УДК 519.854

СТАЦИОНАРНЫЕ РАСПИСАНИЯ ДЛЯ ЦИКЛИЧЕСКИХ СЕТЕВЫХ ГРАФИКОВ С ОГРАНИЧЕНИЯМИ НА РЕСУРСЫ

Н.С.Григорьева

В статье рассматривается метод решения задачи составления оптимальных стационарных расписаний для повторяющегося комплекса взаимосвязанных работ. Комплекс работ задается циклическим сетевым графиком [1].Предлагаемый ниже алгоритм является дальнейшим развитием общей схемы метода решения задачи, описанной в [2]. Формулируются условия конечности алгоритма и обсуждаются модификации, повышающие его эффективность.

1. Постановка задачи

Циклический сетевой график описывает многократное выполнение одного и того же проекта. Он задается графом $\Gamma=\langle M,R\rangle$, в котором множество дуг R составлено из множества реальных работ N, множества фиктивных дуг F и множества дуг переходов H. Предполагается, что каждая дуга лежит на каком-либо контуре и каждый контур содержит хотя бы одну дугу из R. Дуги-переходы из H задают связи между последовательными реализациями проекта, дуги N и F имеют тот же смысл, что и в обычных сетевых графиках.

Множество N разбито на непересекающиеся подмножества

$$N=N_1\cup N_2\cup N_3\cup\ldots\cup N_p,$$

и для выполнения каждой работы из N_i требуется одна бригада i-го типа. Задано число S_i бригад каждого i-го типа. Предполагается, что выполнение работы не прерывается до ее окончания и в любой момент времени число выполняемых работ из N_i не превосходит S_i.

Зададим неотрицательный вектор продолжительностей работ $t[R]$, где $t[FUH]=0[FUH]$. Расписанием будем называть пару векторов $w[N], I[N]$, где $w[u]$ - время начала выполнения работы u, $I[u]$ - номер бригады, выполняющей работу u. Расписание называется стационарным, если времена наступления событий в каждой следующей реализации проекта увеличиваются в сравнении с предыдущими на величину z, которую будем называть шагом расписания.

Требуется составить стационарное расписание с минимальным шагом.

Для решения этой задачи в [2] было предложено сочетать метод дихотомии с методом ветвей и границ. При этом подходе последовательно пересчитывается полуинтервал (a, b), заведомо содержащий искомый шаг расписания $\bar{z}$. Пересчет заключается в том, что выбирается какое-либо z из (a, b), вычисляются границы времен наступления циклического сетевого графика

$$v_{min}[M],\ v_{max}[M],$$

соблюдение которых необходимо и достаточно для построения расписания с шагом z, и далее расписание строится методом ветвей и границ с односторонним обходом дерева вариантов. Следующим полуинтервалом будет $(a, z]$, если расписание удалось построить, и $(z, b]$ - в противном случае. Для контроля существования допустимого решения используются границы времен наступления событий; время начала каждой работы должно удовлетворять следующим неравенствам, где i_u, j_u - начало и конец u:

$$v_{min}[i_u] \le w[u] \le v_{max}[j_u] - t[u].$$

Предлагаемый вариант метода ветвей и границ имеет следующую особенность. Невозможно определить время начала работы бригад на объекте, так как бригады переходят на следующий объект до окончания всей совокупности работ на предыдущем. Приходится начинать обход дерева вариантов, считая, что известен интервал времени, в который все бригады приступят к работе на новом объекте. Обозначим этот интервал для каждой бригады l через

$$dom[l].$$

При продолжении частичного решения для работы u определяется интервал $W[u]=[w_{min}[u], w_{max}[u]]$, в который должно начаться выполнение этой работы. Интервал $W[u]$ определяется так, чтобы он согласовывался с временами наступления событий циклического сетевого графика и с интервалами начала работ, входящих в частичное решение. Таким образом, каждое частичное решение представляет собой несколько частичных расписаний, в которых вектор $I[u]$ определяется однозначно, а $w[u]$ с точностью до интервала $W[u]$. По мере продолжения частичного решения интервалы $W[N]$ будут сужаться до тех пор, пока каждый не превратится в точку. В полном решении время начала каждой работы будет определено однозначно.

