

Рис. 5. Экспериментальное сравнение эффективности предлагаемого метода и эволюционного метода при поиске цепей $\pi(n, n+1)$ минимального веса

дующие правила. Алгоритм инициализировался путем построения популяции здоровых хромосом, которые являются участками цепи, представляющей собой статистический глобальный оптимум. Роль генетических операторов выполняли операторы перекрестного скрещивания участков двух и более хромосом между собой. Хромосомы с наименьшим весом относительно своей длины составляли элиту и подвергались мутации с меньшей вероятностью, чем хромосомы с большим относительным весом. Каждое поколение содержало набор здоровых хромосом, представляющих собой правильную цепь $\pi(n, n+1)$ того или иного веса $h(\pi(n, n+1))$. Такой эпистазис дал возможность просматривать текущее решение в произвольный момент без остановки ГА. Продолжительность работы ГА определялась временем работы предлагаемого алгоритма построения ОГО, оценивающего q решений, $q=n, 5n, 10n$, n – число вершин исходного графа. Таким образом, для каждого $n=20,$

$30, \dots, 120$ ГА возвращал три решения в моменты получения q решений, $q=n, 5n, 10n$, алгоритмом построения ранжированной ОГО.

На основании изложенного можно сделать следующие выводы. Предложенный метод построения окрестности глобального оптимума NPC-задач на ранжированном дереве поиска решений превосходит быстрые эвристики Шермана–Рейтера и Лина и известные методы ИИ по точности решения и может применяться как самостоятельный инструмент при решении NPC-задач.

Достоинством предлагаемого метода является постепенное улучшение текущего субоптимального решения, что позволяет завершать поиск либо по времени, либо по количеству/доле просмотренных цепей, либо по достижению заданного порога весовой функции, либо по завершении построения окрестности глобального оптимума. Недостатком метода являются дополнительные вычислительные затраты, обусловленные ранжированием локальных решений для каждой вершины исходного графа и использованием экспертно задаваемого порога оценки веса глобального решения.

Литература

1. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. СПб: БХВ-Петербург, 2003. 1104 с.
2. Корн Г., Корн Т. Справочник по математике. Определения, теоремы, формулы. СПб: Изд-во «Лань», 2003. 832 с.
3. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность; пер. с англ. М.: Мир, 1985.
4. Макконелл Дж. Основы современных алгоритмов; пер. с англ. М.: Техносфера, 2004. 368 с.

ТЕХНОЛОГИЯ МЯГКИХ ВЫЧИСЛЕНИЙ В ПРОЕКТИРОВАНИИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ

А.А. Мишин (Научно-технологический парк «Дубна», г. Дубна, a.mishin@ntpдubna.ru);
В.Н. Добрынин, к.т.н.; Л.В. Литвинцева, к.ф.-м.н.
(Университет природы, общества и человека «Дубна», г. Дубна)

На конкретных типовых примерах моделей динамических объектов управления демонстрируется эффективность процессов управления в условиях неполноты информации о параметрах структуры объектов управления и непредвиденных (нештатных) ситуаций управления с применением оптимизатора баз знаний на мягких вычислениях. Показано, что использование его в обучении и непредвиденных ситуациях управления приводит к повышению уровня робастности структуры интеллектуальных систем управления.

Ключевые слова: оптимизатор баз знаний, робастность, интеллектуальные системы управления, мягкие вычисления.

SOFT COMPUTING TECHNOLOGY FOR INTELLIGENT CONTROL SYSTEM DESIGN

Mishin A.A. (Technopark «Dubna», Dubna, a.mishin@ntpдubna.ru); Dobrynin V.N., Ph.D.; Litvintseva L.V., Ph.D. (Dubna International University for Nature, Society and Man, Dubna)

Abstract. Effective control processes are demonstrated using Benchmarks of control object models in uncertainty of information about structure parameters and unpredicted control situations based on soft computing knowledge base optimizer (KBO). It is demonstrated that the using of KBO in learning situations and unpredicted control situations increasing robustness levels of intelligent control systems.

Keywords: knowledge base optimizer, robustness, intelligent control systems, soft computing.

В системах проектирования БЗ интеллектуальных систем управления (ИСУ) построение оптимальной структуры нечеткой нейронной сети (ННС) возложено на опытного эксперта. Но даже для него выбор модели нечеткого вывода и лингвистическое описание заданного обучающего сигнала вручную в сложных ситуациях управления является большой проблемой.

Проблематичным остается также определение требуемого соотношения между точностью аппроксимации обучающего сигнала и необходимым уровнем робастности всей структуры ННС. Однако эти проблемы можно решить с помощью программных средств инструментария, названного *оптимизатором БЗ* (ОБЗ) [1, 2].

