Оптимизация периодической обработки информации в специализированных устройствах. Часть 1[1]

В.М. Хачумов

Аннотация. Дается анализ проблемы планирования периодической обработки цифровой информации в устройствах с перестраиваемой структурой. Основное внимание уделяется разработке формализованных моделей вычислительного процесса с совмещением циклов обработки при ограничениях на число исполнительных ресурсов, исследованию эффективности предложенных алгоритмов. Статья, в силу большого объема, разбита на две части. В первой части дается обзор работ в области построения периодических расписаний, вводится модель и алгоритм совмещения циклов многократного выполнения алгоритма. Во второй части предлагается алгоритм оптимизации закрепления исполнительных блоков за фазами алгоритма, дается расширение задачи на случай комплексов алгоритмов с различными видами параллелизма, рассматриваются вопросы оптимизации структуры вычислителя.

Ключевые слова: периодические расписания, оптимизация совмещения циклов, процессорный элемент, специализированное вычислительное устройство.

Введение

Построение периодических расписаний для вычислительных систем имеет большую историю и продолжает оставаться в поле внимания исследователей в различных прикладных областях в настоящее время. Периодическая обработка часто имеет место в процессах реального времени, когда от вычислительного устройства требуется многократное выполнение заданного алгоритма в ответ на периодически поступающие внешние сигналы (заявки) и выдача результата объекту управления. Сюда можно отнести задачи непрерывного контроля и диагностики сложных технических устройств, реализации алгоритмов машинной графики для визуализации сцен, управления динамическими объектами, календарного планирования, обработки сигналов, распознавания речи и т.д.

Актуальным является решение двух задач теории расписаний: задачи построения эффективных алгоритмов оперативного планирования периодической обработки, минимизирующих среднее время однократного выполнения алгоритма или системы алгоритмов, и задачи оптимизации структуры специализированного вычислительного устройства (СВУ), реализу-

[1]Работа выполнена при финансовой поддержке проектов РФФИ № 16-07-00096 а «Методы решения навигационных и траекторных задач в бортовых системах интеллектуального управления автономных летательных аппаратов на основе оптимизации конвейерных, разрядных и параллельных вычислений», № 15-29-06945 офи_м «Развитие моделей, методов и программных средств обработки мультиспектральных снимков, видео-потоков и данных телеметрии для задач космического мониторинга арктической зоны», № 16-29-12839 офи-м «Разработка моделей, методов и инструментальных средств для синтеза оптимизированных технологических цепочек и технологических процессов на основе интегрированных баз знаний и интеллектуальных технологий автоматической генерации и оценки планов».

ющего заданный алгоритм циклической обработки в соответствии с требуемым быстродействием и ограничениями. Задачи периодической обработки информации можно классифицировать следующим образом:

– по типу расписания поступления заявок: с жестко заданными периодами, с возможностью подстройки величин периодов и моментов поступления заявок;

– по типу расписания выполнения заявок: с совмещением и без совмещения циклов обработки, с прерываниями и без прерываний;

– по наличию зависимости между операциями алгоритма обработки: с жестко заданным порядком следования операций, с независимыми операциями, с возможностью частичной перестановки операций;

– по виду параллелизма системы алгоритмов: параллелизм множества объектов, параллелизм смежных операций, параллелизм независимых ветвей, смешанный параллелизм;

– по типу используемых исполнительных ресурсов: функционально ориентированные, универсальные, унифицированные;

– по числу исполнительных ресурсов: единственный ресурс, ограниченное количество ресурсов, неограниченное количество ресурсов;

– по виду целевой функции: среднее время однократного выполнения алгоритма, время k-кратного выполнения, число совмещаемых циклов обработки, число исполнительных ресурсов, число межпроцессорных связей, надежность реализации, время запаздывания;

– по типу математических методов решения задач: комбинаторные методы, методы отсечения, приближенные методы, другие методы.

Актуальность решения задач планирования периодических расписаний требует дополнения существующих постановок за счет анализа новых методов и подходов к с учетом современных тенденций развития вычислительной техники и требований прикладных областей.

1. Подходы к планированию периодических расписаний

Существенный задел в области построения периодических расписаний был сделан в работах за 1960-1980-е гг. Обстоятельный обзор этих источников приведен, например, в монографии [1]. Наиболее полные результаты по организации периодической обработки информации в системе, содержащей один процессор, были получены в работах [2, 3]. При выполнении следующих условий: заявки на выполнение программы i-го типа $(i = 1,2....,n)$ поступают циклически с периодом T_i; длительность выполнения программ считается постоянной и равной t_i; выполнение каждой программы должно быть окончено не позднее момента поступления соответствующей заявки из того же потока; выполнение программ не может быть прервано. Вводится определение циклического расписания. Расписание поступления заявок (ПЗ-расписание) и расписание выполнения программ (ВП-расписание) называются циклическими с периодом T, если все их элементы могут быть представлены в виде: $\alpha_{im} = \alpha_{ij} + kT$, $\alpha_{im} \notin I$, $\alpha_{ij} \in I$ – для ПЗ-расписаний и $\beta_{im} = \beta_{ij} + kT$, $\beta_{im} \notin I$, $\beta_{ij} \in I$ для ВП расписаний. Здесь α_{ls}, β_{ls} – соответственно элементы ПЗ и ВП-расписаний, определяющие начала поступления и выполнения s-й заявки l-го потока; T является наименьшим общим кратным (НОК) всех T_i; k – целое, ненулевое; I – некоторый интервал времени длиной T. Вводится понятие согласованности. Расписание является согласованным, если в некоторый момент времени в систему одновременно поступают заявки от всех потоков. Одним из основных выводов работ [2, 3] является то, что для согласованного расписания достаточно построить допустимое расписание на отрезке времени $(\varphi, \varphi + T)$, начальной и конечной точками которого являются моменты согласования. Далее это расписание может быть распространено (скопировано) на временной шкале для получения циклического расписания.

Аналогичные результаты получены в работах [4, 5], в которых для построения допустимого расписания предлагается специальный алгоритм и доказывается его оптимальность.

Тип используемых процессоров существенно влияет на сложность решения задачи. Со-

гласно [6, 7] можно выделить следующие типы ПЭ: идентичные, независимые, унифицированные. Пусть t_{ij} – полное время, необходимое i-му ПЭ для выполнения j-й задачи. Процессорные элементы являются идентичными, если $t_{ij} = t_{kj}$ для всех i, j, k. Если i-й ПЭ имеет свою скорость s_{ij} выполнения j-й задачи, то ПЭ называются независимыми. В промежуточном случае процессоры называются унифицированными и t_{ij} может быть представлено в виде $t_{ij} = c_i t_j$, где c_i и t_j – параметры, связанные с i-м ПЭ и j-й задачей. В этом случае каждый ПЭ имеет свою скорость s_i выполнения, являющуюся одинаковой для всех задач. Задача составления расписания при $q > 1$ для процессорных элементов разного типа была наиболее полно исследована в работе [6]. Пусть в интервале $[\varphi_0, \varphi_0 + T]$ размещается N задач, где $N = \sum_{i=1}^{n} \frac{T}{T_i}$. Пусть также $\varphi_0, \varphi_1, \ldots, \varphi_k$, где $\varphi_0 \le \varphi_1 \le \ldots \le \varphi_k = \varphi_0 + T$, $k \le 2N+1$ – есть раздельные конечные точки интервалов, в которых размещаются задачи для вычисления, а $t_{ih}^{(k)}$ – время, в течение которого i-й ПЭ работает на h-ю задачу. Допустимое расписание существует, если и только если имеет решение следующая система:

$$\begin{aligned}
&\sum_{i=1}^{q}\sum_{k=0}^{k} s_{ih} t_{ih}^{(k)} = p_h && \text{для всех } h,\\
&\sum_{h=1}^{N} t_{ih}^{(k)} \le \varphi_{k+1} - \varphi_k && \text{для всех } i,k,\\
&\sum_{i=1}^{q} t_{ih}^{(k)} \le \varphi_{k+1} - \varphi_k && \text{для всех } h,k,\\
&t_{ih}^{(k)} \ge 0 && \text{для всех } h,i.k.
\end{aligned} \qquad (1)$$

Допустимое циклическое расписание строится следующим образом: находится решение системы (1); решение системы трансформируется в допустимое расписание для системы задач в интервале $[\varphi_0, \varphi_0 + T]$, что дает циклическое расписание с периодом T. Сложность решения системы составляет для идентичных ПЭ – $O(N^3)$, для унифицированных ПЭ – $O(qN^5)$. Сложность поиска расписания с использованием методики [7, 8] составляет $O(q^2N^3)$.

