

К. А. ДАНИЕЛЯН, Г. П. КАЗАНЧЯН

АВТОМАТИЗАЦИЯ ПЛАНИРОВАНИЯ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

Введение

Основным этапом функционирования системы АНИ-79, общее описание которой дано в работе [1], является планирование алгоритма решения предлагаемой пользователем расчетной задачи. Принципы планирования для решения вычислительных задач были предложены в [2—4]. В этой статье рассматривается работа планировщика и организация обращения к нему.

Пользователь обладает определенной полнотой знания о решаемой задаче, в частности, он может точно определить те объекты (входные данные и конечные результаты), которые нужны для ее формулировки. Система содержит средства для подготовки к реализации на ЭВМ конкретных задач на основе разработанного языка описания запросов (ЯОЗ) [5], близкого к естественному, и транслятора, интерпретирующего предложения этого языка и формирующего необходимый запрос планировщику.

Для планирования нужно иметь сконструированную информационную базу, где хранятся сведения о всех параметрах, характеризующих предметную область, на которой решается задача, и о всех имеющихся в распоряжении программных модулях и их информационных связях. Процесс планирования состоит в выявлении и построении по возможности минимальной цепочки взаимосвязанных модулей, вычисляющих множество выходных параметров задачи по заданному множеству исходных параметров.

1. Язык описания запросов

Планировщик инициируется в режиме использования или решения задач, если задача описана пользователем корректно, что анализируется транслятором с ЯОЗ. Транслятор активизируется подачей директивы ****СПЛАНИРОВАТЬ**, после которой должны быть заданы входные параметры задачи в предложении

>ДАНО: 'наимен. параметра' идентиф. парам.; ...* и выходные параметры в предложении

>НАЙТИ: 'наимен. парам.' идентиф. парам.;...*.

Наименование параметра задается в терминах рассматриваемой предметной области. Идентификатор, представляющий один или два символа, в дальнейшем служит для обозначения соответствующего параметра, заменяя его наименование. Планировщик оперирует только системными номерами параметров, которые ставятся в соответствие с наименованиями параметров при формировании модели предметной области (МПО) в таблице описания параметров—тезаурусе [6].

Значения параметров, условия и ограничения, налагаемые на них, задаются уже непосредственно перед выполнением. В системе АНИ-79 предусмотрена возможность использования сведений о параметрах, запоминаемых в базе данных (БД). Для обращения к БД пользователю нужно применить директиву ****ИСПОЛЬЗОВАТЬ БД**, после которой должны следовать предложения, отвечающие операциям поиска, извлечения и обновления информации, хранящейся в БД.

Если пользователь хочет занести значение одного или нескольких параметров в базу данных для их дальнейшего использования, он должен в предложении

>ЗАПОМНИТЬ: 'наимен. парам'. (знач. парам.);...* перечислить их наименования. Значения могут и не указываться, если они вычисляются в процессе решения задачи или задаются в качестве исходных данных для задачи.

Для использования каких-то значений из БД в процессе планирования их наименования включаются в предложение

>ПРИМЕНИТЬ: 'наимен. парам.';...*.

Когда отпадает необходимость хранения в БД каких-то данных, они уничтожаются при помощи предложения

>УДАЛИТЬ: 'наимен. парам.';...*.

В процессе работы с системой пользователь может собирать сведения о некотором объекте, к которому имеют отношение решаемые им задачи и характеристики которого используются в качестве параметров этих задач. Таким образом, в БД может динамически формироваться таблица описания объектов. В ней в текущий момент времени указываются уже известные характеристики объекта и их значения, хотя физически последние хранятся в общем массиве памяти со значениями параметров, не связанных с определенными объектами. Параметры соотносятся с объектом с помощью предложения

>ВКЛЮЧИТЬ: имя объекта, наимен. парам.;...*.

и, наоборот, параметр выносится из описания объекта предложением

>УДАЛИТЬ: имя объекта, наимен. парам.;...*.

Для проверки информации, хранимой в БД, применяются предложения

>ВЫДАТЬ: наимен. парам.;...*,

считывающее значение этого параметра, и

>ВЫВЕСТИ: имя объекта*,

сообщающее обо всех параметрах, характеризующих указанный объект.

