

РЕШЕНИЕ ОБОБЩЕННОЙ ЗАДАЧИ ДЖОНСОНА С ОГРАНИЧЕНИЯМИ НА СРОКИ ВЫПОЛНЕНИЯ ЗАДАНИЙ И ВРЕМЕНА РАБОТЫ МАШИН. Ч. 2. Приближенные методы решения

Ю.А. Зак

Разработан комплексный алгоритм решения flow-shop problem, использующий основанные на различных гипотезах эвристики и правила предпочтения. Предложены правила построения последовательностей выполнения заданий, позволяющие анализировать различные области притяжения локальных минимумов. Показано, что эффективность построенных расписаний повышается благодаря предложенному в работе алгоритму локальной вариации. Дано сравнение с ранее полученными решениями, приведен числовой пример.

Ключевые слова: оптимальное расписание, последовательность выполнения заданий, эвристики, локальные вариации.

ВВЕДЕНИЕ

В первой части работы [1] была приведена постановка задачи, предложены и обоснованы расчетные формулы для вычисления нижней границы суммарной длины расписания и разработаны точные и приближенные алгоритмы решения задачи на основе метода «ветвей и границ». Вторая часть посвящена развитию приближенных методов решения задачи в традиционной постановке (без учета ограничений на допустимые сроки выполнения заданий и ресурсы работы рабочих станций) в условиях большой размерности.

Еще в 1964 г. Джонсоном [2, 3] было доказано, что для числа рабочих станций $m = 2$ может быть получено точное решение и простое решающее правило сформулированной задачи, заключающееся в следующем. Все множество заданий $I = \{1, 2, \dots, i, \dots, n\}$ может быть разбито на два подмножества $I_1 = \{i : t(i, 1) \leq t(i, 2)\}$ и $I_2 = \{i : t(i, 1) > t(i, 2)\}$, где $t(i, 1)$ и $t(i, 2)$ — соответственно длительности первой и второй операций (работ) i -го здания. Если все задания подмножества I_1 расположены в порядке возрастания, а подмножества I_2 в порядке убывания длительностей операций, то в оптималь-

ной последовательности выполнения заданий вначале следует упорядоченное подмножество заданий I_1 , а затем — упорядоченное подмножество I_2 .

Если число рабочих станций больше двух, то сформулированное Джонсоном [2, 3] и описанное в первой части настоящей работы простое решающее правило уже не справедливо, однако оно может быть использовано для построения различных эвристик при конструировании различных алгоритмов получения приближенных решений.

На основе большого вычислительного опыта успешного решения многих прикладных и тестовых задач (см., например, работы [4–6]) могут быть сформулированы некоторые эвристические принципы построения эффективных последовательностей выполнения заданий, в которых в первую очередь должны выполняться задания, у которых:

- «узкое место» (операция с максимальной продолжительностью) расположена дальше всех от первой станции;
- первая операция имеет самую короткую продолжительность (предпочтение первой операции);
- суммарное время всех операций, исключая первую, наиболее длинное (предпочтение оставшейся длинной операции).



Сформулированные эвристические положения лежат в основе многих эвристических, статистических и генетических алгоритмов решения данного класса задач в условиях большой размерности. Эффективные формулы вычисления нижних границ оптимального решения, приведенные в первой части работы, позволяют эффективно останавливать процесс решения при получении последовательности выполнения заданий с суммарной длительностью, удовлетворяющей поставленным требованиям точности.

Многочисленные эвристические методы и правила предпочтения [2, 4–16], генетические алгоритмы (см., например, работы [10, 17–19]) позволяют зачастую найти решение, принадлежащее области глобального или эффективного локального минимума. Однако никакая из предложенных эвристик не гарантирует попадание в область глобального минимума. В результате локального улучшения последовательности выполнения заданий можно в ряде случаев повысить эффективность и сократить длину построенного расписания. Предлагаемые в литературе правила локального улучшения последовательности выполнения заданий путем обмена местами двух индексов [2, 7, 8, 10, 17] носят либо локальный, либо статистический характер и потому не могут быть интерпретированы как методы локального спуска в область глобального минимума. С этой точки зрения представляет интерес создание гибридных методов, которые на основе хороших эвристик позволяют попасть в зону глобального минимума, где методами локальных перестановок местами заданий осуществляется спуск в точку локального минимума этой области. Принимая во внимание полиномиальную сложность и сравнительно небольшой объем вычислений каждого из эвристических алгоритмов, представляет интерес в таком комплексном алгоритме использовать сразу несколько наиболее эффективных по результатам вычислительных экспериментов [9, 10, 17] эвристик.

