Метод программирования и схем ветвей в процессах решения задач дискретной оптимизации

Содержание

Введение

Дискретная оптимизация как раздел математики существует достаточно давно. Оптимизация - это выбор, т.е. то, чем постоянно приходится заниматься в повседневной жизни. Термином "оптимизация" в литературе обозначают процесс или последовательность операций, позволяющих получить уточненное решение. Хотя конечной целью оптимизации является отыскание наилучшего или "оптимального" решения, обычно приходится довольствоваться улучшением известных решений, а не доведением их до совершенства. Поэтому под оптимизацией понимают скорее стремление к совершенству, которое, возможно, и не будет достигнуто.

Необходимость принятия наилучших решений так же стара, как само человечество. Испокон веку люди, приступая к осуществлению своих мероприятий, раздумывали над их возможными последствиями и принимали решения, выбирая тем или другим образом зависящие от них параметры - способы организации мероприятий. Но до поры, до времени решения могли приниматься без специального математического анализа, просто на основе опыта и здравого смысла.

Возьмем пример: человек вышел утром из дому, чтобы ехать на работу. По ходу дела ему приходится принять целый ряд решений: брать ли с собой зонтик? В каком месте перейти улицу? Каким видом транспорта воспользоваться? И так далее. Разумеется, все эти решения человек принимает без специальных расчетов, просто опираясь на имеющийся у него опыт и на здравый смысл. Для обоснования таких решений никакая наука не нужна, да вряд ли понадобится и в дальнейшем.

Однако возьмем другой пример. Допусти, организуется работа городского транспорта. В нашем распоряжении имеется какое-то количество транспортных средств. Необходимо принять ряд решений, например: какое количество и каких транспортных средств направить по тому или другому маршруту? Как изменять частоту следования машин в зависимости от времени суток? Где поместить остановки? И так далее.

Эти решения являются гораздо более ответственными, чем решения предыдущего примера. В силу сложности явления последствия каждого из них не столь ясны; для того, чтобы представить себе эти последствия, нужно провести расчеты. А главное, от этих решений гораздо больше зависит. В первом примере неправильный выбор решения затронет интересы одного человека; во втором - может отразиться на деловой жизни целого города.

Наиболее сложно обстоит дело с принятием решений, когда речь идет о мероприятиях, опыта, в проведении которых еще не существует и, следовательно, здравому смыслу не на что опереться, а интуиция может обмануть. Пусть, например, составляется перспективный план развития вооружения на несколько лет вперед. Образцы вооружения, о которых может идти речь, еще не существуют, никакого опыта их применения нет. При планировании приходится опираться на большое количество данных, относящихся не столько к прошлому опыту, сколько к предвидимому будущему. Выбранное решение должно по возможности гарантировать нас от ошибок, связанных с неточным прогнозированием, и быть достаточно эффективным для широкого круга условий. Для обоснования такого решения приводится в действие сложная система математических расчетов.

Вообще, чем сложнее организуемое мероприятие, чем больше вкладывается в него материальных средств, чем шире спектр его возможных последствий, тем менее допустимы так называемые "волевые" решения, не опирающиеся на научный расчет, и тем большее значение получает совокупность научных методов, позволяющих заранее оценить последствия каждого решения, заранее отбросить недопустимые варианты и рекомендовать те, которые представляются наиболее удачными.

1. Дискретные оптимизационные задачи

Дискретные оптимизационные задачи находят широкое применение в различных областях, где используются математические методы для анализа происходящих там процессов. Необходимость решения таких задач приводит к тому, что дискретная оптимизация становится важным элементом образования специалистов, связанных с её применением при решении задач, возникающих в приложениях. Поэтому нам представляется, что технология решения задач дискретного программирования должна стать одной из важных составных частей современного математического образования для специалистов по прикладной математике.

Дискретные оптимизационные задачи можно решать двумя методами: метод дискретного программирования и метод ветвей и границ. Они будут рассмотрены на примере задачи коммивояжера.

