ИНФОРМАТИКА

2004 январь-март №1

УДК 519.8

В.С. Гордон, В.Н. Смотряев, А.А. Тарасевич

ПОСТРОЕНИЕ ОПТИМАЛЬНЫХ РАСПИСАНИЙ ПРИ НАЗНАЧЕНИИ ДИРЕКТИВНЫХ СРОКОВ

Приведен обзор методов решения задач теории расписаний, в которых, в отличие от традиционных формулировок, директивные сроки обслуживания требований не заданы априори, а являются параметрами, значения которых необходимо отыскать. Основное внимание уделяется полиномиальным алгоритмам решения задач построения оптимальных расписаний с назначением директивных сроков для систем с одним прибором.

Введение

Задачи определения наилучшей последовательности выполнения работ при заданных ограниченных ресурсах возникают при проектировании и управлении в промышленности и сфере обслуживания. Для решения такого рода задач применяются методы теории расписаний [1, 2]. Сроки, к которым необходимо выполнить работы, называются директивными, и задержка выполнения работ относительно директивных сроков приводит к затратам (теряется доверие клиента, выплачивается штраф и т.д.). Используя терминологию теории расписаний, вместо «выполнение работ» далее будем говорить об обслуживании множества требований приборами; затраты, связанные с задержками в обслуживании относительно директивного срока, отразим введением функции штрафов.

Часто встречаются ситуации, когда выполнение заказа с опережением также ведет к потерям. Например, поставка продукции по железной дороге раньше назначенного срока (когда заказчик не готов разгрузить ее) ведет к простою вагона и, следовательно, к дополнительным затратам. В этих условиях в планировании производства возникла так называемая концепция «точно-в-срок». Согласно этой концепции поступающие требования должны обслуживаться не ранее и не позднее их директивных сроков, и любое отклонение ведет к определенным затратам. Таким образом, функция штрафов зависит как от запаздывания, так и от опережения относительно директивных сроков.

Обычно сроки выполнения работ устанавливает заказчик (заданные директивные сроки). Но нередки ситуации, в которых компания сама предлагает клиенту свои директивные сроки исходя из объема работ и имеющихся ресурсов. Предлагая большой директивный срок, компания рискует потерять доверие клиента либо проиграть своим конкурентам. В таком случае, наряду с задачей поиска наилучшего расписания обслуживания требований, возникает необходимость выбора компромиссных директивных сроков.

Примером может служить тендер на строительство нескольких объектов. Для того чтобы выиграть тендер, строительная компания должна предложить достаточно малые сроки выполнения заказов, которые устроят заказчика и в то же время позволят компании построить эти объекты вовремя, избежав выплаты штрафов за задержку по условиям контракта. Данная ситуация может быть описана задачей построения расписания обслуживания требований одним прибором (строительная компания) и назначения соответствующих директивных сроков завершения обслуживания. Директивные сроки естественным образом зависят от трудоемкости сооружения объекта (например, в виде линейной зависимости от длительности обслуживания требования).

В данной работе приведен обзор методов решения задач теории расписаний, в которых, в отличие от традиционных формулировок, директивные сроки обслуживания требований не заданы априори, а являются параметрами, значения которых необходимо отыскать. Основное внимание уделено результатам, полученным в Объединенном институте проблем информатики Национальной академии наук Беларуси.

Наиболее распространены следующие модели назначения директивных сроков: CON (*constant flow allowance*), SLK (*slack*), TWK (*total work content*) и PPW (*processing plus wait*). В модели CON директивный срок является общим для всех требований; в модели SLK директивный срок определяется как время обслуживания плюс "задержка", одинаковая для всех требований; в модели TWK директивный срок определяется как произведение длительности обслуживания требования на некоторый коэффициент; в модели PPW директивные сроки – это линейные функции длительности обслуживания требования.

Формальные определения этих правил следующие. По правилу CON директивный срок d_j одинаков для всех требований *j*: $d_j = d$. Правило SLK определяет директивный срок требования как $d_j = r_j + p_j + q$, где r_j и p_j – момент поступления и длительность обслуживания требования *j*, а q – общая для всех требований задержка. Для модели TWK директивный срок определяется как $d_j = r_j + kp_j$, где множитель $k > 0$. Модель PPW назначения директивных сроков сочетает в себе модели CON, SLK и TWK: $d_j = r_j + kp_j + q$.

