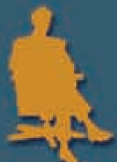
АДАПТИВНЫЕ И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Б. Лю

теория и практика неопределенного программирования



Стохастическое программирование

Нечеткое программирование

Неточное программирование

Нечетко-случайное программирование

Случайно-нечеткое программирование

теория и практика неопределенного программирования

Оглавление

О серии «Адаптивные и интеллектуальные системы»

1. Вот уже более 60 лет, с момента выхода первопроходческой работы У. МакКаллока и У. Питтса [1], не прекращаются попытки математического и компьютерного моделирования естественных (природных) интеллектуальных систем, а также работы по созданию искусственных интеллектуальных систем (ИИС). За прошедшие годы выполнено огромное число проектов, проведены сотни конференций, изданы тысячи книг и многие десятки тысяч статей на эту тему. Казалось бы — все прекрасно, успехи налицо. Однако при более близком рассмотрении результаты всех предпринимавшихся попыток выглядят достаточно скромными, если сопоставлять их с основной декларированной целью — создать ИИС, по уровню интеллектуальных возможностей не уступающую человеку [2].

В самом деле, давайте попытаемся припомнить, много ли известно нам примеров эффективно действующих «умных» машин и устройств: машин-секретарей, слушающих своего хозяина-человека, понимающих о чем он говорит и записывающих его речь в виде текста, разбирающих исчерканные и переправленные им черновики, помогающих подобрать данные по требуемой теме с учетом смысловых нюансов и текущих потребностей хозяина-человека; машин-переводчиков с одного человеческого языка на другой, способных действовать в любой языковой обстановке, в том числе и в сфере обыденного языка с его многочисленными неправильностями, неточностями, недоговоренностями; машин-исследователей, способных не растеряться, оказавшись в обстановке, которая «и не снилась» их создателям — такой машиной может быть, например, космический аппарат, попавший на другую планету, или же робот-спасатель. Этот список может быть продолжен почти неограниченно…

Вообще говоря, налицо тупик. В чем же здесь дело? Почему многолетние, достаточно широкомасштабные исследования и разработки в области ИИС так и не привели пока к решающему успеху? [3] Возможен ли вообще успех или же

---

[1] *McCalloch W. S., Pitts W.* A logical calculus of the ideas immanent in nervous activity. // *Bull. Math. Biophys.*, — 1943. — v. 5. — pp. 115–133. Рус. перевод: *Мак-Каллок У. С., Питтс В.* Логическое исчисление идей, относящихся к нервной активности // В сб.: «Автоматы» под ред. *К. Э. Шеннона* и *Дж. Маккарти.* — М.: Изд-во иностр. лит., 1956. — с. 362–401.

[2] Такого рода системы будем далее для краткости именовать ИИС высшего уровня.

[3] Еще более многолетними, но не намного более успешными являются попытки понять природу естественных интеллектуальных систем, предпринимаемые, в основном, в рамках нейробиологии и психологии.

проблема ИИС выходит за пределы познавательных возможностей человека и человечества? Ведь до сих пор нет ясности даже в таком кардинальном вопросе, как принципиальная реализуемость ИИС высшего уровня. Диапазон точек зрения здесь варьируется от категорического «нет» [1] до решительного «да» [2]. И если успех все же возможен, то каков должен быть путь, который привел бы к решению проблемы создания ИИС, сопоставимой с человеком по своим интеллектуальным возможностям? При этом очевидно, что практически интересные системы не обязательно должны обладать всем набором функций ИИС высшего уровня, список реализуемых ими функций может быть и существенно более скромным. Какими могут быть пути создания подобных систем?

2. Что же представляют собой ИИС? Удовлетворяющее всех определение искусственной интеллектуальной системы, а также того, что представляют собой ИИС, по интеллектуальному уровню сопоставимые с человеком, — вряд ли существует, по крайней мере сейчас, хотя на интуитивном уровне более или менее ясно, о чем идет речь. Полноценное определение изучаемого понятия — это обычно итог [3], один из результатов успешно проведенного научного исследования, но не его стартовая точка. До такого состояния ИИС еще весьма далеко, вопросов пока еще намного больше, чем ответов на них. Взамен недостижимого пока «полноценного» определения, можно построить некоторый его рабочий вариант, пригодный для использования в практических целях. Среди возможных способов формирования такого рабочего определения — предъявление серии примеров того, что заведомо можно отнести к требуемому классу объектов и, возможно, контрпримеров, т. е. перечня объектов, которые заведомо нельзя отнести к требуемому классу.

