Метод ветвей и границ

Виктор Васильевич Лепин

- Метод *Метод ветвей и границ (Branch and Bound)* (далее сокращенно B&B) — это лишь часть большого семейства методов.

- Метод *Метод ветвей и границ (Branch and Bound)* (далее сокращенно B&B) — это лишь часть большого семейства методов.
- Его подэтапы могут выполняться по-разному в зависимости от конкретной задачи, доступных программных инструментов и навыков разработчика алгоритма.

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:
  1. Он рекурсивно разбивает пространство поиска на меньшие пространства,

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:
  1. Он рекурсивно разбивает пространство поиска на меньшие пространства,
  2. а затем максимизирует $f(x)$ на этих меньших пространствах.

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:
  1. Он рекурсивно разбивает пространство поиска на меньшие пространства,
  2. а затем максимизирует $f(x)$ на этих меньших пространствах.
- Расщепление пространства поиска называется ветвлением.

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:
  1. Он рекурсивно разбивает пространство поиска на меньшие пространства,
  2. а затем максимизирует $f(x)$ на этих меньших пространствах.
- Расщепление пространства поиска называется ветвлением.
- В процессе ветвления мы не должны терять допустимые решения:

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:
    1. Он рекурсивно разбивает пространство поиска на меньшие пространства,
    2. а затем максимизирует $f(x)$ на этих меньших пространствах.
- Расщепление пространства поиска называется ветвлением.
- В процессе ветвления мы не должны терять допустимые решения:

- Цель алгоритма B&B — найти значение x, которое максимизирует или минимизирует значение функции $f(x)$, называемой целевой функцией, среди некоторого множества S допустимых или возможных решений.
- Множество S называется пространством поиска или допустимой областью.
- Далее предполагается, что требуется максимизация $f(x)$; это предположение делается без ограничения общности, так как минимальное значение $f(x)$ можно найти, найдя максимум $g(x) = -f(x)$.
- Алгоритм B&B работает по двум принципам:
  1. Он рекурсивно разбивает пространство поиска на меньшие пространства,
  2. а затем максимизирует $f(x)$ на этих меньших пространствах.
- Расщепление пространства поиска называется ветвлением.
- В процессе ветвления мы не должны терять допустимые решения: объединение допустимых областей задач сыновей должно давать допустимую область их родителя.

Определение 1

Ветвлением множества $d \subseteq S$ назовем функцию $b : d \to \{d_1, \ldots, d_k\}$, $d_k \subset d$, $d_k \neq \emptyset$, $\bigcup_{k=1}^{N} d_k = d$, $k = 1, \ldots, N$, разбивающую множество d на несобственные подмножества.

Определение 1

Ветвлением множества $d \subseteq S$ назовем функцию $b : d \to \{d_1, \ldots, d_k\}$, $d_k \subset d$, $d_k \neq \emptyset$, $\bigcup_{k=1}^{N} d_k = d$, $k = 1, \ldots, N$, разбивающую множество d на несобственные подмножества.

- Само по себе рекурсивное ветвление означало бы перебор всех возможных решений методом перебора.

Определение 1

Ветвлением множества $d \subseteq S$ назовем функцию $b : d \to \{d_1, \ldots, d_k\}$, $d_k \subset d$, $d_k \neq \emptyset$, $\bigcup_{k=1}^{N} d_k = d$, $k = 1, \ldots, N$, разбивающую множество d на несобственные подмножества.

- Само по себе рекурсивное ветвление означало бы перебор всех возможных решений методом перебора.
- Чтобы повысить производительность поиска методом перебора, алгоритм B&B отслеживает границы максимума, который он пытается найти, и использует эти границы для «сокращения» пространства поиска, исключая возможные решения, которые, как он может доказать, не будут содержать оптимальное решение.

Определение 2

Числовая функция U называется верхней границей функционала f на множестве d, если выполнены следующие свойства:

1. $U(d) \geq \max_{x \in d} f(x)$;

Определение 2

Числовая функция U называется верхней границей функционала f на множестве d, если выполнены следующие свойства:

1. $U(d) \geq \max_{x \in d} f(x)$;
2. $U(\{x\}) = f(x)$, т. е. на одноэлементном множестве $\{x\}$ значение функции H совпадает со значением функционала от x.

Определение 2

Числовая функция U называется верхней границей функционала f на множестве d, если выполнены следующие свойства:

1. $U(d) \geq \max_{x \in d} f(x)$;
2. $U(\{x\}) = f(x)$, т. е. на одноэлементном множестве $\{x\}$ значение функции H совпадает со значением функционала от x.

Определение 2

Числовая функция U называется верхней границей функционала f на множестве d, если выполнены следующие свойства:

1. $U(d) \geq \max_{x \in d} f(x)$;
2. $U(\{x\}) = f(x)$, т. е. на одноэлементном множестве $\{x\}$ значение функции H совпадает со значением функционала от x.

- Назовем рекордом и обозначим его x^0, наилучшее из найденных допустимое решение.

Определение 2

Числовая функция U называется верхней границей функционала f на множестве d, если выполнены следующие свойства:

1. $U(d) \geq \max_{x \in d} f(x)$;
2. $U(\{x\}) = f(x)$, т. е. на одноэлементном множестве $\{x\}$ значение функции H совпадает со значением функционала от x.

- Назовем рекордом и обозначим его x^0, наилучшее из найденных допустимое решение.
- Величина $f(x^0)$ является нижней границей функционала задачи.

Определение 2

Числовая функция U называется верхней границей функционала f на множестве d, если выполнены следующие свойства:

1. $U(d) \geq \max_{x \in d} f(x)$;
2. $U(\{x\}) = f(x)$, т. е. на одноэлементном множестве $\{x\}$ значение функции H совпадает со значением функционала от x.

- Назовем рекордом и обозначим его x^0, наилучшее из найденных допустимое решение.
- Величина $f(x^0)$ является нижней границей функционала задачи.
- Сначала рекорд x^0 либо произвольное допустимое решение, либо не известен.

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.
- Пусть $t_1, \ldots, t_L$ множество не отсеченных подмножеств решений. (Первоначально $L = 1$, $t_1 = S$.)

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.
- Пусть $t_1, \ldots, t_L$ множество не отсеченных подмножеств решений. (Первоначально $L = 1$, $t_1 = S$.)
- Множество t_i, $1 \leq i \leq L$ отсекается в одном из двух, последовательно проверяемых, случаев:

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.
- Пусть $t_1, \ldots, t_L$ множество не отсеченных подмножеств решений. (Первоначально $L = 1$, $t_1 = S$.)
- Множество t_i, $1 \leq i \leq L$ отсекается в одном из двух, последовательно проверяемых, случаев:
    1. если $U(t_i) \leq f(x^0)$;

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.
- Пусть $t_1, \ldots, t_L$ множество не отсеченных подмножеств решений. (Первоначально $L = 1$, $t_1 = S$.)
- Множество t_i, $1 \leq i \leq L$ отсекается в одном из двух, последовательно проверяемых, случаев:
  1. если $U(t_i) \leq f(x^0)$;
  2. если $U(\{t_i\}) = f(t_i) > f(x^0)$, т. е. одноэлементное множество отсекается всегда.

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.
- Пусть $t_1, \dots, t_L$ множество не отсеченных подмножеств решений. (Первоначально $L = 1$, $t_1 = S$.)
- Множество t_i, $1 \le i \le L$ отсекается в одном из двух, последовательно проверяемых, случаев:
  1. если $U(t_i) \le f(x^0)$;
  2. если $U(\{t_i\}) = f(t_i) > f(x^0)$, т. е. одноэлементное множество отсекается всегда.
- Последнея проверка имеет место, так как одноэлементное множество может быть не удалено по критерию 1.

- Алгоритм ветвей и границ последовательно выполняет итерации (шаги).
- На очередном шаге алгоритма выбирается подмножество решений и проверяется целесообразность его ветвления. В результате проверки может быть установлена бесперспективность ветвления.
- Пусть $t_1, \ldots, t_L$ множество не отсеченных подмножеств решений. (Первоначально $L = 1$, $t_1 = S$.)
- Множество t_i, $1 \le i \le L$ отсекается в одном из двух, последовательно проверяемых, случаев:
  1. если $U(t_i) \le f(x^0)$;
  2. если $U(\{t_i\}) = f(t_i) > f(x^0)$, т. е. одноэлементное множество отсекается всегда.
- Последнея проверка имеет место, так как одноэлементное множество может быть не удалено по критерию 1.
- Значит, в случае 2 происходит смена рекорда $x^0 = t_i$ и нижней границы $f(x^0)$.

Если $t_1, \ldots, t_M$, $M \le L$, множество еще не проверенных подмножеств, то:

- если $M = 0$, то x^0 — оптимальное решение задачи (1) и алгоритм останавливается;

Если $t_1, \dots, t_M$, $M \le L$, множество еще не проверенных подмножеств, то:

- если $M = 0$, то x^0 — оптимальное решение задачи (1) и алгоритм останавливается;
- если $M > 0$, то среди множеств $t_1, \dots, t_M$ выбираем перспективное подмножество, пусть это t_1, и осуществляем его ветвление. Получим подмножества $d_1, \dots, d_N$, $t_2, \dots, t_M$, $L = N + M - 1$. Перенумеруем эти подмножества числами $1, \dots, L$ и повторим шаг алгоритма.

- В приведенном выше описании метода упоминается выбор перспективного множества для дальнейшего ветвления. Выбор такого множества может осуществляться на основании различных стратегий и связан, как правило, с рассматриваемой задачей.

- В приведенном выше описании метода упоминается выбор перспективного множества для дальнейшего ветвления. Выбор такого множества может осуществляться на основании различных стратегий и связан, как правило, с рассматриваемой задачей.
- Если получение верхней границы сопряжено с трудностями, тогда для быстрого нахождения рекорда следует применять так называемую схему одностороннего ветвления, когда разбивается множество минимальной мощности. Таким способом одноэлементное множество и, возможно, допустимое решение (первый рекорд) будут найдены быстро.

- В приведенном выше описании метода упоминается выбор перспективного множества для дальнейшего ветвления. Выбор такого множества может осуществляться на основании различных стратегий и связан, как правило, с рассматриваемой задачей.
- Если получение верхней границы сопряжено с трудностями, тогда для быстрого нахождения рекорда следует применять так называемую схему одностороннего ветвления, когда разбивается множество минимальной мощности. Таким способом одноэлементное множество и, возможно, допустимое решение (первый рекорд) будут найдены быстро.
- С другой стороны, множество, имеющее максимальную верхнюю границу, может с большой вероятностью содержать решение, близкое (по функционалу) к оптимальному, что приведет к получению хорошего рекорда (и нижней границы). Выбор такого множества для дальнейшего разбиения определяет схему всестороннего ветвления.

Если при реализации алгоритма критической является память, тогда схема одностороннего ветвления, как правило, предпочтительнее.

Если при реализации алгоритма критической является память, тогда схема одностороннего ветвления, как правило, предпочтительнее.

Для решения методом ветвей и границ конкретной задачи следует определить:

Если при реализации алгоритма критической является память, тогда схема одностороннего ветвления, как правило, предпочтительнее.

Для решения методом ветвей и границ конкретной задачи следует определить:

- способ представления подмножеств решений;

Если при реализации алгоритма критической является память, тогда схема одностороннего ветвления, как правило, предпочтительнее.

Для решения методом ветвей и границ конкретной задачи следует определить:

- способ представления подмножеств решений;
- схему и способ ветвления;

Если при реализации алгоритма критической является память, тогда схема одностороннего ветвления, как правило, предпочтительнее.

Для решения методом ветвей и границ конкретной задачи следует определить:

- способ представления подмножеств решений;
- схему и способ ветвления;
- алгоритм вычисления верхней границы;

Если при реализации алгоритма критической является память, тогда схема одностороннего ветвления, как правило, предпочтительнее.

Для решения методом ветвей и границ конкретной задачи следует определить:

- способ представления подмножеств решений;
- схему и способ ветвления;
- алгоритм вычисления верхней границы;
- метод нахождения рекорда (если это возможно не только для одноэлементного множества).

- Очевидно, время работы алгоритма зависит от многих факторов.

- Очевидно, время работы алгоритма зависит от многих факторов.
- Теоретически, как отмечалось выше, не исключен полный перебор решений.

- Очевидно, время работы алгоритма зависит от многих факторов.
- Теоретически, как отмечалось выше, не исключен полный перебор решений.
- Практически же следует найти компромисс между точностью и сложностью вычисления верхней границы, что позволит найти решение, близкое к оптимальному, за приемлемое время.

- Очевидно, время работы алгоритма зависит от многих факторов.
- Теоретически, как отмечалось выше, не исключен полный перебор решений.
- Практически же следует найти компромисс между точностью и сложностью вычисления верхней границы, что позволит найти решение, близкое к оптимальному, за приемлемое время.
- Более точное вычисление верхней границы может позволить отсечь больше решений, но потребует и больше времени, что может привести к длительному выполнению одной итерации.

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.

Дерево поиска. Ветвление

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.
- Корень дерева поиска представляет исходную задачу.

Дерево поиска. Ветвление

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.
- Корень дерева поиска представляет исходную задачу.

Дерево поиска. Ветвление

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.
- Корень дерева поиска представляет исходную задачу.
- В ходе решения задачи дерево растет благодаря процессу, называемому ветвлением, который создает двух или более сыновей для одного из листьев текущего дерева поиска.

Дерево поиска. Ветвление

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.
- Корень дерева поиска представляет исходную задачу.
- В ходе решения задачи дерево растет благодаря процессу, называемому ветвлением, который создает двух или более сыновей для одного из листьев текущего дерева поиска.

- Превращение этих принципов в конкретный алгоритм для конкретной задачи оптимизации требует некоторой структуры данных, представляющей наборы возможных решений. Такое представление называется деревом поиска.
- Корень дерева поиска представляет исходную задачу.
- В ходе решения задачи дерево растет благодаря процессу, называемому ветвлением, который создает двух или более сыновей для одного из листьев текущего дерева поиска.
- Каждому узлу приписана подзадача I. Обозначим множество возможных решений подзадачи I через S_I.

Верхняя и нижняя границы

Дерево поиска должно сопровождаться тремя операциями:

- $branch(I)$ создает два или более узла дерева, каждый из которых представляет подмножество S_I. (Обычно подмножества не пересекаются, чтобы алгоритм не мог дважды обратиться к одному и тому же решению-кандидату, но это не требуется. Однако оптимальное решение среди S_I должно содержаться по крайней мере в одном из подмножеств.

Верхняя и нижняя границы

Дерево поиска должно сопровождаться тремя операциями:

- $branch(I)$ создает два или более узла дерева, каждый из которых представляет подмножество S_I. (Обычно подмножества не пересекаются, чтобы алгоритм не мог дважды обратиться к одному и тому же решению-кандидату, но это не требуется. Однако оптимальное решение среди S_I должно содержаться по крайней мере в одном из подмножеств.
- $bound(I)$ вычисляет верхнюю границу значения любого возможного решения в пространстве, представленном I, то есть $bound(I) \geq f(x)$ для всех x из S_I.

Верхняя и нижняя границы

Дерево поиска должно сопровождаться тремя операциями:

- $branch(I)$ создает два или более узла дерева, каждый из которых представляет подмножество S_I. (Обычно подмножества не пересекаются, чтобы алгоритм не мог дважды обратиться к одному и тому же решению-кандидату, но это не требуется. Однако оптимальное решение среди S_I должно содержаться по крайней мере в одном из подмножеств.
- $bound(I)$ вычисляет верхнюю границу значения любого возможного решения в пространстве, представленном I, то есть $bound(I) \geq f(x)$ для всех x из S_I.
- $solution(I)$ вычисляет и выдает оптимальное решение задачи I. (Это не всегда так, операция может выбрать и выдать некоторое допустимое решение из S_I). Если $solution(I)$ возвращает решение, тогда $f(solution(I))$ обеспечивает нижнюю границу для оптимального целевого значения во всем пространстве возможных решений.

- Используя эти операции, алгоритм B&B выполняет рекурсивный поиск сверху вниз по дереву поиска, сформированному операцией ветвления.

- Используя эти операции, алгоритм B&B выполняет рекурсивный поиск сверху вниз по дереву поиска, сформированному операцией ветвления.
- При посещении узла I он проверяет, меньше ли $bound(I)$, чем найденная на данный момент нижняя граница; в таком случае узел можно безопасно исключить из поиска, и рекурсия прекратится.

- Используя эти операции, алгоритм B&B выполняет рекурсивный поиск сверху вниз по дереву поиска, сформированному операцией ветвления.
- При посещении узла I он проверяет, меньше ли $bound(I)$, чем найденная на данный момент нижняя граница; в таком случае узел можно безопасно исключить из поиска, и рекурсия прекратится.
- Этот шаг сокращения обычно реализуется путем поддержания глобальной переменной, которая записывает максимальную нижнюю границу, наблюдаемую среди всех узлов, исследованных до сих пор.

- Приведем общую схему алгоритма ветвей и границ для максимизации произвольной целевой функции f.

- Приведем общую схему алгоритма ветвей и границ для максимизации произвольной целевой функции f.
- Чтобы получить из этой схемы фактический алгоритм, требуется функция bound, которая вычисляет верхние границы f на узлах дерева поиска, а также правило ветвления для конкретной задачи.

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.
3. Повторяйте, пока очередь не станет пустой:

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.
3. Повторяйте, пока очередь не станет пустой:
    1. Удалите узел N из очереди.

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.
3. Повторяйте, пока очередь не станет пустой:
   1. Удалите узел N из очереди.
   2. Если N представляет собой единственное возможное решение x и $f(x) > B$, то x — лучшее решение на данный момент. Запишите его и положите $B \leftarrow f(x)$.

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.
3. Повторяйте, пока очередь не станет пустой:
    1. Удалите узел N из очереди.
    2. Если N представляет собой единственное возможное решение x и $f(x) > B$, то x — лучшее решение на данный момент. Запишите его и положите $B \leftarrow f(x)$.
    3. В противном случае выполните ветвление узла N для создания новых узлов N_i. Для каждого из них:

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.
3. Повторяйте, пока очередь не станет пустой:
    1. Удалите узел N из очереди.
    2. Если N представляет собой единственное возможное решение x и $f(x) > B$, то x — лучшее решение на данный момент. Запишите его и положите $B \leftarrow f(x)$.
    3. В противном случае выполните ветвление узла N для создания новых узлов N_i. Для каждого из них:
        1. Если $bound(N_i) < B$, ничего не делать; поскольку верхняя граница этого узла меньше нижней границы задачи, она никогда не приведет к оптимальному решению и может быть отброшена.

1. Используя эвристику, найдите решение x_h задачи оптимизации. Сохраните значение целевой функции, $B = f(x_h)$. (Если эвристика недоступна, установите B в бесконечность.) B будет обозначать лучшее решение, найденное на данный момент, и будет использоваться в качестве нижней границы для возможных решений.
2. Инициализируйте очередь для хранения частичных решений; включите запись, в которой ни одной из переменных задачи не присвоено значение.
3. Повторяйте, пока очередь не станет пустой:
   1. Удалите узел N из очереди.
   2. Если N представляет собой единственное возможное решение x и $f(x) > B$, то x — лучшее решение на данный момент. Запишите его и положите $B \leftarrow f(x)$.
   3. В противном случае выполните ветвление узла N для создания новых узлов N_i. Для каждого из них:
      1. Если $bound(N_i) < B$, ничего не делать; поскольку верхняя граница этого узла меньше нижней границы задачи, она никогда не приведет к оптимальному решению и может быть отброшена.
      2. В противном случае сохраните N_i в очереди.

- Можно использовать несколько различных структур данных для организации очереди.

- Можно использовать несколько различных структур данных для организации очереди.
- Если используется очередь FIFO, то получаем поиск в ширину.

- Можно использовать несколько различных структур данных для организации очереди.
- Если используется очередь FIFO, то получаем поиск в ширину.
- Стек (очередь LIFO) дает алгоритм поиска в глубину (схема одностороннего ветвления).

- Можно использовать несколько различных структур данных для организации очереди.
- Если используется очередь FIFO, то получаем поиск в ширину.
- Стек (очередь LIFO) дает алгоритм поиска в глубину (схема одностороннего ветвления).
- Алгоритм, в котором исследуется узел с наилудшей границей, может быть получен с помощью очереди с приоритетом, которая сортирует узлы по их верхней границе (схема всестороннего ветвления).

- Можно использовать несколько различных структур данных для организации очереди.
- Если используется очередь FIFO, то получаем поиск в ширину.
- Стек (очередь LIFO) дает алгоритм поиска в глубину (схема одностороннего ветвления).
- Алгоритм, в котором исследуется узел с наилудшей границей, может быть получен с помощью очереди с приоритетом, которая сортирует узлы по их верхней границе (схема всестороннего ветвления).
- Вариант поиска в глубину рекомендуется, когда нет хорошей эвристики для получения начального решения, потому что он быстро дает полные решения и, следовательно, нижние границы.

- Метод B&B также может быть базой для различных эвристик.

- Метод B&B также может быть базой для различных эвристик.
- Например, кто-то может пожелать прекратить ветвление, когда разрыв между верхней и нижней границами становится меньше определенного порога.

- Метод B&B также может быть базой для различных эвристик.
- Например, кто-то может пожелать прекратить ветвление, когда разрыв между верхней и нижней границами становится меньше определенного порога.
- Это используется, когда решение «достаточно хорошо для практических целей» и может значительно сократить требуемые вычисления.

- Метод B&B также может быть базой для различных эвристик.
- Например, кто-то может пожелать прекратить ветвление, когда разрыв между верхней и нижней границами становится меньше определенного порога.
- Это используется, когда решение «достаточно хорошо для практических целей» и может значительно сократить требуемые вычисления.
- Этот тип решения особенно применим, когда используемая функция стоимости является зашумленной или является результатом статистических оценок и поэтому не известна точно, а скорее всего лишь известно, что она находится в диапазоне значений с определенной вероятностью.

Задача о 0/1 рюкзаке

Задача о рюкзаке

Есть несколько разных задач о рюкзаке. Первая и классическая задача — это задача о 0/1 рюкзаке. Это следующая история. Турист планирует поездку в горы. У него много вещей, которые могут пригодиться во время экскурсии. Мы предполагаем, что выполняются следующие условия.