Отметим, что для моделей с последовательными приборами (типа flow shop, job shop и open shop) имеются единичные результаты по задачам построения оптимальных расписаний с назначениями директивных сроков, а модели с параллельными приборами рассмотрены в обзорах [3-5], поэтому далее рассматриваются только модели с одним прибором. Поскольку классификация задач с назначением директивных сроков по сложности их решения приведена в [4, 5], основное внимание далее уделяется полиномиальным алгоритмам построения оптимальных расписаний.

Основные определения и обозначения

В дальнейшем будем использовать следующие обозначения:

n – общее число требований;

r_j – момент поступления (готовности) требования *j*;

p_j – длительность обслуживания требования *j*;

d_j – директивный срок требования *j* – момент, в который желательно завершить обслуживание этого требования;

γ – штраф за единицу назначаемого директивного срока;

s – расписание обслуживания требований; в случае, когда расписание задается последовательностью, для нее будем использовать обозначение σ;

[*j*] – *j*-е по порядку требование в последовательности σ;

S_j и C_j – моменты соответственно начала и завершения обслуживания требования *j* (определяются расписанием);

α_j – величина штрафа за завершение обслуживания требования *j* на единицу времени раньше директивного срока d_j;

$E_j = \max\{0, d_j - C_j\}$ – опережение в обслуживании требования *j*;

$E = \{j \mid C_j \le d_j\}$ – множество требований, обслуженных не позднее директивных сроков;

β_j – величина штрафа за завершение обслуживания требования *j* на единицу времени позже директивного срока d_j;

$T_j = \max\{0, C_j - d_j\}$ – запаздывание требования *j* относительно директивного срока;

$T = \{j \mid C_j > d_j\}$ – множество требований, обслуживание которых было закончено позднее их директивных сроков;

U_j – параметр, показывающий, что требование *j* запаздывает, т.е. $U_j = 1$, если $j \in T$, и $U_j = 0$ в противном случае;

$L_j = C_j - d_j$ – временное смещение – величина, характеризующая смещение момента завершения обслуживания требования *j* от директивного срока;

F – минимизируемая функция, значение которой зависит от расписания *s* и директивных сроков.

Для обозначения задач примем стандартную для теории расписаний схему $A \mid B \mid C$, где $A = 1$ означает систему обслуживания с одним прибором, B характеризует параметры требований, а C задает критерий оптимальности [6,7]. Обозначения, применяемые в позиции *B*, следующие:

r_j – заданы моменты поступления требований;

$d_j := \rho(j)$ – директивные сроки назначаются по правилу $\rho(j)$; например, $d_j := d$ или $d_j := r_j + kp_j + q$, где *k* и *q* – некоторые константы;

pmtn – разрешены прерывания процесса обслуживания;

prec – задано отношение предшествования на множестве требований.

Примечание. *В позиции B может использоваться любая комбинация вышеприведенных обозначений и допускается отсутствие* r_j, *pmtn, prec. Отсутствие* r_j *обозначает одновременное поступление требований.*

Например, $1 \mid d_j := d, prec \mid \sum(\alpha_j E_j + \beta_j T_j + \gamma d)$ обозначает задачу обслуживания одновременно поступивших частично упорядоченных требований одним прибором, в которой необходимо найти общий для всех требований директивный срок и расписание, минимизирующее целевую функцию $\sum_{j=1}^{n}(\alpha_j E_j + \beta_j T_j + \gamma d)$.

Общий директивный срок

В этом разделе внимание уделено задачам, в которых необходимо обслужить все требования к определенному моменту (задачи с общим директивным сроком). Примерами ситуаций, приводящих к таким задачам, могут служить создание компонентов некоторого изделия к моменту окончательной сборки (общему директивному сроку), поставка различных товаров не позднее конца договорного периода или установление срока хранения смеси из нескольких компонент по времени хранения наиболее скоропортящейся составляющей.

Задача $1 \mid d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$. Здесь требования поступают на обслуживание одновременно ($r_j = 0$, $j = 1,2,\ldots,n$), штрафы за отклонение от директивных сроков не зависят от требований ($\alpha_j = \alpha$, $\beta_j = \beta$, $j = 1,2,\ldots,n$) и $\alpha,\beta,\gamma \geq 0$. Нетрудно заметить, что для оптимального директивного срока d^* выполняется неравенство $d^* \leq \sum_{j=1}^{n} p_j$ и для поиска оптимального расписания можно рассматривать расписания без промежуточных простоев прибора, для описания которых достаточно задать последовательность обслуживания требований (обозначим ее через σ). Тогда целевая функция имеет вид

$$F(d,\sigma) = \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \gamma d). \qquad (1)$$

В [8] доказана справедливость следующих свойств:

Если $\gamma \geq \beta$, то оптимальная последовательность σ^* строится по неубыванию длительностей обслуживания требований (правило SPT – сокращение от Shortest Processing Time).