В систему введено понятие проекта. Каждый проект определяет

достаточно четко очерченный класс задач. Это дает возможность в процессе планирования рассматривать именно те программные модули, которые при конструировании МПО были включены в проекты, указанные в описании задачи.

Допускается до восьми проектов, задаваемых номерами от 1 до 8 в предложении

>ПРОЕКТ; Пр,...*.

Тогда планировщик ведет поиск по тем модулям, которые удовлетворяют хотя бы одному из перечисленных проектов. Если это предложение отсутствует, то поиск необходимых модулей ведется по всей модели. Но пользователь может исключить из рассмотрения некоторые модули на время решения своей задачи специальным предложением

>ИСКЛЮЧИТЬ: имя модуля, ...*.

О преимуществах данной директивы будет сказано после описания алгоритма планирования.

Пользователь должен завершить формулирование своей задачи предложением

>КОНЕЦ*.

Предложения ЯОЗ могут вводиться либо с перфокарт, либо с терминальных устройств (консоль, дисплей). Ниже дан пример применения рассмотренного языка описания запросов для конкретной задачи:

>ПРОЕКТ: 1, 5*

>ДАНО: 'КОЭФФИЦИЕНТ ТРЕНИЯ' КТ;

'УГОЛ НАКЛОНА ПЛОСК';

'УСКОРЕНИЕ ТЕЛА' А*

>НАЙТИ: 'РЕЗУЛЬТИР. СИЛА' FR *

* ИСПОЛЬЗОВАТЬ БД *

>ПРИМЕНИТЬ: МАССА ТЕЛА *

>КОНЕЦ *

Транслятор с ЯОЗ после распознавания поступивших предложений и выбора необходимых сведений из таблиц описания модели предметной области формирует запрос планировщику в виде некоторого информационного массива дескрипторов всех параметров задачи, имеющего следующий формат:

Кроме того планировщику передаются номера допустимых проек-

Количество входных параметров задачи	Количество выходных параметров задачи
Системный номер i -го входного параметра задачи	
	Вид: скаляр Φ вектор Ψ матрица Z
R_i	Φ
Размерность	
:	
:	
Системный номер j -го выходного параметра задачи	
	Вид: скаляр Φ вектор Ψ матрица Z
Ψ	Φ
Размерность	
:	
:	

Рис. 1

тов и системные номера блокируемых модулей.

2. Алгоритм планирования

Процесс планирования начинается с выяснения потенциальной возможности решения задачи на данной МПО. Под потенциальной возможностью подразумевается наличие программных модулей, вычисляющих искомые параметры задачи. Удовлетворительный ответ является необходимым, но не достаточным условием разрешимости. Далее планирование разбивается на два этапа: прямой и обратный ход.

Прямой ход состоит в нахождении пути вычисления множества выходных параметров задачи Y по заданному множеству входных параметров X во множестве проектов R . Пусть задано множество программных модулей $M = \{M_i\}$, $i = 1, 2, \dots, n$, причем модуль M_i требует на входе множество параметров P_i , по значениям которых он вычисляет параметры Q_i во множестве проектов R_i . На первом этапе планирования динамически формируется некоторое множество параметров S . Вначале оно включает в себя только входные параметры задачи, а по мере нахождения нужных модулей, пополняется новыми параметрами.

Поиск ведется по тем модулям из множества M , для которых соответствующее множество проектов R_i имеет общие элементы с указанным в описании задачи множеством R .

