

О МЕТОДАХ ОПТИМИЗАЦИИ ПРОЦЕССОВ ПРИНЯТИЯ РЕШЕНИЙ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ УПРАВЛЕНИЯ ПРИ АЛГОРИТМИЧЕСКИХ ОГРАНИЧЕНИЯХ

В.А. ФИЛАТОВ, В. Е. КУЧЕРЕНКО

Рассмотрены вопросы принятия решений при нечетких алгоритмических ограничениях на основе дальнейшего развития методов линейного программирования. Теоретические результаты подтверждены практическими приложениями

The problems of decision making under conditions fuzzy algorithmic restrictions were considered on basis of further development of linear programming methods. Theoretical results were confirmed by practical applications.

ВВЕДЕНИЕ

Важным направлением реализации функций принятия решений в автоматизированных системах управления сложными технологическими объектами является внедрение методов оптимизации [1, 2]. Выбор метода решения — один из важнейших этапов оптимизации. Можно выделить аналитические методы и методы математического программирования.

Группа аналитических методов оптимизации обычно объединяет аналитический поиск экстремума функции, метод множителей Лагранжа, вариационные методы и методы на основе принципа максимума [2]. Аналитический поиск экстремума функций, заданных без ограничений на независимые переменные, применяется к задачам, у которых оптимизируемая функция имеет аналитическое выражение, дифференцируемое во всем диапазоне исследования, а число переменных ограничено. Это один из наиболее исследованных, широко применяемых на практике методов.

Методы математического программирования традиционно включают динамическое программирование, линейное программирование и нелинейное программирование.

Динамическое программирование — эффективный метод решения задач оптимизации многостадийных процессов [2, 3]. Эти процессы можно определить как деятельность, при которой принимаются последовательные решения, направленные на достижение единой цели. Типичные области применения моделей динамического программирования в принятии решений: разработка правил управления запасами, которые устанавливают сроки пополнения запасов и размер заказа; разработка принципов календарного планирования производства и уравнивания занятости в условиях нестабильного спроса на продукцию; распределение капитальных вложений между возможными новыми направлениями их использования; разработка стратегий инвестирования в развитие экономики.

Модели динамического программирования ценны тем, что позволяют принимать огромное количество решений, основываясь на стандартном подхо-

де при минимальном участии человека. Типы задач динамического программирования: задачи последовательного принятия решений; динамические задачи с бесконечным количеством этапов.

В задачах последовательного принятия решений обычно применяют две вычислительные системы: решение задач в прямом направлении, если первый шаг схемы динамического программирования соответствует первому по времени принятию решения; решение задачи в обратном порядке, если первый шаг схемы соответствует последнему по времени принятию решения;

Линейное программирование — метод для решения задач оптимизации с линейными функциями для критерия оптимальности и линейными ограничениями на область изменения переменных [2, 3]. Линейное программирование применяется при решении таких задач, как: оптимальное планирование производства при ограниченном количестве ресурсов; транспортные задачи; задачи о раскрое; задачи о смесях; задачи о максимальном потоке; матричные антагонистические игры.

Подобные задачи решаются итерационными способами и не допускают нелинейных критериев и ограничений.

Методы нелинейного программирования — объединяют в основном следующие способы решения задач оптимизации: градиентные, безградиентные и случайного поиска [2]. Общим для методов нелинейного программирования является то, что их используют при решении задач с нелинейными критериями оптимальности. Все методы нелинейного программирования — это численные методы поискового типа. Суть их в определении набора независимых переменных, дающих наибольшее приращение оптимизируемой функции. Данная группа методов применяется как для детерминированных, так и стохастических процессов. Следует отметить, что методы обладают значительной сложностью.

Указанные подходы относятся к классическим методам оптимизации [2] и предполагают четкое пространство состояний объекта исследования.

Нечеткое математическое программирование [4, 5, 6] выделяет естественную множественность не-

точно (нечетко) определенных критериев, их значений и ограничений. Нечеткий подход облегчает задачу лица, принимающего решения, позволяя ему нечетко, приближенно формулировать критерии, цели и ограничения. Синтез теории нечетких множеств и классических методов оптимизации является основой создания новых методов и эффективных решений, что и определяет актуальность исследований.

