1}^{k} \prod_{j=1}^{i} (\alpha_j + 1) + 2a + 12E_0^2.$$

Из леммы ?? следует, что $\sum_{i=1}^{k}\prod_{j=1}^{i}(\alpha_j+1) \le 2E_0$ и $\sigma_1 \le 12E_0^2+4E_0$. Проделав аналогичные вычисления для машины 2, получаем: $\sigma_2 \le 12E_0^2+4E_0$. Следовательно, $\Sigma C(s) \le 24E_0^2+8E_0$, и расписание s искомое.

2. Пусть существует расписание s такое, что $\Sigma C(s) \le y$. В расписании s работы J_{n+3}, J_{n+4} не могут выполняться на одной машине, так как $C_{n+3}+C_{n+4} \ge (6E_0+1)+(6E_0+1)^2 > y$. Покажем, что в расписании s выполнение этих работ должно начаться не позднее момента времени $2E_0$. Пусть это не так. Согласно утверждению ?? оптимальное расписание работ на каждой машине получается упорядочением их в порядке возрастания величин α_i. Поэтому можно считать, что работы J_{n+3}, J_{n+4} выполняются последними. Предположим, что в расписании s работа J_{n+3} начинает выполняться в момент времени $2E_0x$, где $x>1$. Тогда работа J_{n+4} начинает выполняться не ранее момента времени $2E_0/x$. Так как $2E_0x+2E_0/x > 4E_0$, и все параметры задачи целые, то $2E_0x+2E_0/x \ge 4E_0+1$,

$$\Sigma C(s) \ge s_{n+3}+s_{n+4}+C_{n+3}+C_{n+4}$$

$$\ge 2E_0x+\frac{2E_0}{x}+2E_0x6E_0+\frac{2E_0}{x}6E_0 \ge (6E_0+1)(4E_0+1) = 24E_0^2+10E_0+1.$$

Получили противоречие. Следовательно, к моменту времени $2E_0$ все работы, кроме работ J_{n+3}, J_{n+4}, должны быть выполнены. Тогда в расписании s работы J_{n+1} и J_{n+2} выполняются на разных машинах, так как иначе $C_{max}(s) \ge (\alpha_{n+1}+1)(\alpha_{n+2}+1) = 2\frac{E_0}{E}2E = 4E_0 > 2E_0$. Не ограничивая общности можно считать, что работы J_{n+1} и J_{n+3} выполняются на машине 1, и пусть I_1 — множество работ, выполняемых на этой машине. Положим $I' = I_1 \setminus \{n+1, n+3\}$. Так как $\prod_{i=1}^{n+2}(\alpha_i+1) = 4E_0^2$, то из ?? все машины выполняют работы без простоев в промежутке времени $[1, 2E_0]$. Тогда $2E_0 = \bar{T}_1 = 2E_0/E \prod\limits_{i \in I'} e_i$. Следовательно, $\prod\limits_{i \in I'} e_i = E$, и множество I' искомое. Теорема ?? доказана.

3.4 Многооперационные системы

В этом параграфе установим NP-трудность ряда задач построения оптимального расписания работ в многооперационных системах.

Сначала покажем, что для задачи $F3|p_{i,l} = \alpha_{i,l}(as_{i,l} + b)|C_{max}$ существует такое оптимальное расписание, при котором машины выполняют работы в одинаковой последовательности.

Лемма 15 *Оптимальное расписание s^* задачи $Fm|p_{i,l} = \alpha_{i,l}(as_{i,l} + b)|C_{max}$ можно искать в классе расписаний, при которых машины 1 и 2 выполняют работы в одной и той же последовательности.*

Доказательство. Предположим, что в расписании s машина 1 выполняет операции в последовательности $\pi_1 = (i_1, \ldots, i_n)$, а машина 2 — в последовательности $\pi_2 = (j_1 = i_1, j_2 = i_2, \ldots, j_p = i_p, j_{p+1} = i_{r+1}, \ldots, j_{p+q} = i_r, \ldots, j_n)$, $q > 1$, $r > p \geq 0$.

Из теоремы ?? для расписания s справедливы следующие соотношения

$$s_{i_{r+1},2}(s) = \max\{b/a \prod_{k=1}^{r+1} (\alpha_{i_k,2}a + 1) - b/a, C_{j_p,2}(s)\},$$

$$s_{i_r,2}(s) = \max\{b/a \prod_{k=1}^{r} (\alpha_{i_k,2}a + 1) - b/a, C_{j_{p+q-1},2}(s)\} = C_{j_{p+q-1},2}(s),$$

$$\geq C_{i_{r+1},2}(s) \geq C_{i_{r+1},1}(s) = b/a \prod_{k=1}^{r+1} (\alpha_{i_k,2}a + 1) - b/a.$$

Рассмотрим расписание s', отличающееся от расписания s только тем, что машина 1 выполняет операции в последовательности $\pi_0 = (i_1, \ldots, i_{r-1}, i_{r+1}, i_r, i_{r+2}, \ldots, i_n)$.

Для расписания s', используя теорему ??, получаем следующие соотношения

$$s_{i_{r+1},2}(s') = \max\{b/a(\alpha_{i_{r+1},2}a + 1)\prod_{k=1}^{r-1}(\alpha_{i_k,2}a + 1) - b/a, C_{j_p,2}(s')\},$$

$$s_{i_r,2}(s') = \max\{b/a\prod_{k=1}^{r}(\alpha_{i_k,2}a + 1) - b/a, C_{j_{p+q-1},2}(s')\}.$$

Так как $C_{j_p,2}(s) = C_{j_p,2}(s')$, то $s_{i_{r+1},2}(s) \geq s_{i_{r+1},2}(s')$. Отсюда следует, что $C_{j_{p+q-1},2}(s) \geq C_{j_{p+q-1},2}(s')$ и, следовательно, $s_{i_r,2}(s) \geq s_{i_r,2}(s')$.

Повторяя описанные рассуждения не более чем $O(n^2)$ раз, приходим к расписанию $\bar{s}$ в котором машины 1 и 2 выполняют требования в одной и той же последовательности и выполняется соотношение $C_{i,2}(s) \geq C_{i,2}(\bar{s})$, для всех $J_i \in J$. Тем самым $C_{max}(s) \geq C_{max}(\bar{s})$. Лемма ?? доказана.

Лемма 16 *Оптимальное расписание s^* задачи $Fm|p_{i,l} = \alpha_{i,l}(as_{i,l} + b)|C_{max}$ можно искать в классе расписаний, при которых машины $m-1$ и m выполняют работы в одной и той же последовательности.*

Доказательство леммы ?? аналогично доказательству предыдущей леммы.

Из лемм ?? и ?? непосредственно следует следующее.

Лемма 17 *Пусть число машин $m \leq 3$. Тогда оптимальное расписание задачи $Fm|p_{i,l} = \alpha_{i,l}(as_{i,l} + b)|C_{max}$ можно искать среди расписаний, при которых все машины выполняют работы в одной и той же последовательности.*

Покажем, что если $P \neq NP$, то для любого фиксированного числа U для задачи $F3|p_{i,m} = \alpha_{i,m}s_{i,m}|C_{max}$ не существует полиномиального эвристического алгоритма A с погрешностью $R_A = C_{max}(A)/C^{\star}_{max} < U$.

Более того, установим этот факт даже при условии, что все коэффициенты $\alpha_{i,m}$ операций, выполняемых на машинах 1 и 3 равны между собой. Обозначим данную задачу через $F3|p_{i,m} = \alpha_{i,m}s_{i,m}, \alpha_{i,1} = \alpha_{i,3} = \alpha|C_{max}$.

Для доказательства воспользуемся NP-трудной в сильном смысле задачей 3-РАЗБИЕНИЕ.

