

УДК 004.822

**РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО РЕШЕНИЯ  
ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ В САПР, ОСНОВАННОЙ НА МЕТОДЕ  
ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ****Кривевич С.Д., Григорьев А.В.***Донецкий национальный технический университет г. Донецк**Кафедра прикладной математики и информатики**E-mail: [krilevich@gmail.com](mailto:krilevich@gmail.com)***Аннотация**

**Кривевич С.Д., Григорьев А.В. Разработка системы автоматизированного решения вычислительных задач САПР, основанной на методе программирования в ограничениях.** В предлагаемой работе рассматриваются методы реализации инструментальной системы для решения задач программирования в ограничениях в различных постановках, возникающих в процессе работы проблемно-ориентированных САПР. Система обеспечивает взаимосвязь с различными проблемно ориентированными САПР за счет унификации интерфейса.

**1. Общая постановка проблемы**

Тенденции развития современных САПР приводят ко все большей унификации методов их построения. В настоящее время широко распространение приобретают гибридные САПР, Широко применяются различные базы знаний, системы распознавания образов, логики пространства-времени и т.д. Однако, наиболее востребованными в САПР являются методы обеспечения автоматизации расчетной части, что вызвано большим объемом выполнения расчетов в среде любого САПР. К интеллектуальным методам для решения вычислительных задач САПР можно отнести: нейронные сети, программирование в ограничениях, различные вычислительные сети. Можно отметить, что из перечисленных средств наиболее ориентированы на специфику САПР два последних компонента, а именно - программирование в ограничениях и различные вычислительные сети. Учитывая, что метод программирования в ограничениях является более общим методом, остановим свое внимание на нем. Дадим краткую характеристику метода программирования в ограничениях и выполним анализ уровня его востребованности в современных САПР. Программирование в ограничениях основано на описании модели задачи. Модель представляется в виде совокупности отношений, связывающих параметры задачи. Значения параметров задает пользователь, причем они могут принимать точные, известные значения или не известные, а также интервальные значения, заданные в виде ограничений. Используя описание модели задачи и информацию о значениях параметров, методы программирования в ограничениях (МПО) обеспечивают автоматическое нахождение решения.

Постановка задачи в общем виде, сформулированная в [1], выглядит следующим образом. Пусть на параметры  $x_1, x_2, \dots, x_n$ , областями значений которых являются множества  $X_1, X_2, \dots, X_n$ , заданы ограничения  $C_i(x_1, x_2, \dots, x_n), i=1..k$ . Требуется найти наборы значений  $\langle a_1, a_2, \dots, a_n \rangle (a_i \in X_i)$ , которые бы удовлетворяли всем ограничениям одновременно. Такая постановка задачи называется проблемой удовлетворения ограничений и решается при помощи различных методов и алгоритмов [2-3].

Одним из наиболее развитых подходов в программировании в ограничениях, являются недоопределенные модели, на базе которого уже реализованы различные системы, такие как UniCalc, Немо-Тек, Time-Ex. Из зарубежных систем реализующих парадигму программирования в ограничениях можно выделить такие как Prolog III, ILOG Solver, CHIP и другие. В современных САПР имеется ряд задач, где может быть применен МПО. Однако,

перечисленные выше системы не обеспечивают эффективного решения полного класса задач, подпадающих под возможности методов программирования в ограничениях, характерных для САПР.

Авторами ранее был предложен ряд методов и подходов, обеспечивающих решение задачи построения системы программирования в ограничениях, ориентированной на специфику САПР [4-7]. Однако, осталась не решенной задача построения единого программного комплекса, обеспечивающего решение всего набора задач, возникающих в САПР, для которых необходимо применение метода программирования в ограничениях.

## **2. Постановка задач исследования**

Задачей исследования является построение комплексной системы программирования в ограничениях, ориентированной на решение различных вычислительных задач в САПР. Фактически, для решения поставленной задачи необходимо решить следующие подзадачи:

- 1) Определить типы задач, возникающие в САПР, и сформулировать постановки данных задач;
- 2) Определить структур данных модели;
- 3) Разработать общую динамическую вычислительную модель, способную обеспечить решение всех типов поставленных задач;
- 4) Разработать общие алгоритмы решения различных типов задач в рамках определенных структур модели.

