

1. Закревский А.Д., Балаклея Л.И., Елисеева Н.А. и др. Синтез асинхронных автоматов на ЭВМ. Минск: Наука и техника, 1975. 181 с.
2. S. Baranov. Logic synthesis for control automata // Dordrecht; Boston; London: Kluwer academic publishers, 1994.
3. K. Brayton, R. Rudell, A., Sangiovanni-Vincentelli and A.R. Wong. MIS: A multi-level logic optimization program // IEEE Trans. On computer-aided design. Nov. 1987. Vol. 7. P 1062–1081..
4. I. Levin, M. Karpovskii. On-line self-checking of microprogram control unit // 4th IEEE Intl. On-line testing workshop. Capry, Italy, July 1998. P 152–156.
5. A.Yu. Matrosova, S.A. Ostannin. Self-checking synchronous FSM network design // 4th IEEE Intl. On-line testing workshop. Capry, Italy, July 1998. P. 162–166.
6. Астафьев М.В., Левин И., Матросова А.Ю., Синельников В. Синтез самопроверяемых автоматных сетей в базисе ПЛИМ // Автоматика и телемеханика (в печати).
7. Азубалов Г.П., Оранов А.М. Лекции по теории конечных автоматов. Томск: Изд-во ТГУ, 1984. 185 с.
8. E. Randal. Bryant graph-based algorithms for boolean function manipulation // IEEE transactions on computers. August 1986. Vol. C-35, № 8.
9. A. Yu. Matrosova, S. A. Ostannin. Self-checking FSM design with observing only FSM outputs // 4th IEEE Intl. On-line testing workshop. Capry, Italy, July 1998.

Статья представлена кафедрой программирования факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию 24 февраля 2000 г.

УДК 519.714

В.А. Беляев

НЕКОТОРЫЕ АЛГОРИТМЫ СТРУКТУРНОЙ ДЕКОМПОЗИЦИИ УПРАВЛЯЮЩИХ СИСТЕМ

Работа выполнена при финансовой поддержке РФФИ, грант № 98-01-00288

Рассматривается широкий класс задач структурной декомпозиции, возникающих при исследовании управляющих систем. Предлагаются общая математическая модель, метод и эффективная технология решения рассматриваемых задач. При помощи этой технологии разрабатываются практические алгоритмы решения конкретных задач структурной декомпозиции, возникающих при рассмотрении задач анализа и синтеза управляющих систем.

Управляющие системы представляют собой объекты дискретной природы и характеризуются функцией и структурой. Способы задания функциональных характеристик очень многообразны и включают в себя системы уравнений, формулы, микропрограммы и пр. Структура управляющей системы задается схемой. Под структурной декомпозицией управляющих систем в данной работе понимается задача декомпозиции соответствующих схем. Задачи структурной декомпозиции возникают при исследовании управляющих систем на стыке логического и конструкторского этапов проектирования, когда требуется распределить заданную схему устройства по конструктивным блокам – интегральным микросхемам, базовым ячейкам или стандартным элементам БИС, а те, в свою очередь, – по печатным платам, базовым кристаллам или матричным БИС в заданном монтажном пространстве с обеспечением конструкторских и технологических ограничений.

Схема [1] задается тройкой объектов (X, Z, Z_0) , где X – множество элементов схемы, Z – множество её цепей и $Z_0 \subseteq Z$ – множество её внешних цепей или полюсов. Элемент схемы определяется множеством своих полюсов и весом – некоторой количественной характеристикой сложности элемента. По определению, множества полюсов различных элементов в схеме не пересекаются. Цепь схемы – это подмножество множества полюсов элементов в схеме. Она может дополнительно характеризоваться весом – значением некоторого параметра её физического исполнения (толщины, кратности и т.п.). По определению, различные цепи схемы (элементы множества Z) не пересекаются, и их объединение есть множество всех полюсов элементов схемы. Схема $S=(Y, U, U_0)$ называется подсхемой схемы (X, Z, Z_0) , если $Y \subseteq X$, $u \in U$ тогда и только тогда, когда $u \subseteq z$ для некоторого $z \in Z$ и в u входят все полюсы из Z , принадлежащие элементам в Y , и $u_0 \in U_0$ тогда и только тогда, когда $u_0 \notin Z$ или $u_0 \in Z_0$; в этом случае говорят, что подсхема S порождена подмножест-

вом элементов Y . По определению, вес цепи u и подсхемы совпадает с весом цепи $z \supseteq u$ схемы.

Поскольку порождающее подмножество определяет подсхему однозначно, имеется возможность там, где это не вызывает двусмысленности, рассматриваемые подсхемы отождествлять с их порождающими подмножествами, говоря, например, «подсхема Y » вместо «подсхема (Y, U, U_0) ». В дальнейшем для упрощения изложения мы будем пользоваться этой возможностью без дополнительных оговорок. Соответственно этому к подсхемам будем применять любые теоретико-множественные операции и отношения.