Пусть Q некоторая индивидуальная задача 3-РАЗБИЕНИЕ. По входу задачи Q для произвольного фиксированного числа U построим вход индивидуальной задачи Q_U задачи $F3|p_{i,m} = \alpha_{i,m}s_{i,m}|C_{max}$. Положим $|J| = 4n + 1$, $\alpha_{i,1} = \alpha_{i,3} = U^E - 1$ для всех $1 \leq i \leq 4n + 1$, $\alpha_{i,2} = U^{e_i} - 1,\ \ 1 \leq i \leq 3n$, $\alpha_{i,2} = U^{3E} - 1,\ \ 3n + 1 \leq i \leq 4n + 1$. Для удобства назовем первые $3n$ работ R-работами, а остальные H-работами.

Лемма 18 *Вход задачи Q_U может быть построен за время, ограниченное полиномом от $Length(Q)$ и $Max(Q)$.*

Доказательство. Действительно, $Length(Q) = 3n + \sum\limits_{i \in I} \log_2 e_i$ и $Max(Q) = E$. Для построения входа задачи Q_U нам потребуется вычислить величины $\alpha_{i,m}$. Для вычисления каждой из них требуется не более чем $3E - 1$ операций умножения и одно вычитание. Таким образом для построения входа задачи Q_U потребуется не более чем $O(12n + 3)3E$ элементарных операций и вход задачи Q_U может быть построен за время, ограниченное полиномом от $Length(Q)$ и $Max(Q)$. Лемма **??** доказана.

Лемма 19 *Если задача Q имеет ответ "да", то в задаче Q_U оптимальное значение целевой функции равно $C^*_{max} = U^{(4n+5)E}$.*

Доказательство. Пусть существует такое разбиение I на n непересекающихся подмножеств $I_1, I_2, \ldots, I_n$, что для любого $1 \leq l \leq n$ выполнено соотношение $\sum_{i \in I_l} e_i = E$. Построим расписание s в задаче Q_U следующим образом. Все машины выполняют работы без простоев в порядке, заданном перестановкой $\pi = (3n+1, I_1, 3n+2, I_2, \ldots, 4n, I_n, 4n+1)$. Запись $\pi = (\ldots, j, I_k, l, \ldots)$ означает, что между работами j и l выполняются работы множества I_k в произвольном порядке.

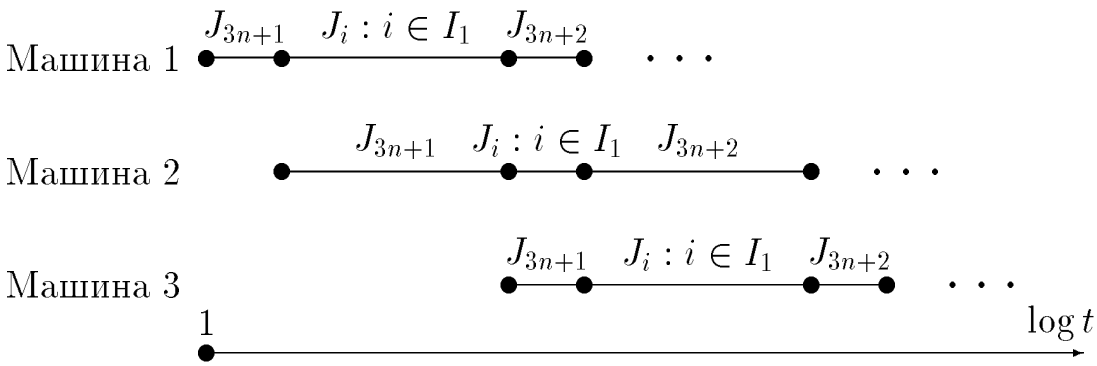

Рисунок 1.

Тогда из ?? получаем, что времена начала и окончания выполнения работы $J_{3n+l}, 1 \leq l \leq n+1$ на машине 1 соответственно равны $s_{3n+l,1} = U^{(4l-4)E}$ и $C_{3n+l,1} = U^{(4l-3)E}$. Следовательно на машине 2 работа J_{3n+1} начинает выполнение в момент U^E, Предположим, что на машине 2 работа J_{3n+l} начинает выполнение в момент $U^{(4l-3)E}$, тогда $C_{3n+l,2} = U^{4lE}$. Обозначим через $s_{I_l,k}$ и $C_{I_l,k}$ моменты начала и окончания выполнения работ из множества I_l на машине k соответственно. Тогда $s_{I_l,1} = U^{(4l-3)E}$ и $C_{I_l,1} = U^{4lE}$. Следовательно, работы множества I_l выполняются на машине 2 без задержки, $s_{I_l,2} = U^{4lE}$, $C_{I_l,1} = U^{(4l+1)\sum_{i \in I_l} e_i} = U^{(4l+1)E} = U^{(4(l+1)-3)E}$ и работа J_{3n+l+1} начинает выполнение в момент $U^{(4(l+1)-3)E}$. По индукции получаем, что машина

2 выполняет все операции, согласно перестановки π без простоев и задержек в промежутке времени $[U^E, U^{(4n+4)E}]$. Аналогично рассуждая, получаем что машина 3 заканчивает выполнение последней операции в момент времени $U^{(4n+5)E}$. Лемма **??** доказана.

Лемма 20 *Существует оптимальное решение задачи Q_U, в котором выполнение каждой операции $O_{i,k}$ начинается в момент $U^{z_{i,k}}$, где $z_{i,k} \geq 0$ — некоторое целое число.*

Доказательство. Пусть s — некоторое допустимое расписание в задаче Q_U и $O_{i,k}$ — первая по времени операция, которая начинает выполнение в момент U^z, где z не является целым. Из целочисленности исходных данных задачи Q, утверждения 2 и построения входа задачи Q_U следует, что если операция начинает выполнение в момент целой степени U, то она и оканчивает выполнение в момент целой степени U. Таким образом, либо операция $O_{i,k}$ — первая на машине k, либо предшествующая ей операция завершилась в момент целой степени U. То же самое относится и к предшествующим ей операциям работы J_i. Тогда существует такой момент U^t, что $t \leq z$ — целое и в расписании s все операции работы J_i, предшествующие $O_{i,k}$ и все операции машины k, , предшествующие $O_{i,k}$ закончили выполнение до момента U^t. Положим $s_{i,k} = U^t$. Полученное расписание останется допустимым и его длина не увеличится. Повторив эту процедуру для остальных операций, начинающихся не в моменты степени U, построим расписание, удовлетворяющее условию леммы. Лемма **??** доказана.

Лемма 21 *Если задача Q имеет ответ "нет", то оптимальное значение в задаче Q_U $C^*_{max} \geq U^{(4n+6)E}$.*

Доказательство. Так как для любой работы $J_i \in J$ $\alpha_{i,1} = U^E$, то машина 2 начинает выполнение не ранее момента E. Обозначим через $C(k)$ время завершения работы машины k. Тогда из ?? получаем

$$C_2 \geq U^E \prod_{i=1}^{4n+1} (\alpha_i + 1) = U^E U^{nE} U^{3(n+1)E} = U^{4(n+1)E}.$$

Отсюда и из того, что для любой работы $J_i \in J$ $\alpha_{i,3} = U^E$, получаем $C^*_{max} \geq U^{(4n+5)E}$. Если на машине 2 есть простой после момента U^E, то $C^*_{max} > U^{(4n+5)E}$ и из леммы ?? получаем $C^*_{max} \geq U^{(4n+6)E}$. По лемме ?? можно считать, что в оптимальном расписании на всех машинах работы выполняются в одинаковой последовательности. Пусть $\pi = \{J_{(1)}, \ldots, J_{(n)}\}$ — перестановка, задающая порядок выполнения работ в оптимальном расписании. Обозначим через $s_{(i),k}$, $C_{(i),k}$ моменты начала и окончания выполнения операции i-й работы из перестановки π на машине k. Назовем две H-работы соседними, если в перестановке π между ними нет других H-работ.