В работе предложены некоторые гибридные алгоритмы и приближенные методы решения задач большой размерности, использующие различные известные ранее эвристики и правила предпочтения, а также локальные улучшения построенных на основе эвристик последовательностей. Как показали вычислительные эксперименты, в ряде случаев предлагаемые методы позволяют повысить эффективность получаемых известными алгоритмами приближенных решений при незначительном объеме вычислений.

Предложенные и исследованные в работе модификации метода локальной вариации построенного допустимого решения в ряде случаев позволяют

улучшить построенную допустимую последовательность выполнения заданий и могут найти применение в различных эвристических, статистических и генетических алгоритмах решения задачи.

В первой части работы была введена система обозначений, приведены постановка и математическая формулировка задачи, а также формульные выражения для оценки нижней границы длины строящегося расписания. В дальнейшем изложении используются введенные ранее обозначения, установленные свойства и формулы.

Нумерация формул второй части работы продолжает нумерацию формул первой части.

1. ЛОКАЛЬНЫЕ ПРИОРИТЕТНЫЕ ПРАВИЛА ДЛЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Далее исследуются локальные свойства последовательностей выполнения заданий, позволяющие осуществить обмен местами двух рядом стоящих заданий, гарантируя выполнение системы ограничений и обеспечение значения критерия оптимальности не хуже, чем в исходной последовательности.

Рассмотрим некоторый частичный план P^s , в котором определена подпоследовательность выполнения заданий $\tilde{J}^s = \{i_1^s, i_2^s, \dots, i_{l-1}^s, i_l^s, \dots, i_R^s\}$ и рассчитаны значения $\theta^s(i, k)$ и $\sigma^s(i, k)$ всех операций (l, k) подпоследовательности \tilde{J}^s . Рассмотрим некоторую другую подпоследовательность, в которой задания i_{l-1}^s и i_l^s поменяны местами. Обозначим соответственно $\theta^s(i, k | i_{l-1}^s, i_l^s)$, $\sigma^s(i, k | i_{l-1}^s, i_l^s)$ и $\theta^s(i, k | i_l^s, i_{l-1}^s)$, $\sigma^s(i, k | i_l^s, i_{l-1}^s)$ времена начала и окончания выполнения всех операций для этих двух подпоследовательностей $\tilde{J}^s(i_{l-1}^s, i_l^s)$ и $\tilde{J}^s(i_l^s, i_{l-1}^s)$.

Утверждение 1. Если для пары заданий i_{l-1}^s и i_l^s некоторой подпоследовательности \tilde{J}^s выполняется равенство $\sigma^s(i_l^s, m) = \sigma^s(i_{l-1}^s, m) + t(i_l^s, m)$, то обмен местами этих заданий в подпоследовательности \tilde{J}^s не может привести к сокращению суммарного времени выполнения всех заданий (локальному минимуму), и потому в смысле локальных вариаций рассматривается как нецелесообразный. ♦

Утверждение 1 позволяет выделить в последовательности только те пары заданий, для которых

$$\sigma^s(i_l^s, m) > \sigma^s(i_{l-1}^s, m) + t(i_l^s, m), \quad (26)$$

и обмен местами которых в данной перестановке может оказаться целесообразным.

Следующие далее утверждения позволяют проверить локальную эффективность такой «перспективной» перестановки, зачастую не производя расчета времен завершения выполнения всех заданий, стоящих за этой парой в последовательности.

Утверждение 2 (правило абсолютного предпочтения). Если справедлива система неравенств

$$\begin{aligned} \sigma^s(i_{l-1}^s, k|l, l-1) &\leq \sigma^s(i_l^s, k|l-1, l), \\ k &= 1, \dots, m, \end{aligned} \quad (27)$$

то подпоследовательность $\tilde{J}^s(i_l^s, i_{l-1}^s)$ имеет абсолютное предпочтение перед подпоследовательностью $\tilde{J}^s(i_l^s, i_{l-1}^s)$. Если подпоследовательность \tilde{J}^s содержит допустимые расписания выполнения заданий, то перестановка в ней местами двух рядом стоящих заданий i_{l-1}^s и i_l^s не приводит к исключению допустимых или более перспективных и оптимальных расписаний. ♦

Если для рассматриваемых подпоследовательностей справедливо неравенство

$$\sigma^s(i_{l-1}^s, m|l, l-1) > \sigma^s(i_l^s, m|l-1, l), \quad (28)$$

то перестановка местами этих двух индексов может привести к увеличению длины расписания или нарушению системы ограничений.

