ИНФОРМАТИКА

2016 январь-март № 1

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 658.512.2

Г.М. Левин[1], Б.М. Розин[1], А.Б. Долгий[2]

ОПТИМИЗАЦИЯ АГРЕГИРОВАНИЯ И РЕЖИМОВ ПОСЛЕДОВАТЕЛЬНО-ПАРАЛЛЕЛЬНОГО ВЫПОЛНЕНИЯ ПЕРЕСЕКАЮЩИХСЯ МНОЖЕСТВ ОПЕРАЦИЙ

Предложены математическая модель и метод для задачи оптимизации агрегирования и режимов последовательно-параллельного выполнения комплекса пересекающихся множеств операций. Метод основан на двухуровневой схеме декомпозиции задачи. На верхнем уровне выбирается вариант агрегирования для заданных групп операций, на нижнем уровне оптимизируются режимы выполнения операций при фиксированном варианте агрегирования.

Введение

Задачам планирования процессов выполнения комплексов операций в системах различного назначения в последние десятилетия уделялось значительное внимание [1–12]. В ряде публикаций (в том числе в работах [9–12] авторов настоящей статьи) предлагаются модели и методы решения ряда задач, связанных с оптимизацией управления интенсивностью выполнения комплекса взаимосвязанных операций в предположении, что структура этого комплекса, а также структура реализующей его системы уже определены. Вместе с тем значительный научный и практический интерес представляет также разработка моделей и методов решения более сложных задач по совместной оптимизации как структуры комплекса, так и управления интенсивностями выполнения его компонентов. Под интенсивностью операции подразумевается некоторый параметр, определяющий время выполнения единицы ее объема.

В статье исследуется одна из задач совместной структурно-параметрической оптимизации при многократном последовательно-параллельном выполнении комплекса пересекающихся между собой множеств операций.

Следует отметить, что понятия «структура комплекса операций» (как и «структура реализующей его системы»), а также «управление интенсивностями (режимами) выполнения операций комплекса» могут иметь различные трактовки, причем в ряде случаев эти трактовки взаимосвязаны. В данной статье исследуется одна из ситуаций, когда структура комплекса определяется выбираемым вариантом агрегирования его операций в непересекающиеся группы (называемые в дальнейшем блоками), каждая из которых исполняется посредством своего устройства системы, причем все операции одного блока в любой момент времени выполняются с одной и той же интенсивностью. Подобная ситуация имеет место, в частности, при обработке деталей на многоинструментальных многопозиционных металлорежущих агрегатных станках, где несколько инструментов могут быть установлены в одной шпиндельной коробке (образуя единый блок технологических переходов) и иметь единую минутную подачу. Объединение операций в блоки, как правило, способствует снижению капитальных затрат на производственную систему и затрат на ее обслуживание, но одновременно приводит к возрастанию текущих материальных и временных затрат, связанных с выполнением каждой из операций комплекса в отдельности. Последнее объясняется тем, что объединение операций в блоки исключает возможность индивидуального выбора для каждой из них наилучших (с точки зрения текущих затрат) интенсивностей их выполнения.

Исследуемая задача заключается в определении варианта агрегирования операций и интенсивностей выполнения полученных блоков (включая единичные, т. е. содержащие только одну операцию), минимизирующего в совокупности стоимость выполнения комплекса работ при ограничении на общую длительность его выполнения. Рассмотренные в [9, 12] задачи являются фрагментами исследуемой, получаемыми при фиксации варианта агрегирования операций и некоторых дополнительных предположений. Наличие в исследуемой задаче комбинаторной составляющей, связанной с поиском наилучшего агрегирования операций, делает эту задачу достаточно сложной, требующей специальных методов решения. Ниже предлагается один из возможных подходов к разработке таких методов.

1. Постановка задачи и ее математическая модель

В производственной системе планируется последовательное выполнение (однократное либо циклически повторяющееся) всех работ комплекса, образованного множеством $I = \{1,\ldots, i,\ldots,n\}$ различных работ, причем этот комплекс включает n_i идентичных работ $i\in I$. Выполнение очередной работы $i\in I$ комплекса заключается в параллельном выполнении всех операций соответствующего подмножества J_i исходного множества $J = \{1, \ldots, m\}$ операций. Подмножества семейства $\mathbf{J}=\{J_1,\ldots, J_i,\ldots, J_n\}$, образующие работы комплекса, могут пересекаться и $\bigcup_{i=1}^{n} J_i = J$. Задано некоторое семейство $\mathbf{W}$ непересекающихся неединичных подмножеств множества J, каждое из которых является потенциальным блоком операций, причем любое подмножество $w\in\mathbf{W}$ может содержать операции из разных подмножеств семейства $\mathbf{J}$ (рис. 1). Подмножества $w\in\mathbf{W}$ выделены на рис. 1 штриховыми линиями.

