

УДК 658.52.011.56

КОНВЕЙЕРНАЯ ЗАДАЧА ТЕОРИИ РАСПИСАНИЙ С ОБРАТНЫМ УПОРЯДОЧЕНИЕМ РАБОТ ПО ТЕХНИЧЕСКОМУ ОБСЛУЖИВАНИЮ АВТОМОБИЛЕЙ

**С.А. Канцедал, профессор, д.т.н., Западодонбасский частный институт
экономики и управления, г. Павлоград, М.В. Костикова, доцент, к.т.н.,
И.В. Скрипина, ст. преподаватель, ХНАДУ**

Аннотация. При планировании выполнения технического обслуживания автомобилей предлагается новый алгоритм, использующий принцип обратного упорядочения работ при составлении расписания. Приводится теоретическое обоснование возможности построения алгоритма, а также его сравнительные характеристики, полученные экспериментальным путем. Предлагается способ улучшения временных характеристик алгоритма.

Ключевые слова: автомобили, техническое обслуживание, теория расписаний, конвейерная задача, алгоритм.

КОНВЕЄРНА ЗАДАЧА ТЕОРІЇ РОЗКЛАДІВ ЗІ ЗВОРОТНИМ УПОРЯДКУВАННЯМ РОБІТ ІЗ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ АВТОМОБІЛІВ

**С.А. Канцедал, профессор, д.т.н., Західнодонбаський приватний інститут економіки
і управління, м. Павлоград, М.В. Костікова, доцент, к.т.н.,
І.В. Скрипіна, ст. викладач, ХНАДУ**

Анотація. При плануванні виконання технічного обслуговування автомобілів пропонується новий алгоритм, який використовує принцип зворотного впорядкування робіт при складанні розкладу. Подано теоретичне обґрунтування можливості побудови алгоритму, а також його порівняльні характеристики, отримані експериментальним шляхом. Пропонується спосіб покращення часових характеристик алгоритму.

Ключові слова: автомобілі, технічне обслуговування, теорія розкладів, конвеєрна задача, алгоритм.

GENERAL PROBLEM OF SCHEDULING THEORY WITH REVERSE ORDERING OF WORKS CONCERNING VEHICLES MAINTENANCE

**S. Kantsedal, Professor, Doctor of Technical Science, West Donbass Private Institute
of Economics and Management, Pavlograd, M. Kostikova, Associate Professor,
Candidate of Technical Science, I. Skrypina, senior lecturer, KhNAHU**

Abstract. A new algorithm, using the principle of reverse ordering of works at drawing up of schedules is offered for application at planning of vehicles maintenance carrying out. Theoretical grounding of algorithm compiling possibility as well as its comparative features obtained experimentally are presented. A method of temporary algorithm features improvement is offered.

Key words: vehicles, maintenance, scheduling theory, conveyor problem, algorithm.

Введение

Рыночная экономика требует минимизации себестоимости любой продукции, чтобы выигрывать соревнования по ценам у конкурентов. У всех предприятий заметную долю средств производства составляет колесная и гусеничная техника, поэтому важным направлением снижения себестоимости является сокращение времени простоя машин в ремонте. Только у крупных предприятий это может быть обеспечено деятельностью собственных хорошо оснащенных ремонтных баз. Для остальных содержание ремонтников, соответствующих помещений и оборудования является тяжким бременем. Все больше владельцев техники понимают невыгодность содержания ремонтных цехов. В свою очередь эффективное функционирование крупных предприятий возможно только при правильной организации производства. В качестве критерия эффективности принято количество автомобилей, отремонтированных за определенный период. Для увеличения количества автомобилей необходимо четко спланировать очередность выполнения работ по техническому обслуживанию и ремонту.

Анализ публикаций

Составление графика очередности выполнения работ по техническому обслуживанию и ремонту автомобилей является задачей теории расписаний и в общем виде еще не решена [1]. Имеется целый ряд частных результатов. Особые трудности при решении задач возникают в случае больших размерностей.

В статьях [2, 3] описаны алгоритмы составления расписаний для производственных систем с общим и конвейерным характером выполнения работ. Отличительной чертой предложенных алгоритмов является прямое упорядочение работ, т.е. работы (операции) предметов при формировании порядков их выполнения на машинах назначаются в возрастающем порядке следования этих операций в технологических маршрутах предметов. Например, для двух предметов a , b с технологическими операциями a_2 , a_1 и b_1 , b_2 , представленными сетью G_T с истоком i_S и стоком i_F , при любых вариантах упорядочения работ на машинах 1 и 2 алгоритмами

предусматривается, что работа a_2 всегда будет предшествовать работе a_1 , а работа b_1 – предшествовать работе b_2 .