Целью данной работы является повышение эффективности и достоверности процессов принятия решений в автоматизированных системах управления за счет расширения существующих классических подходов на случаи нечетко и (или) неточно заданных критериев, целей и ограничений.

1. ПОСТАНОВКА ЗАДАЧИ

Цель, которой придерживаются при выборе и принятии оптимального решения, заключается в том, чтобы выявить наилучший (оптимальный) вариант действия при наличии некоторых ограничений. В работе [7] определено, что методы принятия оптимальных решений могут быть эффективными в имитационно-управляющих моделях, построенных на основе совместного комплексного использования сетей с управляемой структурой, реляционных баз данных и баз знаний.

Пусть задана функция цели оптимизационной задачи линейного программирования

$$F = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad j = \overline{1, n}, \quad (1)$$

где c_j — некоторые коэффициенты, x_j — некоторые переменные.

Для переменных функции (1) существуют ограничения двух видов:

линейные ограничения:

$$AX \leq B; \\ A = \|a_{ij}\|_{mn}, \quad X = \|x_1, x_2, \dots, x_n\|, \quad B = \|b_1, b_2, \dots, b_m\|; \quad (2)$$

алгоритмические ограничения на основе нечетких правил типа

$$\text{if/ then,} \quad (3)$$

которые в реализациях могут быть представлены как

$$\begin{aligned} &\text{if (лингвистическая переменная 1)} \\ &\text{is (нечеткое значение 1)} \\ &\text{then (лингвистическая переменная 2)} \\ &\text{is (нечеткое значение 2),} \end{aligned} \quad (4)$$

причем «нечеткое значение» определено на множестве некоторых функций принадлежности [8].

Необходимо разработать метод, который позволит эффективно решать задачи математического программирования при нечетких алгоритмических ограничениях, на основании которого будет реализо-

ван программный комплекс. Для этой цели, необходимо, используя в качестве базового методы линейного программирования [2, 3] и положения нечеткой логики, определить пути и подходы к вычислению величин $x_j \geq 0, j = \overline{1, n}$, которые оптимизируют функцию (1) с учетом ограничений (2–3).

2. АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ К РЕШЕНИЮ ОПТИМИЗАЦИОННЫХ ЗАДАЧ, СФОРМУЛИРОВАННЫХ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

Существующие решения задачи нечеткого математического программирования могут быть сформулированы как решение задачи максимизации (или минимизации) заданной функции на заданном множестве допустимых альтернатив, которое описывается системой равенств или неравенств [4] вида (1), (2).

Причем описание функций, параметров, от которых зависят эти функции, и самого множества X может быть представлено нечеткими переменными в нечетком пространстве состояний. Таким образом, классическая задача математического программирования превращается в задачу нечеткого математического программирования.

Различают несколько постановок задачи нечеткого математического программирования. Некоторые из них рассмотрены в [4, 5]:

максимизация функции

$$F: X \rightarrow R^1, \quad (5)$$

где R^1 — числовая ось, на заданном нечетком множестве допустимых альтернатив, определяемых функцией принадлежности

$$\mu(X) \rightarrow [0, 1]; \quad (6)$$

нечеткий вариант задачи математического программирования основан на модификации классической задачи математического программирования, которая достаточно подробно обоснована, например в [1, 2].

Возникают важные для практики случаи:

а) если «смягчить» ограничения, т. е. допустить возможность их представления в нечетком пространстве состояний, то получим задачу нечеткого математического программирования;

б) вместо максимизации функции (1) можно стремиться к достижению некоторого приемлемого, наперед заданного значения функции (1) на основе использования функций (6).

В настоящее время известны решения (5) для случаев (6) [4, 5]. Анализ показал, что указанные решения не охватывают важный случай нечетких алгоритмических ограничений типа (3) в смысле (4). Это существенно ограничивает возможности принятия решений для случая нечетких алгоритмических ограничений. В особенности это характерно для практических случаев, когда сталкиваются с примени-

ем точной теории оптимизации к неточным моделям, критериям и ограничениям, где нет обоснованных требований точного определения некоторых величин (производительность, прибыль, временные затраты и т. п.) и где слишком часто проявляются трудности вычислительного характера.