Правила относительного предпочтения. Обозначим K — множество всех рабочих станций, а K_1 — некоторое его подмножество. Если для подпоследовательностей $\tilde{J}^s(i_{l-1}^s, i_l^s)$ и $\tilde{J}^s(i_l^s, i_{l-1}^s)$ справедлива система неравенств

$$\begin{aligned} \sigma^s(i_{l-1}^s, m|l, l-1) &\leq \sigma^s(i_l^s, m|l-1, l), \\ \sigma^s(i_{l-1}^s, k|l, l-1) &\leq \sigma^s(i_l^s, k|l-1, l), \\ k &\in K_1; \end{aligned} \quad (29)$$

$$\sigma^s(i_{l-1}^s, k|l, l-1) > \sigma^s(i_l^s, k|l-1, l), \quad k \in K/K_1, k \neq m; \quad (30)$$

$$\begin{aligned} \theta^s(i_{l-1}^s, k|l, l-1) + \sum_{i \in I_2^s} t(i, k) &\leq H_k, \\ k &= 1, \dots, m, \end{aligned} \quad (31)$$

то подпоследовательность $\tilde{J}^s(i_l^s, i_{l-1}^s)$ имеет относительное предпочтение перед подпоследовательностью $\tilde{J}^s(i_l^s, i_{l-1}^s)$.

В случае относительного предпочтения перестановка двух рядом стоящих индексов может

привести к нарушению ограничений на завершение выполнения отдельных заданий и на времена работы машин, а также к увеличению длины расписания выполнения всего комплекса работ. Поэтому для окончательных выводов о целесообразности перестановки местами этих двух индексов необходимо рассчитать сроки завершения и следующих заданий, стоявших в исходной подпоследовательности \tilde{J}^s после задания i_l^s .

Утверждение 3. Если для некоторого задания j , стоявшего в подпоследовательности \tilde{J}^s после задания i_l^s , справедливы правила абсолютного предпочтения в виде

$$\begin{aligned} \sigma^s(j, k|l, l-1) &\leq \sigma^s(j, k|l-1, l), \\ k &= 1, \dots, m, \end{aligned} \quad (32)$$

то обмен местами индексов i_{l-1}^s и i_l^s не приведет к нарушению системы ограничений на времена работы рабочих систем или к увеличению длины расписания.

Если для какого-либо задания j , стоящего в подпоследовательности \tilde{J}^s после задания i_l^s , условия (31) не выполняются для последней операции задания (для $k = m$), то обмен местами индексов i_{l-1}^s и i_l^s нецелесообразен. ♦

Так как пересчет сроков завершения выполнения всех операций заданий, стоящих в подпоследовательности \tilde{J}^s после задания i_l^s , связан с определенным объемом вычислений, для практических приложений рекомендуется анализировать не более 2—3 следующих за заданием i_l^s заданий.

2. АЛГОРИТМ ЛОКАЛЬНОЙ ВАРИАЦИИ ПОСЛЕДОВАТЕЛЬНОСТИ ВЫПОЛНЕНИЯ ЗАДАНИЙ (LV-АЛГОРИТМ)

Пусть построена некоторая допустимая последовательность выполнения всего множества заданий $\tilde{L} = \{i_1, i_2, \dots, i_n\}$. Рассмотрим, как на основе локальных перестановок местами двух рядом стоящих в последовательности индексов сократить длину расписания выполнения всего комплекса работ. Определим следующие параметры алгоритма:

— параметр ρ определяет число стоящих в последовательности после рассматриваемой пары заданий, которые анализируются в соответствии с правилами абсолютного и относительного предпочтений;

— параметр β определяет стратегию локального поиска: при $\beta = 0$ обмен местом расположения в последовательности пары индексов производится



до тех пор, пока он потенциально может привести к сокращению длины расписания; при $\beta = 1$ перестановка местами двух рядом стоящих в последовательности заданий происходит до тех пор, пока процесс обмена не достигнет двух стоящих последними в последовательности \tilde{L} индексов (одно-разовый просмотр последовательности \tilde{L});

— счетчики $N1$, $N2$ и $N3$, определяющие целесообразность продолжения отдельных шагов алгоритма.

LV-алгоритм состоит из следующих шагов.

Шаг 0 (предварительный). Устанавливаются $N1 = N2 = 0$, $N3 = n - 1$, переход к шагу 1.