Есть несколько разных задач о рюкзаке. Первая и классическая задача — это задача о 0/1 рюкзаке. Это следующая история. Турист планирует поездку в горы. У него много вещей, которые могут пригодиться во время экскурсии. Мы предполагаем, что выполняются следующие условия.

- Каждый предмет имеет положительное значение и положительный вес. (Значение — это степень вклада объекта в успех тура).

Есть несколько разных задач о рюкзаке. Первая и классическая задача — это задача о 0/1 рюкзаке. Это следующая история. Турист планирует поездку в горы. У него много вещей, которые могут пригодиться во время экскурсии. Мы предполагаем, что выполняются следующие условия.

- Каждый предмет имеет положительное значение и положительный вес. (Значение — это степень вклада объекта в успех тура).
- Объекты независимы друг от друга.

Есть несколько разных задач о рюкзаке. Первая и классическая задача — это задача о 0/1 рюкзаке. Это следующая история. Турист планирует поездку в горы. У него много вещей, которые могут пригодиться во время экскурсии. Мы предполагаем, что выполняются следующие условия.

- Каждый предмет имеет положительное значение и положительный вес. (Значение — это степень вклада объекта в успех тура).
- Объекты независимы друг от друга.
- Рюкзак туриста достаточно большой.

Есть несколько разных задач о рюкзаке. Первая и классическая задача — это задача о 0/1 рюкзаке. Это следующая история. Турист планирует поездку в горы. У него много вещей, которые могут пригодиться во время экскурсии. Мы предполагаем, что выполняются следующие условия.

- Каждый предмет имеет положительное значение и положительный вес. (Значение — это степень вклада объекта в успех тура).
- Объекты независимы друг от друга.
- Рюкзак туриста достаточно большой.
- Сила туриста позволяет взять с собой лишь ограниченный общий вес.

Задача о рюкзаке

Есть несколько разных задач о рюкзаке. Первая и классическая задача — это задача о 0/1 рюкзаке. Это следующая история. Турист планирует поездку в горы. У него много вещей, которые могут пригодиться во время экскурсии. Мы предполагаем, что выполняются следующие условия.

- Каждый предмет имеет положительное значение и положительный вес. (Значение — это степень вклада объекта в успех тура).
- Объекты независимы друг от друга.
- Рюкзак туриста достаточно большой.
- Сила туриста позволяет взять с собой лишь ограниченный общий вес.
- Но в рамках этого ограничения по весу турист хочет достичь максимальной общей стоимости.

При математической постановке задачи используются следующие обозначения:

n	количество предметов;
j	индекс предмета;
w_j	вес предмета j;
v_j	ценность предмета j;
b	максимальный вес, который может нести турист.

Для каждого объекта j вводится так называемая *двоичная* или *ноль-один* переменная, скажем, x_j:

$$x_j = \begin{cases} 1 & \text{если предмет } j \text{ берется в поход} \\ 0 & \text{если предмет } j \text{ не берется в поход.} \end{cases}$$

Для каждого объекта j вводится так называемая *двоичная* или *ноль-один* переменная, скажем, x_j:

$$x_j = \begin{cases} 1 & \text{если предмет } j \text{ берется в поход} \\ 0 & \text{если предмет } j \text{ не берется в поход.} \end{cases}$$

Отметим, что

$$w_j x_j = \begin{cases} w_j & \text{если предмет } j \text{ берется в поход,} \\ 0 & \text{если предмет } j \text{ не берется в поход} \end{cases}$$

— вес премета взятого в рюкзак.

- Аналогично $v_j x_j$ — это стоимость объекта в туре. Общий вес ранца

$$\sum_{j=1}^{n} w_j x_j$$

который не может превышать предельный вес.

- Аналогично $v_j x_j$ — это стоимость объекта в туре. Общий вес ранца

$$\sum_{j=1}^{n} w_j x_j$$

который не может превышать предельный вес.

- Следовательно, математическая форма задачи такова:

$$\max \sum_{j=1}^{n} v_j x_j \tag{1}$$

$$\sum_{j=1}^{n} w_j x_j \leq b \tag{2}$$

$$x_j = 0 \text{ или } 1, \quad j = 1, \ldots, n\ . \tag{3}$$

- Сложность задачи вызвана требованием целочисленности. Если ограничение (3) заменить ослабленным ограничением:

$$0 \le x_j \le 1, \quad j = 1, \ldots, n \;, \tag{4}$$

то задача (1), (2), и (4) является задачей ЛП.

Теорема 1

Предположим, что все числа v_j, w_j $(j = 1, \ldots, n)$ положительны и, кроме того, они упорядочены так, что

$$\frac{v_1}{w_1} \ge \frac{v_2}{w_2} \cdots \ge \frac{v_n}{w_n}. \tag{5}$$

Тогда существует индекс p $(1 \le p \le n)$ и оптимальное решение $\mathbf{x}^*$ такие, что

$$x_1^* = x_2^* = \cdots = x_{p-1}^* = 1, \; x_{p+1}^* = x_{p+2}^* = \cdots = x_{p+1}^* = 0.$$

- Сложность задачи вызвана требованием целочисленности. Если ограничение (3) заменить ослабленным ограничением:

$$0 \leq x_j \leq 1, \quad j = 1, \ldots, n \,, \tag{4}$$

то задача (1), (2), и (4) является задачей ЛП.

- (4) означает, что в рюкзаке может находиться не только целый предмет, но и любая его часть.

Теорема 1

Предположим, что все числа v_j, w_j $(j = 1, \ldots, n)$ положительны и, кроме того, они упорядочены так, что

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \cdots \geq \frac{v_n}{w_n}. \tag{5}$$

Тогда существует индекс p $(1 \leq p \leq n)$ и оптимальное решение $\mathbf{x}^*$ такие, что

$$x_1^* = x_2^* = \cdots = x_{p-1}^* = 1, \; x_{p+1}^* = x_{p+2}^* = \cdots = x_{p+1}^* = 0.$$

- Сложность задачи вызвана требованием целочисленности. Если ограничение (3) заменить ослабленным ограничением:

$$0 \leq x_j \leq 1, \quad j = 1, \dots, n\ , \tag{4}$$

то задача (1), (2), и (4) является задачей ЛП.

- (4) означает, что в рюкзаке может находиться не только целый предмет, но и любая его часть.
- Более того, оптимальное решение легко найти.

Теорема 1

Предположим, что все числа v_j, w_j $(j = 1, \dots, n)$ положительны и, кроме того, они упорядочены так, что

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \cdots \geq \frac{v_n}{w_n}. \tag{5}$$

Тогда существует индекс p $(1 \leq p \leq n)$ и оптимальное решение $\mathbf{x}^*$ такие, что

$$x_1^* = x_2^* = \cdots = x_{p-1}^* = 1,\ x_{p+1}^* = x_{p+2}^* = \cdots = x_{p+1}^* = 0.$$

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.
- Любое возможное решение первого варианта возможно и во втором.

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.
- Любое возможное решение первого варианта возможно и во втором.
- Но противоположное утверждение неверно.

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.
- Любое возможное решение первого варианта возможно и во втором.
- Но противоположное утверждение неверно.
- Другими словами, множество возможных решений первой проблемы является собственным подмножеством возможных решений второй. Этот факт имеет два важных следствия:

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.
- Любое возможное решение первого варианта возможно и во втором.
- Но противоположное утверждение неверно.
- Другими словами, множество возможных решений первой проблемы является собственным подмножеством возможных решений второй. Этот факт имеет два важных следствия:
  - Оптимальное значение задачи (1), (2), и (4) является верхней границей оптимального значения задачи (1), (2), и (3).

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.
- Любое возможное решение первого варианта возможно и во втором.
- Но противоположное утверждение неверно.
- Другими словами, множество возможных решений первой проблемы является собственным подмножеством возможных решений второй. Этот факт имеет два важных следствия:
    - Оптимальное значение задачи (1), (2), и (4) является верхней границей оптимального значения задачи (1), (2), и (3).
    - Если оптимальное решение задачи (1), (2), и (4) является допустимым для задачи (1), (2), и (3) то это оптимальное решение и последней задачи.

- Обратите внимание, что в $\mathbf{x}^*$ есть не более одного нецелочисленного компонента. Это свойство будет использовано в алгоритме B&B.
- С точки зрения B&B связь зачачи (1), (2), (3) и (1), (2) и (4) очень важна.
- Любое возможное решение первого варианта возможно и во втором.
- Но противоположное утверждение неверно.
- Другими словами, множество возможных решений первой проблемы является собственным подмножеством возможных решений второй. Этот факт имеет два важных следствия:
  - Оптимальное значение задачи (1), (2), и (4) является верхней границей оптимального значения задачи (1), (2), и (3).
  - Если оптимальное решение задачи (1), (2), и (4) является допустимым для задачи (1), (2), и (3) то это оптимальное решение и последней задачи.
- Эти свойства используются в методе B&B.

Чтобы показать логику B&B рассмотрим пример

$$
\begin{array}{lcccccccccccc}
\max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\
 & 8x_1 & + & 7x_2 & + & 11x_3 & + & 6x_4 & + & 19x_5 & \leq & 25 \\
 & & & & x_1, x_2, x_3, x_4, x_5 = 0 \text{ or } 1 & & & & & & &
\end{array} \qquad (6)
$$

Обратите внимание, что условие

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \cdots \geq \frac{v_n}{w_n}.$$

выполняется так как

$$\frac{23}{8} = 2.875 > \frac{19}{7} \approx 2.714 > \frac{28}{11} \approx 2.545 > \frac{14}{6} \approx 2.333 > \frac{44}{19} \approx 2.316.$$

Обратите внимание, что условие

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \cdots \geq \frac{v_n}{w_n}.$$

выполняется так как

$$\frac{23}{8} = 2.875 > \frac{19}{7} \approx 2.714 > \frac{28}{11} \approx 2.545 > \frac{14}{6} \approx 2.333 > \frac{44}{19} \approx 2.316.$$

Множество допустимых решений нашей задачи обозначим через $\mathcal{F}$, т.е.

$$\mathcal{F} = \{\mathbf{x} \mid 8x_1+7x_2+11x_3+6x_4+19x_5 \leq 25;\ x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1\}.$$

Обратите внимание, что условие

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \cdots \geq \frac{v_n}{w_n}.$$

выполняется так как

$$\frac{23}{8} = 2.875 > \frac{19}{7} \approx 2.714 > \frac{28}{11} \approx 2.545 > \frac{14}{6} \approx 2.333 > \frac{44}{19} \approx 2.316.$$

Множество допустимых решений нашей задачи обозначим через $\mathcal{F}$, т.е.

$$\mathcal{F} = \{\mathbf{x} \mid 8x_1+7x_2+11x_3+6x_4+19x_5 \leq 25;\ x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1\}.$$

Релаксация задачи (6):

$$\begin{array}{lrcrcrcrcrcl} \max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\ & 8x_1 & + & 7x_2 & + & 11x_3 & + & 6x_4 & + & 19x_5 & \leq & 25 \\ \multicolumn{12}{c}{0 \leq x_1, x_2, x_3, x_4, x_5 \leq 1.} \end{array} \tag{7}$$

- Множество допустимых решений (7) обозначим через $\mathcal{R}$, т.е.

$$\mathcal{R} = \{\mathbf{x} \mid 8x_1+7x_2+11x_3+6x_4+19x_5 \le 25;\ 0 \le x_1, x_2, x_3, x_4, x_5 \le 1\}.$$

- Множество допустимых решений (7) обозначим через $\mathcal{R}$, т.е.

  $$\mathcal{R} = \{\mathbf{x} \mid 8x_1{+}7x_2{+}11x_3{+}6x_4{+}19x_5 \leq 25;\ 0 \leq x_1, x_2, x_3, x_4, x_5 \leq 1\}.$$

- Таким образом, разница между (6) и (7) заключается в том, что значение переменных должно быть либо 0, либо 1 в (6), а с другой стороны, они могут принимать любое значение из отрезка $[0, 1]$ в случае (7).

- Множество допустимых решений (7) обозначим через $\mathcal{R}$, т.е.

  $\mathcal{R} = \{\mathbf{x} \mid 8x_1{+}7x_2{+}11x_3{+}6x_4{+}19x_5 \le 25;\ 0 \le x_1, x_2, x_3, x_4, x_5 \le 1\}.$

- Таким образом, разница между (6) и (7) заключается в том, что значение переменных должно быть либо 0, либо 1 в (6), а с другой стороны, они могут принимать любое значение из отрезка $[0, 1]$ в случае (7).
- Поскольку задача (6) сложна, вместо нее решается (7). Оптимальным решением является

$$x_1^* = x_2^* = 1, x_3^* = \frac{10}{11}, x_4^* = x_5^* = 0.$$

- Множество допустимых решений (7) обозначим через $\mathcal{R}$, т.е.

  $\mathcal{R} = \{\mathbf{x} \mid 8x_1+7x_2+11x_3+6x_4+19x_5 \le 25;\ 0 \le x_1, x_2, x_3, x_4, x_5 \le 1\}.$

- Таким образом, разница между (6) и (7) заключается в том, что значение переменных должно быть либо 0, либо 1 в (6), а с другой стороны, они могут принимать любое значение из отрезка $[0, 1]$ в случае (7).
- Поскольку задача (6) сложна, вместо нее решается (7). Оптимальным решением является

$$x_1^* = x_2^* = 1, x_3^* = \frac{10}{11}, x_4^* = x_5^* = 0.$$

- Поскольку значение x_3^* не является целым числом, оптимальное значение 67,54 является лишь верхней границей оптимального значения (6), и необходим дальнейший анализ.

- Множество допустимых решений (7) обозначим через $\mathcal{R}$, т.е.

  $$\mathcal{R} = \{\mathbf{x} \mid 8x_1+7x_2+11x_3+6x_4+19x_5 \le 25;\ 0 \le x_1, x_2, x_3, x_4, x_5 \le 1\}.$$

- Таким образом, разница между (6) и (7) заключается в том, что значение переменных должно быть либо 0, либо 1 в (6), а с другой стороны, они могут принимать любое значение из отрезка $[0, 1]$ в случае (7).
- Поскольку задача (6) сложна, вместо нее решается (7). Оптимальным решением является

  $$x_1^* = x_2^* = 1, x_3^* = \frac{10}{11}, x_4^* = x_5^* = 0.$$

- Поскольку значение x_3^* не является целым числом, оптимальное значение 67,54 является лишь верхней границей оптимального значения (6), и необходим дальнейший анализ.
- Значение 67,54 можно округлить до 67 из-за целочисленности коэффициентов целевой функции.

- Ключевая идея состоит в том, что множества допустимых решений обеих проблем разделены на две части в соответствии с двумя возможными значениями x_3.

- Ключевая идея состоит в том, что множества допустимых решений обеих проблем разделены на две части в соответствии с двумя возможными значениями x_3.
- Переменная x_3 выбрана, так как ее значение не целое. Важность выбора обсуждается ниже.

- Ключевая идея состоит в том, что множества допустимых решений обеих проблем разделены на две части в соответствии с двумя возможными значениями x_3.
- Переменная x_3 выбрана, так как ее значение не целое. Важность выбора обсуждается ниже.
- Пусть

$$\mathcal{F}_0 = \mathcal{F},\ \mathcal{F}_1 = \mathcal{F}_0 \cap \{\mathbf{x} \mid x_3 = 0\},\ \mathcal{F}_2 = \mathcal{F}_0 \cap \{\mathbf{x} \mid x_3 = 1\}$$

и

$$\mathcal{R}_0 = \mathcal{R},\ \mathcal{R}_1 = \mathcal{R}_0 \cap \{\mathbf{x} \mid x_3 = 0\},\ \mathcal{R}_2 = \mathcal{R}_0 \cap \{\mathbf{x} \mid x_3 = 1\}.$$

- Ключевая идея состоит в том, что множества допустимых решений обеих проблем разделены на две части в соответствии с двумя возможными значениями x_3.
- Переменная x_3 выбрана, так как ее значение не целое. Важность выбора обсуждается ниже.
- Пусть

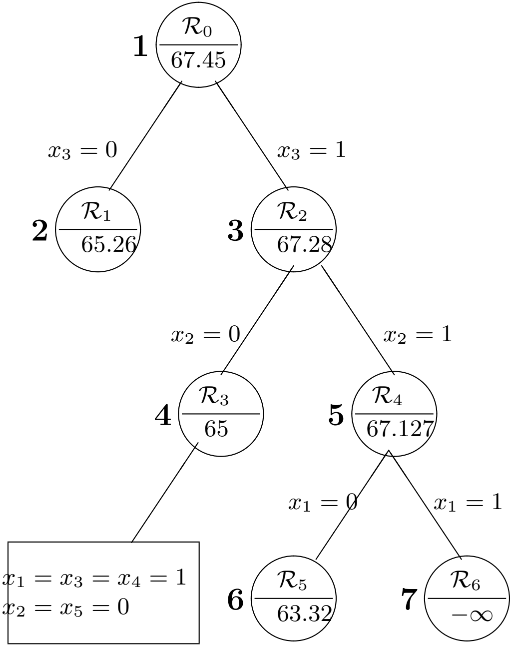
$$\mathcal{F}_0 = \mathcal{F},\ \mathcal{F}_1 = \mathcal{F}_0 \cap \{\mathbf{x} \mid x_3 = 0\},\ \mathcal{F}_2 = \mathcal{F}_0 \cap \{\mathbf{x} \mid x_3 = 1\}$$

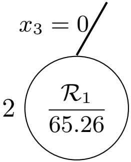
и

$$\mathcal{R}_0 = \mathcal{R},\ \mathcal{R}_1 = \mathcal{R}_0 \cap \{\mathbf{x} \mid x_3 = 0\},\ \mathcal{R}_2 = \mathcal{R}_0 \cap \{\mathbf{x} \mid x_3 = 1\}.$$

- Очевидно

$$\mathcal{F}_1 \subseteq \mathcal{R}_1 \text{ и } \mathcal{F}_2 \subseteq \mathcal{R}_2.$$

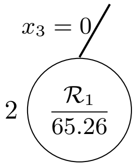

1
$\mathcal{R}_0$
67.45
$x_3 = 0$
$x_3 = 1$
2
$\mathcal{R}_1$
65.26
3
$\mathcal{R}_2$
67.28
$x_2 = 0$
$x_2 = 1$
4
$\mathcal{R}_3$
65
5
$\mathcal{R}_4$
67.127
$x_1 = 0$
$x_1 = 1$
$x_1 = x_3 = x_4 = 1$
$x_2 = x_5 = 0$
6
$\mathcal{R}_5$
63.32
7
$\mathcal{R}_6$
$-\infty$



$$
\begin{aligned}
&\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5) \\
&\mathbf{x} \in \mathcal{R}_1
\end{aligned}
\tag{8}
$$

Следовательно задача является релаксацией задачи

$$
\begin{aligned}
&\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5) \\
&\mathbf{x} \in \mathcal{F}_1.
\end{aligned}
\tag{9}
$$

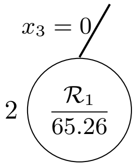


$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_1 \tag{8}$$

Следовательно задача является релаксацией задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{F}_1. \tag{9}$$

- Задача (8) может быть решена по теореме 1, также, но необходимо учитывать, что значение x_3 равно 0.



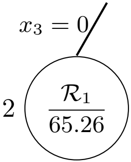
$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_1 \tag{8}$$

Следовательно задача является релаксацией задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{F}_1. \tag{9}$$

- Задача (8) может быть решена по теореме 1, также, но необходимо учитывать, что значение x_3 равно 0.
- Таким образом, оптимальным решением является

$$x_1^* = x_2^* = 1,\ x_3^* = 0,\ x_4^* = 1,\ x_5^* = \frac{4}{19}.$$



$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_1 \qquad (8)$$

Следовательно задача является релаксацией задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{F}_1. \qquad (9)$$

- Задача (8) может быть решена по теореме 1, также, но необходимо учитывать, что значение x_3 равно 0.
- Таким образом, оптимальным решением является

$$x_1^* = x_2^* = 1,\ x_3^* = 0,\ x_4^* = 1,\ x_5^* = \frac{4}{19}.$$

- Оптимальное значение — 65,26, что дает верхнюю границу 65 для оптимального значения задачи (9).

Исследуются остальные подмножества возможных

решений.

Исследуются остальные подмножества возможных решений.

$x_3 = 1$

3 $\dfrac{\mathcal{R}_2}{67.28}$

Оптимальное решение задачи

$$\begin{aligned} \max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5) \\ \mathbf{x} \in \mathcal{R}_2 \end{aligned} \qquad (10)$$

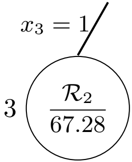
есть

$$x_1^* = 1,\ x_2^* = \frac{6}{7},\ x_3^* = 1,\ x_4^* = x_5^* = 0$$

значение критерия 67.28. Следовательно, 67 это верхняя граница задачи

$$\begin{aligned} \max 23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5 \\ \mathbf{x} \in \mathcal{F}_2 \end{aligned} \qquad (11)$$

Исследуются остальные подмножества возможных решений.



Оптимальное решение задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5) \qquad \mathbf{x} \in \mathcal{R}_2 \tag{10}$$

есть

$$x_1^* = 1,\ x_2^* = \frac{6}{7},\ x_3^* = 1,\ x_4^* = x_5^* = 0$$

Поскольку значение критерия 67.28. Следовательно, 67 это верхняя граница задачи

$$\max 23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5 \qquad \mathbf{x} \in \mathcal{F}_2 \tag{11}$$

верхняя граница (11) выше, чем верхняя граница (9), (67>65) т.е. эта ветвь более перспективна, сначала рассматривается она.

- Снова делим на две ветви в соответствии с двумя значениями x_2, поскольку это нецелая переменная в оптимальное решение (10).