В приведенной выше серии работ структура реализуемых алгоритмов в явном виде не рассматривалась, вместе с тем, можно назвать исследования [9-16], в которых детально анализируется структура циклически реализуемых алгоритмов. В работах [17, 18] рассматривается задача многократной обработки алгоритма, содержащего M независимых блоков/операций, частично повторяющихся в вычислительной системе из Q идентичных ПЭ. Обработка i-го блока требует использования q_i ПЭ в течение временного промежутка времени t_i. При заданной кратности выполнения блоков требуется построить расписание, минимизирующее функцию $t = \sum_{j=1}^{J} x_j / c$, определяющую среднее время обработки алгоритма, где J – количество векторов состояния, определенных на множестве блоков, алгоритма, c – количество обработанных за время T алгоритмов, x_j – число повторений j-го вектора состояния за время T. Вектора состояния, определенные на множестве блоков алгоритма, формируются следующим образом. Каждому произвольному множеству A_j блоков алгоритма ставится в соответствие вектор $p_j = \{a_{1j}, \ldots, a_{Mj}\}$, элементы которого определяются как:

$$a_{ij} = \begin{cases} 1, \text{если } i\text{-ый блок принадлежит множеству } A_j \\ 0, \quad \text{в остальных случаях} \end{cases}.$$

Вектор p_j называется вектором состояния, если $\sum_{i=1}^{M} a_{ij} q_i \le Q$. Показано, что в случае, когда $t_i = const = 1$, нахождение минимума целевой функции t эквивалентно решению задачи линейного программирования с целевой функцией

$$t^{/} = \sum_{j=1}^{J} x_j^{/} \qquad (2)$$

и ограничениями

$$\sum_{j=1}^{J} a_{ij} x_j^{/} \geq cn_i \quad (i = 1,..., M), \qquad (3)$$

$$x_j^{/} \geq 0 \quad (j = 1,..., J). \qquad (4)$$

В связи с большой размерностью задачи (2)–(4) в работе [17] предлагается использовать только определенные «опорные» вектора состояний. При снятии ограничения $t_i = const$ задача становится эквивалентной бесконечномерной целочисленной задаче математического программирования с дробно-линейной целевой функцией [18]. В условиях логической или информационной зависимости операций алгоритма составление расписания значительно усложняется. В работе [9] предлагается решать такую задачу без совмещения разных циклов выполнения алгоритма. Сначала определяется минимальное время t_0 однократного выполнения алгоритма, представленного в виде графа следования. Тогда для выполнения k комплексов необходимо затратить время kt_0. В задаче Джонсона рассматривается оптимальное обслуживание партии заявок (работ) на двух приборах (фазах) [19, 20]. Пусть, например, необходимо выполнить n работ, каждая из которых проходит последовательно фазы обслуживания A и B. Обозначим через A_i, B_i время выполнения i-й $(i = 1,..., n)$ работы на фазах A и B соответственно. Оптимальное расписание таково, что работа i предшествует работе j, если $\min(A_i, B_j) < \min(A_j, B_i)$. При многократном (периодическом) выполнении алгоритма приборы обслуживают заявки в одной и той же последовательности. При числе обслуживающих приборов больше трех приходится использовать эвристические методы построения расписаний. Может быть применен метод последовательного конструирования, анализа и отсеивания вариантов, наиболее известной вычислительной схемой которого является метод ветвей и границ [10, 21].

Заслуживает особого внимания метод организации периодических расписаний с совмещением так называемых «примитивных» процессов [12, 13]. Здесь под примитивным процессом понимается последовательное (конвейерное) прохождение некоторым объектом системы состояний $a_1, a_2,..., a_n$, при этом определен вектор задержек $t_1, t_2,..., t_n$, связанных с пребыванием объекта в точках a_i, и вектор $\sigma_1, \sigma_2,..., \sigma_n$, определяющий время переноса. Если число операторов (ресурсов) обслуживания и операторов переноса не ограничено, то имеет место обычное конвейерное обслуживание примитивных процессов, когда каждый последующий процесс «запускается» на обслуживание через интервал времени $T = \max(t_i + \sigma_i)$. Однако, если число операторов обслуживания и/или число операторов переноса ограничено, возникает задача оптимизации процесса совмещения примитивных процессов. В работах [12, 13] ставится задача организации периодического процесса, совмещающего максимальное число примитивных процессов при заданном числе операторов переноса. Оптимальным считается циклический процесс, совмещающий p примитивных процессов, где p – длина максимальной цепи. Фазы примитивных процессов задаются элементами этой цепи. Это направление получило развитие в серии работ [14-16], где задача обобщается на случай совмещения нескольких независимых примитивных процессов. В работе [22] исследовались вопросы оптимизации совмещения различных циклов выполнения линейного алгоритма при ограничении на число специализированных ПЭ. Ограничение на число ПЭ приводит к многократному их использованию в процессе выполнения алгоритма, что требует выбора наилучшего варианта закрепления ПЭ за операциями алгоритма. Задача нахождения оптимального расписания, минимизирующего среднее время однократного выполнения алгоритма при заданном способе закрепления ПЭ, решалась путем полного перебора вариантов расписаний, что возможно только для сравнительно простых алгоритмов. Из других работ отметим [23], где рассматривается организация периодического процесса на параллельных процессорах при априорно известном распределении

операций реализуемого алгоритма. Периодическое расписание является рациональным способом организации вычислительного процесса, имеющего простое управление [24].

Перейдем к некоторым современным постановкам задачи периодической обработки информации. Среди работ последних лет отметим статью [25], в которой рассматривается проблема построения расписаний для систем, имеющих жесткие ограничения по времени работы. Это является одной из основных проблем при проектировании систем авионики. Разработан алгоритм, решающий задачу построения расписаний точно и находящий оптимальное расписание или близкое к оптимальному с использованием метода ограниченного перебора. Показано, что добавление требования строгой периодичности запуска процессов делает из полиномиально разрешимой задачи NP-сложную. В статьях [26, 27] дается следующее определение периодичности. Расписание для системы из n процессов называется периодическим с периодом α, если для каждого i-го процесса момент времени $t(i;k)$ начала k-й его итерации определен формулой $t(i;k) = t(i;0) + \alpha k$, где $1 \le i \le n$, k – целое. То есть, периоды всех процессов одинаковы и равны α. Систему с разными периодами отдельных процессов можно свести к циклической с одинаковыми периодами. Пусть имеется периодическая система из n задач $T = \{1,...,n\}$ с соответствующими периодами процессов $p_1,..., p_n$ и их начальными смещениями $s_1,...,s_n$. Тогда k-й запуск процесса i происходит в момент времени $s_i + (k-1)p_i$. Длина расписания без коллизий равна наименьшему общему кратному всех периодов процессов $L = HOK(p_1,..., p_n)$ и называется его первым циклом. Пусть эти процессы за весь первый цикл расписания запускаются соответственно $k_1,...,k_n$ раз, где $k_i = L / p_i$. Второй цикл расписания начинается непосредственно за первым. Тогда для каждого процесса его $(k_i + 1)$-й запуск (т.е. уже во втором цикле расписания) будет в момент времени $s_i + ((k_i + 1) - 1)p_i = s_i + k_i p_i = s_i + (L / p_i)p_i = s_i + L$.

