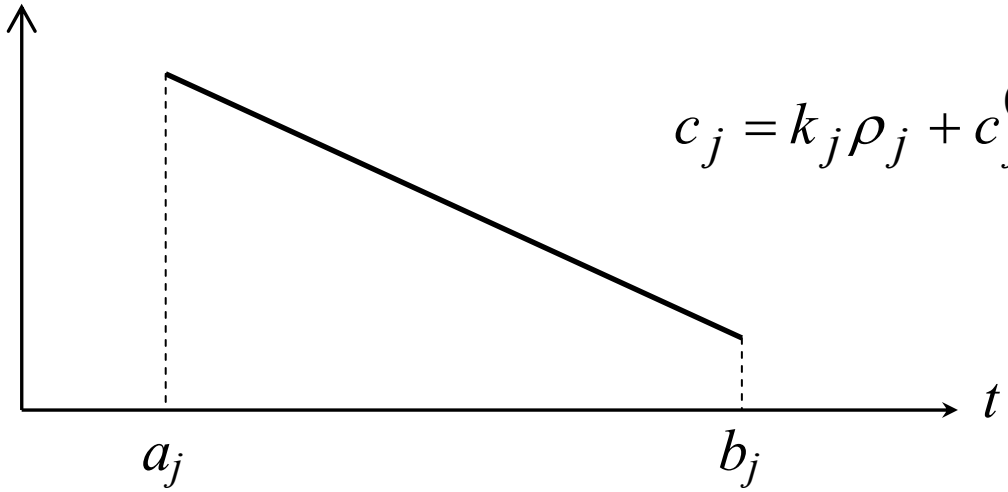
Анализ проекта по критерию «стоимость–длительность»

Для каждой работы $j \in J$ вместо длительности τ_j задан временной интервал $[a_j, b_j]$ и линейная функция c_j стоимости выполнения работы в зависимости от длительности.



Требуется найти минимальные суммарные затраты на выполнение всех работ и соответствующие им длительности работ, чтобы закончить проект к заданному сроку T^*.

Модель линейного программирования

Дано: $J = \{1, \ldots, n\}$ — множество работ;

$C = \{(i, j) \mid i, j \in J\}$ — частичный порядок;

$[a_j, b_j]$ — допустимый интервал выполнения работы j;

$c_j = k_j \rho_j + c_j^0$ — стоимость выполнения работы j;

T^* — время окончания всего проекта.

Найти: Минимальные суммарные затраты и длительность выполнения каждой работы для окончания проекта к заданному сроку T^*.

Модель:

$$\min_{\rho_j} \sum_{j \in J} (k_j \rho_j + c_j^0)$$

при ограничениях

$$s_i + \rho_i \le s_j, \text{ для всех } (i, j) \in C$$

$$s_t \le T^*$$

$$a_j \le \rho_j \le b_j, \quad j \in J.$$

Обратная задача

Дано: $J = \{1,\ldots, n\}$ — множество работ;

$C = \{(i, j) \mid i, j \in J\}$ — частичный порядок;

$[a_j, b_j]$ — допустимый интервал выполнения работы j;

$c_j = k_j \rho_j + c_j^0$ — стоимость выполнения работы j;

C^* — суммарная стоимость выполнения всех работ.

Найти: Наиболее раннее время выполнения проекта при условии, что суммарные стоимость выполнения всех работ не превысит величины C^*.