Если $\gamma < \beta$, то для любой последовательности σ обслуживания требований выполняется следующее:

- оптимальное значение директивного срока равно моменту завершения обслуживания K-го по порядку требования, где $K = \lceil n(\beta - \gamma)/(\alpha + \beta) \rceil$;

- $F(d,\sigma)$ можно преобразовать к виду $F(d,\sigma) = \sum_{j=1}^{n} \lambda_j p_{[j]}$, где $\sigma = ([1],...,[n])$, $\lambda_j = n\gamma + (j-1)\alpha$, если $j \leq K$, и $\lambda_j = \beta(n+1-j)$, если $j > K$.

Поскольку выражение $\sum_{j=1}^{n} \lambda_j p_{[j]}$ принимает минимальное значение в случае, когда наименьшему коэффициенту λ (позиционному весу) ставится в соответствие наибольшее p, следующему наименьшему λ соответствует второе по величине p и т. д., то приходим к следующему алгоритму [8] решения задачи.

Алгоритм 1.

Вход: задача $1 \mid d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$.

Выход: оптимальное значение директивного срока d^* и оптимальная последовательность σ^*.

1. Построим SPT-последовательность σ'. Пронумеруем требования соответственно этой последовательности.
2. Если $\gamma \geq \beta$, то $d^* := 0$ и $\sigma^* = \sigma'$. СТОП.
3. Если $\gamma < \beta$, то $K := \lceil n(\beta - \gamma)/(\alpha + \beta) \rceil$.
4. Для $j \leq K$ вычисляем $\lambda_j = n\gamma + (j-1)\alpha$,
   для $j > K$ вычисляем $\lambda_j = \beta(n+1-j)$.
5. Проранжируем коэффициенты λ_j по убыванию (т.е. ранг самого большого коэффициента λ_j равен 1, а ранг самого малого – n). Построим оптимальную последовательность $\sigma^* = ([1],...,[j],...,[n])$, помещая на j-е место требование, номер i которого равен рангу весового коэффициента λ_j.
6. $d^* := p_{[1]} + p_{[2]} + ... + p_{[K]}$. СТОП.

Трудоемкость алгоритма: $O(n\log n)$.

Замечания. *1. Оптимальная последовательность, полученная с помощью приведенного алгоритма при $\gamma < \beta$, является V-образной, т.е. требования, обслуживаемые до директивного срока, располагаются в порядке невозрастания их времен обслуживания (правило LPT – сокращение от Longest Processing Time), а требования, обслуживаемые после директивного срока, располагаются в порядке неубывания их времен обслуживания (правило SPT).*

2. Пусть в рассмотренной задаче $\alpha = \beta = 1$ и $\gamma = 0$. Тогда из (1) получаем $F(d,\sigma) = \sum_{j=1}^{n}(E_j + T_j) = \sum_{j=1}^{n} |C_j - d|$ и тем самым приходим к задаче (если разделить целевую функцию на n) минимизации среднего абсолютного отклонения от общего директивного срока (задача MAD – сокращение от Mean Absolute Deviation); в этой задаче $d^* = C_{[K]}$, где

$[K]$ – K-е по порядку требование в последовательности σ^*, и $K = n/2$, если n четно, $K = (n+1)/2$, если n нечетно.

3. Пусть в рассмотренной задаче $\gamma = 0$. Тогда формулу (1) можно записать следующим образом: $F(d,\sigma) = \sum_{j=1}^{n}(\alpha E_j + \beta T_j) = \alpha\sum_{j\in E}\left|C_j - d\right| + \beta\sum_{j\in T}\left|C_j - d\right|$, т.е. целевой функцией задачи является взвешенная сумма абсолютного отклонения моментов завершения обслуживания требований от директивных сроков (задача WSAD – сокращение от Weighted Sum of Absolute Deviations); для этой задачи $d^* = C_{[K]}$ и $K = \lceil n\beta/(\alpha+\beta)\rceil$.