1. Если первой выполняется R-работа, то на машине 2 есть простой.

Действительно пусть это работа J_l, $l \leq 3n$. Тогда $C_{(1),1} = U^E$, $C_{(2),1} = U^{2E}$, $C_{(1),2} = U^E U^{e_l} < U^{2E}$ и на машине 2 есть простой.

Пусть первой в оптимальном расписании выполняется H-работа, тогда машина 3 начинает обслуживание с момента U^{4E}, Тогда из ?? получаем

$$C_3 \geq U^{4E} \prod_{i=1}^{4n+1} (\alpha_i + 1) = U^{4E} U^{(4n+1)E} = U^{(4n+5)E}.$$

Следовательно, если на машине 3 есть простой после момента U^{4E}, то из леммы ?? получаем $C^*_{max} \geq U^{(4n+6)E}$.

3. Если в оптимальном расписании между двумя соседними H-работами выполняется не более двух R-работ, то на машине 3 есть простой.

Действительно, пусть это l-я и $l+r$-я работы в перестановке π, $r \leq 3$ и пусть операция l-й работы на машине 2 начинает выполняться в момент $t \geq U^{lE}$. Тогда $C_{(l),2} = tU^{3E}$, $C_{(l+r),2} \geq tU^{6E+(r-1)E/4}$. Если машина 3 выполняет первые $l+r-1$ работу без простоев, то $C_{(l+r-1),3} = U^{4E}U^{(l+r-1)E} = U^{(l+r+3)E}$, $s_{(l+r),3} \geq C_{(l+r),2} \geq tU^{6E+(r-1)E/4} > U^{(l+r+3)E} = C_{(l+r-1),3}$ и на машине 3 между операциями $l+r-1$-й и $l+r$-й работы есть простой.

Из того, что число R-работ равно $n+1$, а число H-работ равно $3n$, имеем, что в оптимальном расписании между двумя соседними H-работами выполняется ровно три R-работы. Поэтому далее будем считать, что в перестановке π H-работы имеют номера $4l+1$, $0 \leq l \leq n$.

4. Пусть в оптимальном расписании между двумя соседними H-работами $J_{(4l+1)}$ и $J_{(4l+5)}$, $0 \leq l \leq n-1$ выполняется три R-работы и $\prod_{i=1}^{3}(\alpha_{(4l+1+i),2}+1) > U^E$, тогда на машине 3 есть простой.

Действительно, пусть операция $4l+1$-й работы на машине 2 начинает выполняться в момент $t \geq U^{(4l+1)E}$. Тогда из утверждения 2 $C_{(4l+5),2} \geq t\prod_{i=0}^{4}(\alpha_{(4l+1+i),2}+1) = tU^{6E}\prod_{i=1}^{3}(\alpha_{(l+i),2}+1) > tU^{7E}$. Если машина 3 выполняет первые $4l+4$ работы без простоев, то $C_{(4l+4),3} = U^{4E}U^{(4l+4)E} = U^{(4l+8)E}$, $s_{(4l+5),3} \geq C_{(4l+5),2} > tU^{7E} \geq U^{(4l+8)E} = C_{(l+r-1),3}$ и на машине 3 между операциями $4l+4$-й и $4l+5$-й работы есть простой.

5. Пусть в оптимальном расписании между двумя соседними H-работами $J_{(4l+1)}$ и $J_{(4l+5)}$ $0 \leq l \leq n-1$ выполняется три R-работы и $\prod_{i=1}^{3}(\alpha_{(4l+1+i),2}+1) < U^E$ и нет простоев на машине 3, тогда на машине 2 есть простой.

Действительно, пусть операция $4l+1$-й работы на машине 1 начинает выполняться в момент $t \geq U^{4lE}$. Тогда $C_{(4l+5),1} = tU^{5E}$ Так как на машине 3 нет простоев, то $\prod_{i=1}^{3}(\alpha_{(4l+1+i),2}+1) \leq U^E$ для всех $0 \leq l \leq n$.

Следовательно, $C_{(4l),2} \leq U^E \prod_{k=1}^{l} U^{3E} \prod_{i=1}^{3}(\alpha_{(4k+1+i),2}+1) \leq U^{(4l+1)E} \leq tU^E$. и $C_{(4l+4,2)} = tU^E U^{3E} \prod_{i=1}^{3}(\alpha_{(4l+1+i),2}+1) < tU^{5E}$. Получаем, что $s_{(4l+5),1} \geq C_{(4l+5),1} > C_{(4l+4,2)}$ и на машине 2 есть простой.

6. Пусть в оптимальном расписании между двумя соседними Н-работами $J_{(4l+1)}$ и $J_{(4l+5)}$ $0 \leq l \leq n-1$ выполняется три R-работы и $\prod_{i=1}^{3}(\alpha_{(4l+1+i),2}+1) = U^E$. Тогда в задаче Q положим $I_l = \{(4l+2),(4l+3),(4l+4)\}$, $0 \leq l \leq n-1$. Нетрудно убедиться, что разбиение множества I на такие подмножества является искомым 3-разбиением в задаче Q. Получаем противоречие, которое заканчивает доказательство леммы. Лемма ?? доказана.

Теорема 29 *Если $P \neq NP$, то для задачи $F3|p_{i,m} = \alpha_{i,m}s_{i,m}, \alpha_{i,1} = \alpha_{i,3} = \alpha|C_{max}$ не существует полиномиального эвристического алгоритма A с погрешностью $R_A = C_{max}(A)/C^{\star}_{max} < U$, где U — произвольное фиксированное число.*

Доказательство. Пусть существует полиномиальный эвристический алгоритм A с погрешностью $R_A < U$. Тогда по данному входу индивидуальной задачи Q задачи 3-разбиение построим вход индивидуальной задачи Q_U задачи $F3|p_{i,m} = \alpha_{i,m}s_{i,m}|C_{max}$ и решим ее алгоритмом A. Из леммы ?? следует, что если индивидуальная задача Q имеет ответ "да", то алгоритм A получит значение в задаче Q_U $C^*_{max} < U^{(4n+5)E}U = U^{(4n+6)E}$, если индивидуальная задача Q имеет ответ "нет", то из леммы ?? алгоритм A получит значение в задаче Q_U $C^*_{max} \geq U^{(4n+6)E}$. Из леммы ?? следует, что время работы алгоритма ограничено полиномом от $Length(Q)$ и $Max(Q)$. Следовательно получаем псевдополиномиальный алгоритм, решающий NP-полную в сильном смысле задачу 3-разбиение, что противоречит гипотезе $P \neq NP$.

Теорема ?? доказана.

Теорема 30 *Задача $F2|p_{i,m} = \alpha_{i,m}s_{i,m}|L_{max}$ является NP-трудной.*

Доказательство. Построим сведение задачи 4-ПРОИЗВЕДЕНИЕ к задаче $F2|p_{i,m} = \alpha_{i,m}s_{i,m}|L_{max}$. Положим $|J| = 5n-1$,

$$\begin{array}{lll} \alpha_{i,1} = 0, & \alpha_{i,2} = e_i - 1, & d_i = 2^{n-1}E^n, \ \ 1 \le i \le 4n; \\ \alpha_{4n+1,1} = E-1, & \alpha_{4n+1,2} = 1, & d_{4n+1} = 2E; \\ \alpha_{4n+i,1} = 2E-1, & \alpha_{4n+i,2} = 1, & d_{4n+i} = 2^iE^i, \ \ 2 \le i \le n-1. \end{array}$$

Покажем, что в этой задаче существует расписание со значением целевой функцией $L_{max}(s) = 0$ тогда и только тогда, когда в I существует разбиение на n непересекающихся подмножеств $I_1, I_2, \ldots, I_n$, таких что $\prod_{i\in I_k} e_i = E$ для $1 \le k \le n$.

