

**А.В. Григорьев** (канд.техн.наук, доц.), **С.Д. Крилевич**  
Донецкий национальный технический университет  
grigorie@r5.dgtu.donetsk.ua, krilevich@gmail.com

## **СИСТЕМА АВТОМАТИЗИРОВАННОГО РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ В САПР, ОСНОВАННАЯ НА МЕТОДЕ ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ**

Рассматриваются методы реализации инструментальной системы для решения задач программирования в ограничениях в различных постановках, возникающих в процессе работы проблемно-ориентированных САПР.

**САПР, вычислительная модель, классификация задач, общие методы решения**

### ***Вступление***

Тенденции развития современных САПР приводят ко все большей унификации методов построения САПР, не зависимо от предметной области, а также – ко все большей их интеллектуализации. В настоящее время широкое распространение приобретают гибридные САПР, включающие как «классические», так и интеллектуальные компоненты. В гибридных САПР широко применяются различные базы знаний, системы распознавания образов, логики пространства-времени и т.д. Однако, наиболее востребованными из методов искусственного интеллекта в современных САПР являются методы обеспечения автоматизации расчетной части, что вызвано большим объемом выполнения расчетов в среде любой САПР. К интеллектуальным методам для решения вычислительных задач САПР можно отнести: различные вычислительные и нейронные сети, а так же теорию неопределенного программирования. В частности, теория неопределенного программирования включает такие разделы программирования как: неопределенное, случайно-нечеткое, нечетко-случайное, неточное, нечеткое [1] и т.д. Не умаляя важности всех названных разделов для практики САПР, тем не менее, отметим, что наиболее ориентирован на специфику САПР такой раздел теории неопределенного программирования, как программирование в ограничениях. В связи с этим все работы, направленные на повышение уровня эффективности применения метода программирования в ограничениях в САПР, имеют безусловную актуальность.

## **Общая постановка проблемы**

Дадим краткую характеристику метода программирования в ограничениях и выполним анализ уровня его востребованности в современных САПР.

Программирование в ограничениях основано на описании модели вычислительной задачи. Такая модель представляется в виде совокупности отношений, связывающих параметры задачи. Значения параметров задает пользователь, причем они могут принимать либо точные, т.е. известные значения, либо – не известные, либо – 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 [2], 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 [3-4].

Одним из наиболее развитых подходов в программировании в ограничениях, являются недоопределенные модели, на базе которых уже реализованы различные универсальные или узкоспециализированные 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 для полного класса задач, подпадающих под возможности методов программирования в ограничениях, характерных для САПР.

С другой стороны, следует отметить существование ряда средств решения вычислительных задач в САПР, представляющих собой инструментальные среды для автоматизации создания и применения баз знаний, построенных на основе системы вычислительных модулей знаний. Основой данных систем чаще являются динамические вычислительные сети, т.е. более простые в математическом отношении модели вычислений. Т.о., такие системы не обеспечивают в полной мере высокую эффективность решения вычислительных задач. Например, система SprutExPro [5] при общей высокой эффективности, кроме названных, обладает дополнительно еще такими недостатками:

- необходимость явно формировать в среде инструментального комплекса базу знаний вычислительных моделей для решения того или иного класса вычислительных задач в рамках графической модели данного объекта;
- необходимость явно ставить вычислительную задачу.

«Идеальная» система такого класса должна обеспечивать:

- автоматическое формирование математической модели объекта по его структуре, т.е. структурной модели объекта как совокупности объектов, связанных структурными связями;

- вычислительные задачи в рамках данной структурной графической модели должны ставиться и решаться автоматически, исходя из заданной структурной модели объекта и ряда типичных этапов построения и исследования модели.

Авторами ранее был предложен ряд методов и подходов, обеспечивающих решение задачи построения системы программирования в ограничениях, ориентированной на специфику САПР [6-9], лишенную указанных недостатков. 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.

### ***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 структуры данных моделей вычислительных задач;
- 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 данных модели.

### ***Решение задачи и результаты исследований***

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

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

1) ***Доопределение неизвестных, но точных значений всех параметров по подмножеству параметров, имеющих точные известные значения.*** 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 значений всех параметров.

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 коллизий, т.е. - некорректных соотношений параметров с последующим их устранением.

5) **Преобразование доопределенной модели в модель с точными значениями.** Предполагает экспертное преобразование модели с интервальными параметрами в модель с точными значениями параметров.

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

1) На входе – модель, у которой задан набор параметров и система отношений между всеми параметрами, при этом определена, т.е. имеет точные или интервальные значения, только часть параметров. Прочие параметры – неопределенны. 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, а так же - 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 параметров. 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 непустоты значений параметров в соответствии с системой отношений, фактически это означает поиск коллизий, т.е. – противоречий в требованиях значений параметров, исходя из различных отношений. 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.

## **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.

### **Определение структур данных модели**

Определим структуры данных модели, которые будет использовать инструментальная система. Предлагаемые структуры данных есть отражение предлагаемой модели мира, описывающей структурные и топологические отношения [10]. Иначе говоря, рассматривается обобщенная модель структуры объектов проектирования количественного уровня абстракции.

Вся модель представляет собой набор объектов, набор точек стыков (ТС) между данными объектами, а так же набор структурных связей (далее - связи) между этими точками стыка. Структурная связь подразумевает некоторую совокупность математических отношений между параметрами, входящими в ТС. В виде структуры данных модель 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>i</sub>)}.