В общем случае, периодическую систему реального времени можно формально определить следующим образом. Периодическая система $T = \{1,...,n\}$, состоящая из n процессов, задается множеством своих процессов T_i, каждый из которых определяется четверкой $T_i = (e_i, d_i, p_i, s_i)$, $1 \le i \le n$, где e_i – длительность i-го процесса (в тактах), т.е. время работы, взятое без прерываний; d_i – относительный предельный срок выполнения i-го процесса, то есть промежуток времени между очередными моментом времени готовности и предельным сроком выполнения; p_i – период i-го процесса, то есть промежуток времени между двумя последовательными моментами времени готовности; s_i – начальное смещение i-го процесса, то есть время начала первой итерации, первый момент времени готовности. Таким образом для k-го вызова i-го процесса заданы следующие временные ограничения:

$$s_i + (k-1)p_i \le t < s_i + (k-1)p_i + d_i, \qquad (5)$$

где k – натуральное число. Для этих параметров процессов выполняются следующие очевидные соотношения: $0 < e_i \le d_i \le p_i$, $1 \le i \le n$. Для смещения чаще всего выполняется условие $s_i \le d_i - e_i$. Для простоты все числовые параметры в (5) полагают целыми, поскольку для периодических систем реального времени случай расписаний с действительными параметрами может быть сведен к случаю с дискретными параметрами. Если задаются начальные смещения, то системы называются полными, в противном случае – неполными. Т.е., для неполных систем процессы задаются тройкой $T_i = (e_i, d_i, p_i)$, $1 \le i \le n$.

Расписание для однопроцессорной системы можно формально задать как числовую функцию S от натурального переменного t ($S : N \to \{0,...,n\}$), которая для каждого момента времени определяет, какой из n процессов работает, или возвращает нуль, если ни один из процессов не запущен в данный момент. Расписание $S(t)$ называется корректным, если для любого натурального k и $i \in \{1,...,n\}$

$S(t)=i$ и выполняется условие $S(s_i+(k-1)p_i)=i$ (то есть, момент запуска каждой итерации строго периодичен). В работах [28, 29] разработан алгоритм построения допустимых периодических расписаний в многопроцессорной системе реального времени для случая, когда запросы на выполнение заданий поступают с заданными периодами. Работы допускают прерывания и переключения с одного процессора на другой. Требования на выполнение работ поступают с заданными периодами. Рассматривается вычислительная система, состоящая из M параллельных процессоров L типов. Число процессоров j-го типа равно m_j $(j=1,2,...,L)$; $\sum_j^L m_j = M$. Производительность процессоров j-го типа равна v_j $(j=1,2,...,L)$; $v_1>v_2>...>v_L$. На вход системы поступает детерминированный поток требований на выполнение работ из множества $N=\{1,2,...,n\}$. Требования на обслуживание работы $i\in N$ поступают циклически с периодом $p_i>0$, начиная с момента времени $p_i^0\ge 0$; k-е требование поступает в момент времени $p_i^0+(k-1)p_i$, $k=1,2,...$ и для него установлен директивный срок $p_i^0+(k-1)p_i+f_i$, $(f_i\le p_i)$. Каждая работа может обслуживаться любым процессором. В фиксированный момент времени каждым процессором может выполняться не более одной работы, а каждая работа обслуживается не более чем одним процессором. Допускаются прерывания работ и переключения их с одного процессора на другой. Задан суммарный объем Q_i работы процессоров, необходимый для однократного исполнения каждой работы $i\in N$, т.е. если τ_{ij}^k – суммарное время обслуживания процессором j-го типа k-го требования работы i, то $Q_i=\sum_{j=1}^L v_j\tau_{ij}^k$. Требуется определить, существует ли допустимое расписание выполнения работ (т.е. расписание, при котором каждое требование работы i обслуживается в своем директивном интервале и суммарный объем работы процессоров по его обслуживанию составляет Q_i), и если оно существует, то найти его. Расписание задается в виде вектор-функции $R(t)=(R_1(t),R_2(t),...,R_n(t))$, где $R_i(t)=j$, если в момент времени t работа i обслуживается одним из процессоров j-го типа, и $R_i(t)=0$, если работа i в момент времени t не выполняется. Вводятся следующие обозначения:

P – НОК $(p_1,p_2,...,p_n)$; $p_0=\max\{p_i^0: i\in N\}$;

$B_m=(p_0+(m-1)P,\ \ p_0+mP\,|\,m=1,2....)$;

$U_{ih}^m(m=1,2,...;\ i\in N,\ h=0,1,...,r_i+1)$ – интервалы, образованные пересечением B_m с интервалами $P_{ki}(k=1,2,...)$. Если $p_0=p_i^0$ или $(p_0-p_i^0)$ кратно p_i, то полагаем $U_{i0}^m=\varnothing$.

Ω – множество расписаний $R^0(t)$ на B_1, для которых $\alpha_{ih}=Q_i$ при всех $i\in N$, $h=1,2,...r_i$ и $\alpha_{i0}+\alpha_{i(r_i+1)}=Q_i$ при всех $i\in N$, где $\alpha_{ih}=\sum_{j=1}^L v_j\tau_{ij}^h$, τ_{ij}^h – суммарная длина интервалов, на которых $R^0(t)\in j$ и которые лежат в U_{ih}^1. Если $\Omega\not\in\varnothing$, то для произвольного $R_i^0(t)\in\Omega$ можно определить допустимое периодическое расписание $R(t)$ с периодом P:

$$R_i(t)=\begin{cases}0, & при\ \ t\le p_i^0,\\ R_i^0(t+mP), & при\ \ t>p_i^0,\end{cases} \qquad (6)$$

где m – такое целое, что $t+mP\in B_1$, $i\in N$. Расписание (6) называется P- периодическим. Для этой задачи разработан полиномиальный алгоритм построения расписания.

Большая активность в решении задач периодической обработки информации наблюдается за рубежом. В работе [30] введен новый вид ограничений, показано, что они могут быть довольно легко добавлены к задаче линейного программирования для построения целого циклического расписания, которая является *NP*-полной. Моделью расписания служит бесконечное выполнение повторяющейся петли на графе. Работа направлена на построение периодических расписаний, таких, что каждая задача будет повторяться λ раз $\forall T_i\in T$, $s_i^q=s_i+(q-1)\lambda$, где $s_i=s_i^1$

есть время начала первого вхождения T_i, и λ – длина периода расписания. Таким образом, термин «периодическое расписание» используется, когда каждое задание повторяется каждые λ единиц времени, причем все задачи имеют одинаковый период.

Ограничения на ресурсы определяется следующим образом.

1. Имеется m типов ресурсов. Доступный ресурс типа $k \in \{1,...,m\}$ обозначается R_k.

2. Имеется набор T из n задач $\{T_i\}, 1 \le i \le n$ с целочисленными временами выполнения $\{p_i\}, 1 \le i \le n$. Каждое вхождение задачи T_i использует r_{ik} единиц (раз) ресурса типа $k \in \{1,...,m\}$ в процессе выполнения, где r_{ik} – целое число.

3. Если существует расписание, то для любой единицы времени t и любого ресурса k общее количество ресурсов, требуемое для всех задач T_i^q, таких, что $s_i^q \le t \le s_i^q + p_i$, не больше, чем R_k.

4. Проблема называется унитарной, если $r_{ik} \in \{0,1\}$.