1.1 Постановка задач дискретного программирования

Под задачей дискретного программирования (дискретной оптимизации) понимается задача математического программирования

F(x0) = min f(x), x є G,

множество допустимых решений которой конечно, т.е. O ≤ |G| = N < ∞, где |G| — число элементов множества G. В силу конечности G все допустимые решения можно пронумеровать: x1, x2, ..., xN, вычислить f(xi), i= 1,2,..., N, и найти наименьшее значение.

Во многих задачах условия дискретности отделены от других условий, т.е. если x = (x1, x2, ... ,xn), то xj є Gj = (x j 1, xj2, ...,xjki), kj > 2. Поэтому N = = $\prod_{j=1}^{n}|G|$ = $\prod_{j=1}^{n}k_j$ > 2n, отсюда видно, что с ростом числа переменных объем вычислительной работы резко возрастает.

1.2 Алгоритм метода ветвей и границ

Рассмотрим задачу в виде:

f(x0)=min f(x), x є G, |G|=N < ∞.

Алгоритм ветвей и границ основан на следующих построениях, позволяющих уменьшить объем перебора.

Вычисление оценки. Пусть G' $\subset$ G, тогда φ(G') называется нижней оценкой, если для любого x є G' выполняется неравенство f(x) ≥ φ(G').

Ветвление (разбиение множества G на подмножества). Положим

G0 = G и разобьем множество G0 на r1 непересекающихся подмножеств

/O

$G_1^1,\ G_2^1,\ \dots,\ G_{r_1}^1,$ G0 = $G_1^1 \bigcup_{i=1}^{r_1} G_i^1,\ G_i^1 \cap G_j^1 =$, i ≠ j.

Этот шаг алгоритма считаем начальным, имеющим номер 0. Рассмотрим шаг алгоритма с номером k. Пусть $G_1^1,\ G_2^1,\ \dots,\ G_{r_1}^1$ — множества, еще не подвергавшиеся разбиению. Выберем одно из этих множеств G_i^k и разобьем его на непересекающиеся подмножества: $G_i^k = \bigcup_{i=1}^{r_1} G_{i,j}^k.$

Выполним модификацию списка множеств, еще не подвергавшихся разбиению. Заменим множество G_i^k множествами $G_{i,j}^k$ и множества, еще не подвергшиеся разбиению, переобозначим: $G_1^{k+1},\ G_2^{k+1},\ \dots,\ G_{r_{k+1}}^{k+1}$.

Эти множества образуют список задач для ветвления. Выберем одно из них и снова повторим процедуру разбиения.

Описанную процедуру разбиения можно представить в виде дерева (рис. 1)