В настоящей статье в целях поиска более качественных алгоритмов составления расписаний обосновываются и исследуются алгоритмы, в которых выбор операций предметов для составления порядков их выполнения на машинах осуществляется начиная с последних операций предметов. То есть в рассматриваемом примере для предметов a и b сначала будут выбраны операции a_1 , b_2 , затем, как предшествующие им операции, a_2 , b_1 . Для описания алгоритмов и их обоснования сформулируем задачи в традиционной постановке [2, 3].

Для достижения указанной цели решались следующие задачи: а) конструирование алгоритмов для общей и конвейерной задач теории расписаний с обратным упорядочением работ; б) программная реализация разработанных алгоритмов; в) проведение экспериментальных исследований алгоритмов.

Цель и постановка задачи

Заданы конечные множества автомобилей $I = \{i | 1 \leq i \leq n\}$ и обслуживающих их постов $Q = \{q | 1 \leq q \leq m\}$. Обслуживание каждого автомобиля $i \in I$ состоит в выполнении постами из множества Q последовательности операций $I_i = \{iq_j | 1 \leq j \leq j_i\}$, специфичной для каждого автомобиля в общей задаче и одинаковой для всех автомобилей в задаче конвейерного типа. Операция $iq_j \in I_i$ выполняется постом $q \in Q$ за время $t_{iq_j} > 0$ без прерывания и может начаться после полного завершения непосредственно предшествующей ей операции $iq_{j-1} \in I_i$. Любой пост $q \in Q$ одновременно может выполнять только одну операцию.

Задача составления расписания состоит в том, чтобы для каждого поста $q \in Q$ найти такой порядок выполнения операций и, как следствие, определить такие моменты их начала, которые бы обеспечивали обслуживание множества всех автомобилей I за минимальное время.

С целью обоснования алгоритмов с обратным упорядочением работ перейдем к описанию задач на языке орграфов, подробно изложенному в [2]. Согласно [2] сеть, рассматриваемая в данном примере, определяет исходные данные задачи и в общем виде может быть представлена как $G_T = (I_I, i_S, V_T, W_T)$, где I_I – множество вершин графа; $\{a_2, a_1, b_1, b_2\}$ – множество технологических операций; i_S, i_F – фиктивные вершины (исток и сток); $V_T = V_I \cup V_S \cup V_F$ – множество дуг, подмножество V_T которых $\{(a_2, a_1), (b_1, b_2)\}$ определяет направления следования технологических операций; $V_S = \{(i_S, a_2), (i_S, b_1)\}$ – дуги, исходящие из истока; $V_T = \{(a_1, i_F), (b_2, i_F)\}$ – дуги, входящие в сток; $W_T: \{I_I \cup \{i_S\} \cup \{i_F\}\} \rightarrow R^+ \cup \{0\}$ – числовая функция, определяющая длительность выполнения операций так, что $W(a_2) = 1$, $W(a_1) = 3$, $W(b_1) = 2$, $W(b_2) = 4$, $W(i_S) = W(i_F) = 0$.

Учитывая то, что решение задачи сводится к поиску наилучших порядков выполнения операций постами $q \in Q$, на графе G_T любые конкретные порядки можно отобразить множеством дополнительных дуг. Пусть $I_q \subset I_I$, $1 \leq q \leq m$ – множества вершин графа G_T , соответствующие множествам операций I_q , выполняемых m постами. Тогда любые порядки выполнения операций постами $q \in Q$ отобразятся множествами V_q , $1 \leq q \leq m$ дуг, связывающих попарно вершины каждого множества I_q так, что для любой пары вершин из I_q дуга исходит из некоторой вершины и заходит в другую вершину, если операция, отображаемая первой вершиной, непосредственно предшествует операции, отображаемой второй вершиной. В результате получаем граф $G = (I_I, i_S, i_F, V, W_T)$, где $V = V_T \cup V_m$, $V_m = \cup_{q=1}^m V_q$.

Множество дуг V_m для сети G представляют дуги $a_2 b_2$, $b_1 a_1$. Они указывают на то, что сначала на посту 1 должна выполняться операция b_1 , затем – операция a_1 . На посту 2

сначала необходимо выполнить операцию a_2 , затем – операцию b_2 . Для такой сети известным способом можно вычислить ранние сроки начала выполнения операций, которые представляют расписание U , и значение критического пути L , которое определит минимальное время обработки автомобилей с заданным порядком выполнения операций на постах. Таким образом, формируя $\prod_{q=1}^m |I_q|!$ различных порядков обработки автомобилей на постах и рассчитывая все соответствующие сети G методом критического пути, можно установить сеть G^* с наименьшей длиной L^* , порядок, который её определяет, и соответствующее расписание U^* .