Задачи декомпозиции схем рассматриваются в следующей постановке. Заданы схема $S=(X, Z, Z_0)$, отношение несовместимости v на множестве X её элементов, тройка положительных чисел w, q и p и система подмножеств $A_1, \dots, A_m$ из X ; требуется разбить S на минимальное число подсхем так, что:

- а) любые два элемента в каждой подсхеме совместимы;
- б) сумма весов элементов каждой подсхемы не превосходит числа w ;
- в) сумма весов цепей каждой подсхемы не превосходит числа q ;
- г) сумма весов полюсов каждой подсхемы не превосходит числа p ;
- е) элементный состав каждой подсхемы является подмножеством, содержащимся хотя бы в одном из заданных подмножеств $A_1, \dots, A_m \subseteq X$.

Здесь и далее элементы схемы называются совместимыми, если они не находятся в заданном отношении v , и подмножество элементов называется множеством совместимости, если все они попарно совместимы. В известном смысле числа w, q и p ограничивают сложность компоненты искомого разбиения, а ограничение а) означает, что множество элементов каждой такой компоненты должно быть множеством совместимости.

Различные подмножества множества ограничений $\{a, b, c, d, e\}$ в постановке задачи определяют частные задачи. Некоторые из них являются хорошо известными комбинаторными задачами. Так, ограничение а) определяют задачу раскраски графа [2] – в нашем случае графа $G=(X, v)$, а ограничение б) – задачу разбиения множества чисел на классы с ограниченной суммой [3], известную как вариант задачи загрузки [4]. Ограничение е) определяет задачу покрытия булевой (с элементами 0 и 1) матрицы $M=||m_{ij}||$ задающей подмножества $L_1, \dots, L_n$, содержащиеся в некоторых заданных множествах $A_1, \dots, A_m \subseteq X$ по правилу

$$\forall i \in \{1, \dots, t\} \forall j \in X = \{1, \dots, n\} (m_{ij} = 1 \leftrightarrow j \in L_i).$$

В каждой частной задаче, определяемой подмножеством D ограничений из $\{a, b, c, d, e\}$, подсхемы, удовлетворяющие выбранному подмножеству D , будем называть допустимыми. Ясно, что понятие допустимости в каждой такой задаче зависит от выбранного подмножества D и распространяется только на рассматриваемую подзадачу.

Понятие допустимости подсхемы (схемы) распространяем на множество порождающих её, элементов. Подсхемы или порождающие подмножества элементов в разбиении будем называть блоками разбиения, а блоки, удовлетворяющие заданному набору ограничений, – допустимыми блоками разбиения. Разбиение с минимальным числом классов в дальнейшем будем называть кратчайшим, а кратчайшее разбиение на допустимые блоки – кратчайшим допустимым разбиением. Тогда сформулированную выше задачу разбиения схемы S на минимальное число допустимых подсхем можно рассматривать как задачу кратчайшего разбиения множества элементов схемы на допустимые блоки.

Наряду с рассмотренными задачами кратчайшей декомпозиции при исследовании управляющих систем возникает ещё одна разновидность декомпозиционных задач – задачи упорядоченной декомпозиции. Для их формулировки введём некоторые определения.

Последовательность подмножеств $X_1, \dots, X_k$ множества X , в которой все непустые подмножества различны и образуют разбиение множества X , называется упорядоченным разбиением данного множества и обозначается $(X_1, \dots, X_k)$; подмножества $X_1, \dots, X_k$ суть классы этого разбиения.

Обратим внимание на основные отличия упорядоченного разбиения от неупорядоченного:

1) классами упорядоченного разбиения может быть и пустое множество;

2) порядок перечисления классов в упорядоченном разбиении существен, любое его изменение порождает, вообще говоря, новое упорядоченное разбиение.

Для формулировки задачи упорядоченного разбиения в условия предыдущей задачи добавим ещё некоторое заданное натуральное число $k > 1$ и числовую неотрицательную функцию F , определённую на любых упорядоченных разбиениях $(Y_1, \dots, Y_k)$ любого подмножества элементов $Y \subseteq X$ схемы S . Тогда задача заключается в нахождении такого упорядоченного разбиения $K=(X_1, \dots, X_k)$ множества X элементов схемы S на k блоков, для которого каждый непустой блок является допустимым в смысле ограничений $\{a, b, c, d, e\}$ и числовая функция $F(K)$ принимает на этом разбиении минимальное значение.

Такое разбиение в дальнейшем называется минимальным допустимым разбиением.

Сформулированную выше задачу о кратчайшей декомпозиции в дальнейшем для краткости будем называть задачей декомпозиции типа I, а задачу о минимальном упорядоченном разбиении – задачей декомпозиции типа II. Если в последней задаче значение функции F положить равным числу непустых блоков в разбиении $(X_1, \dots, X_k)$, то любая задача типа I сводится к задаче о минимальном упорядоченном разбиении.

