

- ский технологический институт. – Томск: Изд-во Томского политехнического университета, 2011. – 231с.
5. Grigoryeva A. A. Information System Of Innovative Products Competitiveness Determining // 7th International Forum on Strategic Technology (IFOST - 2012): Proceedings: in 2 vol., Tomsk, September 18-21, 2012. - Tomsk: TPU Press, 2012 - Vol. 1 - p. 691-694.
  6. Grigoryeva A.A., Grigoreva A.P. System of decision –making support about competitiveness innovations in engineering industry Труды 6 всемирной конференции «Intelligent System for Industrial Automation» Tashkent, Uzbekistan November 25-27, 2010.-С. 116-119.
  7. Захарова А. А. , Григорьева А. А. Нечеткие модели принятия решений об инновационном развитии региона. Стратегический анализ, выбор и контроль. Монография - Saarbrucken : LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012 - 245 с.
  8. Осипов Ю.М. Конкурентоспособность наукоемкой машиностроительной продукции: экономика и менеджмент: моногр. – Томск: Изд-во Томск. гос. ун-та сист. упр. и радиоэлектроники, 2002. – 247 с.
  9. Saaty T.L. Measuring the fuzziness of sets. – Journal of Cybernetics, 1974. v. 4, p. 53-61.
  10. Zadeh L.A. Calculus of fuzzy restrictions. In: Fuzzy Sets and Their Applications to Cognitive and Decision Processes/Ed. By L. A. Zadeh et al. New York: Academic Press, 1975, p. 1-41.

## ОБ ОДНОМ АЛГОРИТМЕ ПОСТРОЕНИЯ РАСПИСАНИЙ В УСЛОВИЯХ ВРЕМЕННЫХ ОГРАНИЧЕНИЙ

*А. Добрынин, Р. Койнов, Е. Аксенова*

*Национальный исследовательский Томский политехнический университет, Филиал в г. Новокузнецке  
654007, Kemerovo Region, Novokuznetsk, Druzhba Av., 39, (3843) 776786, fax: (3843) 46449*

*E-mail: sgd\_nftpu@mail.ru*

### Элементы математической модели

Базовая задача построения производственных расписаний [1], для непрерывного времени, формализуется как задача на графах, в которой узлы представляют собой события, дуги – отдельные процессы или работы. С каждой дугой ассоциирован двухкомпонентный вес, представленный вещественным числом и временной разницей с возможностью их взаимного отождествления. Этапы решения базовой задачи [1], реализованы в рамках модельно – алгоритмического комплекса (МАК) [2] и дают неплохие результаты на практике.

Особый интерес представляет задача, в которой необходимо учитывать ограничения, связанные с невозможностью распределить работы в определенный интервал времени. Сложность заключается в вариативном характере таких ограничений, которые могут изменяться в различных постановках. Рассмотрим элементы математической модели для достаточно общего случая, предполагая, что на периодических интервалах времени  $t + \Delta t$  структура ограничений одинакова.

Одним из элементов математической модели, используемой для построения расписаний в ограничениях, является **вектор кортежей работ**  $\overline{W}$ , полученный в ходе решения задачи [1], где каждая отдельная запись представляет собой параметры отдельной работы, такие как: идентификатор работы (ID), дата начала (beginDate), дата раннего окончания (earlyEndDate), дата позднего окончания (lastEndDate), компонент временного смещения (offsetDate).

$$w_i = \{ID_i, beginDate_i, earlyEndDate_i, lastEndDate_i, offsetDate_i\} \quad (1)$$

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.

Также важнейшим элементом математической модели является логическая **матрица работ и простоев**  $timeMap[d \in Days, h \in Hours]$ , которая описывает временную сетку интервалов проведения работ, такую что:

$$timeMap[d, h] = \begin{cases} 1, & \text{допустимо размещение элемента работы} \\ 0, & \text{простой, размещение не допускается} \end{cases} \quad (2)$$

Для случаев описания детальных временных компонент, матрица работ и простоев может быть трансформирована в **кортеж работ и простоев** (использование более двух временных компо-

нент). В общем случае, структура и вид матрицы или кортежа зависит 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, см. выражение (3).

