

Раздел I

АРХИТЕКТУРА САПР

УДК 658.52.011

В.В. Емельянов, С.И. Ясиновский

Гибридная система для планирования производства на основе генетических алгоритмов, методов имитации и экспертных систем

Решение многих системных задач, связанных со сложными технологическими комплексами (ТК), таких как анализ функционирования, синтез структуры, планирование, управление и др., не обеспечивается чисто математическими методами. Как правило, для принятия решений используют сложное сочетание математики, вычислительных методов (обычно имитационное моделирование на ЭВМ), эвристические, экспериментальные подходы, методы инженерии знаний [чаще всего экспертные системы (ЭС)]. Комплексное использование указанных методов и средств обеспечивает пользователя поддержкой при принятии решений.

Необходимость сращивания имитационных моделей (ИМ) и методов с теми, которые предлагают ЭС приводит к идее гибридных систем (ГС). В настоящее время ГС находятся еще на стадии исследовательского прототипа, инструментальные средства для их создания отсутствуют. Для построения ГС желательно иметь единые средства, с помощью которых можно создавать ИМ, ЭС и реализовывать поисковые алгоритмы. В качестве такого средства может быть использован язык РДО, предназначенный для создания интеллектуальных систем моделирования [1,2].

РДО уже нашел применение для создания реально действующих систем ИМ и ЭС [3,4]. Ниже рассмотрим его применение для построения ГС.

Архитектура ГС

Рассматриваемая ГС предназначена для планирования работы многономенклатурного ТК серийного типа производства. В нее входят:

- 1) ИМ, служащая для составления плана и использующая для этого набор эвристических приоритетных правил [5];
- 2) блок оптимизации подбора приоритетных правил и других переменных планирования, обеспечивающих получение планов работы с наилучшими показателями;
- 3) ЭС, предназначенная для изменения параметров поиска на основе некоторых представлений человека-оператора о перспективности той или иной стратегии поиска.

Имитационная модель ТК

ИМ используется для построения версии расписания при заданных параметрах ТК, для которого проводится планирование, состоит из семи станков и производит детали десяти различных типов. Маршрут обработки деталей представляет собой граф возможных альтернатив. Для каждой операции известны номер станка, который должен выполнять эту операцию, длительность операции и номер наладки. В тестовых прогонах строилось расписание для обработки на участке десяти имеющихся на момент планирования заказов. Каждый заказ включает

детали одного типа и описывается номером типа деталей, количеством деталей, наиболее ранним временем начала и сроком выполнения. Пример портфеля заказов представлен в таблице. Исходными данными для системы краткосрочного планирования являются результаты работы систем долгосрочного планирования типа MRP-II.

Все элементы участка, необходимые при построении расписания, представлены в модели в терминах РДО - метода как ресурсы. Множество ресурсов разбивается на подмножества ресурсов, имеющих одинаковые свойства и описываемых одинаковыми параметрами. ИМ участка включает следующие основные типы ресурсов:

a_Machine_Tool - ресурсы этого типа описывают станки, они имеют такие параметры, как номер, текущий и требуемый номера наладок, состояние, счетчик времени, необходимого для обработки всех деталей, ожидающих обработки на данном станке, суммарное время обработки и переналадки станка, номер функции критического отношения (одна из переменных оптимизации);

an_Order - ресурсы этого типа описывают заказы, их параметры включают исходные характеристики заказов из таблицы, переменные оптимизации (приоритет заказа и номер размера транспортной партии), состояние заказа, количество созданных и обработанных транспортных партий, параметры, используемые для оценки качества расписания - время изготовления заказа и опоздание;

a_Batch - ресурсы этого типа описывают транспортные партии деталей, они создаются в имитационной модели в момент, когда наступает время прибытия соответствующего заказа. Основные параметры ресурсов этого типа: уникальный номер в системе, номер заказа, номер типа деталей, количество деталей (размер заказа), количество выполненных операций, состояние заказа, номер станка, где находится транспортная партия; номер назначенной следующей операции, номер станка, назначенного для выполнения следующей операции, ряд параметров, хранящих характеристики соответствующего заказа и назначенной следующей операции.