2. Метод ветвей и границ построения стационарного расписания

Основными шагами алгоритма являются продолжения частичного решения - назначение бригады на очередную работу и отмена назначения, если частичное решение стало недопустимым. Алгоритм вычисления и пересчета времен наступления событий циклического сетевого графика и интервалов $W[N]$ при продолжении частичного решения, а также алгоритм восстановления информации после отмены назначения не описаны в общей схеме метода в [2]. Изложим здесь эти алгоритмы, попутно уточнив некоторые начальные границы для интервалов.

Рассмотрим, как и в [2], случай с однотипными работами: $p=1,\ N_1=N,\ S_1=S$. Множество бригад обозначим через S.

Начнем с определения исходного интервала (a,b) для шага расписания z. Наименьший возможный шаг расписания z_{min} в [1] предложено находить решением следующей задачи линейного программирования: задан циклический сетевой график Γ и вектор продолжительностей работ $t[R]$. Зададим вектор $L[R]$ таким, что

$$L[H]=1[H],\ L[N\cup F]=0[N\cup F].$$

Требуется найти вектор $x[R]$, удовлетворяющий условиям

$$x[R]\geqslant 0[R], \tag{1}$$

$$\sum_{u\in R^+_{i_u}} x[u]=\sum_{u\in R^-_{i_u}} x[u], \tag{2}$$

$$L[R]\times x[R]=1 \tag{3}$$

и максимизирующий линейную форму

$$t[R] \times x[R]. \qquad (4)$$

Задача, двойственная задаче (I)-(4), - найти вектор $\vartheta[M]$ и число z, удовлетворяющие условиям

$$\vartheta[i_u] + t[u] - z \times L[u] \leq \vartheta[j_u],\ u \in R, \qquad (5)$$

и минимизирующие z. Эта задача называется задачей об оптимальном контуре, метод решения описан в [3]. Решением задачи является вектор $\vartheta[M]$, удовлетворяющий условиям (5), и число

$$z_{min} = 1[C] \times t[C] / 1[C] \times L[C],$$

где C - неразложимый контур, причем

$$\vartheta[i_u] + t[u] = \vartheta[j_u],\ u \in (N \cup F) \cap C,$$
$$\vartheta[i_u] - z_{min} = \vartheta[j_u],\ u \in H \cap C.$$

Расписание с шагом z_{min} можно составить, если число бригад s равно максимальному числу попарно-несравнимых дуг графа $\Gamma' = \langle M, N \cup F \rangle$. Дальнейшее увеличение числа исполнителей не приведет к сокращению сроков выполнения работ. Если число бригад будет меньше, то, возможно, искомый шаг расписания будет больше z_{min}. Вторая оценка снизу для шага расписания будет $\sum_{u \in N} t[u] / S$. Следовательно, можно определить границу интервала (a, b):

$$a = \min(z_{min} - 1, [\sum_{u \in N} t[u]/S).$$

Чтобы определить b, решим задачу составления нестационарного расписания для графа $\Gamma' = \langle M, N \cup F \rangle$, в котором отброшены дуги-переходы H. Полученная продолжительность критического пути будет оценкой сверху для шага расписания $\bar{z}$.

Будем называть ранние наступления событий сетевого графика нижними границами событий, поздние наступления - верхними границами. Вектор $\vartheta[M]$, полученный при решении задачи (I)-(4), дает нижние границы событий, которым должно удовлетворять расписание с шагом z_{min}. Верхние границы для вершин критического контура определяются сразу, они равны нижним границам. Верхние границы событий должны удовлетворять условиям (5), поэтому для каждой дуги $u \in H$ положим $\vartheta_{max}[i_u] = \vartheta_{min}[j_u] + z_{min}$ и вычислим верхние границы остальных событий в графе Γ', как в обычных сетевых графиках.