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

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  $K[d \in Day, h \in Hour, m \in Minute]$  сдвигом будет кортеж  $K[d, h, m + / - 1]$ . Рассматриваемый в статье алгоритм назначения работ (time – labeling) использует модель ограничений, представленных выражением 2.

#### Ключевые идеи алгоритма

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

*Двухкомпонентный, двунаправленный временной итерационный механизм.*

Итератор `workIterator` сдвигает временной кортеж в прямом направлении, итератор `durationIterator` сдвигает временной кортеж в обратном направлении, см. рисунок 1.

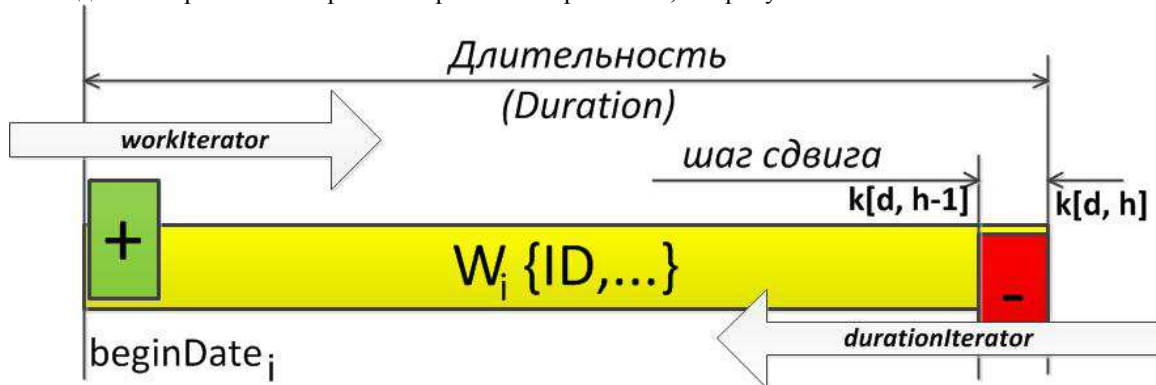


Рис. 1. Двунаправленная итерация по времени

- 1) Механизм сдвига, с учетом смещения, непосредственно влияющий на окрестность работ  $\bar{W}_N \in \bar{W}$ , 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, см. Листинг 1.

```

/*Интерфейс для дуги графа IEdge<T>*/
public interface IEdge<T> : IComparable<IEdge<T>>
{
    ///Временной интервал дуги (выраженный через временную разницу)
    TimeSpan Duration { get; set; }
    //// Стартовый узел для дуги, как INode<T>
    INode<T> start_node { get; }
    //// Конечный узел дуги, как INode<T>
    INode<T> end_node { get; }

    string start_nodeid { get; } ///Стартовый идентификатор узла
    string end_nodeid { get; } ///Конечный идентификатор узла
    string name { get; set; } ///Наименование дуги
    string manager { get; set; }
    double Weigth { get; set; } ///Вес дуги
}
    
```

Листинг 1. Информационная модель дуги графа

Информационная модель дуги графа содержит ссылки на стартовый и конечный узел графа, реализующие поведенческий механизм  $INode < T >$ , также имеется возможность работы по уникальным строковым идентификаторам узлов. Таким образом, итерационный процесс по отдельной дуге графа воздействует на окрестность дуг, расположенных после текущей дуги, см. рисунок 2.

2) Оригинальный механизм временной разметки (time-labeling) с использованием списка запретов.

Суть итерационного механизма заключается в следующем: если на очередном  $i$ - шаге итерации элемент кортежа  $k_i[d, h]$  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  $k_i[d, h + 1]$ , если он отсутствует в списке запретов. В противном случае длительность текущей работы уменьшается на интервал времени  $k_i[d, h - 1]$ , 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 *tabooList*, каждый элемент которого представляет собой кортеж  $k_{taboo}[id, k_i[d, h]]$ . Список запретов создается отдельно для каждой работы  $w_i$  и содержит даты, которые уже использовались ранее для сдвига работы  $w_i$ .