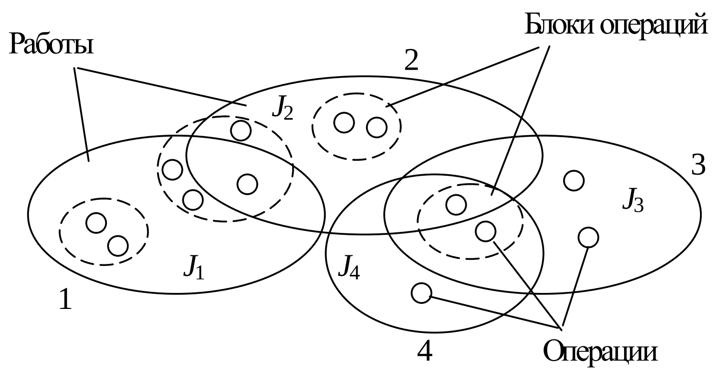

Рис. 1. Пример комплекса работ

Исследуется один из частных случаев возможного агрегирования операций, когда каждое из множеств операций $w\in\mathbf{W}$ может выполняться либо только полностью агрегированным (т. е. как один блок), либо полностью разагрегированным (т. е. каждая операция автономна). В этом случае множество возможных вариантов агрегирования представимо множеством Q всевозможных значений двоичного вектора $q = (q_w|w\in\mathbf{W})$, где $q_w= 0$ при агрегированном выполнении операций множества $w\in\mathbf{W}$ и $q_w= 1$ в противном случае.

Если операции из $w\in\mathbf{W}$ выполняются единым блоком (т. е. агрегированно), то их интенсивности одинаковы. При автономном выполнении этих операций каждая из них может иметь свою отличную от других операций из w интенсивность. В дальнейшем каждая автономно выполняемая операция также рассматривается как блок (единичный блок). Здесь и далее под интенсивностью s_j операции $j\in J$ подразумевается время, затрачиваемое на выполнение единицы ее объема. Для каждой операции $j\in J$ известны диапазон $S_j=[s_{1j}, s_{2j}]$ возможных значений ее интенсивности s_j и объем V_{ij} в содержащей ее работе $i\in I$. Естественно предположить, что для любого блока $w\in\mathbf{W}$ диапазон возможных значений интенсивностей выполнения его операций $S_w= [S_{1w}=\max\{s_{1j}|\ j\in w\},\ S_{2w}=\min\{s_{2j}|\ j\in w\}]\neq\varnothing$, поскольку в противном случае все операции из соответствующего w не могут быть агрегированы в одном блоке и, следовательно, это множество должно быть исключено из $\mathbf{W}$.

Ограничимся случаем, когда интенсивность выполнения каждой операции (а значит, и блока операций) выбирается однократно, не зависит от работы, в составе которой эта операция выполняется, а также не изменяется во времени. Таким образом, если $s=(s_j|\ j\in J)$ – вектор принимаемых значений интенсивностей операций $j\in J$ и $j(w)$ – некоторая фиксированная операция из $w\in\mathbf{W}$, то длительность $t_{ij}(s_j)$ операции j в составе работы i равна $V_{ij}s_j$, длительность $\underline{t}_{iw}(s_{j(w)})$ агрегированного выполнения блока w в составе работы i – $\max\{V_{ij}|j\in J_i\cap w\}s_{j(w)}$, а длительность $t_i(s)$ работы i – $\max\{t_{ij}(s_j)\mid j\in J_i\}$.

Анализ реальных ситуаций показывает, что для многих из них с достаточной для практики точностью можно считать, что общие материальные и временные затраты на выполнение каждой из работ комплекса складываются из двух основных частей. Первой является сумма затрат, связанных с восстановлением ресурсов, расходуемых при выполнении каждой из операций работы. Количество соответствующего ресурса, расходуемого на операцию $j \in J$, обычно зависит от работы i, в составе которой она выполняется, объема V_{ij} операции в этой работе и принятой интенсивности s_j операции. Максимальное количество каждого ресурса, которое может быть в системе, ограничено. Полное восстановление любого из ресурсов осуществляется после его расходования до допустимого уровня по завершении работы, в которой это произошло. Выполнение очередной работы может начаться лишь после завершения процесса восстановления соответствующего ресурса. Материальные и временные затраты на восстановление каждого из ресурсов, как правило, известны. В этом случае можно считать заданными определенные на S_j невозрастающие положительные функции $f_{1ij}(s_j)$ и $f_{2ij}(s_j)$, представляющие зависимости удельных (отнесенных к единице объема) соответственно материальных и временных затрат на выполнение операции j в составе работы i от принимаемой интенсивности s_j этой операции.

Вторую часть представляют материальные и временные затраты, связанные с амортизацией оборудования и его обслуживанием. Эти затраты можно считать пропорциональными длительности $t_i(s)$ работы i при принятых интенсивностях составляющих эту работу операций. Удельные (отнесенные к единице длительности работы) затраты этой категории зачастую различны для разных работ $i\in I$ и зависят от принимаемого варианта $q\in Q$ агрегирования операций, причем во многих реальных ситуациях достаточно предположить, что для всех работ $i\in I$ эти зависимости $C_{pi}(q)$ имеют следующую структуру:

$$C_{pi}(q) = E_{pi} + \sum_{w\in\mathbf{W}} e_{piw}q_w ,$$

где $p = 1$ или $p = 2$ при оценке материальных или временных затрат соответственно; $E_{pi}> 0$ – величина удельных затрат на работу $i\in I$ в случае агрегированного выполнения операций для всех множеств $w\in\mathbf{W}$; $e_{piw} > 0$ – дополнительные удельные затраты на работу $i\in I$ при автономном выполнении операций множества $w\in\mathbf{W}$. В дальнейшем параметры E_{pi} и e_{piw} предполагаются известными для всех $i\in I$, $w\in\mathbf{W}$ и $p = 1, 2$.