- Снова делим на две ветви в соответствии с двумя значениями x_2, поскольку это нецелая переменная в оптимальное решение (10).
- Пусть

$$
\begin{aligned}
\mathcal{F}_3 &= \mathcal{F}_2 \cap \{\mathbf{x} \mid x_2 = 0\}\,, \\
\mathcal{F}_4 &= \mathcal{F}_2 \cap \{\mathbf{x} \mid x_2 = 1\}\,, \\
\mathcal{R}_3 &= \mathcal{R}_2 \cap \{\mathbf{x} \mid x_2 = 0\}\,, \\
\mathcal{R}_4 &= \mathcal{R}_2 \cap \{\mathbf{x} \mid x_2 = 1\}\,.
\end{aligned}
$$

- Снова делим на две ветви в соответствии с двумя значениями x_2, поскольку это нецелая переменная в оптимальное решение (10).
- Пусть

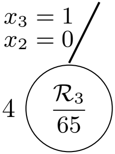
$$\begin{aligned}
\mathcal{F}_3 &= \mathcal{F}_2 \cap \{\mathbf{x} \mid x_2 = 0\} \,, \\
\mathcal{F}_4 &= \mathcal{F}_2 \cap \{\mathbf{x} \mid x_2 = 1\} \,, \\
\mathcal{R}_3 &= \mathcal{R}_2 \cap \{\mathbf{x} \mid x_2 = 0\} \,, \\
\mathcal{R}_4 &= \mathcal{R}_2 \cap \{\mathbf{x} \mid x_2 = 1\} \,.
\end{aligned}$$

- Множества $\mathcal{F}_3$ и $\mathcal{R}_3$ содержат допустимое решение исходных задач, так что x_3 фиксируем равным 1, а x_2 фиксируем равным 0.

- Снова делим на две ветви в соответствии с двумя значениями x_2, поскольку это нецелая переменная в оптимальное решение (10).
- Пусть

$$
\begin{aligned}
\mathcal{F}_3 &= \mathcal{F}_2 \cap \{\mathbf{x} \mid x_2 = 0\} ,\\
\mathcal{F}_4 &= \mathcal{F}_2 \cap \{\mathbf{x} \mid x_2 = 1\} ,\\
\mathcal{R}_3 &= \mathcal{R}_2 \cap \{\mathbf{x} \mid x_2 = 0\} ,\\
\mathcal{R}_4 &= \mathcal{R}_2 \cap \{\mathbf{x} \mid x_2 = 1\} .
\end{aligned}
$$

- Множества $\mathcal{F}_3$ и $\mathcal{R}_3$ содержат допустимое решение исходных задач, так что x_3 фиксируем равным 1, а x_2 фиксируем равным 0.
- В множествах $\mathcal{F}_4$ и $\mathcal{R}_4$ обе переменные фиксируем 1.

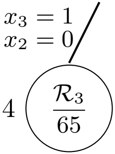


Оптимальное решение релаксированной задачи:

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_3$$

это

$$x_1^* = 1,\ x_2^* = 0,\ x_3^* = 1,\ x_4^* = 1,\ x_5^* = 0.$$



Оптимальное решение релаксированной задачи:

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_3$$

это

$$x_1^* = 1,\ x_2^* = 0,\ x_3^* = 1,\ x_4^* = 1,\ x_5^* = 0.$$

- Поскольку это решение состоит из целых чисел, то это также оптимальное решение рассматриваемой задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{F}_3.$$

$x_3 = 1$
$x_2 = 0$

4 $\dfrac{\mathcal{R}_3}{65}$

Оптимальное решение релаксированной задачи:

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_3$$

это

$$x_1^* = 1,\ x_2^* = 0,\ x_3^* = 1,\ x_4^* = 1,\ x_5^* = 0.$$

- Поскольку это решение состоит из целых чисел, то это также оптимальное решение рассматриваемой задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{F}_3.$$

- Оптимальное значение целевой функции — 65.

$x_3 = 1$
$x_2 = 0$

4 $\frac{\mathcal{R}_3}{65}$

Оптимальное решение релаксированной задачи:

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{R}_3$$

это

$$x_1^* = 1,\ x_2^* = 0,\ x_3^* = 1,\ x_4^* = 1,\ x_5^* = 0.$$

- Поскольку это решение состоит из целых чисел, то это также оптимальное решение рассматриваемой задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5)$$
$$\mathbf{x} \in \mathcal{F}_3.$$

- Оптимальное значение целевой функции — 65.
- Ветвь множеств $\mathcal{F}_3$ и $\mathcal{R}_3$ полностью изучена, т.е. невозможно найти лучшее решение в ней.

- Другая новая ветвь — это когда и x_2, и x_3 имеют фиксированное значение 1.

- Другая новая ветвь — это когда и x_2, и x_3 имеют фиксированное значение 1.
- Если целевая функция оптимизирована на $\mathcal{R}_4$, то оптимальным решением будет

$$x_1^* = \frac{7}{8}, \, x_2^* = x_3^* = 1, \, x_4^* = x_5^* = 0.$$

- Другая новая ветвь — это когда и x_2, и x_3 имеют фиксированное значение 1.
- Если целевая функция оптимизирована на $\mathcal{R}_4$, то оптимальным решением будет

$$x_1^* = \frac{7}{8},\ x_2^* = x_3^* = 1,\ x_4^* = x_5^* = 0.$$

- Применим ту же технику снова, две новые ветви определяются множествами

$$\mathcal{F}_5 = \mathcal{F}_4 \cap \{\mathbf{x} \mid x_1 = 0\},\ \mathcal{F}_6 = \mathcal{F}_4 \cap \{\mathbf{x} \mid x_1 = 1\},$$

$$\mathcal{R}_5 = \mathcal{R}_4 \cap \{\mathbf{x} \mid x_2 = 0\},\ \mathcal{R}_6 = \mathcal{R}_4 \cap \{\mathbf{x} \mid x_2 = 1\}.$$

- Оптимальное решение ветви $\mathcal{R}_5$ есть

$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$

- Оптимальное решение ветви $\mathcal{R}_5$ есть
$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$
- Оптимальное значение целевой функции — 63.32.

- Оптимальное решение ветви $\mathcal{R}_5$ есть

$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$

- Оптимальное значение целевой функции — 63.32.
- Это строго меньше значения целевой функции допустимого решения, найденного в ветви $\mathcal{R}_3$.

- Оптимальное решение ветви $\mathcal{R}_5$ есть

$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$

- Оптимальное значение целевой функции — 63.32.
- Это строго меньше значения целевой функции допустимого решения, найденного в ветви $\mathcal{R}_3$.
- Следовательно, она не может содержать оптимального решения.

- Оптимальное решение ветви $\mathcal{R}_5$ есть

$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$

- Оптимальное значение целевой функции — 63.32.
- Это строго меньше значения целевой функции допустимого решения, найденного в ветви $\mathcal{R}_3$.
- Следовательно, она не может содержать оптимального решения.
- Таким образом, его дальнейшее исследование можно не проводить, хотя наилучшее возможное решение в этой ветви останется неизвестным.

- Оптимальное решение ветви $\mathcal{R}_5$ есть

$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$

- Оптимальное значение целевой функции — 63.32.
- Это строго меньше значения целевой функции допустимого решения, найденного в ветви $\mathcal{R}_3$.
- Следовательно, она не может содержать оптимального решения.
- Таким образом, его дальнейшее исследование можно не проводить, хотя наилучшее возможное решение в этой ветви останется неизвестным.
- Ветвь $\mathcal{R}_6$ недопустима, поскольку объекты 1, 2 и 3 переполняют рюкзак.

- Оптимальное решение ветви $\mathcal{R}_5$ есть

$$x_1^* = 0,\ x_2^* = x_3^* = x_4^* = 1,\ x_5^* = \frac{1}{19}.$$

- Оптимальное значение целевой функции — 63.32.
- Это строго меньше значения целевой функции допустимого решения, найденного в ветви $\mathcal{R}_3$.
- Следовательно, она не может содержать оптимального решения.
- Таким образом, его дальнейшее исследование можно не проводить, хотя наилучшее возможное решение в этой ветви останется неизвестным.
- Ветвь $\mathcal{R}_6$ недопустима, поскольку объекты 1, 2 и 3 переполняют рюкзак.
- Традиционно этот факт обозначается использованием $-\infty$ в качестве оптимального значения целевой функции.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.
- Сразу можно сделать вывод, что это допустимое решение является оптимальным.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.
- Сразу можно сделать вывод, что это допустимое решение является оптимальным.
- Если нет необходимости в альтернативных оптимальных решениях, то исследование этой последней ветви может быть прекращено и метод завершен.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.
- Сразу можно сделать вывод, что это допустимое решение является оптимальным.
- Если нет необходимости в альтернативных оптимальных решениях, то исследование этой последней ветви может быть прекращено и метод завершен.
- Если требуются альтернативные оптимальные решения, исследования необходимо продолжить.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.
- Сразу можно сделать вывод, что это допустимое решение является оптимальным.
- Если нет необходимости в альтернативных оптимальных решениях, то исследование этой последней ветви может быть прекращено и метод завершен.
- Если требуются альтернативные оптимальные решения, исследования необходимо продолжить.
- Нецелочисленная переменная в оптимальном решении ветви это x_5.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.
- Сразу можно сделать вывод, что это допустимое решение является оптимальным.
- Если нет необходимости в альтернативных оптимальных решениях, то исследование этой последней ветви может быть прекращено и метод завершен.
- Если требуются альтернативные оптимальные решения, исследования необходимо продолжить.
- Нецелочисленная переменная в оптимальном решении ветви это x_5.
- Подветви, называемые позже 7-й и 8-й ветвями, определяются уравнениями $x_5 = 0$ и $x_5 = 1$, дают верхние границы 56 и 61 соответственно.

- На данный момент есть только одна ветвь, которая еще не исследована. Это ветвь $\mathcal{R}_1$.
- Верхняя граница здесь равна 65, что равно значению целевой функции найденного допустимого решения.
- Сразу можно сделать вывод, что это допустимое решение является оптимальным.
- Если нет необходимости в альтернативных оптимальных решениях, то исследование этой последней ветви может быть прекращено и метод завершен.
- Если требуются альтернативные оптимальные решения, исследования необходимо продолжить.
- Нецелочисленная переменная в оптимальном решении ветви это x_5.
- Подветви, называемые позже 7-й и 8-й ветвями, определяются уравнениями $x_5 = 0$ и $x_5 = 1$, дают верхние границы 56 и 61 соответственно.
- Таким образом, они не содержат оптимального решения, и метод завершен.

Анализ порядка вычислений в примере

- Проанализируем порядк вычислений в примере, чтобы подчеркнуть общую логику, лежащую в основе метода.

Анализ порядка вычислений в примере

- Проанализируем порядк вычислений в примере, чтобы подчеркнуть общую логику, лежащую в основе метода.
- Выявленные свойства будут использоваться в следующем разделе, когда обсуждается общий метод B&B.

Анализ порядка вычислений в примере

- Проанализируем порядк вычислений в примере, чтобы подчеркнуть общую логику, лежащую в основе метода.
- Выявленные свойства будут использоваться в следующем разделе, когда обсуждается общий метод B&B.
- Задача о рюкзаке NP-трудная, а задача ЛП эффективно решается. Поэтому очень похожая, но гораздо более простая задача ЛП

$$\begin{array}{lcccccccccccc} \max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\ & 8x_1 & + & 7x_2 & + & 11x_3 & + & 6x_4 & + & 19x_5 & \leq & 25 \end{array}$$
$$0 \leq x_1, x_2, x_3, x_4, x_5 \leq 1. \tag{7}$$

была решена вместо задачи о ранце

$$\begin{array}{lcccccccccccc} \max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\ & 8x_1 & + & 7x_2 & + & 11x_3 & + & 6x_4 & + & 19x_5 & \leq & 25 \end{array}$$
$$x_1, x_2, x_3, x_4, x_5 = 0 \text{ or } 1 \tag{6}$$

- Априори нельзя было исключить случай, когда оптимальное решение задачи (7) также является оптимальным решением задачи (6). В итоге оказалось, что оптимальное решение задачи (7) не удовлетворяет всем ограничениям задачи (6), поэтому оно не решение этой задачи.

- Априори нельзя было исключить случай, когда оптимальное решение задачи (7) также является оптимальным решением задачи (6). В итоге оказалось, что оптимальное решение задачи (7) не удовлетворяет всем ограничениям задачи (6), поэтому оно не решение этой задачи.
- Но расчет не оказался бесполезным, поскольку была получена верхняя граница оптимального значения для задачи (6).

- Априори нельзя было исключить случай, когда оптимальное решение задачи (7) также является оптимальным решением задачи (6). В итоге оказалось, что оптимальное решение задачи (7) не удовлетворяет всем ограничениям задачи (6), поэтому оно не решение этой задачи.
- Но расчет не оказался бесполезным, поскольку была получена верхняя граница оптимального значения для задачи (6).
- Эти свойства отражены в определении *релаксации* в следующем разделе.

- Поскольку релаксация не решила задачу (6), она была разделена на подзадачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_1 \tag{9}$$

и

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_2 \tag{11}$$

- Поскольку релаксация не решила задачу (6), она была разделена на подзадачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_1 \tag{9}$$

и

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_2 \tag{11}$$

- Обе подзадачи имеют собственное оптимальное решение, и лучшее из них — оптимальное решение задачи (6).

- Поскольку релаксация не решила задачу (6), она была разделена на подзадачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_1 \tag{9}$$

и

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_2 \tag{11}$$

- Обе подзадачи имеют собственное оптимальное решение, и лучшее из них — оптимальное решение задачи (6).
- Их все еще слишком сложно решить напрямую, поэтому релаксация применялась к ним обоим. Это задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{R}_1 \tag{8}$$

и

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{R}_2 \qquad (10).$$

- Поскольку релаксация не решила задачу (6), она была разделена на подзадачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_1 \qquad (9)$$

и

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{F}_2 \qquad (11)$$

- Обе подзадачи имеют собственное оптимальное решение, и лучшее из них — оптимальное решение задачи (6).
- Их все еще слишком сложно решить напрямую, поэтому релаксация применялась к ним обоим. Это задачи

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{R}_1 \qquad (8)$$

и

$$\max(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5), \mathbf{x} \in \mathcal{R}_2 \qquad (10).$$

- Природа задач (8) и (10) с математической точки зрения такая же, как и у задачи (7).

- Обратите внимание, что объединение множеств возможных решений (8) и (10) является собственным подмножеством релаксации (7), т. е.

$$\mathcal{R}_1 \cup \mathcal{R}_2 \subset \mathcal{R}_0.$$

- Обратите внимание, что объединение множеств возможных решений (8) и (10) является собственным подмножеством релаксации (7), т. е.

$$\mathcal{R}_1 \cup \mathcal{R}_2 \subset \mathcal{R}_0.$$

- Более того, эти два подмножества не имеют общих элементов, т. е.

$$\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset.$$

- Обратите внимание, что объединение множеств возможных решений (8) и (10) является собственным подмножеством релаксации (7), т. е.

$$\mathcal{R}_1 \cup \mathcal{R}_2 \subset \mathcal{R}_0.$$

- Более того, эти два подмножества не имеют общих элементов, т. е.

$$\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset.$$

- Это верно и для всех других случаев.

- Обратите внимание, что объединение множеств возможных решений (8) и (10) является собственным подмножеством релаксации (7), т. е.

$$\mathcal{R}_1 \cup \mathcal{R}_2 \subset \mathcal{R}_0.$$

- Более того, эти два подмножества не имеют общих элементов, т. е.

$$\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset.$$

- Это верно и для всех других случаев.
- Причина в том, что ветвление, то есть определение подзадач (9) и (11), было сделано таким образом, чтобы оптимальное решение релаксации, то есть оптимальное решение (7), было отрезано.

- Политика ветвления также влияет на верхние границы.

- Политика ветвления также влияет на верхние границы.
- Пусть $\nu(\mathcal{S})$ — оптимальное значение задачи, при которой целевая функция не изменяется, а множество допустимых решений — $\mathcal{S}$.

- Политика ветвления также влияет на верхние границы.
- Пусть $\nu(\mathcal{S})$ — оптимальное значение задачи, при которой целевая функция не изменяется, а множество допустимых решений — $\mathcal{S}$.
- Используя эти обозначения, оптимальные значения целевой функции исходной и ослабленной задач находятся в соотношении

$$\nu(\mathcal{F}) \leq \nu(\mathcal{R}).$$

- Политика ветвления также влияет на верхние границы.
- Пусть $\nu(\mathcal{S})$ — оптимальное значение задачи, при которой целевая функция не изменяется, а множество допустимых решений — $\mathcal{S}$.
- Используя эти обозначения, оптимальные значения целевой функции исходной и ослабленной задач находятся в соотношении

$$\nu(\mathcal{F}) \leq \nu(\mathcal{R}).$$

- Если подмножество $\mathcal{R}_k$ разделить на $\mathcal{R}_p$ и $\mathcal{R}_q$, то

$$\nu(\mathcal{R}_k) \geq \max\{\nu(\mathcal{R}_p), \nu(\mathcal{R}_q)\}. \tag{12}$$

- Политика ветвления также влияет на верхние границы.
- Пусть $\nu(\mathcal{S})$ — оптимальное значение задачи, при которой целевая функция не изменяется, а множество допустимых решений — $\mathcal{S}$.
- Используя эти обозначения, оптимальные значения целевой функции исходной и ослабленной задач находятся в соотношении
$$\nu(\mathcal{F}) \leq \nu(\mathcal{R}).$$
- Если подмножество $\mathcal{R}_k$ разделить на $\mathcal{R}_p$ и $\mathcal{R}_q$, то
$$\nu(\mathcal{R}_k) \geq \max\{\nu(\mathcal{R}_p), \nu(\mathcal{R}_q)\}. \tag{12}$$
- Обратите внимание, что в рассматриваемой задаче всегда выполняется строгое неравенство:
$$\nu(\mathcal{R}_0) > \max\{\nu(\mathcal{R}_1), \nu(\mathcal{R}_2)\},$$
$$\nu(\mathcal{R}_1) > \max\{\nu(\mathcal{R}_7), \nu(\mathcal{R}_8)\},$$
$$\nu(\mathcal{R}_2) > \max\{\nu(\mathcal{R}_3), \nu(\mathcal{R}_4)\},$$
$$\nu(\mathcal{R}_4) > \max\{\nu(\mathcal{R}_5), \nu(\mathcal{R}_6)\}.$$
(Значения $\nu(\mathcal{R}_7)$ и $\nu(\mathcal{R}_8)$ были только упомянуты.)

- Если сравнить верхние границы определенной величины, можно сделать вывод, что чем меньше, тем лучше, поскольку она ближе к оцениваемому значению.

- Если сравнить верхние границы определенной величины, можно сделать вывод, что чем меньше, тем лучше, поскольку она ближе к оцениваемому значению.
- Уравнение, подобное (12), верно для нерелаксированных задач, т.е. если $\mathcal{F}_k = \mathcal{F}_p \cup \mathcal{F}_q$, то

  $$\nu(\mathcal{F}_k) = \max\{\nu(\mathcal{F}_p), \nu(\mathcal{F}_q)\}, \tag{13}$$

  но из-за сложности решения подзадач практически невозможно использовать (13) для получения дополнительной информации.

- Пусть для рассматриваемой подзадачи решена ее релаксация,

- Пусть для рассматриваемой подзадачи решена ее релаксация,
- тогда дальнейшее ее исследование не требуется, если

- Пусть для рассматриваемой подзадачи решена ее релаксация,
- тогда дальнейшее ее исследование не требуется, если
  - решение релаксации — целочисленное, (нерелаксированное) оптимальное решение подзадачи совпадает с решением релаксации (в нашем примере это случй $\mathcal{F}_3$), или

- Пусть для рассматриваемой подзадачи решена ее релаксация,
- тогда дальнейшее ее исследование не требуется, если
  - решение релаксации — целочисленное, (нерелаксированное) оптимальное решение подзадачи совпадает с решением релаксации (в нашем примере это случай $\mathcal{F}_3$), или
  - доказано, что релаксация не имеет решений (случай $\mathcal{F}_6$), или

- Пусть для рассматриваемой подзадачи решена ее релаксация,
- тогда дальнейшее ее исследование не требуется, если
    - решение релаксации — целочисленное, (нерелаксированное) оптимальное решение подзадачи совпадает с решением релаксации (в нашем примере это случй $\mathcal{F}_3$), или
    - доказано, что релаксация не имеет решений (случай $\mathcal{F}_6$), или
    - ее верхняя граница не превосходит наилудшего среди уже найденных значений допустимых решений (случаи $\mathcal{F}_1$ и $\mathcal{F}_5$).

- Пусть для рассматриваемой подзадачи решена ее релаксация,
- тогда дальнейшее ее исследование не требуется, если
  - решение релаксации — целочисленное, (нерелаксированное) оптимальное решение подзадачи совпадает с решением релаксации (в нашем примере это случй $\mathcal{F}_3$), или
  - доказано, что релаксация не имеет решений (случай $\mathcal{F}_6$), или
  - ее верхняя граница не превосходит наилудшего среди уже найденных значений допустимых решений (случаи $\mathcal{F}_1$ и $\mathcal{F}_5$).

- Пусть для рассматриваемой подзадачи решена ее релаксация,
- тогда дальнейшее ее исследование не требуется, если
  - решение релаксации — целочисленное, (нерелаксированное) оптимальное решение подзадачи совпадает с решением релаксации (в нашем примере это случй $\mathcal{F}_3$), или
  - доказано, что релаксация не имеет решений (случай $\mathcal{F}_6$), или
  - ее верхняя граница не превосходит наилудшего среди уже найденных значений допустимых решений (случаи $\mathcal{F}_1$ и $\mathcal{F}_5$).

Если первое или третье из этих условий выполнено, то все возможные решения подзадачи перечисляются явным образом.

- Подзадачи, которые генерируются на одной и той же итерации представлены двумя ветвями на дереве поиска.

- Подзадачи, которые генерируются на одной и той же итерации представлены двумя ветвями на дереве поиска.
- Они братья и сестры и имеют одного родителя.

- Подзадачи, которые генерируются на одной и той же итерации представлены двумя ветвями на дереве поиска.
- Они братья и сестры и имеют одного родителя.
- Рисунок 1 визуализирует ход вычислений с использованием отношения родитель-потомок.

- Подзадачи, которые генерируются на одной и той же итерации представлены двумя ветвями на дереве поиска.
- Они братья и сестры и имеют одного родителя.
- Рисунок 1 визуализирует ход вычислений с использованием отношения родитель-потомок.
- Дерево поиска модифицируется конструктивными шагами, когда формируются новые ветви, а также шагами сокращения, когда некоторые ветви могут быть удалены, поскольку выполняется один из трех вышеупомянутых критериев.

