**Введение**

Большое количество производственных задач можно сформулировать в виде математической задачи на экстремум. К некоторым из этих задач применимы классические методы оптимизации и математического анализа. Но для значительной части задач производства и управления подобные методы не применимы или малоэффективны. Нахождение экстремума классическими методами зачастую приводит к тому, что на следующем этапе решения приходится решать новые задачи, ещё более сложные, чем поставленные первоначально. Даже в тех случаях, когда задача формулируется, как задача математического программирования, применение необходимых условий для построения оптимального решения очень часто не приводит к нужному результату. В производственных задачах оптимизации к максимуму целевой функции легче бывает подойти поэтапно, чем аналитически решить систему уравнений, определяемую возможными ограничениями поставленной задачи. И даже, в том случае, когда задача математического программирования может быть решена, проверка найденного решения может оказаться довольно сложной и тем сложнее, чем больше аргументов у функции.

Курсовая работа посвящена моделированию оптимизационных задач средствами динамического программирования.

Целью курсовой работы является обоснование метода динамического программирования при моделировании оптимизационных задач, решение задач аналитически и с помощью программных средств.

Объект исследования – оптимизационные модели, решаемые методом динамического программирования.

Предмет исследования – алгоритм метода динамического программирования.

Задачи работы:

1. изучить общий подход динамического программирования;
2. выявить оптимизационные модели, решаемые методом динамического программирования;
3. продемонстрировать применение метода динамического программирования при решение оптимизационных задач аналитически;
4. выполнить программную реализацию некоторых моделей.

Актуальность, выполненной работы можно обосновать тем, что решение ряда оптимизационных задач, например, производственного управления, можно упростить, если процесс управления осуществляется поэтапно, заменив нахождение точек экстремума целевой функции многих переменных многократным нахождением точек экстремума функции одного или небольшого числа переменных, что возможно, если воспользоваться методом динамического программирования.

Результатом работы является программы решения нескольких оптимизационных задач, таких как задача о рюкзаке, задачи о замене оборудования, задача о распределении инвестиций.

**Метод динамического программирования**

Динамическое программирование является разделом математической науки, в основе которой лежит изучение экстремальных задач управления (методы вычислительной математики, которые применяются для поиска экстремумов функций), планирование и разработка методов их решения, в котором процесс принятия решения разбивается на отдельные этапы. Также можно сказать, что динамическое программирование – это способ решения сложных задач путём разбиения их на более простые подзадачи. Он применим к задачам с оптимальной подструктурой, выглядящим как набор перекрывающихся подзадач, сложность которых чуть меньше исходной.

В общей постановке задачу динамического программирования можно сформулировать следующим образом. Управляемая физическая система , характеризуется определённым набором параметров. Требуется построить оптимальное решение , на множестве допустимых решений, переводящее систему из начального состояния в конечное состояние , обеспечив целевой функции нужный экстремум.

Алгоритмы динамического программирования используются для поиска решения не сразу для всей сложной задачи, а для поиска оптимального решения для нескольких более простых задач аналогичного содержания, на которые распадается исходная задача. Также данные алгоритмы могут применяться для подсчёта количества этих решений.

То есть в динамическом программировании задача разделяется на подзадачи, и решения этих подзадач объединяются вместе для достижения общего решения основной задачи.

Подзадачи могут быть рекурсивно вложены в более крупные задачи, тогда методы динамического программирования применимы, и существует связь между значением более крупной проблемы и значениями подзадач. В литературе такое соотношение называется уравнением Беллмана.

При использовании алгоритмического подхода «разделяй и властвуй», подзадача может быть решена несколько раз. Динамическое программирование решает каждую из этих подзадач только один раз, тем самым уменьшив количество вычислений, а затем сохраняет результат, избегая повторного вычисления на более позднем этапе, когда требуется решение для этой подзадачи.

Динамическое программирование используется для решения задач оптимизации (например, поиск кратчайшего пути), где может существовать множество решений, но интерес представляет только поиск оптимального решения.

Задачи должны иметь свойства перекрывающихся подзадач. Иными словами, решаемая задача может быть разбита на подзадачи, которые многократно используется, причём рекурсивный алгоритм решает одну и ту же подзадачу много раз, а не создаёт новую подзадачу. Например, числа Фибоначчи. В конечном итоге, оптимальное решение может быть построено из оптимальных решений подзадач.