Выбираем некоторое z из интервала (a, b), определенного выше. Необходимо определить границы событий для расписания с шагом $\mathcal{Z}$. Нижние границы остаются без изменения при всех $\mathcal{Z} > \mathcal{Z}_{min}$. Для каждой дуги-перехода из H положим

$$\vartheta_{max}[i_u] = \vartheta_{min}[j_u] + \mathcal{Z}$$

и вычислим верхние границы остальных событий, как указано выше. Заметим, что при $\mathcal{Z} > \mathcal{Z}_{min}$ верхние границы событий $\vartheta_{max}[i]$ увеличатся на $\mathcal{Z} - \mathcal{Z}_{min}$ для всех $i \in C$, так как

$$\vartheta_{max}[i_u] = \vartheta_{min}[j_u] + \mathcal{Z},$$

а $\vartheta_{min}[i_u]$ не меняется. Верхние границы остальных событий из M зависят также от технологической последовательности работ и их продолжительностей, которые не меняются, следовательно, $\vartheta_{max}[i]$ увеличится на $\mathcal{Z} - \mathcal{Z}_{min}$ для всех $i \in M$. Суммарное время простоев бригад при выполнении проекта с шагом расписания $\mathcal{Z}$ равно $\tau s = \mathcal{Z} \times s - \sum_{u \in N} t[u]$.

Для данного объекта определим интервал времени начала работ для всех $\ell \in S$: $dom[\ell] = [0, t_c]$, где t_c - время перехода последней бригады с предыдущего объекта. Под временем $t = 0$ мы понимаем время перехода на новый объект первой бригады, освободившейся от работ на предыдущем объекте. Следовательно,

$$t_c = \max_{i \in M}(\vartheta_{max}[i]) - \mathcal{Z} = \max_{i \in M}(\bar{\vartheta}_{max}[i] +$$
$$+ \mathcal{Z} - \mathcal{Z}_{min}) - \mathcal{Z} = \max_{i \in M}(\bar{\vartheta}_{max}[i]) - \mathcal{Z}_{min},$$

где $\bar{\vartheta}_{max}[i]$ - верхние границы событий при шаге расписания $\mathcal{Z}_{min}$. Заметим, что величина t_c не меняется при изменении $\mathcal{Z}$. Таким образом, мы определили начальные значения параметров для организации метода ветвей и границ, причем при изменении они легко пересчитываются.

Рассмотрим повышение уровня перебора в методе ветвей и границ. На каждом шаге частичное решение σ характеризуется следующими параметрами:

для каждой бригады ℓ заданы: интервал $D[\ell] = [d_{min}[\ell]$, время $d_{max}[\ell]]$, в которое бригада закончила выполнение работ σ; интервал $dom[\ell]$, причем $D[\ell] = dom[\ell] + \sum_{u \in P(\ell)} t[u] + \sum_{u \in P(\ell)} \tau s[u]$, где $P(\ell)$ - множество работ, которые бригада ℓ должна выполнять;

для каждой вершины i заданы границы событий $v_{max}[i]$, $v_{min}[i]$;

для каждой работы из частичного решения известен интервал ее начала $W[u]=[w_{min}[u], w_{max}[u]]$ и время простоя бригады перед выполнением работы $u: rs[u]$; известна суммарная величина простоев бригады при выполнении работ $res_\sigma = \sum_{u\in\sigma} rs[u]$.

Чтобы восстанавливать информацию о частичном решении после отмены назначения, необходимо хранить уровень перебора k, на котором произошло сужение интервалов $dom[l]$, и значение интервалов до изменения. Хранение массива $dom[S]$ можно организовать по принципу магазина. После отмены назначения, имея массив $dom[S]$, можно восстановить интервалы $W[N]$ предыдущего шага с помощью процедуры корректировки интервалов, которая будет описана ниже.