- Подзадачи, которые генерируются на одной и той же итерации представлены двумя ветвями на дереве поиска.
- Они братья и сестры и имеют одного родителя.
- Рисунок 1 визуализирует ход вычислений с использованием отношения родитель-потомок.
- Дерево поиска модифицируется конструктивными шагами, когда формируются новые ветви, а также шагами сокращения, когда некоторые ветви могут быть удалены, поскольку выполняется один из трех вышеупомянутых критериев.
- Метод останавливается, когда не остается подмножества, которое еще предстоит исследовать.

- Метод B&B и неявное перечисление могут легко взаимодействовать. Неявное перечисление использует так называемые тесты и получает последствия для значений переменных.

- Метод B&B и неявное перечисление могут легко взаимодействовать. Неявное перечисление использует так называемые тесты и получает последствия для значений переменных.
- Например, если значение переменной x_3 зафиксировано на 1, то из неравенства на вместимость рюкзака сразу следует, что x_5 должна быть равна 0, в противном случае вместимость рюкзака будет превышена. Это верно для всей ветви 2.

Как ускорить метод

- Метод B&B и неявное перечисление могут легко взаимодействовать. Неявное перечисление использует так называемые тесты и получает последствия для значений переменных.
- Например, если значение переменной x_3 зафиксировано на 1, то из неравенства на вместимость рюкзака сразу следует, что x_5 должна быть равна 0, в противном случае вместимость рюкзака будет превышена. Это верно для всей ветви 2.
- С другой стороны, если значение целевой функции должно быть не менее 65, что является значением найденного допустимого решения, то в ветви 1 можно сделать вывод, что пятый объект должен находиться в рюкзаке, т.е. переменная x_5 должна быть равна 1, так как общая стоимость остальных объектов 1, 2 и 4 составляет всего 56.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.
- Обе ветви 1 и 2 имеют по 16 случаев.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.
- Обе ветви 1 и 2 имеют по 16 случаев.
- Если можно определить значение переменной, то количество возможных случаев сокращается вдвое.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.
- Обе ветви 1 и 2 имеют по 16 случаев.
- Если можно определить значение переменной, то количество возможных случаев сокращается вдвое.
- В приведенном примере это означает, что осталось расследовать только 8 случаев в обоих ветвях.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.
- Обе ветви 1 и 2 имеют по 16 случаев.
- Если можно определить значение переменной, то количество возможных случаев сокращается вдвое.
- В приведенном примере это означает, что осталось расследовать только 8 случаев в обоих ветвях.
- Это небольшой пример. Но в случае более крупных задач процесс сокращения намного выше.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.
- Обе ветви 1 и 2 имеют по 16 случаев.
- Если можно определить значение переменной, то количество возможных случаев сокращается вдвое.
- В приведенном примере это означает, что осталось расследовать только 8 случаев в обоих ветвях.
- Это небольшой пример. Но в случае более крупных задач процесс сокращения намного выше.
- Например, если в ветви 21 свободная, т.е. нефиксированная, переменная, но можно определить значение одной из них, то не нужно исследовать 1 048 576 случаев.

Как ускорить метод

- Почему такие последствия ускоряют работу алгоритма?
- В примере 5 двоичных переменных, поэтому количество возможных случаев составляет $32 = 2^5$.
- Обе ветви 1 и 2 имеют по 16 случаев.
- Если можно определить значение переменной, то количество возможных случаев сокращается вдвое.
- В приведенном примере это означает, что осталось расследовать только 8 случаев в обоих ветвях.
- Это небольшой пример. Но в случае более крупных задач процесс сокращения намного выше.
- Например, если в ветви 21 свободная, т.е. нефиксированная, переменная, но можно определить значение одной из них, то не нужно исследовать 1 048 576 случаев.
- Конечно, применение тестов требует дополнительных расчетов. Таким образом, необходимо найти хороший компромисс.

Общий метод B&B

- Цель этого раздела — дать общее описание метода B&B.

Общий метод B&B

- Цель этого раздела — дать общее описание метода B&B.
- Частные реализации общей схемы обсуждаются в следующих разделах.

Общий метод B&B

- Цель этого раздела — дать общее описание метода B&B.
- Частные реализации общей схемы обсуждаются в следующих разделах.
- Метод B&B основан на понятии **релаксация.** Оно еще не определено. Поскольку существует несколько видов релаксаций, первый подраздел посвящен этому понятию.

Общий метод B&B

- Цель этого раздела — дать общее описание метода B&B.
- Частные реализации общей схемы обсуждаются в следующих разделах.
- Метод B&B основан на понятии **релаксация.** Оно еще не определено. Поскольку существует несколько видов релаксаций, первый подраздел посвящен этому понятию.
- Общая схема обсуждается во втором подразделе.

Релаксации

- Релаксации обсуждается в два этапа. Есть несколько техник, позволяющих определить релаксацию для конкретной задачи.

Релаксации

- Релаксации обсуждается в два этапа. Есть несколько техник, позволяющих определить релаксацию для конкретной задачи.
- Нет правила выбора среди них. Это зависит от разработчика алгоритма, который выбирает подходящую для алгоритма.

Релаксации

- Релаксации обсуждается в два этапа. Есть несколько техник, позволяющих определить релаксацию для конкретной задачи.
- Нет правила выбора среди них. Это зависит от разработчика алгоритма, который выбирает подходящую для алгоритма.
- Различные типы обсуждаются в первой части, озаглавленной “Релаксации конкретной задачи”.

Релаксации

- Релаксации обсуждается в два этапа. Есть несколько техник, позволяющих определить релаксацию для конкретной задачи.
- Нет правила выбора среди них. Это зависит от разработчика алгоритма, который выбирает подходящую для алгоритма.
- Различные типы обсуждаются в первой части, озаглавленной "Релаксации конкретной задачи".
- В ходе решения задачи (6) были сгенерированы подзадачи, которые были задачами о рюкзаке. У них были свои собственные релаксации, которые не были полностью независимы от релаксациий друг друга и исходной задачи.

Релаксации

- Релаксации обсуждается в два этапа. Есть несколько техник, позволяющих определить релаксацию для конкретной задачи.
- Нет правила выбора среди них. Это зависит от разработчика алгоритма, который выбирает подходящую для алгоритма.
- Различные типы обсуждаются в первой части, озаглавленной "Релаксации конкретной задачи".
- В ходе решения задачи (6) были сгенерированы подзадачи, которые были задачами о рюкзаке. У них были свои собственные релаксации, которые не были полностью независимы от релаксаций друг друга и исходной задачи.
- Ожидаемые общие свойства и структура анализируются на втором этапе под заголовком "Релаксации для класса задач".

Релаксации конкретной задачи

- Описание проблемы (6) состоит из трех частей: (1) целевая функция, (2) алгебраические ограничения и (3) требование, чтобы переменные были двоичными.

- Описание проблемы (6) состоит из трех частей: (1) целевая функция, (2) алгебраические ограничения и (3) требование, чтобы переменные были двоичными.
- Такая структура типична для задач оптимизации ВЦП.

Релаксации конкретной задачи

- Описание проблемы (6) состоит из трех частей: (1) целевая функция, (2) алгебраические ограничения и (3) требование, чтобы переменные были двоичными.
- Такая структура типична для задач оптимизации ВЦП.
- В общей постановке задачу оптимизации можно представить в виде

$$\max f(\mathbf{x}) \tag{14}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Основная логика создания релаксации (7) заключается в том, что ограничение $\mathbf{x} \in \mathcal{X}$ (16) было заменено более слабым.

Релаксация неалгебраических ограничений

- Основная логика создания релаксации (7) заключается в том, что ограничение $\mathbf{x} \in \mathcal{X}$ (16) было заменено более слабым.
- В конкретном случае было разрешено, что переменные могут принимать любое значение от 0 до 1.

Релаксация неалгебраических ограничений

- Основная логика создания релаксации (7) заключается в том, что ограничение $\mathbf{x} \in \mathcal{X}$ (16) было заменено более слабым.
- В конкретном случае было разрешено, что переменные могут принимать любое значение от 0 до 1.
- Обычно (16) заменяется на требование, чтобы переменные принадлежали множеству, скажем, $\mathcal{Y}$, которое больше, чем $\mathcal{X}$, т.е. отношение $\mathcal{X} \subseteq \mathcal{Y}$ должно выполняться.

- Основная логика создания релаксации (7) заключается в том, что ограничение $\mathbf{x} \in \mathcal{X}$ (16) было заменено более слабым.
- В конкретном случае было разрешено, что переменные могут принимать любое значение от 0 до 1.
- Обычно (16) заменяется на требование, чтобы переменные принадлежали множеству, скажем, $\mathcal{Y}$, которое больше, чем $\mathcal{X}$, т.е. отношение $\mathcal{X} \subseteq \mathcal{Y}$ должно выполняться.
- Более формально релаксация задачи (14)-(16) — это задача

$$\max f(\mathbf{x}) \tag{14}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{Y}. \tag{17}$$

Релаксация неалгебраических ограничений

- Основная логика создания релаксации (7) заключается в том, что ограничение $\mathbf{x} \in \mathcal{X}$ (16) было заменено более слабым.
- В конкретном случае было разрешено, что переменные могут принимать любое значение от 0 до 1.
- Обычно (16) заменяется на требование, чтобы переменные принадлежали множеству, скажем, $\mathcal{Y}$, которое больше, чем $\mathcal{X}$, т.е. отношение $\mathcal{X} \subseteq \mathcal{Y}$ должно выполняться.
- Более формально релаксация задачи (14)-(16) — это задача

$$\max f(\mathbf{x}) \tag{14}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{Y}. \tag{17}$$

- Этот тип релаксации может быть применен, если можно устранить большую трудоемкость решения, изменив характер переменных.

Релаксация алгебраических ограничений

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.

Релаксация алгебраических ограничений

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.
- Способ такого ослабления заключается в следующем.

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.
- Способ такого ослабления заключается в следующем.
- Предположим, что в (15) имеется m неравенств.

Релаксация алгебраических ограничений

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.
- Способ такого ослабления заключается в следующем.
- Предположим, что в (15) имеется m неравенств.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.

Релаксация алгебраических ограничений

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.
- Способ такого ослабления заключается в следующем.
- Предположим, что в (15) имеется m неравенств.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Тогда для любого $\mathbf{x} \in \mathcal{X}$, удовлетворяющего (15), также выполняется неравенство

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \leq \sum_{i=1}^{m} \lambda_i b_i. \tag{18}$$

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.
- Способ такого ослабления заключается в следующем.
- Предположим, что в (15) имеется m неравенств.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Тогда для любого $\mathbf{x} \in \mathcal{X}$, удовлетворяющего (15), также выполняется неравенство

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \leq \sum_{i=1}^{m} \lambda_i b_i. \tag{18}$$

- Тогда релаксация — это оптимизация целевой функции (14) при условиях (18) и (16).

Релаксация алгебраических ограничений

- Существует аналогичная техника, при которой неравенства $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ (15) ослабляются, а ограничение $\mathbf{x} \in \mathcal{X}$ (16) не изменяется.
- Способ такого ослабления заключается в следующем.
- Предположим, что в (15) имеется m неравенств.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Тогда для любого $\mathbf{x} \in \mathcal{X}$, удовлетворяющего (15), также выполняется неравенство

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \leq \sum_{i=1}^{m} \lambda_i b_i. \tag{18}$$

- Тогда релаксация — это оптимизация целевой функции (14) при условиях (18) и (16).
- Неравенства (18) называют суррогатными ограничениями.

- Задача

$$\begin{array}{rrcrcrcrcrcl}
\max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\
 & 5x_1 & + & 4x_2 & + & 6x_3 & + & 3x_4 & + & 5x_5 & \leq & 14 \\
 & 2x_1 & - & 2x_2 & - & 3x_3 & + & 5x_4 & + & 6x_5 & \leq & 4 \\
 & 1x_1 & + & 5x_2 & + & 8x_3 & - & 2x_4 & + & 8x_5 & \leq & 7 \\
\multicolumn{12}{c}{x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1}
\end{array} \tag{19}$$

является общей задачей булевой оптимизации.

- Задача

$$
\begin{array}{lrcrcrcrcrcl}
\max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\
 & 5x_1 & + & 4x_2 & + & 6x_3 & + & 3x_4 & + & 5x_5 & \leq & 14 \\
 & 2x_1 & - & 2x_2 & - & 3x_3 & + & 5x_4 & + & 6x_5 & \leq & 4 \\
 & 1x_1 & + & 5x_2 & + & 8x_3 & - & 2x_4 & + & 8x_5 & \leq & 7 \\
\end{array} \qquad (19)
$$

$$x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1$$

является общей задачей булевой оптимизации.

- Если $\lambda_1 = \lambda_2 = \lambda_3 = 1$, то релаксация, полученная таким образом, является задачей (6).

- Задача

$$\begin{array}{rrcrcrcrcrcrcl} \max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\ & 5x_1 & + & 4x_2 & + & 6x_3 & + & 3x_4 & + & 5x_5 & \leq & 14 \\ & 2x_1 & - & 2x_2 & - & 3x_3 & + & 5x_4 & + & 6x_5 & \leq & 4 \\ & 1x_1 & + & 5x_2 & + & 8x_3 & - & 2x_4 & + & 8x_5 & \leq & 7 \\ \multicolumn{12}{c}{x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1} \end{array} \tag{19}$$

  является общей задачей булевой оптимизации.

- Если $\lambda_1 = \lambda_2 = \lambda_3 = 1$, то релаксация, полученная таким образом, является задачей (6).

- Обе задачи принадлежат к классу NP-трудных.

- Задача

$$\begin{array}{lrcrcrcrcrcl}
\max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & \\
& 5x_1 & + & 4x_2 & + & 6x_3 & + & 3x_4 & + & 5x_5 & \le & 14 \\
& 2x_1 & - & 2x_2 & - & 3x_3 & + & 5x_4 & + & 6x_5 & \le & 4 \\
& 1x_1 & + & 5x_2 & + & 8x_3 & - & 2x_4 & + & 8x_5 & \le & 7 \\
\multicolumn{12}{c}{x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1}
\end{array} \tag{19}$$

  является общей задачей булевой оптимизации.
- Если $\lambda_1 = \lambda_2 = \lambda_3 = 1$, то релаксация, полученная таким образом, является задачей (6).
- Обе задачи принадлежат к классу NP-трудных.
- Однако задача о рюкзаке значительно проще с практической точки зрения, чем общая задача, поэтому релаксация может иметь смысл.

- Задача

$$\begin{array}{rrcrcrcrcrcrl} \max & 23x_1 & + & 19x_2 & + & 28x_3 & + & 14x_4 & + & 44x_5 & & & \\ & 5x_1 & + & 4x_2 & + & 6x_3 & + & 3x_4 & + & 5x_5 & \leq & 14 & \\ & 2x_1 & - & 2x_2 & - & 3x_3 & + & 5x_4 & + & 6x_5 & \leq & 4 & \quad (19) \\ & 1x_1 & + & 5x_2 & + & 8x_3 & - & 2x_4 & + & 8x_5 & \leq & 7 & \\ \end{array}$$
$$x_1, x_2, x_3, x_4, x_5 = 0 \text{ или } 1$$

является общей задачей булевой оптимизации.

- Если $\lambda_1 = \lambda_2 = \lambda_3 = 1$, то релаксация, полученная таким образом, является задачей (6).
- Обе задачи принадлежат к классу NP-трудных.
- Однако задача о рюкзаке значительно проще с практической точки зрения, чем общая задача, поэтому релаксация может иметь смысл.
- Обратите внимание, что в этой конкретной задаче оптимальное решение задачи о рюкзаке, то есть (1,0,1,1,0), удовлетворяет ограничениям (19), таким образом, это также оптимальное решение последней задачи.

- Суррогатное ограничение — не единственный вариант ослабления алгебраических ограничений.

- Суррогатное ограничение — не единственный вариант ослабления алгебраических ограничений.
- Область, определяемая нелинейными граничными поверхностями, может быть апроксимирована касательными плоскостями.

- Суррогатное ограничение — не единственный вариант ослабления алгебраических ограничений.
- Область, определяемая нелинейными граничными поверхностями, может быть апроксимирована касательными плоскостями.
- Например, если допустимой областью является единичная окружность, задаваемая неравенством

$$x_1^2 + x_2^2 \leq 1,$$

то она может быть апроксимирована квадратом:

$$-1 \leq x_1, x_2 \leq 1.$$

- Суррогатное ограничение — не единственный вариант ослабления алгебраических ограничений.
- Область, определяемая нелинейными граничными поверхностями, может быть апроксимирована касательными плоскостями.
- Например, если допустимой областью является единичная окружность, задаваемая неравенством

$$x_1^2 + x_2^2 \leq 1,$$

то она может быть апроксимирована квадратом:

$$-1 \leq x_1, x_2 \leq 1.$$

- Если оптимальное решение на увеличенной области, например, точка (1,1) не находится в исходной допустимой области, то необходимо найти *отсечение*, которое удаляет ее из ослабленной области, но не отсекает какую-либо часть исходной допустимой области.

- Это делается, например, неравенством

$$x_1 + x_2 \leq \sqrt{2}.$$

- Это делается, например, неравенством

$$x_1 + x_2 \leq \sqrt{2}.$$

- Новая осслабленная задача определяется введением разреза.

- Это делается, например, неравенством

$$x_1 + x_2 \leq \sqrt{2}.$$

- Новая ослабленная задача определяется введением разреза.
- Этот метод аналогичен одному из методов ослабления целевой функции, обсуждаемых ниже.

Ослабление целевой функции

- В некоторых случаях сложность задачи вызвана целевой функцией.

Ослабление целевой функции

- В некоторых случаях сложность задачи вызвана целевой функцией.
- Если возможно использовать более простую целевую функцию, скажем $h(\mathbf{x})$, то для получения верхней границы условие

$$\forall\, \mathbf{x} \in \mathcal{X} : \; h(\mathbf{x}) \geq f(\mathbf{x}) \tag{20}$$

должно выполняться.

Ослабление целевой функции

- В некоторых случаях сложность задачи вызвана целевой функцией.
- Если возможно использовать более простую целевую функцию, скажем $h(\mathbf{x})$, то для получения верхней границы условие

$$\forall \mathbf{x} \in \mathcal{X} : \; h(\mathbf{x}) \geq f(\mathbf{x}) \tag{20}$$

должно выполняться.

- Тогда релаксацией является задача

$$\max h(\mathbf{x}) \tag{21}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

Ослабление целевой функции

- Этот тип релаксации типичен, если метод B&B применяется для решения задачи (непрерывной) нелинейной оптимизации.

Ослабление целевой функции

- Этот тип релаксации типичен, если метод B&B применяется для решения задачи (непрерывной) нелинейной оптимизации.
- Важным подклассом задач нелинейной оптимизации является так называемая задача выпуклого программирования.

Ослабление целевой функции

- Этот тип релаксации типичен, если метод B&B применяется для решения задачи (непрерывной) нелинейной оптимизации.
- Важным подклассом задач нелинейной оптимизации является так называемая задача выпуклого программирования.
- Это снова относительно простой подкласс. Поэтому разумно произвести релаксацию этого типа, если это возможно.

- Этот тип релаксации типичен, если метод B&B применяется для решения задачи (непрерывной) нелинейной оптимизации.
- Важным подклассом задач нелинейной оптимизации является так называемая задача выпуклого программирования.
- Это снова относительно простой подкласс. Поэтому разумно произвести релаксацию этого типа, если это возможно.
- Задача (14) - (16) — это задача выпуклого программирования, если $\mathcal{X}$ — выпуклое множество, функции $g_i(\mathbf{x})$ $(i = 1, \ldots, m)$ выпуклые, а целевая функция $f(\mathbf{x})$ вогнутая.

- Этот тип релаксации типичен, если метод B&B применяется для решения задачи (непрерывной) нелинейной оптимизации.
- Важным подклассом задач нелинейной оптимизации является так называемая задача выпуклого программирования.
- Это снова относительно простой подкласс. Поэтому разумно произвести релаксацию этого типа, если это возможно.
- Задача (14) - (16) — это задача выпуклого программирования, если $\mathcal{X}$ — выпуклое множество, функции $g_i(\mathbf{x})$ $(i = 1, \ldots, m)$ выпуклые, а целевая функция $f(\mathbf{x})$ вогнутая.
- Таким образом, релаксация может быть задачей выпуклого программирования, если нарушается только последнее условие.

Ослабление целевой функции

- Этот тип релаксации типичен, если метод B&B применяется для решения задачи (непрерывной) нелинейной оптимизации.
- Важным подклассом задач нелинейной оптимизации является так называемая задача выпуклого программирования.
- Это снова относительно простой подкласс. Поэтому разумно произвести релаксацию этого типа, если это возможно.
- Задача (14) - (16) — это задача выпуклого программирования, если $\mathcal{X}$ — выпуклое множество, функции $g_i(\mathbf{x})$ $(i = 1, \ldots, m)$ выпуклые, а целевая функция $f(\mathbf{x})$ вогнутая.
- Таким образом, релаксация может быть задачей выпуклого программирования, если нарушается только последнее условие.
- Тогда достаточно найти вогнутую функцию $h(\mathbf{x})$ такую, что выполняется (20).

Пример

- Например, функция одной переменной $f(x) = 2x^2 - x^4$ не является вогнутой в интервале $[\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}]$. [1]

[1] Непрерывная функция называется вогнутой, если ее вторая производная не отрицательна. $f^{''}(x) = 4 - 12x^2$, положительна в открытом интервале $\left(\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right)$.

- Например, функция одной переменной $f(x) = 2x^2 - x^4$ не является вогнутой в интервале $[\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}]$. [1]
- Таким образом, если это целевая функция в задаче оптимизации, то в релаксации ее, можно заменить вогнутой функцией $h(x)$ такой, что $\forall x \in [\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}] : f(x) \leq h(x)$.

[1] Непрерывная функция называется вогнутой, если ее вторая производная не отрицательна. $f^{''}(x) = 4 - 12x^2$, положительна в открытом интервале $\left(\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right)$.

Пример

- Например, функция одной переменной $f(x) = 2x^2 - x^4$ не является вогнутой в интервале $[\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}]$. [1]
- Таким образом, если это целевая функция в задаче оптимизации, то в релаксации ее, можно заменить вогнутой функцией $h(x)$ такой, что $\forall x \in [\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}] : f(x) \leq h(x)$.
- Легко проверить, что $h(x) = \frac{8}{9} - x^2$ удовлетворяет требованиям.