## **3. Решение задачи и результаты исследований**

### **3.1. Типы задач разрабатываемой системы программирования в ограничениях**

Определим типы задач, которые будет решать разрабатываемая система программирования в ограничениях должна:

1) **Доопределение неизвестных значений по имеющимся значениям параметров:** Соответствует классической задаче построения и использования вычислительных сетей, имеет два режима – общий и локальный.

2) **Доопределение неопределенной модели:** Соответствует классической постановке задачи метода программирования в ограничениях, при котором на основании недоопределенной модели (с интервальными или точными значениями) требуется получить более доопределенную модель (с интервальными или точными значениями). Итоговая модель отличается более суженными интервалами.

3) **Отслеживание влияния одних параметров на другие:** - на основании недоопределенной модели (с интервальными или точными значениями), определенными для важнейших параметров, исследовать возможные границы значений зависимых параметров.

4) **Проверка коллизий, поиск их источников и вариантов устранения:** По модели с точными значениями выполнить пересчет - проверку (удовлетворения системе ограничений); цель перерасчета – поиск некорректных соотношений параметров.

5) **Преобразование доопределенной модели в модель с точными значениями:** Предполагает экспертное преобразование модели.

Определим постановку каждой задачи в рамках разрабатываемой системы в соответствии с ее номером в списке задач:

1) На входе – модель, у которой определена только часть параметров и система отношений между всеми параметрами. Задача – доопределить неизвестные параметры. Результат – модель, у которой все параметры имеют значения (точные или интервальные).

2) На входе - недоопределенная модель с набором параметров и совокупностью отношений, связывающих эти параметры. Задача - доопределить параметры модели в соответствии с системой отношений, уменьшив интервалы значений параметров. Результат - модель с более узкими интервалами значений параметров.

3) На входе - недоопределенная модель с параметрами и системой отношений, приоритетные параметры, для которых мы отслеживаем влияние. Задача – на основе

приоритетных параметров определить возможные интервальные значения остальных параметров. Оценить степень влияния приоритетных параметров на изменение остальных. Результат – модель с доопределенными интервалами значений зависимых параметров и оценка степени влияния приоритетных параметров.

4) На входе - модель с точными значениями параметров. Задача – пересчет и проверка совпадения значений параметров в соответствии с системой отношений, поиск коллизий. Результат – сообщение о корректности исходной модели и, в случае возникновения коллизий, указание их источников с предоставлением способа разрешения.

5) На входе - модель с интервальными значениями параметров и система отношений. Задача – нахождение точных значений каждого параметра в соответствии с системой ограничений. Результат – модель с точными значениями параметров.

### **3.2. Предлагаемые методы решения задач разрабатываемой системы программирования в ограничениях**

Предлагается следующий порядок решения задачи: 1) Определение структур данных модели; 2) Разработка общей динамической вычислительной модели, способной обеспечить решение всех типов поставленных задач; 3) Разработка общих алгоритмов решения различных типов задач в рамках определенных структур модели.

#### **3.2.1. Определение структур данных модели**

Определим структуры данных модели, которые будет использовать система.

Вся модель представляет собой набор объектов, набор точек стыков (ТС) и набор связей между этими точками. Структурно это выглядит следующим образом:

{Модель: (ОБЪЕКТ<sub>1</sub>, ..., ОБЪЕКТ<sub>n</sub>), (ТС<sub>1</sub>, ..., ТС<sub>m</sub>), (СВЯЗЬ<sub>1</sub>, ..., СВЯЗЬ<sub>l</sub>)}.

Объект состоит из набора характерных точек (ХТ) и набора связей между этими точками. Структурно это выглядит так: {Объект : (ХТ<sub>1</sub>, ..., ХТ<sub>n</sub>), (СВЯЗЬ<sub>1</sub>, ..., СВЯЗЬ<sub>m</sub>)}.

Точки стыков являются частными случаями характерных точек, являясь, по сути, ссылками на характерные точки объектов в которых производится стыковка объектов при помощи определенного типа структурной связи (typeСвязи). Структурно это выглядит так: {ТС : ХТ }.