Следует отметить, что во многих случаях алгоритмы перебора могут работать быстрее, чем алгоритмы динамического программирования, но они не гарантируют оптимальное решение задачи.

**Уравнение Беллмана**

Согласно Беллману, основной принцип оптимальности управления многошаговыми процессами может быть словесно выражен следующим образом: «Оптимальное поведение обладает тем свойством, что, каковы бы ни были исходное состояние и первоначальное решение, последующие решения должны составлять оптимальное поведение относительно состояния, получающегося в результате первоначального решения». Иными словами, любой участок оптимальной траектории, в том числе и завершающий, также являются оптимальным, а ошибки в управлении, приводящие к отклонениям от оптимальной траектории, впоследствии не могут быть исправлены.

Рассмотрим функции . Функции представляют собой максимальные значения сумм частных целевых функций , вычисляемые по всем допустимым «укороченным» наборам управлений . Иными словами, – условно-оптимальное значение целевой функции при переводе системы из состояния после шага с номером в конечное состояние ; условность оптимального значения состоит в том, что оно относится не ко всему процессу, а к его заключительной части, и зависит от выбора состояния , являющегося начальным для «укороченного» процесса. Тем самым функции , называются функциями Беллмана, характеризуют экстремальные свойства управляемой системы на последних шагах процесса. При этом имеет место соотношение:

|  |  |
| --- | --- |
|  | (1) |

Формула 1. Соотношение функции Беллмана.

Это выражение справедливо, так как состояние уже является конечным, дальнейших изменений состояний системы не происходит, и соответствующий экономический эффект равен 0.

Принцип оптимальности Беллмана, лежащий в основе метода динамического программирования решений рассматриваемых задач, выражается следующим основным функциональным уравнением:

|  |  |
| --- | --- |
|  | (2) |

Формула 2. Функциональное уравнение Беллмана.

в котором индекс изменяется по номерам всех шагов процесса в обратном порядке: ;

– значение целевой функции на -м шаге.

По своей структуре функциональное уравнение Беллмана является рекуррентным. Это означается, что в последовательности функций каждая предшествующая выражается через последующую.

Уравнение Беллмана было впервые применено к теории управления техническими системами и другим темам прикладной математики, а затем стало важным инструментом экономической теории.

Чтобы понять функциональное уравнение Беллмана, необходимо понять несколько основных понятий. Во-первых, любая задача оптимизации имеет какую-то цель: минимизация времени в пути, минимизация затрат, максимизация прибыли и т. д. Математическая функция, которая описывает эту задачу, называется целевой функцией.

Динамическое программирование разбивает задачу многопериодного планирования на более простые этапы в разные моменты времени. Следовательно, необходимо отслеживать, как ситуация с решениями меняется со временем. Информация о текущей ситуации, которая необходима для принятия правильного решения, называется «состоянием». Например, чтобы решить, сколько потреблять и тратить в каждый момент времени, люди должны знать (среди прочего) своё первоначальное богатство. Следовательно, богатство будет одной из их переменного состояния, но, вероятно, будут и другие.

Переменные, выбранные в любой данный момент времени, часто называю контрольными переменными. Например, учитывая их текущее состояние, люди могут решить, сколько потреблять сейчас. Выбор управляющих переменных теперь может быть эквивалентен выбору следующего состояния; в более общем случае на следующее состояние влияют другие факторы, помимо текущего контроля. Например, в простейшем случае сегодняшнее богатство и потребление могут точно определять завтрашнее богатство хотя, как правило, другие факторы также влияют на завтрашнее богатство.

Подход динамического программирования описывает оптимальный план, находя правило, которое сообщает, какими должны быть элементы управления, учитывая любое возможное значение состояния. Например, потребление зависит только от богатства, мы бы искали правило потребление как функцию богатства. Такое правило, определяющее элементы управления как функцию, называется функцией политики.

Наконец, по определению, оптимальным правилом принятия решения является то, которое достигает наилучшего возможного значения цели. Наилучшее возможное значение цели, записанное как функция состояния, называется функцией значения.

Ричард Беллман показал, что задача динамической оптимизации в дискретном времени может быть сформулирована в рекурсивной пошаговой форме, известной как обратная индукция, записав взаимосвязь между функцией значения в одном периоде и функцией значения в следующем периоде.