В принципе, любую задачу декомпозиции можно решить тривиальным перебором всех возможных разбиений множества X , выбором среди них допустимых, а среди последних – минимальных или кратчайших. Объём перебора при таком подходе оценивается числом всех возможных разбиений n -элементного множества, которое вычисляется по следующей рекуррентной формуле

$$[1]: R_n = \sum_{i=1}^n C_{n-1}^{i-1} R_{n-i}, \text{ где } R_0=1. \text{ Число таких разбиений с}$$

ростом n растёт значительно быстрее, чем 2^n , поэтому тривиальный подход осуществим лишь для задач с числом элементов в разбиваемом множестве порядка нескольких десятков. Для практических же целей требуется находить декомпозиции множеств, содержащих сотни и даже тысячи элементов.

Известны подходы к решению некоторых из рассматриваемых задач, основанные на использовании математического (часто нелинейного) программирования с булевыми переменными [5,6]. В этом случае любое разбиение представляется набором булевых переменных x_{ij} , принимающих значение 1 если элемент $i \in X$ помещается в блок с номером $j \in \{1, \dots, n\}$, и значение 0 в противном случае, и строится путём последовательного определения компонент в наборе. Решение сформулированной задачи сводится к поиску таких значений переменных x_{ij} , которые представляют допустимые минимальные или кратчайшие разбиения. Однако «потолок» методов, использующих эти подходы, также не превышает нескольких десятков элементов в разбиваемом множестве.

Среди комбинаторных подходов, получивших широкое развитие, известны [7,8] два альтернативных по способу построения разбиений подхода, использующих дерево поиска и некоторые приёмы сокращения числа перебираемых вариантов на нём. Подходы отличаются способом построения дерева поиска.

В первом подходе на каждом шаге ветвления выбирается (формируется) очередной допустимый блок разбиения; число ярусов в дереве при этом не превышает числа блоков в искомым разбиениях, а коэффициент ветвления в каждой вершине равен числу вариантов выбора очередного допустимого блока.

При втором подходе формирование блоков разбиения ведётся параллельно: на каждом шаге ветвления рассматривается очередной элемент множества X и для него подбирается подходящий блок, в который он может быть включён без нарушения условий допустимости. Количество ярусов в таком дереве равно $n=|X|$, и коэффициент ветвления в каждой вершине не превышает k , где k – наибольшее число блоков в строящихся разбиениях.

Подход, развиваемый в данной работе, относится к первому способу организации дерева поиска. В

его основе лежит метод сокращённого обхода дерева поиска [9]. В работе метод применяется для решения задач декомпозиции типа I и II. Для этого вводится общая математическая модель задач декомпозиции – так называемая *T*-задача, формулируется параметрический метод и алгоритм сокращённого обхода дерева поиска и предлагается технология решения широкого класса задач декомпозиции.

T-задача

Вводимая *T*-задача ставится следующим образом. Пусть имеется тройка определённых каким-либо эффективным способом объектов $T = \langle X, P, F \rangle$, где X есть конечное множество, P – множество некоторых подмножеств в X и F – функция, сопоставляющая каждому (возможно, упорядоченному) разбиению R^Y любого подмножества $Y \subseteq X$ на классы из P некоторое неотрицательное число $F(R^Y)$, называемое сложностью разбиения R^Y . Требуется найти, если возможно, такое (упорядоченное) разбиение R множества X на классы из P , которое доставляет минимум функции F .

Под интерпретацией *T*-задачи понимается всякая конкретная задача декомпозиции, которая может быть сформулирована как *T*-задача для некоторой тройки $T = \langle X, P, F \rangle$. Иначе говоря, некая задача A считается интерпретацией *T*-задачи (сокращённо – *T*-интерпретацией), если для неё возможно определить подходящую тройку объектов $\langle X, P, F \rangle$ и указать какое-либо соответствие, относящее каждому решению *T*-задачи для этой тройки некоторые решения задачи A и наоборот, каждому решению задачи A – некоторое решение *T*-задачи.

Многие классические комбинаторные задачи являются в действительности некоторыми *T*-интерпретациями. Мы продемонстрируем это на примере лишь одной задачи – о покрытии булевой матрицы.