В работе [31] рассмотрено периодическое расписание работы системы, состоящей из набора m - однотипных процессоров. Процессоры выполняют множество уровней обработки $L = \{L_1, L_2, ..., L_\alpha\}$. Уровень $L_i \in L$ определяется как $L_i(S_i, \omega_{l_i}, f_\varphi(l_i), \phi_i, D_i)$, где S_i – время старта (начала) уровня L_i, $\omega_i \in N^*$ – худшее (максимальное) время вычисления, $f_{STP}(i) \in N$ – худшее время передачи (связи) L_i при выполнении STP (Self-Timed Periodic)-расписания, $\phi_i \ge \omega_i + f_{STP}(i)$ есть граница периода, D_i – относительное время завершения уровня L_i, где $D_i = \max\limits_{k=1\to\beta_j} D_k$, $\beta_j \in N^*$ представляет число заявок (задач) на каждом уровне. Уровень L_i вызывается в момент времени $t = S_1 + (i-1)\phi$ и заканчивается до момента $t = S_1 + (i-1)\phi + \omega_i + f_\varphi(i)$. Если $D_i = \phi$, то L_i в таком случае имеется явный предельный срок выполнения (implicit-deadline). Если $D_i < \phi$, то L_i имеет условный предельный срок выполнения (constrained-deadline). Задачи распределяются по уровням, выполняются периодически с неявными сроками и самосинхронизируются (self-timed manner). Это возможно потому, что задачи уровня $k+1$ используют данные, полученные на уровне k.

В работе [32] предложено построение конвейера как модели вычислений для потоковых приложений, таких как обработка сигнала/изображения. Требования реального времени для таких приложений включают задания, сроки и величины задержки. Для этой задачи предложен эффективный метод периодического планирования с применением ациклических графов с отказом от некоторой эффективности решения ради простоты. Другую постановку задачи с привлечением графов можно найти в работе [33].

2. Построение периодических расписаний с совмещением циклов обработки для вычислительных устройств с динамическими связями

Дадим краткую характеристику алгоритмам быстрого построения периодических расписаний с оптимизацией совмещения циклов обработки для устройств с настраиваемой структурой. Здесь воспользуемся изложением результатов работ автора [34-37].

2.1. Основные понятия и определения

Выбор моделей вычислительного алгоритма, специализированного устройства, периодического вычислительного процесса. Пусть обработка цифровой информации производится в соответствии с некоторым алгоритмом A, содержащим m операций/операторов, среди которых могут быть одинаковые. Множество разных операций, выполняемых в A, обозначим $a = \{a_1, a_2, ..., a_l\}$, $1 \le l \le m$, при этом операция a_j в алгоритме A выполняется p_j раз, где $1 \le p_j \le m$, $\sum\limits_{j=1}^{l} p_j = m$. В простейшем случае

алгоритм обработки может быть представлен последовательностью m операций, в которой результат i-й операции служит входом для (i+1)-й операции. Таким образом, вход первой операции является входом алгоритма A, а результат последней операции – его выходом. Тогда алгоритм A можно представить в виде

$$A = a_{j_1}, a_{j_2}, ..., a_{j_m} \quad (7)$$

Запись (7) показывает, что на k-й фазе обработки информации выполняется операция a_{j_k}. Запись (7) есть логическая схема алгоритма (ЛСА) частного вида, в которой отсутствуют логические условия и опущены операторы начала и конца. Понятно, что такая запись является моделью некоторого множества алгоритмов, имеющих одинаковый строго определенный порядок следования операций, но отличающихся, в общем случае, видом информационных связей и структурой данных. Описание конкретного алгоритма обработки необходимо дополнить схемой информационных связей, устанавливающей связи операторов с множеством входных промежуточных и выходных переменных

Определение 2.1. Алгоритм (7), в котором строго определен порядок выполнения операций, назовем локальным алгоритмом (ЛА) обработки цифровой информации. Как будет показано позже на конкретных примерах, многие реальные алгоритмы периодической обработки могут быть представлены с помощью записи (7).

Понятно, что более сложные структуры алгоритмов обработки могут быть представлены композицией ЛА. Пусть модель вычислительного устройства представлена набором Q процессорных элементов $\{q_1, q_2, ..., q_l\}$ l-типов, функционально-специализированных в соответствии с набором операций ЛА. Здесь q_j представляет набор k_j одинаковых ПЭ, $q_j = \{q_{j1}, q_{j2}, ..., q_{jk_j}\}$ Пусть также t_j – время, необходимое для полного выполнения операции a_j процессорным элементом из q_j. Предполагается, что каждый ПЭ имеет свою управляемую память, в которую загружена микропрограмма выполнения операции, входные и выходной регистры для приема и выдачи операндов.

Переход от структуры алгоритма к структуре СВУ заключается в установлении соответствия между операциями ЛА и набором ПЭ. При этом каждой операции i-го типа ставится в соответствие единственный ПЭ из набора того же типа, а каждой информационной связи – физический канал связи. Если $k_j = p_j$, $\forall j \in \{1,2,...,l\}$, то имеется единственный способ закрепления ПЭ за операциями ЛА при однократном использовании каждого из них. Полученную таким образом структуру называют структурой с постоянными связями. Обычно $k_j \le p_j$ и существует множество возможных вариантов закрепления ПЭ, характеризуемых многократным использованием элементов. Подобные структуры имеют динамические связи и требуют введения коммутатора для переключения входов ПЭ.

Определение 2.2. ЛА назовем отмеченным локальным алгоритмом (ОЛА), если в (7) кроме выполняемых операций указаны закрепленные за ними ПЭ. ОЛА имеет следующий вид:

$$A^* = a_{j_1}^{l_1}, a_{j_2}^{l_2}, ..., a_{j_m}^{l_m} \quad (8)$$

Запись $a_{j_k}^{l_k}$ в (8) означает, что ПЭ с номером i_k из набора q_k выполняет операцию a_{j_k} на k-й фазе реализации ЛА. В общем случае одному ЛА может соответствовать некоторое множество ОЛА. Для организации вычислительного процесса необходимо указать момент начала выполнения ЛА. В этом случае при выполнении определенных условий ОЛА определяет расписание однократного выполнения алгоритма обработки, показывающее шаг за шагом, какая операция за какой следует и на каком ПЭ выполняется. При многократном выполнении алгоритма обработки информации в режиме совмещения вычислительных процессов предполагается выполнение следующих условий [11, 22].

Условия для построения расписания:

– выполнение ЛА не может быть прервано (расписание без прерываний);

– каждая операция ЛА выполняется одновременно не более чем на одном ПЭ;

– каждый ПЭ выполняет одновременно не более одного оператора ЛА;

– такт работы СВУ является составным и равен $\Delta t = t_{\text{вып}} + t_{\text{обм}}$, где $t_{\text{вып}} = \max\limits_{i=1,...,l}\{t_i\}$ – время выполнения самой длительной операции алгоритма, $t_{\text{обм}}$ – время межпроцессорного обмена.

Для простоты полагаем $\Delta t = 1$ и ограничения на число и вид межпроцессорных связей не накладываются. Перечисленные Условия не являются окончательными и будут уточняться в процессе изложения результатов работы. Однократное выполнение ОЛА назовем его циклом.

Пусть алгоритм (8) выполняется многократно, причем начала циклов совпадают с моментами $\varphi = \varphi_1, \varphi_2, ...;\ \varphi_1 \le \varphi_2 \le ...$ При многократной обработке, в дополнение к перечисленным условиям, будем оставлять неизменным порядок закрепления ПЭ за фазами ЛА, что является одним из необходимых требовании обработки без прерываний. Указанное расписание назовем циклическим с периодом T, если для всех его элементов выполняется соотношение $\varphi_j + kT = \varphi_{j+k}$. Будем говорить, что для циклического расписания с периодом T имеет место совмещение вычислительных процессов, если хотя бы для одного элемента расписания выполняется условие $\varphi_{j+1} < \varphi_j + t_0$, где t_0 – время в тактах, необходимое для выполнения одного цикла ОЛА. Процесс совмещенной обработки может быть представлен, таким образом, совокупностью взаимодействующих однократных процессов, описываемых ОЛА. Полученное расписание является допустимым, если выполняются все Условия. Качество расписания будем оценивать величиной среднего времени (в тактах), необходимого для получения одного результата обработки информации в установившемся режиме работы СВУ.