1. Пусть такое разбиение существует. Рассмотрим следующее расписание. На первой машине выполняются работы с номерами от $4n+1$ до $5n-1$ в порядке возрастания номеров. На второй машине работы выполняются в последовательности $(\pi_k, 4n+k)$, $1 \le k \le n$, где π_k произвольная перестановка работ с номерами из множеств I_k. Нетрудно убедиться, что такое расписание является допустимым.

2. Пусть существует расписание s такое, что $L_{max}(s) = 0$. Так как $\max\limits_{i\in I} d_i = 2^{n-1}E^n$ и $\prod\limits_{i=1}^{5n-1}(\alpha_{i,2}+1) = 2^{n-1}E^n$, то машина 2 работает без простоев в промежутке $[1, 2^{n-1}E^n]$. Так как работа выполнение $4n+1$ заканчивается до директивного срока $d_{4n+1} = 2E$, то она выполняется без задержек в промежутке $[1, 2E]$ Следовательно, $s_{1,1} = 1$, $C_{1,1} = E$, $s_{1,2} = E$, $C_{1,2} = 2E$. Работа $4n+i$, $2 \le i \le n-1$ заканчивается до директивного срока $d_{4n+i} = 2^iE^i$, и аналогично вышеизложенному на машине 1 она начинает выполняться не позже момента $s_{i,1} = 2^{i-2}E^{i-1}$. С другой стороны выполнение этой работы не может начаться раньше

этого срока, так как на машине 1 в это время выполняется работа $i-1$. Следовательно получаем $s_{i,1} = 2^{i-2}E^{i-1}$, $C_{i,1} = s_{i,2} = 2^{i-1}E^i$, $C_{i,2} = 2^iE^i$, для $2 \le i \le n-1$.

Из приведенных выше рассуждений следует, что на машине 2 работы с номерами от 1 до n выполняются в промежутках $[2^{i-1}E^{i-1}; 2^{i-1}E^i]$, $1 \le i \le n$. Пусть I_k — множество работ, выполняемых в промежуток $[2^{k-1}E^{k-1}; 2^{k-1}E^i]$. Тогда из **??** следует, что

$$E = \prod_{i \in E_k} (v_{i,2}^2 + 1) = \prod_{i \in E_k} e(i),$$

и набор множеств I_k задает искомое разбиение.

Теорема **??** доказана.

Перейдем к рассмотрению задач с системами открытого типа.

Теорема 31 *Задача* $O3|p_{i,m} = \alpha_{i,m}s_{i,m}, \alpha_{i,3} = \alpha|C_{max}$ *является* NP-*трудной.*

Доказательство. Построим сведение задачи ПРОИЗВЕДЕНИЕ РАЗМЕРОВ к задаче $O3|p_{i,m} = \alpha_{i,m}s_{i,m}|C_{max}$. Пусть $\prod_{i \in I} e_i = E_0$. Обозначим через $\bar{E} = 2E_0$. Без ограничения общности можно считать, что n — четное.

Положим $|I| = n+4$,

$$\begin{array}{llll}
\alpha_{i,1} = 0, & \alpha_{i,2} = e_i^2 - 1, & \alpha_{i,3} = \bar{E}^2 - 1, & 1 \le i \le n; \\
\alpha_{n+1,1} = 0, & \alpha_{n+1,2} = 4E_0^2/E^2 - 1, & \alpha_{n+1,3} = \bar{E}^2 - 1; & \\
\alpha_{n+2,1} = 0, & \alpha_{n+2,2} = 4E^2 - 1, & \alpha_{n+1,3} = \bar{E}^2 - 1 & \\
\alpha_{n+3,1} = \bar{E}^{n-1} - 1, & \alpha_{n+3,2} = \bar{E}^{n-1} - 1, & \alpha_{n+3,3} = \bar{E}^2 - 1; & \\
\alpha_{n+4,1} = \bar{E}^{n+1} - 1, & \alpha_{n+4,2} = \bar{E}^{n-3} - 1, & \alpha_{n+4,3} = \bar{E}^2 - 1 &
\end{array}$$

Покажем, что в построенной задаче расписание s, при котором $C(s) \le \bar{E}^{2n}$, существует тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что произведение размеров его элементов равно E.

1. Пусть такое подмножество существует. Тогда расписание s может быть построено следующим образом. Первая машина сначала выполняет работу J_{n+3} и затем работу J_{n+4}. На второй машине первыми в произвольном порядке выполняются работы с номерами из множества I', следом работы J_{n+1}, J_{n+4}, J_{n+3}, J_{n+2} и затем оставшиеся работы в произвольном порядке. Третья машина выполняет работы в следующей последовательности $J_{n+4}, J_1, J_2, \ldots, J_n, J_{n+1}, J_{n+2}, J_{n+3}$.

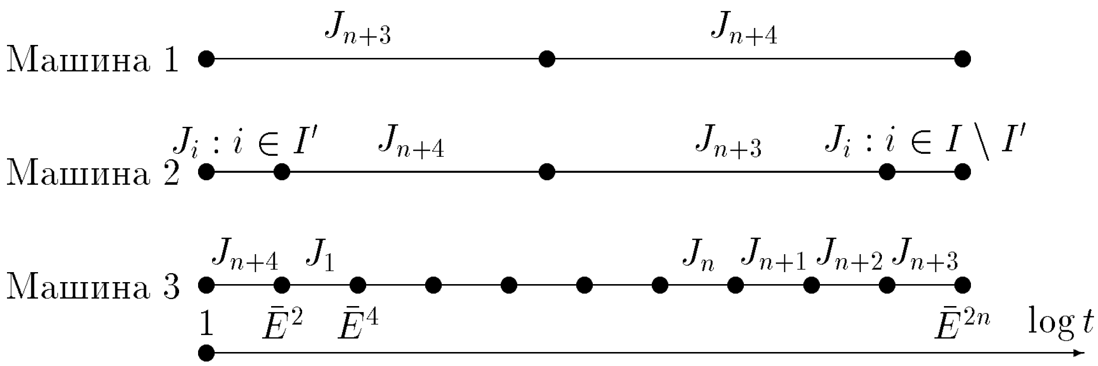

Рисунок 2.

Нетрудно убедиться, что все машины работают без простоев в промежутке $[1, \bar{E}^{2n}]$ и $C_{max}(s) \le \bar{E}^{2n}$.