Задача $1 \mid r_j, d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$. В отличие от предыдущей задачи, здесь требования поступают на обслуживание неодновременно. При $\alpha = 0$ задача эквивалентна NP-трудной в сильном смысле задаче $1 \mid r_j \mid \sum C_j$ [9]. Если $\beta = 0$, то $d^* = 0$, поскольку увеличение директивного срока ведет к увеличению значения целевой функции; в этом случае любое допустимое расписание оптимально. Если $\gamma \ge \beta$, опять имеем $d^* = 0$; однако в этом случае задача эквивалентна задаче $1 \mid r_j \mid \sum C_j$ и является NP-трудной в сильном смысле. В общем случае задача NP-трудна в сильном смысле, поскольку даже при $\alpha = 0$ она является таковой. Далее будем рассматривать случай, когда $\gamma < \beta$.

При одинаковых длительностях либо при заданной последовательности обслуживания требований задача полиномиально разрешима. Алгоритмы решения основаны на следующих свойствах [10]:

Для заданной последовательности требований $\sigma = (1,2,\ldots,n)$ в оптимальном расписании $s(\sigma)$ не существует задержек между требованиями $(1,2,\ldots,u)$, где $u = \left\lceil \dfrac{n(\beta-\gamma)}{\beta} \right\rceil$.

Если $p_j = p$ и требования пронумерованы в порядке неубывания r_j, то существует оптимальное расписание, которое задается последовательностью $(1,2,\ldots,n)$.

Алгоритм 2.

Вход: задача $1 \mid r_j, d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$ при заданной последовательности $\sigma = (1,2,\ldots,n)$ обслуживания требований.

Выход: оптимальное значение директивного срока d^* и оптимальное расписание s^*.

1. Построим начальное расписание s^0 с моментами начала обслуживания требований, определяемыми по следующему правилу: $S_1(s^0) := r_1$, $S_j(s^0) := \max\{r_j, C_{j-1}(s^0)\}$.
2. Определим $u := \left\lceil \dfrac{n(\beta-\gamma)}{\beta} \right\rceil$.
3. Сдвинем требования $1,\ldots,u-1$ к требованию u таким образом, чтобы между ними не было перерывов в обслуживании.
4. Определим $K := \left\lceil \dfrac{n(\beta-\gamma)}{\alpha+\beta} \right\rceil$ и получим $d^* := C_{[K]}$.
5. СТОП.

Трудоемкость алгоритма: $O(n)$.

Алгоритм 3.

Вход: задача $1 \mid r_j, p_j = p, d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$.

Выход: оптимальное значение директивного срока d^* и оптимальное расписание s^*.

1. Перенумеруем требования в порядке неубывания их моментов поступления и зададим последовательность $\sigma = (1,2,\ldots,n)$.
2. Используем алгоритм 2 для получения результата.

Трудоемкость алгоритма: $O(n\log n)$.

Одним из способов получения приближенного решения для общего случая задачи $1 \mid r_j, d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$ является использование оптимального расписания для задачи $1 \mid d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$. Таким образом, приходим к следующему эвристическому алгоритму.

Алгоритм 4.

Вход: задача $1 \mid r_j, d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$.

Выход: директивный срок d^h и расписание s^h, являющиеся приближенным решением задачи.

1. При помощи алгоритма 1 построим оптимальную последовательность требований для задачи $1 \mid d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$.
2. Для получения приближенного решения исходной задачи $1 \mid r_j, d_j := d \mid \sum(\alpha E_j + \beta T_j + \gamma d)$ используем построенную последовательность, применяя алгоритм 2.

Трудоемкость алгоритма: $O(n\log n)$.

Оценка погрешности алгоритма 4: $f(d^h, s^h) / f(d^*, s^*) \le 1 + R$, где $f(d^h, s^h)$ и $f(d^*, s^*)$ – значения целевого функционала для решения, полученного с использованием данного алгоритма, и , соответственно, для оптимального решения, а $R = \max_{j \in N}\{r_j\}$.

Директивные сроки, зависящие от длительностей обслуживания требований

В данном разделе рассматриваются задачи с назначениями директивных сроков по правилам SLK, TWK и PPW.