В данной статье рассматривается программная архитектура, принятая за основу ОБЗ. Внимание сконцентрировано и на описании конкретных результатов моделирования ИСУ сложными существенно-нелинейными, динамически неустойчивыми объектами управления (ОУ).

Программная реализация оптимизатора БЗ

Для построения ОБЗ ИСУ была выбрана модульная схема, которая подразумевает построение пользователем модели ИСУ из набора функциональных блоков, таких как модули ввода и вывода данных, лингвистических переменных, нечеткого вывода и других. При этом пользователь может произвольно соединять эти блоки, формируя модели сложных ИСУ.

В основе модульной модели ИСУ лежат классы *Model*, *DataArchive* и *Block*.

Класс *Model* инкапсулирует все объекты, необходимые для построения модели ИСУ, и обеспечивает взаимодействие между ними. Он отвечает за ведение списка имеющихся в модели ИСУ модулей, отслеживает корректность связей между ними, предоставляет интерфейсы для передачи данных модели для вычисления, а также получения результатов.

Класс *Block* является базовым для реализаций различных модулей ИСУ. Он предоставляет общий интерфейс для проведения расчетов, а также поддерживает базовые функции, необходимые большинству модулей.

Класс *DataArchive* обеспечивает хранение и извлечение данных, передаваемых внутри модели. В нем хранятся данные, порождаемые выходными портами всех модулей в системе, а также входные и выходные данные модели.

В оптимизаторе реализованы два генетических алгоритма – обычный генетический алгоритм, обеспечивающий минимизацию вещественной целевой функции, а также алгоритм типа *NSGA*, находящий множество парето-оптималь-

ных решений для задачи минимизации нескольких целевых функций. Эти алгоритмы могут использоваться с двумя типами хромосом, в которых кодирование решений происходит с помощью бинарной ДНК или набором вещественных чисел.

Процесс оптимизации протекает следующим образом. Сначала выбирается объект для оптимизации и создается объект, реализующий оптимизационный алгоритм. Параметры алгоритма запрашиваются у пользователя с помощью диалогового окна и соответствующего класса для обработки ввода пользователя. Хромосомы первого поколения по очереди декодируются оптимизируемым объектом, и для них вычисляются соответствующие значения функции полезности. После этого оптимизационный алгоритм создает новое поколение, используя определенные генетические операции и сортировку хромосом на базе вычисленных значений функций полезности. Процесс продолжается до завершения оптимизации согласно выбранным пользователем условиям.

Одной из основных проблем практического применения генетической оптимизации для создания ИСУ является необходимость проведения большого числа вычислений функции полезности [3]. Время оптимизации может быть сокращено за счет применения оптимизации с использованием обучающего сигнала и математических моделей ОУ. На финальном этапе решения, показавшие себя достаточно хорошо на моделях, проверяются на реальном ОУ.

Разработанный ОБЗ обеспечивает такие возможности, как

- оптимизация по заданному обучающему сигналу;
- оптимизация с использованием проверки работы ИСУ на модели, реализованной во внешней системе, или на ОУ;
- проверка соответствия начального состояния ОУ некоторым заданным начальным условиям;
- одновременная работа нескольких оптимизационных алгоритмов;
- поиск и выявление повторных запросов на вычисление функции полезности с дальнейшим использованием значения, полученного при первой проверке, либо усреднением значения функции по результатам нескольких испытаний в зависимости от предпочтений пользователя;
- приостановление оптимизации и продолжение ее в дальнейшем с помощью различных настроек алгоритмов или даже разных алгоритмов;
- многократное тестирование решений с целью повышения точности измерения функции пригодности.

Модульный подход к построению модели ИСУ диктует построение интерфейса пользовате-

ля в виде набора блоков и связей между ними. Пользователь может добавлять или удалять блоки, модифицировать связи между ними.

Для обеспечения возможности создания интерфейсов различного типа ОБЗ был выполнен в виде библиотеки, содержащей основные функции оптимизатора, а также те части интерфейса, которые могут использоваться в разных проектах. Эта библиотека затем может применяться для создания ОБЗ с различными интерфейсами.

Были разработаны два интерфейса: предусматривающий возможность манипулирования с блоками и связями между ними и предназначенный для создания ПИД-регуляторов.

Во втором случае пользовательский интерфейс исключает доступ пользователя к блочной структуре модели. Пользователю предоставляется непосредственный доступ к управлению коэффициентами контроллеров и пространством поиска для оптимизации.

Пример использования ОБЗ

Рассмотрим результаты моделирования структуры ИСУ с применением ОБЗ.