Существуют условия, которым должна удовлетворять общая задача оптимизации, чтобы её можно было описать методом динамического программирования:

* задача оптимизации формулируется как итоговый многошаговый процесс управления;
* целевая функция, является аддитивной и равна сумме целевых функций каждого шага оптимизации;
* выбор управления на каждом шаге зависит только от состояния самой системы на этом шаге и не влияет на предшествующие шаги (отсутствие обратной связи);
* состояние системы после каждого шага управления зависит исключительно от предшествующего состояния системы и этого управляющего воздействия (нет последействия), причём оно может быть записано в виде уравнения состояния системы;
* на каждом шаге управление зависит от конечного числа управляющих переменных, а состояния системы от конечного числа параметров;
* оптимальное управление представляет собой вектор , который определяется как последовательность оптимальных пошаговых управлений, число которых и определяет количество шагов задачи.

Основные свойства задач, в которых можно применять метод динамического программирования:

* задача должна допускать интерпретацию как многошаговый процесс принятия решения;
* задача должна быть определена для любого числа шагов и иметь структуру, не зависящую от их числа;
* при рассмотрении задачи на каждом шаге должно быть задано множество параметров, описывающих состояние системы;
* выбор решения (управления) на каждом шаге не должен оказывать влияния на предыдущие решения.

**Задачи, решаемые методом динамического программирования**

Особенность производственных задач, решаемых методом динамического программирования, заключается в том, что процесс, протекающий в системе, зависит либо от времени (от этапов), либо имеет многоступенчатую структуру. Метод решения задач динамического программирования состоит в том, что оптимальное управление строится поэтапно шаг за шагом. На каждом этапе оптимизируется только один шаг, но при этом учитывается изменение всего процесса, так как управление, оптимизирующее целевую функцию только для данного шага, может привести к неоптимальному эффекту всего процесса. Управление на каждом шаге должно быть оптимальным с точки зрения процесса в целом.

В основе вычислительных алгоритмов динамического программирования лежит принцип оптимальности Р. Беллмана. (Формула 2) Данный принцип включается в себя три основных этапа:

1. предварительный этап;
2. этап условной оптимизации;
3. этап безусловной оптимизации.

Предварительный этап проводится с целью уменьшения вычислительной работы на последующем этапе решения и, по существу, заключается в нахождении всех допустимых значений управлений и фазовых переменных , то есть фактически область определения функций или, в более сложных случаях, множеств, содержащих эти области определения. Иными словами, на данном этапе отбрасываются все заведомо неподходящие, нереализуемые значения фазовых и управляющих переменных. Проводится предварительный этап в естественном порядке от первого шага к последнему: , а опираются соответствующие расчёты на уравнение процесса . Данный этап особенно удобен при табличном способе задания функций, фигурирующих в условии задачи.

Условная оптимизация осуществляется поэтапно при движении из конечного состояния системы в первоначальное состояние путём построения на каждом этапе условно-оптимального управления и нахождения условно-оптимального значения функции цели для каждого шага.

Безусловная оптимизация осуществляется в обратном направлении: от первого шага к последнему, в результате чего находятся уже оптимальные управления на каждом шаге с точки зрения всего процесса. На втором этапе отрабатываются рекомендации, полученные на этапе условной оптимизации.

Много интересных многошаговых процессов принятия решений возникает при управлении производственными процессами. Рассмотрим некоторые из них.

Задача о замене оборудования. В настоящее время промышленные предприятия производят замену оборудования в сроки, диктуемые не на основе интуиции, а на основе математических расчётов. Управленцы промышленных предприятий должны владеть методами составления плана замены машинного оборудования, в целях оптимизации его использования. Задачу по замене оборудования можно рассматривать как многоэтапный процесс, к которому применимы методы динамического программирования. Применение методов динамического программирования позволяет максимизировать прибыль или минимизировать затраты.

Задача оптимального распределения инвестиций (ресурсов). В производственной практике очень часто возникают задачи на оптимальное распределение ресурсов между предприятиями или внутри предприятия. Кроме того, к задача оптимального распределения инвестиций можно отнести ещё ряд задач. Например, задачу о размещение по торговым и складским помещениям какого-либо товара, задачу о распределении средств между различными отраслями промышленности и т. п.

Задача о рюкзаке (Задача о загрузке транспортного средства). В рюкзак требуется погрузить несколько видом предметов, так, чтобы ценность рюкзака была максимальной. Также можно переформулировать данную задачу в задачу о загрузке транспортного средства. В транспортное средство требуется погрузить несколько видом груза, так, чтобы результат загрузки был эффективным. Например, максимизировать стоимость груза, размещённого в транспортном средстве, известна грузоподъёмность транспортного средства, вес единицы груза и соответствующая эффективность.