ОПРЕДЕЛЕНИЕ. Частичное решение σ будем называть допустимым, если выполнены следующие условия для $u \in \sigma$:

1) $w_{min}[u] \geq v_{min}[i_u]$,
2) $w_{min}[u]+t[u] \leq v_{min}[j_u]$,
3) $w_{min}[u_1]+t[u_1]+rs[u_2]=w_{min}[u_2]$,

если работа u_2 выполняется непосредственно за u_1 одной бригадой;

4) $w_{max}[u] \geq v_{max}[i_u]$,
5) $w_{max}[u]+t[u] \leq v_{max}[j_u]$,
6) $w_{max}[u_1]+t[u_1]+rs[u_2]=w_{max}[u_2]$,
7) $w_{min}[u] \leq w_{max}[u]$,
8) $\sum_{u\in\sigma} rs[u] \leq res$.

Условия 1)–3) относятся к нижним границам интервалов, аналогичные условия 4)–6) – к верхним. Равенства 3) и 6) связывают работы, закрепленные за одной бригадой, остальные неравенства относятся к границам событий сетевого графика.

При каждом продолжении частичного решения границы интервалов $W[N]$ для работ из $u\in\sigma$ будут изменяться. В процедуре корректировки интервалов $W[N]$ выделим два шага.

ШАГ 1. Если уменьшилось значение $v_{max}[i_0]$, то $v_{max}[i]$ пересчитываются по путям, входящим в i_0 в графе $\langle M, NUF\rangle$ в соответствии с условием:

$$v_{max}[i_u]+t[u]\le v_{max}[j_u],$$
т.е. как в обычных сетевых графиках.

Аналогично пересчитывается $v_{min}[i]$ по путям, исходящим из i_o при увеличении $v_{min}[i_o]$.

Заметим, что при таком пересчете границы событий $v_{max}[M]$, $v_{min}[M]$ будут удовлетворять условию цикличности (5).

ШАГ 2. Если уменьшилось значение $w_{max}[u_o]$ для $u_o\in P(\ell)$, то $w_{max}[u]$ пересчитываются для всех работ $u\in P(\ell)$ по условию 6). Аналогично пересчитываются $w_{min}[u]$ при увеличении $w_{min}[u_o]$.

Рассмотрим назначение бригады на работу и переход от частичного решения σ к $\sigma'=\sigma\cup\{ut\}$. Для бригады $\ell_o\in S$ такой, что $d_{min}[\ell_o]=\min\limits_{\ell\in S}(d_{min}[\ell])$, определяется множество допустимых работ

$$\bar N=\{u\in N\,/\,u\notin\sigma,\ N^+_{iu}\subset\sigma,\ v_{min}[i_u]-d_{max}[\ell_o]\ge \tau es-\tau es_o\}.$$

Если множество $\bar N$ пусто или нашлась работа $up\in\bar N$ такая, что

$$d_{min}[\ell_o]>v_{max}[j_{up}]-t[up],$$

то перейти к понижению уровня перебора. Иначе, выбирается работа $ut\in\bar N$, на которой достигается минимум

$$v_{max}[j_u]-t[u],\ u\in\bar N.$$

Теперь необходимо определить интервал $W[ut]$ и скорректировать интервалы $W[u]$ для $u\in\sigma$. Сначала определяются верхние границы интервалов, затем нижние.

Положим

$$w_{max}[ut]=\min(v_{max}[j_{ut}]-t[ut],d_{max}[\ell_o]).$$

Если

$$w_{max}[ut]=d_{max}[\ell_o],\ w_{max}[ut]<v_{max}[i_{ut}],$$

то положить $v_{max}[i_{ut}]=w_{max}[ut]$ и выполнить шаг I. Если $w_{max}[ut]=v_{max}[j_{ut}]-t[ut]$, то выполнить шаг 2.

После шага I проверить неравенство 5) для $u\in\sigma$, если для $uk\in\sigma$ оно нарушается, то положить

$$w_{max}[uk]=v_{max}[j_{uk}]-t[uk]$$

и выполнить шаг 2.

После шага 2 проверить неравенство 4) для $u\in P(\ell)$; если для $uk \in P(\ell)$ неравенство нарушается, то положить $w_{max}[u] = v_{max}[i_u]$ и выполнить шаг I.