Пусть $S = \prod_{j\in J} S_j$; $q\in Q$ и $S(q)$ – множество таких векторов $s\in S$, что для любых $w\in\mathbf{W}$ и $j\in w$ значение $s_j = s_{j(w)}$ при $q_w = 0$. Заметим, что в силу оговоренного ранее $S(q)\neq\varnothing$ для любых $q\in Q$.

Согласно принятым предположениям о специфике рассматриваемого комплекса операций и его свойствах при фиксированных значениях векторов $q\in Q$ и $s\in S(q)$, представляющих вариант агрегирования операций и их интенсивности, общие средние материальные ($p = 1$) и временные ($p = 2$) затраты на однократное выполнение всего комплекса работ имеют следующий вид:

$$F_p(q,s) = \sum_{i\in I} n_i[(E_{pi} + \sum_{w\in\mathbf{W}} e_{piw}q_w)\max\{V_{ij}s_j \mid j\in J_i\} + \sum_{j\in J_i} V_{ij} f_{pij}(s_j)].$$

Пусть T^0 – максимально допустимое значение общих затрат времени на однократное выполнение всех работ комплекса, определяемое исходя из заданной производительности. Тогда исследуемая задача определения варианта агрегирования операций и их интенсивностей, минимизирующих общие материальные затраты на однократное выполнение всего комплекса операций при обеспечении требуемой производительности, сводится к следующей задаче смешанного нелинейного программирования:

$$F_1(q, s) \to \min; \tag{1}$$

$$F_2(q, s) \le T^0; \tag{2}$$

$$q \in Q; \tag{3}$$

$$s \in S(q). \tag{4}$$

В ряде публикаций (например, [2, 3, 6]) описываются производственные системы, планирование функционирования которых включает решение подобных задач.

2. Метод решения

Отметим следующие особенности задачи (1)–(4) (называемой далее задачей **A**), которые могут быть использованы при выборе подходов к ее решению:

– наличие двух групп разнородных переменных ($|\mathbf{W}|$-мерного двоичного вектора $q \in Q$ и m-мерного вектора s с компонентами, принимающими значения из заданных отрезков);

– одинаковая структура функций $F_1(q, s)$ и $F_2(q, s)$;

– если векторы q', $q'' \in Q$ таковы, что $q'_w \le q''_w$ для всех $w \in \mathbf{W}$, то $S(q') \subseteq S(q'')$ и $F_p(q', s) \le F_p(q'', s)$; если при этом $q' \ne q''$, то $F_p(q', s) < F_p(q'', s)$ для любых $s \in S(q')$ и $p = 1, 2$.

Для решения задачи **A** можно воспользоваться следующей двухуровневой схемой (рис. 2). На нижнем уровне решается подзадача **B**(q) нелинейного программирования по определению для фиксированного значения $q \in Q$ такого значения $s^*(q)$ вектора $s \in S(q)$, которое минимизирует функцию $F_1(q, s)$ при условии, что $F_2(q, s) \le T^0$. На верхнем уровне решается подзадача **C** дискретного программирования по поиску такого значения q^* вектора $q \in Q$, которое минимизирует функцию $\Phi(q) = F_1(q, s^*(q))$, при этом предполагается, что $\Phi(q) = \infty$, если подзадача **B**(q) не имеет решения.

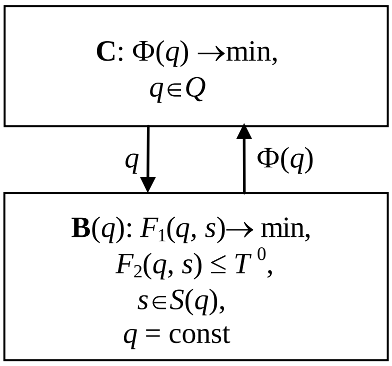

Рис. 2. Схема декомпозиции задачи **A**

Поскольку q^* и $s^*(q)$ являются решениями подзадач **C** и **B**(q) соответственно, то $(q^*, s^*) = (q^*, s^*(q^*))$ – решение исходной задачи **A**. Если же q^* и $s^*(q)$ являются лишь приближенными решениями соответствующих подзадач, то вектор (q^*, s^*) может быть принят в качестве приближенного решения задачи **A**.

Отметим некоторые из возможных подходов к решению выделенных подзадач. Для более детального рассмотрения подзадачи **B**(q) нижнего уровня введем некоторые дополнительные обозначения, опуская при этом фиксируемое в подзадаче значение вектора q.