Шаг 1. Если $N1 = N3$ и $N2 = 0$, то переход к шагу 4. Если $N1 = N3$ и $N2 > 0$, то при $\beta = 0$ полагается $N1 = 0$, $N2 = 0$ и переход к шагу 2, а при $\beta = 1$ — алгоритм заканчивает работу. Если $N1 < N3$, то переход к шагу 2.

Шаг 2. Если для пары индексов $i_{l-1}^s = M1$ и $i_l^s = M1 + 1$ выполняются условия (26), то проверяются для них условия абсолютного предпочтения (32). В случае их выполнения производится обмен местами этих заданий в последовательности, в счетчики $N1$ и $N2$ добавляется 1 и переход к шагу 4. Если условия (26) не выполняются, либо при выполнении условия (26) справедливо неравенство (28), $M1$ увеличивается на единицу и переход к шагу 1. При выполнении условия (26) совместно с правилами относительного предпочтения (29)—(31) $M1$ увеличивается на единицу и переход к шагу 3.

Шаг 3. Определяем число анализируемых заданий, стоящих в последовательности после задания i_l^s , сроки завершения которых должны быть рассчитаны для определения локальной перестановки этих двух индексов: $\rho1 = \min(\rho, R - M1 - 1)$. Для заданий $j = i_p^s$, $p = l + 1, \dots, l + \rho1$, подпоследовательности $J^s(i_l^s, i_l^s - 1)$ вычисляются сроки завершения всех операций $\sigma^s(j, k|l, l - 1)$, $k = 1, \dots, m$, и проверяется выполнение условий (27) и (28). Если для какого-либо j выполняются неравенства (27), то значение счетчика $N2$ увеличивается на 1 и переход к шагу 1. Если условия (28) выполняются, либо ни для одного из этих $\rho1$ заданий ни одна из систем неравенств (27) или (28) не выполняется, то переход к шагу 1.

Шаг 4. Для всех заданий $j = i_p^s$, $p = l + 1, \dots, l + R$, новой подпоследовательности $J^s(i_l^s, i_{l-1}^s)$ пересчитываются времена завершения операций на каждой машине. Переход к шагу 1.

3. ГИБРИДНЫЕ ПРИБЛИЖЕННЫЕ И ЭВРИСТИЧЕСКИЕ МЕТОДЫ РЕШЕНИЯ FLOW-SHOP PROBLEM

Предложенные Джонсоном [3] простые правила предпочтения для получения точного решения в случае двух машин нашли широкое применение при построении различных эвристических приближенных алгоритмов решения задачи в случае, когда число машин $m > 2$. Далее на основе предложенных ранее отдельных эвристик конструируется некоторый комплексный алгоритм генерирования, локального улучшения и оценки степени приближения построенных на основе различных эвристик расписаний с последующим отбором наилучшего из них. Построенное расписание может быть в дальнейшем улучшено описанным в § 2 алгоритмом локальных вариаций.

В конструируемом алгоритме используются следующие эвристики и правила предпочтения.

- Эвристика [6], согласно которой все машины разделяются на две группы. Для каждого задания суммируются времена обработки на машинах каждой группы

$$T_1(i, r) = \sum_{k=1}^r t(i, k), \quad T_2(i, r) = \sum_{k=r+1}^m t(i, k), \\ i = 1, \dots, n, \quad r = 0, \dots, m, \quad (33)$$

и задача в дальнейшем решается построением расписания с помощью алгоритма Джонсона [3] для случая двух машин. По результатам решения таких $(m - 1)$ проблем выбирается наилучшее решение. Сложность такого алгоритма равна $O(nm^2 + mn \log n)$ [6].

- Предложенные в работе [7] выражения в виде

$$T_1(i) = \sum_{k=1}^m (m - k + 1)t(i, k), \\ T_2(i) = \sum_{k=1}^m kt(i, k) \quad (34)$$

с дальнейшим одноразовым применением алгоритма Джонсона.

- Построение приоритетных правил с присвоением каждому заданию некоторого ранга в соответствии с алгоритмом [20] (engl-slope index), вычисляемого по формуле

$$V(i) = \sum_{k=1}^m (2k - m - 1)t(i, k) = \\ = 2 \sum_{k=1}^m kt(i, k) - (m - 1) \sum_{k=1}^m t(i, k), \quad (35)$$

затем задания располагаются в последовательности в соответствии с монотонно убывающим рангом.