Задача $1 \mid prec, C_j \le d_j := p_j + q \mid \varphi(F, q)$. В этой задаче директивные сроки назначаются по правилу SLK, требуется найти оптимальное значение задержки q и расписание, при котором ни одно из требований не запаздывает, а значение целевой функции φ минимально. Множество требований частично упорядочено. Функция φ неубывающая по обоим аргументам, где F, в свою очередь, – неубывающая функция от опережений E_j, которая имеет вид либо $F = \sum_{j=1}^{n} w_j E_j$, либо $F = \sum_{j=1}^{n} w_j \exp(\eta E_j)$, где $\eta > 0$. Особое внимание уделяется случаю, когда граф, задающий отношение порядка, является последовательно-параллельным.

В статье [11] предложена общая схема решения указанной задачи, включающая две стадии: основную (алгоритм 5) и вспомогательную (алгоритм 6). На вспомогательной стадии решается задача без назначения директивных сроков (при фиксированном значении q). Алгоритмы 5 и 6 применимы для произвольного графа, описывающего отношение предшествования, однако следует иметь в виду, что в этом случае задача NP-трудна [11]. Для случая последовательно-параллельного графа задача полиномиально разрешима: временная сложность алгоритма 6 в этом случае полиномиальна, и решение задачи алгоритмами 5 и 6 требует $O(n^2 \log n)$ операций. При построении алгоритмов используется следующее свойство [11].

Существует оптимальное расписание, при котором одно из требований, не имеющих потомков, обслуживается последним и завершает обслуживание в свой директивный срок, а остальные требования завершают обслуживание ранее директивных сроков.

Алгоритм 5.

Вход: задача $1 \mid prec, C_j \le d_j := p_j + q \mid \varphi(F, q)$.

Выход: оптимальное значение q^* задержки и последовательность σ^*, доставляющая минимальное значение φ^* целевой функции.

1. Пусть r – число требований, у которых нет потомков, $r \le n$. Перенумеруем требования так, чтобы эти r требований были первыми. Найдем убывающую последовательность $p^{(1)}, p^{(2)}, \ldots, p^{(t)}$ всех различных значений p_j, $j = 1,2,\ldots,r$. Найдем возрастающую последовательность $q^{(u)} = p(N) - p^{(u)}$, $u = 1,2,\ldots,t$, где $p(N) = \sum_{j=1}^{n} p_j$.
2. Найдем множество $N(q^{(1)})$, состоящее из всех требований j, $1 \le j \le r$, для которых $p_j = p^{(1)}$. Решим вспомогательную задачу $1 \mid prec, C_j \le d_j = p_j + q \mid F$ при $q = q^{(1)}$, предполагая, что только требования множества $N(q^{(1)})$ можно поставить на последнее место в последовательности. В результате получим значение $F(q^{(1)})$ и соответствующую оптимальную последовательность $\sigma^{(1)}$. Вычислим $\varphi^* = \varphi(F(q^{(1)}), q^{(1)})$. Полагая $q^* = q^{(1)}$, $\sigma^* = \sigma^{(1)}$ и $u = 2$, переходим к следующему шагу.
3. Найдем множество требований N', состоящее из всех тех требований j, $1 \le j \le r$, для которых $p_j = p^{(u)}$. Определим $N(q^{(u)}) = N(q^{(u-1)}) \cup N'$. Решим вспомогательную задачу $1 \mid prec, C_j \le d_j = p_j + q \mid F$ при $q = q^{(u)}$, предполагая, что только требования множества $N(q^{(u)})$ можно поставить на последнее место в последовательности. В результате получим наименьшее значение F' функции F и соответствующую оптимальную последовательность σ' требований. Если $F(q^{(u-1)}) > F'$, то положим $F(q^{(u)}) = F'$; иначе положим $F(q^{(u)}) = F(q^{(u-1)})$. Если $\varphi^* > \varphi(F(q^{(u)}), q^{(u)})$, то положим $\varphi^* = \varphi(F(q^{(u)}), q^{(u)})$, $\sigma^* = \sigma'$ и $q^* = q^{(u)}$. Полагая $u := u + 1$, повторяем этот шаг, пока не получим $u > t$.
4. СТОП.

Алгоритм 6 решает вспомогательную задачу на шагах 2 и 3 алгоритма 5.

Алгоритм 6.

Вход: задача $1 \mid prec, C_j \le d_j = p_j + q \mid F$ при $F = \sum_{j=1}^{n} w_j E_j$ либо $F = \sum_{j=1}^{n} w_j \exp(\eta E_j)$.

Выход: последовательность σ^* требований, которая определяет оптимальное расписание, и соответствующее ей значение целевой функции.