В процессе прямого планирования формируется также упорядоченное множество активных модулей C . В него последовательно включаются те модули из числа допустимых, все входы которых уже являются элементами множества S . При выборе очередного модуля параметры, вычисляемые им, засылаются в S . По мере расширения множества C , а значит, и S , производится следующая проверка: входят

ли все выходные параметры задачи во множество S , т. е. $Y \subseteq S$? В случае положительного ответа первый этап планирования считается завершенным. В противном случае поиск продолжается. Если для данной совокупности входных параметров X все возможности дальнейшего расширения множества S будут исчерпаны в рамках сконструированной МПО, то пользователю сообщается о недостаточности входных данных. Он может вновь запустить задачу, но уже с указанием недостающих параметров, т. е. задача будет решаться для нового множе-

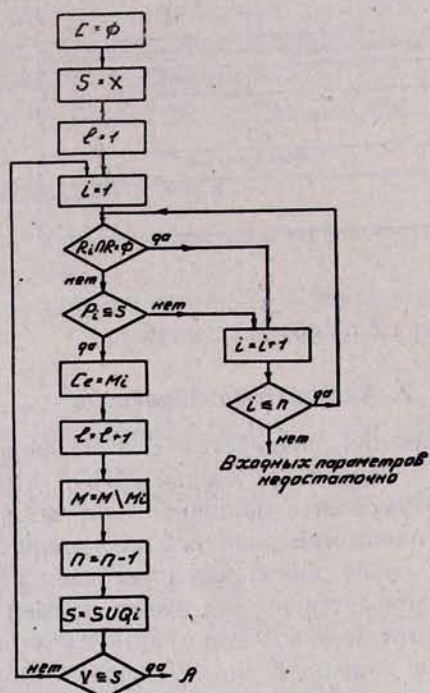


Рис. 2. Прямой ход планирования

ства X . Итак, в результате прямого хода при удачном его завершении будет создана последовательность модулей C , выполнение которых в полученной очередности приводит к вычислению искомым параметров задачи Y (см. Рис. 2). Но множество C может включать и модули, не влияющие на получение результатов. Эти избыточные модули исключаются из C на втором этапе планирования. Здесь динамически формируется множество параметров T , совпадающее сначала с заданным множеством Y . Избыточность последовательности модулей C выявляется при следующей проверке: есть ли среди выходов очередного модуля M_i , выбранного с конца последовательности C , параметры из множества T ? Если $Q_i \cap T = \emptyset$, то модуль исключается из множества C . В противном случае, когда модуль необходим для получения искомым параметров, производится корректировка множества T . Из него извлекаются параметры, составляющие множество $Q_i \cap T$, и наоборот, пополняется T теми входами выбранного модуля, которые не

являются элементами множества X . Просмотр последовательности модулей C ведется до тех пор, пока все параметры из T будут найдены, т. е. $T = \emptyset$. Полученная таким образом упорядоченная совокупность модулей C и будет искомой цепочкой модулей, выполнение которых в указанной последовательности приведет к нахождению выходных параметров задачи. Выбранная цепочка может не требовать всех входных данных задачи, о чем сообщается пользователю, и он может не указывать их значений для решения своей задачи (см. рис. 3).

