

УДК 004.023

МЕТОД ВЕТВЕЙ И ГРАНИЦ С ЭВРИСТИЧЕСКИМИ ОЦЕНКАМИ ДЛЯ КОНВЕЙЕРНОЙ ЗАДАЧИ ТЕОРИИ РАСПИСАНИЙ

© 2008 г.

М.Х. Прилуцкий, В.С. Власов

Нижегородский госуниверситет им. Н.И. Лобачевского

vlasov_nn@mail.ru

Поступила в редакцию 30.04.2008

Рассматривается конвейерная задача построения оптимального по быстродействию расписания. Приводятся эвристические и точные алгоритмы и схема последовательного использования изложенных алгоритмов.

Ключевые слова: метод ветвей и границ, эвристическая оценка, конвейерная задача, теория расписаний.

Введение

На содержательном уровне общая задача теории расписаний может быть описана следующим образом [1]. На нескольких станках должны пройти обработку несколько деталей. Для каждой детали задан порядок ее обработки на станках. Обработку детали на станке обычно называют операцией. Каждый станок одновременно не может выполнять более одной операции, а операции обрабатываются на станках без перерывов. Задана матрица времен выполнения операций на станках. Требуется определить такой порядок выполнения операций на станках, который минимизирует время завершения изготовления всех деталей.

Существует широкий класс прикладных задач, формализация которых приводит их к классу задач теории расписаний. Это задачи производственного календарного планирования, сетевого планирования и управления, задачи управления проектами [2–4]. В настоящее время, в связи с развитием средств информатики и вычислительной техники, методы теории расписаний применяются при решении задач организации больших информационных систем – создания аппаратно-программных комплексов параллельной обработки данных, задач усовершенствования микроархитектуры процессоров (одна из особенностей современных процессоров – наличие блока диспетчера обработки, который планирует расписание обработки микроопераций с учетом их взаимозависимости по данным и наличным ресурсам).

Особое место среди задач теории расписаний принадлежит так называемым перестановочным (конвейерным) расписаниям, когда порядок выполнения операций для каждой детали одинаков. Не меняя общности в конвейерных

задачах можно считать, что для каждой детали нумерация станков (обычно в естественном порядке) соответствует порядку выполнения операций. Кроме того, предполагается, что на каждом станке порядок выполнения операций одинаков для каждой детали. Примером такой системы может служить конвейер (сборочная линия). Вообще, в качестве конвейерной системы может рассматриваться любая совокупность станков, которые выполняют все операции в одном и том же порядке. Для такой системы вовсе не обязательно, чтобы каждая деталь состояла из операций, выполняемых на каждом станке, или чтобы все операции начинались и заканчивались определенными станками. Существенно лишь, что все перемещения операции, связанные с окончанием ее выполнения на одном станке и началом выполнения на другом, должны происходить в одном направлении. В конвейерных задачах теории расписаний допустимое решение определяется перестановкой из n чисел (по числу деталей), а оптимальное решение содержится среди $n!$ допустимых решений. Как общая задача теории расписаний, так и конвейерная задача относятся к классу NP-трудных, что обуславливает необходимость построения приближенных алгоритмов их решения.

Математическая модель

Пусть i – номер станка, j – номер детали, $T = \|t_{ij}\| - m \times n$ действительная матрица, элемент $t_{ij} \geq 0$ которой определяет время выполнения детали j на станке i , $i = \overline{1, m}$, $j = \overline{1, n}$.

Требуется найти такую матрицу $X = \|x_{ij}\|$ размерности $m \times n$, элемент которой x_{ij} – определяет момент начала выполнения детали j на

станке i , $i = \overline{1, m}$, $j = \overline{1, n}$, для которой выполняются ограничения:

$$x_{ij} \geq x_{i-1j} + t_{i-1j}, \quad i = \overline{2, m}, \quad j = \overline{1, n}, \quad (1)$$

(обработка детали на станке может начаться не раньше, чем эта деталь завершит обработку на предыдущем станке),

$$x_{ij} \geq x_{ik} + t_{ik}, \quad \text{или} \quad x_{ik} \geq x_{ij} + t_{ij}, \quad i = \overline{1, m}, \\ j = \overline{1, n}, \quad k = \overline{1, n}, \quad (2)$$