Модель:

$$\min s_t$$

при ограничениях

$$s_i + \rho_i \le s_j, \text{ для всех } (i, j) \in C,$$

$$\sum_{j \in J} (k_j \rho_j + c_j^0) \le C^*,$$

$$a_j \le \rho_j \le b_j, \quad j \in J.$$

Пакеты линейного программирования

ABACUS 2.3	2004	http://www.informatik.uni-koeln.de/abacus/
BCP	2004	http://www.coin-or.org/
BonsaiG 2.8	2004	http://www.cs.sfu.ca/~lou/BonsaiG/dwnldreq.html
CBC	2004	http://www.coin-or.org/.
GLPK 4.2	2004	http://www.gnu.org/software/glpk/glpk.html
lp solve 5.1	2004	http://groups.yahoo.com/group/lp_solve/
MINTO 3.1	2004	http://coral.ie.lehigh.edu/minto/
SYMPHONY 5.0	2004	http://www.branchandcut.org/SYMPHONY/

Пример

	Длительность	$T_P(старт)$	$T_П(старт)$	Затраты
A	3	0	5	2 100
B	5	0	0	5 000
C	3	5	5	1 800
D	4	8	8	4 800
E	8	12	12	32 000
F	2	8	11	1 000
G	4	10	13	2 800
H	2	10	18	7 000
I	5	5	15	4 000
J	3	14	17	30 000

Итого: 90 500

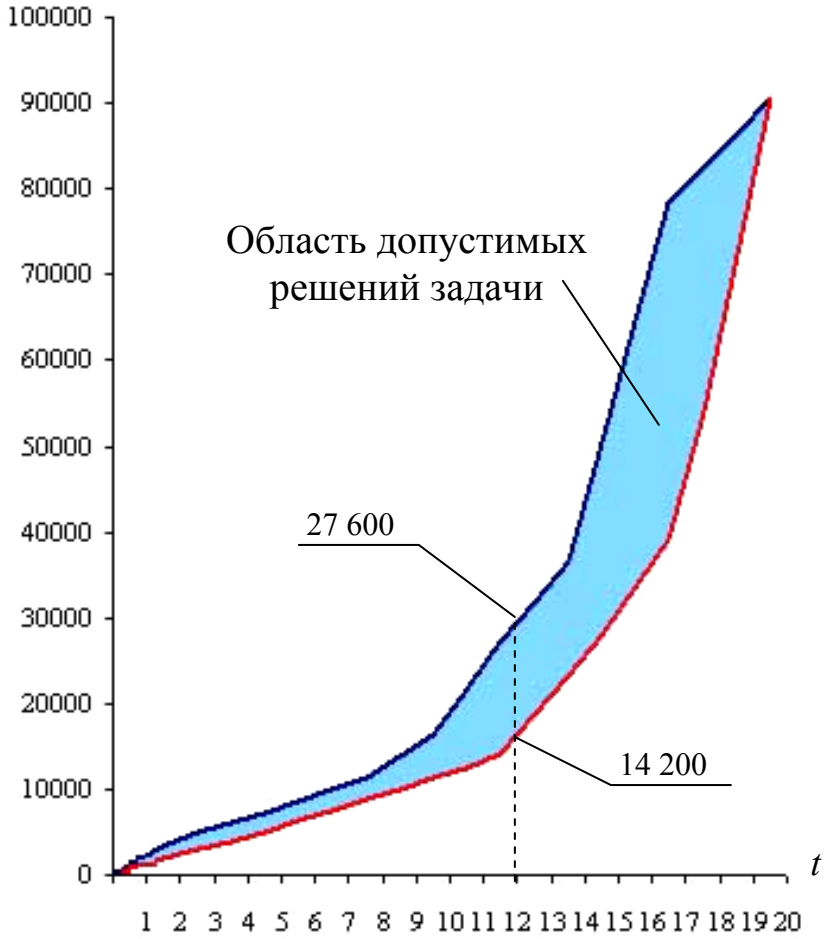
Еженедельные затраты при наиболее раннем старте работ

	Недели																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	700	700	700																	
B	1000	1000	1000	1000	1000															
C						600	600	600												
D									1200	1200	1200	1200								
E													4000	4000	4000	4000	4000	4000	4000	4000
F									500	500										
G											700	700	700	700						
H											3500	3500								
I						800	800	800	800	800										
J															10000	10000	10000			
В неделю	1700	1700	1700	1000	1000	1400	1400	1400	2500	2500	5400	5400	4700	4700	14000	14000	14000	4000	4000	4000
Всего за проект	1700	3400	5100	6100	7100	8500	9900	11300	13800	16300	21700	27100	31800	36500	50500	64500	78500	82500	86500	90500