Задача состоит в нахождении для заданной булевой (с элементами 0 и 1) матрицы M такого наименьшего по мощности подмножества её строк, называемого кратчайшим покрытием матрицы, в котором для каждого столбца M имеется покрывающая его строка, т. е. строка, содержащая единицу в этом столбце. Пусть матрица M имеет n столбцов, обозначенных $1, \dots, n$ соответственно. Положим $X = \{1, \dots, n\}$. Будем говорить, что булев вектор $a_1, \dots, a_n$ представляет собой множество $Y \subseteq X$, если $\forall i \in X (a_i = 1 \Leftrightarrow i \in Y)$. Пусть множество P содержит в себе каждое такое и только такое подмножество $Y \subseteq X$, для которого в матрице M существует строка, представляющая некоторое множество $Z \supseteq Y$. Пусть также $F(R^Y) = |R^Y|$. Тогда разбиению $R_0 = \{X_1, \dots, X_k\}$, являющемуся решением *T*-задачи для указанной тройки $T = \langle X, P, F \rangle$, соответствует кратчайшее покрытие матрицы M , состоящее из некоторых её строк $b_1, \dots, b_k$ представляемые которыми подмножества в X содержат классы $X_1, \dots, X_k$ соответственно.

Метод сокращённого обхода дерева поиска

Сформулируем теперь метод сокращённого обхода дерева поиска. Для заданной тройки $T = \langle X, P, F \rangle$ рассмотрим некоторый алгоритм φ , перечисляющий в любом множестве $Y \subseteq X$ (возможно, в зависимости от некоторого условия α) некоторые подмножества, принад-

лежащие P и (только в случае, когда упорядоченность искомого разбиения не существенна!) содержащие некоторый фиксированный элемент в Y . Совокупность всех подмножеств в Y , перечисляемых алгоритмом φ для заданного α , обозначается $\varphi_\alpha(Y)$ или $\varphi(Y)$, если эта совокупность не зависит от α , т. е. безусловна. В частности, когда $\varphi(Y)$ состоит из всех указанных подмножеств в Y , алгоритм φ обозначается $\tilde{\varphi}$. Пусть также ψ есть операция, сопоставляющая (возможно, тоже в зависимости от некоторого условия β) множеству $Z = \varphi_\alpha(Y)$ некоторое подмножество $\psi_\beta(Z) \subseteq Z$. Результат операции ψ , не зависящей от какого-либо условия β , обозначается через $\psi(Z)$. Если, кроме того, ψ есть тождественная операция, т. е. $\psi(Z) = Z$, то сама она обозначается буквой I . В дальнейшем φ называется алгоритмом перечисления (условным или безусловным), а ψ – операцией сокращения (также условной или безусловной).

Построим для рассматриваемых алгоритма φ и операции ψ дерево D_φ^ψ с корнем (дерево поиска) следующим образом. Вершины и рёбра дерева сопоставляются подмножествам множества X . Вершины, соответствующие пустому множеству, называются листьями дерева. Корень (вершина 1-го яруса) ставится в соответствие множеству X . Пусть таким образом уже построены вершины ≥ 1 ярусов и рёбра $(i-1)$ ярусов дерева D_φ^ψ . Рассмотрим произвольную вершину v i -го яруса и сопоставленное ей множество $Y \subseteq X$. Вершина v называется концевой, если $\psi_\beta(\varphi_\alpha(Y)) = \emptyset$. Для каждой не концевой вершины v i -го яруса вычисляется множество $\{Z_1, \dots, Z_i\} = \psi_\beta(\varphi_\alpha(Y))$. Множествам $(Y - Z_1), \dots, (Y - Z_i)$ сопоставляются вершины $(i+1)$ -го яруса, смежные вершине v , а ориентированным рёбрам, соединяющим v с ними, подмножества $Z_1, \dots, Z_i$ соответственно. Построение дерева D_φ^ψ заканчивается, когда на некотором его ярусе каждая вершина оказывается концевой.

Для любого пути s в дереве D_φ^ψ через $R(s)$ обозначается множество всех подмножеств, сопоставленных рёбрам пути s и указанных в порядке вхождения соответствующих рёбер в путь s . По построению дерева это есть упорядоченное разбиение разности множеств, сопоставленных началу и концу пути s , на классы из P , которое называется разбиением, соответствующим пути s . Пусть S_φ^ψ обозначает множество всех путей в D_φ^ψ , соединяющих корень с листьями дерева, а $\hat{S}_\varphi^\psi$ обозначает подмножество всех тех путей из S_φ^ψ , соответствующие которым разбиения доставляют минимум функции F . Пусть также $R_\varphi^\psi = \{R(s) : s \in S_\varphi^\psi\}$ и $\hat{R}_\varphi^\psi = \{R(s) : s \in \hat{S}_\varphi^\psi\}$. Тогда из построения дерева D_φ^ψ и определения алгоритма $\tilde{\varphi}$ следует, что множество R_φ^ψ содержит всевозможные упорядоченные разбиения множества X на классы из P и $\hat{R}_\varphi^\psi$ и образует множество всех решений *T*-задачи.