Начальное представление о требуемом уровне ИИС непосредственно вытекает из примеров, упоминавшихся выше. Чтобы яснее представить, о чем идет речь, рассмотрим в качестве характерного примера один из важнейших классов ИИС — интеллектуальные автономные системы (ИАС).

ИАС — это системы, обладающие высоким «уровнем самостоятельности», в том числе умеющие:

- достигать поставленных *целей* в высокодинамичной среде со значительным числом разнородных неопределенностей в ней;
- корректировать поставленные цели, а также формировать *новые цели* и комплексы целей, исходя из заложенных в ИАС ценностных и нормативных установок (мотивации);
- добывать *новые знания*, накапливать *опыт* решения разнообразных задач, обучаться на этом опыте, модифицировать свое поведение (реакции

[1] *Дрейфус У.* Чего не могут вычислительные машины: Критика искусственного разума. — М.: Прогресс, 1979.

[2] *McCarthy J.* What is Artificial Intelligence?
`http://www-formal.stanford.edu/jmc/whatisai.html`

[3] *Wang P.* On the working definition of intelligence // Indiana University, Center for Research on Concepts and Cognition (CRCC); CRCC Technical Report No. 94, 1994. — 32 pp.
`http://citeseer.nj.nec.com/wang95working.html`

на изменение ситуации) на основе полученных знаний и накопленного опыта;

- адаптироваться к виду задач, в решении которых возникает необходимость, в том числе обучаться решению задач, *не предусмотренных* первоначальным проектом системы;
- образовывать *«коллективы»* из ИАС (*сообщества* ИАС), нацеленные на взаимодействие их членов при решении некоторой общей задачи; эти коллективы должны располагать возможностями самоструктуризации (разнородность элементов коллективов ИАС, разнородность и динамичность связей между ИАС), исходя из текущей и/или прогнозируемой ситуации;
- осуществлять *самовоспроизведение* с привлечением местных сырьевых и энергетических ресурсов, возможно, с изменениями в «геноме» системы (для поддержки процессов эволюции в сообществах ИАС).

В первую очередь ИАС — это «самодостаточные» системы, на которые может возлагаться решение определенного комплекса прикладных задач в полном объеме. Имеется, однако, еще одна важная «ниша» для ИАС — это *интеллектуальные автономные подсистемы* в сложных системах, включающих человека-оператора (пилотируемые самолеты, вертолеты, космические аппараты, надводные, подводные или наземные аппараты и т. п.). Такие подсистемы нацеливаются на то, чтобы в максимальной степени самостоятельно решать поставленные перед ними задачи. Их использование дает возможность существенно повысить качество реализации критически важных функций, уменьшить рабочую нагрузку человека-оператора и повысить за счет этого безопасность и эффективность эксплуатации соответствующих сложных систем согласно их целевому назначению.

Структурно «деятельность» ИАС в некоторой среде можно разделить на *три сферы*:

1. Восприятие текущей ситуации (ситуация = внешняя-ситуация + внутренняя-ситуация) — *сенсорные функции.*
2. Формирование реакции («ответа») на текущую или прогнозируемую ситуацию (виды возможных реакций: изменение состояния ИАС в ее фазовом пространстве, реконфигурация, реструктуризация, адаптация целей, самообучение, самоорганизация и т. п.) — *управляющие функции.*
3. Реализация (сформированной) реакции на текущую или прогнозируемую ситуацию — *эффекторные функции.*

Заданное сочетание необходимых «умений» ИАС и требуемых конкретных форм ее «деятельности» определяет состав функций, которые должна реализовывать ИАС того или иного вида (примеры таких функций: зрение в различных диапазонах электромагнитного излучения, совмещение сенсорных данных из разных источников в единую информационную картину, выявление отказов в системах ИАС, компенсация обнаруженных отказов путем реконфигурации/реструктуризации и т. п.).

3. По аналогии с традиционными автоматическими и автоматизированными системами будем различать *системы-роботы* (или просто роботы) и *роботизированные системы*. Оба этих класса систем представляют собой наиболее очевидную сферу применения ИАС.

К системам-роботам, которые целесообразно было бы реализовывать на уровне ИАС, можно отнести самолетыроботы, вертолеты-роботы, космические аппараты-роботы, надводные и подводные аппараты-роботы, безлюдные (автоматические) системы, предназначенные для решения сложных комплексов задач без вмешательства человека и т. п. Разница между аппаратами-роботами и традиционными автоматическими аппаратами состоит в существенно более высоком уровне «самостоятельности» поведения роботов по сравнению с традиционными автоматами, в способности роботов обучаться, накапливать и использовать опыт в ходе решения поставленных задач.