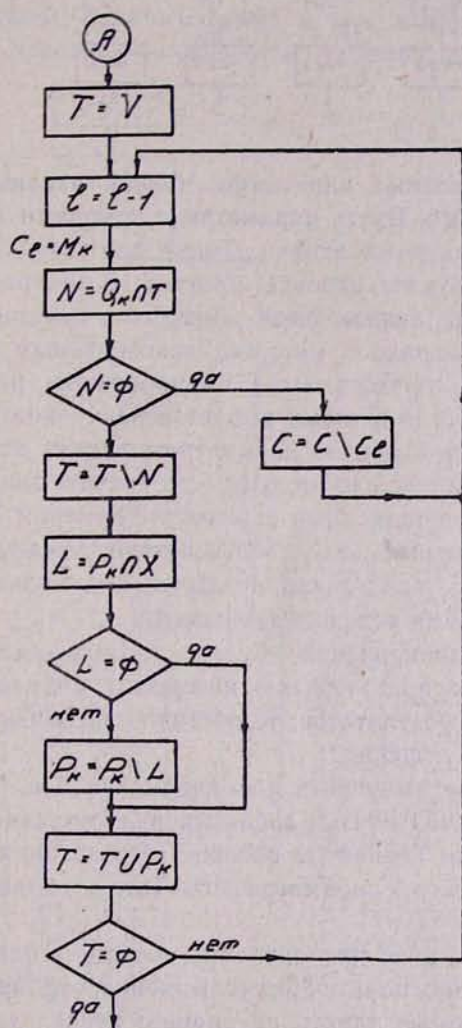


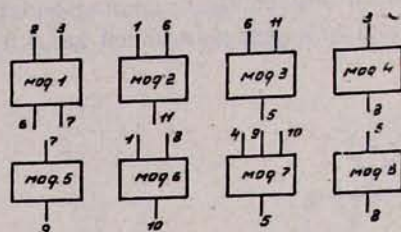
Рис. 3. Обратный ход планирования

Планировщик кроме цепочки модулей посылает на выполнение заново сформированную таблицу описания параметров. Он заполняет ее использованными входными параметрами задачи и всеми промежу-

точными параметрами, встречающимися при прохождении по спланированной цепочке.

Теперь рассмотрим примеры планирования цепочки с использованием директивы >ИСКЛЮЧИТЬ: имя модуля,...*.

Допустим, имеются следующие 8 модулей:



Входные и выходные параметры модуля заданы здесь своими системными номерами. Пусть параметры с номерами 1, 2, 3, 4 являются входными параметрами задачи. Тогда для нахождения параметра 5 системой спланируется цепочка модулей с номерами 1, 2, 3. При этом параметр 4 не используется. Допустим пользователь не хочет, чтобы его задача решалась методом, используемым в модуле 3. Заблокировав его, он предоставляет планировщику решить задачу по иному алгоритму. Тогда система выдает цепочку модулей: 1, 4, 5, 6, 7.

В случае, когда искомым параметром задачи является параметр 8, цепочка будет состоять из модуля 4, а значит, для решения задачи достаточно задать единственный параметр с номером 3. Но может оказаться нецелесообразным для пользователя включать модуль 4 в рассмотрение, тогда, аналогично предыдущему, пользователь выносит его из модели на время решения этой задачи.

Результатом планирования будет последовательность модулей 1, 2, 3, 8. Если же пользователю интересны модули только с точки зрения получаемых результатов, то очевидно, он выберет первый, более экономный вариант решения.

Пусть требуется вычислить оба параметра 5 и 8, и пользователь директивой >ИСКЛЮЧИТЬ* заблокировал модули 3 и 4. В этом случае планировщик не найдет решения без задания дополнительных параметров, например, параметра 10. Тогда спланируется цепочка 1, 5, 7, 8.

Таким образом, хотя предлагаемый алгоритм планирования выбирает не всегда наилучший вариант решения, он предоставляет возможность пользователю влиять на процесс поиска для выявления наиболее эффективной цепочки модулей. Эффективность цепочки определяется требованиями самого пользователя и зависит как от эффективности самих программных модулей с точки зрения применяемых в них методов вычисления или времени работы модуля, так и от длины цепочки, от количества требуемых входных параметров, причем с учетом данных, которыми располагает пользователь. Определив для себя

некоторый критерий эффективности, пользователь выберет наиболее предпочтительный алгоритм решения.

3. Организация циклов

В системе АНИ-79 пользователю предоставляется возможность задавать входной параметр множеством значений с тем, чтобы получить соответствующий спектр значения искомым параметров задачи или найти экстремальное значение некоторого выходного параметра. Тогда, как и в общем случае, по наименованиям параметров планируется цепочка модулей. Задача состоит в том, чтобы выделить из полученной цепочки некоторую подцепочку, которая вычисляет макси-