1. Для каждого требования m, которое можно поставить в последнюю позицию последовательности, введем вспомогательную задачу P_m со множеством $N \setminus m$ требований. Найдем последовательность $\sigma_m = (\pi_m, m)$, где π_m – оптимальная последовательность для задачи P_m. Задача P_m для функции $F = \sum_{j=1}^{n} w_j E_j$ – это задача $1 \mid prec \mid (-\sum w_j C_j)$, а для функции $F = \sum_{j=1}^{n} w_j \exp(\eta E_j)$ – это задача $1 \mid prec \mid Y(\sigma, n-1)$ минимизации функции $Y(\sigma, n-1) = \sum_{k=1}^{n-1} w'_{\sigma(k)} \exp(-\eta \sum_{j=1}^{k} p_{\sigma(j)})$, где $w'_j = w_j \exp(\eta(p_j + p(N)))$ и

$\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n))$ – последовательность требований, соответствующая допустимому расписанию.

2. Среди всех найденных последовательностей σ_m найдем ту, которой соответствует наименьшее значение функции $F(\sigma_m)$, и возьмем ее в качестве оптимальной последовательности σ^*.
3. СТОП.

Заметим, что на шаге 1 решение задачи $1 | prec | (-\sum w_j C_j)$ (или задачи $1 | prec | Y(\sigma, n-1))$ для графа отношения порядка $\vec{G}$ сводится к решению задачи $1 | prec | \sum w_j C_j$ (или, соответственно, $1 | prec | \sum w_j \exp(\eta C_j))$ для графа $\bar{G}$, полученного из $\vec{G}$ заменой всех дуг на противоположные. Эти задачи для последовательно-параллельных ограничений предшествования решаются за время $O(n \log n)$ [1]. Отсюда следует полиномиальная реализация алгоритмов 5 и 6.

Задача $1 | pmtn, prec, r_j, d_j := r_j + p_j + q | \gamma q + T_{\max}$, где директивные сроки назначаются по правилу SLK, $\gamma > 0$ – стоимость единицы задержки q, а $T_{\max} = \max_{1 \le j \le n} T_j$. Расписание, минимизирующее $\gamma q + T_{\max}$ при заданном значении q, строится с применением алгоритма [12, 13] решения задачи $1 | pmtn, prec, r_j | F_{\max}$, где $F_{\max} = \max_{1 \le j \le n} F_j$. Этот алгоритм, называемый по первым буквам фамилий авторов алгоритмом GT-BLLR, применим для произвольных неубывающих (от моментов завершения обслуживания) функций штрафа F_j и имеет временную сложность $O(n^2)$. Расписание, построенное для задачи $1 | pmtn, prec, r_j, d_j := r_j + p_j + q | \gamma q + T_{\max}$ по алгоритму GT-BLLR, оптимально независимо от величины q, что позволяет для решения задачи применить следующий алгоритм [14].

Алгоритм 7.

Вход: задача $1 | pmtn, prec, r_j, d_j := r_j + p_j + q | \gamma q + T_{\max}$.

Выход: оптимальное значение q и оптимальное расписание s^*.

1. Строим оптимальное расписание s^* по алгоритму GT-BLLR. Пусть l – требование с максимальным запаздыванием в s^*.
2. Находим оптимальное значение q^* по правилу: $q^* = C_l - r_l - p_l$, если $0 < \gamma < 1$; $q^* = 0$, если $\gamma > 1$; $q^* \ge C_l - r_l - p_l$, если $\gamma = 0$; $0 \le q^* \le C_l - r_l - p_l$, если $\gamma = 1$.
3. СТОП.

На шаге 2 в последних двух случаях q^* может принимать произвольные значения в указанных пределах.

Задача $1 | d_j := kp_j | \sum(\alpha_j E_j + \beta_j T_j)$. Директивные сроки назначаются по правилу TWK. Очевидно, все требования с одинаковой длительностью обслуживания имеют одинаковый директивный срок. Пусть m – число различных длительностей обслуживания p_j и пусть $t_1 < t_2 < \dots < t_m$ – эти различные значения. Пусть J_i – множество требований длительности t_i. Все требования из J_i имеют общий директивный срок, и известно [8, 15], что в оптимальном расписании одно из требований завершает обслуживание точно в этот срок. Пусть q_i – это число требований из J_i, завершивших обслуживание при оптимальном расписании позже директивного срока. В [16] предлагается следующий алгоритм сложности $O(n\log n)$.

Алгоритм 8.