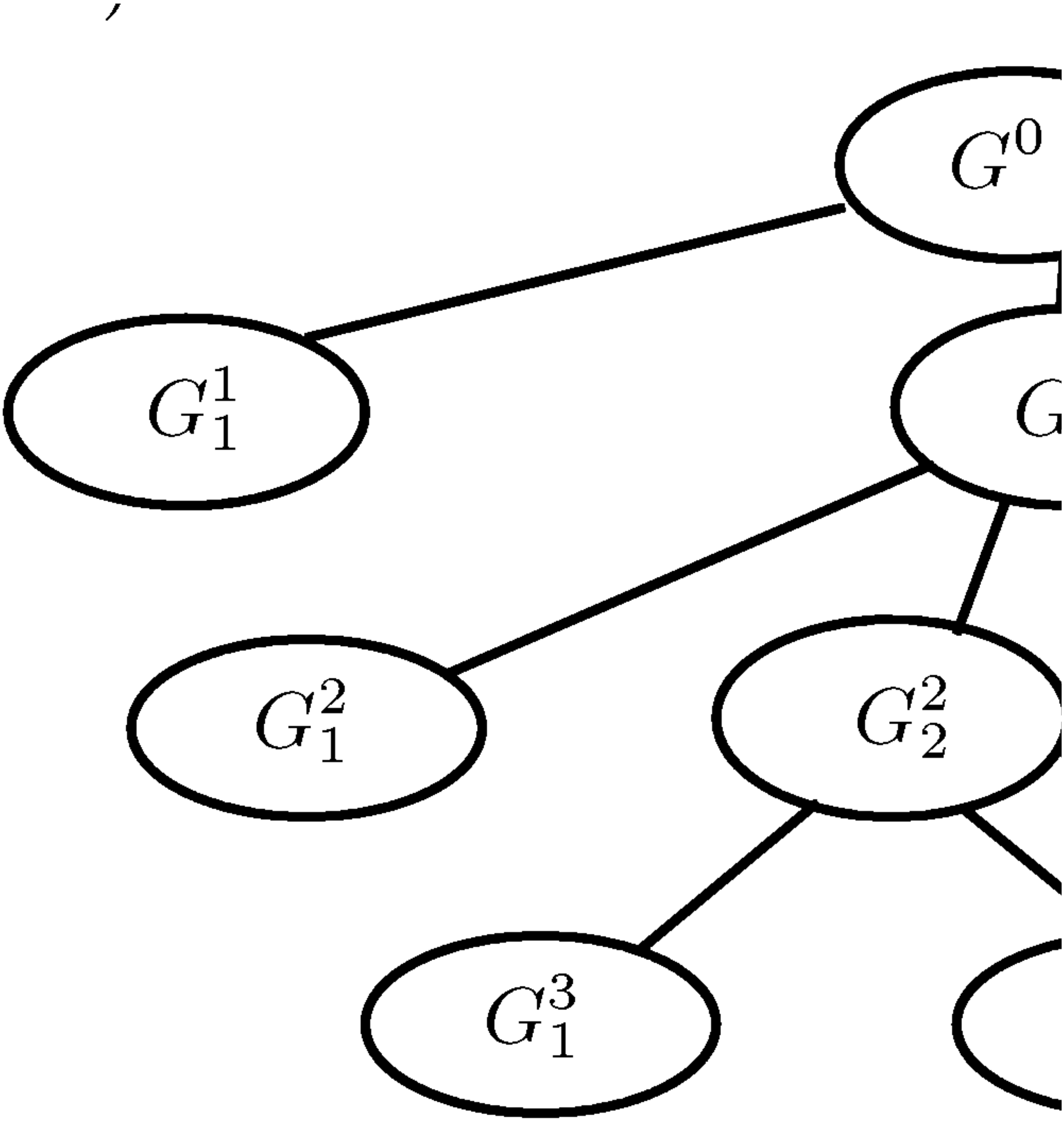

Рис. 1

Пересчет оценок. Если G1 ⊂ G2, то $\min_{x \in G_1} f(x) \geq \min_{x \in G_2} f(x)$.

Поэтому, разбивая в процессе ветвления подмножество G’ ⊂ G на непересекающиеся подмножества $G_1^1, G_2^1, \dots,$ G's, G' = $\bigcup_{i=1}^{s} G_i'$, будем предполагать, что φ(G1’) ≥ φ (G’), причем хотя бы для некоторых номеров i0 выполняется строгое неравенство φ(G_{i_0}') ≥ φ (G’).

Вычисление планов (допустимых решений). Если на шаге ветвления с номером k известен план xk, на шаге с номером (k + 1) — план xk+1 и если f(xk+1) < f(xk), то план xk забывается и вместо него сохраняется план xk+1. Наилучшее из полученных допустимых решений принято называть рекордом.

Признак оптимальности. Пусть G = $\bigcup_{i=1}^{s} G_i$. Тогда план $\bar{x}$ является оптимальным, т.е. $\bar{x} = \bar{x}^0$, если выполняется условие

f($\bar{x}$) = φ(Gv) ≤ φ(Gi), i=1, 2, ... , s.

Оценка точности приближенных решений. Пусть G = $\bigcup_{i=1}^{s} G_i$,

φ0 = $\min_{1\le i\le s}$ φ(Gj), xk — рекорд; тогда имеет место следующее неравенство:

φ0 ≤ f(x0) ≤ f(xk).

Разность Δ = f(xk) - φ0 является оценкой гарантированного отклонения рекорда xk от оптимума x0. Из приведенного неравенства следует, что для ветвления необходимо выбрать множество с минимальным значением нижней оценки.

Правило отсева. Пусть снова G = $\bigcup_{i=1}^{s} G_i$, x0 — оптимум, xk — рекорд. Если φ(Gr) > f(xk), то множество Gr можно отсеять, т.е. исключить из дальнейшего рассмотрения, так как оно не может содержать оптимальных решений. Действительно, пусть x є G; тогда в силу определения оценки f(x) ≥ φ(G') имеем f(x) ≥ φ(Gr) > f(xk) ≥ f(x0).