Еженедельные затраты при наиболее позднем старте работ

	Недели																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A						700	700	700												
B	1000	1000	1000	1000	1000															
C						600	600	600												
D									1200	1200	1200	1200								
E													4000	4000	4000	4000	4000	4000	4000	4000
F												500	500							
G														700	700	700	700			
H																			3500	3500
I																800	800	800	800	800
J																		10000	10000	10000
В неделю	1000	1000	1000	1000	1000	1300	1300	1300	1200	1200	1200	1700	4500	4700	4700	5500	5500	14800	18300	18300
Всего за проект	1000	2000	3000	4000	5000	6300	7600	8900	10100	11300	12500	14200	18700	23400	28100	33600	39100	53900	72200	90500

Управление финансами



Затраты к концу 11-й недели

	Завершенность (%)	Общие затраты (план)	Текущие затраты (план)	Текущие затраты (факт)	Перерасход
A	100	2100	2100	2300	200
B	100	5000	5000	4900	–100
C	100	1800	1800	1800	0
D	75	4800	3600	4600	1000
E	0	32000	0	0	0
F	100	1000	1000	1200	200
G	25	2800	700	1400	700
H	50	7000	3500	5400	1900
I	20	4000	800	500	–300
J	0	30000	0	0	0
Итого		90500	18500	22100	3600

Календарное планирование с ограниченными ресурсами

Три типа ресурсов:

- *Возобновляемые ресурсы* (рабочее время, эфирное время, производственные мощности,…) если не потратили, то пропадают.
- *Складируемые ресурсы* (пиломатериалы, деньги, зерно, ГСМ, …) если не потратили, то можно потратить позже.
- *Невозобновляемые ресурсы* (кредиты, суммарный запас непополняемых материалов,…) можно потратить в любой момент выполнения проекта.

Постановка задачи

Дано:

$J = \{1,\ldots, n\}$ — множество работ;

$C = \{(i, j) \mid i, j \in J\}$ — частичный порядок;

$M_j, j \in J$ — моды (варианты) выполнения работы;

K^{ν} — множество невозобновляемых ресурсов;

K^{ρ} — множество возобновляемых ресурсов;

K^{σ} — множество складируемых ресурсов;

$r_{kj}^{m}(\tau) \geq 0$ — потребность в ресурсе $k \in K^{\rho} \cup K^{\sigma}$ в интервале времени $[\tau–1, \tau]$, считая от начала выполнения работы j в моде m;

$r_{kj}^{m} \geq 0$ — потребность в невозобновляемом ресурсе $k \in K^{\nu}$ при выполнении работы j в моде m;

$q_k(t) \geq 0$ — количество ресурса $k \in K^{\rho} \cup K^{\sigma}$, выделяемое в момент времени t на выполнение всех работ;

$q_k > 0$ — количество невозобновляемого ресурса $k \in K^{\nu}$, выделяемое в момент $t = 0$ на выполнение всех работ в течение всего интервала планирования;

$\rho_j^m \geq 0$ — длительность работы j в моде m;

$d_j \geq 0$ — директивный срок окончания работы j.

Переменные задачи:

$m(j)$ — мода работы j;

$s_j \geq 0$ — начало выполнения работы j

$A(t) = \{j \in J \mid s_j < t \leq s_j + \rho_j^m\}$ — множество работ, выполняемых в момент времени t.

Допустимое решение задачи: назначение моды $m(j)$ и времени старта каждой работы так, чтобы выполнить

- условия предшествования;
- ограничения по ресурсам;
- директивные сроки окончания каждой работы.

Задача состоит в том, чтобы найти допустимое решение задачи с минимальным временем окончания всех работ.