Заметим, что наряду с применяемыми можно воспользоваться и другими правилами ранжирования заданий, предложенных в работах [4, 5] и др. Алгоритм [11] со сложностью $O(mn^3)$ вытекает из выражения (33) при $r = 0$, когда все задания упорядочиваются в соответствии с убыванием значений $T_2(i, 0)$.

Отличительные особенности предлагаемого алгоритма:

- возможность анализа эффективности полученного решения на основе вычисленной по формуле (1) или (12) оценки (см. первую часть [1] работы);

- процедура локального улучшения каждого построенного на основе различных эвристик расписания с помощью LV-алгоритма.

Существенные преимущества описанного в § 2 LV-алгоритма относительно известных в литературе (см., например, [2, 7, 8, 10, 17]) методов локальной вариации построенных последовательностей (перестановок) заключаются в:

- целенаправленности выбора рядом стоящей пары индексов для обмена;

- использовании решающих правил для оценки возможности локального улучшения расписания, что требует существенно меньшего объема вычислений;

- достижении локального минимума для рассматриваемой последовательности.

Длина каждого вновь построенного и локально улучшенного расписания сравнивается с оценкой нижней границы оптимального решения, вычисленной в соответствии с выражением (1), и лучшим решением, полученным по результатам проведенных вычислений. При достижении требуемой точности полученного результата процесс решения завершается.

Рассматриваемая задача многоэкстремальная. Какая-то из используемых и построенных из различных соображений эвристик должна обеспечить попадание в некоторую область локального или глобального минимума со значением критерия оптимальности, близким к длине оптимального расписания. Дальнейший спуск в точку локального минимума этой области методом локальных вариаций в ряде случаев должен повысить эффективность полученного эвристическими алгоритмами решения.

3.1. Алгоритм В

Алгоритм приближенного решения задачи состоит из следующих шагов.

Шаг 0 (предварительный). Полагается $E = \infty$, $r = 0$. Устанавливается требуемая точность решения задачи $\varepsilon \geq 1$ (в частности, если $\varepsilon = 1$, ставится требование получение точного решения). Вводится некоторый параметр алгоритма Z , определяющий логическую последовательность выполнения шагов. Полагается $Z = 0$. Рассчитывается нижняя граница длины оптимального расписания по формуле (1), по значению которой определяется F_{\min} , переход к шагу 1.

Шаг 1. Рассчитываются модифицированные времена выполнения каждого из заданий $T_1(i)$ и $T_2(i)$, $i = 1, \dots, n$, в соответствии с формулой (34). Переход к шагу 2.

Шаг 2. Строится последовательность выполнения заданий на основе алгоритма Джонсона. Полагается F — длина построенного расписания. Переход к шагу 3.

Шаг 3. Если $F < E$, то полагается $E = F$ и запоминается построенная последовательность как наилучшее из полученных к данному моменту решений. Если $E \leq \varepsilon E_{\min}$, то получено решение задачи, и алгоритм завершает работу. В противном случае, если $Z = 0$, переход к шагу 4, положив $Z = 1$. Если $Z \neq 0$, переход к шагу $(Z + 4)$.

Шаг 4. Выполняется LV-алгоритм локальной вариации последовательности выполнения заданий. Переход к шагу 3.

Шаг 5. Расчет рангов выполнения заданий в соответствии с формулой (35), строится последовательность выполнения заданий в соответствии с монотонно убывающим рангом. Полагается F — длина построенного расписания. Переход к шагу 3, положив $Z = 2$.

Шаг 6. Если $r \leq m$, рассчитываются модифицированные времена выполнения каждого из заданий $T_1(i)$ и $T_2(i)$, $i = 1, \dots, n$, в соответствии с формулой (33). Переход к шагу 2, положив $r = r + 1$. Если $r > m$, то алгоритм заканчивает работу. Запомненная в памяти последовательность выполнения заданий с длиной расписания равной E является решением задачи. ♦

Разработано программное обеспечение, позволяющее решать практические задачи размерностью до нескольких десятков и даже сотен заданий и выводить результаты решения как в форме таблиц, так и в графической форме — в виде диаграмм Ганта.