Характерная точка представляется собой набор параметров и тип самой точки (typeХТ). Сами параметры могут принимать интервальные, точные и неопределенные значения. Структурно это выглядит так: {ТС: (X<sub>1</sub>, ..., X<sub>n</sub>), typeХТ }

Связи представляют собой набор характерных точек (или точек стыка) и тип связи. Связи могут быть внутренними, т.е. связывающими характерные точки внутри объекта, и внешними, т.е. связывающими точки стыков различных объектов. Структурно это выглядит так: {СВЯЗЬ: (ХТ<sub>1</sub>,...,ХТ<sub>n</sub>), typeСвязи}.

Связи определяют совокупности функций (отношений) между параметрами характерных точек. Наборы этих функций (системы функций) определяются по типам характерных точек и типу их связи. Структурно система функций выглядит так:

{R: (typeХТ<sub>1</sub>, ..., typeХТ<sub>n</sub>), typeСвязи, (ФУНКЦИЯ<sub>1</sub>, ..., ФУНКЦИЯ<sub>m</sub>)}.

Сами функции представляются в виде набора аргументов, их числа (Num) и приведенного к некоторому общему виду выражения (exp), т.е. однородного уравнения – отношений этих аргументов (ARG<sub>i</sub>). Структурно функции выглядят так:

{ФУНКЦИЯ: (ARG<sub>1</sub>, ..., ARG<sub>n</sub>), Num, exp }.

#### **3.2.2. Разработка общей динамической вычислительной модели.**

Общая вычислительная модель  $M=(X=(X^1, X^i, X^0), R)$  состоит из множеств:

▪  $X$  – множество параметров, значения которых могут принимать различный вид:

–  $X^1$  – обозначает одно точное значение;

–  $X^i$  – обозначает, что значение параметра задано в интервальном виде;

–  $X^0$  – обозначает не определенное значение, в этом случае принято считать границы интервалов такого значения как бесконечные;

- $R$  – множество отношений, связывающих эти параметры

Здесь задание параметра в интервальном виде -  $X^1$  подразумевает, что значение ограничивается двумя интервалами. Например,  $x = [a, b]$  – означает, что параметр  $x$ , принимает значение в интервале от  $a$  до  $b$ , где  $a$  – нижний интервал, а  $b$  – верхний, т.е. значение параметра ограничено неравенством:  $a \leq x \leq b$ .

### 3.2.3. Разработка общих алгоритмов решения различных типов задач в рамках определенных структур модели.

Приведем общие алгоритмы решения задач в рамках определенных структур модели. Для начала необходимо привести структурную модель к виду общей динамической вычислительной модели (ДВМ), в рамках которой будут решаться задачи методом программирования в ограничениях. Суть построения ДВМ сводится к определению двух множеств – множества параметров ( $X$ ) и множества отношений между ними ( $R$ ), также подразумевается и третье множество – это множество ограничений ( $I(x)$ ), обозначающее интервалы значений известных параметров, если они не определены, то принимают бесконечные интервалы значений. Обобщенный алгоритм построения такой модели приведен далее.

**Алгоритм 1.** Построения ДВМ по структурной модели.

Исходные данные - набор ТС и набор связей между ТС.

Обозначения в ДВМ:  $X$  - множество параметров,  $X'$  – подмножество параметров,  $R$  - множество отношений (система функций),  $R'$  – подмножество отношений.

*Этап 1:* обработка внешних связей объектов

Для каждой связи исходной модели выполняются следующие действия:

- определяется тип текущей связи;
- Для каждой ТС (или ХТ):
  - определяется тип характерной точки;
  - выбирается множество параметров характерной точки –  $X'$ ;
  - по типам характерных точек и типу связи определяется система функций –  $R'$ .

$\Rightarrow X = X \cup X'$ ;  $R = R + R'$ .

*Этап 2:* обработка внутренних связей объектов.

Для каждого объекта исходной модели:

- Повторяется Этап 1;

$\Rightarrow X = X \cup X'$ ;  $R = R + R'$ .

Таким образом, формируется множество аргументов и множество систем функций для исходной модели, причем каждое отношение представленное уравнением в структурной модели интерпретируются в набор уравнений, для каждого из ее аргументов и называются функциями интерпретации. Такие интерпретации функций можно называть ограничениями.

Алгоритм решения определенных типов задач зависит от самой модели, сформированной в рамках постановки задачи. В общем же виде задача доопределения исходных моделей решается при помощи методов программирования в ограничениях. Такая задача называется «Задачей удовлетворения ограничений».