Таким образом, решение *T*-задачи, если оно существует, можно найти, выбрав некоторый путь $\hat{s} \in \hat{S}_\varphi^\psi$, пере-

числя все пути в S_φ^I и выписав соответствующее ему разбиение $R(s)$. Однако в практических ситуациях перебор всех путей в S_φ^I оказывается фактически невыполнимым даже с помощью ЭВМ, в связи с чем возникает необходимость в его сокращении, т.е. в замене перебором части множества S_φ^I , содержащей пути из S_φ^I .

Одна из возможностей сокращения указанного перебора связана с использованием алгоритма φ и операции ψ . Алгоритм φ и операция ψ называются допустимыми для T -задачи, если в S_φ^W имеется хотя бы один путь из S_φ^I . Решение T -задачи для допустимых φ и ψ может быть найдено перебором только множества S_φ^W .

Другая возможность для сокращения перебора вытекает из некоторых свойств функции F в тройке $T = \langle X, P, F \rangle$. В широком классе интерпретаций T -задачи функция F обладает свойством монотонности, которое позволяет ввести числовую неотрицательную функцию F_0 , определённую на подмножествах в X так, что для любого $Y \subseteq X$ и для любых разбиений R^Y и R^{X-Y} соответственно множеств Y и $X-Y$ справедливо: $F(R^{X-Y} \cup R^Y) \geq F(R^{X-Y}) + F_0(Y)$.

Смысл функции F_0 , называемой граничной или функцией нижней оценки, состоит в том, что значение $F_0(Y)$ есть нижняя граница величины, которая при вычислении сложности $F(R^{X-Y} \cup R^Y)$ разбиения множества X добавляется к сложности $F(R^{X-Y})$ имеющегося разбиения множества $X-Y$ за счёт возможного разбиения остатка Y .

Метод сокращённого обхода дерева поиска основан на выполнении специальной процедуры обхода некоторой части дерева D_φ^W для допустимых алгоритма перечисления φ и операции сокращения ψ . Эта процедура представляет собой последовательность операций двух типов – спуск и подъём (по дереву D_φ^W), осуществляемых при определённых условиях. Спуск есть переход из вершины v i -го яруса в смежную вершину $(i-1)$ -го яруса ($i > 1$), а подъём есть переход из вершины v i -го яруса в вершину z $(i+1)$ -го яруса, соединённую с v не пройденным ребром. После выполнения операции подъёма ребро vz объявляется пройденным. Подъём осуществляется всегда, когда не выполнены условия для спуска и существует не пройденное ребро, инцидентное с вершиной v . Спуск выполняется, если вершина v концевая или (в случае монотонной функции F) $F_0(Y) \geq F(R) - F(R(s_v))$, где Y – подмножество в X , сопоставленное вершине v ; s_v – путь из корня в v ; R – известное в момент выполнения данной операции разбиение множества X на классы из P . Перед началом обхода в качестве R может быть принято любое разбиение R_0 , представляющее собой приближённое (эвристическое) решение T -задачи, отыскиваемое с помощью некоторого быстродействующего алгоритма. В случае, если такое разбиение неизвестно, то условно считается $R_0 = \emptyset$ и $F(R_0) = \infty$. Всякий раз, когда в процессе обхода достигается вершина v , являющаяся

листом дерева, за R принимается разбиение $R(s_v)$. Обход начинается в корне дерева D_φ^W и заканчивается в вершине v , в которой либо $F(R(s_v)) = F_0(X)$ (тогда v есть лист дерева), либо ни одна из указанных операций неприменима (тогда v есть корень). По окончании обхода разбиение R является решением T -задачи, если $R \neq \emptyset$.

Предлагается конкретный алгоритм, реализующий описанную процедуру обхода дерева поиска применительно к T -задаче с монотонной функцией F .

Алгоритм обхода

1. $X^{(1)} := X$, $R := R_0$, $R' := \emptyset$, $i := 1$.
2. Вычисляется множество $\{Z_1^{(i)}, \dots, Z_n^{(i)}\} = \psi_\beta(\varphi_\alpha(Y))$; $k_i := 0$.
3. $k_i := k_i + 1$; если $k_i > l_n$ то п. 4; в противном случае $X^{(i+1)} := X^{(i)} - Z_{k_i}^{(i)}$, $R' := R' \cup \{Z_{k_i}^{(i)}\}$, $i := i + 1$. Если $X^{(i)} = \emptyset$, то п. 5; иначе при условии $F_0(X^{(i)}) \geq F(R) - F(R')$ выполняется п. 4; в противном случае – п. 2.
4. Если $i \geq 1$, то п. 6; иначе $i := i - 1$, $R := \{Z_{k_i}^{(i)}, \dots, Z_{k_{i+1}}^{(i+1)}\}$ и п. 3.
5. Если $F(R') < F(R)$, то $R := R'$ и, если $F(R) = F_0(X)$, то п. 6, в остальных случаях – п. 4.
6. Конец; если $R \neq \emptyset$, то R есть решение T -задачи, в противном случае решения не существует.