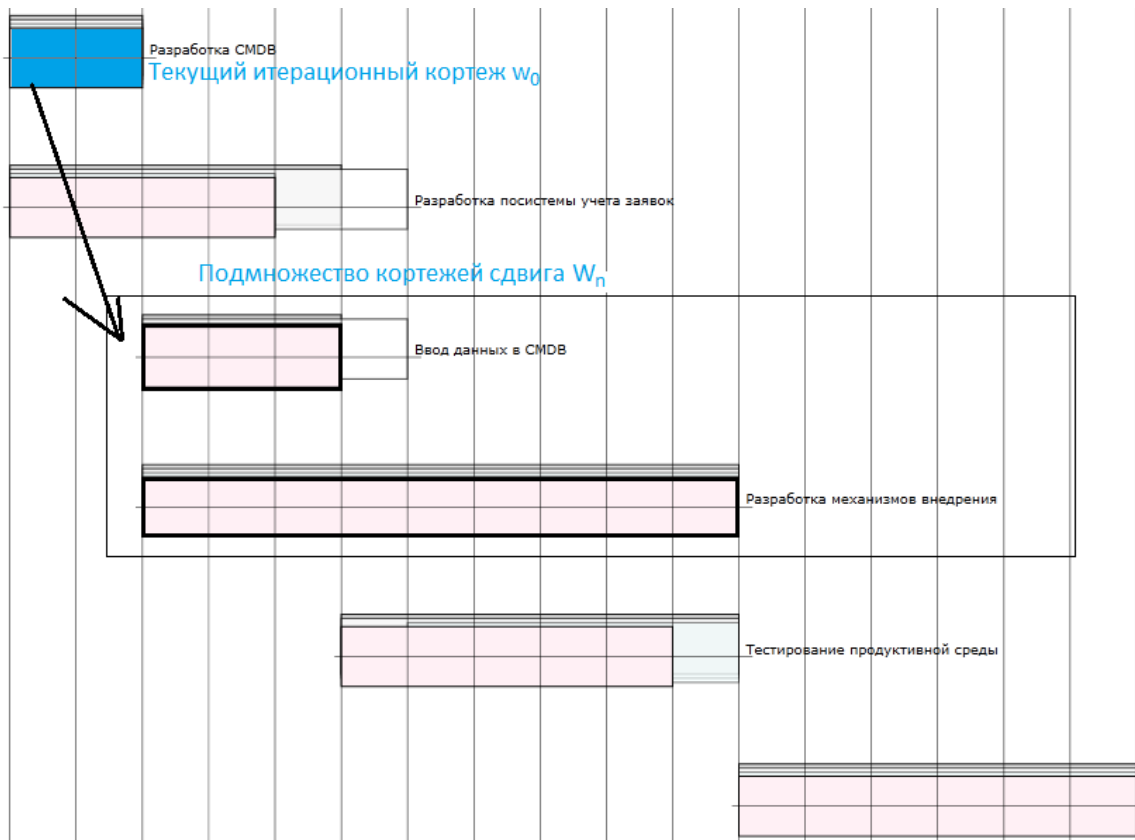


Рис. 2. Подмножества сдвига при итерационном движении

**Содержательное описание алгоритма**

Опираясь на описанные выше идеи, сформулируем алгоритм построения расписаний, пригодный для построения расписаний для сколь угодно сложных практических случаев временных ограничений, при условии их однородности.

1. Сортировка вектора  $w_i$  по возрастанию даты начала работы  $beginDate$ .
2. Определение даты начала проекта  $prjDate$  как  $w_0\{..., beginDate, \dots\}$
3. Двухнаправленная итерационная процедура по каждой работе  $w_i \in \overline{W}$ ,  $i = 0 \dots N - 1$ , выполняем действия по формированию кортежей ее размещения во времени. Формирование вектора кортежей  $\overline{LBL}$ , каждый элемент которого содержит идентификатор работы и дату начала разметки для временного сдвига.
4. Визуализация вектора кортежей  $\overline{LBL}$  с использованием механизма рендеринга WPF.