Вычисления продолжаются до тех пор, пока неравенства I)–6) не будут выполнены. Если $d_{max}[\ell_o]$ оказалось меньше $v_{max}[j_{ut}] - t[ut]$, то положим

$$w_{max}[ut] = v_{max}[i_{ut}] = v_{min}[t_{ut}],$$
$$rs[ut] = v_{min}[i_{ut}] - d_{max}[\ell_o].$$

После чего процедура повторяется. Затем определяется нижняя граница $W[ut]$. Положим.

$$w_{min}[ut] = max(v_{min}[i_{ut}], d_{min}[\ell_o]).$$

Далее пересчитываются нижние границы интервалов аналогично верхним. Частичное решение σ' считается построенным, когда все условия I)–8) выполнены. В теореме I доказывается, что допустимое частичное решение будет построено за конечное число шагов.

ЛЕММА I. Если частичное решение σ допустимо, то для любых работ u_1, $u_2 \in \sigma$ величина $|w_{min}[u_1] - w_{min}[u_2]|$ не меньше длины пути из i_{u_1} в i_{u_2} в графе Γ'.

ТЕОРЕМА I. Пусть σ – допустимое частичное решение, его продолжение $\sigma' = \sigma \cup \{u\}$ строится по алгоритму, описанному выше. Тогда допустимое частичное решение σ' будет получено за конечное число шагов.

Пересчет интервалов частичного решения σ наиболее трудоемкая часть алгоритма. Но если для всех бригад $\ell \in S$ интервал $dom[\ell]$ превратился в точку, то все неравенства выполнены и процедура корректировки интервалов не нужна. Оценкой сверху для итераций с таким пересчетом будет $t_c(S-2)$. Вычислительные эксперименты показывают, что достаточно быстро все интервалы превращаются в точку, и далее алгоритм решает обычную сетевую задачу распределения ресурсов.

3. Повышение эффективности алгоритма составления расписания

3.I. Для реализации метода ветвей и границ основное значение имеет стратегия ветвления и отбрасывание недопустимых вариантов. В нашей задаче приходится продолжать каждое частичное решение до тех пор, пока не будет построено полное решение или пока частичное решение не станет недопустимым, поскольку в общем случае частичные решения несравнимы. Чтобы убедиться в отсутствии расписания при данном $\mathcal{Z}$, необходимо проверить все частичные решения, поэтому особенно важно быстрое опознание недопустимых вариантов. Следующие леммы позволяют сократить перебор вариантов.

ОПРЕДЕЛЕНИЕ. Частичное решение недопустимо, если никакое его продолжение не укладывается в директивные сроки.

Частичное решение будет недопустимым, если:

I) существует $u \notin \sigma, u \in N$ такая, что

$$v_{max}[j_u] - t[u] < \min_{l \in S}(d_{min}[l]);$$

2) для всех работ $u \in N, u \notin \sigma$

$$v_{min}[i_u] > \min_{l \in S}(d_{max}[l]) + ws - ws_{\sigma}.$$

ЛЕММА 2. Пусть продолжение $\sigma' = \sigma \cup \{ut\}$ частичного решения σ недопустимо, так как для $uk \notin \sigma$ выполнено I). $\sigma'' = \sigma \cup \{uk\}$ недопустимо, так как для работы $ut \in \sigma$ выполнено условие I), причем величина $d_{min}[l_1] = \min_{l \in S, l \neq l_1}(d_{min}[l])$ не изменялась при переходе к частичным решениям и σ''; тогда σ – недопустимое частичное решение.

ЗАМЕЧАНИЕ. Если величина $d_{min}[l_1]$ изменялась при продолжении частичного решения σ , то может существовать его допустимое продолжение.

ЛЕММА 3. Пусть частичное решение $\epsilon' = \sigma \cup \{ut\}$ недопустимо, так как работа $uk \in \sigma$ не укладывается в директивные сроки. Тогда можно выделить подмножество работ из N ,

$u \in \sigma$, дающих недопустимые продолжения,

$N_0 = \{u \in N \mid u \in \sigma, \sigma \cup \{u\}$ недопустимы$\}$.