Для применения данного алгоритма к T -задаче с немонотонной функцией F достаточно в п. 3 и 5 алгоритма включить блокировку сравнений с граничной функцией F_0 . В тех случаях, когда можно удовлетвориться приближённым решением T -задачи, но желательно, чтобы это решение было по возможности более точным (т.е. с меньшим значением функции F), в этих случаях в п. 4 алгоритма перед операцией $i := i - 1$ следует поставить переключатель по времени – операцию, существующую обычно в развитых языках программирования, которая по исчерпанию ресурса времени, отпущенного вычислителю (ЭВМ) на реализацию алгоритма, выводит алгоритм на точку (п. 6), а до того не меняет порядка работы алгоритма. В отсутствие переключателя по времени или при нежелании воспользоваться им допустимый объём вычислений может быть задан ограничением области поиска решения. Так, если в п. 4 алгоритма условие $i \leq 1$ заменить условием $i \leq r$, то поиск будет ограничен лишь теми путями в дереве, которые проходят через некоторую одну и ту же вершину r -го яруса. Результат $R = \emptyset$ в таких случаях должен рассматриваться как свидетельство невозможности получения данным алгоритмом какого-либо решения T -задачи в указанное время или в указанной области поиска. Наконец, если начальное разбиение $R_0 \neq \emptyset$ неизвестно и требуется найти хотя бы одно приближённое решение T -задачи, то в п. 5 алгоритма после операции $R := R'$ достаточно поставить безусловный переход на точку – п. 6.

Технология решения задач структурной декомпозиции

Технология решения задач структурной декомпозиции данным методом заключается в следующем:

1. Определяется тройка объектов $T=\langle X, P, F \rangle$, подходящая решаемой задаче.

2. Подбираются для T допустимые алгоритм перечисления φ и операция сокращения Ψ .

3. В случае монотонной функции F задаётся функция нижней оценки F_0 .

4. Указывается некоторый алгоритм ρ построения начального разбиения R_0 .

5. С помощью алгоритма сокращённого обхода дерева D_φ^Ψ отыскивается решение T -задачи, если оно существует.

Эффективность алгоритма обхода и всей технологии в целом зависит от «сокращающих способностей» алгоритма перечисления и операции сокращения и от степени точности функции нижней оценки и алгоритма начального разбиения. Названные объекты являются *параметрами* метода поиска и выбираются эвристически для каждой конкретной задачи декомпозиции.

Например, для задачи покрытия булевой матрицы предлагаются следующие параметры метода. Пусть матрица M имеет m строк с номерами $1, \dots, m$ и A_i есть подмножество (столбцов) в X , представленное строкой с номером i . Алгоритм φ в данном случае перечисляет в любом множестве $Y \subseteq X$ непустые подмножества $A_i \cap Y$ для $i \in \{1, \dots, m\}$, которые содержат некоторый фиксированный (например, с наименьшим номером) столбец j_0 в Y ; операция ψ отбирает среди этих подмножеств максимальные (не входящие в другие) подмножества; $F_0(Y)$ – любая эффективно вычисляемая оценка снизу для числа строк в кратчайшем покрытии множества Y подмножествами $A_i \cap Y$, $i=1, \dots, m$; например, $F_0(Y) = \lceil |Y| / \max_i |A_i \cap Y| \rceil$; за начальное разбиение R_0 может быть принято любое разбиение $\{X_1, \dots, X_r\}$ множества X , классы которого суть части множеств $A_{i_1}, \dots, A_{i_n}$, представляемые строками матрицы, образующими некоторое её приближённо кратчайшее покрытие, найденное каким-либо из эвристических алгоритмов: $X_1 = A_{i_1}, X_j = A_{i_j} - \bigcup_{k=1}^{j-1} A_{i_k}, j=2, \dots, r$.

Разбиение множества чисел

Задача заключается в нахождении для заданных множества W положительных чисел и числа w , не меньшего любого числа в W , такого разбиения $\hat{R}$ множества W на классы, в котором сумма чисел в каждом классе разбиения не превосходит w и число классов минимально. Всякое такое разбиение $\hat{R}$ множества W называется *w-минимальным*.

Данная задача является формальным эквивалентом задачи распределения конструктивно-функциональных узлов по блокам ограниченной вместимости w , если объёмы распределяемых узлов выражены числами, образующими множество W . Её можно рассматривать также как задачу размещения заданных массивов данных или программ по страницам памяти; в этом случае w есть размер страницы, а числа в W суть размеры размещаемых массивов.

T-интерпретация

Подмножество чисел из W , сумма которых не превосходит w , называется *w-совместимым*. Пусть $X=W$, $F(R^Y) = |R^Y|$ и множество P состоит из всех w -совместимых подмножеств в X . Тогда w -миним

альное разбиение множества W совпадает непосредственно с решением T -задачи для $\langle X, P, F \rangle$. Вводятся следующие параметры метода сокращённого обхода дерева поиска для данной задачи.