### Блок схема алгоритма

Представленная в данном разделе блок – схема алгоритма построена с опорой на процесс отладки работающей реализации на языке программирования C# в рамках модельно-алгоритмического комплекса (МАК) построения расписаний [2]. Алгоритм был опробован на 10 тестовых структурах графов, при произвольной генерации значений для матрицы временных ограничений. Блок схема представлена на рисунке 3.

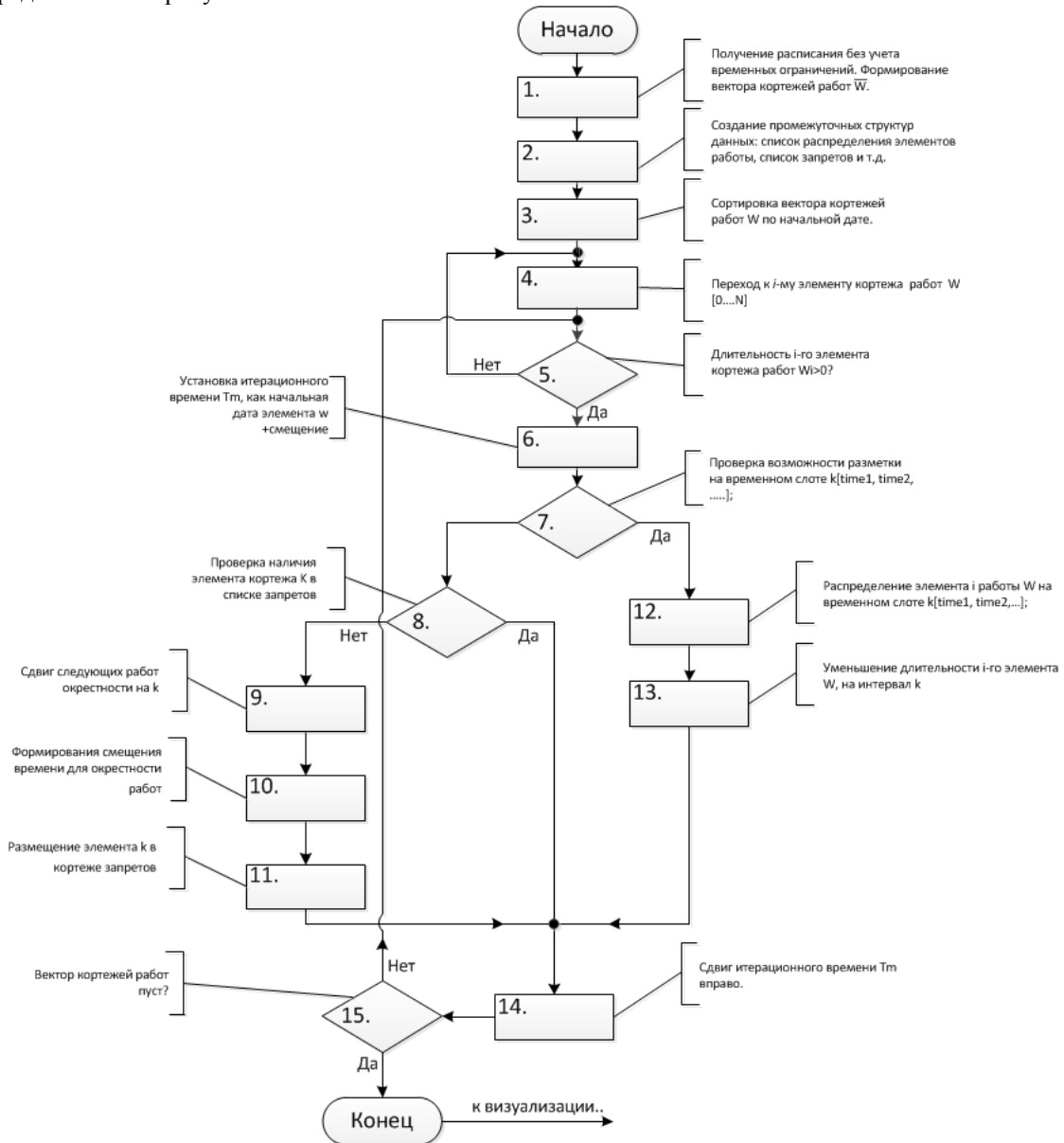


Рис. 3. Блок – схема алгоритма разметки (labeling) работ

### Вопросы реализации

Рассмотренный в работе алгоритм реализован в составе модельно – алгоритмического комплекса МАК[2], опробован на множестве модельных структур графов работ (более 10), при производных способах задания кортежей временных ограничений.