(на станке одновременно не может обрабатываться более одной детали),

$$\text{если } x_{ij} \geq x_{ik}, \text{ то } x_{sj} \geq x_{sk}, \quad i = \overline{1, m}, \\ s = \overline{1, m}, \quad j = \overline{1, n}, \quad (3)$$

(порядок обработки деталей одинаков на всех станках – перестановочность расписания),

$$x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad (4)$$

(естественные условия на переменные), и достигает минимального значения критерий

$$F(X) = \max_{j=1, n} (x_{mj} + t_{mj}) \quad (5)$$

(минимизация времени завершения обработки всех деталей).

Исследование математической модели

Поставленная задача содержит логические условия (2)–(3) и нелинейный критерий (5). Путем введения дополнительных переменных (в том числе булевых) и ограничений задачу (1)–(5) можно свести к задаче частично-целочисленного линейного программирования.

Введем переменные: $y_{ijk} = 1$, если на i -ом станке деталь j обрабатывается раньше детали k , и $y_{ijk} = 0$ – в противном случае, $i = \overline{1, m}$, $j = \overline{1, n}$, $k = \overline{1, n}$. Пусть Q – достаточно большое число. Тогда ограничения (2) эквивалентны ограничениям:

$$(Q + t_{ik}) y_{ijk} + x_{ij} - x_{ik} \geq t_{ik}, \\ (Q + t_{ij})(1 - y_{ijk}) + x_{ik} - x_{ij} \geq t_{ij}, \quad (6) \\ i = \overline{1, m}, \quad j = \overline{1, n}, \quad k = \overline{1, n}.$$

Действительно пусть $x_{ij} = x_{ik}$, тогда если $y_{ijk} = 1$, то из ограничений (6) получаем: $Q + t_{ik} + x_{ij} - x_{ik} = Q + t_{ik} \geq t_{ik}$, что всегда верно; $x_{ik} - x_{ij} = 0 \geq t_{ij}$, а поскольку $t_{ij} \geq 0$, то $t_{ij} = 0$. Это означает, что деталь с номером j не проходит обработку на станке с номером i , $i = \overline{1, m}$, $j = \overline{1, n}$, $k = \overline{1, n}$.

Если $y_{ijk} = 1$, то $t_{ik} = 0$, т.е. деталь с номером j не обрабатывается на станке с номером i , $i = \overline{1, m}$, $j = \overline{1, n}$, $k = \overline{1, n}$.

Пусть $x_{ij} > x_{ik}$, т.е. деталь с номером j выполняется после детали с номером k . Тогда $y_{ijk} = 0$ и из условий (6) получаем $x_{ij} - x_{ik} \geq t_{ik}$, т.е. условия (2) выполняются, $i = \overline{1, m}$, $j = \overline{1, n}$, $k = \overline{1, n}$.

Пусть $x_{ij} < x_{ik}$, т.е. деталь с номером k выполняется после детали с номером j . Тогда $y_{ijk} = 1$ и из условий (6) получаем $x_{ij} - x_{ik} \geq t_{ik}$, т.е. условия (2) выполняются, $i = \overline{1, m}$, $j = \overline{1, n}$, $k = \overline{1, n}$.

Логические условия (3) эквивалентны ограничениям

$$y_{ijk} = y_{sjk}, \\ i = \overline{1, m}, \quad s = \overline{1, m}, \quad j = \overline{1, n}, \quad k = \overline{1, n}. \quad (7)$$

Условия (7), как и условия (3), означают, что если на одном станке некоторая деталь обрабатывается раньше другой детали, то этот же порядок обработки будет и для любого другого станка.