Пусть $I_j = \{i \in I | j \in J_i\}$ – множество работ, содержащих операцию $j \in J$. Для данного фиксированного варианта агрегирования операций положим: $\mathbf{W}^0 = \{w \in \mathbf{W} | q_w = 0\}$ – множество реализуемых в этом варианте блоков; $\mathbf{W}_i^0 = \{w \in \mathbf{W}^0 | w \cap J_i \ne \varnothing\}$ – множество блоков, реализуемых в работе $i \in I$; $J^1 = \{j \in J | j \notin w \in \mathbf{W}^0\}$ – множество операций, реализуемых в этом варианте автономно; $J_i^1 = J_i \cap J^1$ – множество операций, реализуемых автономно в работе $i \in I$; $\tilde{E}_{pi} = n_i (E_{pi} + \sum_{w \in \mathbf{W} \setminus \mathbf{W}^0} e_{piw})$; $\bar{V}_{iw} = \max\{V_{ij} | j \in J_i \cap w\}$; $\tilde{f}_{pj}(s_j) = \sum_{i \in I_j} n_i V_{ij} f_{pij}(s_j)$; $\tilde{f}_{pw}(s_w) = \sum_{j \in w} \tilde{f}_{pj}(s_w)$, $w \in \mathbf{W}^0$; $\bar{s} = ((s_j | j \in J^1), (s_w | w \in \mathbf{W}^0))$ и $\bar{S} = \prod_{j \in J^1} S_j \times \prod_{w \in \mathbf{W}^0} S_w$. В этих обозначениях функции $F_p(q, s)$, $p = 1, 2$, представимы функциями

$$\tilde{F}_p(\bar{s}) = \sum_{i\in I}\tilde{E}_{pi}\max\{\max\{V_{ij}s_j \mid j\in J_i^1\},\quad \max\{\bar{V}_{iw}s_w \mid w\in \mathrm{W}_i^0\}\} + \sum_{j\in J^1}\tilde{f}_{pj}(s_j) + \sum_{w\in \mathbf{W}^0}\tilde{f}_{pw}(s_w),\ p = 1, 2,$$

и задача $\mathbf{B}(q)$ при фиксированном значении $q\in Q$ заключается в определении такого значения вектора $\bar{s}\in\bar{S}$, которое минимизирует функцию $\tilde{F}_1(\bar{s})$ при условии, что $\tilde{F}_2(\bar{s})\le T^0$.

Возможные подходы к решению задачи $\mathbf{B}(q)$ во многом зависят от свойств фукций $f_{pij}(\bullet)$ и соотношений объемов V_{ij} операций j в различных содержащих их работах $i\in I$ и потенциальных блоках операций $w\in\mathbf{W}$. Выделим следующих два распространенных на практике частных случая: а) когда функции $f_{pij}(s_j)$ выпуклы для всех p = 1, 2, $i\in I$ и $j\in J_i$ (задача $\mathbf{B}_1(q)$); б) когда для каждой операции $j\in J^1$ ее объемы V_{ij} одинаковы и равны V_j для всех работ $i\in I_j$ и когда для каждого блока $w\in\mathbf{W}^0$ значения $\bar{V}_{iw}$ одинаковы и равны $\bar{V}_w$ для всех таких работ $i\in I$, что $w\cap J_i\neq\varnothing$ (задача $\mathbf{B}_2(q)$). В остальных случаях разработка эффективных методов решения подзадачи $\mathbf{B}(q)$ требует дополнительных исследований.

В задаче $\mathbf{B}_1(q)$ при принятых предположениях функции $\tilde{f}_{pj}(s_j)$ и $\tilde{f}_{pw}(s_w)$ являются выпуклыми на отрезках S_j и S_w соответственно для всех $j\in J^1$ и $w\in\mathbf{W}^0$, и для ее решения применимы известные методы выпуклого программирования. В силу сепарабельности последних двух слагаемых в функциях $\tilde{F}_1(\bar{s})$ и $\tilde{F}_2(\bar{s})$ для решения задачи $\mathbf{B}_1(q)$ может быть адаптирован, в частности, подход, предложенный в [12] для решения аналогичной задачи и основанный на ее аппроксимации задачей линейного программирования.

Построим кусочно-линейные аппроксимации функций $\tilde{f}_{pj}(s_j)$ и $\tilde{f}_{pw}(s_w)$ на отрезках S_j и S_w соответственно, полагая

$$\tilde{f}_{pj}(s_j) \approx \max\{a_{pjk}s_j + b_{pjk} \mid k = 1, \ldots, r_{pj}\},\ p = 1, 2,\ j\in J; \tag{5}$$

$$\tilde{f}_{pw}(s_w) \approx \max\{c_{pwk}s_w + d_{pwk} \mid k = 1, \ldots, u_{pw}\},\ p = 1, 2,\ w\in\mathbf{W}, \tag{6}$$

где a_{pjk}, b_{pjk}, c_{pwk}, d_{pwk}, r_{pj} и u_{pw} являются параметрами этой аппроксимации, причем параметры r_{pj} и u_{pw} во многом определяют ее точность. Заметим, что параметры аппроксимации не зависят от значения вектора q и, следовательно, эти аппроксимации могут быть построены заранее до непосредственного решения исходной задачи **A** и использоваться в задачах $\mathbf{B}_1(q)$ при различных значениях вектора q.