2.2. Оперативное планирование периодических расписаний для ЛА

Процесс построения расписания может быть представлен в виде следующей схемы:

1. генерация вариантов закрепления ПЭ за фазами выполнения ЛА (вариантов ОЛА);

2. отыскание для каждого ОЛА наилучшего расписания с совмещением циклов обработки;

3. выбор из множества локально лучших расписаний глобально лучшего.

Решение задачи построения оптимального расписания можно получить путем полного перебора всех возможных вариантов по указанной схеме. Однако в общем случае такой путь оказывается невозможным из-за большого числа вариантов [9]. В этой связи будем уделять основное внимание вопросам оперативного планирования периодических расписаний без существенного снижения их качества. Решение задачи начнем с разработки алгоритмов оптимизации совмещения циклов выполнения ОЛА [35, 36].

2.2.1. Оптимизация совмещения циклов выполнения ОЛА

Будем говорить, что для ОЛА (8) существует совмещение S_1 с шагом T_1, иначе $S_1 = S_1(T_1)$, $1 \le T_1 \le m$, если для $k > 1$ выполняется одно из условий:

1) $k + T_1 > m$,

2) $(j_{k+T_1} - j_k)^2 + (i_{k+T_1} - i_k)^2 > 0$.

В условии 2) первое выражение в скобках обращается в нуль при совпадении типов совмещаемых операций, а второе выражение – при совпадении номеров ПЭ. Условие нарушается если один и тот же ПЭ используется одновременно для выполнения двух операций.

Для $n > 1$ будем говорить, что для ОЛА существует совмещение $S_n = S_n(T_1,...,T_n)$, если существуют совмещения $S_{n-1}(T_1,...,T_{n-1})$, $S_{n-1}(T_2,...,T_n)$ и при этом выполняется одно из следующих условий:

1) $k + \sum\limits_{i=1}^{n} T_i > m$,

2) $(j_{k+\sum\limits_{i=1}^{n} T_i} - j_k)^2 + (i_{k+\sum\limits_{i=1}^{n} T_i} - i_k)^2 > 0$.

Совмещение $S_1(T_1)$ назовем максимальным, если не существует совмещения $S_1(T)$ с $T < T_1$. Для $n > 1$ совмещение $S_n(T_1,...,T_n)$ назовем максимальным, если максимально совмещение $S_{n-1}(T_1,...,T_{n-1})$ и не существует совмещения $S_n(T_1,...,T_{n-1},T)$ с $T < T_n$. Пусть для алгоритма (8) дано совмещение $S_n = S_n(T_1,...,T_n)$. Если

обозначить $t(S_n)=\sum_{i=1}^{n}T_i/n$, то очевидно $1\le t(S_n)\le m$. Если существует $\lim_{n\to\infty}t(S_n)=t(A^*)$, то нетрудно заметить, что $t(A^*)$ определяет среднее значение времени, необходимого для получения одного результата обработки в установившемся режиме работы СВУ, т.е. среднее значение времени однократного выполнения ОЛА. Так как величина $t(A^*)$ при фиксированном ОЛА зависит от выбора варианта совмещения, то задача отыскания $t(A^*)_{\min}$, соответствующего режиму работы СВУ с максимальной загрузкой, может быть решена путем полного перебора возможных вариантов, число которых, в общем случае, может быть значительным. Однако с достаточной степенью точности задачу отыскания $t(A^*)_{\min}$ можно заменить задачей нахождения максимального совмещения. Покажем, что если совмещения S_n являются максимальными, то существует $\lim_{n\to\infty}t(S_n)=t(A^*)$ и имеется простой алгоритм его нахождения. Формализуем постановку и решение задачи максимального совмещения циклов обработки ОЛА.

Для каждого ПЭ выпишем первоначально из заданного ОЛА порядковые номера (фазы) закрепленных за ними операций. Если φ_1 и φ_2 – две разные фазы закрепления ПЭ, то величину $\tau=|\varphi_2-\varphi_1|$ назовем сдвигом фаз. Таким образом, для всех ПЭ i-го типа можно выписать множество сдвигов фаз $H_i(A^*)$, а всему варианту ОЛА поставить в соответствие множество $H(A^*)=\bigcup_{i=1}^{l}H_i(A^*)=\{\tau_1,...,\tau_r\}$. Введем далее множество $G(A^*)$, такое, что $G(A^*)=N_p\setminus H(A^*)$, где $N_p=\{1,...,p+1\}$, $p=\max\{\tau:\tau\in H(A^*)\}$. Нетрудно убедиться, что множество сдвигов $H(A^*)$ определяет запрещенные шаги совмещения циклов обработки периодического расписания, а $G(A^*)$ – разрешенные шаги. Информация о множествах $H(A^*), G(A^*)$ и длине ЛА m служит формальной моделью ОЛА и является достаточной для построения допустимых периодических расписаний, в том числе оптимального.

Предлагается итеративный быстро сходящийся процесс построения периодического расписания, основанный на принципах максимального совмещения.

Пусть дано совмещение $S_n(T_1,...,T_{n-1},T)$. Для тех s, $1\le s\le k$, для которых $1+\sum_{i=k-s+1}^{k}T_i\le m$, введем обозначение $\gamma_n^s=1+\sum_{i=k-s+1}^{k}T_i$. Введем операцию $M\ominus\{q\}=\{x:x=y-q>0,\ y\in M\}$, где $M\subset N$, $q\in N$, N – множество натуральных чисел. Кроме того, пусть $m(M)=\min\{y:y\in M\}$. Введем вектор состояния i-го шага совмещения P_i. С учетом этого алгоритм построения периодического расписания сформулируем следующим образом.

Алгоритм 2.1 (максимального совмещения)

1. Пусть для выбранного варианта ОЛА. $H(A^*)=H_1$, $G_1=N_p\setminus H_1$, $T_1=m(G_1)$, $P_1=(1,\gamma_1^1)$. Тогда существует совмещение $S_1(T_1)$ и оно максимально.

2. $H_1^{/}=H_1\ominus\{T_1\}$, $H_2=H_1\cup H_{11}^{/}$, $G_2=N_p\setminus H_2$, $T_2=m(G_2)$, $P_2=(1,\gamma_2^1,\gamma_2^2)$. Тогда существует совмещение $S_2(T_1,T_2)$ и оно максимально.

n^0. Пусть $H_{n-1}^{/}=\bigcup_{k=1}^{n-1}(H_1\ominus\{\sum_{i=k}^{n-1}T_i\})$, $H_n=H_1\cup H_{n-1}^{/}$, $G_n=N_p\setminus H_n$, $T_n=m(G_n)$, $P_n=(1,\gamma_n^1,...,\gamma_n^n)$. Тогда существует совмещение $S_n(T_1,...,T_n)$ и оно максимально.

Процесс продолжается до такого n^0, для которого найдется такое k^0, $0\le k^0\le n^0$, что $P_{n^0}=P_{k^0}$. Тогда период T циклического расписания работы СВУ будет равен $T=\sum_{i=k^0}^{n^0-1}T_i$, откуда $t(A^*)=\frac{T}{n^0-k^0}$. Приведем доказательство максимальности совмещения, полученного с

помощью алгоритма 2.1, и доказательство существования периодического расписания.

Доказательство

Без ограничения общности можно считать, что $k_j = 1,\ j = 1,...,l$.

1. Если $T_1 + p \le m$, то $a_{T_1+p} \ne a_p$, $p = 1,...,(m - T_1)$, т.к. в противном случае $T_1 \in H_1$, т.е. $T_1 \notin G_1$, что противоречит выбору T_1. Покажем максимальность совмещения $S_1(T_1)$. Если существует совмещение $S_1(p_0)$, $1 \le p_0 \le m$, то $p_0 \notin H_1$ т.е. $p_0 \in G_1$, откуда $p_0 \ge T = m(G_1)$.