2. Пусть существует расписание s такое, что $C(s) \le \bar{E}^{2n}$. Предположим, что все машины работают без простоев с момента 1. Вычислим времена завершения работы каждой из машин. Из (3.1) получим

$$T_1 = (\alpha_{n+3,1} + 1)(\alpha_{n+4,1} + 1) = \bar{E}^{n-1}\bar{E}^{n+1} = \bar{E}^{2n},$$

$$T_2 = \prod_{i=1}^{n+4}(\alpha_{i,2} + 1) = 16E_0^2\bar{E}^{n-1}\bar{E}^{n-3}\prod_{i=1}^{n} e_i^2 = 16E_0^4\bar{E}^{2n-4} = \bar{E}^{2n},$$

$$T_3 = \prod_{i=1}^{n+4}(\alpha_{i,2} + 1) = \bar{E}^{2n}.$$

Итак из полученного видно, что в расписании s все машины работают без простоев в промежутке $[1, \bar{E}^{2n}]$ и $C_{max}(s) \le \bar{E}^{2n}$. В противном случае получим противоречие с предположением $C(s) \le \bar{E}^{2n}$. Заметим, что из этого факта и из того, что $\alpha_{i,3} = \bar{E}^2 - 1$, для всех $1 \leq i \leq n+4$, следует, что на машине 3 все работы начинают и заканчивают выполнение в четные степени $\bar{E}$. Вычислим времена завершения работ J_{n+3} и J_{n+4} в предположении, что их выполнение начинается в момент 1 и происходит без задержек. Имеем

$$C_{n+3} = \prod_{i=1}^{3} (\alpha_{n+3,i} + 1) = \bar{E}^{n-1}\bar{E}^{n-1}\bar{E}^2 = \bar{E}^{2n},$$

$$C_{n+4} = \prod_{i=1}^{3} (\alpha_{n+4,i} + 1) = \bar{E}^{n+1}\bar{E}^{n-3}\bar{E}^2 = \bar{E}^{2n},$$

из полученного видно, что в расписании s работы J_{n+3} и J_{n+4} выполняются с момента 1 без задержек. Тогда на машине 2 работы J_{n+3} и J_{n+4} выполняются друг за другом в промежутке $[\bar{E}^2, \bar{E}^{2n-2}]$. Действительно, если первая из этих работ начинает выполнение на машине 2 в момент $0 < t < \bar{E}^2$, то операция этой работы на машине 3 не может быть выполнена в промежутке $[1, \bar{E}^{2n}]$. Аналогично, если вторая работа заканчивается на машине 2 в момент $\bar{E}^{2n-2} < t < \bar{E}^{2n}$, то операция этой работы на машине 3 не может быть выполнена в промежутке $[1, \bar{E}^{2n}]$. Пусть работа J_{n+3} начинает выполнение на машине 2 в момент 1, тогда на машине 1 первой выполняется работа J_{n+4}, следовательно, $s_{n+3,1} = E^{n+1}$, $C_{n+3,2} = E^{n-1}$, и на машине 3 работа J_{n+3} начинается в E^{n-1} и завершается вE^{n+1}. Получаем противоречие с тем, что в расписании s на машине 3 все работы начинают и заканчивают выполнение в четные степени $\bar{E}$. Аналогично рассуждая получаем, что в расписании s на машине 2 ни одна из работ J_{n+3} и J_{n+4} не может быть начата в момент 1 или завершена в момент $\bar{E}^{2n}$. Следовательно, на машине 2 $n+2$ оставшиеся работы выполняются в интервалах времени $[1, \bar{E}^2]$ и $[\bar{E}^{2n-2}, \bar{E}^{2n}]$. Так как

$$(\alpha_{n+1,2} + 1)(\alpha_{n+2,2} + 1) = (4E_0^2/E^2)(4E^2) = 16E_0^2 = 4\bar{E}^2 > \bar{E}^2,$$

то работы J_{n+1} и J_{n+2} не могут в расписании s обе выполняться в одном интервале. Пусть для определенности работа J_{n+1} в расписании s выполняется в интервале $[1, \bar{E}^2]$. Обозначим через I_1 множество номеров работ, выполняемых в этом интервале. Положим $I' = I_1 \setminus \{n+1\}$. Имеем $\prod_{i \in I_1}(\alpha_i + 1) = \bar{E}^2 = 4E_0^2$, Тогда из (3.1) имеем $\bar{E}^2 = 4E_0^2 = 4\frac{E_0^2}{E^2}\prod_{i \in I'} e_i^2$. Отсюда получаем $\prod_{i \in I'} e_i = E$, и множество I' искомое. Теорема ?? доказана.

Теорема 32 *Задача $O2|p_{i,m} = \alpha_{i,m}s_{i,m}|L_{max}$ является NP-трудной.*

Доказательство. Построим сведение задачи ПРОИЗВЕДЕНИЕ РАЗМЕРОВ к задаче $O2|p_{i,m} = \alpha_{i,m}s_{i,m}|L_{max}$. Положим $T = 1$, $I = |n+3|$, $\alpha_{i,j} = e_i - 1$, $1 \le i \le n$, $\alpha_{n+1,j} = 2E_0/E - 1$, $\alpha_{n+2,j} = 2E - 1$, $\alpha_{n+3,j} = 2E_0 - 1$, $1 \le j \le 2$, $d_i = 8E_0^3$ $1 \le i \le n+2$, $d_{n+3} = 4E_0^2$.

Покажем, что в построенной задаче существует расписание s, не нарушающее директивных сроков тогда и только тогда, когда в I есть подмножество $I' \subseteq I$ такое, что произведение размеров его элементов равно E.

1. Пусть такое подмножество существует. Построим расписание s следующим образом. Сгруппируем работы в три множества: $N_1 = \{i | i \in I'\} \cup \{n+1\}$, $N_2 = \{n+3\}$, $N_3 = \{i | i \in I \setminus I'\} \cup \{n+2\}$. На первой машине работы выполняются в последовательности (π_1, π_2, π_3), на второй машине — в последовательности (π_2, π_3, π_1), где π_k произвольная перестановка работ из множества N_k, $1 \le k \le 3$. Так как $\prod_{i \in N_k}(\alpha_{i,j} + 1) = 2E_0$, $1 \le j \le 3$, то в построенном расписании все машины работают без простоев в промежутке $[1, 8E_0^3]$, и выполнение всех работ завершается до директивных сроков.

2. Пусть существует расписание s такое, что $L(s) = 0$. Так как выполнение работы $n+3$ в расписании s завершается к моменту времени

$4E_0^2$, то она выполняется с момента времени 1 без задержек. Следовательно, существует машина, которая выполняет работу $n+3$ в промежуток времени $[2E_0, 4E_0^2]$. Без ограничения общности можно считать, что это есть машина 1. Тогда остальные работы выполняются на ней в промежутки времени $[1, 2E_0], [4E_0^2, 8E_0^3]$. Так как $(\alpha_{n+1,1}+1)(\alpha_{n+2,1}+1) = 4E_0 > 2E_0$, то работы $n+1$ и $n+2$ на машине 1 выполняются в разные промежутки времени. Пусть I' — множество номеров работ, которые выполняются в том же промежутке, что и работа $n+1$. Так как машина 1 работает без простоев, то из леммы 2 следует, что

$$2E_0 = (\alpha_{n+1,1}+1)\prod_{i\in I'}(\alpha_{i,1}+1) = 2\frac{E_0}{E}\prod_{i\in I'} e_i.$$

Поэтому, $\prod\limits_{i\in I'} e_i = E$, и множество I' искомое. Теорема 35 доказана.

3.5 Полиномиально-разрешимые случаи.

Как было отмечено выше, для почти всех критериев результаты по NP-трудности задач с постоянными длительностями переносятся на задачи с простым линейным ростом длительностей. Аналогичная картина наблюдается и для некоторых полиномиально-разрешимых задач. Для каждой из таких задач можно построить полиномиальный алгоритм в некотором смысле эквивалентный алгоритму для решения такой же задачи с постоянными длительностями. Ниже делается попытка предложить некоторый метод, позволяющий по уже существующему алгоритму, построить другой алгоритм, решающий эквивалентную задачу.

Пусть Π — оптимизационная задача, I_Π — индивидуальная задача, s_i— допустимое решение задачи I_Π, $1 \le i \le n$ и $f(s_i)$ — значение целевой функции на допустимом решении s_i. Пусть $\gamma(x) : R \to R$ —

строго возрастающая функция.