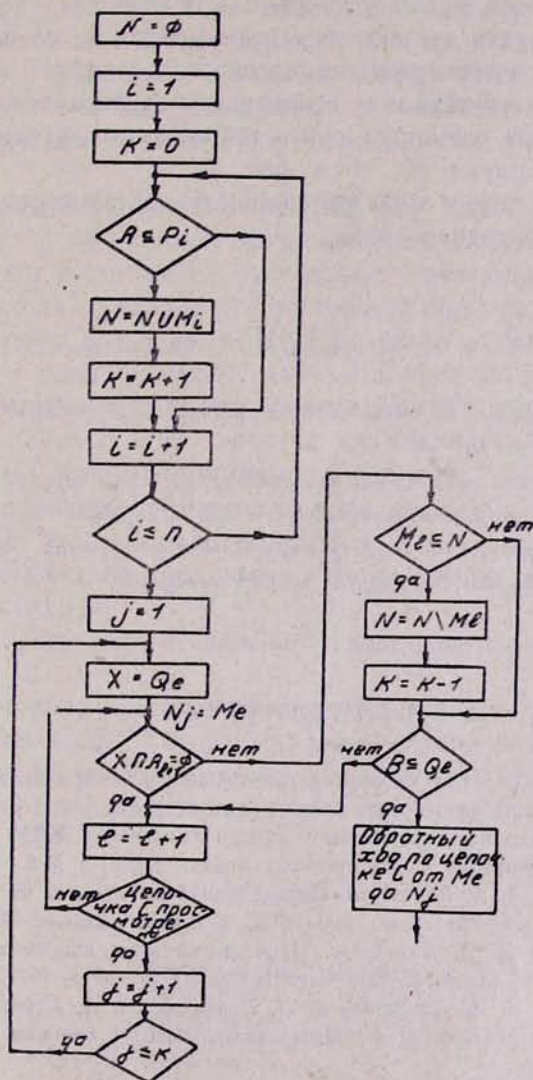


Рис. 4. Организация циклов

зируемый параметр. В процессе выполнения именно эта подцепочка будет прокручиваться в цикле. Число шагов цикла равно числу значений, придаваемых входному параметру. Если на параметр, вычисляемый циклической подцепочкой, налагается условие, то происходит преждевременный выход из цикла при нарушении этого условия.

Рассмотрим алгоритм выделения циклической последовательности программных модулей из уже спланированной цепочки C .

Пусть входной параметр задачи A задан диапазоном значений, и должно быть найдено максимальное значение выходного параметра. Из цепочки C выделяются все модули, требующие параметр A на входе, в некоторое множество N .

Далее, для каждого модуля N_i из множества N выполняется прямой ход планирования по цепочке C . В качестве исходных параметров новой задачи считаются выходы модуля N_i , а искомым — параметр B . Если соответствующая последовательность модулей будет найдена, то уже начиная с ее последнего модуля проводится обратный ход планирования, сканируя опять по модулям исходной цепочки C , до выбранного модуля N_j . (см. рис. 4).

Полученная таким образом подпоследовательность и будет искомой циклической подцепочкой.

4. Ա. ԴԱՆԵԼՅԱՆ, Գ. Պ. ՂԱԶԱՆՉՅԱՆ

ԽՆԴՐԻ ԼՈՒՄԱՆ ԱԼԳՈՐԻԹՄԻ ՊԼԱՆԱՎՈՐՄԱՆ ԱՎՏՈՄԱՏԱՑՈՒՄԸ

Հոդվածում դիտարկվում են խնդիրների լուծման ավտորիթմների ավտոմատիկ պլանավորման ավտորիթմները: Պլանավորման պահանջը տրվում է խնդիրների նկարագրման լեզվով, որը մոտ է բնականին և օգտագործում է նախագծման կոնկրետ տիրույթի տերմինոլոգիան:

ЛИТЕРАТУРА

1. К. А. Абгарян. Базовая система автоматизации научно-технических и опытно-конструкторских разработок. В настоящем сборнике.
2. Э. Х. Тыгу. Решение задач на вычислительных моделях ЖВМ и МФ, 10, 1970, 3.
3. Э. Х. Тыгу. Решатель вычислительных задач. ЖВМ и МФ, 11, 1971, 4.
4. С. С. Гайсарян, А. А. Калентьев. Планирование вычислений на обобщенных вычислительных моделях. Деп. рук., РЖ «Математика», 5В—642в, 1977, 5.
5. С. С. Гайсарян, А. А. Калентьев. Проблемный язык описания проектных задач. Деп. рук., РЖ «Автоматика и вычислительная техника», 1977, 6.
6. А. А. Абрамян, К. А. Даниелян, С. П. Казанчян, М. А. Узунов. Общее описание программной реализации отдельных частей базовой системы АНИ-79. В настоящем сборнике.