Предположим, что $S_{n-1}(T_1,...,T_{n-1})$ является максимальным совмещением. Покажем, что существует совмещение $S_n(T_1,...,T_n)$ и оно максимально. Сначала покажем, что для всех p и r, $1 \le r \le n$, для которых $T_n + \sum_{i=r}^{n} T_i \le m$, имеет место $a_p \ne a_{p+\sum_{i=r}^{n} T_i}$ или $T_n \notin H_n$.

Пусть $r = n$, тогда $T_n \notin H_1$, т.к. $T_n = m(G_n) = m(N_p \setminus (H_1 \cup H'_{n-1}))$. Далее пусть $r = n-1$, тогда, если $T_n + T_{n-1} \in H_i$, то $T_n \in H_i \oplus \{T_{n-1}\}$, что противоречит тому, что $T_n \notin H_n$. Аналогично делается проверка и для остальных r. Максимальность $S_n(T_1,...,T_n)$ очевидна.

2. Из определения чисел γ_k^s вытекает, что количество элементов в векторе P_n для любого $n \ge 0$ не превышает m. Из того, что $\gamma_k^s \le m$, $j_k \le l$, $i_k \le p_k \le m$ следует, что различных P_n может быть только конечное число и поэтому числа n^0 и k^0, о которых говорится в алгоритме, существуют. Легко проверить, что $T_{s+p_0} = T_s$ для $s \ge k^0$, что определяет существование периодического расписания.

Утверждения алгоритма доказаны.

Применение алгоритма 2.1 может приводить, в общем случае, к совмещениям циклов обработки ЛА с разными шагами. Если такой режим нежелателен, то следует перейти к расписанию, для которого совмещения выполняются с постоянным шагом.

Определение 2.3. Совмещение назовем равномерным с шагом T, если $T_i = const = T$. При равномерном совмещении шаг равен периоду циклического расписания.

Для равномерного совмещения минимальное T может быть найдено как

$$T = \min\{\tau : k\tau \notin H(A^*), \tau \in G(A^*), \forall k \in N\}. \quad (9)$$

В дальнейшем, если при построении циклического расписания имеет место соотношение $T \le t(A^*)$, где T – минимальный период для расписания с равномерным совмещением, а $t(A^*)$ – среднее время получения результата в расписании с максимальным совмещением, предпочтение будем отдавать расписанию с равномерным совмещением (9).

Из алгоритма 2.1 непосредственно вытекает следующее важное следствие.

Следствие 2.1. Если A_1^* и A_2^* – два варианта закрепления ПЭ за фазами ЛА, то $t(A_1^*) = t(A_2^*)$, если $H(A_1^*) = H(A_2^*)$.

Приведенные алгоритмы и следствие открывают возможность оперативного построения периодических расписаний с совмещением циклов.

Заключение

Составление периодических расписаний связано, как правило, с решением задач линейного целочисленного программирования большой размерности. Размерность зависит от числа и типа используемых ПЭ, модели вычислительного процесса, вида входного алгоритма и т.д. В статье большое внимание уделяется разработке методов поиска решений с привлечением элементов эвристики. Предложены простые модели периодического вычислительного процесса с совмещением циклов обработки для вычислительного устройства с динамическими связями.

Литература

1. Артамонов Е.И., Исмаилов Ш-М.А., Кокаев О.Г., Хачумов В.М. Специализированные алгоритмы и устройства обработки массивов данных. Махачкала: Дагестанское книжное издательство, 1993. 304 с.

2. Гильман А.Л., Хаит Я.Г. Расписания в задачах организации периодической обработки информации // Известия АН СССР. Техническая кибернетика. 1970. №3. С. 125-130.
3. Бейлин А.М., Гильман А.Л. Планирование периодической обработки информации в системе, работающей без прерываний // Изв. АН СССР. Техническая кибернетика. 1973. №3. С. 113-116.
4. Leung J.Y-T., Merril M.L. 1980. A note on preemptive scheduling of periodic real-time tasks. Information processing letters. 11(3):115–118.
5. Lui C.L., Layland J.W. 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the Association for Computing Machinery. 20(1):46–61. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.5086&rep=rep1&type=pdf (дата обращения: 15.02.2017).
6. Lawler E.L., Martel C.U. 1981. Scheduling periodically occurring tasks on multiple processors. Information processing letters. 12(1):9–12. URL: http://www.sciencedirect.com/science/article/pii/0020019081900661 (дата обращения: 15.02.2017).
7. Lawler E.L., Labetoulle J. 1978. On preemptive scheduling of unrelated parallel processors by linear programming. Journal of the ACM. 25(4):612–619. URL: http://www.columbia.edu/~cs2035/courses/ieor6400.F07/ll.pdf (дата обращения: 15.02.2017).
8. Gonzalez T., Sahni S. 1976. Open shop scheduling to minimize finish time. Journal of the ACM (JACM). 23(4):665–679. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.394.1507&rep=rep1&type=pdf (дата обращения: 15.02.2017).
9. Шурайц Ю.М. Оптимизация исследования ресурсов при многократном выполнении комплекса работ. Диссертация кандидата технических наук. М., 1979. 166 с.
10. Гудман С., Хидетниеми С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. 366 с.
11. Конвей Р.В., Максвелл В.А., Миллер Л.Б. Теория расписаний. М.: Наука, 1975. 360 с.
12. Айзенштат В.С. Совмещение примитивных циклических процессов // Доклады АН БССР. 1963. Т.VII. №3. С. 148-151.
13. Айзенштат В.С. Многооператорные циклические процессы // Доклады АН БССР. 1963. Т.VII. №4. С. 224-227.
14. Метельский А.С. О некоторых периодических процессах // Доклады АН БССР. 1963. Т.VII. №9. С. 584-587.
15. Метельский А.С. К системе периодических процессов // Доклады АН БССР. 1963. Т.VII. №11. С. 724-728.
16. Метельский А.С. Периодические процессы // Доклады АН БССР. 1965. Т.IX. №12. С. 788-790.
17. Бронштейн И.И., Трахтенгерц Э.А., Шурайц Ю.М. Определение оптимального плана реализации алгоритма на многопроцессорной вычислительной машине // Автоматика и телемеханика. 1975. №4. С. 122-127.
18. Шурайц Ю.М. Оптимальное распределение ресурсов при многократном выполнении комплекса работ. Актуальные вопросы теории и практики управления. М. Наука, 1977. С. 122-125.
19. Джонсон С.М. Оптимальные двух- и трехоперационные календарные планы производства с учетом подготовительно-заключительного времени. Календарное планирование. М.: Прогресс, 1966. С. 33-41.
20. Layland J.W. 1975. Development of real-time hardware/software systems. The deep space network progress report. 42-28:57–66.
21. Танаев В.С., Шкурба В.Б. Введение в теорию расписаний. М.: Наука, 1975. 256 с.
22. Федорович О.Е., Горлова В.И. Перечисление возможных вариантов состава модульных структур с учетом фаз обработки информации. Теория автоматизированного проектирования. Межвуз. темат. сборник научн. трудов. Харьков, 1980. Вып.2. С. 92-97.
23. Reiter R. 1968. Sheduling parallel computations. Journal of the Association for Computing Machinery. 15(4):590–599.
24. Солих Р. Задача календарного планирования для циклически повторяющегося производства // Журнал вычислительной математики и математической физики. 1973. Т.13. №2. С. 326-342.
25. Третьяков А. Автоматизация построения расписаний для периодических систем реального времени. Труды Института системного программирования РАН, 2012. Т.22. С. 375-400. URL: http://www.ispras.ru/proceedings/docs/2012/22/isp_22_2012_375.pdf (дата обращения: 15.02.2017).
26. Brucker P., Kampmeyer T. 2005. Tabu search algorithms for cyclic machine scheduling problems. Journal of Scheduling. 8:303–322.
27. Brucker P., Kampmeyer T. 2008. A general model for cyclic machine scheduling problems. Discrete Applied Mathematics. 156(13):2561–2572.
28. Фуругян М.Г. Построение периодических расписаний в многопроцессорных системах реального времени. Материалы 7 международной конференции «Управление развитием крупномасштабных систем». ИПУ РАН, Москва, 2013. С. 444-446.
29. Фуругян М.Г. Построение периодических расписаний в многопроцессорных АСУ реального времени//Автоматика и телемеханика. 2000. №9. С. 169-172. URL: http://www.mathnet.ru/links/8e4787617aa0bc93619b6f743ac90985/at364.pdf (дата обращения: 15.02.2017).
30. Hanzalek Z., Hanen C. 2015. The impact of core precedences in a cyclic RCPSP with precedence delays. Journal of Scheduling. 18(3):275–284. URL: http://link.springer.com/article/10.1007/s10951-014-0399-4 (дата обращения: 15.02.2017).
31. Dkhil A., Do X.K., Dubrulle P., Louise S., Rochange C. 2014. Self-timed periodic scheduling for cyclo-static dataflow model. In: Workshop on Architecture, Languages, Compilation and Hardware support for Emerging ManY-core systems. ALCHEMY 2014, 10 June 2014 – 12 June 2014 (Cairns, Australia). 18 p. URL: http://oatao.univ-toulouse.fr/12956/1/Dkhil_12956.pdf (дата обращения: 15.02.2017).
32. Tendulkar P., Poplavko P., Maler O. 2014. Strictly periodic scheduling of acyclic synchronous dataflow graphs using SMT solvers. Verimag Research Report no TR-2014-5 01-May-2014. URL: http://www-verimag.imag.fr/TR/TR-2014-5.pdf (дата обращения: 15.02.2017).