Определение 2 *Будем говорить, что задача Π изоморфна задаче $\bar{\Pi}$ относительно функции γ, если*

1. Для любой индивидуальной задачи I_Π существует индивидуальная задача $I_{\bar{\Pi}}$, такая, что для любого допустимого решения s_i задачи I_Π решение $\bar{s}_i = \gamma(s_i)$ является допустимым для задачи $I_{\bar{\Pi}}$, и для любого допустимого решения $\bar{s}_i$ задачи $I_{\bar{\Pi}}$ решение $s_i = \gamma^{-1}(\bar{s}_i)$ является допустимым решением задачи I_Π.

2. $\bar{f}(\gamma(s_i)) = \gamma(f(s_i))$ для любого допустимого решения s_i задачи I_Π.

Индивидуальные задачи, описанные в определении, будем называть изоморфными. Рассмотрим задачи $F2||C_{max}$ $O2||C_{max}$.

Лемма 22 *1. Относительно функции $\gamma = 2^x - b/a$ задача $F2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|C_{max}$ изоморфна задаче $F2||C_{max}$.*
2. Относительно функции $\gamma = 2^x - b/a$ задача $O2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|C_{max}$ изоморфна задаче $O2||C_{max}$.

Доказательство. Докажем пункт 1. Числовыми параметрами, характеризующими каждую индивидуальную задачу с постоянными длительностями, являются длительности выполнения операций $\bar{p}_i$. В задачах с растущими длительностями числовыми параметрами являются величины α_i. Обозначим через Ω и $\bar{\Omega}$ множества всех операций в задачах $F2||C_{max}$ и $F2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|C_{max}$ соответственно, Пусть $\bar{s}_i$, $i \in \bar{\Omega}$ — допустимое расписание некоторой индивидуальной задаче $I_{\bar{\Pi}}$. Построим индивидуальную задачу I_Π, изоморфную индивидуальной задаче $I_{\bar{\Pi}}$. Положим $\alpha_{i,m} = 2^{\bar{p}_{i,m}}/a - 1$. Убедимся, что в построенной задаче $s_i = 2^{\bar{s}_i} - b/a$ является допустимым решением. Обозначим через τ_i и $\bar{\tau}_i$ момент завершения операций в задачах I_Π и $I_{\bar{\Pi}}$ соответственно:

$$\tau_i = s_i + \alpha_i(as_i + b) = 2^{\bar{s}_i} - \frac{b}{a} + \alpha_i a(2^{\bar{s}_i} - \frac{b}{a}) + \alpha_i b$$

$$= 2^{\bar{s}_i}(1 + \alpha_i a) - \frac{b}{a} = 2^{\bar{s}_i} 2^{\bar{p}_i} - \frac{b}{a} = 2^{\bar{\tau}_i} - \frac{b}{a}. \tag{3.4}$$

Для допустимости решения необходимо выполнение неравенств, связанных с величинами s_i и τ_i. Поэтому из (3.3) и монотонности функции $2^x - \frac{b}{a}$ следует, что если $\bar{s}_i$ — допустимое решение в задаче $I_{\bar{\Pi}}$, то s_i — допустимое решение в задаче I_Π. Аналогично проверяется и обратное утверждение. Совпадение целевых функций следует из равенства $\max\limits_{i \in I} 2^{\tau_i} - \dfrac{b}{a} = 2^{\max_{i \in I} \tau_i} - \dfrac{b}{a}$. Второе утверждение леммы доказывается аналогично. Лемма 25 доказана.

Следствие 4 *Если $\bar{s}_i$ — оптимальное решение задачи $I_{\bar{\Pi}}$, то s_i — оптимальное решение задачи I_Π, изоморфной задаче $I_{\bar{\Pi}}$.*

В [?], [?] для точного решения задач $F2||C_{max}$, $O2||C_{max}$ приведены полиномиальные алгоритмы с трудоемкостью $O(n \log n)$ и $O(n)$, соответственно. Обозначим длительности операций работы i на первой и второй машинах через a_i и b_i.

Алгоритм $\overline{AF}$.

1. Множество работ I разбивается на два подмножества $I_1 = \{i | a_i \leq b_i\}$ и $I_2 = \{i | a_i > b_i\}$.

2. Находится перестановка π, в которой сначала идут операции множества I_1 в порядке неубывания значений a_i, а затем операции множества I_2 в порядке неубывания значений b_i.

3. Находится расписание, обслуживающее операции в порядке заданной перестановкой π на обеих машинах.

Алгоритм $\overline{AO}$.

1. Множество работ I разбивается на два подмножества $I_1 = \{i | a_i \le b_i\}$ и $I_2 = \{i | a_i > b_i\}$.

2. Среди операций множества I_1 выбирается операция j_1 такая, что $b_{j_1} \ge \max\limits_{i \in I_1} a_i$, а среди операций множества I_2 выбирается операция j_2 такая, что $a_{j_2} \ge \max\limits_{i \in I_1} b_i$.

3. Вычисляется $\alpha = \sum\limits_{i \in I} a_i$, $\beta = \sum\limits_{i \in I} b_i$.

4. В силу симметрии задачи можно считать, что $\alpha + b_{j_2} \le \beta + a_{j_1}$. Находится перестановка $\pi_a = (\pi_{I_1 \setminus j_1}, \pi_{I_2 \setminus j_2}, j_2, j_1)$ и $\pi_b = (j_1, \pi_{I_1 \setminus j_1}, \pi_{I_2 \setminus j_2}, j_2)$, где $\pi_{I_1 \setminus j_1}$ — произвольная перестановка из $I_1 \setminus j_1$, а $\pi_{I_2 \setminus j_2}$ — произвольная перестановка из $I_2 \setminus j_2$.

5. Строится расписание, обслуживающее операции в порядке заданной перестановкой π_a на машине 1 и перестановкой π_b на машине 2.

В обоих алгоритмах операция сразу ставится на выполнение, если все предшествующие операции выполнены и машина свободна.

Алгоритмы AO и AF работают следующим образом В алгоритмах $\overline{AO}$ и $\overline{AF}$ величины a_i заменяются на $\alpha_i = (a_{i,1} + 1)a$, b_i заменяются на $\beta_i = (a_{i,2} + 1)a$ и каждая операция сложения двух чисел $x_1 + x_2$ — на следующую операцию $(x_1 + b/a)(x_2 + b/a) - b/a$.

Алгоритм AF.

1. Множество работ I разбивается на два подмножества $I_1 = \{i | \alpha_i \le \beta_i\}$ и $I_2 = \{i | \alpha_i > \beta_i\}$.

2. Находится перестановка π, в которой сначала идут операции множества I_1 в порядке неубывания значений α_i, а затем операции множества I_2 в порядке неубывания значений β_i.

3. Находится расписание, обслуживающее операции в порядке заданной перестановкой π на обеих машинах.

Алгоритм AO.

1. Множество работ I разбивается на два подмножества $I_1 = \{i|\alpha_i \leq \beta_i\}$ и $I_2 = \{i|\alpha_i > \beta_i\}$.

2. Среди операций множества I_1 выбирается операция j_1 такая, что $\beta_{j_1} \geq \max\limits_{i \in I_1} \alpha_i$, а среди операций множества I_2 выбирается операция j_2 такая, что $\alpha_{j_2} \geq \max\limits_{i \in I_1} \beta_i$.

3. Вычисляется $\alpha = \prod\limits_{i \in I}(\alpha_i + b/a) - b/a$, $\beta = \prod\limits_{i \in I}(\beta_i + b/a) - b/a$.

4. В силу симметрии задачи можно считать, что $(\alpha + b/a)(\beta_{j_2} + b/a) \leq (\beta + b/a)(\alpha_{j_1} + b/a)$.