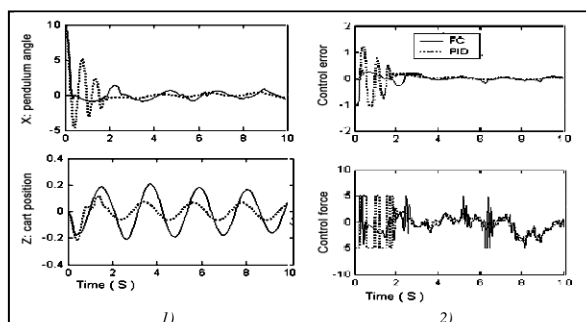
Движение динамической системы «перевернутый маятник – каретка перемещения» описывается следующими уравнениями:

$$\ddot{\theta} = \frac{g \sin \theta + \cos \theta \left(\frac{u + \xi(t) + \{+a_1 \dot{z} + a_2 z\} - m \dot{\theta}^2 \sin \theta}{m_c + m} \right) - k \dot{\theta}}{l \left(\frac{4}{3} - \frac{m \cos^2 \theta}{m_c + m} \right)}$$

$$\ddot{z} = \frac{u + \xi(t) + \{-a_1 \dot{z} - a_2 z\} + m l (\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta)}{m_c + m} \quad (1)$$

где g – ускорение свободного падения (9.8 м/сек²); m_c – масса каретки; m – масса маятника; l – половина длины маятника; $\xi(t)$ – стохастическое воздействие; u – управляющая сила, действующая на каретку.

Уравнения для скорости производства энтропии в ОУ и ПИД-регуляторе имеют следующий



Примечание: X (pendulum angle) – угол отклонения маятника; Z (cart position) – положение каретки; control error – ошибка управления; control force – управляющая сила.

Сравнение результатов моделирования управления ОУ с помощью НР (FC) и традиционного ПИД (PID)-регулятора

вид:

$$\frac{d}{dt} S_{\theta} = \frac{k \dot{\theta}^2 + \frac{m \dot{\theta}^3 \sin 2\theta}{m_c + m}}{l \left(\frac{4}{3} - \frac{m \cos^2 \theta}{m_c + m} \right)}; \quad \frac{d}{dt} S_z = a_1 \dot{z}^2; \quad \frac{d}{dt} S_u = k_a \dot{e}^2. \quad (2)$$

Физической моделью рассматриваемой динамической системы (1) является мотоцикл с учетом биомеханических характеристик водителя или ее обобщение на трехмерный случай системы «одноколесный робот – велосипед».

При наличии рэлеевского стохастического шума, действующего на каретку, и при наличии времени задержки сигнала в системе измерения положения маятника необходимо перевести маятник из начального положения в целевое вертикальное ($\theta=0$) и удерживать ОУ в заданном вертикальном положении. Зададим следующие значения параметров: $m_c=1$; $m=0.1$; $l=0.5$; $k=0.4$; $a_1=0.1$; $a_2=5$ и начального положения $[\theta_0; \dot{\theta}_0; z_0; \dot{z}_0]=[10; 0.1; 0; 0]$. Введем также ограничение на силу управления: $-5.0 < u < 5.0$ [N].

Рассмотрим ИСУ, содержащую нечеткий ПИД-регулятор, управляющий движением маятника, с использованием разработанного инструментария ОБЗ.

На рисунке показаны результаты моделирования движения системы в двух случаях управления: (1) – с помощью нечеткого регулятора, БЗ которого построена с использованием разработанного ОБЗ; (2) – с помощью классического ПИД-регулятора с коэффициентами усиления $K=[80 \ 15 \ 60]$, полученными усреднением коэффициентов усиления НР.

Результаты моделирования показывают следующее. С точки зрения оптимизации системы автоматического управления по критериям качества управления, таким как минимум ошибки управления, минимум производства энтропии в объекте управления (то есть минимум тепловых потерь, потерь полезной работы и энергии), а также с учетом ограничений на управляющую силу ИСУ, разработанная на основе ОБЗ, является более эффективной, чем традиционные ПИД-регуляторы.

Сравнение результатов движения маятника и каретки перемещения, управляющей силы, термодинамических характеристик ОУ и регуляторов (потери полезной работы), законов управления коэффициентами усиления нечеткого и традиционного ПИД-регуляторов проведено в следующих условиях: гауссовский шум, воздействующий на ОУ; время задержки в системе измерения, равное 0.002 сек. Результаты моделирования показывают, что построенная БЗ НР, управляющего движением перевернутого маятника, является робастной; с точки зрения критериев качества управления, таких как минимум ошибки управления, минимум производства энтропии в объекте управления и

системе управления, а также с учетом минимума управляющей силы разработанная ИСУ эффективнее традиционных ПИД-регуляторов.