Вход: задача $1 \mid d_j := kp_j \mid \sum(\alpha_j E_j + \beta_j T_j)$.

Выход: оптимальное значение *k* и оптимальное расписание.

1. Решаем m независимых задач для множеств J_i, $i = 1,2,\ldots,m$, каждая из которых представляет собой задачу с общим директивным сроком и одинаковыми длительностями. Для их решения применим алгоритм из [15].
2. Оптимальное расписание получаем стыковкой m решений, полученных на первом шаге, при выборе значения *k* таким, чтобы выполнялись условия: $(k - |J_i| + q_i)t_i \geq (k + q_{i-1})t_{i-1}$ для всех $i = 2,3,\ldots,m$ и $(k - |J_i| + q_i)t_i \geq 0$ для всех $i = 1,2,\ldots,m$. Эти условия обеспечивают неперекрываемость интервалов обслуживания для каждого из множеств J_i.

Заметим, что при внесении в целевую функцию параметра γk, ограничивающего значения директивных сроков, задача $1 \mid d_j := kp_j \mid \gamma k + \sum(\alpha_j E_j + \beta_j T_j)$ становится NP-трудной [16].

Задача $1 \mid d_j := kp_j \mid \Phi(k) + \alpha T_{\max}$, где $\alpha > 0$ и $\Phi(k)$ – выпуклая неубывающая функция. Директивные сроки назначаются по правилу TWK. Для решения задачи предложен следующий алгоритм сложности $O(n\log n)$ [17].

Алгоритм 9.

Вход: задача $1 \mid d_j := kp_j \mid \Phi(k) + \alpha T_{\max}$.

Выход: оптимальная последовательность обслуживания требований.

1. Оптимальную последовательность строим по правилу SPT (требования обслуживаются в порядке неубывания длительностей обслуживания, а значит, и директивных сроков).

Задача $1 \mid pmtn, prec, r_j, d_j := kp_j^c \mid \Phi(k,c) + \alpha T_{\max}$, где $\Phi(k,c)$ – выпуклая неубывающая от *k* и *c* функция, $c \geq 1$ и $\Phi(0,0) = 0$. Директивные сроки назначаются по так называемому TWK-power правилу, т.е. длительности обслуживания возводятся в степень *c* и, кроме этого, умножаются на коэффициент *k*.

Алгоритм 10 [18].

Вход: задача $1 \mid pmtn, prec, r_j, d_j := kp_j^c \mid \Phi(k,c) + \alpha T_{\max}$.

Выход: оптимальное расписание обслуживания требований.

1. Оптимальное расписание строим по алгоритму GT-BLLR.

Полученное расписание оптимально независимо от значений *k* и *c* [18].

Задача $1 \mid pmtn, prec, r_j, d_j := kp_j + q \mid \Phi(k,q) + \alpha T_{\max}$, где $\Phi(k,q)$ – выпуклая неубывающая от *k* и *q* функция и $\Phi(0,0) = 0$. Директивные сроки назначаются по правилу PPW.

Алгоритм 11 [18].

Вход: задача $1 \mid pmtn, prec, r_j, d_j := kp_j + q \mid \Phi(k,q) + \alpha T_{\max}$.

Выход: оптимальное расписание обслуживания требований.

1. Оптимальное расписание строим по алгоритму GT-BLLR.

Полученное расписание оптимально независимо от значений k и q [18]. Способ поиска оптимальных значений k, q и c в алгоритмах 9-11 зависит от конкретного вида функции Φ.

Задача $1 \mid d_j := kp_j + q \mid \max_{1\le j\le n}\left|L_j\right|$. Директивные сроки назначаются по правилу PPW.

Алгоритм 12 [19, 20].

Вход: задача $1 \mid d_j := kp_j + q \mid \max_{1\le j\le n}\left|L_j\right|$.

Выход: оптимальные значения параметров k и q, оптимальная последовательность обслуживания требований.

1. Оптимальную последовательность строим по правилу SPT.
2. Для нахождения оптимальных значений параметров k и q применяем простую графическую процедуру [20] сложности $O(n\log n)$.

Заключение

В данной статье рассмотрены алгоритмы полиномиальной временной сложности решения задач построения оптимальных расписаний для систем с одним прибором и назначением директивных сроков. Вообще говоря, полиномиальные алгоритмы применимы к решению только специальных случаев задач теории расписаний с назначением директивных сроков, поскольку в общем случае эти задачи NP-трудны. Однако эти алгоритмы можно использовать при создании и тестировании эвристических и приближенных алгоритмов, пригодных для решения задач в общем случае.