Находятся перестановки $\pi_a = (\pi_{I_1 \setminus j_1}, \pi_{I_2 \setminus j_2}, j_2, j_1)$ и $\pi_b = (j_1, \pi_{I_1 \setminus j_1}, \pi_{I_2 \setminus j_2}, j_2)$, где $\pi_{I_1 \setminus j_1}$ — произвольная перестановка из $I_1 \setminus j_1$, а $\pi_{I_2 \setminus j_2}$ — произвольная перестановка из $I_2 \setminus j_2$.

5. Строится расписание, обслуживающее операции в порядке заданной перестановкой π_a на машине a и перестановкой π_b на машине b.

Пусть I_Π — произвольная индивидуальная задача $F2|p_{i,m} = \alpha_{i,m}(as_{i,m}+b)|C_{max}$, а $I_{\bar{\Pi}}$ изоморфная ей индивидуальная задача $F2||C_{max}$. Алгоритм $\overline{AF}$ находит оптимальное решение $\bar{s}_i$ задачи $F2||C_{max}$. Покажем, что алгоритм AF находит решение $s_i = 2^{\bar{s}_i} - b/a)$, которое, как было отмечено выше, также является оптимальным. Действительно, в начальный момент значению каждого числового параметра x в задаче $I_{\bar{\Pi}}$ соответствует значение $2^x - b/a)$ в задаче I_Π. Так как функция $2^x - b/a)$ строго возрастающая, операция сравнения двух величин дает одинаковый результат. Пусть на некотором шаге в алгоритме $\overline{AF}$ вычисляется число $\bar{x} = x_1 + x_2$. Тогда в алгоритме AF на этом шаге получается число $x = 2^{x_1}2^{x_2} - b/a) = 2^{x_1+x_2} - b/a) = 2^x - b/a)$. Следовательно, алгоритм AF получит решение $s_i = 2^{\bar{s}_i} - b/a)$. Необходимо отметить, что по сравнению с первоначальными алгоритмами одна элементарная операция заменена на четыре других элементарных операции, и, следовательно трудоемкость полученных алгоритмов увеличится не более, чем в четыре раза. Проведя аналогичные рассуждения и для алгоритма AO получаем, следующее утверждение.

Теорема 33 *1.Алгоритм AF находит оптимальное расписание задачи $F2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|C_{max}$ с трудоемкостью $O(n \log n)$.*

2.Алгоритм AO находит оптимальное расписание задачи $O2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|C_{max}$ с трудоемкостью $O(n)$.

В заключение параграфа рассмотрим задачи $F2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|L_{max}$ и $O2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|L_{max}$. Как было доказано в предыдущем параграфе обе задачи являются NP-трудными. Поэтому, представляет интерес построение для них приближенных полиномиальных алгоритмов. Однако из специфики задач минимизации максимального временного смещения видно, что целевая функция в оптимальном решении может принимать нулевые и даже отрицательные значения. Поэтому при построении приближенных алгоритмов в литературе по теории расписаний принято рассматривать эквивалентные задачи, в которых для каждой работы J_i дополнительно задано время доставки (delivery) $q_i \geq 0$. Время окончания работы J_i в таких задачах вычисляется по формуле

$$C_i = s_i + p_i + q_i. \tag{3.5}$$

Пусть $D = \max_{J_i \in J} d_i$ Положив для каждой работы J_k $q_k = D - d_k$, получим, что на любом допустимом решении целевые функции двух задач отличаются на константу D и, следовательно задачи эквивалентны между собой. Обозначим задачи с временами доставки через

$F2|p_{i,m} = \alpha_{i,m}(as_{i,m}+b), q_i|C_{max}$ и $O2|p_{i,m} = \alpha_{i,m}(as_{i,m}+b), q_i|C_{max}$. Тогда верно следующее утверждение.

Следствие 5 *1.Алгоритм AF находит расписание задачи $F2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b), q_i|C_{max}$ с погрешностью $R_A = 2$.*

2.Алгоритм AO находит расписание задачи $O2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b), q_i|C_{max}$ с погрешностью $R_A = 2$.

Доказательство неопсредственно следует из теоремы ??. Обозначим через C^*_{max} оптимальное решение задачи $F2|p_{i,m} = \alpha_{i,m}(as_{i,m}+b)|C_{max}$, а через C^q_{max} оптимальное решение задачи $F2|p_{i,m} = \alpha_{i,m}(as_{i,m}+b), q_i|C_{max}$. Тогда из (3.4) и неотрицательности величин q_i получаем, что $C^q_{max} \geq C^*_{max}$ и $C^q_{max} \geq \max_{J_i \in J} q_i$. С другой стороны так как алгоритм AF находит оптимальное расписание задачи $F2|p_{i,m} = \alpha_{i,m}(as_{i,m} + b)|C_{max}$, то

$$C_{max}(AF) \leq C^*_{max} + \max_{J_i \in J} q_i \leq 2C^q_{max}.$$

Заключение

Сформулируем основные результаты выполненных исследований.

1. Исследован новый широкий класс задач теории расписаний, в которых длительность работ зависит от момента начала ее выполнения.

2. Для случая, когда длительность каждой работы пропорциональна линейной функции изучена взаимосвязь получающихся задач с классическими задачами теории расписаний. В частности,

а) в задачах с одной машиной и различными критериями оптимизации построены точные полиномиальные алгоритмы обобщающие известные алгоритмы для задач с фиксированными длительностями.

б) Разработан метод, позволяющий по известным полиномиальным алгоритмам для задач с фиксированными длительностями строить полиномиальные алгортимы для задач с длительностями пропорциональными линейной функции. На основе этого метода построены точные полиномиальные алгоритмы для задач минимизации длины расписания в системах поточного и открытого типа на двух машинах.

в) Исследованы задачи с параллельными машинами и многооперационные задачи при условии, что длительность каждой работы пропорциональна моменту начала ее выполнения. Установлена NP-трудность в сильном смысле задачи минимизации длины расписания в системе поточного типа из трех и более машин. Показана NP-трудность задач минимизации длины расписания на двух и более параллельных машинах и в системе открытого типа из трех и более машин. Установлена NP-трудность задач минимизации максимального запаздывания в системах поточного и открытого типов из двух и более машин.

г) Рассмотрена задача минимизации суммарного времени завершения выполнения всех работ на двух идентичных машинах. Показано, что если длительность каждой работы пропорциональна моменту начала ее выполнения, то задача становится NP-трудной, в то время, как в случае фиксированных длительностей задача является полинлмиально-разрешимой.

3. Для случая, когда длительность каждой работы пропорциональна произвольной функции выделены новые широкие классы полиномиально-разрешимых задач. А именно,

а) построены точные полиномиальные алгоритмы для задачи минимизации длины расписания на одной машине для непрерывной выпуклой (или вогнутой) функции.

б) построен точный полиномиальный алгоритм для задачи минимизации суммарного времени завершения выполнения всех работ на одной машине для непрерывной выпуклой функции.

в) при дополнительных ограничениях на исходные данные задач построены точные полиномиальные алгоритмы для задач минимизации максимального запаздывания и взвешенного суммарного времени завершения выполнения всех работ на одной машине для непрерывной выпуклой (или вогнутой) функции.

4. Для случая, когда длительность каждой работы пропорциональна ступенчатой функции, установлена NP-трудность задач минимизации длины расписания даже если функция имеет только один скачок вверх или вниз.

5. Установлена NP-трудность задачи минимизации длины расписания, в которых длительность каждой работы определяется произвольной выпуклой неубывающей кусочно-линейной функцией.

Список литературы

[1] Бухштаб А.А. Теория чисел. М.: Учпедгиз, 1960.

[2] Гэри М.,Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.:Мир,1982.

[3] Кононов А.В. О расписаниях работ на одной машине с длительностями, нелинейно зависящими от времени *Дискретный анализ и исследование операций* Т.2, N 1. 1995. С. 21-35.