К сожалению, для общей задачи, в отличие от задачи конвейерного типа, не каждый наугад выбранный порядок выполнения операций порождает ациклический граф G , допускающий применение метода критического пути. Для исключения таких случаев применяется универсальная процедура построения ациклических графов, изложенная и доказанная [2]. Она предусматривает построение ациклического графа G начиная с вершины i_S . Убедимся в том, что ациклический граф может быть построен начиная с вершины i_F , т.е. в порядке, обратном технологическим последовательностям выполнения операций.

Пусть i_F – отмеченная вершина графа G , а $I_F \subseteq I_I$ – множество неотмеченных его вершин, т.е. таких вершин из I_I , исходящие дуги которых заходят только в отмеченные вершины. Пусть I_q^F – отмеченные вершины из множества I_q , $q \in Q$. Процедура включает следующие действия: 1) выделить множество I_F ; 2) выбрать вершину $iq \in I_F$; 3) если $|I_q \setminus \{I_q^F \cup \{iq\}\}| \geq 1$, каждую вершину множества $I_q \setminus \{I_q^F \cup \{iq\}\}$ соединить дугой с выбранной вершиной iq ; 4) отметить вершину iq ; 5) выделить множество I_F ; 6) если $I_F = \emptyset$, остановиться; иначе вернуться к пункту 2.

Правильность изложенной процедуры вытекает из следующего. Процесс выбора и пометки вершин $iq \in I_F$ последовательный.

Поэтому образование цикла в формируемом графе G означало бы, что выбор некоторой вершины $iq \in I_F$, уже выбранной и помеченной, повторился. В свою очередь это бы означало, что первичный выбор этой вершины был осуществлен так, что некоторые исходящие из нее дуги не заходили в отмеченные вершины. Поскольку последнее противоречит действиям процедуры, выбор помеченных вершин не повторяется и последовательный процесс их пометки будет доведен до конца без повторения. В результате полученный граф G окажется ациклическим.

К построенному таким способом графу G может быть применен метод критического пути, в результате чего найдены ранние сроки начала выполнения операций – расписание U и длина критического пути L – минимальное время обработки автомобилей с тем порядком выполнения операций на постах, который был определен в результате построения графа G . Принципиальная возможность формирования всех ациклических графов G и выбора графа с наименьшей длиной критического пути обеспечивается различными вариантами выбора вершин iq из множеств I_F на втором шаге процедуры. Реализуя процедуру в режиме перебора всех вариантов выбора вершин $iq \in I_F$, теоретически можно получить все ациклические графы G .

В практическом отношении реализация алгоритмов с обратным упорядочением работ несколько отличается от традиционно принятой схемы. В алгоритмах с прямым упорядочением работ одновременно с построением ациклического графа G осуществляется составление расписания U и в конце – вычисляется длина L [2]. Если принять эту схему для алгоритмов с обратным упорядочением работ и отсчет времени вести справа налево, используя элементы матриц $M_I = \|q_{ij}\|$, $T = \|t_{ij}\|$ от последних к первым [1], можно вычислить длину L , найти порядки выполнения операций $P = \|i_{qp}\|$. Однако составить расписание U из-за обратного отсчета времени нельзя. Поэтому все алгоритмы с обратным упорядочением работ реализуются в два этапа. Сначала по традиционной схеме определяется порядок $P = \|i_{qp}\|$ и вычисляет-

ся L , затем на основании матриц $M_I = \|q_{ij}\|$, $P = \|i_{qp}\|$ составляется расписание, соответствующее P и L .

Пусть, как и в [2], U_{iq} , U_q – моменты начала некоторой работы $i \in I_I$ на посту $q \in Q$ и загрузки поста $q \in Q$. Тогда $U_{iq}^S = \max(U_{iq}, U_q)$ – самое раннее начало работы $i \in I_I$ на посту $q \in Q$, а $U_{iq}^F = U_{iq}^S + t_{iq}$ – время ее завершения. Процедура составления расписания U на основании матриц M_I и P включает следующие действия: 1) положить $S = 1$; в матрицах M_I , P выделить множества операций I_I^S , I_P^S , непосредственно предшествующие которым уже отмечены; 2) $\forall iq \in I_I^S$ положить $U_{iq} = 0$; $\forall q \in Q$ положить $U_q = 0$; 3) выделить множество $I^S = I_I^S \cap I_P^S$; 4) для всех операций $iq \in I^S$ вычислить U_{iq}^S , U_{iq}^F , положить $U_q = U_{iq}^F$, занести значения U_{iq}^S в матрицу U и отметить в M_I и P операции множества I^S ; 5) положить $S = S + 1$; в матрицах M_I , P выделить множества I_I^S , I_P^S ; 6) если $I_I^S \neq \emptyset$, для всех новых операций iq , поступивших в I_I^S , положить $U_{iq} = U_{iq}^F$ и вернуться к пункту 3; иначе остановиться.