Построим множество N_0. Частичное решение $\sigma' = \sigma \cup \{ut\}$ недопустимо, следовательно, $\vartheta_{max}[j_{uk}] - t[uk] < \min_{l \in S} d'_{min}[l] = d'_{min}[l']$, где $d'_{min}[S]$ – раннее освобождение бригад после выполнения частичного решения σ'.

Пусть бригада l_1 такая, что

$$d'_{min}[l_1] = \min_{l \in S_0} (d'_{min}[l]),$$

$$S_0 = \{l \in S \mid d_{min}[l] \le \vartheta_{max}[j_{uk}] - t[uk]\},$$

$$\beta(uk) = \vartheta_{max}[j_{uk}] - t[uk].$$

Тогда обозначим

$$\alpha = \begin{cases} -1, & l_1 = l_0, \\ \beta(uk) + \vartheta_{min}[uk] - d'_{min}[l_1], & l_1 \ne l_0, \end{cases}$$

$$N_0 = \{u \in N \mid \vartheta_{min}[i_u] > \alpha, \max(d_{min}[l_0], \vartheta_{min}[i_u] + t[u] > \beta(uk)\}.$$

3.2. Если существует расписание при данном $\mathfrak{T}$, то удачная стратегия выбора работы для продолжения частичного решения позволяет быстро построить допустимое расписание. Рассмотрим несколько различных способов выбора работы.

1. Выбирается работа ut такая, что

$$\vartheta_{max}[j_{ut}] - t[ut] = \min_{u \in N_k} (\vartheta_{max}[j_u] - t[u]),$$

где N_k – множество допустимых работ на данном шаге. Такой метод следит в первую очередь за выполнением условия 1), тогда как условие 2) может нарушаться. Такой выбор не учитывает раннее начало работы $\vartheta_{min}[i_{ut}]$, поэтому бригада может простаивать, хотя в это время она может выполнять неприоритетную работу.

2. Выбирается работа $ut \in N_k$, на которой достигается минимум

$$\vartheta_{min}[i_{ut}] = \min_{u \in N_k} (\vartheta_{min}[i_u].$$

В этом случае экономно расходуется резерв простоев $\mathcal{W}$, но легко можно нарушить условие 1) и решение станет недопустимым.

3. В первую очередь выбирается работа с минимальным резервом времени $\vartheta_{max}[j_u] - t[u] - \vartheta_{min}[i_u]$. Это наименее

удачный вариант, так как раннее и позднее начала могут быть велики. Условия I) и 2) могут быстро нарушиться.

4. Выбирается работа ut, на которой достигается минимум суммы $v_{max}[j_u]-t[u]+v_{min}[i_u]$, $u \in N_k$. Тогда из работ с одинаковым ранним началом первой выбирается работа с минимальным поздним. Аналогично при одинаковых поздних началах выбираем ту работу, у которой наименьшее раннее начало. Таким образом, это правило позволяет следить за выполнением условий I) и 2).

Все четыре варианта были опробованы на ЭВМ. Допустимое расписание за наименьшее число итераций было построено при стратегии выбора 4. Стратегии 2 и 3 требовали значительно большего числа итераций.

3.3. Стационарное расписание, которое составляется по одному сетевому графику, позволяет планировать многократное выполнение работ по одному и тому же проекту. При построении расписания будем начинать с момента времени $t=0$. В это время можно начинать первые работы нового объекта, а также последние работы предыдущего объекта, раннее начало которых $v_{min}[i_u] \geq \mathcal{Z}$. Таким образом, мы составим расписание сначала для работ, которые должны начинаться в интервале времени $[\mathcal{Z}, \mathcal{Z}+t_c]$ и $[0, t_c]$. В силу цикличности это один и тот же интервал времени. Эта модификация будет особенно эффективной, если объем работ, которые надо выполнить в конце проекта, больше объема работ, выполняемого в начале проекта. Тогда основное сужение интервалов $W[N]$ произойдет на первых итерациях алгоритма. Трудоемкость данного алгоритма по сравнению с алгоритмом составления расписания для обычной сетевой задачи увеличивается за счет процедуры корректировки интервалов для работ частичного решения σ и зависит от числа работ частичного решения. Когда все интервалы превращаются в точку, дальнейший ход алгоритма не отличается от обычной задачи. Поэтому существенно, чтобы пересчет происходил на первых итерациях алгоритма.