[4] Кононов А.В. Комбинаторная сложность составления расписаний для работ с простым линейным ростом длительностей *Дискретный анализ и исследование операций* Т.3, N 2. 1996. С. 15-32.

[5] Кононов А.В. Задачи теории расписаний на одной машине с длительностями работ, пропорциональными произвольной функции *Дискретный анализ и исследование операций* Т.5, N 3. 1998. С. 17-37.

[6] Кононов А.В. Задачи теории расписаний с длительностями, зависящими от времени. Алгоритмы и сложность. *Тезисы докладов. Второй Сибирский Конгресс по Прикладной и Индустриальной Математике (ИНПРИМ-96).* , Новосибирск 1996, с 137.

[7] Кононов А.В. Задачи теории расписаний с увеличением длительности работ за нарушение нормативных сроков. *Тезисы докладов.*

Международная конференция Проблемы оптимизации и экономические приложения. , Омск 1997, с 97.

[8] Мельников О.Н. Шафранский Я.М. Параметрическая задача теории расписаний. *Кибернетика.* 1979. N 3. C: 53-57.

[9] Пападимтриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность.*Мир,* 1985.-510 с.

[10] Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. *Мир,* 1980.-476 с.

[11] Танаев В.С., Гордон В.С., Шафранский Я.М. Теория расписаний. Одностадийные системы. *Наука. Москва.* 1984.

[12] Танаев В.С., Сотсков Ю.Н., Струсевич В.А. Теория расписаний. Многостадийные системы. *Наука. Москва.* 1989.

[13] Танаев В.С. Шкурба В.В. Введение в теорию расписаний. *Наука. Москва.* 1975.

[14] Alidaee B. A Heuristic Solution Procedure to Minimize Makespan on a Single Machine with Non-linear Cost Functions. *J. Opnl. Res. Soc.* V.41, 1990. P. 1065-1068.

[15] Alidaee B., Landram F. Scheduling Deteriorating Jobs on a single Processor. *Opns. Res. Letts.* V.17, 1996. P.507-510.

[16] Browne S. Yechiali U. Scheduling deterioration jobs on a single processor. *Oper. Res.* V.38, 1990. P. 495-498.

[17] Chen Z-L. A note on single-processor scheduling with time-dependent execution times. *Operations research letters*, 1995. V. 17 , №3. P. 127-129.

[18] Chen Z-L. Parallel machine scheduling with time dependent processing times. *Discrete Applied Mathematics*, V. 70 (1996), P. 81-93.

[19] Chen Z-L. Erratum to "Parallel machine scheduling with time dependent processing times."*Discrete Applied Mathematics , V. 75 (1997), P. 103.*

[20] Cheng, T.C., Ding Q. The complexity of scheduling starting time dependent tasks with release times. Information Processing Letters *V.65, 1998, 75-79.*

[21] Conway R.W., Maxwell W.L., Miller L.W. Theory of scheduling. Addison-Wesley, Reading, MA.

[22] Du J.,Leung J.Y.-T. Minimizing total tardiness on one machine is NP-hard. Math. Oper .Res. *1990. v.15, N. 3. P. 483-495.*

[23] Garey M.R. Johnson D.S. Computers and Intractability: a Guide to the Theory of NP-Completness. Freeman, San Francisco.

[24] Gawiejnowicz S. Brief survey of continuous models of scheduling. Foundations of Computing and Decision Sciences, vol. 21 (1996), no. 2, 81–100.

[25] Gawiejnowicz S., Pankowska L. Scheduling jobs with varying processing times. Information Processing Letters *V.54, 1995. P. 175-178.*

[26] Glazebrook K.D. Single-Machine Scheduling of stochastic Jobs Subject to Deterioration or Delay. Nav.Res.Logist.Quart. *V.39, 1992. P. 613-633.*

[27] *Gonzalez T., Sahni S. Open shop scheduling to minimize finish time.* J. Assoc. Comput. Mach. *V. 23. 1976. P. 665-679.*

[28] *Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G. Optimization and approximation in deterministic sequencing and scheduling: a survey* Ann.Discrete Math. *V.5. 1979. P. 287-326.*

[29] *Gupta J.N.D. Gupta S.K. Single facility scheduling with nonlinear processing times.* Computers ind. Engng. *V.14, N 4, 1988. P. 387-393.*

[30] *K.I.-J.Ho, J.Y.-T.Leung, W.D.Wei, Complexity of scheduling tasks with time-dependent execution times* Information Processing Letters *V.48, 1993, 315-320.*

[31] *Jackson J.R. Scheduling a production line to minimize maximum tardiness.* Res.rep. 43, Management Science Research Project, University of California, Los-Angeles.

[32] *Jonson S.M. Optimal two and three stage production schedules with set up times included* Naval.Res.Logist.Quart. *1954. V.1, N.1. P.61-68.*

[33] *Karp R.M. Reducibility among combinatorial problems.* R.E.Miller, J.W.Thatcher (eds.) Complexity of Computer Computations, Plenum Press, New York *1972. P.85-103.*

[34] *Kononov A.V. On schedules of a single machine jobs with processing times nonlinear in time. Operations Research and Discrete Analysis. Mathematics and Its Applications, Kluwer Academic Publishers, Dordrecht. 1997, 109-123.*

[35] *Kononov A.V. Scheduling Problems with Linear Increasing Processing Times.* Operations Research Proceedings 1996, Springer *1997. P. 208-212.*

[36] *Kononov A.V. Complexity of Scheduling Problems with the Time-Dependent Processing Times.* Third Workshop on Models and Algorithms for Planning and Scheduling Problems. Abstracts. *Cambridge, 1997, P. 29.*

[37] *Kononov A.V. Exact polynomial algorithms and complexity of some parametric scheduling problems on a single machine.* International Symposium on Combinatorial Optimization. Abstracts. *Brussel, 1998, P. 96.*

[38] *Kovalyov M.Y., Kubiak W. A fully polynopmial Approximation Scheme for Minimizing Makespan of Deteriorating Jobs* Journal of Heuristics *V 3, 1998. P.287-297.*

[39] *Kunnarthur A.S., Gupta S.K. Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem.* Europian J. Oper. Res. *V. 47, N 1. 1990. P. 56-64.*

[40] *Kubiak W., van de Velde S.L. Scheduling deteriorating jobs to minimize makespan.* Report LPOM-94-12, Laboratory of Production and Operations Management, Department of Mechanical Engineering, Univercity of Twente, *1994.*

[41] *Lawler E.L. A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness.* Ann. of Discrete Math. Amsterdam:North-Holland *1977. V 1. P.331-342.*

[42] *Lawler E.L. Optimal sequensing of a single machine subject to precedence constraints.* Management Sci. *1973, V 19, P. 544-546.*

[43] *E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, Sequensing and scheduling: Algorithms and complexity, in Handbooks in* Operation Research and Management Science, *Vol 4, North Holland, 1993, 445-522.*

[44] *Mosheiov G. Scheduling jobs under simple linear deterioration.* Computers Operations Research. *V.21, N 6, 1994. P. 653-659.*

[45] *Mosheiov G. V-shaped policies for scheduling deteriorating jobs.* Oper. Res. *V.39, N 6, 1991. P. 979-991.*

[46] *Moore J.M. An n job, one machine sequensing algorithm for minimizing the number of late jobs.* Management Sci. *V.15, P.102-109.*

[47] *Pilliboothamgudi, Sundararaghavan S., Kunnathur A.S. Single machine scheduling with start time dependent processing times: Some solvable cases.* European Journal of Operational Research, *V. 78 (1994), no. 3, 394–403.*

[48] *Woeginger G.J. Scheduling with time-dependent execution times. Information Processing Letters, vol. 54 (1995), 155–156.*