Правило φ(Gr) > f(xk) гарантирует, что в процессе работы алгоритма ни одно из подмножеств Gr, в которых содержится точное решение x0, не будет отсеяно. Более сильное правило φ(Gr) ≥ f(xk) гарантирует, что хотя бы одно оптимальное решение будет найдено, оно и применяется при практическом решении задач.

Конечность алгоритма. Конечность алгоритма следует из конечности множества G.

Под методом ветвей и границ понимается алгоритм решения задачи, имеющий древовидную структуру поиска оптимального решения и использующий результаты решения оценочных задач. Древовидная структура называется обычно деревом ветвления.

Эффективность алгоритма ветвей и границ определяется числом решенных задач. Решение задачи состоит из двух основных этапов. На первом этапе находится оптимальное решение (или близкое к нему). На втором этапе производится доказательство оптимальности полученного решения. Второй этап, как правило, оказывается более трудоемким, чем первый. Это означает, что число подзадач, решаемых до получения оптимума, может оказаться существенно меньше числа подзадач, решаемых для доказательства оптимальности.

2. Постановка задачи коммивояжера

Классическая задача коммивояжера (ЗК) формулируется следующим образом: имеется полный взвешенный ориентированный граф без петель G с множеством вершин N = {1, 2,…, n}; веса всех дуг неотрицательны; в этом графе требуется найти гамильтонов цикл минимального веса.

Исходную информацию по ЗК считаем представленной в виде матрицы размера nxn. S = {sij}, где sij – вес дуги (i, j) графа G, i = $\overline{1,n}$, j = $\overline{1,n}$, i ≠ j; все элементы главной диагонали матрицы S являются нулями.

В типовой интерпретации вершины 1, 2,…, n графа G – это города. Дуги отображают возможные элементарные переходы. Коммивояжеру, изначально находящемуся в городе 1, необходимо обойти все остальные города, побывав в каждом из них ровно по одному разу, и затем вернуться в город 1. Веса дуг графа трактуются как длины соответствующих элементарных переходов. Требуется найти имеющий минимальную длину допустимый (т.е. удовлетворяющий наложенным требованиям) маршрут коммивояжера. С учетом других возможных интерпретаций, на матрицу S требование симметричности не налагается, не считается обязательным и выполнение неравенства треугольника.

3. Задача коммивояжера методом динамического

программирования

Под методом ветвей и границ понимается алгоритм решения задачи, имеющий древовидную структуру поиска оптимального решения и использующий результаты решения оценочных задач. Древовидная структура называется обычно деревом ветвления.

Один из основных алгоритмов решения ЗК основан на принципе динамического программирования. При изложении этого алгоритма будем придерживаться терминологии, соответствующей приведенной типовой интерпретации задачи.

Пусть i – произвольный город (i O N), а V – любое подмножество городов, не содержащее города 1 и города i. Через M(i, V) обозначим совокупность путей, каждый из которых начинается в городе i, завершается в городе 1 и проходит в качестве промежуточных только через города множества V, заходя в каждый из них ровно по одному разу. Через B(i, V) обозначим длину кратчайшего пути множества M(i, V). Для решаемой задачи B(i, V) – функция Беллмана. Как очевидно, B(1, {2, 3, …, n}) – искомая минимальная длина простого (без самопересечений) замкнутого пути, проходящего через все города.

Если V – одноэлементное множество, V ={j}, где j ≠ 1 и j ≠ i, то совокупность M (i, V) состоит из единственного пути μ = (i, j, 1). Поэтому

$B(i,\{j\}) = s_{ij} + s_{j1}$ i ON, j O {2, 3,…, n}, j ≠ i.(1.1)

Предположим, что значения функции B(i, V) для всех i ON {1} и всех возможных k-элементных (k < n – 1) множеств V уже вычислены. Тогда значение B(i, V'), где V' – произвольное (k + 1)-элементное подмножество совокупности N {1, i}, вычисляется по формуле

$B(i,V') = \min_{j\in V'}(s_{ij} + B(j,V'\setminus\{j\}))$. (1.2)

Уравнения (1.1)–(1.12) – рекуррентные соотношения динамического программирования для решения задачи коммивояжера, они обеспечивают реализацию обратного метода Беллмана. Вычислительная сложность задачи равна $C \times 2^{n-2} \times n^2$,где C – произвольная константа (C > 0), n – число городов.

