

Вестник

Астраханского
государственного
технического
университета

Серия

Управление,
вычислительная техника
и информатика

4 ²⁰¹⁴
октябрь



ВЕСТНИК

АСТРАХАНСКОГО
ГОСУДАРСТВЕННОГО
ТЕХНИЧЕСКОГО
УНИВЕРСИТЕТА

СЕРИЯ

УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА

НАУЧНЫЙ ЖУРНАЛ

ИЗДАЕТСЯ С АПРЕЛЯ 1993 ГОДА

ВЫХОДИТ ЧЕТЫРЕ РАЗА В ГОД

ИНДЕКСИРОВАНИЕ ЖУРНАЛА,
ВКЛЮЧЕНИЕ В БАЗЫ ДАННЫХ

База данных Ulrich's Periodicals Directory (США)

Каталог журналов открытого доступа – Directory of Open Access Journals, DOAJ (Швеция)

База Applied Science & Technology Source компании EBSCO Publishing (США)

CiteFactor – Каталог индексирования международных исследовательских журналов (Канада)

Общероссийский математический портал Math-Net.ru

Российский индекс научного цитирования – РИНЦ (Россия)

Реферативный журнал и база данных ВИНТИ РАН (Россия)

4 2014
октябрь

АСТРАХАНЬ
ИЗДАТЕЛЬСТВО АГТУ

СОДЕРЖАНИЕ

УПРАВЛЕНИЕ, МОДЕЛИРОВАНИЕ, АВТОМАТИЗАЦИЯ

<i>Головки С. В., Кононенко С. В., Романенко Н. Г.</i> Технологическая схема экспериментальной установки отслеживания положения судна на базе микропроцессорных устройств.....	7
<i>Обухов А. С., Шкодин В. С., Есауленко В. Н.</i> Устройство для измерения температуры в скважине в процессе бурения	17
<i>Корниенко И. Г., Чистякова Т. Б., Орданьян С. С., Рыбин Д. С., Полосин А. Н.</i> Программный комплекс для исследования процессов получения твёрдых сплавов.....	23
<i>Магомедов Ш. Г.</i> Преобразование представлений чисел в модулярной арифметике в системах остаточных классов с разными основаниями.....	32

КОМПЬЮТЕРНОЕ ОБЕСПЕЧЕНИЕ И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

<i>Брумиштейн Ю. М., Бондарев А. А., Кузьмина А. Б.</i> Системный анализ вопросов информационной безопасности использования ПЭВМ в домашних условиях.....	40
<i>Зубов М. В., Пустыгин А. Н., Старцев Е. В.</i> Численное моделирование анализа исходного кода с использованием промежуточных представлений	55
<i>Козлов А. Л., Селиванов Е. П.</i> Соотношения для альтернативного определения количества информации при различных законах распределения.....	67
<i>Шарафутдинова Е. Н., Авдеев Т. В., Бакаев М. А.</i> Проектирование и реализация интегрированной информационной системы для МЧС России	73

СИСТЕМЫ ТЕЛЕКОММУНИКАЦИЙ И СЕТЕВЫЕ ТЕХНОЛОГИИ

<i>Сорокин А. А., Дмитриев В. Н., Чан Куок Тоан, Резников П. С.</i> Оценка результатов использования протокола RIP в системах связи с динамической топологией сети методом имитационного моделирования	85
--	----

УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ И ЭКОНОМИЧЕСКИХ СИСТЕМАХ

<i>Демурич В. Б.</i> Метод принятия многокритериальных решений в информационной системе современного гостиничного предприятия	94
<i>Добрынин А. С., Кулаков С. М., Койнов Р. С., Грачёв А. В.</i> Формирование расписаний в задачах временного планирования	103
<i>Лантес В. В.</i> Структура управления учебной деятельностью в автоматизированных обучающих системах.....	112

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

<i>Дедов А. В., Попов Г. А.</i> Построение математических моделей предварительного диагноза заболеваний печени на основе методов регрессионного анализа	124
---	-----

ПРАВИЛА ДЛЯ АВТОРОВ ЖУРНАЛА

«ВЕСТНИК АСТРАХАНСКОГО ГОСУДАРСТВЕННОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА. СЕРИЯ: УПРАВЛЕНИЕ, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАТИКА».....	137
--	-----