Рассмотрим дополнительные ограничения

$$x_{mj} + t_{mj} \leq z, \quad j = \overline{1, n}. \quad (8)$$

Конвейерная задача теории расписаний (1)–(5) эквивалентна задаче (1), (6)–(8) с естественными условиями на переменные

$$x_{ij} \geq 0, \quad y_{ijk} \in \{0, 1\}, \\ i = \overline{1, m}, \quad j = \overline{1, n}, \quad k = \overline{1, n}, \quad (9)$$

и критерием

$$F(X, Y, z) = z \rightarrow \min. \quad (10)$$

Полученная задача (1), (6)–(10) является задачей частично-целочисленного линейного программирования и при небольших значениях параметров может быть решена известными методами целочисленного линейного программирования.

Покажем, что решение задачи (1), (6)–(10) определяет перестановку, которая соответствует оптимальному по быстродействию расписанию обработки деталей на станках.

Пусть $X^0 = \|x_{ij}^0\|_{m \times n}$ – матрица времен начала выполнения операций в оптимальном решении задачи. Для нахождения перестановки, соответствующей оптимальному решению задачи, $\rho^0 = (\rho_1, \rho_2, \dots, \rho_n)$ – достаточно воспользо-

ваться соотношениями: $\rho_k = \arg \min_{j \in J_k} x_{ij}^0$, где

$$J_k = \{1, 2, \dots, n\} \setminus \{\rho_1, \rho_2, \dots, \rho_{k-1}\}, k = \overline{1, n}.$$

При значениях параметров, соответствующих реальным прикладным задачам (десятки станков и сотни деталей), для решения конвейерной задачи теории расписаний предлагается применить метод ветвей и границ с эвристическими схемами получения оценок.

Метод ветвей и границ решения конвейерной задачи теории расписаний

Основные процедуры метода ветвей и границ, применяемые при решении конвейерной задачи теории расписаний, включают в себя индивидуальные процедуры – процедура оценок и процедура ветвления, и универсальные процедуры метода ветвей и границ – процедура отсева и процедура останова.

Процедура оценок включает в себя определение верхней (для рассматриваемой задачи достижимой) оценки V и нижней оценки H . Поскольку любая перестановка из n чисел определяет допустимое решение задачи, то в качестве верхней (достижимой) оценки можно рассматривать значение критерия (5), определяемое некоторой перестановкой.

Для определения нижней оценки на первом шаге используем формулу

$$H^0 = \sum_{i=1}^{m-1} \min_{j=1, n} t_{ij} + \sum_{j=1}^n t_{mj}.$$

Для определения нижней оценки с учетом того, что работа k включена в строящееся расписание первой:

$$H^k = \sum_{i=1}^m t_{ik} + \sum_{j=1}^n t_{mj}, \quad j \neq k.$$

Для определения нижней оценки с учетом того, что в строящееся расписание включены работы k и l :

$$H^{kl} = \sum_{i=1}^m t_{ik} + Dif^m(k, l) + t_{ml} + \sum_{j=1}^n t_{mj}, \\ j \neq k, j \neq l,$$

где функция $Dif^m(k, l)$ определяет задержку начала выполнения работы l на станке m после выполнения на нем работы k и определяется как

$$Dif^i(k, l) = \begin{cases} 0, & \text{если } i = 1; \\ \max\{0, (x_{i-1, k} + t_{i-1, k} + t_{i-1, l} + \\ + Dif^{i-1}(k, l)) - (x_{ik} + t_{ik})\}, & \text{иначе.} \end{cases}$$

Аналогично определяется значение нижней оценки после того, как часть работ уже включены в строящееся расписание.

Генерируя по некоторым правилам перестановки, можно находить значения критерия (5) им соответствующие и в качестве верхней оценки выбирать минимальное значение критерия. Для генерации перестановок ниже приводятся различные стохастические и детерминированные алгоритмы, настроенные на получение перестановок, соответствующих локальным решениям конвейерной задачи.

Процедура ветвления рассматривает все допустимые варианты построения перестановок.

Процедура отсева предполагает, что если значение верхней (достижимой) оценки в одной из вершин дерева ветвлений не больше значения нижней оценки в другой вершине, то вторая вершина исключается из рассмотрения.

Процедура останова определяет окончание процесса вычислений. Если осталась неотброшенная лишь одна вершина, в которой значения оценок совпадают, то найдено оптимальное решение задачи, которое определяется перестановкой, соответствующей верхней оценке.