Тогда приближенное решение задачи $\mathbf{B}_1(q)$ при фиксированном q может быть получено в результате решения следующей задачи линейного программирования:

$$\sum_{i\in I}\tilde{E}_{1i}t_i + \sum_{j\in J^1}y_{1j} + \sum_{w\in\mathbf{W}^0}z_{1w} \to \min; \tag{7}$$

$$\sum_{i\in I}\tilde{E}_{2i}t_i + \sum_{j\in J^1}y_{2j} + \sum_{w\in\mathbf{W}^0}z_{2w} \le T^0; \tag{8}$$

$$t_i - V_{ij}s_j \ge 0,\ i\in I,\ j\in J_i^1; \tag{9}$$

$$t_i - \bar{V}_{iw}s_w \ge 0,\ i\in I,\ w\in\mathbf{W}_i^0; \tag{10}$$

$$y_{pj} - a_{pjk}s_j \ge b_{pjk},\ p = 1, 2,\ j\in J^1,\ k = 1, \ldots, r_{pj}; \tag{11}$$

$$z_{pw} - c_{pwk}s_w \ge d_{pwk},\ p = 1, 2,\ w\in\mathbf{W}^0,\ k = 1, \ldots, u_{pw}; \tag{12}$$

$$s_j\in S_j,\ j\in J^1; \tag{13}$$

$$s_w\in S_w,\quad w\in\mathbf{W}^0. \tag{14}$$

Искомыми в этой задаче являются векторы $t=(t_i \mid i\in I)$, $y=(y_{pj} \mid p=1,2, j\in J^1)$, $z=(z_{pw} \mid p=1,2, w\in\mathbf{W}^0)$ и $\bar{s}\in\bar{S}$. Если $(t^*, y^*, z^*, \bar{s}^*)$ – ее решение, то вектор s с компонентами $s_j = s_w^*$ при $j\in w\in\mathbf{W}^0$ и $s_j = s_j^*$

при $j\in J^1$ может быть принят в качестве приближенного решения $s^*(q)$ задачи $\mathbf{B}_1(q)$. Несовпадение минимальных значений целевых функций этих задач определяется точностью аппроксимаций (5) – (6) функций $\tilde{f}_{pj}(s_j)$ и $\tilde{f}_{pw}(s_w)$ в окрестности решения исходной задачи $\mathbf{B}_1(q)$.

Для решения аппроксимирующих задач (7) – (14) при фиксированных значениях $q\in Q$ могут быть использованы существующие программные средства типа CPLEX.

В задаче $\mathbf{B}_2(q)$ в соответствии со сделанными предположениями функции $\tilde{F}_1(\bar{s})$, $\tilde{F}_2(\bar{s})$ имеют вид

$$\tilde{F}_p(\bar{s}) = \sum_{i\in I}\tilde{E}_{pi}\max\{\max\{V_j s_j \mid j\in J_i^1\}, \max\{\bar{V}_w s_w \mid w\in W_i^0\}\} + \sum_{j\in J^1}\tilde{f}_{pj}(s_j) + \sum_{w\in\mathbf{W}^0}\tilde{f}_{pw}(s_w),\ p = 1, 2.$$

Для ее решения может быть применен подход, аналогичный использованному в [9] для решения подобной задачи. Этот подход основан на комбинации методов множителей Лагранжа и динамического программирования.

Введем в рассмотрение функцию Лагранжа

$$L(\lambda,\bar{s}) = \lambda\tilde{F}_1(\bar{s}) + (1-\lambda)\tilde{F}_2(\bar{s}) = \sum_{i\in I}\tilde{\tilde{E}}_i(\lambda)\max\{\max\{V_j s_j \mid j\in J_i^1\},\ \max\{\bar{V}_w s_w \mid w\in \mathbf{W}_i^0\}\} +$$
$$+\sum_{j\in J^1}\varphi_{1j}(\lambda,s_j) + \sum_{w\in\mathbf{W}^0}\varphi_{2w}(\lambda,s_w)$$

с множителем $\lambda\in[0,1]$ и подзадачу $\mathbf{D}(\lambda)$ минимизации этой функции по $\bar{s}\in\bar{S}$ при фиксированном значении $\lambda\in[0,1]$. Здесь и далее параметр q по-прежнему опущен, $\tilde{\tilde{E}}_i(\lambda) = \lambda\tilde{E}_{1i} + (1-\lambda)\tilde{E}_{2i}$, $\varphi_{1j}(\lambda,s_j) = \lambda\tilde{f}_{1j}(s_j) + (1-\lambda)\tilde{f}_{2j}(s_j)$ и $\varphi_{2w}(\lambda,s_w) = \lambda\tilde{f}_{1w}(s_w) + (1-\lambda)\tilde{f}_{2w}(s_w)$.