Примерами роботизированных систем могут служить пилотируемые летательные аппараты (самолеты, вертолеты, космические корабли), в состав систем и бортового оборудования которых входят (на правах подсистем) интеллектуальные автономные системы. Здесь обязательным (и активным) элементом системы в целом является человек. Совсем необязательно при этом, чтобы человек находился непосредственно на борту роботизированного аппарата. Он может осуществлять свои функции (контроль, управление, целеуказание и т. п.), находясь вне аппарата, в том числе и на значительном удалении от него. Примером такого рода аппаратов могут служить роботизированные дистанционно пилотируемые летательные аппараты (ДПЛА), а также другие дистанционно управляемые аппараты, когда оператор и управляемый им аппарат разнесены в пространстве.

В качестве примеров систем, роботизация которых позволила бы резко повысить их эффективность, можно назвать: системы организации воздушного движения; средства управления энергетическими системами; средства поддержки процессов контроля и управления производствами традиционных видов (машиностроительными, химическими и нефтехимическими, добывающими и т. п.); средства управления в чрезвычайных обстоятельствах — ликвидация последствий стихийных бедствий, техногенных катастроф и другие, где ИАС можно было бы использовать как средства поддержки процессов формирования и принятия решений («интеллектуальные помощники») в человеко-машинных системах, позволяющие работать в средах со значительным числом разнородных неопределенностей, в условиях больших потоков данных и жестких временны́х ограничений.

Еще одна очевидная сфера применения ИАС — роботизированная бытовая техника различного назначения, а также сообщества роботизированных бытовых устройств («роботизированное жилище»).

4. Надо признать, что к созданию полномасштабных ИИС высших уровней мы пока еще не готовы. Как и куда надо двигаться, чтобы решение этой задачи стало возможным? При этом следует понимать, что создание систем такого уровня — дело не сегодняшнего и даже не завтрашнего дня. Нет ответов еще на многие принципиальные вопросы, без которых такие системы построить невозможно.

Но ведь кроме гипотетических полномасштабных ИИС существует еще и целый ряд проблем, важных и интересных с практической точки зрения, которые уже сейчас поддаются решению средствами, накопленными в арсенале ИИС. Что это за задачи, как их решать? Какие методы и средства из арсенала ИИС можно использовать для этих целей?

5. Итак, есть чувство неудовлетворенности тем, как соотносятся полученные результаты с затраченными на них ресурсами. И неизбежно возникает в связи с этим целая серия вопросов. Что же все-таки удалось сделать за прошедшие 60 лет, а что из задуманного не получилось и почему? Разрешима ли вообще задача создания ИИС? Какими видятся сейчас проблемы, связанные с созданием ИИС? Каковы пути решения этих проблем? На какой научной и технической базе следует их решать? Что и когда можно получить в области создания ИИС, каковы перспективы этой области знаний и есть ли они? Какие практические задачи можно решать уже сейчас, используя аппарат, полученный в исследованиях по ИИС?

Чтобы дать развернутый, подробный ответ на эти вопросы, требуется посвятить им не один десяток обстоятельных монографий, часть из которых еще только предстоит написать.

6. Серия «Адаптивные и интеллектуальные системы» как раз и нацелена на то, чтобы попытаться познакомить читателя с проблемами и результатами их решения в разнообразных областях, связанных с созданием ИИС. Предполагается издавать как переводные книги, так и книги отечественных авторов, посвященные таким вопросам как мягкие вычисления (включая искусственные нейронные сети, нечеткие системы, эволюционные вычисления) и системы, основанные на знаниях, а также работам из смежных областей (нейробиология, нелинейная динамика, теория систем и т. п.), в тех их аспектах, которые могут быть полезны для решения проблемы создания ИИС. В прикладном плане должны затрагиваться все элементы жизненного цикла ИИС, а также все виды их функций — сенсорные, управляющие и эффекторные.

И в заключение — почему серия получила именно такое название. Адаптивные системы в контексте серии понимаются в широком плане, а именно, как системы, которые могут модифицировать свое поведение применительно к меняющимся условиям их существования (среда существования системы, цели ее существования и т. д., и т. п.) То есть адаптивная система — это система, которая располагает механизмами, позволяющими ей жить и работать в условиях разнообразных и, как правило, многочисленных неопределенностей. Интеллект — хотя и далеко не единственный, но один из важнейших механизмов такого рода, что и послужило основанием для упоминания его в наименовании серии.