33. Periodic Scheduling – CMSC 451: Design and Analysis of Algorithms Spring 2015. URL: http://www.cs.umd.edu/class/spring2015/cmsc451/periodicsch.pdf (дата обращения: 15.02.2017).
34. Хачумов В.М. Периодические расписания с совмещением циклов обработки данных. Труды Первой Всероссийской научной конференции «Методы и средства обработки информации». М.: МГУ, 2003. С. 522-527.
35. Хачумов В.М., Юмагулов М.Г. Метод анализа и синтеза оптимальных структур вычислительных устройств для систем управления. В сб.: Автоматизация проектирования и конструирования. Тезисы докладов II Всесоюзного совещания. Часть 2. М.: Институт проблем управления, 1983. С. 106-109.
36. Хачумов В.М., Юмагулов М.Г. Об оптимизации метода совмещения вычислительных процессов в специализированных устройствах. В кн.: Управление в сложных нелинейных системах. М.: Наука, 1984. С. 148-152.
37. Хачумов В.М. Модели конвейерного медицинского технологического процесса // Искусственный интеллект и принятие решений. 2009. №3. С. 25-32.

Хачумов Вячеслав Михайлович. Заведующий лабораторией ИСА ФИЦ ИУ РАН. Окончил Ленинградский кораблестроительный институт в 1971 году. Доктор технических наук, профессор. Автор более 200 печатных работ. Область научных интересов: вычислительные алгоритмы, распознавание образов, методы обработки сигналов и изображений, искусственный интеллект. E-mail: vmh48@mail.ru

Optimization of periodic information processing in specialized devices. Part 1

V.M. Khachumov

Abstract. The analysis of a problem of planning of periodic digital information processing in devices with tunable structure is given. The main attention is paid to development of the formalized models of calculating process with combination of operation cycles with restrictions on number of the executive resources, to a research of offered algorithms efficiency. Article, owing to large volume, is broken into two parts. In the first part the review of operations in plot area of the work in constructing periodic schedules is given, the model and an algorithm of combination of cycles of repeated algorithm execution is entered. In the second part the algorithm of optimization of fixing of the executive units to algorithm phases is offered, extension of the task on a case of complexes of the algorithms responding to different types of parallelism is given questions of calculator structure optimization are considered.
Keywords: periodic schedules, optimization of combination of cycles, processor element, specialized computing device.

References

1. Artamonov, E.I., Sh-M.A.Ismailov, O.G.Kokaev, and V.M. Khachumov, 1993. Specializirovannye algoritmy i ustrojstva obrabotki massivov dannyh [Specialized algorithms and devices for processing array data]. Mahachkala: Dagestanskoe knizhnoe izdatel'stvo [Makhachkala, Dagestan book publishing house]. 304 p.
2. Gilman, A.L., and Ja.G. Hait, 1970. Raspisanija v zadachah organizacii periodicheskoj obrabotki informacii, 1970. [Schedules the tasks of the organization of the periodic information processing]. Izv. AN SSSR. Tehnicheskaja kibernetika [Proceedings of the USSR Academy of Sciences. Technical Cybernetics]. 3:125-130.
3. Bejlin, A.M., and A.L. Gilman. 1973. Planirovanie periodicheskoj obrabotki informacii v sisteme, rabotajushhej bez preryvanij [The scheduling information processing in the system running without interruption]. Izv. AN SSSR. Tehnicheskaja kibernetika [Proceedings of the USSR Academy of Sciences. Technical Cybernetics]. 3:113-116.
4. Leung, J.Y-T., and M.L. Merril. 1980. A note on preemptive scheduling of periodic real-time tasks. Information processing letters. 11(3):115–118.
5. Lui, C.L., and J.W. Layland. 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the Association for Computing Machinery. 20(1):46–61. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.5086&rep=rep1&type=pdf (accessed February 15, 2017).
6. Lawler, E.L., and C.U. Martel. 1981. Scheduling periodically occurring tasks on multiple processors. Information processing letters. 12(1):9–12. Available at: http://www.sciencedirect.com/science/article/pii/0020019081900661 (accessed February 15, 2017).
7. Lawler, E.L., and J. Labetoulle. 1978. On preemptive scheduling of unrelated parallel processors by linear programming. Journal of the ACM. 25(4):612–619. Available at: http://www.columbia.edu/~cs2035/courses/ieor6400.F07/ll.pdf (accessed February 15, 2017).