Основным достоинством метода ветвей и границ является то, что в любой момент времени, прекратив вычисления, лучшее решение («рекорд»), соответствующее верхней (достижимой) оценке, может быть принято за приближенное решение задачи. Основным недостатком метода ветвей и границ является необходимость определения оценок в каждой вершине дерева ветвления, а при большом числе исходных параметров число вершин становится значительным, что не позволяет провести рассмотрение всех вершин дерева ветвлений.

В данной работе предлагается схема метода ветвей и границ, в которой используются две стратегии – стратегия получения совокупности верхних оценок и стратегия отбрасывания направлений (возможно в ущерб оптимальности), позволяющая продолжать процесс решения задачи.

Получение совокупности верхних оценок

Стохастические алгоритмы

А₁. Алгоритм Simulated Annealing. Алгоритм, использующий Simulated Annealing метод для нахождения псевдооптимальной перестановки [5]. В этом подходе используется аналогия между процессом нахождения псевдооптимального решения и моделью охлаждения термодинамической системы. Проводится аналогия между:

- энергией термодинамической системы (E) и обобщенным критерием оптимальности задачи (F), полагая, что их значения совпадают,

• перестановкой (ρ), задающей порядок выполнения работ, и состоянием термодинамической системы (π).

Вводятся случайные операторы перехода из одного состояния системы в другое или, другими словами от одной перестановки работ к другой. Задается эмпирический параметр T – температура термодинамической системы. Из известной формулы распределения вероятностей энергий для термодинамической системы Больцмана следует, что вероятность перехода системы из состояния с энергией E_1 в состояние с энергией E_2 при температуре T , $P(\Delta E) = e^{-\Delta E/kT}$, где $\Delta E = E_2 - E_1$ и k – постоянная Больцмана (коэффициент пропорциональности, задающий отношение между энергией и температурой). Оптимизационный алгоритм работает следующим образом: для некоторого начального параметра T_0 в состоянии π_0 находится E_0 – значение энергии термодинамической системы. Применяется случайный оператор перехода из состояния π_0 в π_1 . Находится значение E_1 для этого состояния. Если $\Delta E \geq 0$ или $P(\Delta E) < \xi$, где ξ – случайное число, равномерно распределенное в диапазоне $[0, 1]$, то π_1 принимается за текущее состояние системы и шаг работы алгоритма повторяется. Иначе шаг повторяется при том же самом состоянии системы π_0 . После kn (n – число деталей, k – управляемый параметр) повторений этого шага предполагается, что происходит «охлаждение термодинамической системы». Условием остановки алгоритма является отсутствие улучшений значения критерия задачи.

A₂. Генетический алгоритм. В основу эволюционно-генетических алгоритмов заложена идея наследственности в биологических популяциях [6]. Структура данных алгоритма состоит из набора хромосом. Алгоритм представляет собой итерационный процесс, который продолжается до тех пор, пока не будет смоделировано заданное число поколений или выполнен критерий останова.

Конвейерное расписание в рассмотренной постановке однозначно определяется перестановкой из n элементов, т.е. является перестановочным. Перестановочное представление формально определяется генетическим представлением, которое задает хромосому в виде упорядоченной последовательности из n генов

(n -мерного вектора). Ген, соответствующий j -ой компоненте, определяет номер детали, которая будет выполняться j -ой по порядку, $j = \overline{1, n}$. Работа генетического алгоритма основана на использовании ряда операторов кроссовера и мутации, которые, используя допустимые генотипы «родительских» особей, сохраняют допустимость генотипов потомков.

Схема работы кроссовера порядка заключается в случайном выборе двух родительских особей s_1 и s_2 . Случайным образом выбирается точка кроссовера, разрезающая генотип родителей на 2 части. Непрерывная последовательность деталей из первой строки копируется в генотип потомка. Формируется блок заполнения из деталей строки s_2 , которых нет в генотипе потомка. Элементы блока заполнения копируются в генотип потомка, основываясь на относительном порядке их следования в s_2 .