Собственно объект состоит из набора характерных точек (ХТ) и набора связей между этими точками. Набор ХТ позволяет однозначно построить графическую модель объекта в соответствующем графическом редакторе, где каждому типу объекта соответствует библиотечная программа трехмерной (или плоской) визуализации данного объекта. 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 подмножество характерных точек объектов, для которых производится стыковка при помощи определенного типа структурной связи (typeСвязи). Структурно это выглядит так: {ТС : ХТ<sub>k</sub>}.

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

Связи представляют собой тип связи, а так же набор характерных точек (или точек стыка). 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:

$\{R: (\text{typeXT1}, \dots, \text{typeXTn}), \text{typeСвязи}, (\text{ФУНКЦИЯ1}, \dots, \text{ФУНКЦИЯm})\}$ .

Сами функции представляются в виде набора аргументов (ARGi), с указанием их числа (Nam), а так же выражения (exp), приведенного к некоторому общему виду (однородному уравнению), задающего отношение между этими аргументами. Структурно функции выглядят так:

$\{\text{ФУНКЦИЯ}: (\text{ARG1}, \dots, \text{ARGn}), \text{Num}, \text{exp}\}$ .

### ***Разработка общей динамической вычислительной модели***

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

▪  $X$  – множество параметров, значения которых могут принимать такие различные виды:

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

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

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

▪  $R$  – множество отношений, связывающих эти параметры.

Здесь задание параметра в интервальном виде -  $X^i$  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:  $a \leq x \leq b$ .

### ***Разработка общих алгоритмов решения различных типов задач в рамках определенных структур данных модели***

Приведем общие алгоритмы решения задач в рамках определенных структур данных 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 упомянутые интервалы не определены, то считаем, что они 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:

1) Определяется тип текущей связи;

2) 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. При этом каждое отношение, представленное уравнением в структурной модели, интерпретируются в набор уравнений для каждого из ее аргументов, называемых 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 обход структурной модели по связям в ТС, каждая такая связь представляется в виде ДВМ и производится поиск решений. 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;
- 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. Лю Б. Теория и практика неопределенного программирования / Б. Лю ; пер. с англ. – М.: Бином. Лаборатория знаний, 2005. - 416 с.: ил. (Адаптивные и интеллектуальные системы).
2. Нариньяни А.С. Программирование в ограничениях и недоопределенные модели / А.С.Нариньяни, В.В.Телерман, Д.М.Ушаков, И.Е.Швецов // Информационные технологии. - 1998. - №7. - С. 13-22.
3. Телерман В.В. Удовлетворение ограничений в задачах математического программирования / В.В.Телерман, Д.М. Ушаков // Вычислительные технологии. 1998. - Т. 3. - № 2. - С. 45-54.
4. Нариньяни А.С. Недоопределенность в системах представления и обработки знаний / А.С. Нариньяни // Изв. АН СССР. Техн. кибернетика. - 1986. - № 5. – С. 3 – 28.
5. Евгеньев Г. SprutExPro - SprutExPro: программирование для непрограммистов OMCON / Г. Евгеньев, С. Борисов // САПР и Графика. - 2002. - №1.
6. Григорьев А.В. Методы построения функций в специализированной оболочке для создания интеллектуальных САПР / А.В. Григорьев // Искусственный интеллект. – 2001. – №3. – С. 40–53.
7. Григорьев А.В. Методы и средства работы с графическими моделями в САПР парогазовых установок СПРУТ / А.В. Григорьев // Моделирование и компьютерная графика: материалы 1-й международной научно-технической конференции, г. Донецк, 04-07 октября 2005 г. - 2005. – С. 77-86.
8. Григорьев А.В. Решение дифференциальных уравнений в интеллектуальных САПР методом программирования в ограничениях / А.В. Григорьев // Научные труды



Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника». – 2003. - Выпуск 70. – С. 108–116.

9. Григорьев А.В. Интерфейс табличного процессора EXCEL и специализированной оболочки для синтеза интеллектуальных САПР и АСНИ / А.В.Григорьев, А.В.Бондаренко, А.В.Шойхеденко // Информатика, кибернетика и вычислительная техника (ИКВТ-97). - 1997. – Вып. 1. - С. 229-238.
10. Григорьев А.В. Комплекс моделей САПР как система взаимосвязанных уровней о действительности / А.В. Григорьев // Научные труды Донецкого государственного университета. Серия: Информатика, кибернетика и вычислительная техника. - 2000. – Вып. 10. - С. 155-167.

*Надійшла до редакції 20.09.2010*

*Рецензент: канд.техн.наук, доц. Костюкова Н.С.*

**О.В. Григор'єв, С.Д. Крилевич**

Донецький національний технічний університет

**Система автоматизованого вирішення обчислювальних завдань в САПР, заснована на методі програмування в обмеженнях.** Розглядаються методи реалізації інструментальної системи для вирішення завдань програмування в обмеженнях у різних постановках, що виникають у процесі роботи проблемно-орієнтованих САПР.

**САПР, обчислювальна модель, класифікація завдань, загальні методи вирішення**

**A.V. Grigoriev, S.D. Krilevich**

Donetsk National Technical University

**The System of Automated Solutions of Computational Problems in CAD Based on the Method of Constraint Programming.** The paper consider the methods of implementing software tool for solving constraint programming in various formulations that arise in the process of problem-oriented CAD.

**CAD, computational model, the classification task, the general methods of solution**