А. С. Добрынин, С. М. Кулаков, Р. С. Койнов, А. В. Грачёв

ФОРМИРОВАНИЕ РАСПИСАНИЙ В ЗАДАЧАХ ВРЕМЕННОГО ПЛАНИРОВАНИЯ

Важнейшим условием составления расписаний в проектной деятельности является учет множества временных ограничений, которые привязаны к периодическим интервалам времени проведения работ. В классических постановках задач класса JSSP (Job Shop Scheduling Problem) не уделяется должное внимание периодическим интервалам времени графиков работ большинства организаций и предприятий. Постановка и решение задачи планирования, предлагаемые нами, предполагают введение минимального отрезка времени, что позволяет гибко описывать структуру периодических временных ограничений. Рассматривается алгоритм временного сдвига, используемый для получения точного расписания в условиях периодических временных интервалов. Решение задачи подразумевает два этапа: на первом осуществляется построение непрерывного расписания (диаграмма Ганта), на втором происходит его последующая корректировка с учетом периодических временных ограничений произвольной формы. На втором этапе используется описанный алгоритм сдвига, вычислительная сложность которого зависит от минимального интервала времени и его размерности (длительности проекта). Апробация алгоритма в рамках модельно-алгоритмического комплекса на графах показывает несущественный рост сложности вычислений при значительном увеличении объема исходных данных. Таким образом, данный алгоритм имеет перспективы применения в крупных промышленных системах составления расписаний. Рассматриваемая задача планирования работ актуальна для предприятий и организаций, участвующих в разработке и внедрении проектов, поставщиков ИТ-услуг и т. д. Представленные механизмы и методики могут использоваться при разработке подсистем планирования в различных отраслях науки и техники (производство, транспорт, проектно-процессная деятельность).

Ключевые слова: распределение работ, расписание, ИТ-сервис, планирование работ, временные ограничения, сервисное управление.

Введение

Составление календарных графиков проведения работ имеет большое прикладное значение в управлении деятельностью человеческих коллективов, организаций и предприятий. Нами рассматриваются механизмы решения на компьютере одной из задач класса JSSP (Job Shop Scheduling Problem) [1–3], которая заключается в распределении работ во времени при наличии временных ограничений. Временные ограничения формируются естественно – исходя из периодических графиков проведения работ, используемых при организации труда сотрудников. Рассматриваемая задача планирования работ актуальна для предприятий и организаций, участвующих в разработке и внедрении проектов, поставщиков ИТ-услуг и т. д. Представленные механизмы и методики могут использоваться при разработке подсистем планирования в различных отраслях науки и техники (производство, транспорт, проектно-процессная деятельность).

Формализация задачи

Рассматривается задача, в которой известны перечень работ, длительность и последовательность их выполнения. Базовая задача построения производственных расписаний [4] для непрерывного времени формализуется как задача на графах (сетях), в которой узлы представляют собой события, дуги – отдельные процессы или работы. Особый интерес представляет задача, в которой необходимо учитывать ограничения, связанные с невозможностью распределить работы в определенный интервал времени, что характерно для практических ситуаций на большинстве производств. Сложность заключается в вариативном характере таких ограничений, которые могут изменяться в различных постановках. Рассмотрим частную JSSP-постановку для общего случая, полагая, что на периодических интервалах времени (рабочая неделя, месяц) структура ограничений существенно не меняется.

Дано:

1. Вектор работ (процессов) $\overline{JB}$, для которых справедливы принципы:
 - а) композиции или агрегирования, когда отдельная работа (процесс) может включать в себя перечень более простых работ (процессов) $JB_i \equiv \{jb_{i1}, jb_{i2}, \dots, jb_{im}\}$; и наоборот, более мелкие работы или процессы могут укрупняться;
 - б) связности, когда для двух различных процессов JB_i и JB_j справедливо соотношение

$$\gamma_{JB_i, JB_j} = \begin{cases} 1, & \text{если } JB_i \rightarrow JB_j, \\ -1, & \text{если } JB_j \rightarrow JB_i, \\ 0, & \text{если } JB_i \text{ и } JB_j \text{ независимы.} \end{cases} \quad (1)$$

2. Орграф, характеризующий последовательность выполнения работ $jb_i \equiv e_i \in \overline{E}$, и его информационная модель $\overline{G} = (\overline{V}, \overline{E})$.

3. Нормативная, согласованная минимальная, средняя и максимальная длительность отдельных работ (процессов) $T_{\min}, T_{\max}, T_{\text{avg}}$.

4. Минимальный нормативный «оконный» интервал времени Δt (временной слот), характеризующий отрезок непрерывного времени $[t, t + \Delta t]$, кратный производственным графикам работ.