Пусть $Y = y_1, \dots, y_r \subseteq X, y_1 \geq \dots \geq y_r$ и для любых натуральных p и q для $1 \leq p \leq q \leq r$ величина $\sigma_{p,q}$ определяется формулой: $\sigma_{p,q} = \sum_{i=p}^q y_i$. Алгоритм φ перечисляет

некоторые максимальные w -совместимые подмножества в Y , отбираемые по следующим правилам, применяемым в указанном порядке.

Алгоритм φ

1. Если для некоторых i и j ($i \neq j$) справедливо $y_i + y_j = w$, то $\varphi(Y) = \{y_i, y_j\}$ для наименьшего такого i .

2. Если для некоторых i и j ($i < j \leq r$) имеет место $y_i + \sigma_{i+1,j} \leq w < y_i + y_j$, где $\sigma_{i+1,i} = 0$, то $\varphi(Y) = \{y_i, y_{i+1}, \dots, y_j\}$ для наименьшего такого i .

3. Если для некоторых i, j и k ($i < j \leq k < r$) справедливы неравенства $y_i + y_j \leq w, y_j \geq \sigma_{k+1,r}, y_i + y_k + y_r > w$, то $\varphi(Y) = \{y_i, y_j\}$ для наименьших i и j с указанными свойствами.

4. Во всех остальных случаях алгоритм φ перечисляет все максимальные w -совместимые подмножества в Y , содержащие наибольшее число y_1 . Приводится конкретная реализация этого пункта алгоритма.

Результат применения операции ψ к множеству $Z \subseteq \varphi(Y)$ вычисляется как нижняя грань множества Z , упорядоченного отношением $\leq$ предшествования, определяемого для любых двух подмножеств $A = \{y_{i_1}, \dots, y_{i_p}\}$ и $B = \{y_{j_1}, \dots, y_{j_q}\}$ из Y следующим образом:

$$A \leq B \leftrightarrow [p=q \wedge \forall i \in \{1, \dots, p\} (i \leq j_i)] \vee [p=2 \wedge A \cap B \neq \emptyset \wedge \sigma(A) \geq \sigma(B)],$$

где $i_1 < \dots < i_p, j_1 < \dots < j_q$ и $\sigma(C) = \sum_{y \in C} y$.

Граничная функция F_0 определяется формулой: $F_0(Y) = \max(k, l)$, где $l = \lceil \sigma(Y)/w \rceil$ и k есть наименьшее целое такое, что в ряду $y_1 \geq \dots \geq y_r$ для $\{y_1, \dots, y_r\} = Y$ имеет место $y_k + y_{k+1} \leq w$.

Разбиение графа на подграфы

Пусть имеется граф $G=(V, U)$ и его подграф $G_0=(V_0, U_0)$, порождённый некоторым подмножеством вершин $V_0 \subseteq V$ таким, что никакие две вершины из $V - V_0$ не смежны. Пусть всем вершинам и рёбрам графа G_0 приписаны некоторые натуральные числа, называемые их весами. Вес ребра $u \in U_0$ обозначается через $p(u)$, а вес вершины $i \in V_0$ – через w_i . Каждое множество $Y \subseteq V_0$ характеризуется величинами $w(Y)$ и $q(Y)$, определяемыми следующим образом: $w(Y) = \sum_{i \in Y} w_i, q(Y) = |V_Y| + \sum_{u \in U_Y} p(u)$,

где V_Y – множество вершин из $V - V_0$, смежных одновременно вершинам в Y и вершинам в $V_0 - Y$; U_Y – множество рёбер из U_0 , инцидентных одновременно вершинам в Y и вершинам в $V_0 - Y$. Пусть, кроме того, заданы два натуральных числа q и w , причём $w \geq w_i$ для каждого $i \in V_0$. Требуется найти (если возможно) разбиение $\hat{R}$ множества V_0 с минимальным числом классов $V_1, \dots, V_d$ так, чтобы $w(V_i) \leq w$ и $q(V_i) \leq q$ для каждого $i=1, \dots, d$. Всякое

такое разбиение $\hat{R}$ множества V_0 называется (q, w) -минимальным разбиением в графе G .