Ю. В. Тюменцев,
редактор серии

Предисловие редактора перевода

Книга Баодина Лю «Теория и практика неопределенного программирования» открывает серию «Адаптивные и интеллектуальные системы». Выбор этот не случаен, поскольку в данной книге идет речь о многих вещах, важных для тематики искусственных интеллектуальных систем. В частности, рассматриваются и сопоставляются элементы, принципиально важные для таких систем: различные виды неопределенности — случайность (randomness), нечеткость (fuzziness) и неточность (roughness), разнообразные их комбинации, а также методы работы с ними. Эти вопросы излагаются на примере задач математического программирования, которые находят широкое практическое применение. Для решения подобных задач привлекаются средства как традиционного характера (оптимизационные алгоритмы, имитационное моделирование, статистические алгоритмы), так и средства менее традиционного плана (искусственные нейронные сети, генетические алгоритмы, средства нечеткой логики, средства на основе понятия неточной системы).

Многие из вопросов, затрагиваемых в книге Б. Лю, не освещались или почти не освещались в литературе на русском языке. К ним относится, например, тематика неточных величин (rough variables) и неточных множеств (rough sets) [1]. Это обстоятельство вызвало ряд трудностей, связанных с используемой при переводе терминологией. В частности, термин rough переводился, как правило, как «неточный», хотя в ряде отечественных публикаций использовался термин «приближенный». Сделано это было по следующим соображениям. Во-первых, в литературе на русском языке термин «приближенный» скорее соответствует английскому approximate. Чтобы не возникало нежелательных ассоциаций, уводящих от сути дела, требовалось воспользоваться другим термином, менее «нагруженным» уже имеющимися смыслами. Предпочтение было отдано термину «неточный» в том числе и потому, что тогда соответствующий

[1] О неточных множествах совсем кратко (с. 21–22) было сказано в книге *Аверкин А. Н., Батыршин И. З., Блишун А. Ф., Силов В. Б., Тарасов В. Б.* Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. *Д. А. Поспелова.* — М.: Наука, 1986. — 312 с. — (Серия «Проблемы искусственного интеллекта»). Им также посвящена часть материала (см. разд. 14.3, с. 603–621) недавно вышедшей книги *Вагин В. Н., Головина Е. Ю., Загорянская А. А., Фомина М. В.* Достоверный и правдоподобный вывод в интеллектуальных системах. Под ред. *В. А. Вагина* и *Д. А. Поспелова.* — М.: Физматлит, 2004. — 704 с.

вид неопределенности оказывается в одном ряду с прочими «НЕ-факторами» (нечеткость, недоопределенность, неполнота и т. п.), исследуемыми и используемыми в рамках мягких вычислений. Именно поэтому rough variable — это неточная величина, rough set — неточное множество. Однако для rough space в качестве русского эквивалента принят термин «пространство приближений» как более отвечающий смыслу соответствующего понятия.

Еще одно решение было связано с терминами expected value и variance. Применительно к стохастическим задачам термин expected value следовало бы традиционно переводить как «математическое ожидание». Однако в данной книге этот термин используется в значительно более широком контексте, в частности, он употребляется применительно к нечетким (fuzzy), неточным (rough) и комбинированным (random-fuzzy, fuzzy-random, fuzzy-rough, rough-fuzzy, random-rough, rough-random, bifuzzy, birough и т. п.) моделям, где термин «математическое ожидание» не употребляется. Учитывая это, чтобы обеспечить единообразие терминологической базы в сходных условиях, всем моделям класса expected value model дано общее наименование «модели ожидаемого значения», подразумевая при этом — «ожидаемого среднего значения». Как конкретно понимается в том или ином случае термин «среднее значение» — зависит от класса моделей. В частности, в стохастических моделях (моделях стохастического программирования) это будет математическое ожидание. С термином variance было несколько проще. Для общего случая использовался вариант «рассеяние», в специальных случаях этот термин может иметь другой перевод. В частности, для стохастического случая это будет, очевидно, дисперсия.

Трудности возникли также в связи с понятием «шанс», которое обычно ассоциируется со случайностью событий. В данной книге, однако, понятие «шанс» и многочисленные производные от него трактуются значительно шире — как относящиеся к любому отклонению от жесткой предопределенности, причем совсем не обязательно это отклонение имеет вероятностный характер. Один из примеров такого использования данного термина связан с ограничениями, входящими в рассматриваемые задачи математического программирования. Эти ограничения в оригинале книги именуются chance constraints. Если речь идет о задаче стохастического программирования, то это хорошо известный случай вероятностных ограничений. Когда рассматривается задача нечеткого (fuzzy) программирования, получаем возможностные ограничения. В других случаях, когда нет общеупотребительного частного наименования для рассматриваемого вида ограничений, используется общий термин «ограничение на шансы».