Номер заказа	1	2	3	4	5	6	7	8	9	10
Номер типа детали	1	2	3	4	5	6	7	8	9	10
Количество деталей	50	100	100	400	100	150	100	100	50	100
Время прибытия, ч	0.0	8.0	0.0	0.0	8.0	8.0	0.0	0.0	16.0	8.0
Плановый срок, ч	36.0	66.0	34.0	62.0	38.0	58.0	70.0	54.0	38.0	44.0

Протекающий в производственной системе процесс описывается модифицированными производственными правилами на РДО-языке. Каждое правило описывает возможное действие в моделируемой системе. Для рассматриваемой задачи создана простая ИМ, в которой описаны следующие действия, имеющие место в реальной системе:

- обработка транспортной партии деталей на станке: длительность обработки вычисляется умножением количества деталей в транспортной партии, на штучное время для данной операции;

- транспортировка партий между станками или между центральным складом и станком; время транспортировки между каждой парой мест погрузки-выгрузки задается матрицей;

- переналадка станка; время переналадки зависит от номеров текущей и требуемой наладок и определяется по заданной матрице, одинаковой для всех станков;

- создание транспортных партий для прибывшего заказа в момент прибытия: количество транспортных партий и количество деталей в них определяется размером транспортной партии и количеством деталей в заказе (последняя транспортная партия содержит остаток заказа, и ее размер может быть меньше заданного).

Чтобы описать возможные операции в терминах модифицированных продукционных правил РДО-метода необходимо: перечислить ресурсы, которые участвуют в действии (релевантные ресурсы), записать арифметическое выражение для вычисления длительности действия: для каждого релевантного ресурса определить предусловия (Булево выражение, которое описывает условия, когда может начаться действие), и новые значения параметров после начала и окончания действия (арифметические выражения, чтобы вычислить новые значения). Ситуации, в которых необходимо принимать некоторое решение, называются точками решения. В ИМ мы рассматриваем следующие точки решения:

- после того, как выполнение некоторого действия завершилось, необходимо передать партию станку, назначенному для выполнения следующего действия. Если есть только одна возможность, то это сделать просто, но иногда имеется несколько возможностей для следующего действия. В этом случае мы выбираем операцию со станком, который имеет меньшее значение текущего полного времени процесса обработки на данном станке всех назначенных партий. Этот параметр станка вычисляется динамически в время имитации;

- когда некоторый станок закончил обработку партии, мы должны решить, какая следующая партия будет на нем обрабатываться. В этом случае приоритетная функция вычисляется для каждой партии, которая назначена на выполнение для данного станка, но еще не обработана, и затем выбирается партия с минимальным значением приоритета.

Эти точки решения также описываются в РДО-методе продукционными правилами. Имеется лишь небольшое различие между правилами, описывающими точки решения, и правилами имитации. Правило, описывающее процесс, содержит выражение для вычисления длительности операций и два конвертора - для начала и конца. Решающие же правила не содержат времени и имеют только один конвертор, они являются обычными продукционными правилами.

Функция приоритетов для выбора очередной партии, обрабатываемой на станке, базируется на следующих принципах. Все партии, назначенные на данный станок, разбиваются на несколько групп в зависимости от значения отношения времени, необходимого для завершения обработки партии, к времени, оставшемуся до планового срока. Большой приоритет дается тем партиям, для которых это значение отношения наименьшее. Внутри группы выбирается партия, требующая наименьшего времени переналадки. Поэтому, изменяя количество групп, можно придать большее значение минимизации опоздания партии (больше групп) или уменьшить время переналадки (меньше групп)

Блок оптимизации

Цель блока оптимизации - улучшение решения за счет подбора эвристик в приоритетах, при этом достижение действительного оптимума менее важно для таких сложных систем, как рассматриваемая производственная система. Применение точных методов оптимизации, помимо большой сложности, нецелесообразно еще и потому, что алгоритм планирования, который оптимизируется, сам по себе эвристический.

Использование методов направленного поиска не представляется эффективным в силу нерегулярности области определения функции качества. Методы случайного поиска не обеспечивают быстрой сходимости поиска, хотя и более экономичны, чем точные методы. Предлагается решать задачу оптимизации с помощью подхода, реализующего простейший генетический алгоритм (ПГА) [6,7].

ПГА хорошо зарекомендовали себя при решении сложных задач поиска в плохо обусловленных пространствах большой размерности, задач адаптации систем к изменяющейся среде и т.п. Они просты в вычислении, не накладывают

ограничений на пространство поиска, реализуют вероятностный характер поиска с накоплением информации в популяции о заранее неизвестном пространстве поиска. В этом они близки к алгоритмам искусственного интеллекта. Другой положительной для наших целей чертой ПГА является то, что они ищут семейство точек в пространстве поиска, а не одну точку. Последнее позволяет параллельно подниматься на несколько вершин пространства поиска (оптимизационные алгоритмы работают "от точки к точке").