В связи с изложенным выше, принимаем решение о целесообразности разработки метода принятия решений при нечетких алгоритмических ограничениях в задачах математического программирования.

3. РАЗРАБОТКА МЕТОДА И АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧ С НЕЧЕТКИМИ ОГРАНИЧЕНИЯМИ

3.1. Разработка метода решения задач с нечеткими ограничениями

Рассмотрим случай, когда, согласно постановки задачи, существуют два типа ограничений: детерминированные линейные и нечеткие алгоритмические. Для решения подобных задач может быть предложена схема, представленная на рис. 1.

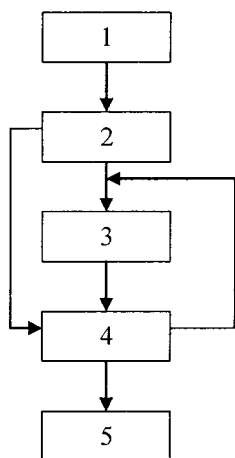


Рис. 1. Схема решения задач с нечеткими алгоритмическими ограничениями (1 – формирование исходных данных; 2 – формализация модели; 3 – реализация метода математического программирования; 4 – логический вывод на основе нечетких правил базы знаний (БЗ); 5 – получение и анализ искомого результатов)

В модуле 1 (рис. 1), на основе поставленных целей и знаний о предметной области, осуществляется формирование исходных данных.

В модуле 2 (рис. 1), согласно (1–4), реализуются следующие действия: составляется и формализуется математическая модель задачи; формулируется целевая функция; выделяются ограничения. Ограничения, представленные нечеткими высказываниями, вводятся в БЗ.

В модуле 3 (рис. 1) для поставленной задачи определяется и обосновывается метод математического линейного программирования. Далее применяется классический метод решения задачи линейного

программирования (1) – (3) без учета алгоритмических нечетких ограничений.

В модуле 4 (рис. 1), на основе правил (4), реализуется нечеткий логический вывод с учетом нечетких алгоритмических ограничений БЗ и полученных в модуле 3 результатов. Процедуры логического вывода реализуются после этапа настройки БЗ.

Так на первом этапе из постановки задачи выделяются лингвистические переменные. На основании знаний экспертов о предметной области для каждого нечеткого значения назначается соответствующая функция принадлежности $\mu(x)$ и осуществляется их настройка:

«малое» значение

$$\mu(x, a_1, c_1) = \frac{1}{1 + e^{-a_1(x-c_1)}}; \quad (7)$$

«среднее» значение

$$\mu(x, \sigma_2, c_2) = e^{-\frac{(x-c_2)^2}{2\sigma_2^2}}; \quad (8)$$

«большое» значение

$$\mu(x, a_3, c_3) = \frac{1}{1 + e^{-a_3(x-c_3)}}, \quad (9)$$

где $c_1, c_2, c_3, a_1, a_3, \sigma_2$ – параметры настройки функций $\mu(x)$.

Далее составляются правила типа (3), (4), которые уточняются совместно с экспертами. Нечеткие правила являются исходными данными для процедур логического вывода. Определяются и формируются также правила, выполнение которых осуществляет выход и формирование искомого результата и отчетных данных.

Прямой нечеткий логический вывод может быть представлен как [8]:

$$\begin{array}{l} \text{if } x \text{ is } \mu_1(x) \text{ then } y \text{ is } \mu_2(y) \\ x \text{ is } \mu_1'(x) \\ \text{-----} \\ y \text{ is } \mu_2'(y) \end{array} \quad (10)$$

где $\mu_1(x), \mu_2(y)$ – соответствующие функции принадлежности, $\mu_1'(x)$ – известное значение функции принадлежности, $\mu_2'(y)$ – искомое значение функции принадлежности.

If/then правила (3), (4) при известном значении функции принадлежности $\mu_1'(x)$ антецедента позволяют нам определить искомое значение функции принадлежности $\mu_2'(y)$ следствия.