Математическая модель

$$\min\left\{\max_{j\in J}(s_j+\rho_j^{m(j)})\right\} \tag{1}$$

при ограничениях

$$s_j+\rho_j^{m(j)}\le d_j,\quad j\in J; \tag{2}$$

$$s_i+\rho_i^{m(i)}\le s_j,\,(i,j)\in C; \tag{3}$$

$$\sum_{j\in A(t)} r_{kj}^{m(j)}(t-s_j)\le q_k(t),\ \ k\in K^{\rho},\,t\ge 0; \tag{4}$$

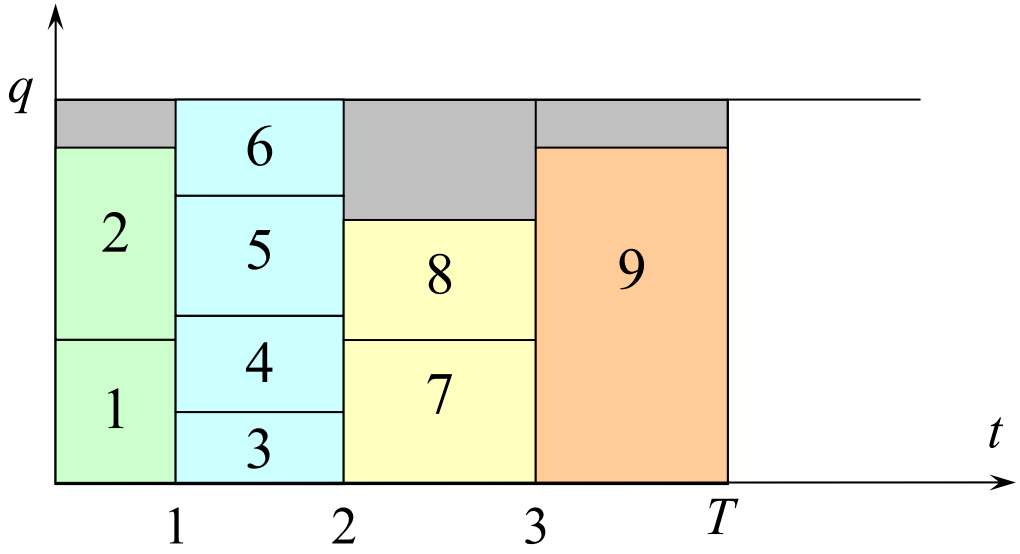
$$\sum_{t'=1}^{t}\sum_{j\in A(t')} r_{kj}^{m(j)}(t'-s_j)\le\sum_{t'=1}^{t} q_k(t'),\ \ k\in K^{\sigma},\,t\ge 0; \tag{5}$$

$$\sum_{j\in J} r_{kj}^{m(j)}\le q_k,\quad k\in K^{\nu}; \tag{6}$$

$$m(j)\in M_j,\ \ s_j\ge 0,\ \ j\in J. \tag{7}$$

Теорема. Задача (1) – (7) является NP–трудной при наличии хотя бы одного возобновляемого ресурса и $|M_j| = 1$ для всех $j \in J$.

Доказательство. Пусть $|M_j| = 1$ для всех $j \in J$, $K^\sigma = K^\upsilon = \varnothing$, $|K^\rho| = 1$, $q_k(t) = q$ для всех $t > 0$ и $k \in K^\rho$. Рассмотрим задачу (1) – (7) при $\rho_j = 1$ для всех $j \in J$ и без условий предшествования на множестве работ. Получим задачу упаковки в контейнеры, которая является NP–трудной. ■



Частные случаи модели

Для $|M_j|=1$, $j \in J$, рассмотрим следующие частные случаи:

1. $K^{\rho} = K^{\sigma} = \varnothing$, $K^{\nu} \neq \varnothing$ — только невозобновляемые ресурсы.

Тривиальный случай, проверяем хватает ли ресурсов и если да, то отбрасываем все ресурсные ограничения.

2. $K^{\rho} = K^{\nu} = \varnothing$, $K^{\sigma} \neq \varnothing$ — только складируемые ресурсы.