Наконец, еще одно принципиальное решение связано с наименованием направления uncertainty programming. Правильным было бы именовать данную область как «математическое программирование в условиях неопределенности». Но для краткости, по аналогии с термином «стохастическое программирование» в переводе чаще всего используется термин «неопределенное программирование». Аналогично этому, термины «неопределенная среда», «неопределенная функция» и т. п. — это сокращенные варианты для более полных (и более корректных) выражений «среда, содержащая неопределенности», «функция, содержащая неопределенности» и т. п.

Остальные терминологические решения, принятые при переводе данной книги, оговариваются в соответствующих местах текста в примечаниях редактора перевода. Кроме того, книга снабжена русско-английским и англо-русским указателями терминов.

Работа по переводу книги распределилась следующим образом: предисловие и главы с 1 по 7 — Ю. Т. Каганов, главы с 8 по 24 и предметный указатель — Ю. В. Тюменцев.

Моей жене Цзиньлань

Предисловие

Принятие решений в реальной жизни обычно осуществляется в условиях неопределенности. Как мы формулируем задачи оптимизации в неопределенной среде? Как мы решаем эти задачи? Главная цель этой книги состоит именно в том, чтобы построить теорию математического программирования (МП) в условиях неопределенности (неопределенного программирования) как ответ на эти вопросы.

Под неопределенным программированием будем понимать теорию оптимизации в неопределенной среде. Основные направления неопределенного программирования в себя: стохастическое программирование, нечеткое МП, неточное МП, нечетко-случайное программирование, случайно-нечеткое программирование, случайно-неточное программирование, неточно-случайное программирование, нечетко-неточное программирование, неточно-нечеткое программирование, бислучайное программирование, бинечеткое программирование, бинеточное программирование и неопределенное программирование с множественной неопределенностью.

Эта книга дает замкнутое, достаточно полное и современное представление о теории математического программирования в условиях неопределенности. В ней рассматривается значительное число вопросов, связанных как теоретическими аспектами моделирования в условиях неопределенности, так и с соответствующими прикладными проблемами, в числе которых: транспортные задачи, моделирование систем управления запасами, задачи составления кормовых смесей, моделирование производственного процесса, проблемы водоснабжения, задача размещения и распределения оборудования, задача распределения капиталовложений, задача топологической оптимизации, задача маршрутизации движения транспорта, оптимизации резервирования, задача о критическом пути, задача составления расписания параллельно действующих машин.

К настоящему времени исследователями из различных областей разработано большое количество различных алгоритмов, которые принято именовать интеллектуальными, в частности, генетические алгоритмы, нейронные сети, алгоритмы типа имитации отжига и табу-поиска. Естественная идея состоит в том, чтобы объединить эти алгоритмы и получить более эффективные и мощные гибридные алгоритмы оптимизации. Значительное число такого рода алгоритмов, позволяющих решать разнообразные задачи, связанные с моделями неопределенного программирования, рассматривается в данной книге. Автором поддерживается веб-сайт, на котором представлены исходные файлы на языке C++, реализующие рассмотренные гибридные алгоритмы. Адрес этого сайта: `http://orsc.edu.cn/~liu/uncertain_programming`

Эта книга состоит из 7 частей. В Части I вводятся основные концепции математического программирования, генетические алгоритмы и нейронные сети. Часть II содержит различные методы формирования случайных чисел и связана с применением закона больших чисел, а также содержит статистическое моделирование, стохастические модели ожидаемого значения, стохастическое программирование с вероятностными ограничениями, стохастическое событийное программирование, гибридные алгоритмы и их применение для решения различных задач. В Части III книги вводятся понятия пространства возможностей, нечеткой величины, меры возможностей, меры необходимости, меры правдоподобия, нечеткий оператор ожидаемого значения, нечеткое моделирование и теория нечеткого программирования. Часть IV посвящена понятиям пространства приближений, неточных величин, меры доверия, неточного оператора ожидаемого значения, а также неточному программированию. Попутно обсуждается также интервальное программирование. В Части V рассматриваются нечетко-случайные величины, соответствующий им оператор ожидаемого значения, мера шансов нечетко-случайного события, нечетко-случайное имитационное моделирование и нечетко-случайное программирование. В Части VI обсуждаются случайно-нечеткие величины, и отвечающий им оператор ожидаемого значения, мера шансов случайно-нечеткого события, случайно-нечеткое имитационное моделирование и случайно-нечеткое программирование. Завершает книгу Часть VII, в которой приводится широкий спектр величин с многократной неопределенностью, а также дается набросок теории неопределенного программирования.