5. Временные ограничения, связанные с периодичностью выполнения работ трудовым коллективом, представленные в виде матрицы работ и простоев. В задачах построения производственных расписаний целесообразно использовать интерпретацию, когда известно, что производственные процессы четко привязаны к конкретным дням недели:

$$M[\text{DayOfWeek}[d], (t, t + \Delta t)] = \begin{cases} 1, & \text{допустимо размещение элемента работы,} \\ 0, & \text{простой, размещение не допускается.} \end{cases} \quad (2)$$

6. Критерий оптимальности. Минимизация совокупного времени выполнения всех процессов. Пусть ребро v орграфа $\overline{G}$ между смежными вершинами i, j связано с отрезком времени $\Delta t_{i,j}$, который характеризует продолжительность процесса (работы). Тогда, на основе выражения (1), справедливо выражение

$$\sum_{i=1}^N \sum_{j=1}^N \gamma_{JB_i, JB_j} \Delta t_{i,j} \rightarrow \min. \quad (3)$$

Требуется:

1. Построить расписание $\overline{R}$ работ $\overline{JB}$ во времени для непрерывного случая, т. е. фактически получить и визуализировать вектор кортежей $\overline{W}$, каждый элемент которого содержит связанную с работой временную информацию.

$$\overline{W} \equiv \overline{R} = \{JB_i \rightarrow [T_{\min}, T_{\text{avg}}, T_{\max}]\}.$$

2. Найти наиболее ранние и наиболее поздние моменты времени начала и завершения каждой работы.

$$\forall jb \in \overline{JB}; T_{\min}, T_{\max} = F(\overline{R}).$$

3. Определить критический путь в графе.
4. Построить расписание в условиях временных ограничений с использованием критерия (3).

Подходы к решению

Реализация полноценных инструментов решения представленной задачи на компьютере создает трудности, поскольку необходимы алгоритмы решения в общем виде, в том числе с использованием конструкторов графов (сетей).

Для решения задачи используется модельно-алгоритмический комплекс [5], написанный на языке программирования C# – WPF, который предусматривает три уровня взаимодействия моделей и алгоритмов.

1. Базовый уровень моделей, который включает в себя структуры данных графа (узел, дуга, граф и т. д.).

2. Уровень промежуточных алгоритмов, который предусматривает реализацию низкоуровневых алгоритмов, таких как быстрая сортировка узлов графа в зависимости от положения в топологии (алгоритм Форда – Фалкерсона).

3. Уровень алгоритмов, включая алгоритмы визуализации, поиска критического пути на графе, построения расписаний.

Все уровни модельно-алгоритмического комплекса связаны друг с другом «слабой» связью посредством интерфейсов, что позволяет при необходимости заменять отдельные компоненты решения другими.

Решение представленной задачи включает два основных этапа:

1. Построение расписания работ без учета временных ограничений (получение вектора кортежей работ в привязке к непрерывному времени и его визуализация).

2. Построение расписания, с учетом временных ограничений, представленных выражением (2) в постановке задачи.

Механизм решения задачи

Комплексное решение задачи на компьютере включает в себя следующие основные этапы.

1. *Топологическая сортировка узлов графа* (модификация алгоритма Форда – Фалкерсона) $\bar{G} = (\bar{V}, \bar{E})$, в результате которой вектор узлов сортируется в порядке, определяемом компаратором. Упорядочение вершин бесконтурного ориентированного графа осуществляется согласно частичному порядку, заданному ребрами орграфа на множестве его вершин.

2. *Определение начальной и конечной вершины графа* $\bar{G}$ как первого и последнего узла v_0, v_N вектора $\bar{V}$ отсортированных узлов графа.

3. *Определение «критического» пути в графе*. Критический путь определяется с помощью алгоритма, рассмотренного в [5].

4. *Определение раннего срока* $t_p(v_j)$ свершения события v_j как раннего срока, необходимого для выполнения всех работ, предшествующих данному событию v_j :

$$t_p(v_j) = \max(t_p(v_i) + t(v_i, v_j)); (v_i, v_j) \in U_{v_j}^+; (v_i, v_j) \in \bar{V},$$

где $t_p(v_j)$ – ранний срок свершения события i ; $t_p(v_i, v_j)$ – продолжительность (вес) работы соответственно между событиями v_i, v_j ; $U_{v_j}^+$ – множество работ (направленных дуг), входящих в событие v_j .