3.4. Для хранения информации о дереве перебора вариантов введем векторы $g[N]$ и $gt[N]$, причем $g[u]=k$, если работа назначена на уровне перебора k, $g[u]=0$, если работа не назначена. Информацию о запрещенных работах будем хранить в массиве $gt[N]$. Используем тот факт, что работы для продолжения частичного решения выбираются по определенному правиду: выбирается ut такая, что

$$v_{max}[j_{ut}]-t[ut]=\min_{u\in N_k}(v_{max}[j_{ut}]-t[ut]),$$

где N_k - множество допустимых работ.

Если минимум достигается на нескольких работах, то выбирается работа с меньшим номером. Тогда, если на уровне перебора k нет запрещенных работ, то $gt[k]=0$, если $gt[k]=i$, то i - номер последней запрещенной работы. Легко переходить к альтернативе, при отмене частичного решения достаточно положить $gt[k]=j$, где j - номер работы uk. Если делаем шаг назад, то полагаем $gt[k]=0$.

4. Алгоритм составления расписания при разнотипных работах

До сих пор мы рассматривали задачу с однотипными работами, где $p=1, N_1=N, S_1=S$. Алгоритм, описанный выше, легко применяется для задачи с несколькими типами работ; необходимо только учитывать, что работы из N_i должны выполняться бригадами i-го типа. Определим для этой задачи начальные значения интервалов:

$$a=\max(z_{min}, \max_{i\in N_i}(\sum_{u\in N_i}t[u]/S_i)).$$

Для работ каждого типа определим минимальное раннее начало:

$$v_i=\min(t_c, \min_{u\in N_i}(v_{min}[i_u]))$$

и максимальное позднее окончание

$$vm_i=\max(z, \max_{u\in N_i}(v_{max}[j_u])).$$

Тогда начальный интервал для бригады l типа i будет

$$dom[l]=[v_i, vm_i-z].$$

Резерв простоев определяется для бригад каждого типа:

$$rs[i]=z\times S_i-\sum_{u\in N_i}t[u].$$

5. Результаты вычислительных экспериментов

Программа, реализующая данный алгоритм составления расписания, была написана на языке АЛГОЛ-60 для БЭСМ-6. Длина программ - 212 строк. Решались задачи с параметрами M =45, N = =110, z_{min} =107, δ=7. Наилучший результат (задача решалась при трех значениях z): z =111 - допустимое решение было построено за 111 итераций, z =109 - допустимое решение было построено за 389 итераций, z =108 - за 401 итерацию допустимого решения построить не удалось.

Время трансляции программы и счета - 183 сек. За это время было выполнено 900 итераций.

Отметим, что выбор работы при продолжении решения осуществлялся по правилу 4, учитывалась возможность сокращения перебора, следующая из леммы 2. Правило, следующее из леммы 3, не было реализовано; можно было назначать работы из конца сетевого графика в начало частичного решения.

В том же варианте программы, если работа выбирается по правилу 2, при z =112 за 401 итерацию не удалось построить допустимое решение, тогда как выбор по правилу 4 дает решение за 136 итераций при z =112.

ЛИТЕРАТУРА

1. РОМАНОВСКИЙ И.В. Циклические варианты моделей сетевого графика. - Исследование операций и статическое моделирование, 1972, вып. 1. Л.: изд-во ЛГУ, с. 145-152.
2. ГРИГОРЬЕВА Н.С., РОМАНОВСКИЙ И.В. Задачи составления циклических расписаний.-Кибернетика, 1978, № 1 . Киев, с.75-79.
3. РОМАНОВСКИЙ И.В. Алгоритмы решения экстремальных задач. - М.: Наука, 1977.

Поступила в ред.-изд. отдел
3.10.1980 г.