Вычисление значений функций принадлежности для (10) осуществляем как нахождение

$$\mu_2'(y) = \bigvee_x [\mu_1'(x) \wedge \mu(x, y)], \quad (11)$$

где $\mu(x, y)$ – функция принадлежности известного из (10) отношения $\tilde{R}(x, y)$.

На этапе нечеткого логического вывода путем его реализации согласно (10), (11) и дефаззификации [8] осуществляется анализ результатов (11), полученных с помощью предыдущего модуля. Если полученные после логического вывода значения не удовлетворяют условиям правил выхода вида

$$\text{if } \min\{\mu_i(y)\} \geq \mu^*(y) \text{ then } \{\text{результаты удовлетворительны, вывод данных}\}, \quad (12)$$

то выполняется обратный логический вывод, необходимый для того, чтобы определить, какие еще дискретные ограничения необходимо добавить к начальной математической модели для обеспечения искомого результата.

Обратный нечеткий логический вывод может быть представлен как [8]:

$$\begin{array}{l} \text{if } x \text{ is } \mu_1(x) \text{ then } y \text{ is } \mu_2(y) \\ y \text{ is } \mu_2'(y) \\ \text{-----} \\ x \text{ is } \mu_1'(x) \end{array} \quad (13)$$

If/then правила (3), (4) при известном значении функции принадлежности $\mu_2'(y)$ консеквента позволяют нам определить искомое значение функции принадлежности $\mu_1'(x)$ антецедента.

Вычисление значений функций принадлежности для (10) осуществляем как нахождение

$$\mu_1'(x) = \bigvee_y [\mu(x, y) \wedge \mu_2'(y)]. \quad (14)$$

Модуль 5 (рис. 1) реализует преобразование искомого результата в вид, удобный для восприятия лицом, принимающим решения; формирует рекомендации и отчетные данные.

3.2. Алгоритмическая реализация предложенного метода

Алгоритм решения задачи оптимизации процессов принятия решений с нечеткими алгоритмическими ограничениями представим в виде выполнения таких действий.

Шаг 1. На основе поставленных целей и знаний о предметной области осуществляем формирование исходных данных.

Шаг 2. Запишем функцию цели задачи линейного программирования. Необходимо определить величины $x_j \geq 0, j = \overline{1, n}$, которые максимизируют (минимизируют) функцию (1).

Шаг 3. Выделяются ограничения двух видов: линейных вида (2); нечетких алгоритмических правил (4).

Шаг 4. Формируем БЗ на основе нечетких правил.

Шаг 5. С учетом сложности решения задачи определяется и обосновывается классический метод математического линейного программирования.

Шаг 6. Решение задачи линейного программирования (1–3) классическим методом математического линейного программирования без учета алгоритмических нечетких ограничений.

Шаг 7. Из постановки задачи выделяются лингвистические переменные и осуществляется настройка БЗ согласно (7–9).

Шаг 8. На основе полученных, согласно шагу 6, результатов и правил (4) реализуется нечеткий логический вывод с учетом нечетких алгоритмических ограничений БЗ (10–11).

Шаг 9. Дефаззификация нечетких решений согласно (10–11) на основе метода «центра масс» [8].

Шаг 10. Осуществляем анализ результатов (11) на их соответствие условиям правил выхода (12).

Шаг 11. Если правила (12) выполнены и условие консеквента справедливо (true), то осуществляем преобразование искомого результата в вид, удобный для восприятия лицом, принимающим решения, формируем рекомендации и отчетные данные. Останов.

Шаг 12. Если правила (12) выполнены и условие консеквента не справедливо (false), то выполняется, согласно (13–14), обратный логический вывод, определяются, какие еще ограничения из (2) необходимо добавить к начальной математической модели для обеспечения искомого результата.

Шаг 13. Переход к шагу 6.

Шаг 14. Останов.

Важно также оценить сложность решения поставленной задачи при нечетких алгоритмических ограничениях. Как показал анализ, временные затраты на выполнение предлагаемой версии алгоритма в основном зависят от размерности БЗ, числа переменных n , и количества правил продукций l и в реализации не превышают

$$Z(n, l) \leq \alpha(\beta n + \lambda l)^2, \quad (15)$$

где α – коэффициент, учитывающий производительность вычислительной системы; β, λ – некоторые коэффициенты, определяемые конкретными реализациями.