5. *Определение позднего срока* $t_n(v_i)$ свершения события v_i как наиболее позднего момента времени, после которого остается столько времени до критического срока, сколько необходимо для завершения всех работ, следующих за этим событием:

$$t_n(v_i) = \min(t_n(v_j) - t(v_i, v_j)); (v_i, v_j) \in U_{v_i}^-; (v_i, v_j) \in \bar{V}.$$

6. *Определение резерва времени события* v_i в соответствии с выражением

$$RT(v_i) = t_n(v_i) - t_p(v_i).$$

7. *Определение полного резерва времени* для работы – по сути, максимального количества времени, на которое можно задержать выполнение работы или увеличить ее продолжительность, не изменяя длительность критического срока.

$$RT_n(v_i, v_j) = t_n(v_j) - t_p(v_j) - t(v_i, v_j).$$

8. *Определение свободного резерва времени* для работы – максимального количества времени, на которое можно увеличить продолжительность данной работы, не изменяя при этом начальных сроков последующих работ при условии, что предшествующее событие наступило в свой срок.

$$RT_c(v_i, v_j) = t_p(v_j) - t_p(v_i) - t(v_i, v_j).$$

9. *Построение расписания в условиях временных ограничений (алгоритм сдвига)*. Сдвиг всех временных характеристик работ в условиях невозможности их размещения на интервалах времени $t + \Delta T$.

Ключевые идеи алгоритма временного сдвига

Процедура временного сдвига, используемая в алгоритме последовательного распределения работ, позволяет повысить точность построения расписаний при переходе от непрерывного к дискретному времени, когда определенные интервалы времени не могут быть использованы для планирования. Работа крупных технологических агрегатов и производств часто связана с наличием временных интервалов, которые желательны (или нежелательны) для проведения работ проектной деятельности. Например, развертывание программного обеспечения на предприятиях должно осуществляться в интервалы времени, свободные от производственной деятельности, которые формируются как $n \cdot \Delta t$, где n – количество минимальных отрезков времени. Рассматриваемый нами механизм временного сдвига позволяет учитывать производственные простои для более точного распределения работ в задачах планирования.

Основой для работы алгоритма временного сдвига является непрерывное расписание $\bar{R}$, которое может быть представлено вектором кортежей (набором записей) $\bar{W}$. Элемент набора записей (запись) w_i , полученный в ходе решения рассмотренной выше задачи, содержит поля, такие как идентификатор работы (*ID*), дата начала (*beginDate*), дата раннего окончания (*earlyEndDate*), дата позднего окончания (*lastEndDate*), компонент временного смещения (*offsetDate*).

$$w_i = \{ID_i, beginDate_i, earlyEndDate_i, lastEndDate_i, offsetDate_i\}.$$

С точки зрения процедуры составления расписаний отдельный кортеж (запись) представляет набор связанных данных по отношению к некоторому идентификатору работы, часть из которых используется алгоритмом построения расписаний.

Важнейшим элементом математической модели является также логическая матрица работ и простоев $timeMap[t \in Time, (t + \Delta t)]$, формируемая с учетом окна планирования, которая описывает временную сетку интервалов проведения работ такую, что

$$timeMap[t \in Time, (t + \Delta t)] = \begin{cases} 1, & \text{допустимо размещение на интервале } t + \Delta t, \\ 0, & \text{простой, размещение не допускается.} \end{cases}$$

Для случаев описания детальных временных компонент матрица работ и простоев может быть трансформирована в кортеж работ и простоев (преобразована из пространственной в табличную форму). В общем случае структура и вид матрицы или кортежа зависят от размерности времени, требуемой точности задания отрезков времени и динамики процессов. В задачах построения производственных расписаний целесообразно использовать «сжатую» интерпретацию, когда известно, что производственные процессы четко привязаны к конкретным дням недели:

$$M[\text{DayOfWeek}[t \in \text{Time}], (t + \Delta t)] = \begin{cases} 1, & \text{допустимо размещение элемента работы,} \\ 0, & \text{простой, размещение не допускается.} \end{cases}$$

Введем понятие левого и правого *временного сдвига*, которое будет означать единичное приращение минимальной компоненты кортежа в сторону уменьшения или увеличения времени. Таким образом, для кортежа $K[d \in \text{Day}, h \in \text{Hour}, m \in \text{Minute}]$ сдвигом будет кортеж $K[d, h, m \pm 1]$. Рассматриваемый нами алгоритм назначения работ (time – labeling) использует модель ограничений, представленных выражением (2).

Введем понятия «окрестности слева и справа» для работы w_i , которые означают все работы, входящие в начальное событие w_i (слева), и работы, выходящие из конечного события (справа).