Предполагается, что читатель знаком с основными идеями математического программирования и обладает некоторыми навыками программирования на языке C++. Для того, чтобы облегчить чтение книги, в нее включено также изложение ряда основополагающих понятий. Книга ориентирована на исследователей, инженеров и студентов в области исследования операций, организационного управления, теории систем, информатики и различных областей техники. Читатели получают возможность изучить многочисленные новые идеи в области создания и использования оптимизационных моделей. Материал книги может послужить для них справочной базой и стимулом для дальнейших исследований в данной области.

Особую благодарность я хотел бы выразить профессору Chi-fa Ku за то, что он ввел меня в эту область исследований и за его постоянную поддержку в процессе работы. Многочисленным коллегам я обязан появлением новых идей и информации, положенных в основу этой книги. Особенно мне хотелось отметить таких коллег, как K. Iwamura, M. Gen, T. Odanaka, A.O. Esogbue, K.K. Lai, Q. Zhang, и M. Lu. Я хотел бы поблагодарить также моих студентов Q. Lü, X. Wang, R. Zhao, J. Zhong, H. Ling, G. Miao, J. Gao, J. Peng, Y. -K. Liu, J. Zhou, G. Wang, H. Ke, and Y. Jiang, которые проделали огромную работу и внесли многочисленные исправления. Я также многим обязан грантам, выделенным Национальным фондом естественных наук, Министерством образования, а также Министерством науки и технологий Китайской народной республики. Наконец я выражаю глубокую благодарность профессору Янушу Кацпшику (Janusz Kacprzyk) за предложение опубликовать эту книгу в его серии, а также редакторскому коллективу издательства «Шпрингер» за прекрасное сотрудничество и полезные замечания.

Баодин Лю
Университет Циньхуа
(Tsinghua University)
http://orsc.edu.cn/~liu

26 февраля 2002 года

Часть I
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Глава 1
Математическое программирование

Один из наиболее широко используемых подходов в области исследования операций — *математическое программирование* (МП). Его можно определить как средство минимизации (или максимизации) заданного количественного показателя (иногда нескольких таких показателей), который принято называть *целевой функцией*, при наличии набора ограничений, представленных в виде равенств и неравенств.

Среди известных в настоящее время разделов математического программирования можно упомянуть следующие: линейное программирование, нелинейное программирование, многокритериальное программирование, целевое программирование, целочисленное программирование, динамическое программирование, многоуровневое (иерархическое) программирование, стохастическое программирование, нечеткое программирование, неточное программирование, нечетко-случайное программирование, неточно-случайное программирование, нечетко-неточное программирование, неточно-нечеткое программирование, случайно-случайное (бислучайное) программирование, нечетко-нечеткое (бинечеткое) программирование, неточно-неточное (бинеточное) программирование, а также математическое программирование в условиях многократной (трехкратной, четырхкратной и т. д.) неопределенности.

В одной главе рассказать сколько-нибудь подробно обо всех этих концепциях математического программирования невозможно, поэтому здесь будут введены лишь основные понятия и средства МП, необходимые для дальнейшего изложения. Более основательно разобраться в этих концепциях читатель сможет по мере освоения материала книги [1].

1.1. Линейное программирование

Один из наиболее важных инструментов оптимизации — *линейное программирование* (ЛП) определяется так: имеется линейная функция, которую следует максимизировать с учетом набора линейных ограничений. Основная (канони-

[1] По различным аспектам математического программирования (большей частью — детерминированного) существует обширная литература. См., например, [1–72] в списке дополнительной литературы. — *Прим. ред.*

ческая) форма задачи ЛП имеет следующий вид:

$$\begin{cases} \max C^T\boldsymbol{x} \\ \text{при ограничениях:} \\ \qquad A\boldsymbol{x} = B, \\ \qquad \boldsymbol{x} \geqslant 0, \end{cases} \tag{1.1}$$

где $C = (c_1, c_2, \ldots, c_n)^T$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$, $A = (a_{ij})_{m\times n}$ и $B = (b_1, b_2, \ldots, b_m)^T$. В основной задаче линейного программирования (1.1) все независимые переменные x_i, $i = 1, 2, \ldots, n$, предполагаются неотрицательными. Это условие выполняется почти для всех реальных задач. Если это не так, например, переменная x_i может принимать как положительные так и отрицательные значения, тогда ее можно заменить выражением $x_i' - x_i''$, где x_i' и x_i'' — две новые неотрицательные переменные, т. е. $x_i' \geqslant 0$ и $x_i'' \geqslant 0$. Таким способом исходная задача линейного программирования может быть преобразована в эквивалентную ей задачу ЛП с неотрицательными переменными. Во многих реальных задачах некоторые ограничения записываются со знаками неравенства $\leqslant$ или $\geqslant$. Каждое неравенство может быть обращено в равенство добавлением неотрицательной переменной к ограничению типа $\leqslant$ или вычитанием неотрицательной переменной из ограничения типа $\geqslant$. Новые переменные, добавленные к ограничениям, называются *дополнительными переменными* (избыточными переменными). Исходные независимые переменные называются *структурными переменными* (основными переменными).