Схема работы кроссовера позиции заключается в случайном выборе двух родительских особей s_1 и s_2 . Формируется n -битовая строка, в которой случайным образом размещается m единиц ($1 < m < n$) и n нулей. Элементы родительской особи s_1 , находящиеся в позициях, которым n -битовой строке соответствует единица, копируются в генотип потомка. Формируется блок заполнения из деталей строки s_2 , которых нет в генотипе потомка. Элементы блока заполнения копируются в генотип потомка, основываясь на относительном порядке их следования в s_2 .

Схема работы классического оператора мутации заключается в случайном выборе родительской особи s_p , элементы которой копируются в генотип потомка. Случайным образом выбираются 2 точки мутации α_1 и α_2 . Элементы потомка, находящиеся на этих позициях меняются местами.

Схема работы циклического оператора мутации заключается в случайном выборе родительской особи s_p , элементы которой копируются в генотип потомка. Случайным образом выбираются 2 точки мутации α_1 и α_2 . Элементы потомка, находящиеся между этими позициями отображаются в обратном порядке (т.е. элемент, стоявший первым в этом блоке становится последним, второй – предпоследним и т.д.)

A₃. Алгоритм ant colonies. В основе алгоритма ant colonies используется аналогия между решением оптимизационной задачи и моделированием поведения колонии «муравьев» [7]. Колония муравьев рассматривается как много-агентная система, в которой каждый агент (муравей) функционирует по простым правилам, и при этом поведение всей системы (колонии му-

равьев) приводит к приемлемым результатам. Это качество многоагентной системы следует из так называемой «роевой» логики. При своем движении агент пометает свой путь, и чем больше агентов прошло по данному маршруту, тем больше значения этих пометок и тем больше агентов этот маршрут привлекает. В конвейерной задаче построения оптимального по быстродействию расписания «длина» пути определяется значением критерия, соответствующего расписанию, построенного агентом – время завершения выполнения всех работ, включенных в строящееся расписание. В основе предлагаемого алгоритма лежит квадратная матрица $P(t) = \|p_{ij}(t)\|_{n \times n}$, определяющая состояние системы на такт работы алгоритма t , к которому каждый из агентов уже построил t перестановок. В основу стратегии построения перестановки отдельным агентом заложены как элементы стохастики, так и «предыстория» жизни системы, которая отображается матрицей $P(t)$. Каждая перестановка однозначно определяет конвейерное расписание, а тем самым и значение критерия задачи, соответствующее этому расписанию. Элемент матрицы $p_{ij}(t)$ определяет величину пометок к такту t , которая соответствует тому, что работа с номером i будет входить в искомую перестановку j -ой по порядку. Система функционирует конечное число T_0 тактов. Найденная перестановка и определяет искомое конвейерное расписание.

Эвристические детерминированные алгоритмы

А₄. Алгоритм, основанный на решении задачи о назначениях. Тот факт, что перестановки из n натуральных чисел можно представить в виде квадратных бистохастических булевых матриц, позволил для исследования конвейерной задачи теории расписаний использовать хорошо разработанный аппарат решения задач о назначениях [8]. Пусть $Y = \|y_{ij}\|$ – квадратная матрица порядка n , где $y_{ij} = 1$, если деталь i выполняется j -ой по порядку и $y_{ij} = 0$ – в противном случае, $i = \overline{1, n}$, $j = \overline{1, n}$. Рассмотрим следующую задачу о назначениях:

$$Q(Y^0) = \max \left\{ \sum_{i=1}^n \sum_{j=1}^n c_{ij} y_{ij} \mid \sum_{i=1}^n y_{ij} = 1, j = \overline{1, n}; \sum_{j=1}^n y_{ij} = 1, i = \overline{1, n}; y_{ij} \in \{0, 1\}, i = \overline{1, n}, j = \overline{1, n} \right\}. \tag{11}$$