Для упрощения понимания сути работы алгоритма в целом выделим несколько ключевых идей:

1. *Двухкомпонентный, двунаправленный временной итерационный механизм.* Итератор *workIterator* сдвигает временной кортеж в прямом направлении, итератор *durationIterator* сдвигает временной кортеж в обратном направлении (рис. 1).

2. *Механизм сдвига, с учетом смещения, непосредственно влияющий на окрестность работ $\overline{W}_N \in \overline{W}$, расположенных справа относительно текущей работы w_i .* Возможность поиска работ, расположенных в окрестности текущей w_i справа или слева, достигается за счет реализации в информационной модели дуги графа ссылок на стартовый и конечный узел.

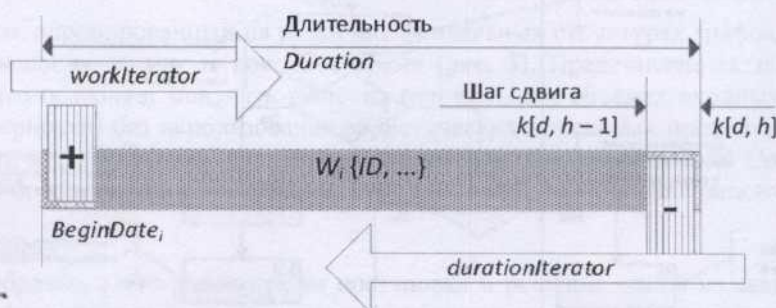


Рис. 1. Двунаправленная итерация по времени

3. *Механизм временной разметки (time-labeling) с использованием списка запретов.* Суть итерационного механизма заключается в следующем: если на очередном i -шаге итерации элемент кортежа $k_i[t \in \text{Time}, (t + \Delta t)]$ для работы w_i не может быть распределен, происходит сдвиг всех временных характеристик работ окрестности справа w_i , с учетом смещения для следующей работы, на интервал времени $k_i[t \in \text{Time}, (t + 2\Delta t)]$, если он отсутствует в списке запретов. В противном случае длительность текущей работы уменьшается на интервал времени $k_i[t \in \text{Time}, t]$, при этом сдвига временных характеристик работ окрестности справа w_i не происходит. Так как имеется n -работ окрестности слева для текущей работы w_i , итерирование каждой из которых приводит к сдвигам w_i , целесообразно использовать список запретов, каждый элемент которого представляет собой кортеж $k_{\text{taboo}}[id, k_i[t \in \text{Time}, (t + \Delta t)]]$. Список запретов создается отдельно для каждой работы w_i и содержит даты, которые уже использовались ранее для сдвига работы w_i .

Блок-схема алгоритма временного сдвига

Представленная в данном разделе блок-схема алгоритма реализована в рамках модельно-алгоритмического комплекса (МАК) построения расписаний [5] (рис. 2).

Большой вклад в теорию и практику решения задач составления расписаний внесли российские ученые В. С. Танаев, Ю. Н. Сотсков, В. А. Струевич, А. В. Куренков, Ю. С. Быков, Г. Б. Рубальский и др. В течение последних лет защищен ряд диссертаций, посвященных задачам разработки производственных и учебных расписаний, среди которых можно выделить работы молодых ученых С. В. Вережкина, К. С. Галузина, С. А. Костина, О. Н. Лопатеевой, Макаревой, Г. Ф. Назимовой и др. Следует также отметить большой вклад зарубежных специалистов в исследование данной проблематики. Мировую известность получили работы таких ученых, как Р. В. Конвей, В. Л. Максвелл, Л. В. Миллер; М. В. Картер, Е. Берк, А. Schaerf, De Werra и др.