Полиномиально разрешимый случай, алгоритм Гимади дает точное решение задачи.

3. $K^{\sigma} = K^{\nu} = \varnothing$, $K^{\rho} \neq \varnothing$ — только возобновляемые ресурсы.

Наиболее трудный случай, применяем эвристические алгоритмы.

Задача со складируемыми ресурсами

Определение. Расписание $S = \{s_j \geq 0, j \in J\}$ называют *полудопустимым*, если оно удовлетворяет условиям предшествования (3) и директивным срокам (2), но быть может не удовлетворяет ограничениям по ресурсам.

Упражнение 1. Полудопустимое расписание существует $\Leftrightarrow$ наиболее ранние сроки выполнения работ $T_p(x)$ удовлетворяют директивным срокам (2).

Определение. Полудопустимое расписание $S^T = \{s_j^T \geq 0, j \in J\}$ для $T \geq T_{Kp}$ называется *T–поздним*, если $s_j^T + \rho_j \leq T$, для всех $j \in J$ и ни одно из значений s_j^T не может быть увеличено без нарушения этого условия или полудопустимости расписания.

Алгоритм вычислении T-позднего расписания аналогичен алгоритму вычисления наиболее поздних моментов, если перед началом вычислений положить

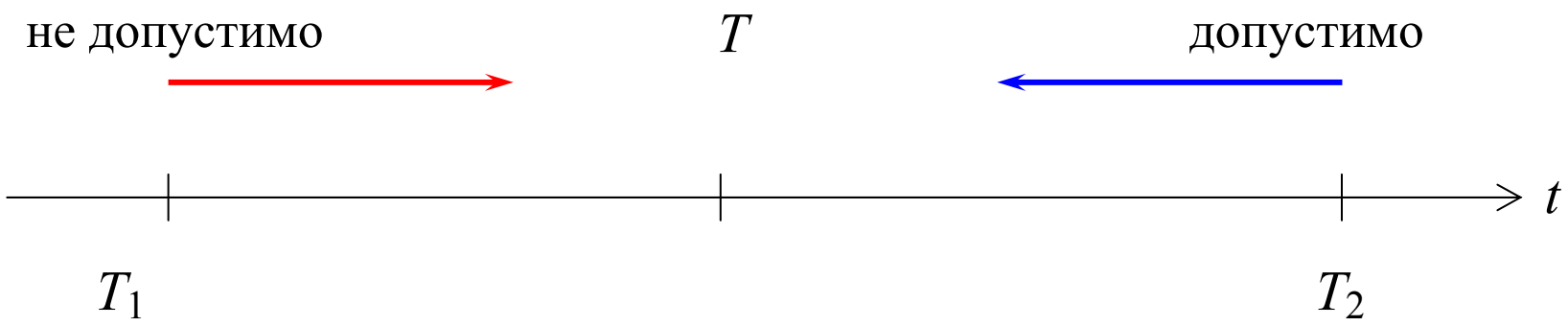
$$T_П(y) = \min \{d_j , T\}, \ j=(x,y) \in J.$$

Теорема. Пусть T^* — длина оптимального расписания. Тогда T^*–позднее расписание является оптимальным.

Доказательство: По определению T^*–позднее расписание удовлетворяет всем ограничениям кроме, быть может, ресурсных ограничений (5). Ограничения (4)–(6) игнорируются, так как $K^\rho = K^\nu = \varnothing$. Покажем, что ограничения (5) также выполняются.