Многочисленные вычислительные эксперименты показали эффективность предлагаемых алгоритмов. Решения с длиной расписания, не превышающей 1–3 % вычисленной по формулам (1) нижней границы, были получены за полиномиальное время. Кроме того, в процессе решения часто



Таблица 1

Исходные данные примера

i	$k = 1$	$k = 2$	$k = 3$	$k = 4$
1	15	18	25	30
2	12	27	30	5
3	25	30	40	18
4	35	5	14	20
5	40	10	5	5
6	10	20	30	10
7	15	35	40	10
8	18	12	30	15
9	20	25	30	25
10	15	18	10	45

удается построить несколько расписаний, удовлетворяющих требованиям точности, длина которых отличается несущественно. Если последовательности выполнения заданий в таких расписаниях совершенно различны, то наиболее приемлемое из них может выбираться из других производственных соображений или согласно другому критерию оптимальности.

4. ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

Времена выполнения каждой из четырех операций для 10 заданий представлены в табл. 1. Нижняя граница длины расписания, вычисленная по формуле (1) определяется машиной 4 и равна 289. Решение примера, полученное с помощью алгоритма В, сведено в табл. 2.

Расчитанные на шаге 1 алгоритма А в соответствии с эвристикой [7] модифицированные времена вы-

полнения заданий представлены в столбцах 2 и 3 табл. 2. Последовательность выполнения заданий, рассчитанная для этих времен алгоритмом Джонсона [3] с длиной расписания 304, представлена в столбцах 4–8. Далее на шаге 3 с помощью LV-алгоритма была осуществлена локальная вариация построенной последовательности. Целесообразным был лишь обмен местами заданий 3 и 7, 7 и 2, а также 4 и 2 (см. условия (2)). Как показали правила абсолютного и локального приоритета (3) и (27), (4) и (28), первые два обмена не приводят к более эффективному результату. Обмен местами заданий 4 и 2 привел к расписанию, длина которого равна нижней границе. Следовательно, это расписание оптимально, и необходимость в выполнении дальнейших шагов отпадает.

Предложенный в работе алгоритм локальной вариации уже после выполнения его первого шага позволил существенно (на 15 временных интервалов) повысить эффективность построенного на основе эвристических правил расписания.

Отметим, что построенная в соответствии с правилами предпочтения (10) [20] последовательность выполнения заданий $\Pi = \{10, 1, 9, 6, 8, 7, 3, 2, 4, 5\}$ с длиной расписания $F = 315$, которая, будучи улучшена алгоритмами локальной вариации, имеет вид $\Pi = \{10, 1, 9, 6, 8, 3, 7, 4, 2, 5\}$ при $F = 300$, приводит в совершенно другую область локального минимума. В другие области локального минимума приводит построение модифицированных времен в соответствии с эвристиками [6, 11] и др. (см. формулы (9)) и последующими шагами 6 алгоритма В. Например, $\Pi = \{6, 2, 7, 1, 10, 8, 9, 3, 4, 5\}$ и $F = 313$ при $k = 1$; $\Pi = \{10, 1, 9, 4, 3, 8, 6, 7, 5, 2\}$ и $F = 320$ при $k = 3$; $\Pi = \{8, 6, 10, 1, 9, 7, 3, 5, 4, 2\}$ и $F = 309$ при $k = 2$. Последнее расписание LV-алгоритмом преобразовывается в другое оптимальное решение: $\Pi = \{8, 6, 10, 1, 9, 3, 7, 5, 4, 2\}$ и $F = 289$.

Таблица 2

Эффективное и оптимальное решения примера

i	$T_1(i)$	$T_2(i)$	Последовательность	Времена завершения операций и заданий				Оптимальная последовательность	Времена завершения операций и заданий			
				$k = 1$	$k = 2$	$k = 3$	$k = 4$		$k = 1$	$k = 2$	$k = 3$	$k = 4$
1	194	246	6	10	30	60	70	6	10	30	60	70
2	194	176	10	25	48	70	115	10	25	48	70	115
3	288	277	8	43	60	100	130	8	43	60	100	130
4	203	167	1	58	78	125	160	1	58	78	125	160
5	205	95	9	78	103	155	185	9	78	103	155	185
6	170	180	3	103	133	195	213	3	103	133	195	213
7	255	245	7	118	168	235	245	7	118	168	235	245
8	183	192	2	130	195	265	270	4	153	173	249	269
9	240	260	4	165	200	279	299	2	165	200	279	284
10	179	261	5	205	215	284	304	5	205	215	284	289

Из двух различных оптимальных решений $\Pi_1 = \{6, 10, 8, 1, 9, 3, 7, 4, 2, 5\}$ и $\Pi_2 = \{8, 6, 10, 1, 9, 3, 7, 5, 4, 2\}$, различающихся последовательностями выполнения заданий, может быть выбрано любое из них, обеспечивающее лучшее значение какого-то другого критерия эффективности.