Вышеперечисленные задачи не составляют полный список производственных задач, решаемых динамическим программированием. Аппарат динамического программирования используется и в численных решениях классических функциональных уравнений, обыкновенных дифференциальных уравнений и дифференциальных уравнений с частными производными.

В разделе рассматриваются решения ряда производственных задач методом динамического программирования аналитически.

**Задача о замене оборудования**

Постановка задачи:

Разработать оптимальную стратегию замены оборудования возраста лет в плановом периоде продолжительностью лет, если известны:

– стоимость продукции, производимой в течение года на оборудовании возраста лет ();

– ежегодные расходы, связанные с эксплуатацией оборудования возраста лет ();

– остаточная стоимость оборудования возраста лет ();

– стоимость нового оборудования и расходы, связанные с установкой, наладкой и запуском.

В начале каждого года имеется две возможности: сохранить оборудование и получить прибыль или заменить его и получить прибыль . Прибыль от использования оборудования в последнем -м году планового периода запишется в следующем виде:

|  |  |
| --- | --- |
|  | (3) |

Формула 3. Прибыль от использования оборудования в -м году.

А прибыль от использования оборудования в период с -го по -й год:

|  |  |
| --- | --- |
|  | (4) |

Формула 4. Прибыль от использования оборудования в период с -го по -й год.

где – прибыль от использования оборудования в период с -го по -й год.

В случае, если оба управления («сохранить» и «заменить») приводят к одной и той же прибыли, то целесообразно выбрать управление «сохранить».

В курсовой работе был приведён пример аналитического решения данной задачи, а также перед разработкой программной реализации в среде MS Excel был автоматизирован процесс нахождения матрицы Беллмана в соответствии с его функциональным уравнением. Также было принято решение реализовать данный алгоритм на языке программирования Python. По описанной ранее математической модели было реализовано программное обеспечение для решения производственных задач на основе метода динамического программирования. Подробное описание программной реализации находится в тексте курсовой работы.

**Задача о рюкзаке**

Постановка задачи:

Дано предметов, -предмет имеет массу и стоимость . Необходимо выбрать из этих предметов такой набор, чтобы суммарная масса не превосходила заданной величины (вместимость рюкзака), а суммарная стоимость была максимальной.

Можно сформулировать данную задачу следующим образом. Пусть дано предметов, – вместимость рюкзака, – набор положительных целых весов, – соответствующий ему набор положительных целых стоимостей. Нужно найти набор бинарных величин , который обеспечит максимальную стоимость рюкзака, где , если предмет включён в набор, , если предмет не включён.

|  |  |
| --- | --- |
|  | (4) |

Формула 4. Целевая функция задачи о рюкзаке.

|  |  |
| --- | --- |
|  | (5) |

Формула 5. Ограничения задачи о рюкзаке.

Решение задачи о рюкзаке. Пусть есть максимальная стоимость предметов, которые можно уложить в рюкзак вместимости , если можно использовать только первые предметов, то есть назовём этот набор допустимых предметов для . Исходя из этого соображения получается, что и .

Найдём . Возможны два варианта:

1. Если предмет не попал в рюкзак. Тогда равно максимальной стоимости рюкзака с такой же вместимостью и набором допустимых предметов , то есть .
2. Если попал в рюкзак. Тогда равно максимальной стоимости рюкзака, где вес уменьшаем на вес -ого предмета и набор допустимых предметов плюс стоимость , то есть .

То есть это можно представить в следующем виде:

|  |  |
| --- | --- |
|  | (6) |

Формула 6. Возможные варианты для .

Получается, что: . Стоимость искомого набора равна , так как нужно найти максимальную стоимость рюкзака, где все предметы допустимы и вместимость рюкзака .

После того как мы определили максимальную стоимость рюкзака нужно восстановить предметы, которые будут входить в искомый рюкзак. Будем определять входит ли предмет в искомый набор. Начинаем с предмета , где . Для этого сравниваем со следующими значениями:

1. Максимальная стоимость рюкзака с такой же вместимостью и набором допустимых предметов , то есть
2. Максимальная стоимость рюкзака с вместимостью на меньше и набором допустимых предметов плюс стоимость , то есть .