Решение $\boldsymbol{x}$ является *допустимым* для задачи ЛП (1.1), если оно удовлетворяет условиям $A\boldsymbol{x} = B$ и $\boldsymbol{x} \geqslant 0$. Набор всех допустимых решений называется *допустимым множеством.* Допустимое решение $\boldsymbol{x}^*$ называется *оптимальным решением* задачи ЛП (1.1), если $C^T\boldsymbol{x} \leqslant C^T\boldsymbol{x}^*$ для всех допустимых решений $\boldsymbol{x}$.

Множество S называется *выпуклым*, если все точки прямолинейного отрезка, соединяющего любые две точки из S, также принадлежат S. Другими словами, S является выпуклым множеством, тогда и только тогда, когда выпуклая линейная комбинация $\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in S$ для любых $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$ и $0 \leqslant \lambda \leqslant 1$. Легко убедиться в том, что решение $\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ также является допустимым. Следовательно, допустимое множество задачи ЛП всегда выпукло.

Точка $\boldsymbol{x}$ называется *угловой точкой* выпуклого множества S, если $\boldsymbol{x} \in S$ и $\boldsymbol{x}$ не может быть представлена в виде выпуклой линейной комбинации двух других различных точек из S. Было показано, что оптимальное решение задачи ЛП (1.1) соответствует угловой точке допустимого множества этой задачи при условии, что данное множество ограничено. Этот факт лежит в основе *симплексного алгоритма*, разработанного Г. Б. Данцигом [53] и представляющего собой очень эффективный метод решения задач линейного программирования. Попросту говоря, симплекс-метод проверяет только угловые точки допустимого множества, не рассматривая все остальные его точки. Вначале симплекс-метод выбирает некоторую угловую точку в качестве стартовой. Следующая угловая точка выбирается таким образом, чтобы увеличить значение целевой функции в задаче (1.1). Процедура повторяется до тех пор, пока происходит увеличение целевой функции. Последняя выбранная угловая точка является оптимальным решением. Для решения задач линейного программирования большой размерности или специальной структуры разработан ряд усовершенствованных методов, в частности, *модифицированный симплекс-метод*, *двойственный симплекс-метод*, *прямо-двойственный метод*[1], *метод декомпозиции Данцига—Вулфа* и *метод внутренней точки Кармаркара*.

1.2. Нелинейное программирование

Существует большое количество реальных задач, которые могут быть сформулированы как задачи нелинейного программирования (НЛП). В этих задачах нелинейные члены могут присутствовать как в целевой функции, так и в ограничениях. В общем виде задача НЛП может быть записана следующим образом:

$$\begin{cases} \min\ f(\boldsymbol{x}) \\ \text{при ограничениях:} \\ \qquad g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1, 2, \ldots, p. \end{cases} \tag{1.2}$$

Здесь минимизируется вещественнозначная функция $f(\boldsymbol{x})$ нескольких действительных переменных $\boldsymbol{x}$ с учетом ограничений $g_j(\boldsymbol{x})$. Если ограничения в (1.2) отсутствуют, то соответствующий вариант задачи НЛП именуют *задачей безусловного математического программирования*. Если функции $f(\boldsymbol{x})$ и $g_j(\boldsymbol{x})$, $j = 1, 2, \ldots, p$, выпуклы, получаем задачу *выпуклого программирования*. Если функция $f(\boldsymbol{x})$ может быть выражена в виде суммы $f(\boldsymbol{x}) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)$, тогда о задаче НЛП говорят как о задаче *сепарабельного программирования*. Если $f(\boldsymbol{x})$ — квадратичная функция, а все функции $g_j(\boldsymbol{x}), j = 1, 2, \ldots, p$, линейны, то соответствующая задача НЛП именуется задачей *квадратичного программирования*. Задача НЛП называется задачей *геометрического программирования*, если функции $f(\boldsymbol{x})$ и $g_j(\boldsymbol{x})$, $j = 1, 2, \ldots, p$, могут быть выражены как $\sum_j a_j \prod_{i=1}^n x_i^{b_{ij}}$, при этом $a_j > 0$ для всех индексов j.