Пример 1.1 Решить задачу коммивояжера, определяемую матрицей:

$$S = \begin{pmatrix} 0 & 4 & 3 & 4 \\ 2 & 0 & 5 & 2 \\ 6 & 2 & 0 & 1 \\ 1 & 5 & 4 & 0 \end{pmatrix}$$

Сначала, пользуясь формулой (1.1), определяем значения B(i, {j}):

B(2, {3}) = 5 + 6 = 11; B(3, {2}) = 2 + 2 = 4; B(4, {2}) = 5 + 2 = 7;

B(2, {4}) = 2 + 1 = 3; B(3, {4}) = 1 + 1 = 2; B(4, {3}) = 4 + 6 = 10.

Далее по формуле (1.2) последовательно получаем (в левой части каждого из ниже записанных равенств выделены те значения параметра j, на которых при подсчете реализуется указанный в правой части (1.2) минимум):

B(2, {3, 4}) = min [s23 + B(3,{4}); s24 + B(4,{3})] = min(5 + 2; 2 + 10)=7;

B(3, {2, 4}) = min [s32 +B(2,{ 4}); s34 + B(4,{ 2})] = min(2 + 3; 1 + 7)=5;

B(4, {2, 3}) = min [s42 + B(2,{ 3}); s43 + B(3,{ 2})] = min(5 + 11; 4 +4)=8;

B(1, {2, 3, 4}) = min [s12 + B(2,{3,4}) s13 + B(3,{ 2,4}) s14 + B(4,{2,3 })] =

= min (4 +7; 3 +5; 4 + 8) = 8.

Итак, оптимальное значение критерия в рассматриваемом примере равно 8.

Выполненные выделения позволяют определить оптимальный маршрут. Он следующий:

1 ® 3 ® 2 ® 4 ® 1.

Для записи соотношений, по которым реализуется прямой метод Беллмана, введем новые обозначения. Пусть M'(V, i) – совокупность путей, каждый из которых начинается в городе 1, проходит в качестве промежуточных только через города подмножества V, заходя в каждый из них ровно по одному разу, и завершается в городе i; здесь, как и ранее, i – произвольный город (i О N), а V – любое подмножество N, не содержащее городов 1 и i. Длину кратчайшего пути множества M'(V, i) обозначим B*(V, i). Как очевидно, B*({2, 3, …, n}, 1) – искомая минимальная длина простого (без самопересечений) замкнутого пути, проходящего через все города. Если V – одноэлементное множество, V = {j}, где j ≠ 1 и j ≠ i , то совокупность M'(V, i) состоит из единственного пути μ = (1, j, i). Поэтому

$$B^*(\{j\},i) = s_{1j} + s_{ji},\ i \in N,\ j \in \{2,3,\dots,n\},\ j \neq i \quad (1.3)$$

Предположим, что значения функции B*(V, i) для всех i О N и всех возможных k-элементных (k < n – 1) множеств V уже вычислены. Тогда значение B*(V', i), где V' – произвольное (k + 1)- элементное подмножество совокупности N {1, i}, вычисляется по формуле

$$B^*(V',i) = \min_{j \in V'}(B^*(V' \setminus \{j\}, j) + s_{ji}) \quad (1.4)$$

Уравнения (1.3)–(1.4) – рекуррентные соотношения динамического программирования для решения классической задачи коммивояжера, они обеспечивают реализацию прямого метода Беллмана.

Пример 1.2 Методом прямого счета решить задачу коммивояжера, определяемую матрицей:

$$S = \begin{pmatrix} 0 & 4 & 3 & 4 \\ 2 & 0 & 5 & 2 \\ 6 & 2 & 0 & 1 \\ 1 & 5 & 4 & 0 \end{pmatrix}$$

(заметим, что матрица S в данном примере та же, что и в предыдущем).