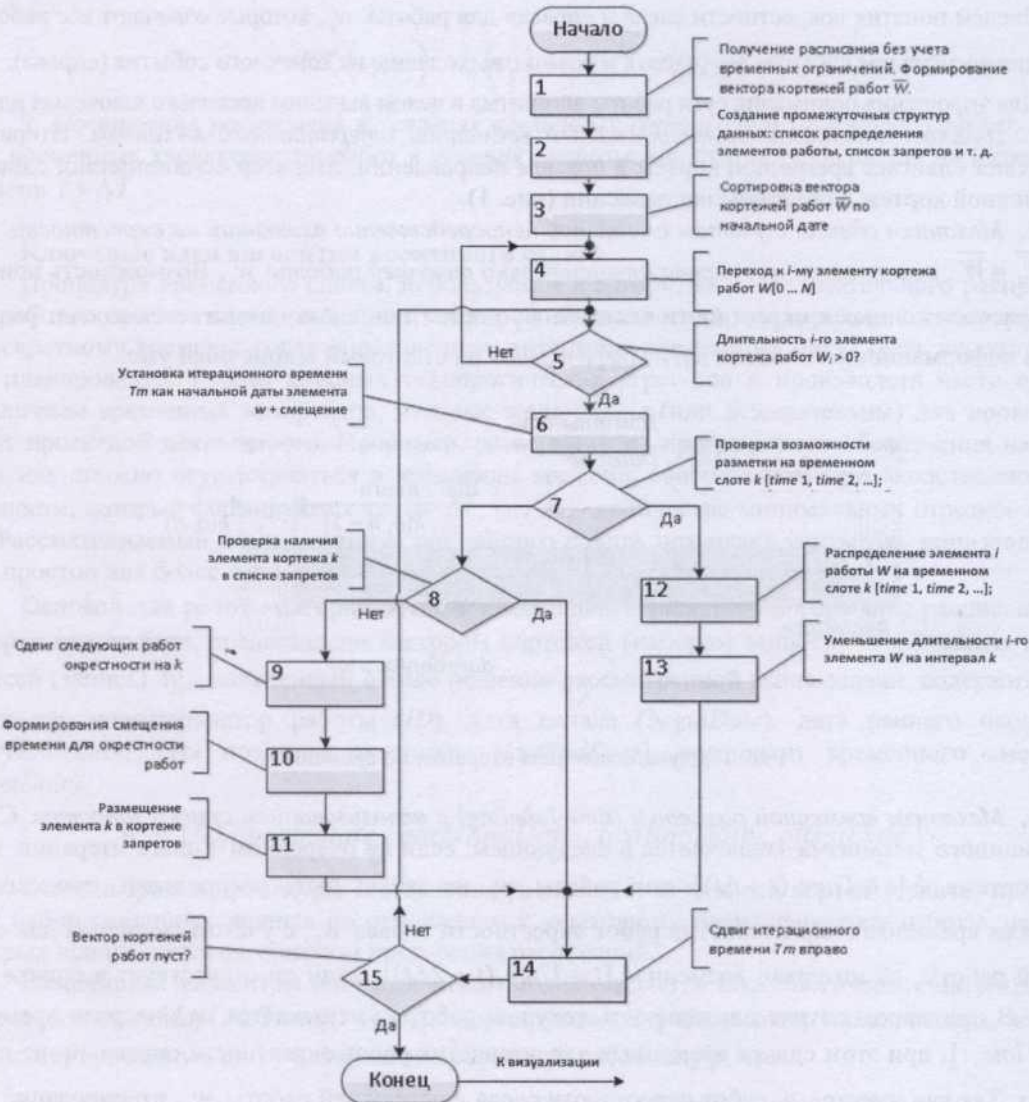


Рис. 2. Блок-схема алгоритма разметки (labeling) работ

Метод последовательного распределения для простейших задач предусматривает сдвиг всех работ, расположенных после текущей, что в значительной мере повышает его вычислительную сложность. Адаптация алгоритма, рассматриваемого нами, предполагает воздействие непосредственно на подмножество работ, следующих за текущей, что позволяет на практике понизить вычислительную сложность предлагаемого алгоритма.

Эвристические (метаэвристические) методы, такие как метод обратного хода, ветвей и границ, применяются, как правило, для решения NP-трудных задач, для которых не найдено быстрое и точное решение. В значительной степени большинство эвристических процедур опи-

рается на имитацию человеческого мышления, природные процессы и элементы теории вероятности (генетические алгоритмы [6], методы «муравьиной колонии» [7], имитация «отжига», поиск с запретами [8]). Поскольку большинство эвристических алгоритмов в процессе поиска использует элементы «нечеткой логики», вычислительная сложность их выше, чем у предлагаемого алгоритма (рис. 3).