Можно заметить, что при построении мы выбирали максимум их этих значений и записывали в . Тогда будем сравнивать с , если равны, тогда -предмет не входит в искомый набор, иначе входит.

В курсовой работе был приведён пример аналитического решения данной задачи, а также перед разработкой программной реализации в среде MS Excel был автоматизирован процесс нахождения матрицы Беллмана в соответствии с его функциональным уравнением. Также была применена надстройка «Поиск решения» для нахождения оптимального решения тестовой задачи. Ещё было принято решение реализовать данный алгоритм на языке программирования Rust. По описанной ранее математической модели было реализовано программное обеспечение для решения производственных задач на основе метода динамического программирования. Подробное описание программной реализации находится в тексте курсовой работы.

Так же хочется добавить, что данную задачу можно модифицировать и получить задачу о загрузке транспортного средства или задачу о целочисленном рюкзаке. Она будет формулироваться следующим образом.

Предположим, что имеется транспортное средство загружаемое различными типами предметов с весом и стоимостью . Максимальная грузоподъёмность равна . Определить максимальную стоимость груза, вес которого не более .

Математическая постановка задачи формулируется следующим образом.

|  |  |
| --- | --- |
|  | (6) |

Формула 6. Целевая функция задачи о загрузке транспортного средства.

|  |  |
| --- | --- |
|  | (7) |

Формула 7. Ограничения задачи о загрузке транспортного средства.

**Задача о распределении инвестиций**

Постановка задачи:

В производственное объединение входят предприятий . Руководство объединения решило инвестировать в свои предприятия условных единиц в общей сумме. Проведённые маркетинговые исследования прогнозируют величину ожидаемой прибыли каждого из предприятий в зависимости от объёма инвестированных средств. Требуется найти такое распределение инвестиций между предприятиями, которое обеспечило бы максимум суммарной ожидаемой прибыли.

Решение задачи о распределении инвестиций

Число шагов в данной задаче равно . Пусть – количество средств, имеющихся в наличии перед данным шагом, и характеризующих состояние системы на каждом шаге. Управление на -ом шаге выберем – количество средств, инвестируемых в -ое предприятие. Выигрыш на -ом шаге – это прибыль, которую приносит -ое предприятие при инвестировании в него средств . Если через выигрыш в целом обозначить общую прибыль , то он будет выглядеть следующим образом.

|  |  |
| --- | --- |
|  | (8) |

Формула 8. Общая прибыль в задаче о распределении инвестиций.

Если в наличии имеются средства в количестве условных единиц и в -ое предприятие инвестируется условных единиц, то для дальнейшего инвестирования остаётся условных единиц. Таким образом, если на -ом шаге система находилась в состоянии и выбрано управление , то на -шаге система будет находиться в состоянии , и, следовательно, функция перехода в новое состояние имеет вид: .

На последнем -ом шаге оптимальное управление соответствует количеству средств, имеющихся в наличии, а выигрыш равен доходу, приносимым последним предприятием.

Согласно принципу оптимальности Беллмана, управления на каждом шаге нужно выбирать так, чтобы оптимальной была сумма выигрышей на всех оставшихся до конца процесса шагах, включая выигрыш на данном шаге. Тогда функциональное уравнение Беллмана примет вид:

|  |  |
| --- | --- |
|  | (9) |

Формула 9. Общая прибыль в задаче о распределении инвестиций

В курсовой работе был приведён пример аналитического решения данной задачи, а также перед разработкой программной реализации в среде MS Excel был автоматизирован процесс нахождения оптимизации в соответствии с его функциональным уравнением. Также была применена надстройка «Поиск решения» для нахождения оптимального решения тестовой задачи. Подробное описание программной реализации находится в тексте курсовой работы.

**Заключение**

Тема курсовой работы посвящена актуальной проблеме реализации алгоритмов решения производственных оптимизационных задач на основе метода динамического программирования.

В ходе выполнения курсовой работы достигнуты следующие результаты:

1. Проанализирован общий подход динамического программирования к решению некоторых типов производственных оптимизационных задач.
2. Приводится аналитическое решение представленных задач с помощью метода динамического программирования.
3. Выполнена реализация алгоритма представленных оптимизационных задач.

Основным результатом выполненной курсовой работы является программная реализация алгоритма решения производственных задач методом динамического программирования.

Результаты работы можно применить для решения практических задач управления ресурсами экономических и производственных систем.