Рассмотренная в статье архитектура ОБЗ позволила создать инструментарий для проектирования ИСУ сложной конфигурации. На основе предложенного инструментария ОБЗ могут быть рассмотрены актуальные задачи формирования БЗ для проектирования робастных НР, например, задача координационного управления коэффициентами усиления двух ПИД-регуляторов, представляющая самостоятельный интерес для теории и систем управления. Использование инструмента-

рия ОБЗ позволяет одновременно реализовать процесс проектирования робастных БЗ на основе алгоритмов обучения и адаптации.

Литература

1. Litvintseva L.V., Takahashi K., Ulyanov S.S. et al. Intelligent robust control design based on new types of computations. Note del Polo Ricerca, Universita degli Studi di Milano Publ., 2004. Vol. 60.
2. Сорокин С.В., Литвинцева Л.В., Ульянов С.В. Оптимизатор баз знаний на мягких вычислениях // Нечеткие системы и мягкие вычисления. 2008. № 1.
3. Ulyanov S.V. System and method for stochastic simulation of nonlinear dynamic systems with a high degree of freedom for soft computing applications. US patent № 2004/0039555 A1. 2004.

РАЗМЕЩЕНИЕ УЗЛОВ И БЛОКОВ РАДИОЭЛЕКТРОННОЙ И ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ НА ОСНОВЕ БИОНИЧЕСКИХ МЕТОДОВ

(Работа выполнена при частичной поддержке РФФИ, грант № 09-01-00492 (РНП.2.1.2.1652))

С.А. Бушин (Компания «Астор-Трейд», г. Москва, sergey@incotex.ru);

В.В. Курейчик, д.т.н.

(Таганрогский технологический институт Южного федерального университета, vkur@tsure.ru)

В статье предложены модели и бионический алгоритм, позволяющие выполнять свертывание и декомпозицию схем большой размерности. Это дает возможность осуществлять синтез различных вариантов размещения путем построения иерархической многоуровневой декомпозиционной структуры модели. При этом временная сложность алгоритма не выходит из области полиномиальной сложности. Полученные результаты позволяют говорить об эффективности метода и целесообразности его использования в решениях других задач оптимизации.

Ключевые слова: задача размещения, генетический алгоритм, математическая модель, гиперграф.

RADIO-ELECTRONIC AND ELECTRONIC COMPUTING EQUIPMENT COMPONENTS AND BLOCKS PLACEMENT BASED ON BIONIC METHODS

Bushin S.A. («Astor-Trade» Company, Moscow, sergey@incotex.ru); **Kureychik V.V., Ph.D.** (Taganrog Institute of Technology Southern Federal University, vkur@tsure.ru)

Abstract. Models and bionic algorithm allowing making the circuits of big dimensions folding and decomposition are proposed. It allows making synthesis of different placement variants by means of hierarchical multilevel decomposition model structure construction. The algorithm running time does not leave the polynomial complexity area. The results obtained allow speaking about the method effectiveness and appropriateness of its use in the other optimization problems solution.

Keywords: placement problem, genetic algorithm, mathematical model, hypergraph.

Стремительный прогресс в технологии создания узлов и блоков радиоэлектронной и электронно-вычислительной аппаратуры (РЭА и ЭВА) обуславливает потребность в новых средствах автоматизированного проектирования. Качественный рост сложности объекта проектирования привел к качественным изменениям в методологии проектирования, к повышению роли математического обеспечения САПР. Это позволяет в области синтеза топологии выйти на следующий уровень программного обеспечения САПР. Исходными для проектирования являются спецификация прибора и технические требования к нему. Спецификация определяет логическую цель проектирования, технические требования – физические ограничения. Синтез топологий является од-

ним из важнейших этапов в общей проблеме проектирования РЭА и ЭВА. В этой связи разработка алгоритмов проектирования топологии актуальна для новых поколений РЭА и ЭВА.

В работе рассмотрена одна из важнейших задач конструкторского проектирования РЭА и ЭВА – задача размещения. Она относится к классу NP (неопределенно распознаваемых за полиномиальное время) или является NP -полной, и для нее не известен алгоритм, растущий в полиномиальной степени. В этой связи разработка эффективных полиномиальных алгоритмов – актуальная и важная задача. В настоящее время появились генетические алгоритмы, ориентированные на решение задач размещения различного уровня иерархии [1–5]. Эти алгоритмы позволяют получать наборы