[1] Непрерывная функция называется вогнутой, если ее вторая производная не отрицательна. $f^{''}(x) = 4 - 12x^2$, положительна в открытом интервале $\left(\frac{-\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right)$.

- Пусть $\mathbf{x}^*$ — оптимальное решение ослабленной задачи (21), (15) и (16).

- Пусть $\mathbf{x}^*$ — оптимальное решение ослабленной задачи (21), (15) и (16).
- Оно является решением исходной задачи, если оптимальное решение имеет одинаковое значение целевой функции в исходной и ослабленной задачах, т. е. $f(\mathbf{x}^*) = h(\mathbf{x}^*)$.

- Другая причина, по которой применяется этот тип релаксации, заключается в том, что в некоторых случаях целевая функция не известна в замкнутой форме, однако ее можно определить в любой заданной точке.

- Другая причина, по которой применяется этот тип релаксации, заключается в том, что в некоторых случаях целевая функция не известна в замкнутой форме, однако ее можно определить в любой заданной точке.
- Это может произойти даже в том случае, если целевая функция вогнутая.

- Другая причина, по которой применяется этот тип релаксации, заключается в том, что в некоторых случаях целевая функция не известна в замкнутой форме, однако ее можно определить в любой заданной точке.
- Это может произойти даже в том случае, если целевая функция вогнутая.
- Предположим, что значение $f(\mathbf{x})$ известно в точках $\mathbf{y}_1, \ldots, \mathbf{y}_k$.

- Другая причина, по которой применяется этот тип релаксации, заключается в том, что в некоторых случаях целевая функция не известна в замкнутой форме, однако ее можно определить в любой заданной точке.
- Это может произойти даже в том случае, если целевая функция вогнутая.
- Предположим, что значение $f(\mathbf{x})$ известно в точках $\mathbf{y}_1, \ldots, \mathbf{y}_k$.
- Если $f(\mathbf{x})$ вогнутая, то она гладкая, т.е. ее градиент существует.

- Другая причина, по которой применяется этот тип релаксации, заключается в том, что в некоторых случаях целевая функция не известна в замкнутой форме, однако ее можно определить в любой заданной точке.
- Это может произойти даже в том случае, если целевая функция вогнутая.
- Предположим, что значение $f(\mathbf{x})$ известно в точках $\mathbf{y}_1, \ldots, \mathbf{y}_k$.
- Если $f(\mathbf{x})$ вогнутая, то она гладкая, т.е. ее градиент существует.
- Градиент определяет касательную плоскость, которая находится над функцией.

- Уравнение касательной плоскости в точке $\mathbf{y_p}$ имеет вид [2]

$$\nabla(f(\mathbf{y_p}))(\mathbf{x} - \mathbf{y_p}) = 0.$$

[2] Градиент считается вектором-строкой.

- Уравнение касательной плоскости в точке $\mathbf{y_p}$ имеет вид [2]

$$\nabla(f(\mathbf{y_p}))(\mathbf{x}-\mathbf{y_p}) = 0.$$

- Следовательно, во всех точках области определения функции $f(\mathbf{x})$ имеем

$$h(\mathbf{x}) = \min\{f(\mathbf{y_p}) + \nabla(f(\mathbf{y_p}))(\mathbf{x}-\mathbf{y_p}) \mid p = 1, \ldots, k\} \geq f(\mathbf{x}).$$

[2] Градиент считается вектором-строкой.

- Уравнение касательной плоскости в точке $\mathbf{y_p}$ имеет вид [2]

$$\nabla(f(\mathbf{y_p}))(\mathbf{x}-\mathbf{y_p})=0.$$

- Следовательно, во всех точках области определения функции $f(\mathbf{x})$ имеем

$$h(\mathbf{x})=\min\{f(\mathbf{y_p})+\nabla(f(\mathbf{y_p}))(\mathbf{x}-\mathbf{y_p})\mid p=1,\ldots,k\}\geq f(\mathbf{x}).$$

- Очевидно, что функция $h(\mathbf{x})$ апроксимирует функцию $f(\mathbf{x})$.

[2] Градиент считается вектором-строкой.

Пример

- Идея метода проиллюстрирована на следующем числовом примере.

Пример

- Идея метода проиллюстрирована на следующем числовом примере.
- Предположим, что «неизвестная» вогнутая функция должна быть максимизирована на отрезке [0,5].

Пример

- Идея метода проиллюстрирована на следующем числовом примере.
- Предположим, что «неизвестная» вогнутая функция должна быть максимизирована на отрезке [0,5].
- Метод может стартовать из любой точки интервала, находящейся в допустимой области.

Пример

- Идея метода проиллюстрирована на следующем числовом примере.
- Предположим, что «неизвестная» вогнутая функция должна быть максимизирована на отрезке [0,5].
- Метод может стартовать из любой точки интервала, находящейся в допустимой области.
- Пусть 0 будет стартовой точкой.

Пример

- Идея метода проиллюстрирована на следующем числовом примере.
- Предположим, что «неизвестная» вогнутая функция должна быть максимизирована на отрезке [0,5].
- Метод может стартовать из любой точки интервала, находящейся в допустимой области.
- Пусть 0 будет стартовой точкой.
- Согласно предположениям, хотя замкнутая формула функции неизвестна, можно определить значения функции и ее производной.

Пример

- Идея метода проиллюстрирована на следующем числовом примере.
- Предположим, что «неизвестная» вогнутая функция должна быть максимизирована на отрезке [0,5].
- Метод может стартовать из любой точки интервала, находящейся в допустимой области.
- Пусть 0 будет стартовой точкой.
- Согласно предположениям, хотя замкнутая формула функции неизвестна, можно определить значения функции и ее производной.
- Теперь получены значения $f(0) = -4$ и $f^{'}(0) = 4$.

Пример

- Идея метода проиллюстрирована на следующем числовом примере.
- Предположим, что «неизвестная» вогнутая функция должна быть максимизирована на отрезке [0,5].
- Метод может стартовать из любой точки интервала, находящейся в допустимой области.
- Пусть 0 будет стартовой точкой.
- Согласно предположениям, хотя замкнутая формула функции неизвестна, можно определить значения функции и ее производной.
- Теперь получены значения $f(0) = -4$ и $f^{'}(0) = 4$.
- Общая формула касательной в точке $(x_0, f(x_0))$:

$$y = f^{'}(x_0)(x - x_0) + f(x_0).$$

- Следовательно, уравнение первой касательной имеет вид $y = 4x - 4$, поэтому первая задача оптимизации ставится так

$$\begin{gathered} \max\ h \\ h \leq 4x - 4 \\ x \in [0, 5]. \end{gathered}$$

- Следовательно, уравнение первой касательной имеет вид $y = 4x - 4$, поэтому первая задача оптимизации ставится так

$$\begin{gathered} \max\ h \\ h \le 4x - 4 \\ x \in [0, 5]. \end{gathered}$$

- Поскольку $4x - 4$ — монотонно возрастающая функция, оптимальное решение — $x = 5$.

- Следовательно, уравнение первой касательной имеет вид $y = 4x - 4$, поэтому первая задача оптимизации ставится так

$$\begin{gathered} \max\ h \\ h \leq 4x - 4 \\ x \in [0, 5]. \end{gathered}$$

- Поскольку $4x - 4$ — монотонно возрастающая функция, оптимальное решение — $x = 5$.
- Тогда значения $f(5) = -9$ и $f'(5) = -6$ предоставляют метод вычисления функции.

- Следовательно, уравнение первой касательной имеет вид $y = 4x - 4$, поэтому первая задача оптимизации ставится так

$$\begin{gathered} \max h \\ h \leq 4x - 4 \\ x \in [0, 5]. \end{gathered}$$

- Поскольку $4x - 4$ — монотонно возрастающая функция, оптимальное решение — $x = 5$.
- Тогда значения $f(5) = -9$ и $f^{'}(5) = -6$ предоставляют метод вычисления функции.
- Уравнение второй касательной: $y = -6x + 21$. Таким образом, вторая оптимизационная проблема ставится так

$$\begin{gathered} \max h \\ h \leq 4x - 4, \quad h \leq -6x + 21 \\ x \in [0, 5]. \end{gathered}$$

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2,5$.

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2,5$.
- Затем вычисляются значения $f(2.5) = -0.25$ и $f^{'}(2.5) = -1$, и уравнение касательной прямой имеет вид $y = -x + 2.25$.

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2, 5$.
- Затем вычисляются значения $f(2.5) = -0.25$ и $f^{'}(2.5) = -1$, и уравнение касательной прямой имеет вид $y = -x + 2.25$.
- Следующая задача оптимизации —

$$\max h$$
$$h \leq 4x - 4, \quad h \leq -6x + 21, \quad h \leq -x + 2.25$$
$$x \in [0, 5].$$

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2,5$.
- Затем вычисляются значения $f(2.5) = -0.25$ и $f^{'}(2.5) = -1$, и уравнение касательной прямой имеет вид $y = -x + 2.25$.
- Следующая задача оптимизации —

$$\max h$$
$$h \leq 4x - 4, \quad h \leq -6x + 21, \quad h \leq -x + 2.25$$
$$x \in [0, 5].$$

- Оптимальное решение — $x = 1,25$. Это точка пересечения первой и третьей касательных.

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2,5$.
- Затем вычисляются значения $f(2.5) = -0.25$ и $f^{'}(2.5) = -1$, и уравнение касательной прямой имеет вид $y = -x + 2.25$.
- Следующая задача оптимизации —

$$\max h$$
$$h \leq 4x - 4, \quad h \leq -6x + 21, \quad h \leq -x + 2.25$$
$$x \in [0, 5].$$

- Оптимальное решение — $x = 1,25$. Это точка пересечения первой и третьей касательных.
- Обе новые точки пересечения находятся в интервале [0,5].

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2,5$.
- Затем вычисляются значения $f(2.5) = -0.25$ и $f^{'}(2.5) = -1$, и уравнение касательной прямой имеет вид $y = -x + 2.25$.
- Следующая задача оптимизации —

$$\max h$$
$$h \leq 4x - 4, \quad h \leq -6x + 21, \quad h \leq -x + 2.25$$
$$x \in [0, 5].$$

- Оптимальное решение — $x = 1,25$. Это точка пересечения первой и третьей касательных.
- Обе новые точки пересечения находятся в интервале [0,5].
- В общем случае некоторые точки пересечения могут быть недопустимыми.

- Поскольку вторая касательная линия является монотонно убывающей функцией, оптимальное решение находится в точке пересечения двух касательных линий, что дает $x = 2,5$.
- Затем вычисляются значения $f(2.5) = -0.25$ и $f^{'}(2.5) = -1$, и уравнение касательной прямой имеет вид $y = -x + 2.25$.
- Следующая задача оптимизации —

$$\max h$$
$$h \le 4x - 4, \quad h \le -6x + 21, \quad h \le -x + 2.25$$
$$x \in [0, 5].$$

- Оптимальное решение — $x = 1,25$. Это точка пересечения первой и третьей касательных.
- Обе новые точки пересечения находятся в интервале [0,5].
- В общем случае некоторые точки пересечения могут быть недопустимыми.
- Далее метод идет по тому же пути. Функция $f(x) = -(x-2)^2$ апроксимирует “неизвестную” функцию.

Лагранжева релаксация

- В другой релаксации, называемой релаксацией Лагранжа, изменяются как целевая функция, так и ограничения.

Лагранжева релаксация

- В другой релаксации, называемой релаксацией Лагранжа, изменяются как целевая функция, так и ограничения.
- Основная идея заключается в следующем.

Лагранжева релаксация

- В другой релаксации, называемой релаксацией Лагранжа, изменяются как целевая функция, так и ограничения.
- Основная идея заключается в следующем.
- Переменные должны удовлетворять двум различным типам ограничений, т.е. они должны удовлетворять:

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

Лагранжева релаксация

- В другой релаксации, называемой релаксацией Лагранжа, изменяются как целевая функция, так и ограничения.
- Основная идея заключается в следующем.
- Переменные должны удовлетворять двум различным типам ограничений, т.е. они должны удовлетворять:

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Причина того, что ограничения написаны в двух частях, заключается в том, что природа этих двух наборов ограничений различна.

Лагранжева релаксация

- В другой релаксации, называемой релаксацией Лагранжа, изменяются как целевая функция, так и ограничения.
- Основная идея заключается в следующем.
- Переменные должны удовлетворять двум различным типам ограничений, т.е. они должны удовлетворять:

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Причина того, что ограничения написаны в двух частях, заключается в том, что природа этих двух наборов ограничений различна.
- Сложность задачи вызвана требованием *удовлетворить обоим* ограничениям.

- В другой релаксации, называемой релаксацией Лагранжа, изменяются как целевая функция, так и ограничения.
- Основная идея заключается в следующем.
- Переменные должны удовлетворять двум различным типам ограничений, т.е. они должны удовлетворять:

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Причина того, что ограничения написаны в двух частях, заключается в том, что природа этих двух наборов ограничений различна.
- Сложность задачи вызвана требованием *удовлетворить обоим* ограничениям.
- Значительно проще выполнять условия только по одному типу ограничений. *Возможно следует устраненить одного из них?*

- Предположим, что количество неравенств в (15) равно m.

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Лагранжевая релаксация задачи (14) - (16) это

$$\max\ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \tag{22}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Лагранжевая релаксация задачи (14) - (16) это

$$\max\ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \tag{22}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Обратите внимание, что целевая функция (22) штрафует нарушение ограничений, например пытается использовать слишком много ресурсов и вознаграждает за их экономию.

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Лагранжевая релаксация задачи (14) - (16) это

$$\max\ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \tag{22}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Обратите внимание, что целевая функция (22) штрафует нарушение ограничений, например пытается использовать слишком много ресурсов и вознаграждает за их экономию.
- Первый набор ограничений исчез из задачи.

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Лагранжевая релаксация задачи (14) - (16) это

$$\max\ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \tag{22}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Обратите внимание, что целевая функция (22) штрафует нарушение ограничений, например пытается использовать слишком много ресурсов и вознаграждает за их экономию.
- Первый набор ограничений исчез из задачи.
- В большинстве случаев релаксация Лагранжа намного проще, чем исходная задача.

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Лагранжевая релаксация задачи (14) - (16) это

$$\max\ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \tag{22}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Обратите внимание, что целевая функция (22) штрафует нарушение ограничений, например пытается использовать слишком много ресурсов и вознаграждает за их экономию.
- Первый набор ограничений исчез из задачи.
- В большинстве случаев релаксация Лагранжа намного проще, чем исходная задача.
- Далее задача (14) - (16) также обозначается (P), а релаксация Лагранжа обозначается как $(L(\lambda))$.

- Предположим, что количество неравенств в (15) равно m.
- Пусть $\lambda_i \geq 0$ $(i = 1, \ldots, m)$ — фиксированные числа.
- Лагранжевая релаксация задачи (14) - (16) это

$$\max\ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \tag{22}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

- Обратите внимание, что целевая функция (22) штрафует нарушение ограничений, например пытается использовать слишком много ресурсов и вознаграждает за их экономию.
- Первый набор ограничений исчез из задачи.
- В большинстве случаев релаксация Лагранжа намного проще, чем исходная задача.
- Далее задача (14) - (16) также обозначается (P), а релаксация Лагранжа обозначается как $(L(\lambda))$.
- Обозначения отражают тот факт, что задача релаксации Лагранжа зависит от выбора чисел λ_i. Числа λ_i называются множителями Лагранжа.

Не очевидно, что $(L(\lambda))$ действительно является релаксацией (P).

Не очевидно, что $(L(\lambda))$ действительно является релаксацией (P).

Теорема

Предположим, что и (P), и $(L(\lambda))$ имеют оптимальные решения. Тогда для любого неотрицательного λ_i $(i = 1, \ldots, m)$ выполняется неравенство $\nu(L(\lambda)) \geq \nu(P)$

Не очевидно, что $(L(\lambda))$ действительно является релаксацией (P).

Теорема

Предположим, что и (P), и $(L(\lambda))$ имеют оптимальные решения. Тогда для любого неотрицательного λ_i $(i = 1, \ldots, m)$ выполняется неравенство $\nu(L(\lambda)) \geq \nu(P)$

Доказательство. Утверждение состоит в том, что оптимальное значение $(L(\lambda))$ является верхней границей оптимального значения (P). Пусть $\mathbf{x}^*$ — оптимальное решение (P). Это очевидно выполнимо в обеих задачах. Следовательно, для всех i выполняются неравенства $\lambda_i \geq 0$, $b_i \geq g_i(\mathbf{x}^*)$. Таким образом, $\lambda_i(b_i - g_i(\mathbf{x}^*)) \geq 0$ откуда следует, что

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i(b_i - g_i(\mathbf{x}^*)).$$

Не очевидно, что $(L(\lambda))$ действительно является релаксацией (P).

Теорема

Предположим, что и (P), и $(L(\lambda))$ имеют оптимальные решения. Тогда для любого неотрицательного λ_i $(i = 1, \ldots, m)$ выполняется неравенство $\nu(L(\lambda)) \geq \nu(P)$

Доказательство. Утверждение состоит в том, что оптимальное значение $(L(\lambda))$ является верхней границей оптимального значения (P). Пусть $\mathbf{x}^*$ — оптимальное решение (P). Это очевидно выполнимо в обеих задачах. Следовательно, для всех i выполняются неравенства $\lambda_i \geq 0$, $b_i \geq g_i(\mathbf{x}^*)$. Таким образом, $\lambda_i(b_i - g_i(\mathbf{x}^*)) \geq 0$ откуда следует, что

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i(b_i - g_i(\mathbf{x}^*)).$$

Здесь правая часть — значение целевой функции допустимого решения задачи $(L(\lambda))$, т.е.

$$\nu(P) = f(\mathbf{x}^*) \leq f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i(b_i - g_i(\mathbf{x}^*)) \leq \nu(L(\lambda)).$$

Существует еще одна связь между (P) и $(L(\lambda))$, которая также важна с точки зрения понятия релаксации.

Существует еще одна связь между (P) и $(L(\lambda))$, которая также важна с точки зрения понятия релаксации.

Теорема

Пусть $\mathbf{x}_L$ — оптимальное решение релаксации Лагранжа. Если

$$\mathbf{g}(\mathbf{x}_L) \leq \mathbf{b} \tag{23}$$

и

$$\sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x}_L)) = 0, \tag{24}$$

то $\mathbf{x}_L$ оптимальное решение задачи (P).

Существует еще одна связь между (P) и $(L(\lambda))$, которая также важна с точки зрения понятия релаксации.

Теорема

Пусть $\mathbf{x}_L$ — оптимальное решение релаксации Лагранжа. Если

$$\mathbf{g}(\mathbf{x}_L) \leq \mathbf{b} \tag{23}$$

и

$$\sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x}_L)) = 0, \tag{24}$$

то $\mathbf{x}_L$ оптимальное решение задачи (P).

Доказательство.(23) означает, что $\mathbf{x}_L$ является допустимым решением (P). Для любого допустимого решения $\mathbf{x}$ задачи (P) из оптимальности $\mathbf{x}_L$ следует что

$$f(\mathbf{x}) \leq f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x})) \leq f(\mathbf{x}_L) + \sum_{i=1}^{m} \lambda_i (b_i - g_i(\mathbf{x}_L)) = f(\mathbf{x}_L),$$

т.е. $\mathbf{x}_L$ по крайней мере не хуже $\mathbf{x}$.

- Важность условий (23) и (24) заключается в том, что они дают критерий оптимальности, т.е. если точка, порожденная множителями Лагранжа, удовлетворяет им, то она оптимальна в исходной задаче.

- Важность условий (23) и (24) заключается в том, что они дают критерий оптимальности, т.е. если точка, порожденная множителями Лагранжа, удовлетворяет им, то она оптимальна в исходной задаче.
- Смысл (23) в том, что оптимальное решение задачи Лагранжа является допустимым в исходной задаче, а смысл (24) заключается в том, что значения целевой функции $\mathbf{x}_L$ равны в двух задачах, как и в случае предыдущей релаксации.

- Важность условий (23) и (24) заключается в том, что они дают критерий оптимальности, т.е. если точка, порожденная множителями Лагранжа, удовлетворяет им, то она оптимальна в исходной задаче.
- Смысл (23) в том, что оптимальное решение задачи Лагранжа является допустимым в исходной задаче, а смысл (24) заключается в том, что значения целевой функции $\mathbf{x}_L$ равны в двух задачах, как и в случае предыдущей релаксации.
- Это также указывает на то, что оптимальные решения двух задач в некоторых случаях совпадают.

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.
- Этим свойством обладает релаксация Лагранжа.

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.
- Этим свойством обладает релаксация Лагранжа.
- Это можно показать для задачи (19).

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.
- Этим свойством обладает релаксация Лагранжа.
- Это можно показать для задачи (19).
- Пусть $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 3$.

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.
- Этим свойством обладает релаксация Лагранжа.
- Это можно показать для задачи (19).
- Пусть $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 3$.
- Тогда целевая функция (22) следующая

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.
- Этим свойством обладает релаксация Лагранжа.
- Это можно показать для задачи (19).
- Пусть $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 3$.
- Тогда целевая функция (22) следующая

- Существует практическое необходимое условие для того, чтобы релаксация была полезной, а именно: ослабленную задачу решить легче, чем исходную.
- Этим свойством обладает релаксация Лагранжа.
- Это можно показать для задачи (19).
- Пусть $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 3$.
- Тогда целевая функция (22) следующая

$$
\begin{aligned}
&(23x_1 + 19x_2 + 28x_3 + 14x_4 + 44x_5) + (14 - 5x_1 - x_2 - 6x_3 - 3x_4 - 5x_5) \\
&+3(4 - 2x_1 - x_2 + 3x_3 - 5x_4 - 6x_5) + 3(7 - x_1 - 5x_2 - 8x_3 + 2x_4 - 8x_5) \\
&= 47 + (23 - 5 - 6 - 3)x_1 + (19 - 1 - 3 - 15)x_2 + (28 - 6 + 9 - 24)x_3 \\
&+(14 - 3 - 15 + 5)x_4 + (44 - 5 - 18 - 24)x_5 \\
&= 47 + 9x_1 + 0x_2 + 7x_3 + x_4 - 3x_5.
\end{aligned}
$$

- Единственное ограничение — все переменные двоичные.