Данная задача является одной из разновидностей задачи компоновки схемы управляющей системы в конструктивные блоки. Граф G в этом случае выступает в качестве модели компоновки схемы: вершины в V_0 сопоставляются элементам схемы, а рёбра в U_0 – множествам электрических цепей в схеме, объединяющих пары полюсов соответствующих элементов. Вес такого ребра равен мощности сопоставленного ему множества цепей. Вершинами в $V-V_0$ представляются электрические цепи схемы, объединяющие более двух полюсов (промежуточные узлы связи). Ребро, соединяющее вершину в V_0 и вершину в $V-V_0$, указывает на инцидентность в схеме соответствующих элемента и электрической цепи. Поскольку объём w конструктивного блока ограничен, а размеры w_i реальных физических элементов m_i конечны, то в каждом блоке i при компоновке схемы может быть размещено лишь множество таких элементов, общий объём которых не превышает w . Отсюда следует ограничение на $w(V_i)$ в постановке задачи. Кроме того, конструктивные блоки соединяются между собой посредством разъёмов с ограниченным числом q контактов. Тем самым накладывается ограничение на число $q(V_i)$ внешних связей блока i , т.е. связей элементов в блоке i с элементами других блоков. При подсчёте этого числа все проводники, принадлежащие одной и той же электрической цепи, отождествляются с одним проводником, вследствие чего величина $q(V_i)$ вычисляется по приведённой выше формуле при $Y=V_i$. Общее количество конструктивных блоков, по которым распределяются элементы схемы, должно быть наименьшим; отсюда – тре-

бование минимальности числа классов в разбиении $\hat{R}$.

Т-интерпретация

Множество $Y \subseteq V_0$ называется q -совместимым (w -совместимым), если $q(Y) \leq q$, а соответственно, $w(Y) \leq w$. Множество Y , являющееся одновременно q -совместимым и w -совместимым, называется (q, w) -совместимым. Пусть $X = V_0 \setminus P$ – множество всех (q, w) -совместимых подмножеств в X и $F(R^Y) = R^Y$. Тогда решение T -задачи для $\langle X, P, F \rangle$ является (q, w) -минимальным разбиением в графе G .

Параметры метода

Каждое (q, w) -совместимое подмножество в $Y \subseteq X$ является q -совместимым подмножеством некоторого w -совместимого подмножества $Q \subseteq Y$. В связи с этим алгоритм перечисления Φ для рассматриваемой задачи предложено представлять как композицию (последовательное выполнение) двух алгоритмов – Φ_2 и Φ_3 . Алгоритм Φ_2 в любом $Y \subseteq X$ перечисляет все максимальные w -совместимые подмножества Q , содержащие вершину y_0 наибольшего веса, а алгоритм Φ_3 для каждого $Q \in \Phi_2(Y)$ перечисляет все такие его q -совместимые подмножества, содержащие вершину $y_0 \in Y$, которые являются объединениями некоторых максимальных плотных подмножеств из Q . Множество $A \subseteq X$ называется *плотным*, если для любого непустого подмножества $C \subseteq A$ имеет место $q(C) > q(A)$. Приводится конкретная реализация алгоритма перечисления. Операция ψ исключает из $Z = \Phi(Y)$ всякое такое множество $A \in Z$, для которого в Z найдётся множество $B \supset A$ со свойством $q(A) \geq q(B)$. Допустимость этой операции доказывается соответствующей теоремой. Граничная функция F_0 определяется так же, как в задаче о разбиении множества чисел.

ЛИТЕРАТУРА.

1. Агibalов Г.П. Дискретные автоматы на полурешётках. Томск: Изд-во Том. ун-та, 1993. 227 с.
2. Бергс К. Теория графов и её применения. М.: ИЛ, 1962. 319 с.
3. Агibalов Г.П., Беллев В.А. Метод сокращённого обхода дерева поиска и его применение в синтезе интегральных схем // Управляющие системы и машины. 1977. № 6. С. 99–103.
4. Агibalов Г.П., Беллев В.А. Технология решения комбинаторно-логических задач методом сокращённого обхода дерева поиска. Томск: Изд-во Том. ун-та, 1981. 125 с.
5. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.
6. A.M. Geoffrion, R.E. Marsten. Integer programming: a framework and state-of-the-art survey // Management Science. 1972. Vol. 18. № 9.
7. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980. 476 с.
8. Ершов А.П. Введение в теоретическое программирование. М.: Наука, 1977. 288 с.
9. Закревский А.Д. Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 512 с.

Статья представлена кафедрой защиты информации и криптографии факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию 1 марта 2000 г.

УДК 519.7

В.А. Бузанов

ЗАДАЧИ ДИАГНОСТИКИ ДЛЯ СЛОЖНЫХ ФУНКЦИЙ

Работа выполнена при финансовой поддержке РФФИ, грант № 98-01-00288

Предлагается способ применения специальных интервальных операций при вычислениях в задачах диагностики, если объект диагностики задан простейшей композицией произвольных функций – сложной функцией. Предполагается, что функции-компоненты композиции заданы их интервальными формами.

Для упрощения вычислений ранее [1] были предложены специальные интервальные операции для решения основных задач диагностики. При этом имелось в виду следующее.

Пусть $X = X_1 \times \dots \times X_i \times \dots \times X_n$ – декартово произведение произвольных множеств X_i . Элементы множества X называются векторами или наборами.