Пусть P – множество различных решений системы ограничений (1)–(4), а B – множество различных n -мерных бистохастических булевых матриц. Рассмотрим соответствие $\Gamma = (Q, P, B)$, где $Q = P \rightarrow B$. Соответствие Γ является функциональным, инъективным, всюду определенным и сюръективным, т.е. взаимнооднозначным (биективным). Отсюда, исследование задачи построения конвейерных расписаний можно проводить посредством решения задач о назначениях, для чего необходимо связать значения критерия задачи (11), зависящие от коэффициентов функционала, со значениями критерия задачи (1)–(5). Это можно сделать, например, используя в качестве матрицы, определяющей коэффициенты критерия задачи (11), найденную в результате работы алгоритма **ant colonies** матрицу $P(t) = \|p_{ij}(t)\|_{n \times n}$. Кроме того, поскольку в задаче о назначениях в общем случае существует множество оптимальных решений, имеет смысл рассматривать все оптимальные решения у задачи (11), для каждого решения находить соответствующую перестановку, строить по ней допустимое расписание и выбирать из построенных расписаний наилучшее с точки зрения критерия (5).

А₅. Фронтальный алгоритм. Предлагаемый алгоритм основан на идеологии «жадных алгоритмов» – аналоге метода градиента для непрерывных задач. Для каждой детали находится величина $\tau_j = \sum_{i=1}^m t_{ij}$ – суммарная трудоемкость детали, $j = \overline{1, n}$. Строится перестановка из n натуральных чисел (номера деталей) ρ , в которой номера деталей соответствуют невозрастающей (неубывающей) величин τ_j , $j = \overline{1, n}$. Это дает возможность получения перестановок на основе следующих двух стратегий: в первую очередь на выполнение ставится деталь с максимальной трудоемкостью, а в конце – с минимальной и наоборот, на выполнение вначале ставится деталь с минимальной трудоемкостью, а в конце – с максимальной.

А₆. Использование алгоритма Джонсона для задачи с 2 станками. Разобьем множество всех деталей на классы джонсоновских задач для двух станков. Общее число всевозможных вариантов будет C_n^2 . Тогда находя для каждого варианта точное решение алгоритмом Джонсона, фиксируем полученную перестановку и находим для нее значение критерия (5). Перестановка, соответствующая минимальному значе-

нию критерия (5) и определит допустимое решение исходной задачи.

Замечание. Используя алгоритм Джонсона для 2 станков можно сформировать множество из нескольких лучших решений и соответствующих им перестановок, которые могут быть использованы при комбинировании предложенных алгоритмов.

Метод комбинирования алгоритмов

Поскольку большинство из описанных алгоритмов на входе может использовать перестановку (или набор перестановок) в качестве исходных данных, а результатом работы алгоритма также является перестановка, то для наиболее эффективного решения задачи при большом числе исходных параметров, предлагается последовательное использование алгоритмов, когда «лучшая» перестановка (или набор «лучших» перестановок), полученная одним из алгоритмов, используется как начальная для другого алгоритма.

Указываются начальные и конечные алгоритмы и переходы между ними с использованием промежуточных алгоритмов.

Замечание. С помощью предложенных алгоритмов можно генерировать различные схемы построения допустимого расписания, определяющего верхнюю оценку в методе ветвей и границ:

$$\left. \begin{array}{l} A_1 \rightarrow \rightarrow \\ A_3 \rightarrow A_4 \rightarrow \\ A_5 \rightarrow \rightarrow \\ A_6 \rightarrow \rightarrow \end{array} \right\} \rightarrow A_2 \rightarrow V(\rho_0).$$

Вычислительный эксперимент

Для решения рассмотренного класса задач построена и реализована диалоговая программная система (среда Microsoft Windows XP, среда программирования Visual Studio.Net 2005, язык программирования C#). В программном комплексе реализованы описанные алгоритмы. Программное средство обладает интуитивно-понятным интерфейсом, в ходе своей работы выводит пользователю всю необходимую информацию и время работы используемого алгоритма. Для проведения вычислительного эксперимента был программно реализован генератор задач, реализующий исходные данные для конвейерных задач с заданным числом станков и деталей и случайными значениями времен выполнения операций в интервале от 1 до 100.

Эксперименты проводились на машине со следующей конфигурацией: процессор AMD

Athlon(tm) 64 X2 Dual Core processor 6000 + + 3.01 ГГц, 2 Гб оперативной памяти.