- Единственное ограничение — все переменные двоичные.
- Это означает, что если коэффициент в целевой функции положителен, то в оптимальном решении задачи Лагранжа переменная должна быть равна 1, а если коэффициент отрицательный, то переменная должно быть 0.

- Единственное ограничение — все переменные двоичные.
- Это означает, что если коэффициент в целевой функции положителен, то в оптимальном решении задачи Лагранжа переменная должна быть равна 1, а если коэффициент отрицательный, то переменная должно быть 0.
- Поскольку коэффициент при x_2 равен нулю, есть два оптимальных решения: (1,0,1,1,0) и (1,1,1,1,0).

- Единственное ограничение — все переменные двоичные.
- Это означает, что если коэффициент в целевой функции положителен, то в оптимальном решении задачи Лагранжа переменная должна быть равна 1, а если коэффициент отрицательный, то переменная должно быть 0.
- Поскольку коэффициент при x_2 равен нулю, есть два оптимальных решения: (1,0,1,1,0) и (1,1,1,1,0).
- Первый удовлетворяет условию оптимальности, поэтому является оптимальным решением.

- Единственное ограничение — все переменные двоичные.
- Это означает, что если коэффициент в целевой функции положителен, то в оптимальном решении задачи Лагранжа переменная должна быть равна 1, а если коэффициент отрицательный, то переменная должно быть 0.
- Поскольку коэффициент при x_2 равен нулю, есть два оптимальных решения: (1,0,1,1,0) и (1,1,1,1,0).
- Первый удовлетворяет условию оптимальности, поэтому является оптимальным решением.
- Второй вариант недопустим.

Что общего во всех релаксациях?

У них есть три общих свойства.

1. Все допустимые решения допустимы и в осслабленной задаче.

Что общего во всех релаксациях?

У них есть три общих свойства.

1. Все допустимые решения допустимы и в ослабленной задаче.
2. Оптимальное значение ослабленной задачи — это верхняя граница оптимального значения исходной задачи.

Что общего во всех релаксациях?

У них есть три общих свойства.

1. Все допустимые решения допустимы и в ослабленной задаче.
2. Оптимальное значение ослабленной задачи — это верхняя граница оптимального значения исходной задачи.
3. Бывают случаи, когда оптимальное решение релаксированной задачи оказывается оптимальным и в исходной.

Что общего во всех релаксациях?

У них есть три общих свойства.

1. Все допустимые решения допустимы и в ослабленной задаче.
2. Оптимальное значение ослабленной задачи — это верхняя граница оптимального значения исходной задачи.
3. Бывают случаи, когда оптимальное решение релаксированной задачи оказывается оптимальным и в исходной.

- Последнее свойство не может быть востребовано для всех частных случаев, так как тогда ослабленная задача является эквивалентной формой исходной и, скорее всего, требует примерно таких же вычислительных усилий, то есть не слишком помогает.

Что общего во всех релаксациях?

У них есть три общих свойства.

1. Все допустимые решения допустимы и в осслабленной задаче.
2. Оптимальное значение ослабленной задачи — это верхняя граница оптимального значения исходной задачи.
3. Бывают случаи, когда оптимальное решение релаксированной задачи оказывается оптимальным и в исходной.

- Последнее свойство не может быть востребовано для всех частных случаев, так как тогда ослабленная задача является эквивалентной формой исходной и, скорее всего, требует примерно таких же вычислительных усилий, то есть не слишком помогает.
- Следовательно, первые два свойства присутствуют в определении релаксации конкретной задачи.

Определение 3

Пусть f, h — две функции, отображающие n-мерное евклидово пространство в действительные числа. Далее пусть $\mathcal{U}, \mathcal{V}$ — два подмножества n-мерного евклидова пространства. Задача

$$\max\{h(\mathbf{x}) \mid \mathbf{x} \in \mathcal{V}\} \tag{25}$$

это ослабление задачи

$$\max\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{U}\} \tag{26}$$

если
(i) $\mathcal{U} \subset \mathcal{V}$ и
(ii) известно *априори*, т.е. без решения проблем, что $\nu(25) \geq \nu(26)$.

- Никакого точного определения понятия *класса задач* не будет.

Релаксация класса задач

- Никакого точного определения понятия *класса задач* не будет.
- В оптимизации есть много классов задач. Несколько примеров — это задача о рюкзаке, более общая булева оптимизация, задача коммивояжера, линейное программирование, выпуклое программирование и т. д.

Релаксация класса задач

- Никакого точного определения понятия *класса задач* не будет.
- В оптимизации есть много классов задач. Несколько примеров — это задача о рюкзаке, более общая булева оптимизация, задача коммивояжера, линейное программирование, выпуклое программирование и т. д.
- Далее под классом задач понимается только бесконечный набор задач.

Релаксация класса задач

- Никакого точного определения понятия *класса задач* не будет.
- В оптимизации есть много классов задач. Несколько примеров — это задача о рюкзаке, более общая булева оптимизация, задача коммивояжера, линейное программирование, выпуклое программирование и т. д.
- Далее под классом задач понимается только бесконечный набор задач.
- Одним из ключевых шагов в решении (6) было то, что задача была разделена на подзадачи, и даже подзадачи были разделены на дополнительные подзадачи, и так далее.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.
- При фиксировании значения переменной задача о рюкзаке становится еще одной задачей о рюкзаке меньшего размера.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.
- При фиксировании значения переменной задача о рюкзаке становится еще одной задачей о рюкзаке меньшего размера.
- То же самое верно почти для всех задач оптимизации, т. е. ограничение значения одной переменной (введение либо нижней границы, либо верхней границы, либо точного значения) создает новую задачу из того же класса.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.
- При фиксировании значения переменной задача о рюкзаке становится еще одной задачей о рюкзаке меньшего размера.
- То же самое верно почти для всех задач оптимизации, т. е. ограничение значения одной переменной (введение либо нижней границы, либо верхней границы, либо точного значения) создает новую задачу из того же класса.
- Но ограничение одной переменной — не единственный возможный способ разделить задачу на подзадачи.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.
- При фиксировании значения переменной задача о рюкзаке становится еще одной задачей о рюкзаке меньшего размера.
- То же самое верно почти для всех задач оптимизации, т. е. ограничение значения одной переменной (введение либо нижней границы, либо верхней границы, либо точного значения) создает новую задачу из того же класса.
- Но ограничение одной переменной — не единственный возможный способ разделить задачу на подзадачи.
- Иногда имеет смысл вводить специальные ограничения на набор переменных.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.
- При фиксировании значения переменной задача о рюкзаке становится еще одной задачей о рюкзаке меньшего размера.
- То же самое верно почти для всех задач оптимизации, т. е. ограничение значения одной переменной (введение либо нижней границы, либо верхней границы, либо точного значения) создает новую задачу из того же класса.
- Но ограничение одной переменной — не единственный возможный способ разделить задачу на подзадачи.
- Иногда имеет смысл вводить специальные ограничения на набор переменных.
- Например, из первого ограничения (19) легко увидеть, что не более двух из переменных x_1, x_3 и x_5 могут быть равны 1.

- Разделение должно выполняться таким образом, чтобы подзадачи принадлежали к одному классу задач.
- При фиксировании значения переменной задача о рюкзаке становится еще одной задачей о рюкзаке меньшего размера.
- То же самое верно почти для всех задач оптимизации, т. е. ограничение значения одной переменной (введение либо нижней границы, либо верхней границы, либо точного значения) создает новую задачу из того же класса.
- Но ограничение одной переменной — не единственный возможный способ разделить задачу на подзадачи.
- Иногда имеет смысл вводить специальные ограничения на набор переменных.
- Например, из первого ограничения (19) легко увидеть, что не более двух из переменных x_1, x_3 и x_5 могут быть равны 1.
- Таким образом, можно разделить задачу на две подзадачи, введя новое ограничение, для одной ветви $x_1 + x_3 + x_5 = 2$, а для другой $x_1 + x_3 + x_5 \leq 1$.

- Приведенные задачи все еще относятся к классу бинарной оптимизации.

- Приведенные задачи все еще относятся к классу бинарной оптимизации.
- То же самое не работает в случае задачи о ранце, поскольку у нее должно быть только одно ограничение, т.е. если к задаче добавляется второе неравенство, то новая задача выходит из класса задач о ранце.

- Приведенные задачи все еще относятся к классу бинарной оптимизации.
- То же самое не работает в случае задачи о ранце, поскольку у нее должно быть только одно ограничение, т.е. если к задаче добавляется второе неравенство, то новая задача выходит из класса задач о ранце.
- Разделение задачи на подзадачи означает, что множество возможных решений делится на подмножества, не исключая случая, когда одно или несколько подмножеств оказываются пустым множеством. $\mathcal{R}_5$ и $\mathcal{R}_6$ дают такой пример.

- Приведенные задачи все еще относятся к классу бинарной оптимизации.
- То же самое не работает в случае задачи о ранце, поскольку у нее должно быть только одно ограничение, т.е. если к задаче добавляется второе неравенство, то новая задача выходит из класса задач о ранце.
- Разделение задачи на подзадачи означает, что множество возможных решений делится на подмножества, не исключая случая, когда одно или несколько подмножеств оказываются пустым множеством. $\mathcal{R}_5$ и $\mathcal{R}_6$ дают такой пример.
- Еще одна важная особенность резюмируется в формуле (12). В ней говорится, что верхняя граница оптимального значения, полученного из неразделенной задачи, не более точна, чем верхняя оценка, полученная из разделенных задач.

- Приведенные задачи все еще относятся к классу бинарной оптимизации.
- То же самое не работает в случае задачи о ранце, поскольку у нее должно быть только одно ограничение, т.е. если к задаче добавляется второе неравенство, то новая задача выходит из класса задач о ранце.
- Разделение задачи на подзадачи означает, что множество возможных решений делится на подмножества, не исключая случая, когда одно или несколько подмножеств оказываются пустым множеством. $\mathcal{R}_5$ и $\mathcal{R}_6$ дают такой пример.
- Еще одна важная особенность резюмируется в формуле (12). В ней говорится, что верхняя граница оптимального значения, полученного из неразделенной задачи, не более точна, чем верхняя оценка, полученная из разделенных задач.
- Наконец, дальнейшее исследование подмножества $\mathcal{F}_1$ можно было бы прекратить, поскольку $\mathcal{R}_1$ не давал более высокую верхнюю границу, поскольку значение целевой функции оптимального решения на $\mathcal{R}_3$, который одновременно лежит и в $\mathcal{F}_3$, т.е. подзадача, определенная на множестве $\mathcal{F}_3$, решена.

Определение релаксации класса задач отражает тот факт, что релаксация и определение подзадач (ветвление) не являются полностью независимыми. В определении предполагается, что метод ветвления задан априори.

Определение релаксации класса задач отражает тот факт, что релаксация и определение подзадач (ветвление) не являются полностью независимыми. В определении предполагается, что метод ветвления задан априори.

Определение 4

Пусть $\mathcal{P}$ и $\mathcal{Q}$ — два класса задач. Класс $\mathcal{Q}$ является релаксацией класса $\mathcal{P}$ если существует отображение R со следующими свойствами.

1. R отображает задачи из $\mathcal{P}$ в задачи из $\mathcal{Q}$.

Определение релаксации класса задач отражает тот факт, что релаксация и определение подзадач (ветвление) не являются полностью независимыми. В определении предполагается, что метод ветвления задан априори.

Определение 4

Пусть $\mathcal{P}$ и $\mathcal{Q}$ — два класса задач. Класс $\mathcal{Q}$ является релаксацией класса $\mathcal{P}$ если существует отображение R со следующими свойствами.

1. R отображает задачи из $\mathcal{P}$ в задачи из $\mathcal{Q}$.
2. Если задача (P) $\in \mathcal{P}$ отображается в (Q) $\in \mathcal{Q}$ то (Q) является релаксацией задачи (P) в смысле определения 3.

Определение релаксации класса задач отражает тот факт, что релаксация и определение подзадач (ветвление) не являются полностью независимыми. В определении предполагается, что метод ветвления задан априори.

Определение 4

Пусть $\mathcal{P}$ и $\mathcal{Q}$ — два класса задач. Класс $\mathcal{Q}$ является релаксацией класса $\mathcal{P}$ если существует отображение R со следующими свойствами.

1. R отображает задачи из $\mathcal{P}$ в задачи из $\mathcal{Q}$.
2. Если задача $(\mathrm{P}) \in \mathcal{P}$ отображается в $(\mathrm{Q}) \in \mathcal{Q}$ то (Q) является релаксацией задачи (P) в смысле определения 3.
3. Если (P) делится на (P_1),…,(P_k) и эти задачи отображаются в (Q_1),…,(Q_k), то выполняется неравенство

$$\nu(\mathrm{Q}) \geq \max\{\nu(\mathrm{Q}_1), \ldots, \nu(\mathrm{Q}_k)\} \tag{27}$$

Определение релаксации класса задач отражает тот факт, что релаксация и определение подзадач (ветвление) не являются полностью независимыми. В определении предполагается, что метод ветвления задан априори.

Определение 4

Пусть $\mathcal{P}$ и $\mathcal{Q}$ — два класса задач. Класс $\mathcal{Q}$ является релаксацией класса $\mathcal{P}$ если существует отображение R со следующими свойствами.

1. R отображает задачи из $\mathcal{P}$ в задачи из $\mathcal{Q}$.
2. Если задача (P) $\in \mathcal{P}$ отображается в (Q) $\in \mathcal{Q}$ то (Q) является релаксацией задачи (P) в смысле определения 3.
3. Если (P) делится на $(\mathrm{P}_1),\ldots,(\mathrm{P}_k)$ и эти задачи отображаются в $(\mathrm{Q}_1),\ldots,(\mathrm{Q}_k)$, то выполняется неравенство

$$\nu(\mathrm{Q}) \geq \max\{\nu(\mathrm{Q}_1), \ldots, \nu(\mathrm{Q}_k)\} \tag{27}$$

4. Существует бесконечное множество пар (P), (Q) таких, что оптимальное решение (Q) также оптимально в (P).

Общая схема метода B&B

Метод B&B делит задачу на подзадачи и пытается исследовать каждую подзадачу с помощью релаксации. Подзадача далее не делится в одном из следующих случаев:

Общая схема метода B&B

Метод B&B делит задачу на подзадачи и пытается исследовать каждую подзадачу с помощью релаксации. Подзадача далее не делится в одном из следующих случаев:

1. Оптимальное решение релаксированной подзадачи удовлетворяет ограничениям нерелаксированной подзадачи, и значения ее релаксированной и нерелаксированной целевой функции равны.

Общая схема метода B&B

Метод B&B делит задачу на подзадачи и пытается исследовать каждую подзадачу с помощью релаксации. Подзадача далее не делится в одном из следующих случаев:

1. Оптимальное решение релаксированной подзадачи удовлетворяет ограничениям нерелаксированной подзадачи, и значения ее релаксированной и нерелаксированной целевой функции равны.
2. Невыполнимость ослабленной подзадачи означает, что неослабленная подзадача также недопустима.

Общая схема метода B&B

Метод B&B делит задачу на подзадачи и пытается исследовать каждую подзадачу с помощью релаксации. Подзадача далее не делится в одном из следующих случаев:

1. Оптимальное решение релаксированной подзадачи удовлетворяет ограничениям нерелаксированной подзадачи, и значения ее релаксированной и нерелаксированной целевой функции равны.
2. Невыполнимость ослабленной подзадачи означает, что неослабленная подзадача также недопустима.
3. Верхняя граница, обеспечиваемая ослабленной подзадачей, меньше (в случае поиска альтернативного оптимального решения) или меньше или равна (если ненужны альтернативные оптимальные решения), чем значение целевой функции наилуджшего известного допустимого решения.

Алгоритм может остановиться, если исследованы все подмножества (ветви). Если задача нелинейного программирования решаются методом B&B, то конечность алгоритма не всегда может быть гарантирована.

Алгоритм может остановиться, если исследованы все подмножества (ветви). Если задача нелинейного программирования решаются методом B&B, то конечность алгоритма не всегда может быть гарантирована.
В типичной итерации алгоритм выполняет следующие шаги.

- Выбирает лист дерева поиска, т.е. подзадачу еще не разделенную на подзадачи.

Алгоритм может остановиться, если исследованы все подмножества (ветви). Если задача нелинейного программирования решаются методом B&B, то конечность алгоритма не всегда может быть гарантирована.
В типичной итерации алгоритм выполняет следующие шаги.

- Выбирает лист дерева поиска, т.е. подзадачу еще не разделенную на подзадачи.
- Подзадача делится на дополнительные подзадачи (ветви) и определяются условия их отсечения.

Алгоритм может остановиться, если исследованы все подмножества (ветви). Если задача нелинейного программирования решаются методом B&B, то конечность алгоритма не всегда может быть гарантирована.
В типичной итерации алгоритм выполняет следующие шаги.

- Выбирает лист дерева поиска, т.е. подзадачу еще не разделенную на подзадачи.
- Подзадача делится на дополнительные подзадачи (ветви) и определяются условия их отсечения.
- Каждая новая ослабленная подзадача решается и проверяется, принадлежит ли она одному из вышеупомянутых случаев. Если да, то ветвь отсекается, и дальнейшее исследование не требуется. Если нет, то ветвь необходимо сохранить для дальнейшего разветвления.

Алгоритм может остановиться, если исследованы все подмножества (ветви). Если задача нелинейного программирования решаются методом B&B, то конечность алгоритма не всегда может быть гарантирована.
В типичной итерации алгоритм выполняет следующие шаги.

- Выбирает лист дерева поиска, т.е. подзадачу еще не разделенную на подзадачи.
- Подзадача делится на дополнительные подзадачи (ветви) и определяются условия их отсечения.
- Каждая новая ослабленная подзадача решается и проверяется, принадлежит ли она одному из вышеупомянутых случаев. Если да, то ветвь отсекается, и дальнейшее исследование не требуется. Если нет, то ветвь необходимо сохранить для дальнейшего разветвления.
- Если найдено новое допустимое решение, которое лучше, чем до сих пор лучшее, то даже сохраненные ветви, имеющие верхнюю границу меньше, чем значение нового наилучшего допустимого решения, могут быть удалены без дальнейшего исследования.

В дальнейшем предполагается, что релаксация удовлетворяет определению 4.
Исходная задача:

$$\max f(\mathbf{x}) \tag{14}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

В дальнейшем предполагается, что релаксация удовлетворяет определению 4.
Исходная задача:

$$\max\ f(\mathbf{x}) \tag{14}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

Таким образом, множество возможных решений есть

$$\mathcal{F} = \mathcal{F}_0 = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{b}; \mathbf{x} \in \mathcal{X}\}. \tag{28}$$

В дальнейшем предполагается, что релаксация удовлетворяет определению 4.
Исходная задача:

$$\max\ f(\mathbf{x}) \tag{14}$$

$$\mathbf{g}(\mathbf{x}) \le \mathbf{b} \tag{15}$$

$$\mathbf{x} \in \mathcal{X}. \tag{16}$$

Таким образом, множество возможных решений есть

$$\mathcal{F} = \mathcal{F}_0 = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \le \mathbf{b}; \mathbf{x} \in \mathcal{X}\}. \tag{28}$$

Ослабленная задача, удовлетворяющая требованиям определения 4, имеет вид

$$\max\ h(\mathbf{x})$$

$$\mathbf{k}(\mathbf{x}) \le \mathbf{b}$$

$$\mathbf{x} \in \mathcal{Y},$$

где $\mathcal{X} \subseteq \mathcal{Y}$ и для всех точек области определения целевых функций $f(\mathbf{x}) \le h(\mathbf{x})$ и для всех точек области определения функций ограничений $\mathbf{k}(\mathbf{x}) \le h(\mathbf{x})$.

- Таким образом, множество допустимых решений релаксации есть

$$\mathcal{R} = \mathcal{R}_0 = \{\mathbf{x} \mid \mathbf{k}(\mathbf{x}) \le \mathbf{b}; \mathbf{x} \in \mathcal{Y}\}.$$

- Таким образом, множество допустимых решений релаксации есть

$$\mathcal{R} = \mathcal{R}_0 = \{\mathbf{x} \mid \mathbf{k}(\mathbf{x}) \leq \mathbf{b}; \mathbf{x} \in \mathcal{Y}\}.$$

- Пусть $\mathcal{F}_k$ будет ранее определенным подмножеством допустимых решений булевой подзадачи.

- Таким образом, множество допустимых решений релаксации есть

$$\mathcal{R} = \mathcal{R}_0 = \{\mathbf{x} \mid \mathbf{k}(\mathbf{x}) \leq \mathbf{b}; \mathbf{x} \in \mathcal{Y}\}.$$

- Пусть $\mathcal{F}_k$ будет ранее определенным подмножеством допустимых решений булевой подзадачи.
- Предположим, что оно разбито на подмножества $\mathcal{F}_{t+1}, \ldots, \mathcal{F}_{t+p}$, т.е.

$$\mathcal{F}_k = \bigcup_{l=1}^{p} \mathcal{F}_{t+l}.$$

- Таким образом, множество допустимых решений релаксации есть

$$\mathcal{R} = \mathcal{R}_0 = \{\mathbf{x} \mid \mathbf{k}(\mathbf{x}) \leq \mathbf{b}; \mathbf{x} \in \mathcal{Y}\}.$$

- Пусть $\mathcal{F}_k$ будет ранее определенным подмножеством допустимых решений булевой подзадачи.
- Предположим, что оно разбито на подмножества $\mathcal{F}_{t+1}, \ldots, \mathcal{F}_{t+p}$, т.е.

$$\mathcal{F}_k = \bigcup_{l=1}^{p} \mathcal{F}_{t+l}.$$

- Пусть $\mathcal{R}_k$ and $\mathcal{R}_{t+1}, \ldots, \mathcal{R}_{t+p}$ — возможные наборы ослабленных подзадач.