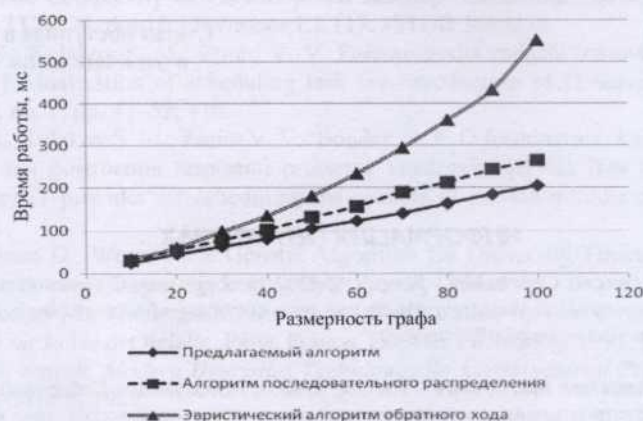


Рис. 3. Вычислительная сложность алгоритма

Алгоритм, апробированный на различных модельных структурах графов, позволяет получить точное решение задачи за конечное время (рис. 3). Представленная задача не является NP-трудной, что позволяет получать решения при больших объемах входных данных (графов большой размерности) без использования эвристических поисковых процедур. При разработке и тестировании использовалось следующее аппаратное обеспечение: Intel Core Quad 1,5 ГГц/4 Гб ОЗУ/160 Гб; программное обеспечение: MS Windows 7, MS Dot Net Framework 4.5, WPF C#.

Заключение

Таким образом, нами рассмотрены постановка и решение одной из задач планирования, которая является актуальной для организаций, осуществляющих проектную деятельность. Приведена структура временных ограничений, которая позволяет уточнять временные интервалы с учетом периодических графиков работ (неделя, месяц и т. д.), используемых в большинстве практических случаев. Предлагаемый подход позволяет решать задачу в общем виде для любых структур исходных данных с использованием конструктора графов. Алгоритм был протестирован; получены результаты для диапазона вершин графа от 10 до 100 при произвольных связях между ними.

Предложенный алгоритм показал лучшую производительность по сравнению с другими известными решениями, что открывает перспективы его использования в задачах планирования.

Развитие предлагаемых идей подразумевает построение крупных информационных систем управления проектами с возможностью ресурсно-ролевого управления активами на всех стадиях жизненного цикла систем.

СПИСОК ЛИТЕРАТУРЫ

1. *Job shop scheduling* // URL: http://en.wikipedia.org/wiki/Job_shop_scheduling (дата обращения: 03.05.2014).
2. Karger D. A Better Algorithm for an Ancient Scheduling Problem / D. Karger, S. Phillips, E. Torng // Proc. Fifth ACM Symp. Discrete Algorithms. 1994.
3. Garey M. R. The Complexity of Flowshop and Jobshop Scheduling / M. R. Garey // Mathematics of Operations Research. 1976. 1 (2). P. 117–129. doi:10.1287/moor.1.2.117. JSTOR 3689278.
4. Добрынин А. С. Формализация задачи составления расписаний для стадии внедрения ИТ-сервиса / А. С. Добрынин, С. М. Кулаков, В. В. Зимин // Научное обозрение: теория и практика. 2013. № 2. С. 47–52, 110.
5. Добрынин А. С. О формировании комплекса инструментальных средств ИТ-провайдера для построения расписаний процесса внедрения сервиса / А. С. Добрынин, С. М. Кулаков, В. В. Зимин, Н. Ф. Бондарь // Научное обозрение. 2013. № 8. С. 93–101.

6. *Burke E.* A Genetic Algorithm for University Timetabling / E. Burke, D. Elliman, R. Weare // Proceedings of the AISB workshop on Evolutionary Computing, University of Leeds, 1994.
7. *Colomi A.* Distributed Optimization by Ant Colonies / A. Colomi, M. Dorigo, V. Maniezzo // Actes de la première conférence européenne sur la vie artificielle. Paris, France, Elsevier Publishing, 1991. P. 134–142.
8. *Glover F.* Tabu search // Modern Heuristics Techniques for Combinatorial Problem. C. R. Reeves (ed.), Scientific Publications, Oxford, 1989.

Статья поступила в редакцию 16.05.2014,
в окончательном варианте – 25.06.2014

ИНФОРМАЦИЯ ОБ АВТОРАХ

Добрынин Алексей Сергеевич – Россия, 654007, Новокузнецк; Сибирский государственный индустриальный университет; соискатель кафедры «Автоматизация и информационные системы»; serpentfly@mail.ru.

Кулаков Станислав Матвеевич – Россия, 654007, Новокузнецк; Сибирский государственный индустриальный университет; д-р техн. наук, профессор; зав. кафедрой «Автоматизация и информационные системы»; serpentfly@mail.ru.

Койнов Роман Сергеевич – Россия, 654007, Новокузнецк; Сибирский государственный индустриальный университет; соискатель кафедры «Автоматизация и информационные системы»; koynov_rs@mail.ru.