Приведенный числовой пример наглядно иллюстрирует целесообразность использования в одном алгоритме нескольких различных эвристик с последующим локальным улучшением каждого из полученных решений.

5. РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

Автором был проведен вычислительный эксперимент по решению тестовых задач большой размерности. Размерность задач — число заданий m и число машин n — выбиралась случайным образом. Во всех рассмотренных примерах ограничения на сроки завершения выполнения заданий и времена работы машин не учитывались. Времена выполнения различных операций в каждом из заданий выбирались случайным образом в диапазоне $t_{ik} \in [5, 25]$. Каждый из примеров был решен с применением различных эвристик, после чего полученное расписание преобразовывалось описанными алгоритмами локальной вариации (спуск в соответствующую область локального минимума). Результаты решения некоторых характерных примеров сведены в табл. 3. Жирным шрифтом выделены полученные наиболее эффективные приближенные решения, а курсивом — наилучшее из решений, полученных различными эвристическими алгоритмами.

Приведенная таблица показывает, что эффективность каждой из известных в литературе эвристик для различных задач различна, с помощью каждой из них могут быть получены отличающиеся друг от друга решения, принадлежащие различным областям притяжения локального минимума. Метод локальной вариации позволяет зачастую на 2—6 % повысить эффективность полученного эвристическими алгоритмами решения. Причем в некоторых случаях полученное даже существенно менее эффективное, чем на основе других эвристик, решение предложенными в работе методами локальной вариации может быть преобразовано в самое эффективное расписание. Учитывая, что значение критерия оптимальности в точном решении задачи, как правило, на 2—4 % превосходит значение нижней границы, вычисленной на начальном этапе решения, полученные результаты вычислительных экспериментов свидетельствуют об эффективности предлагаемых приближенных методов.

ЗАКЛЮЧЕНИЕ

Разработаны модифицированные алгоритмы локальной вариации построенных последовательностей выполнения заданий, гарантирующие сохранение всех расписаний, содержащих допустимые и оптимальные решения, и осуществляющие спуск в точку локального минимума. Предложенные алгоритмы требуют существенно меньшего объема вычислений по сравнению с применяемыми на практике алгоритмами локального улучшения построенных расписаний.

Для традиционной flow-shop problem (при отсутствии ограничений на сроки выполнения зада-

Таблица 3

Результаты вычислительных экспериментов: эффективность применения различных эвристик и метода локальных вариаций

№ п/п	Размерность задачи		Отклонения (в процентах) полученного различными алгоритмами решения от нижней границы длины оптимального расписания (до и после применения алгоритмов локальной вариации)							
	m	n	[6]		[7]		[20]		[11]	
			до	после	до	после	до	после	до	после
1	25	6	7,3	1,88	8,1	3,57	<i>6,4</i>	2,42	9,35	3,26
2	50	7	6,43	4,37	7,12	4,18	7,81	5,23	<i>5,16</i>	3,92
3	70	8	8,56	3,38	<i>8,44</i>	5,66	10,13	4,76	9,64	6,28
4	85	5	<i>7,81</i>	6,43	8,71	5,21	9,05	5,07	10,12	7,32
5	100	7	<i>9,38</i>	6,98	9,67	6,33	11,24	8,43	10,83	7,52
6	110	10	11,19	7,05	12,32	6,89	<i>9,68</i>	7,59	11,82	8,17
7	120	6	<i>9,87</i>	7,28	11,43	8,14	10,77	7,13	9,89	7,93
8	135	5	<i>10,33</i>	9,15	11,01	8,42	10,81	6,98	11,14	7,76



ний и времени работы машин) разработаны комплексные алгоритмы решения, использующие основанные на различных гипотезах и известные в литературе эвристики и правила предпочтения и позволяющие осуществить попадание в различные области притяжения локальных минимумов. Эффективность работы алгоритмов повышается благодаря предложенным в работе методам локальной вариации и сравнения длины построенных расписаний.

Разработано программное обеспечение, позволяющее решать практические задачи размерностью до нескольких сотен заданий и выводить результаты решения как в форме таблиц, так и в графической форме — в виде диаграмм Ганта.