Работа выполнена при частичной поддержке БРФФИ (проекты Ф02Р-013 и Ф03МС-039).

Список литературы

1. Танаев В.С., Гордон В.С., Шафранский Я.М. Теория расписаний. Одностадийные системы. – М.: Наука, 1984. – 382 с.

2. Танаев В.С., Сотсков Ю.Н., Струсевич В.А. Теория расписаний. Многостадийные системы. – М.: Наука, 1989. – 328 с.

3. Cheng T.C.E., Gupta M.C. Survey of scheduling research involving due date determination decisions // European J. Opl. Res. – 1989. – 38, – P.156-166.

4. Gordon V., Proth J.-M., Chu C. A survey of the state-of-the-art of common due date assignment and scheduling // European J. Opl. Res. – 2002. – V.139. – P.1-25.

5. Gordon V., Proth J.-M., Chu C. Due date assignment and scheduling: SLK, TWK and other due date assignment models // Production Planning & Control. – 2002. – V. 13. – N.2. – P. 117-132.

6. Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B. Sequencing and scheduling : algorithms and complexity // Handbooks in Operations Research and Management Science. – Amsterdam: North Holland, 1993. – V. 4. – P. 445-522.

7. Chen B., Potts C.N., Woeginger G.J. A review of machine scheduling: complexity, algorithms and approximability // Handbook of Combinatorial Optimization. – Dordrecht: Kluwer, 1998. – P. 21-169.

8. Panwalkar S.S., Smith M.L., Seidmann A. Common due date assignment to minimize total penalty for the one machine scheduling problem // Oper. Res. – 1982. – V.30. – P.391-399.

9. Lenstra J.K., Rinnooy Kan A.H.G., Brucker P. Complexity of machine scheduling problems // Ann. Discrete Math. – 1977. – V.1 – P. 343-362.

10. Cheng T.C.E., Chen Z-L., Shakhlevich N.V. Common due date assignment and scheduling with ready times // Comp. Oper. Res. – 2002. – V. 29. – P.1957-1967.

11. Gordon V.S., Strusevich V.A. Earliness penalties on a single machine subject to precedence constraints: SLK due date assignment // Comp. Oper. Res. – 1999. – V. 26. – P.157-177,

12. Гордон В.С., Танаев В.С. О минимаксных задачах теории расписаний с одним прибором // Изв. АН БССР. Сер. физ.-мат. наук. – 1983. – Т. 3. – С. 3-9.

13. Baker K.R., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G. Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints // Opns. Res. – 1983. – V. 31. – P. 381-386.

14. Gordon V.S. A note on optimal assignment of slack due-dates in single-machine scheduling // European J. Opl. Res. – 1993. – V. 70. – P. 311-315.

15. Hoogeveen J. A., van de Velde S. L. Scheduling around a small common due date // European J. Opl. Res. – 1991. – V. 55. – P. 237-242.

16. Chu C., Gordon V. TWK due date determination and scheduling: NP-hardness and polynomially solvable case // Int. J. Mathl. Algorithms.– 2001. – V. 2. – P. 251-267.

17. Cheng T.C.E. Optimal assignment of slack due-dates and sequencing in a single-machine shop // Appl. Math. Lett. – 1989. – V. 2. – P. 333-335.

18. Cheng T.C.E., Gordon V.S. Optimal assignment of due-dates for preemptive single-machine scheduling // Mathl. Comput. Modelling.– 1994. – V. 20. – P. 33-40.

19. Kahlbacher H.G., Cheng T.C.E. Processing-plus-wait due dates in single-machine scheduling // J. Opt. Theory Appl. – 1995. – V. 85. – P. 163-186.

20. Cheng T.C.E., Kovalyov M.Y. Complexity of parallel machine scheduling with processing-plus-wait due dates to minimize maximum absolute lateness // European J. Opl. Res. – 1999. – V. 114. – P. 403-410.

Поступила 22.01.04

Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова,6
e-mail: gordon@newman.bas-net.by

V.S. Gordon, V.N. Smotrjaev, A.A. Tarasevich

SCHEDULING UNDER DUE DATE ASSIGNMENT

A survey of scheduling methods is given for the problems where due dates, in contrast to traditional formulations, are not given a priori but are to be found. A special attention is focused on polynomial algorithms for the single machine scheduling and due date assignment problems.