Пусть $\bar{s}^*(\lambda)$ – решение этой задачи при некотором значении $\lambda\in[0,1]$ и $H(\lambda)=\tilde{F}_2(\bar{s}^*(\lambda))$. Очевидно, что если $H(0) > T^0$, то задача $\mathbf{B}_2(q)$ неразрешима (и, следовательно, соответствующее $\Phi(q) =\infty$), а если $H(1) \le T^0$, то $\bar{s}^*(1)$ – решение этой задачи. Поскольку функция $H(\lambda)$ не убывает на отрезке [0,1], то иначе решение задачи $\mathbf{B}_2(q)$ может быть получено по двухуровневой схеме. На нижнем уровне для фиксированного значения $\lambda\in(0,1)$ решается подзадача $\mathbf{D}(\lambda)$ по определению $\bar{s}^*(\lambda)$, на верхнем отыскивается наибольший корень $\lambda^*\in(0,\ 1)$ уравнения $H(\lambda)=T^0$ с монотонной левой частью (подзадача $\mathbf{L}$).

Если множество $\{(\tilde{F}_1(\bar{s}),\ \tilde{F}_2(\bar{s}))\mid \bar{s}\in\bar{S}\}$ эффективно выпукло (см., в частности, [13]), то $\bar{s}^*(\lambda^*)$ – точное решение задачи $\mathbf{B}_2(q)$, в противном случае $\bar{s}^*(\lambda^*)$ может быть принято в качестве ее приближенного решения.

Решение подзадачи $\mathbf{L}$ может быть получено известными методами, обычно используемыми для нахождения корней уравнений с монотонной левой частью, а решение задачи $\mathbf{D}(\lambda)$ при фиксированном λ может быть сведено к отысканию кратчайшего пути в некотором бесконтурном орграфе $\mathbf{G}$. Для более детального представления идеи этого сведения введем следующие определения и обозначения:

$M_1(I') = \bigcup_{i\in I'} J_i^1$, $M_2(I') = \bigcup_{i\in I'}\mathbf{W}_i^0$ и $M(I') = (M_1(I'), M_2(I'))$ для $I'\subseteq I$;

$\mathbf{\Omega}$ – семейство подмножеств $I'\subseteq I$, для которых не существует другого подмножества $I''\subseteq I$, что $I'\subset I''$ и $M(I') = M(I'')$. Предполагается, что $\mathbf{\Omega}$ включает также и пустое подмножество;

$\tilde{\Omega}$ – множество таких пар $(I^1, I^2)\in\mathbf{\Omega}\times\mathbf{\Omega}$, что $I^1\subset I^2$ и не существует такого $I^3\in\mathbf{\Omega}$, что $I^1\subset I^3\subset I^2$;

$N_k(I',I'') = M_k(I'')\backslash M_k(I')$, $k = 1, 2$, для $(I',I'')\in\tilde{\Omega}$;

$\tau_{1k}(J') = \max\{V_j s_{kj} | j\in J'\}$ и $\tau_{2k}(W') = \max\{\bar{V}_w S_{kw} | w\in W'\}$, $J'\subseteq J$, $W'\subseteq \mathbf{W}$ и $k = 1, 2$;

$s_j(\tau) = \min[s_{2j}, \tau/V_j]$, $j \in \mathbf{J}$, и $\underline{s}_w(\tau) = \min[S_{2w}, \tau/\overline{V}_w]$, $w \in \mathbf{W}$;

$$\psi(\lambda, I', J', W', \tau) = \sum_{i\in I'} \tilde{\tilde{E}}_i(\lambda)\tau + \sum_{j\in J'} \phi_{1j}(\lambda, s_j(\tau)) + \sum_{w\in W'} \phi_{2w}(\lambda, \underline{s}_w(\tau)),\ I' \subseteq I,\ J' \subseteq J,\ W' \subseteq \mathbf{W}$$

и $\tau \geq \max[\tau_{11}(J'), \tau_{21}(W')]$;

$\overline{\tau}(\lambda, I', J', W', \tau')$ – значение $\tau\in[\max\{\tau', \tau_{11}(J'), \tau_{21}(W')\}, \max\{\tau', \tau_{12}(J'), \tau_{22}(W')\}]$, которому соответствует наименьшее значение функции $\psi(\lambda, I', J', W', \tau)$, где $\tau' > 0$.

Очевидно, что $M_k(I') \subset M_k(I'')$ по крайней мере для одного из $k = 1, 2$ для любой пары $(I', I'')\in \tilde{\Omega}$.

Построение орграфа **G** возможно по следующей рекуррентной схеме:

1. Пара $(\varnothing,0)$ представляет начальную вершину этого орграфа.

2. Если вершина (I',τ') принадлежит **G**, то дуга $((I',\tau'),(I'',\tau''))$ также принадлежит **G**, если $(I',I'')\in \tilde{\Omega}$ и $\tau''\in[\max\{\tau',\tau_{11}(N_1(I',I'')),\tau_{21}(N_2(I',I''))\}, \overline{\tau}(\lambda,I''\backslash I',N_1(I',I''),N_2(I',I''),\tau')]$. Длина этой дуги принимается равной $\psi(\lambda,I''\backslash I', N_1(I',I''), N_2(I',I''), \tau'')$.

Следует заметить, что параметр τ' в парах (I',τ') представляет максимальное из предполагаемых времен выполнения работ из множества I'. Размерность орграфа **G** определяется как мощностью множества $\tilde{\Omega}$, так и длинами отрезков возможных значений параметра τ' в парах (I',τ') и принимаемой при построении орграфа дискретизацией этих отрезков.