8. Gonzalez, T., and S. Sahni. 1976. Open shop scheduling to minimize finish time. Journal of the ACM (JACM). 23(4):665–679. Available at:: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.394.1507&rep=rep1&type=pdf (accessed February 15, 2017).
9. Shurajc, Ju.M. 1979. Optimizacija issledovanija resursov pri mnogokratnom vypolnenii kompleksa rabot. Dis... kand.tehn.nauk. [Optimization studies with multiple resources performing the work. Thesis of candidate of technical Sciences]. Moscow, 166 p.
10. Gudman, S., and S. Hidetniemi. 1981. Vvedenie v razrabotku i analiz algoritmov [Introduction to the design and analysis of algorithms]. Moscow: Mir, 366 p.
11. Konvej, R.V., V.A. Maksvell, and L.B. Miller. 1975. Teorija raspisanij [The scheduling theory]. Moscow: Nauka, 360 p.
12. Ajzenshtat, B.C. 1963. Sovmeshhenie primitivnyh ciklicheskih processov [The combination of primitive cyclic processes]// Doklady AN BSSR [Reports of Academy of Sciences of the Byelorussian SSR]. VII. 3: 148-151.
13. Ajzenshtat, B.C. 1963. Mnogooperatornye ciklicheskie process [Multi-statement cyclic processes]. Doklady AN BSSR [Reports of Academy of Sciences of the Byelorussian SSR]. VII. 4: 224-227.
14. Metelskij, A.S. 1963. O nekotoryh periodicheskih processah [Some periodic processes]. Doklady AN BSSR [Reports of Academy of Sciences of the Byelorussian SSR]. VII. 9: 584-587.
15. Metelskij, A.S. 1963. K sisteme periodicheskih processov [To the system of periodic processes]. Doklady AN BSSR [Reports of Academy of Sciences of the Byelorussian SSR]. VII. 11: 724-728.
16. Metelskij, A.S. 1965. Periodicheskie processy [Periodic processes]. Doklady AN BSSR [Reports of Academy of Sciences of the Byelorussian SSR]. IX. 12: 788-790.
17. Bronshtejn, I.I., Je. A., Trahtengerc, and Ju. M. Shurajc. 1975. Opredelenie optimal'nogo plana realizacii algoritma na mnogoprocessornoj vychislitel'noj mashine [Determining the optimal plan for the implementation of the algorithm on a multiprocessor computer]. Avtomatika i telemehanika [Automation and remote control]. 4:122-127.
18. Shurajc, Ju.M. 1977. Optimalnoe raspredelenie resursov pri mnogokratnom vypolnenii kompleksa rabot [Optimal resource allocation in case of multiple execution of complex of works]. Aktualnye voprosy teorii i praktiki upravlenija [Relevant issues of the theory and practice of control]. Moscow: Nauka: 122-125.
19. Dzhonson, S.M. 1966. Optimal'nye dvuh- i trehoperacionnye kalendarnye plany proizvodstva s uchetom podgotovitel'no-zakljuchitel'nogo vremeni. Kalendarnoe planirovanie [Optimal two - and trihomonatidne calendar plans of production, given the preparatory-final time. Scheduling]. Moscow: Progress: 33-41.
20. Layland, J.W. 1975. Development of real-time hardware/software systems. The deep space network progress report. 42-28:57–66.
21. Tanaev, B.C., and V.B. Shkurba. 1975. Vvedenie v teoriju raspisanij [Introduction to scheduling theory]. Moscow: Nauka. 256 p.
22. Fedorovich, O.E., and V.I. Gorlova. 1980. Perechislenie vozmozhnyh variantov sostava modul'nyh struktur s uchetom faz obrabotki informacii. Teorija avtomatizirovannogo proektirovanija [Enumeration of the possible variants of modular structures, taking into account the phases of information processing. Theory of computer-aided design]. Mezhvuz. temat. sbornik nauchn. Trudov [Interuniversity theme. collection of scientific works]. Kharkov. 2:92-97.
23. Reiter, R. 1968. Sheduling parallel computations. Journal of the Association for Computing Machinery. 15(4):590–599.
24. Solih, R. 1973. Zadacha kalendarnogo planirovanija dlja ciklicheski povtorjajushhegosja proizvodstva [Task scheduling for cyclic production]. Zhurnal vychislitel'noj matematiki i matematicheskoj fiziki [Computational mathematics and mathematical physics]. 13(2):326-342.
25. Tret'jakov, A. 2012. Avtomatizacija postroenija raspisanij dlja periodicheskih sistem real'nogo vremeni [Automation of scheduling for periodic real-time systems]. Trudy Instituta sistemnogo programmirovanija RAN [Proceedings of Institute of system programming of RAS]. 22:375-400. Available at: http://www.ispras.ru/proceedings/docs/2012/22/isp_22_2012_375.pdf (accessed February 15, 2017).
26. Brucker, P., and T. Kampmeyer. 2005. Tabu search algorithms for cyclic machine scheduling problems. Journal of Scheduling. 8:303–322.
27. Brucker, P., and T. Kampmeyer. 2008. A general model for cyclic machine scheduling problems. Discrete Applied Mathematics. 156(13):2561–2572.
28. Furugjan, M.G. 2013. Postroenie periodicheskih raspisanij v mnogoprocessornyh sistemah real'nogo vremeni [The construction of periodic schedules in multiprocessor real-time automation]. Materialy 7 mezhdunarodnoj konferencii «Upravlenie razvitiem krupnomasshtabnyh sistem» [The materials of the 7 international conference "Managing the development of large-scale systems"]. IPU RAN, Moskva [Institute of control problems of RAS, Moscow]. 444-446.
29. Furugjan, M.G. 2000. Postroenie periodicheskih raspisanij v mnogoprocessornyh ASU realnogo vremeni [The construction of periodic schedules in multiprocessor real-time automation]. Avtomatika i telemehanika [Automation and remote control]. 9:169-172. Available at: http://www.mathnet.ru/links/8e4787617aa0bc93619b6f743ac90985/at364.pdf (accessed February 15, 2017).
30. Hanzalek, Z., and C. Hanen. 2015. The impact of core precedences in a cyclic RCPSP with precedence delays. Journal of Scheduling. 18(3):275–284. Available at: http://link.springer.com/article/10.1007/s10951-014-0399-4 (accessed February 15, 2017).
31. Dkhil, A., Do X.K., P. Dubrulle, S. Louise, and C. Rochange. 2014. Self-timed periodic scheduling for cyclo-static dataflow model. In: Workshop on Architecture, Languages, Compilation and Hardware support for Emerging ManYcore systems.

ALCHEMY 2014, 10 June 2014 – 12 June 2014 (Cairns, Australia). 18 p. Available at: http://oatao.univ-toulouse.fr/12956/1/Dkhil_12956.pdf (accessed February 15, 2017).
32. Tendulkar, P., P. Poplavko, and O. Maler. 2014. Strictly periodic scheduling of acyclic synchronous dataflow graphs using SMT solvers. Verimag Research Report no TR-2014-5 01-May-2014. Available at: http://www-verimag.imag.fr/TR/TR-2014-5.pdf (accessed February 15, 2017).
33. Periodic Scheduling – CMSC 451: Design and Analysis of Algorithms Spring 2015. Available at: http://www.cs.umd.edu/class/spring2015/cmsc451/periodicsch.pdf (accessed February 15, 2017).
34. Khachumov, V.M. 2003. Periodicheskie raspisanija s sovmeshheniem ciklov obrabotki dannyh [Periodic schedules with combining cycles of data processing]. Trudy Pervoj Vserossijskoj nauchnoj konferencii «Metody i sredstva obrabotki informacii» [Proceedings of the First all-Russian scientific conference "Methods and means of information processing"]. Moscow: MGU. 522-527.
35. Khachumov, V.M., and M.G. Jumagulov. 1983. Metod analiza i sinteza optimal'nyh struktur vychislitel'nyh ustrojstv dlja sistem upravlenija [The method of analysis and optimal synthesis of structures of computing devices for control systems]. V sb.: Avtomatizacija proektirovanija i konstruirovanija. Tezisy dokladov II Vsesojuznogo soveshhanija. Chast' 2 [In proceedings: automation of design and construction. Abstracts of the II all-Union meeting. Part 2]. Moscow: Institute of control problems, RAS. 106-109.
36. Khachumov, V.M., and M.G. Jumagulov. 1984. Ob optimizacii metoda sovmeshhenija vychislitel'nyh processov v special-izirovannyh ustrojstvah [About optimization of the method of combining computational processes in specialized devices]. V kn.: Upravlenie v slozhnyh nelinejnyh sistemah [In the book: Management in complex nonlinear systems]. Moscow: Nauka. 148-152.
37. Khachumov, V.M. 2009. Modeli konvejernogo medicinskogo tehnologicheskogo processa [Conveyor model of medical technological process]. Iskusstvennyj intellekt i prinjatie reshenij [Artificial intelligence and decision making]. 3: 25-32.

Khachumov Vyacheslav Mikhailovich. Head of the Laboratory “Intelligent Control Methods” of the Institute for System Analysis, Federal Research Centre “Information and Control” of the Russian Academy of Sciences. He graduated from the Leningrad Shipbuilding Institute in 1971. Doctor of Technical Sciences, Professor. Author of over 200 publications. Research interests: computational algorithms, pattern recognition, signal and image processing, artificial intelligence.
E-mail: vmh48@mail.ru