Значение верхней оценки (15) работы алгоритма подтверждено на тестовых примерах при оптимизации загрузки групп оборудования, что допускает эффективную программную реализацию.

4. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМЫХ РЕШЕНИЙ

Разработанные методы и алгоритмы реализованы в виде версии программного продукта (ПП), ориентированного на решение практических задач в автоматизированных системах принятия решений. При разработке ПП использовался объектно-ориентированный подход на языке программирования высокого уровня C++ в визуальной среде разработки Borland C++ Builder 6 [9].

Программное обеспечение выполнено в виде отдельных модулей: модуль для решения классической симплекс задачи; модуль выполнения операций с нечеткими переменными и правилами на основе операторов Мамдани [8, 10]. Каждый из программных модулей имеет самостоятельное значение. Их объединение позволяет получить комплекс программ, с помощью которых возможна автоматизация решения оптимизационных задач с нечеткими алгоритмическими ограничениями.

Каждый модуль составляет собой библиотеку классов, ориентированных на определенную область и графическую оболочку, обеспечивающую работу с библиотекой. Рассмотрим эти модули.

Модуль для решения классической симплекс задачи. Данный модуль предназначен для решения классической симплекс задачи, работает с канонической формой задачи. Имеет функции ручного ввода данных, их сохранения в файл и последующего чтения.

Для функционирования модуля были реализованы следующие классы: TSimplexTable и TSimplexAlgoritm. TSimplexTable позволяет динамически создавать симплекс-таблицу требуемой размерности. Также класс работает с vcl компонентом TStringGrid (функция чтения данных с таблицы и отображения их в нее). TSimplexAlgoritm – класс, реализующий работу симплекс-алгоритма и работающий с TSimplexTable таблицами. Основные функции: Init() – инициализация алгоритма (получение начальных данных из StringGrid, создание необходимых симплекс-таблиц и т.д.); CalculateTable() – перерасчет таблицы и анализ на возможность продолжения работы алгоритма; DoStep() – формирование новой таблицы, исходя из подсчитанных значений разрешающего столбца и строки.

Модуль выполнения операций с нечеткими переменными и правилами. В данном модуле реализуются такие функции: добавление лингвистических переменных; назначение лингвистическим переменным нечетких значений; настройка функций принадлежности для нечетких значений; составление «if/then» правил; нечеткий логический вывод.

Для функционирования модуля были реализованы следующие классы: TMInterval, TMFunction, TFuzzyVar, TLinguisticVar, TFRule, TFIS. TMInterval – интервал, на котором значение функции принадлежности описывается с помощью простого правила (аналитического выражения). В нем определены поддерживаемые стандартные функции принадлежности: trimf, trapmf, gbellmf и параметры для настроек этих функций. TMFunction – функция принадлежности, показывает степень принадлежности на интервале от 0 до 1 для нечеткой переменной. TFuzzyVar – нечеткая переменная, характеризуется тройкой $(N, X, R(N, x))$, где N – название переменной; X – универсальное множество с базовой переменной x ; $R(N, x)$ – нечеткое подмножество множества X , представляющее собой нечеткое ограничение на зна-

чения переменной x , обусловленное N . TLinguisticVar – лингвистическая переменная. Более точно структура лингвистической переменной описывается набором (N, T, X, G, M) , в котором N – название этой переменной; T – терм-множество N , т. е. совокупность ее лингвистических значений; X – универсальное множество с базовой переменной x ; G – синтаксическое правило, которое может быть задано в форме бесконтекстной грамматики, порождающей термы множества T ; M – семантическое правило, которое каждому лингвистическому значению t ставит в соответствие его смысл $M(t)$, причем $M(t)$ обозначает нечеткое подмножество множества X . TFRule – класс, разработанный для представления правила, его проверки. В классе реализованы методы просчета нечеткого отношения $R(x, y)$, необходимого для осуществления логического вывода на основе правил, реализованы функции визуализации правила в виде графиков. TFIS – fuzzy interface system. Реализует основные функции модуля. Представляет собой базу знаний с функциями добавления знаний (правил), предоставляет пользователю возможность выполнения логического вывода на основании нечетких правил.