Следуя [9], можно показать, что задача $\mathbf{D}(\lambda)$ сводится к нахождению в орграфе **G** кратчайшего пути из начальной вершины $(\varnothing,0)$ в множество конечных вершин вида $(I'=I,\tau)$. Если дуга $((I',\tau'),(I'',\tau''))$ принадлежит такому пути, то в решении $\overline{s}^*(\lambda)$ задачи $\mathbf{D}(\lambda)$ принимается значение $s_j^*(\lambda) = s_j(\tau'')$ для всех $j\in N_1(I',I'')$ и $s_w^*(\lambda) = s_w(\tau'')$ для всех $w\in N_2(I',I'')$.

Пусть $H(I',\tau')$ – длина кратчайшего пути в орграфе **G** из его начальной вершины $(\varnothing,0)$ в вершину (I',τ'). При построении орграфа **G** и поиске в нем кратчайшего пути в множество конечных вершин промежуточная вершина (I',τ') может быть исключена из дальнейшего рассмотрения, если определится другая такая вершина (I'',τ''), что $I' \subseteq I''$, $\tau' \geq \tau''$ и $H(I',\tau') \geq H(I'',\tau'')$.

В задаче **C** верхнего уровня число возможных значений вектора q равно $2^{|\mathbf{W}|}$, поэтому ее решение полным перебором всего множества Q требует значительных затрат времени даже при сравнительно небольших значениях $|\mathbf{W}|$ и практически нереализуемо при больших $|\mathbf{W}|$. Для сокращения перебора могут быть использованы известные методы, основанные на идеях случайного поиска, эвристиках и метаэвристиках в сочетании с предлагаемым ниже вариантом метода последовательной фиксации переменных (МПФП).

Пусть имеется некоторое значение q^0 вектора $q\in Q$, полученное с использованием перечисленных выше эвристических подходов. Для последующей оптимизации q^0 предлагается следующий алгоритм МПФП. Алгоритм сводится к пошаговой фиксации значений (0 либо 1) одной из незафиксированных на предыдущих шагах компонент вектора q^0. Число оцениваемых векторов $q\in Q$ на очередном шаге алгоритма последовательно сокращается от $|\mathbf{W}|$ до 1, при этом среди возможных вариантов выбирается единственный с наименьшим значением целевой функции.

Ниже для описания итерации алгоритма используются следующие обозначения:

Ψ – список блоков $w\in\mathbf{W}$ с фиксированными в дальнейшем значениями q_w для текущего вектора $q^{тек}$;

$\underline{q}(q,w)$ – вектор из множества Q, который отличается от вектора $q\in Q$ только компонентой q_w.

Итерация алгоритма МПФП:

1. Положим $q^{тек} = q^0$; $L_{тек} = \Phi(q^0)$, $\Psi = \varnothing$.

2. Если $\Psi = \mathbf{W}$, то переход к п. 4. Иначе переход к п. 3.

3. Посредством решения серии задач $\mathbf{B}(q)$ находится $w^* = \arg\min\{\Phi(\underline{q}(q,w))|w\in\mathbf{W}\backslash\Psi\}$. Если $L_{тек} \leq L^* = \Phi(\underline{q}(q,w^*))$, то переход к п. 4. Иначе полагается $L_{тек} = L^*$, $q^{тек} = \underline{q}(q,w^*)$, $\Psi = \Psi\cup\{w^*\}$ и переход к п. 2.

4. Вектор $q^{тек}$ принимается в качестве текущего решения q^0 задачи **C**. Конец итерации.

Итерация алгоритма МПФП может повторяться для текущего решения q^0, пока функция $\Phi(q^0)$ убывает. Число задач $\mathbf{B}(q)$, решаемых на каждой итерации алгоритма, не превышает $O(|\mathbf{W}|^2)$.

Одно из возможных направлений развития алгоритма МПФП связано с проверкой целесообразности изменения одновременно нескольких компонент вектора $q^{тек}$, отбираемых по результатам решения задач $\mathbf{B}(q)$ в п. 3.

Замечание. В некоторых случаях заранее известно, что оптимальные интенсивности $s_j^*(q_w)$ всех операций $j \in w$ некоторого блока $w \in \mathbf{W}$ одинаковы (например, когда операции $j \in w$ идентичны). Тогда для такого блока можно заранее положить $q_w^* = 0$ в решении q^* задачи **C**. Это свойство позволяет сократить размерность данной задачи.

Заключение

Предложены математическая модель и метод решения задачи комплексной оптимизации агрегирования в блоки и интенсивностей последовательно-параллельного выполнения комплекса пересекающихся множеств операций. Рассмотрены частные случаи задачи, когда интенсивность любой операции остается неизменной в составе любого множества, при этом зависимости материальных и временных затрат на выполнение любой операции в составе конкретного множества от ее интенсивности представимы выпуклыми функциями, либо объемы любой операции (блока операций) в составе различных включающих ее множеств одинаковы. Выбор степени агрегирования заранее определенных подмножеств операций осуществляется из двух крайних вариантов: либо все операции, входящие в состав подмножества, выполняются одним общим устройством и имеют общую интенсивность, либо интенсивности всех операций такого подмножества выбираются индивидуально.