В задаче НЛП (1.2) вектор $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ называется *вектором независимых переменных (вектором решений)*. Он состоит из n действительных переменных $x_1, x_2, \ldots, x_n$, именуемых *компонентами вектора решений*. Функция f вектора переменных $\boldsymbol{x}$ носит название *целевой функции*. Система неравенств $g_j(\boldsymbol{x}) \leqslant 0$, $j = 1, 2, \ldots, p$, называется *набором ограничений*. Множество, определяемое как

$$S = \left\{ \boldsymbol{x} \in \Re^n \;\middle|\; g_j(\boldsymbol{x}) \leqslant 0,\, j = 1, 2, \ldots, p \right\}, \tag{1.3}$$

называется областью допустимых решений задачи НЛП (*допустимым множеством*). Решение $\boldsymbol{x}$ из S называется *допустимым решением*. Задача нелинейного программирования (1.2) состоит в поиске решения $\boldsymbol{x}^* \in S$ такого, что

$$f(\boldsymbol{x}^*) \leqslant f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in S. \tag{1.4}$$

[1] В отечественной литературе этот метод обычно именуют *методом последовательного сокращения невязок. — Прим. ред.*

Решение $\boldsymbol{x}^*$ называется *оптимальным решением* и, в случае задачи (1.2), *минимальным решением.* Значение целевой функции $f(\boldsymbol{x})$ при $\boldsymbol{x} = \boldsymbol{x}^*$ называется *оптимумом.* Задачу максимизации, записываемую в виде

$$\begin{cases} \max\ f(\boldsymbol{x}) \\ \text{при ограничениях:} \\ \qquad g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1, 2, \ldots, p, \end{cases} \tag{1.5}$$

можно путем умножения целевой функции на -1 преобразовать в задачу минимизации, решаемую при тех же самых ограничениях. Иногда множество ограничений составляют не только неравенства, но также и равенства, например:

$$\begin{cases} g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1, 2, \ldots, p, \\ h_k(\boldsymbol{x}) = 0, \quad k = 1, 2, \ldots, q. \end{cases} \tag{1.6}$$

Если в задаче НЛП имеется q ограничений типа равенства и существует возможность решить систему уравнений $h_k(\boldsymbol{x}) = 0, \quad k = 1, 2, \ldots, q$, представив q переменных через остальные, то можно исключить эти переменные из задачи НЛП, воспользовавшись полученным представлением. Можно также исключить ограничения типа равенства, используя метод Лагранжа, основанный на преобразовании задачи с ограничениями в задачу без ограничений.

В ходе развития НЛП было разработано значительное число методов оптимизации, которые стали классическими. Эти методы учитывали специфику соответствующих задач НЛП и основывались на математической теории в сочетании с анализом структуры решаемых задач. Один из выдающихся результатов в области теории НЛП известен как «условия Куна—Таккера». Чтобы сформулировать эти условия, дадим вначале некоторые определения. Говорят, что ограничение типа неравенства $g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1, 2, \ldots, p$ является активным в точке $\boldsymbol{x}^*$, если $g_j(\boldsymbol{x}^*) = 0$. Точка $\boldsymbol{x}^*$, удовлетворяющая условию $g_j(\boldsymbol{x}^*) \leqslant 0$, называется регулярной, если векторы градиентов $\nabla g_j(\boldsymbol{x})$ всех активных ограничений линейно независимы. Пусть $\boldsymbol{x}^*$ — регулярная точка ограничений задачи НЛП (1.2). Предположим также, что все функции $f(\boldsymbol{x})$ и $g_j(\boldsymbol{x}), j = 1, 2, \ldots, p$, дифференцируемы. Если $\boldsymbol{x}^*$ является решением типа локального минимума, то существуют множители Лагранжа $\lambda_j, j = 1, 2, \ldots, p$, такие, что удовлетворяются следующие *условия Куна—Таккера*

$$\begin{cases} \nabla f(\boldsymbol{x}^*) + \sum\limits_{j=1}^{p} \lambda_j \nabla g_j(\boldsymbol{x}^*) = 0, \\ \lambda_j g_j(\boldsymbol{x}^*) = 0, \quad j = 1, 2, \ldots, p, \\ \lambda_j \geqslant 0, \quad j = 1, 2, \ldots, p. \end{cases} \tag{1.7}$$

Если все функции $f(\boldsymbol{x})$ и $g_j(\boldsymbol{x}), j = 1, 2, \ldots, p$, выпуклы и дифференцируемы, а точка $\boldsymbol{x}^*$ удовлетворяет условиям Куна—Таккера (1.7), можно доказать, что $\boldsymbol{x}^*$ является решением типа глобального минимума для задачи (1.2).

Рассмотрим теперь задачу безусловной оптимизации, заключающейся в минимизации некоторой вещественнозначной функции, определенной в области $\Re^n$. На практике вычисление первых или вторых производных функции