- Таким образом, множество допустимых решений релаксации есть

$$\mathcal{R} = \mathcal{R}_0 = \{\mathbf{x} \mid \mathbf{k}(\mathbf{x}) \leq \mathbf{b}; \mathbf{x} \in \mathcal{Y}\}.$$

- Пусть $\mathcal{F}_k$ будет ранее определенным подмножеством допустимых решений булевой подзадачи.
- Предположим, что оно разбито на подмножества $\mathcal{F}_{t+1}, \ldots, \mathcal{F}_{t+p}$, т.е.

$$\mathcal{F}_k = \bigcup_{l=1}^{p} \mathcal{F}_{t+l}.$$

- Пусть $\mathcal{R}_k$ and $\mathcal{R}_{t+1}, \ldots, \mathcal{R}_{t+p}$ — возможные наборы ослабленных подзадач.
- Для удовлетворения требования (27) определения 4 предполагается, что

$$\mathcal{R}_k \supseteq \bigcup_{l=1}^{p} \mathcal{R}_{t+l}.$$

Подзадачи идентифицируются множеством допустимых решений. Неизвестные подзадачи хранятся в списке. Алгоритм выбирает из списка подзадачу для дальнейшего ветвления. При формальном описании общего метода B&B используются следующие обозначения.

Подзадачи идентифицируются множеством допустимых решений. Неизвестные подзадачи хранятся в списке. Алгоритм выбирает из списка подзадачу для дальнейшего ветвления. При формальном описании общего метода B&B используются следующие обозначения.

$\hat{z}$	значение целевой функции наилучшего допустимого решения, найденного на данный момент;
$\mathcal{L}$	список неизведанных подмножеств допустимых решений;
t	количество ветвей, созданных на данный момент;
$\mathcal{F}_0$	множество всех возможных решений;
r	индекс подмножества, выбранного для ветвления;
$p(r)$	количество ветвей, генерируемых из $\mathcal{F}_r$;
$\mathbf{x}_i$	оптимальное решение ослабленной подзадачи, определенной на $\mathcal{R}_i$;
z_i	верхняя граница целевой функции на подмножестве $\mathcal{F}_i$;
$\mathcal{L} + \mathcal{F}_i$	операция добавления подмножества $\mathcal{F}_i$ к списку $\mathcal{L}$;
$\mathcal{L} - \mathcal{F}_i$	операция удаления подмножества $\mathcal{F}_i$ из списка $\mathcal{L}$;

Обратите внимание, что $y_i = \max\{h(\mathbf{x}) \mid \mathbf{x} \in \mathcal{R}_i\}$.

Схема алгоритма дана ниже. Она просто описывает основные идеи метода и не содержит никаких инструментов ускорения.

Algorithm 1: метод ветвей и границ

```
ẑ ← −∞;
ℒ ← { ℱ_0 };
t ← 0;
while ℒ ≠ ∅ do
    определение r;
    ℒ ← ℒ − ℱ_r;
    определение p(r);
    определение ветвления ℱ_r ⊂ ℛ_1 ∪ ... ∪ ℛ_p(r);
    for i ← 1 to p(r) do
        ℱ_{t+i} ← ℱ_r ∩ ℛ_i;
        вычисление (x_{t+i}, z_{t+i});
        if z_{t+i} > ẑ then
            if x_{t+i} ∈ ℱ then
                ẑ ← z_{t+i};
            else
                ℒ ← ℒ + ℱ_{t+i};
    t ← t + p(r);
    for i ← 1 to t do
        if z_i ≤ ẑ then
            ℒ ← ℒ − ℱ_i;
return x
```

- Операции в строках 5, 7, 8 и 11 зависят от конкретного класса проблемы и навыков разработчика алгоритма.

- Операции в строках 5, 7, 8 и 11 зависят от конкретного класса проблемы и навыков разработчика алгоритма.
- Ослабленная подзадача решается в строке 14.

- Операции в строках 5, 7, 8 и 11 зависят от конкретного класса проблемы и навыков разработчика алгоритма.
- Ослабленная подзадача решается в строке 14.
- Подробный пример обсуждается в следующем разделе.

- Операции в строках 5, 7, 8 и 11 зависят от конкретного класса проблемы и навыков разработчика алгоритма.
- Ослабленная подзадача решается в строке 14.
- Подробный пример обсуждается в следующем разделе.
- Работа со списком также требует тщательного рассмотрения.

- Операции в строках 5, 7, 8 и 11 зависят от конкретного класса проблемы и навыков разработчика алгоритма.
- Ослабленная подзадача решается в строке 14.
- Подробный пример обсуждается в следующем разделе.
- Работа со списком также требует тщательного рассмотрения.
- Цикл в строках 17 и 18 может выполняться неявным образом. Если выбранная подзадача в строке 5 имеет нижнюю верхнюю границу, то есть $z_r \leq \hat{z}$, то подзадача исследована и выбирается новая подзадача.

- Однако наиболее важным вопросом является количество необходимых операций, включая конечность алгоритма.

- Однако наиболее важным вопросом является количество необходимых операций, включая конечность алгоритма.
- Метод не обязательно конечен. В частности при решении задачи нелинейного программирования может возникнуть бесконечное число шагов.

- Однако наиболее важным вопросом является количество необходимых операций, включая конечность алгоритма.
- Метод не обязательно конечен. В частности при решении задачи нелинейного программирования может возникнуть бесконечное число шагов.
- Бесконечный цикл может возникнуть даже в том случае, если число допустимых решений конечно. Это может быть вызвано «неосторожной» процедурой ветвления. Ветвь может принадлежать пустому множеству. Предположим, что процедура ветвления порождает подмножества из $\mathcal{F}_r$ такие, что одно из подмножеств $\mathcal{F}_{t+1}, \ldots, \mathcal{F}_{t+p(r)}$ равно $\mathcal{F}_r$, а остальные — пустые множества. Таким образом, существует индекс i такой, что

$$\mathcal{F}_{t+i} = \mathcal{F}_r, \quad \mathcal{F}_{t+1} = \ldots = \mathcal{F}_{t+i-1} = \mathcal{F}_{t+i+1} = \ldots = \mathcal{F}_{t+p(r)} = \emptyset. \quad (29)$$

Если такая же ситуация повторяется при разветвлении $\mathcal{F}_{t+i}$ то возможен бесконечный цикл.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.
- После первого ветвления имеется не более 2^{n-1} возможных решений в каждом из двух узлов первого уровня.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.
- После первого ветвления имеется не более 2^{n-1} возможных решений в каждом из двух узлов первого уровня.
- Это число уменьшается вдвое с каждым ветвлением, т.е. в ветви на уровне k имеется не более 2^{n-k} возможных решений.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.
- После первого ветвления имеется не более 2^{n-1} возможных решений в каждом из двух узлов первого уровня.
- Это число уменьшается вдвое с каждым ветвлением, т.е. в ветви на уровне k имеется не более 2^{n-k} возможных решений.
- Это означает, что на уровне n существует не более $2^{n-n} = 2^0 = 1$ допустимого решения.

- Предположим, что задача $0/1$-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.
- После первого ветвления имеется не более 2^{n-1} возможных решений в каждом из двух узлов первого уровня.
- Это число уменьшается вдвое с каждым ветвлением, т.е. в ветви на уровне k имеется не более 2^{n-k} возможных решений.
- Это означает, что на уровне n существует не более $2^{n-n} = 2^0 = 1$ допустимого решения.
- Фактически на этом уровне есть ровно один $0/1$-вектор, и можно решить, допустим он или нет.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.
- После первого ветвления имеется не более 2^{n-1} возможных решений в каждом из двух узлов первого уровня.
- Это число уменьшается вдвое с каждым ветвлением, т.е. в ветви на уровне k имеется не более 2^{n-k} возможных решений.
- Это означает, что на уровне n существует не более $2^{n-n} = 2^0 = 1$ допустимого решения.
- Фактически на этом уровне есть ровно один 0/1-вектор, и можно решить, допустим он или нет.
- Следовательно, после генерации всех ветвей на уровне n проблема может быть решена.

- Предположим, что задача 0/1-оптимизации с n переменными решается с помощью B&B, а ветвление всегда выполняется в соответствии с двумя значениями свободной переменной.
- Как правило, неизвестно, насколько велико количество возможных решений.
- Существует не более 2^n возможных решений, так как это количество векторов из нулей и единиц.
- После первого ветвления имеется не более 2^{n-1} возможных решений в каждом из двух узлов первого уровня.
- Это число уменьшается вдвое с каждым ветвлением, т.е. в ветви на уровне k имеется не более 2^{n-k} возможных решений.
- Это означает, что на уровне n существует не более $2^{n-n} = 2^0 = 1$ допустимого решения.
- Фактически на этом уровне есть ровно один 0/1-вектор, и можно решить, допустим он или нет.
- Следовательно, после генерации всех ветвей на уровне n проблема может быть решена.
- Эта идея обобщается в следующей теореме конечности.

Предположим, что
(i) Множество $\mathcal{F}$ конечно.
(ii) Существует конечное множество $\mathcal{U}$ такое, что выполняются следующие условия. Если в ходе метода ветвей и границ создается подмножество $\hat{\mathcal{F}}$, тогда существует подмножество $\hat{\mathcal{U}}$ из $\mathcal{U}$, такое что $\hat{\mathcal{F}} \subseteq \hat{\mathcal{U}}$. Кроме того, если процедура ветвления создает оболочку $\mathcal{R}_1 \cup \ldots \cup \mathcal{R}_p \supseteq \hat{\mathcal{F}}$, тогда $\hat{\mathcal{U}}$ имеет такое разбиение, что

$$\hat{\mathcal{U}} = \hat{\mathcal{U}}_1 \cup \cdots \cup \hat{\mathcal{U}}_p,\ \hat{\mathcal{U}}_i \cap \hat{\mathcal{U}}_j = \emptyset (i \neq j)$$
$$\hat{\mathcal{F}} \cap \hat{\mathcal{R}} j \subseteq \hat{\mathcal{U}}_j (j = 1, \ldots, p)$$

И кроме того

$$1 \leq \mid \hat{\mathcal{U}}_j \mid < \mid \hat{\mathcal{U}} \mid \quad (j = 1, \ldots, p). \tag{30}$$

(iii) Если множество $\hat{\mathcal{U}}$, принадлежащий множеству $\hat{\mathcal{F}}$, имеет только один элемент, тогда ослабленная подзадача также решает нерелаксированную подзадачу.
Метод *ветвей и границ* останавливается после конечного числа шагов. Если $\hat{z} = -\infty$, то допустимого решения нет. В противном случае $\hat{z}$ равно оптимальному значению целевой функции.

Замечание. Отметим, что упомянутые выше задачи типа $\hat{\mathcal{U}}_j$, определяются так

$$\hat{\mathcal{U}}_j = \{\mathbf{x} \in \{0,1\}^n \mid x_k = \delta_{kj},\ k \in I_j\},$$

где $I_j \subset \{1, 2, \dots, n\}$ — множество фиксированных переменных и $\delta_{kj} \in \{0, 1\}$ — фиксированное значение, удовлетворяющие условиям теоремы.

Замечание. Отметим, что упомянутые выше задачи типа $\hat{\mathcal{U}}_j$, определяются так

$$\hat{\mathcal{U}}_j = \{\mathbf{x} \in \{0,1\}^n \mid x_k = \delta_{kj},\ k \in I_j\},$$

где $I_j \subset \{1, 2, \ldots, n\}$ — множество фиксированных переменных и $\delta_{kj} \in \{0, 1\}$ — фиксированное значение, удовлетворяющие условиям теоремы.

- Предположим, что процедура *Branch-and-Bound* выполняет бесконечно много шагов.

Замечание. Отметим, что упомянутые выше задачи типа $\hat{\mathcal{U}}_j$, определяются так

$$\hat{\mathcal{U}}_j = \{\mathbf{x} \in \{0,1\}^n \mid x_k = \delta_{kj},\ k \in I_j\},$$

где $I_j \subset \{1, 2, \ldots, n\}$ — множество фиксированных переменных и $\delta_{kj} \in \{0,1\}$ — фиксированное значение, удовлетворяющие условиям теоремы.

- Предположим, что процедура *Branch-and-Bound* выполняет бесконечно много шагов.
- Поскольку множество $\mathcal{F}$ конечно, следует, что существует по крайней мере одно подмножество $\mathcal{F}$ скажим $\mathcal{F}_r$ такое, что оно определяет бесконечное множество ветвей, подразумевая, что описанная ситуация в (29) встречается бесконечно много раз.

Замечание. Отметим, что упомянутые выше задачи типа $\hat{\mathcal{U}}_j$, определяются так

$$\hat{\mathcal{U}}_j = \{\mathbf{x} \in \{0,1\}^n \mid x_k = \delta_{kj},\ k \in I_j\},$$

где $I_j \subset \{1, 2, \ldots, n\}$ — множество фиксированных переменных и $\delta_{kj} \in \{0, 1\}$ — фиксированное значение, удовлетворяющие условиям теоремы.

- Предположим, что процедура *Branch-and-Bound* выполняет бесконечно много шагов.
- Поскольку множество $\mathcal{F}$ конечно, следует, что существует по крайней мере одно подмножество $\mathcal{F}$ скажим $\mathcal{F}_r$ такое, что оно определяет бесконечное множество ветвей, подразумевая, что описанная ситуация в (29) встречается бесконечно много раз.
- Следовательно, существует бесконечная последовательность индексов, скажем $r_0 = r < r_1 < \cdots$, такая, что $\mathcal{F}_{r_{j+1}}$ создается при разветвлении $\mathcal{F}_{r_j}$ и $\mathcal{F}_{r_{j+1}} = \mathcal{F}_{r_j}$.

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$| \mathcal{U}_{r_0} | > | \mathcal{U}_{r_1} | > \cdots \geq 1.$$

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$|\mathcal{U}_{r_0}| > |\mathcal{U}_{r_1}| > \cdots \geq 1.$$

- Это невозможно, потому что $\mathcal{U}$ — конечные множества.

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$| \mathcal{U}_{r_0} | > | \mathcal{U}_{r_1} | > \cdots \geq 1.$$

- Это невозможно, потому что $\mathcal{U}$ — конечные множества.
- Конечность $\mathcal{F}$ означает, что оптимальное решение существует тогда и только тогда, когда $\mathcal{F}$ непусто, т.е. задача не может быть неограниченной, и если допустимое решение существует, то верхняя грань целевой функции является ее максимумом.

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$| \mathcal{U}_{r_0} |>| \mathcal{U}_{r_1} |> \cdots \geq 1.$$

- Это невозможно, потому что $\mathcal{U}$ — конечные множества.
- Конечность $\mathcal{F}$ означает, что оптимальное решение существует тогда и только тогда, когда $\mathcal{F}$ непусто, т.е. задача не может быть неограниченной, и если допустимое решение существует, то верхняя грань целевой функции является ее максимумом.
- Начальное значение $\hat{z}$ равно $-\infty$.

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$|\mathcal{U}_{r_0}| > |\mathcal{U}_{r_1}| > \cdots \geq 1.$$

- Это невозможно, потому что $\mathcal{U}$ — конечные множества.
- Конечность $\mathcal{F}$ означает, что оптимальное решение существует тогда и только тогда, когда $\mathcal{F}$ непусто, т.е. задача не может быть неограниченной, и если допустимое решение существует, то верхняя грань целевой функции является ее максимумом.
- Начальное значение $\hat{z}$ равно $-\infty$.
- Его можно изменить только в строке 18 алгоритма, и если его изменить, то оно будет равно значению целевой функции допустимого решения.

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$|\mathcal{U}_{r_0}| > |\mathcal{U}_{r_1}| > \cdots \geq 1.$$

- Это невозможно, потому что $\mathcal{U}$ — конечные множества.
- Конечность $\mathcal{F}$ означает, что оптимальное решение существует тогда и только тогда, когда $\mathcal{F}$ непусто, т.е. задача не может быть неограниченной, и если допустимое решение существует, то верхняя грань целевой функции является ее максимумом.
- Начальное значение $\hat{z}$ равно $-\infty$.
- Его можно изменить только в строке 18 алгоритма, и если его изменить, то оно будет равно значению целевой функции допустимого решения.
- Таким образом, если допустимого решения нет, то оно остается $-\infty$.

- С другой стороны, параллельная последовательность $\mathcal{U}$ множеств должна удовлетворять неравенствам

$$|\,\mathcal{U}_{r_0}\,|>|\,\mathcal{U}_{r_1}\,|>\cdots\geq 1.$$

- Это невозможно, потому что $\mathcal{U}$ — конечные множества.
- Конечность $\mathcal{F}$ означает, что оптимальное решение существует тогда и только тогда, когда $\mathcal{F}$ непусто, т.е. задача не может быть неограниченной, и если допустимое решение существует, то верхняя грань целевой функции является ее максимумом.
- Начальное значение $\hat{z}$ равно $-\infty$.
- Его можно изменить только в строке 18 алгоритма, и если его изменить, то оно будет равно значению целевой функции допустимого решения.
- Таким образом, если допустимого решения нет, то оно остается $-\infty$.
- Следовательно, если вторая половина утверждения неверна, то в конце алгоритма $\hat{z}$ равно значению целевой функции неоптимального допустимого решения или остается $-\infty$.

- Пусть r — максимальный индекс такой, что $\mathcal{F}_r$ по-прежнему содержит оптимальное решение.

- Пусть r — максимальный индекс такой, что $\mathcal{F}_r$ по-прежнему содержит оптимальное решение.
- Тогда

$$z_r \geq \text{оптимальное значение} > \hat{z}.$$

- Пусть r — максимальный индекс такой, что $\mathcal{F}_r$ по-прежнему содержит оптимальное решение.
- Тогда
$$z_r \geq \text{оптимальное значение} > \hat{z}.$$
- Следовательно, невозможно, чтобы ветвь, содержащая оптимальное решение, была удалена из списка в цикле строк 22 и 23, так как $z_r > \hat{z}$.

- Пусть r — максимальный индекс такой, что $\mathcal{F}_r$ по-прежнему содержит оптимальное решение.
- Тогда
$$z_r \geq \text{оптимальное значение} > \hat{z}.$$
- Следовательно, невозможно, чтобы ветвь, содержащая оптимальное решение, была удалена из списка в цикле строк 22 и 23, так как $z_r > \hat{z}$.
- Также очевидно, что подзадача
$$\max\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{F}_r\}$$
не была решена, иначе ограничение $z_r = \hat{z}$ должно выполняться.

- Пусть r — максимальный индекс такой, что $\mathcal{F}_r$ по-прежнему содержит оптимальное решение.
- Тогда
$$z_r \geq \text{оптимальное значение} > \hat{z}.$$
- Следовательно, невозможно, чтобы ветвь, содержащая оптимальное решение, была удалена из списка в цикле строк 22 и 23, так как $z_r > \hat{z}$.
- Также очевидно, что подзадача
$$\max\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{F}_r\}$$
не была решена, иначе ограничение $z_r = \hat{z}$ должно выполняться.
- Тогда остается только одна возможность, что $\mathcal{F}_r$ был выбран для ветвления один раз в ходе работы алгоритма.

- Пусть r — максимальный индекс такой, что $\mathcal{F}_r$ по-прежнему содержит оптимальное решение.
- Тогда
$$z_r \geq \text{оптимальное значение} > \hat{z}.$$
- Следовательно, невозможно, чтобы ветвь, содержащая оптимальное решение, была удалена из списка в цикле строк 22 и 23, так как $z_r > \hat{z}$.
- Также очевидно, что подзадача
$$\max\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{F}_r\}$$
не была решена, иначе ограничение $z_r = \hat{z}$ должно выполняться.
- Тогда остается только одна возможность, что $\mathcal{F}_r$ был выбран для ветвления один раз в ходе работы алгоритма.
- Оптимальное решение должно содержаться в одном из его подмножеств, скажем, $\mathcal{F}_{t+i}$, что противоречит предположению, что $\mathcal{F}_r$ имеет наивысший индекс среди ветвей, содержащих оптимальное решение.

- Если задача оптимизации содержит только ограниченные целочисленные переменные, то множества $\mathcal{U}$ являются множествами целочисленных векторов в определенных блоках.

- Если задача оптимизации содержит только ограниченные целочисленные переменные, то множества $\mathcal{U}$ являются множествами целочисленных векторов в определенных блоках.
- В случае некоторых задач планирования, где должен быть определен оптимальный порядок задач, даже релаксации имеют комбинаторную природу, потому что они состоят из перестановок.

- Если задача оптимизации содержит только ограниченные целочисленные переменные, то множества $\mathcal{U}$ являются множествами целочисленных векторов в определенных блоках.
- В случае некоторых задач планирования, где должен быть определен оптимальный порядок задач, даже релаксации имеют комбинаторную природу, потому что они состоят из перестановок.
- Тогда тоже возможно $\mathcal{U} = \mathcal{R}$. В обоих случаях естественным образом выполняется условие (iii) теоремы.

Смешанное целочисленное программирование с ограниченными переменными

- Многие решения имеют как непрерывный, так и дискретный характер.

- Многие решения имеют как непрерывный, так и дискретный характер.
- Например, при производстве электроэнергии дискретное решение — включить или выключить оборудование.

- Многие решения имеют как непрерывный, так и дискретный характер.
- Например, при производстве электроэнергии дискретное решение — включить или выключить оборудование.
- Оборудование может производить электрическую энергию в относительно широком диапазоне.

- Многие решения имеют как непрерывный, так и дискретный характер.
- Например, при производстве электроэнергии дискретное решение — включить или выключить оборудование.
- Оборудование может производить электрическую энергию в относительно широком диапазоне.
- Таким образом, если первое решение — включить, то второе решение должно быть принято на уровне произведенной энергии.

- Многие решения имеют как непрерывный, так и дискретный характер.
- Например, при производстве электроэнергии дискретное решение — включить или выключить оборудование.
- Оборудование может производить электрическую энергию в относительно широком диапазоне.
- Таким образом, если первое решение — включить, то второе решение должно быть принято на уровне произведенной энергии.
- Это постоянное решение.

- Многие решения имеют как непрерывный, так и дискретный характер.
- Например, при производстве электроэнергии дискретное решение — включить или выключить оборудование.
- Оборудование может производить электрическую энергию в относительно широком диапазоне.
- Таким образом, если первое решение — включить, то второе решение должно быть принято на уровне произведенной энергии.
- Это постоянное решение.
- Правильная математическая модель таких задач должна содержать как дискретные, так и непрерывные переменные.

- Этот раздел посвящен задаче смешанного целочисленного линейного программирования с ограниченными целочисленными переменными.

- Этот раздел посвящен задаче смешанного целочисленного линейного программирования с ограниченными целочисленными переменными.
- Предполагается, что существует n переменных и их подмножество, скажем, $\mathcal{I} \subseteq \{1, \ldots, n\}$ должнны быть целыми числами.