В качестве параметров ПГА, посредством подбора которых обеспечивается быстрота стохастической сходимости поиска и соответственно эффективность процедуры планирования, могут использоваться:

- длина хромосомы L ;
- размер популяции N ;
- вероятность скрещивания P_c ;
- вероятность мутации P_m .

Оператор с помощью ЭС (или сама ЭС в автоматическом режиме) выполняет функции "селекционера", целенаправленно изменяя указанные параметры ПГА, а также применяя другие правила для сохранения особей с высоким значением функции пригодности из поколения в поколение, регулируя число особей с одинаковым значением функции пригодности, уменьшая количество вычислений для особей, перешедших из предыдущего поколения, и т.п.

Как уже говорилось, ПГА реализуется в единой среде с ИМ и ЭС на языке РДО. Соответственно его алгоритмы используют формализмы РДО-метода, рассматриваемые ниже.

Для реализации ПГА используются ресурсы трех типов:

ресурсы типа тип_Система (один), применяются для хранения общей и вспомогательной информации;

ресурсы типа Поколение, хранящие общую для одного поколения информацию;

ресурсы типа Особь, представляющие собой отдельные особи.

Трехоператорный ПГА выполняет итерации, включающие расшифровку битовой строки и расчет значений функций пригодности для всех особей, воспроизведение, скрещивание и мутацию. Отдельный блок предназначен для генерации первого поколения. Генерация поколений заканчивается при выполнении какого-либо условия. В описываемом примере программа заканчивает работу, сгенерировав заданное число поколений, в общем же случае прекращение итераций может быть любым.

С учетом особенностей РДО-языка для описания ПГА использовано 12 продукционных правил. Каждый блок, кроме блока скрещивания, реализуется двумя правилами, блок скрещивания - тремя. Еще одно правило добавлено для уничтожения старых поколений по мере генерации новых, что позволяет уменьшить время вычислений. В каждом блоке, описанном двумя правилами, одно правило реализует собственно действия данного блока над очередной особью, а другое переключает режим работы ПГА, если действия данного блока произведены над всеми особями. В блоке скрещивания используется та же схема, но основное действие блока реализуется двумя правилами.

Для реализации общего цикла ПГА и циклов по особям внутри каждого блока используются два параметра ресурса Система Режим и Счетчик. Рассмотрим эту схему подробнее на примере блока генерации первого поколения. В начальный момент значение параметра "Счетчик" равно нулю, параметра Режим "Инициализация" Правило Генерация первой популяции создает одну за другой особи первого поколения. Чтобы это правило применялось, необходимо одновременное выполнение трех условий:

текущий номер поколения должен быть меньше или равен заданному количеству поколений (это условие присутствует почти во всех правилах и введено для остановки ПГА, когда сгенерировано требуемое количество поколений;

остановка происходит потому, что ни одно правило не может быть применено, когда текущий номер поколения превышает требуемое количество поколений);

текущий режим работы ПГА должен быть равен параметру "Инициализация";

- значение параметра "Счетчик" должно быть меньше заданного размера поколения, это говорит о том, что еще не все особи первого поколения сгенерированы.

В конверторе правила значение параметра "Счетчик" увеличивается на 1. Это правило применяется последовательно до тех пор, пока не перестанет выполняться третья часть предусловия, т. е. пока значение параметра Система_Счетчик не станет больше или равно значению параметра Система_Размер_поколения. Таким образом будет создано нужное количество особей первого поколения.

Предусловие второго правила блока генерации первого поколения Конец генерации первого поколения отличается только тем, что значение параметра "Счетчик" должно быть равно заданному размеру поколения, это говорит о том, что все особи первого поколения сгенерированы. В конверторе этого правила параметр Система_Счетчик обнуляется, и значение параметра Система_Режим устанавливается равным параметру "Расшифровка_и_расчет_ФП" (ПГА переходит к блоку расшифровки битовой строки и расчета функции пригодности). Заметим, что здесь рассмотрены не все действия, предусматриваемые правилами блока генерации первого поколения, а лишь часть, связанная с организацией вычислений.