Сначала, пользуясь формулой (1.3), определяем значения B*({j }, i):

B*({2}, 3) = 4 + 5 = 9; B*({3}, 2) = 3 + 2 = 5; B*({4}, 2) = 4 + 5 = 9;

B*({2}, 4) = 4 + 2 = 6; B*({3}, 4) = 3 + 1 = 4; B*({4}, 3) = 4 + 4 = 8.

Далее по формуле (1.4) последовательно получаем (в левой части каждого из ниже записанных равенств выделены те значения параметра j, на которых при подсчете реализуется указанный в правой части (1.4) минимум):

B*({2, 3}, 4) = min [B*({2}, 3) + s34; B*({3}, 2) + s24] = min(9 + 1; 5 + 2)= 7;

B*({2, 4}, 3) = min [B*({2}, 4) + s43; B*({4}, 2) + s23] = min(6 + 4; 9 + 5)= 10;

B*({3, 4}, 2}) = min [B*({3}, 4) + s42; B*({4}, 3) + s32] = min(4 + 5; 8

Страницы: **1** 2 3

Похожие рефераты:
Задачи линейного программирования. Алгоритм Флойда
Постановка задач линейного программирования. Примеры экономических задач, сводящихся к задачам линейного программирования. Допустимые и оптимальные решения. Алгоритм Флойда — алгоритм для нахождения кратчайших путей между любыми двумя узлами сети.
Анализ и выбор решений на основе нечеткой монотонной экспертной информации
Выбор представления экспертной информации о предметной области в системе; выбор и (или) обоснование подхода к принятию решения (ПР) на основе этой информации; разработка алгоритмов, реализующих выбранный подход к ПР.
Механизм бектрекинга
Составление алгоритма нахождения подмножества с заданной суммой элементов. Оценка количества отсекаемых вариантов. Добавление механизма бектрекинга в переборное решение задачи: организация отскока в переборе вариантов назад, если идти вперёд нельзя.
Методы решения задачи о рюкзаке
Задача о ранце как задача комбинаторной оптимизации. Задача о загрузке, рюкзаке, ранце. Постановка и NP-полнота задачи. Классификация методов решения задачи о рюкзаке. Динамическое программирование. Метод ветвей и границ. Сравнительный анализ методов.
Основные принципы решения транспортной задачи
Постановка задачи о коммивояжере. Нахождение оптимального решения с применением метода ветвей и границ. Основной принцип этого метода, порядок его применения. Использование метода верхних оценок в процедуре построения дерева возможных вариантов.
Дискретное программирование
Модель дискретного программирования как способ нахождения максимума целевой функции на множестве. Коды на языках Java и C# для решения заданий с неделимостями. Применение методов итерации Гомори и "ветвей и границ". Описание решения комбинаторных задач.
Алгоритмы сортировки, поиска кратчайшего пути в графе и поиска покрытия, близкого к кратчайшему
Алгоритм сортировки Шейкер: математическое описание задачи и описание алгоритма. Алгоритм покрытия: построение одного кратчайшего покрытия. Описание схемы и работы алгоритма на графах: нахождение кратчайшего пути. Контрольные примеры работы алгоритмов.
Методы исследования операций
Методы исследования операций и их использование в организационном управлении. Общая задача линейного программирования и некоторые методы ее решения. Теория двойственности и двойственные оценки в анализе решений линейных оптимизационных моделей.
Алгоритмизация задач
Средства формализации процесса определения спецификаций. Назначение языка (PSL) и анализатора определения задач (PSA). Разработка алгоритма решения задачи, критерии оценки его сложности. Локальный и глобальный уровни повышения эффективности алгоритмов.
Оптимальное распределение средств на расширение производства
Применение принципа оптимальности в задачах на рациональное распределение средств на расширение производства. Понятие динамического программирования и теоретические основы рекуррентной природы вычислительной схемы Беллмана в среде Microsoft Excel.
Выбор параметров контроля с использованием метода динамического программирования и метода ветвей и границ
Сущность и особенности выполнения метода динамического программирования. Решение математической задачи, принцип оптимальности по затратам, ручной счёт и листинг программы. Применение метода ветвей и границ, его основные преимущества и недостатки.
Эйлеровы и гамильтоновы графы
Министерство народного образования Республики Дагестан Дагестанский Государственный Университет Курсовая работа Программирование задач на графах
Проектирование модели для определения времени простоя станков на машиностроительном предприятии
Форма организации основного переменно-поточного производства. Особенности переналадки станков как задача динамического программирования. Общая характеристика алгоритма формирования метода ветвей и границ. Сущность понятия комбинаторная конфигурация.
Алгоритмы на графах. Независимые и доминирующие множества
Методы поиска подмножеств множества вершин V графа G, удовлетворяющих определенным условиям и свойствам. Понятие независимых множеств и порядок их генерации. Определение доминирующего множества. Основные этапы решения задачи о наименьшем разбиении.
Использование метода ветвей и границ при адаптации рабочей нагрузки к параметрам вычислительного процесса
Нахождение рационального порядка следования запросов для обеспечения максимального критерия эффективности использования компонентов вычислительного процесса в системе. Метод ветвей и границ для максимально быстрого выполнения вычислительного процесса.
Динамическое программирование и вариационное исчисление
Постановка задачи динамического программирования. Поведение динамической системы как функция начального состояния. Математическая формулировка задачи оптимального управления. Метод динамического программирования. Дискретная форма вариационной задачи.
Комбинаторные задачи
Решение задач по информатике, перебор различных комбинаторных конфигураций объектов и выбор наилучшего, с точки зрения условия задачи. Генерация k-элементных подмножеств, всех подмножеств данного множества, всех перестановок n-элементного множества.
Симплекс метод решения задачи линейного программирования
Описание симплекс метода решения задачи линейного программирования. Решение задачи методом Литла на нахождение кратчайшего пути в графе, заданном графически в виде чертежа. Из чертежа записываем матрицу расстояний и поэтапно находим кратчайший путь.
Решение линейных интегральных уравнений
Основные леммы и теоремы для решения линейных интегральных уравнений методом итераций. Применение информационных технологий для вычисления функции, построение алгоритма для определения уравнения по ядру и отрезку интегрирования и правой части уравнения.
Лекции по вычислительной математике
Бельский Аркадий Александрович. Вычислительная математика. Часть 2. Лекция 1 Вычислительная математика Специальность ПО 5-й семестр Конспект лекций