Предлагаемая разработка обладает относительно малой сложностью и высокой достоверностью принимаемых решений. Отдельные модули: модуль для решения классической симплекс задачи; модуль выполнения операций с нечеткими переменными и правилами на основе операторов Мамдани – обладают достаточно высокой универсальностью и хорошо адаптируются к особенностям объекта исследования. Применение ПП в задачах оптимизации загрузки технических средств сложных технологических объектов при принятии ответственных решений в распределенных автоматизированных системах подтвердило возможности его эффективного использования в реальном масштабе времени, сложность реализации не превышала расчетной согласно (19).

ВЫВОДЫ

1. Сформулирована и решена задача оптимизации процессов принятия решений при нечетких алгоритмических ограничениях для одного класса задач линейного программирования, являющаяся дальнейшим развитием существующих решений и имеющая важные преимущества над существующими решениями. Это позволило расширить возможности существующих подходов, повысить достоверность принимаемых решений.

2. Получили дальнейшее развитие методы принятия решений за счет возможности учета нечетких алгоритмических ограничений в условиях неточных, нечетких или плохо формализуемых ограничений.

3. Для предлагаемых методов впервые разработаны алгоритмические и программные средства эффективного их применения при решении практических

задач. Определена верхняя оценка вычислительной сложности, которая в практических приложениях не превышает квадратичной.

4. Практическая значимость полученных результатов заключается в разработанных и реализованных алгоритмических и программных средствах, которые ориентированы на современные информационные технологии. Алгоритмические и программные средства внедрены при разработке автоматизированных систем принятия решений на сложных технологических объектах.

5. Перспективными направлениями исследований следует считать дальнейшее развитие возможностей по адаптации методов решений прикладных задач в условиях высокоавтоматизированных технологических объектов производственного назначения.

Литература: 1. *Болтянский В.Г.* Оптимальное управление дискретными системами. — М.: Наука. Главн. ред. физ.-мат. литературы, 1973. — 448 с. 2. *Линейное и нелинейное программирование* / Под ред. И.Н. Ляшенко. — К.: Вища шк., 1979. — 372 с. 3. *Бондаренко М.Ф., Гвоздинский А.Н.* Оптимизационные задачи в системах принятия решений. — Харьков, 1998. — 215 с. 4. *Алтушин А.Е., Семухин М.В.* Модели и алгоритмы принятия решений в нечетких условиях. — Тюмень: ТГУ, 2000. — 352 с. 5. *Орловский С.А.* Проблемы принятия решений при нечеткой исходной информации. — М: Наука, 1981. — 203 с. 6. *Бобко В.Д., Золотухин Ю.Н., Нестеров А.А.* Основа нечеткого логического

контроллера. http://www.idisys.iae.nsk.su/ref_98/folder4/doc12.html. 7. *Кучеренко В.Е.* Модели анализу процесів обробки даних в автоматизованих системах // Радиоелектроніка і інформатика. — 2004. — № 3. — С. 103–107. 8. *Tsoukalas L.H., Uhrig R.E.* Fuzzy and Neural Approaches in Engineering. New York: John Wiley & Sons, Inc, 1997. 587 p. 9. *Архангельский А.* Программирование в среде C++ Builder. — 2003. — 634 с. 10. *Mamdani E.H.* Applications of fuzzy set theory to control system: A survey fuzzy automata and decision processes / M.M. Gupta, G.H. Saridis and B.R. Gaines, eds. New York: North - Holland, 1977. P. 1–13.

Поступила в редколлегию 21.04.2005



Филатов Валентин Александрович, кандидат технических наук, доцент, доцент кафедры искусственного интеллекта Харьковского национального университета радиотехники. Научные интересы: распределенные интеллектуальные системы.

Кучеренко Валерий Евгеньевич, аспирант кафедры искусственного интеллекта Харьковского национального университета радиотехники. Научные интересы: модели и системы принятия решений.