Литература.

1. Добрынин А.С., Кулаков С.М., Зимин В.В. Формализация задачи составления расписаний для стадии внедрения ИТ-сервиса // Научное обозрение: теория и практика. – 2013. -№2. –С. 47-52, 110.
2. 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.
3. Р.С. Мартин, М. Мартин. Принципы, паттерны и методики быстрой разработки приложений на языке программирования C#. -М.:Символ-Плюс, 2013.-786 с.
4. OGC-ITIL V3-2 Service Transition, TSO. – 2007. с.

### О ПОСТРОЕНИИ АДАПТИВНОЙ СИСТЕМЫ УПРАВЛЕНИЯ НАГРЕВАТЕЛЬНЫМИ ОБЪЕКТАМИ НА ОСНОВЕ НЕЙРОСЕТЕВОЙ ОПТИМИЗАЦИИ ПАРАМЕТРОВ ПИД-РЕГУЛЯТОРА

*Ю.И. Еременко, д.т.н., проф., Д.А. Полеценко, к.т.н., доц., А.И. Глуценко, к.т.н.  
Старооскольский технологический институт им. А.А. Угарова (филиал) НИТУ «МИСиС»  
309516, Белгородская обл., г. Старый Оскол, м-н Макаренко, д.42, тел. 8(4725) 45-12-00  
E-mail: strondutt@mail.ru*

Задача совершенствования работы систем управления технологическими процессами на сегодняшний день является весьма актуальной. Это связано с тем, что в большинстве случаев на предприятиях управление нелинейными по своей природе объектами, параметры которых имеют тенденцию изменяться по заранее неизвестному закону, ведется с помощью линейных ПИД- и ПИ-регуляторов. Подобное несоответствие зачастую приводит к снижению качества регулирования и к росту энергозатрат на ведение технологических процессов, поскольку конкретные значения коэффициентов подобных линейных регуляторов позволяют оптимально вести технологический процесс только для конкретного состояния объекта. Однако при его переходе в другие состояния (например, смена задания по температуре, загрузка печи) данные значения коэффициентов уже не позволяют получать требуемых по качеству переходных процессов. Подобная проблема особенно актуальна для отраслей, в которых широко используются различные нагревательных объекты - печи нагрева литой заготовки, дуговые печи и пр. - для которых принудительное охлаждение невозможно, ввиду чего наличие даже небольшого перерегулирования требует длительного времени на охлаждение [1].

Фактически, для каждого состояния объекта управления необходимо подбирать свои значения коэффициентов регулятора. Однако, в реалиях производства чаще всего используют всего один набор таких коэффициентов для всех режимов и не производят их перенастройку ввиду трудоемкости данного процесса.

Одним из возможных решений данной проблемы является построение адаптивных систем управления технологическими процессами [2], автоматически подбирающих коэффициенты ПИД-регулятора. Всю совокупность методов построения таких систем можно разбить на две группы. Первая группа - это классические методы, такие как метод Циглера-Никольса [3], частотный метод А.Г. Александрова [4], метод, заложенный в основу работы адаптивных ПИД-регуляторов фирмы Siemens [1]. Вторая группа - это методы, основанные на методологии искусственного интеллекта [5-7].

Для решения задач управления среди интеллектуальных методов исследователи выделяют, в частности, нейронные сети (НС), поскольку НС обладают нелинейными свойствами и способностью к обучению, что придает адаптивные свойства нейросетевым системам управления. Сигеру Омату в [8] предложил несколько схем нейроуправления. Анализ ряда работ, в частности [5-7], позволяет сделать вывод о том, что среди таких схем наиболее реализуемой с практической точки зрения является схема управления на основе автонастройки коэффициентов ПИД-регулятора  $K_p$ ,  $K_i$ ,  $K_d$  с помощью НС (рис.1, где  $e$  – рассогласование).