Категории

- Авиация и космонавтика (327)
- Административное право (149)
- Арбитражный процесс (29)
- Архитектура (130)
- Астрология (4)
- Астрономия (186)
- Банковское дело (2917)
- Безопасность жизнедеятельности (1880)
- Биографии (3553)
- Биология (3097)
- Биология и химия (1111)
- Биржевое дело (79)
- Ботаника и сельское хоз-во (2362)
- Бухгалтерский учет и аудит (4899)
- Валютные отношения (70)
- Ветеринария (56)
- Военная кафедра (636)
- География (3743)
- Геодезия (60)
- Геология (1084)
- Геополитика (49)
- Государство и право (13144)
- Гражданское право и процесс (543)
- Делопроизводство (32)
- Деньги и кредит (116)
- Естествознание (137)
- Журналистика (637)
- Зоология (40)
- Издательское дело и полиграфия (271)
- Инвестиции (120)
- Иностранный язык (4422)
- Информатика (74)
- Информатика, программирование (7324)
- Исторические личности (436)
- История (10671)
- История техники (672)
- Кибернетика (83)
- Коммуникации и связь (2320)
- Компьютерные науки (75)
- Косметология (20)
- Краеведение и этнография (528)
- Краткое содержание произведений (963)
- Криминалистика (136)
- Криминология (53)
- Криптология (5)
- Кулинария (955)
- Культура и искусство (5433)
- Культурология (586)
- Литература : зарубежная (2101)
- Литература и русский язык (5460)
- Логика (69)
- Логистика (29)
- Маркетинг (4459)
- Математика (2533)
- Медицина, здоровье (9201)
- Медицинские науки (100)
- Международное публичное право (78)
- Международное частное право (38)
- Международные отношения (2158)
- Менеджмент (8171)

-
- Обратная связь
- Реклама
- Добавить работу