Пусть S^* — оптимальное решение и S^{T^*} — T^*–позднее расписание. Из определения T–позднего расписания следует, что $s_j^* \le s_j^{T^*}$, для всех $j \in J$. Так как S^* удовлетворяет (5), то есть в каждый момент времени ресурсов достаточно, то эти же ресурсы могут быть использованы и позже в силу их складируемости. Значит, для $S_j^{T^*}$ тоже хватает ресурсов и (5) верно. Следовательно $S_j^{T^*}$ — оптимальное расписание. ■

Идея алгоритма Гимади



Алгоритм Гимади

1. Если наиболее ранние сроки выполнения работ $T_p(x)$ не удовлетворяют директивным срокам, то STOP, задача не имеет решения.
2. Положить $T_1 = T_{Kp}$.
3. Если T_1–позднее расписание допустимо по ресурсам, то STOP, получено оптимальное решение задачи.
4. Положить $T_2 = \max_{j \in J} d_j$.
5. Если T_2–позднее расписание недопустимо по ресурсам, то STOP, задача не имеет решения.
6. Положить $T := \lceil (T_1 + T_2)/2 \rceil$.
7. Построить T–позднее расписание S^T.
8. Если S^T допустимо по ресурсам, то $T_2 = T$, иначе $T_1 = T$.
9. Если $T_2 - T_1 > 1$, то вернуться на 6, иначе STOP, получено оптимальное решение S^{T_2}.

Задача с возобновляемыми ресурсами

Будем предполагать, что $q_k(t) = q_k$, и $r_{jk}(\tau) = r_{jk}, j \in J, k \in K^{\rho}$, т.е. выделение и потребление ресурсов не зависит от времени.

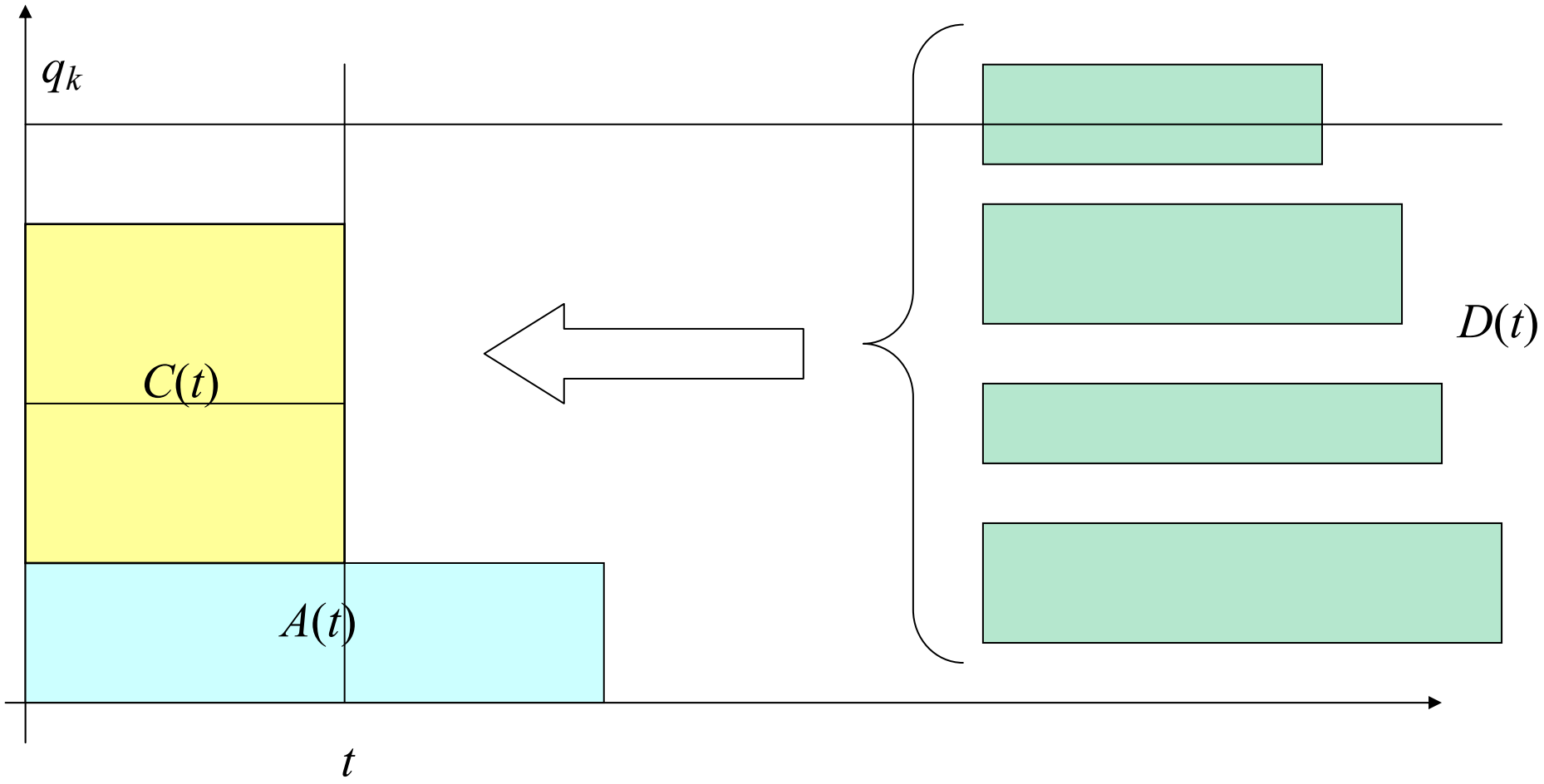
Для произвольного расписания $S = \{s_j \geq 0, j \in J\}$ и момента времени $t \geq 0$ определим три множества:

$C(t) = \{ j \in J \mid s_j + \rho_j < t\}$ — завершенные работы

$A(t) = \{ j \in J \mid s_j < t \leq s_j + \rho_j \}$ — выполняемые работы

$D(t) = \{ j \in J \setminus (C(t) \cup A(t)) \mid P(j) \subseteq C(t), \sum_{i \in A(t)} r_{ik} + r_{jk} \leq q_k, k \in K^{\rho}\}$ — работы, которые можно начать в момент времени t согласно условиям предшествования и ограничениям по ресурсам, $P(j)$ — множество предшественников работы j.

Идея алгоритма



Основной вопрос: как выбрать подмножество $D' \subseteq D(t)$?

Приближенный алгоритм

1. $t_0 := 0$, $C(t_0) := \varnothing$, $A(t_0) := \varnothing$, $l := 0$;
2. Пока $|\, C(t_l) \cup A(t_l) \,| < |\, J \,|$ выполнять

   2.1. $l := l + 1$

   2.2. $t_l := \min \{ s_j + \rho_j \,|\, j \in A(t_{l-1}) \}$

   2.3. Найти множества $D(t_l)$, $A(t_l)$, $C(t_l)$

   2.4. Положить $Q_k := q_k - \sum_{j \in A(t_l)} r_{jk}, \quad k \in K^{\rho}$

   2.5. Выбрать $D' \subseteq D(t_l)$ так, чтобы $\sum_{j \in D'} r_{jk} \le Q_k$ для всех $k \in K^{\rho}$

   2.6. Положить $s_j := t_l$ для всех $j \in D'$ и $A(t_l) := A(t_l) \cup D'$.

Выбор множества D'

Обозначим через $R_j = \sum_{k \in K^\rho} r_{jk} / q_k$, $j \in D(t_l)$ долю ресурсов, потребляемых работой j. Тогда D' целесообразно выбирать по решению задачи о рюкзаке:

$$\max \sum_{j \in D(t_l)} R_j x_j$$

при ограничениях $\sum_{j \in D(t_l)} r_{jk} x_j \le q_k - \sum_{i \in A(t_l)} r_{ik}, \quad k \in K^\rho,$

$$x_j \in \{0, 1\}, \; j \in D(t_l).$$

Полагаем $D' = \{ j \in D(t_l) \mid x_j^* = 1 \}$, где x_j^* — оптимальное решение задачи.

Упражнение 2. Показать, что любое правило выбора подмножества D', имеющее полиномиальную трудоемкость, не может гарантировать получение оптимального решения, если $P \neq NP$.