В общем случае правила, выполняющие основные функции блоков, увеличивают значение счетчика особей и применяются до тех пор, пока значение счетчика не станет равным заданному количеству особей в поколении. Правила, заканчивающие текущий блок, обнуляют параметр Счетчик для его корректного использования в следующем блоке и изменяют режим, после чего начинают выполняться условия применимости правил другого блока. Параметр Система_Режим может принимать пять значений, каждое из которых соответствует блоку ПГА: "Инициализация", "Расшифровка_и_расчет_ФП", "Воспроизведение", "Скрещивание", "Мутация"

Результаты работы ГС

С помощью ПГА проводилась оптимизация расписания для ТК, состоящего из семи станков, портфель заказов содержал 10 позиций, всего над деталями выполнялось 64 операции (от 3 до 13 операций на деталь). Управляемыми переменными при оптимизации расписания являлись приоритеты заказов. Приоритет заказа был задан целым числом в диапазоне от 1 до 8, увеличение значения приоритета ускоряет обработку заказа. Приоритет кодировался тремя битами, длина битовой строки-хромосомы равнялась 30 битам. Функция оценки качества получаемого расписания S представляла собой сумму штрафов для каждого заказа. Значение штрафа вычислялось следующим образом. Если заказ опоздал к плановому сроку, то штраф равнялся значению величины опоздания в квадрате, если изготовление заказа завершалось раньше планового срока, но не более, чем на 4 часа, штраф равнялся нулю, если же опережение было больше четырех часов, то штраф равнялся значению величины опережения минус 4 часа. Задача оптимизации формулировалась следующим образом: найти сочетание приоритетов заказов, минимизирующее значение штрафной функции.

Функция пригодности особей вычислялась по следующей формуле: $f(w_i) = \frac{1}{1 + S/w_i}$,

где S - функция оценки качества расписания.

Поскольку функция оценки качества расписания неотрицательна, функция пригодности меняется от 1 при $S=0$ (наилучшее расписание) до 0 при S , стремящемся к бесконечности.

Наилучшие результаты получаются при $N=200$, лучшее значение $f(1W_i, 0)$ равно 0,654, что соответствует значению функции S , равному 10,58. Среднее значение функции пригодности увеличивается за 10 поколений от 0,177 до 0,455.

Заключение

Как видно из приведенного материала, язык РДО обеспечивает реализацию ГС, имеющей в составе ИМ,ЭС и блок оптимизации и, следовательно, обладает большой выразительной мощностью. ПГА позволяет проводить оптимизацию в столь сложной задаче как составление расписания, однако требуется дополнительное исследование по выбору его параметров, обеспечивающих более высокую сходимость и эффективность, поскольку при решении этой и подобных задач оптимизации большую часть времени занимает расчет функций пригодности, что делает весьма актуальным минимизацию общего числа особей за все время эволюции популяции.

ЛИТЕРАТУРА

1. Горнев В.Ф., Емельянов В.В., Овсянников М.В. Оперативное управление в ГПС. М.: Машиностроение., 1990. 341с.
2. Емельянов В.В., Ясиновский С.И. Представление знаний о дискретном производственном процессе в системах моделирования и управления // Приборы и системы управления, 1991, N 6, с. 1-3.
3. Емельянов В.В., Ясиновский С.И. Продукционный имитатор дискретных процессов и систем // Вестник машиностроения, 1992, N 5, с. 41-44.
4. Ясиновский С.И. Логический вывод в гибридных системах // Вестник МГТУ. Сер. Приборостроение". 1994. N 1, с. 88-95.
5. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. - М.: Наука, 1975. - 358 с.
6. Holland J.N. Adaptive algorithm for discovering and using general patterns in groving knowledge-bases // Int. Journ. of Policy Analysis and Information Systems, 1980, P. 217-240.
7. Holland J.N. Escaping Britteness: The possibilities of General-purpose Learning Algorithms Applied to Parallel Rule-Based System // Machine Learning: An artificial intelligence approach / R.S. Michalski, J.G.Carbonell, T.M.Mitchell (eds). Los Altos, CA: Morgan Kaufmann, 1986. Vol.2. P. 593-623.

УДК 681.3 - 519.8

В.М. Курейчик, В.П. Карелин, В.М. Глушань **Специализированные средства аппаратной поддержки интеллектуальных САПР и систем принятия решений**

Интеллектуализация САПР предполагает имитацию деятельности конструктора в части накопления знаний об используемых алгоритмах проектирования, их модификации, способах оптимизации, проверки корректности проектируемых схем и т.п. [1].

Процесс проектирования один из наиболее интеллектуальных видов деятельности человека, поэтому уровень требований, предъявляемых к интеллектуальным САПР, превышает современный уровень технологии обработки информации. Автоматизировать удастся только часть процесса проектирования.

Конструктор в процессе проектирования использует различные функции интеллекта: интуицию, творчество, ассоциацию, индукцию, силлогизмы,