Правильность процедуры вытекает из того, что для всех $1 \leq S \leq |I_I|$ множества $I^S \neq \emptyset$, так как порядки операций P определены ациклическими графами G .

Экспериментальные исследования алгоритмов

Экспериментальные исследования алгоритмов с обратным упорядочением работ были предприняты с целью получения статистических характеристик их погрешностей, а также времен счета задач и сравнения этих характеристик с аналогичными характеристиками ранее исследованных алгоритмов с прямым упорядочением работ. Для этого были отобраны лучшие алгоритмы KN , KNK , дающие, согласно [2, 3], наименьшую погрешность, преобразованы в алгоритмы с обратным упорядочением работ. Статистиче-

ские данные обрабатывались средствами пакета *STATISTICA 6* фирмы *StatSoft*. В качестве результирующих данных для ряда размеров задач, определяемых количеством автомобилей n и постов m , вычислялись следующие статистические характеристики: Δ_{\min} , Δ_S , Δ_{\max} – минимальная, средняя и максимальная погрешности решения задач (в %) по отношению к наименьшей нижней границе длины L [2]; S_{KO} , d_l , d_r – стандартное отклонение, левая и правая границы 95 %-го доверительного интервала для средней погрешности; t_s – среднее время решения задачи (сек). Для получения этих характеристик по каждому размеру программно генерировалось сто случайных задач с временами выполнения операций, заключенными в интервале (1, 100).

Данные экспериментов показывают, что, во-первых, алгоритм с обратным упорядочением работ обладает таким же характером изменения средней погрешности в зависимости от размера задачи при $n/m = \text{const}$ и в зависимости от отношения n/m при $n \times m = \text{const}$, как и алгоритм с прямым упорядочением работ. Во-вторых, не наблюдается существенного превосходства алгоритма с обратным упорядочением работ по отношению к алгоритму *KN*. Полученные данные показывают, что для него лишь более интенсивно падение средней погрешности при увеличении n/m и менее интенсивен ее рост с увеличением размера задачи при $n/m = 4$. В-третьих, алгоритм с обратным упорядочением работ обладает столь же низким разбросом случайных погрешностей и узким доверительным интервалом для средней ее величины, как и алгоритм с прямым упорядочением работ, что свидетельствует о высокой статистической устойчивости изучаемых характеристик алгоритма.

Результаты экспериментов показывают, что для задачи конвейерного типа (алгоритмы *KNK*, *KNKOB*) так же, как и для общей задачи теории расписаний, алгоритм с обратным упорядочением работ по статистикам погрешности не превосходит алгоритм с прямым упорядочением работ. Можно сказать, что в практическом отношении эти алгоритмы равноценны. В заключение отметим, что процессорное время решения задач

с большим отношением n/m , например, для задачи конвейерного типа при $n/m = 4$, оно составляет приблизительно 28,5 с, может быть существенно снижено путем использования управляющего параметра ε [2, 3]. Эксперимент показал, что при $\varepsilon = 0,2$ для задачи конвейерного типа размера $n \times m = 80 \times 20$ и отношения $n/m = 4$ это время будет составлять 20,3 с вместо 28,5 с при увеличении средней погрешности на 0,78 %, для общей задачи – 4,6 с вместо 11,2 с – при увеличении средней погрешности на 0,27 %.

Выводы

Научная новизна разработки состоит в конструировании и экспериментальном исследовании алгоритмов решения общей и конвейерной задач теории расписаний с обратным упорядочением работ при составлении расписания.

Практическая значимость исследований состоит в том, что на основании обширных, исчерпывающих экспериментальных данных дается правдоподобное заключение о практическом применении алгоритмов.

Литература

1. Танаев В.С. Введение в теорию расписаний / В.С. Танаев, В.В. Шкурба. – М.: Наука, 1975. – 256 с.
2. Канцедал С.А. Приближенный алгоритм для решения общей задачи теории расписаний с высокой точностью / С.А. Канцедал, М.В. Костикова // Радиоэлектроника и информатика. – 1999. – № 4 (9). – С. 97–105.
3. Канцедал С.А. Применение алгоритмов решения общей задачи для составления расписаний в конвейерной системе / С.А. Канцедал, М.В. Костикова // Автоматизированные системы управления и приборы автоматики. – 2003. – Вып. 123. – С. 13–21.

Рецензент: В.П. Волков, профессор, д.т.н., ХНАДУ.

Статья поступила в редакцию 21 ноября 2011 г.