Грачёв Александр Викторович – Россия, 654007, Новокузнецк; Сибирский государственный индустриальный университет; соискатель кафедры «Прикладная информатика и программирование»; ansel@zaoproxy.ru.



A. S. Dobrynin, S. M. Kulakov, R. S. Koynov, A. V. Grachev

SETTING UP OF SCHEDULES IN THE TIME PLANNING TASKS

Abstract. The most important condition in the scheduling of the project activity is keeping the set time limits, which are linked to periodic intervals of work time. Classic JSSP (Job Shop Scheduling Problem) tasking did not pay enough attention to periodic intervals schedules of most organizations and enterprises. Formulation and solution of the planning problem presented in this article involves the introduction of a minimum length of time that allows the flexibility to describe the structure of periodic time constraints. The article deals with the time shift algorithm used to obtain a precise schedule in terms of periodic time intervals. Solution of this problem involves two steps: the first step is to set up a continuous schedule (Gantt chart), the second comprises its subsequent periodic adjustment to the time constraints of an arbitrary shape. The second step is used to solve the problem of shift algorithm, the computational complexity of which depends on the minimum amount of time and its dimension (duration of the project). Algorithm testing within the model-algorithmic complex on the graphs shows insignificant increase in computational complexity with a significant increase in the volume of initial data. Thus, the algorithm has a data application prospects in large-scale industrial systems of scheduling. The considered problem of scheduling is urgent for companies and organizations involved in developing and implementing the projects, IT service providers, etc. The submitted mechanisms and techniques can be used in the development of planning subsystems in various branches of science and technology (production, transportation, design-process activities).

Key words: labeling, schedule, IT-service, job shop schedule, time constraints, service management.

REFERENCES

1. *Job shop scheduling*. Available at: http://en.wikipedia.org/wiki/Job_shop_scheduling (accessed: 03.05.2014).
2. Karger D., Phillips S., Torng E. A Better Algorithm for an Ancient Scheduling Problem. *Proc. Fifth ACM Symp. Discrete Algorithms*, 1994.
3. Garey M. R. The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, 1976, 1 (2), pp. 117–129. doi:10.1287/moor.1.2.117. JSTOR 3689278.
4. Dobrynin A. S., Kulakov S. M., Zimin V. V. Formalizatsiia zadachi sostavleniia raspisanii dlia stadii vnedreniia IT-servisa [Formalization of scheduling task for introduction of IT-service]. *Nauchnoe obozrenie: teoriia i praktika*, 2013, no. 2, pp. 47–52, 110.
5. Dobrynin A. S., Kulakov S. M., Zimin V. V., Bondar' N. F. O formirovanii kompleksa instrumental'nykh sredstv IT-provaidera dlia postroeniia raspisanii protsessa vnedreniia servisa [On formation of the complex of technological tools of IT-provider for scheduling the process of service introduction]. *Nauchnoe obozrenie*, 2013, no. 8, pp. 93–101.
6. Burke E., Elliman D., Weare R. A Genetic Algorithm for University Timetabling. *Proceedings of the AISB workshop on Evolutionary Computing*, University of Leeds, 1994.
7. Colomi A., Dorigo M., Maniezzo V. Distributed Optimization by Ant Colonies. *Actes de la première conférence européenne sur la vie artificielle*, Paris, France, Elsevier Publishing, 1991, pp. 134–142.
8. Glover F. *Tabu search. Modern Heuristics Techniques for Combinatorial Problem*. C. R. Reeves (ed.), Scientific Publications, Oxford, 1989.

The article submitted to the editors 16.05.2014,
in the final version – 25.06.2014

INFORMATION ABOUT THE AUTHORS

Dobrynin Alexey Sergeevich – Russia, 654007, Novokuznetsk; Siberian State Industrial University; Candidate of the Department "Automation and Information Systems"; serpent-fly@mail.ru.

Kulakov Stanislav Matveevich – Russia, 654007, Novokuznetsk; Siberian State Industrial University; Doctor of Technical Sciences, Professor; Head of the Department "Automation and Information Systems"; serpentfly@mail.ru.

Koynov Roman Sergeevich – Russia, 654007, Novokuznetsk; Siberian State Industrial University; Candidate of the Department "Automation and Information Systems"; koynov_rs@mail.ru.

Grachev Aleksander Viktorovich – Russia, 654007, Novokuznetsk; Siberian State Industrial University; Candidate of the Department "Applied Informatics and Programming"; anse1@zaoproxy.ru.