Приведем общий алгоритм решения задачи удовлетворения ограничений на определенной вычислительной модели.

**Алгоритм 2.** Удовлетворение ограничений.

Исходные данные:  $X$  – множество параметров,  $R$  – множество отношений.

Обозначения:  $x^{(t)}$  – вектор недоопределенных значений на шаге  $t$ ,  $Q^{(t)}$  – множество активных ограничений на шаге  $t$ ,  $R(x^{(t)})$  – решение функции  $c$  из множества  $R$  над вектором  $x^{(t)}$ , результатом которого является ограничение, представляющее собой интервал  $[a, b]$ .

Шаг 0:  $x^{(0)} := X$ ;  $Q^{(0)} := R$ ;

Шаг  $t+1$ : Если  $Q^{(t)} = 0$ , то СТОП

Иначе: выбираем  $c$  из  $Q^{(t)}$ ,  $x^{(t+1)} := R_c(x^{(t)})$

Если значение интервала уменьшилось, то:

$$Q^{(t+1)} := Q^{(t)} \cup \{d \in R(x^{(t)}) \mid x^{(t+1)} \in \arg d, d \neq c\} \setminus \{c\}$$

// добавить в список активных ограничений функции, аргументом которых является текущий параметр, и убрать из списка текущую функцию.

Таким образом, производится пересчет значений параметров на наборе функций до тех пор, пока он уменьшает интервалы значений.

Аналогичным образом строятся алгоритмы для решения отдельных задач. В частности, для решения задачи доопределения неизвестных параметров по известным, производится обходом структурной модели по связям в ТС, каждая такая связь представляется в виде ДВМ и производится поиск решений. А при локальном режиме данной задачи, пользователь сам выделяет части структурной модели, на которых необходимо произвести расчеты. Более детальный алгоритм подразумевает проверку полученных расчетов на коллизии, возникающие, когда значения параметров образуют пустое множество и отслеживание параметров непосредственно влияющих на возникновения таких несоответствий.

### **Выводы**

В работе выполнен анализ проблем применения методов программирования в ограничениях в САПР. Определены постановки задач разрабатываемой системы программирования в ограничениях. Определены структуры данных ее модели. Разработана общая динамическая вычислительная модель, способная обеспечить решение всех типов поставленных задач. Определены общие алгоритмы решения задачи в рамках этой системы. Перспективным направлением работы является разработка программного комплекса реализующего работу данной системы и обеспечивающего решения поставленных задач в САПР.

### **Список литературы**

1. Нариньяни А.С., Телерман В.В., Ушаков Д.М., Швецов И.Е. Программирование в ограничениях и недоопределенные модели // Информационные технологии №7, 1998. М., Издательство “Машиностроение”. - С. 13-22.
2. Телерман В.В., Ушаков Д.М. Удовлетворение ограничений в задачах математического программирования // Вычислительные технологии. 1998. Т. 3. № 2. С. 45-54
3. Нариньяни А.С. Недоопределенность в системах представления и обработки знаний // Изв. АН СССР. Техн. кибернетика, 1986. № 5. – С. 3 – 28.
4. Григорьев А.В. Методы построения функций в специализированной оболочке для создания интеллектуальных САПР // Искусственный интеллект. – Донецк, 2001 – №3 – С. 40–53.
5. Григорьев А.В. Методы и средства работы с графическими моделями в САПР парогазовых установок СПРУТ / Моделирование и компьютерная графика: Материалы 1-й международной научно-технической конференции, г Донецк, 04-07 октября 2005 г. — Донецк, ДонНТУ, Министерство образования и науки Украины, 2005. – С. 77-86. 1-я международная научно-техническая конференция «Моделирование и компьютерная графика», 04-07 октября 2005 г., г. Донецк, Украина.
6. Григорьев А.В. Решение дифференциальных уравнений в интеллектуальных САПР методом программирования в ограничениях // Научные труды Донецкого национального технического университета. Выпуск 70. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2003) – Донецк: ДонНТУ, 2003. – С. 108–116.
7. Григорьев А.В., Бондаренко А.В., Шойхеденко А.В. Интерфейс табличного процессора EXCEL и специализированной оболочки для синтеза интеллектуальных САПР и АСНИ. В кн. Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник трудов ДонГТУ, Выпуск 1. Донецк: ДонГТУ, 1997. - С. 229-238.