ЛИТЕРАТУРА

1. *Зак Ю.А.* Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения заданий и времена работы машин. Ч. 1. Точные методы решения // Проблемы управления. — 2010. — № 3. — С. 17–25.
2. *Johnson S.M.* Optimal two- and tree-stage production schedules with setup times included // Research Logistics Quarterly. — 1954 — Vol. 1. — P. 61–68.
3. *Domschke W., Scholl A., VoЯ S.* Produktionsplanung. Ablauforganisatorische Aspekte. — Berlin: Springer Verlag, 1997. — 456 p.
4. *Hundal T.S., Rajgopal J.* An extension of Palmer's heuristic for the flow-shop scheduling problem // International Journal of Production Research. — 1988. — Vol. 26. — P. 1119–1124.
5. *Gupta J.N.D.* A functional heuristic algorithm for the flow-shop scheduling problem // Operational Research Quarterly. — 1971. — Vol. 2. — P. 39–47.
6. *Cambell H.G., Dudek R.A., Smith M.L.* A heuristic algorithm for the n job, m machine sequencing problem // Management Science. — 1970. — Vol. 16. — P. 630–637.
7. *Dannenbring D.G.* A evaluation of flow shop sequencing heuristics // Management Science. — 1977. — Vol. 23. — P. 1174–1182.
8. *Ho J.C., Chang Y.-L.* A new heuristic for the n -job, M -machine problem // European Journal of Operational Research. — 1991. — Vol. 52. — P. 194–202.
9. *Ishibuchi H., Misaki S., Tanaka H.*: Modived simulated annealing algorithms for the flow shop sequencing problem // European Journal of Operational Research. — 1995. — Vol. 81. — P. 388–398.
10. *Ogbu F.A., Smith D.K.* Simulated annealing for the permutation flow shop problem // OMEGA. — 1991. — P. 64–67.
11. *Nawaz M., Enscore E.E., Ham I.* A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem // OMEGA. — 1983. — Vol. 11. — P. 91–95.
12. *Brucker P.* Scheduling algorithms. — Berlin: Springer Verlag, 1995.
13. *Blazewicz J., Domschke W., Pesch E.* The job shop scheduling problem: Conventional and new solution techniques // European Journal of Operational Research. — 1996. — Vol. 93. — P. 1–33.
14. *Lageweg B.J., Lenstra J.K., Rinnooy Kann A.H.G.* A general bounding scheme for the permutation flow-shop problem // Operations Research. — 1978. — Vol. 26. — P. 53–67.
15. *Танаев В.С., Ковалёв М.Я., Шафранский Я.М.* Теория расписаний. Групповые технологии. — Минск: Ин-т техн. кибернетики НАН Беларуси, 1998. — 289 с.
16. *Танаев В.С., Сотсков Ю.Н., Струевич В.А.* Теория расписаний. Многостадийные системы. — М., URSS, 1989. — 328 с.
17. *Ogbu F.A., Smith D.K.* The application of the simulated annealing algorithm to the solution of the $n/m/C_{\max}$ flow shop problem // Computer & Operations Research. — 1990. — Vol. 17. — P. 243–253.
18. *Cleveland G.A., Smith S.F.* Using genetic algorithms to schedule flow shop releases // Proc. of the third international conference on genetic algorithms, Morgan Kaufmann, San Mateo, 1989. — P. 160–169.
19. *Reeves C.R.* A genetic algorithm for flow shop sequencing // Computer & Operations Research. — 1995. — Vol. 22. — P. 5–13.
20. *Palmer D.S.* Sequencing job through a multi-stage process in the minimum total time — a quick method of obtaining a near optimum // Operations Research Quarterly. — 1965. — Vol. 16. — P. 101–107.

Статья представлена к публикации членом редколлегии В.Н. Бурковым.

Зак Юрий Александрович — д-р техн. наук, пенсионер, последнее место работы — Европейский центр по мехатронике (European Centre for Mechatronics) г. Аахен, Германия, ☎ +49 (0) 241/543255, ✉ yuriy_zack@hotmail.com.

НОВАЯ КНИГА

Левин В.И. Логические методы в теории надежности: монография. — Пенза: Изд-во Пензенской гос. технол. академии, 2010. — 110 с.

Разработана математическая модель надежности сложных систем управления, пригодная для создания эффективных и экономичных методов количественного изучения надежности таких систем. Она представляет собой динамико-автоматную модель, описывающую надежностные процессы в системах в виде функций от надежностных процессов в элементах этих систем. Разработан математический аппарат непрерывной логики, позволяющий адекватно описывать созданную динамико-автоматную модель надежностных процессов в системах управления. Благодаря этому стали возможными практическое моделирование, расчет и анализ надежности указанных систем. Разработанный аппарат представляет собой обобщение булевой логики на непрерывный случай.

Книгу можно заказать по e-mail: f_os@pgta.ru