Описана двухуровневая декомпозиционная схема решения рассматриваемой задачи. Метод приближенного решения задачи верхнего уровня основан на комбинации эвристических методов и методе последовательной фиксации переменных. Для подзадачи нижнего уровня поиска оптимальных интенсивностей операций при фиксированном варианте агрегирования предложены методы решения для двух частных случаев исходной задачи: комбинация методов Лагранжевой релаксации и динамического программирования для случая равных объемов любой операции (блока операций) в разных работах и метод аппроксимации задачей линейного программирования для случая выпуклых функций затрат на операции.

В качестве направления дальнейших исследований предполагается рассмотреть новые постановки задач планирования структуры управления и интенсивности выполнения комплексов операций, в которых:

– варианты агрегирования операций в составе заданных подмножеств операций определяются различными комбинациями подмножеств этих операций;

– интенсивности выполнения операций могут зависеть от работы, в составе которой эти операции выполняются;

– функции затрат на выполнение операций не являются выпуклыми и объемы операций (блоков операций) различны для разных работ.

Список литературы

1. Alting, L. Computer Aided Process Planning: the state-of-the-art survey / L. Alting, H. Zhang // Int. J. Prod. Res. – 1989. – Vol. 27, № 4. – P. 553–585.

2. Halevi, G. Process and Operation Panning / G. Halevi. – Springer, 2003. – 335 p.

3. Bukchin, J. Design of flexible assembly line to minimize equipment cost / J. Bukchin, M. Tzur // IIE Transactions. – 2000. – Vol. 32. – P. 585–598.

4. Gupta, A.K. Optimization of due-date objectives in scheduling semiconductor batch manufacturing / A.K. Gupta, A.I. Sivakumar // Intern. J. of Machine Tools and Manufacture. – 2006. – Vol. 46. – P. 1671–1679.

5. Burkov, V.N. Models and methods of multiprojects' management / V.N. Burkov, D.A. Novikov // Systems Science. – 1999. – Vol. 256, № 2. – P. 5–14.

6. Dolgui, A. Graph approach for optimal design of transfer machine with rotary table / A. Dolgui, N. Guschinsky, G. Levin // Intern. J. of Production Research. – 2009. – Vol. 47, № 2. – P. 321–341.

7. Dolgui, A. Enhanced mixed integer programming model for a transfer line design problem / A. Dolgui, N. Guschinsky, G. Levin // Computers and Industrial Engineering. – 2012. – Vol. 62, № 2. – P. 570–578.

8. Левин, Г.М. Оптимизация режимов параллельной многоинструментальной обработки деталей на агрегатном оборудовании с учетом групповой смены инструментов / Г.М. Левин, Б.М. Розин // Информатика. – 2011. – № 3. – С. 33–47.

9. Левин, Г.М. Оптимизация последовательно-параллельного выполнения комплекса взаимосвязанных операций / Г.М. Левин, Б.М. Розин // Весці НАН Беларусі. Сер. фіз.-мат. навук. – 2013. – № 1. – С. 111–116.

10. Levin, G. Optimization of Multi-tool Cutting Modes for Batch Manufacturing in Large Series Machining Environment / G. Levin, B. Rozin, A. Dolgui // Proc. of the 14th IFAC Symp. on Information Control Problems in Manufacturing (INCOM’12), Bucharest, Romania, May 23–25, 2012. – Elsevier Science, 2012. – P. 444–448.

11. Rozin, B. Optimization of Multi-tool Cutting Modes in Multi-item Batch Manufacturing System / B. Rozin, G. Levin, A. Dolgui // Proc. of the IFAC Conf. on Manufacturing Modelling, Management and Control (MIM'2013), Russia, June 19–21, 2013. – Elsevier Science, 2013. – P. 766–771.

12. Левин, Г.М. Линейная аппроксимация задачи оптимизации интенсивностей последовательно-параллельного выполнения пересекающихся множеств операций / Г.М. Левин, Б.М. Розин, А.Б. Долгий // Информатика. – 2014. – № 3. – С. 44–51.

13. Подиновский, В.В. Парето-оптимальные решения многокритериальных задач / В.В. Подиновский, В.Д. Ногин. – М. : Наука. Глав. ред. физ.-мат. лит-ры, 1982. – 256 с.

Поступила 24.11.2015

*[1]Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: {levin; rozin}@newman.bas-net.by*

*[2]Ecole Nationale Superieure des Mines, CNRS,
UMR6597 IRCCYN, F-44307 Nantes
Cedex 3, France
e-mail : alexandre.dolgui@mines-nantes.fr*

G.M. Levin, B.M. Rozin, A.B. Dolgui

OPTIMIZATION OF AGGREGATION AND SEQUENTIAL-PARALLEL EXECUTION MODES OF INTERSECTING OPERATION SETS

A mathematical model and a method for the problem of optimization of aggregation and of sequential-parallel execution modes of intersecting operation sets are proposed. The proposed method is based on the two-level decomposition scheme. At the top level the variant of aggregation for groups of operations is selected, and at the lower level the execution modes of operations are optimized for a fixed version of aggregation.