После каждого эксперимента находилось значение лучшей верхней оценки F («рекорд»), максимальное значение нижней оценки H и время решения задачи t . Эксперимент проводился для 6 серий по 5 задач в каждой для 10, 20, 50, 100, 500 и 1000 работ. Результаты эксперимента приведены в табл. 1, в которой в графе t значение 10:01 означает, что после 10 минут работа программы прерывалась, и определялось значение «рекорда» F .

Таблица 1

Число работ	Число станков	F	H	t
10	5	790	790	00:51
10	5	790	790	00:46
10	5	779	779	00:51
10	5	695	695	00:52
10	5	789	789	00:52
20	5	1205	1205	10:01
20	5	1227	1216	10:01
20	5	1107	1107	10:01
20	5	1157	1157	10:01
20	5	1337	1337	10:01
50	5	3107	3107	10:01
50	5	2554	2249	10:01
50	5	2750	2700	10:01
50	5	2775	2599	10:01
50	5	2721	2547	10:01
100	5	5271	4590	10:01
100	5	5773	4929	10:01
100	5	5232	4898	10:01
100	5	5526	4682	10:01
100	5	5179	5178	10:01
500	10	27032	25551	10:01
500	10	27188	25253	10:01
500	10	27312	25380	10:01
500	10	27224	25945	10:01
500	10	27334	25949	10:01
1000	10	53193	51825	10:01
1000	10	52916	50461	10:01
1000	10	53096	49841	10:01
1000	10	53041	49515	10:01
1000	10	52984	49979	10:01

В табл. 2 приведены δ_{cp} – средние относительные отклонения значений критерия задачи на найденном решении от значений нижних (недостижимых) оценок $F - H/H$.

Таблица 2

Число работ	Число станков	δ_{cp}
10	5	0
20	5	$\approx 0,001$
50	5	$\approx 0,058$
100	5	$\approx 0,113$
500	10	$\approx 0,063$
1000	10	$\approx 0,054$

Заключение

Учитывая, что при проведении вычислительного эксперимента отклонения значений критерия на найденных решениях определялись не относительно значений критерия на оптимальном решении задачи, а относительно значений нижних (недостижимых) оценок, и при этом отклонения значений критерия относительно значений нижних оценок составляет 5–6% (лишь для 100 работ 11.3%), можно сделать вывод о приемлемости предложенного подхода для решения конвейерных задач теории расписаний.

Список литературы

1. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. М.: Наука, 1975. 256 с.
2. Прилуцкий М.Х., Батишев Д.И., Гудман Э.Д., Норенков И.П. Метод декомпозиций для решения комбинаторных задач упорядочения и распределения ресурсов // Информационные технологии. 1997. № 1. С. 29–33.
3. Прилуцкий М.Х., Власов С.Е. Многостадийные задачи теории расписаний с альтернативными вариантами выполнения работ // Системы управления и информационные технологии. 2005. № 2. С. 44–48.
4. Прилуцкий М.Х. Многокритериальные многоиндексные задачи объёмно-календарного планирования // Известия АН. Теория и системы управления. 2007. № 1. С. 78–82.
5. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by Simulated Annealing // Science. 1983. V. 220. No. 4598. P. 671–680.
6. Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA. 1989. 412 p.
7. Dorigo M., Maniezzo V., Colomi A. The Ant System: Optimization by a colony of cooperating objects // IEEE Transactions on systems, man, and cybernetics – Part B. 1996. 26(1). P. 29–41.
8. Прилуцкий М.Х., Кумагина Е.А. Задача упорядочения работ как задача о назначениях // Вестник Нижегородского государственного университета. Математическое моделирование и оптимальное управление. Нижний Новгород: Изд-во ННГУ, 1999. Вып. 21. С. 18–24.

THE METHOD OF PATHS AND BOUNDARIES WITH HEURISTIC EVALUATION FOR A CONVEYOR PROBLEM OF SCHEDULING THEORY

M.Kh. Prilutsky, V.S. Vlasov

A conveyor problem to construct an optimal operating speed schedule has been considered. Heuristic and strict algorithms are given as well as a scheme of their sequential implementation.