- Этот раздел посвящен задаче смешанного целочисленного линейного программирования с ограниченными целочисленными переменными.
- Предполагается, что существует n переменных и их подмножество, скажем, $\mathcal{I} \subseteq \{1, \ldots, n\}$ должнны быть целыми числами.
- Модель имеет m линейных ограничений в форме уравнения, и каждая целочисленная переменная имеет явную целочисленную верхнюю границу.

- Этот раздел посвящен задаче смешанного целочисленного линейного программирования с ограниченными целочисленными переменными.
- Предполагается, что существует n переменных и их подмножество, скажем, $\mathcal{I} \subseteq \{1, \ldots, n\}$ должнны быть целыми числами.
- Модель имеет m линейных ограничений в форме уравнения, и каждая целочисленная переменная имеет явную целочисленную верхнюю границу.
- Также предполагается, что все переменные должны быть неотрицательными.

- Этот раздел посвящен задаче смешанного целочисленного линейного программирования с ограниченными целочисленными переменными.
- Предполагается, что существует n переменных и их подмножество, скажем, $\mathcal{I} \subseteq \{1, \ldots, n\}$ должнны быть целыми числами.
- Модель имеет m линейных ограничений в форме уравнения, и каждая целочисленная переменная имеет явную целочисленную верхнюю границу.
- Также предполагается, что все переменные должны быть неотрицательными.
- Более формально математическая задача состоит в следующем.

$$\max \mathbf{c}^T \mathbf{x} \tag{31}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{32}$$

$$\forall\, j \in \mathcal{I}: \;\; x_j \leq g_j \tag{33}$$

$$x_j \geq 0 \;\; j = 1, \ldots, n \tag{34}$$

$$\forall\, j \in \mathcal{I}: \;\; x_j \;\; \text{— целые} \;, \tag{35}$$

где $\mathbf{c}$ и $\mathbf{x}$ — n-мерные векторы, $\mathbf{A}$ — матрица $m \times n$, $\mathbf{b}$ — m-мерный вектор и, наконец, все g_j $(j \in \mathcal{I})$ — натуральное число.

В математическом анализе задачи, приведенном ниже, явные ограничения на верхнюю границу (33) использоваться не будут. Читатель может считать, что они формально включены в другие алгебраические ограничения (32).

- Есть технические причины, по которым алгебраические ограничения в (32) заявлены в форме уравнений.

- Есть технические причины, по которым алгебраические ограничения в (32) заявлены в форме уравнений.
- В методе используется релаксация линейного программирования.

- Есть технические причины, по которым алгебраические ограничения в (32) заявлены в форме уравнений.
- В методе используется релаксация линейного программирования.
- Задача линейного программирования решается симплекс-методом, которому нужна эта форма.

- Есть технические причины, по которым алгебраические ограничения в (32) заявлены в форме уравнений.
- В методе используется релаксация линейного программирования.
- Задача линейного программирования решается симплекс-методом, которому нужна эта форма.
- Но, вообще говоря, уравнения и неравенства можно преобразовать одно в другое эквивалентным образом.

- Есть технические причины, по которым алгебраические ограничения в (32) заявлены в форме уравнений.
- В методе используется релаксация линейного программирования.
- Задача линейного программирования решается симплекс-методом, которому нужна эта форма.
- Но, вообще говоря, уравнения и неравенства можно преобразовать одно в другое эквивалентным образом.
- Даже в числовом примере, обсуждаемом ниже, используется форма неравенств.

Сначала анализируется числовой пример. Ход метода обсуждается с геометрической точки зрения. При этом некоторые технические детали остаются скрытыми. Обсуждается симплекс-метод и связанные с ним темы. Все технические детали могут быть описаны только в их владении. Наконец, анализируются некоторые стратегические моменты алгоритма.

- Решаемая задача:

$$\begin{array}{llcrcrcr} \max & x_0 & = & 2x_1 & + & x_2 & & \\ & & & 3x_1 & - & 5x_2 & \leq & 0 \\ & & & 3x_1 & + & 5x_2 & \leq & 15 \\ & & & & x_1,\, x_2 \geq 0 & & & \\ & & & & x_1,\, x_2 \text{ — целые.} & & & \end{array} \tag{36}$$

Геометрический анализ числового примера

- Решаемая задача:

$$\begin{array}{lllllll} \max & x_0 & = & 2x_1 & + & x_2 & & \\ & & & 3x_1 & - & 5x_2 & \leq & 0 \\ & & & 3x_1 & + & 5x_2 & \leq & 15 \\ \end{array} \tag{36}$$
$$x_1,\, x_2 \geq 0$$
$$x_1,\, x_2 \text{ — целые.}$$

- Для получения релаксации в задаче опускаются ограничения целостности. Таким образом, получается задача линейного программирования с двумя переменными.

Геометрический анализ числового примера

- Решаемая задача:

$$
\begin{array}{rrcrcrcr}
\max & x_0 & = & 2x_1 & + & x_2 & & \\
 & & & 3x_1 & - & 5x_2 & \leq & 0 \\
 & & & 3x_1 & + & 5x_2 & \leq & 15 \\
\end{array}
\qquad (36)
$$
$$x_1,\, x_2 \geq 0$$
$$x_1,\, x_2 \text{ — целые.}$$

- Для получения релаксации в задаче опускаются ограничения целостности. Таким образом, получается задача линейного программирования с двумя переменными.
- Ветвление осуществляется по нецелой переменной. И x_1, и x_2 имеют дробные значения.

- Решаемая задача:

$$
\begin{array}{lclcrcl}
\max \; x_0 & = & 2x_1 & + & x_2 & & \\
& & 3x_1 & - & 5x_2 & \le & 0 \\
& & 3x_1 & + & 5x_2 & \le & 15 \\
& & & x_1,\, x_2 \ge 0 & & & \\
& & & x_1,\, x_2 \text{ — целые.} & & &
\end{array}
\tag{36}
$$

- Для получения релаксации в задаче опускаются ограничения целостности. Таким образом, получается задача линейного программирования с двумя переменными.
- Ветвление осуществляется по нецелой переменной. И x_1, и x_2 имеют дробные значения.
- Чтобы количество ветвей было как можно меньше, за один шаг будем создать только две новые ветви.

- Нумерация ветвей следующая.

- Нумерация ветвей следующая.
- Первоначальный набор возможных решений — № 1.

- Нумерация ветвей следующая.
- Первоначальный набор возможных решений — № 1.
- Когда генерируются две новые ветви, то ветвь, принадлежащая меньшим значениям переменной ветвления, имеет меньший номер.

- Нумерация ветвей следующая.
- Первоначальный набор возможных решений — № 1.
- Когда генерируются две новые ветви, то ветвь, принадлежащая меньшим значениям переменной ветвления, имеет меньший номер.
- Ветви, не имеющие допустимого решения, также нумеруются.

- Оптимальное решение релаксации: $x_1 = 2,5$, $x_2 = 1,5$, а оптимальное значение — $\frac{13}{2}$, как видно из рисунка 1.

- Оптимальное решение релаксации: $x_1 = 2,5$, $x_2 = 1,5$, а оптимальное значение — $\frac{13}{2}$, как видно из рисунка 1.
- Оптимальным решением является точка пересечения прямых, определяемых уравнениями

$$3x_1 - 5x_2 = 0$$

и

$$3x_1 + 5x_2 = 15.$$

- Оптимальное решение релаксации: $x_1 = 2,5$, $x_2 = 1,5$, а оптимальное значение — $\frac{13}{2}$, как видно из рисунка 1.
- Оптимальным решением является точка пересечения прямых, определяемых уравнениями

$$3x_1 - 5x_2 = 0$$

и

$$3x_1 + 5x_2 = 15.$$

- Если ветвление основано на переменной x_1, то они определяются неравенствами

$$x_1 \leq 2 \quad \text{и} \quad x_1 \geq 3\ .$$

- Оптимальное решение релаксации: $x_1 = 2,5$, $x_2 = 1,5$, а оптимальное значение — $\frac{13}{2}$, как видно из рисунка 1.
- Оптимальным решением является точка пересечения прямых, определяемых уравнениями

$$3x_1 - 5x_2 = 0$$

  и

$$3x_1 + 5x_2 = 15.$$

- Если ветвление основано на переменной x_1, то они определяются неравенствами

$$x_1 \leq 2 \quad \text{и} \quad x_1 \geq 3\ .$$

- Обратите внимание, что оптимальное значение x_1 равно 2,5.

- Оптимальное решение релаксации: $x_1 = 2,5$, $x_2 = 1,5$, а оптимальное значение — $\frac{13}{2}$, как видно из рисунка 1.
- Оптимальным решением является точка пересечения прямых, определяемых уравнениями

$$3x_1 - 5x_2 = 0$$

и

$$3x_1 + 5x_2 = 15.$$

- Если ветвление основано на переменной x_1, то они определяются неравенствами

$$x_1 \leq 2 \quad \text{и} \quad x_1 \geq 3\ .$$

- Обратите внимание, что оптимальное значение x_1 равно 2,5.
- Переменная x_2 будет создавать ветви

$$x_2 \leq 1 \quad \text{и} \quad x_2 \geq 2\ .$$

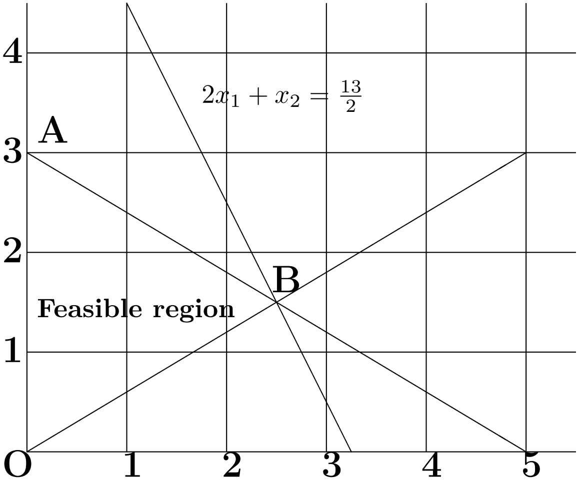


Рис.: Решение релаксации линейного программирования задачи (36). Допустимая область (треугольник $\overline{\text{OAB}}$), оптимальное решение ($x_1 = 2.5$, $x_2 = 1.5$), и оптимальное значение целевой функции, представленной линией $2x_1 + x_2 = \frac{13}{2}$.

Ни одна из них не соответствует пустому множеству. Таким образом, выгоднее ветка по x_1. Геометрически это означает, что множество допустимых решений релаксированной задачи обрезается прямой $x_1 = 2$. Таким образом, новый набор становится четырехугольником $\overline{OACD}$ на рисунке 2. Оптимальное решение на этом множестве — $x_1 = 2$, $x_2 = 1.8$. Это точка C на рисунке.

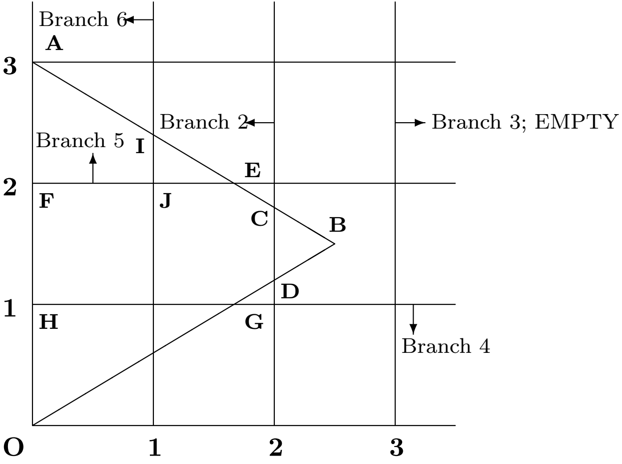


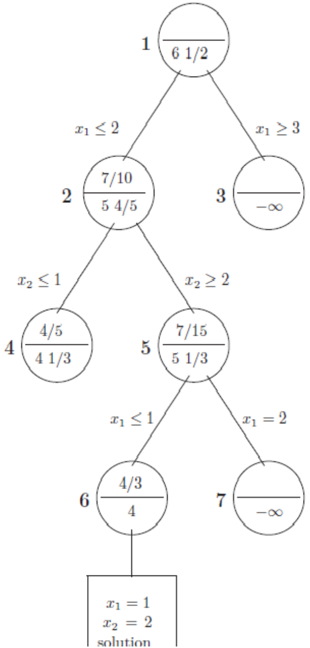
Рис.: Геометрия хода решения. Координаты точек: O=(0,0), A=(0,3), B=(2.5,1.5), C=(2,1.8), D=(2,1.2), E=($\frac{5}{3}$,2), F=(0,2), G=($\frac{5}{3}$,1), H=(0,1), I=(1,2.4), и J=(1,2). Возможные области релаксации следующие. Ветвь 1: $\overline{OAB}$, Ветвь 2: $\overline{OACD}$, Ветвь 3: пустое множество, Ветвь 4: $\overline{OHG}$, Ветвь 5: $\overline{AEF}$, Ветвь 6: $\overline{AIJF}$, Теперь ветвление возможно только по переменной x_2. Ветви 4 и 5 порождаются разрезами $x_2 \leq 1$ и $x_2 \geq 2$ соответственно. Возможными областями ослабленных задач являются $\overline{OHG}$ ветви 4 и $\overline{AEF}$ ветви 5. Метод продолжает исследование ветви 5. Причина будет указана в следующем подразделе, когда обсуждаются быстро вычисляемые верхние границы. С другой стороны, очевидно, что набор $\overline{AEF}$ более перспективен, чем $\overline{OHG}$, если Читатель учитывает положение контура, то есть линии уровня целевой функции на Рис. 2. Алгебраические детали, обсуждаемые в следующем подразделе, служат для реализации решений в более высоких измерениях, которые можно увидеть в 2-мерном случае.

Ветви 6 и 7 определяются неравенствами $x_1 \leq 1$ и $x_1 \geq 2$ соответственно. Последний снова пуст. Возможная область ветви 6 — $\overline{AIJF}$. Оптимальное решение в этом четырехугольнике — точка I. Обратите внимание, что в $\overline{AIJF}$ есть только три целых точки: (0,3), (0,2) и (1,2). Таким образом, оптимальное целочисленное решение этой ветви — (1,2). Есть методика, которая может помочь преодолеть непрерывный оптимум. В этом случае она достигает непосредственно точки J, то есть оптимального целочисленного решения ветви, как это также будет показано в следующем разделе. Прямо сейчас предположим, что целочисленное оптимальное решение со значением целевой функции 4 обнаружено.

На данном этапе алгоритма единственная неопознанная ветвь — это ветвь 4 с допустимой областью $\overline{OHG}$. Очевидно, что оптимальным решением является точка $G = (\frac{5}{3}, 1)$. Значение целевой функции равно $\frac{13}{3}$. Таким образом, область не может содержать более лудшего решения, чем известное (1,2). Таким образом, алгоритм останавливается.

Первый в истории общий метод решения задач линейного программирования был открыт Джорджем Данцигом и назван *симплекс-методом.* Существует множество вариантов симплекс-метода. Основным является так называемый *двойственный симплекс-метод.* Хотя симплекс-метод обсуждается в предыдущем томе, здесь обобщены основные знания.

Любой вид симплекс-метода — это так называемый алгоритм вращения. *Важным свойством алгоритмов вращения является то, что они генерируют эквивалентные формы системы уравнений и — в случае линейного программирования — целевой функции.* Практически это означает, что алгоритм работает с уравнениями. Столько переменных, сколько существует линейно независимых уравнений, выражается с помощью других переменных, а дальнейшие следствия выводятся из текущей эквивалентной формы уравнений.


1
6 1/2
$x_1 \leq 2$
$x_1 \geq 3$
2
7/10
5 4/5
3
$-\infty$
$x_2 \leq 1$
$x_2 \geq 2$
4
4/5
4 1/3
5
7/15
5 1/3
$x_1 \leq 1$
$x_1 = 2$
6
4/3
4
7
$-\infty$
$x_1 = 1$
$x_2 = 2$
solution

Ход решения задачи (36). Верхние числа в схемах объяснены в подразделе **??**. Это исправления предыдущих оценок, полученных на первом шаге поворота симплекс-метода. Нижние числа — это (непрерывные) верхние оценки, полученные в ветви.

Если в задаче есть неравенства, то они переформулируются путем введения неотрицательных переменных резервов. Например, в случае LP-релаксации задачи (36) эквивалентная форма задачи имеет вид

$$\begin{array}{lllllllllllll} \max & x_0 & = & 2x_1 & + & x_2 & + & 0x_3 & + & 0x_4 & & \\ & & & 3x_1 & - & 5x_2 & + & x_3 & + & 0x_4 & = & 0 \\ & & & 3x_1 & + & 5x_2 & + & 0x_3 & + & x_4 & = & 15 \\ & & & \multicolumn{8}{l}{x_1,\, x_2\, x_3,\, x_4 \geq 0.} \end{array} \qquad (37)$$

Обратите внимание, что все переменные присутствуют во всех уравнениях, включая целевую функцию, но допускается, чтобы некоторые коэффициенты были нулевыми. Текущая версия (37) может рассматриваться как форма, в которой переменные x_3 и x_4 выражаются через x_1 и x_2, а выражение подставляется в целевую функцию.

Если $x_1 = x_2 = 0$, то $x_3 = 0$ и $x_4 = 15$, поэтому решение возможно. Обратите внимание, что значение целевой функции равно 0, и если возможно увеличить значение любого из x_1 и x_2 и по-прежнему получить допустимое решение, тогда будет получено лучшее возможное решение. Это правда, потому что в методе используются эквивалентные формы целевой функции. Метод позволяет получить лучшее возможное решение за счет поворота. Пусть x_1 и x_2 - две выраженные переменные. Если пропустить некоторые этапы поворота, эквивалентная форма (37) будет

$$\begin{array}{llcccccccccc}
\max & x_0 & = & 0x_1 & + & 0x_2 & - & \frac{7}{30}x_3 & - & \frac{13}{30}x_4 & + & \frac{13}{2} \\
 & & & x_1 & + & 0x_2 & + & \frac{1}{6}x_3 & + & \frac{1}{6}x_4 & = & \frac{5}{2} \\
 & & & 0x_1 & + & x_2 & - & \frac{1}{10}x_3 & + & \frac{1}{10}x_4 & = & \frac{3}{2} \\
 & & & \multicolumn{9}{l}{x_1,\ x_2\, x_3,\ x_4 \geq 0.}
\end{array} \tag{38}$$

On the enumeration tree

О дереве поиска

Одна из критических точек B& B — хранение дерева поиска. Когда ветвь опознана, тогда даже некоторые из ее предков могут стать полностью опознанными при условии, что текущая ветвь была последней неизведанной подветвью предков. Предки также сохраняются, иначе восстановить наследника невозможно. Поскольку B&B гибко использует дерево поиска, может возникнуть необходимость хранить большой объем информации в ветвях. Это может вызвать проблемы с памятью. С другой стороны, было бы слишком дорого с точки зрения вычислений проверять предков каждый раз, если ветвь становится понятной. В этом разделе даются некоторые идеи, как найти компромисс.

В первую очередь необходимо решить, какие данные описывают ветку. Есть два варианта. Во-первых, вся необходимая информация хранится по каждой ветке. Он включает в себя все определяющие ограничения ветвления. В этом случае одно и то же ограничение сохраняется много раз, потому что ветвь на более высоком уровне может иметь много дочерних ветвей. На самом деле количество ветвей очень велико в случае крупномасштабных проблем, поэтому для этого решения требуется очень большой объем памяти.

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,

По техническим причинам также используются три других атрибута:

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,
- граница целевой функции на ветви,

По техническим причинам также используются три других атрибута:

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,
- граница целевой функции на ветви,
- индекс переменной ветвления,

По техническим причинам также используются три других атрибута:

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,
- граница целевой функции на ветви,
- индекс переменной ветвления,
- ветвление, определяющее ограничение переменной ветвления.

По техническим причинам также используются три других атрибута:

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,
- граница целевой функции на ветви,
- индекс переменной ветвления,
- ветвление, определяющее ограничение переменной ветвления.

По техническим причинам также используются три других атрибута:

- логическая переменная, показывающая, разбита ли ветвь на подветви,

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,
- граница целевой функции на ветви,
- индекс переменной ветвления,
- ветвление, определяющее ограничение переменной ветвления.

По техническим причинам также используются три других атрибута:

- логическая переменная, показывающая, разбита ли ветвь на подветви,
- другая логическая переменная, показывающая, существует ли какая-либо неизвестная подветвь ветви,

Другой вариант заключается в том, что сохраняется только та информация, которая необходима для полной реконструкции ветви. Например такая:

- родительская ветвь, т.е. ветвь, из которой она была создана напрямую,
- граница целевой функции на ветви,
- индекс переменной ветвления,
- ветвление, определяющее ограничение переменной ветвления.

По техническим причинам также используются три других атрибута:

- логическая переменная, показывающая, разбита ли ветвь на подветви,
- другая логическая переменная, показывающая, существует ли какая-либо неизвестная подветвь ветви,
- и указатель на следующий элемент в списке ветвей.

Таким образом, ветвь может быть описана **записью** следующего вида:

$$
\begin{array}{l}
record\ Branch \\
begin \\
\quad \begin{array}{ll}
Parent & : Branch; \\
Bound & : integer; \\
Variable & : integer; \\
Value & : integer; \\
Decomposition & : Boolean; \\
Descendant & : Boolean; \\
suc & : Branch;
\end{array} \\
end;
\end{array}
$$

Значение атрибута Parent равно **none** тогда и только тогда, когда ветвь является начальной ветвью, то есть исходной задачей. Это корень B&B-дерева. Реконструкция ограничений, определяющих конкретную ветвь, является наиболее простой, если предполагается, что ветви определяются фиксацией свободной переменной. Предположим, что Node — это переменная типа Branch. Вначале ее значение — это реконструируемая ветвь. Тогда алгоритм реконструкции следующий.

```
Branch-Reconstruction
1 while Node ≠ none
2       do x[Node.Variable] ← Node.Value;
3             . . .
4             Node ← Node.Parent;
5 return Node
```

Значение ранее фиксированной переменной устанавливается равным соответствующему значению в строке 2. Возможны дальнейшие операции (строка 4). Узел становится своей родительской ветвью в строке 5. Если это **none**, то корень передается и все исправления выполнены.

Иногда необходимо выполнить какие-то операции над всеми элементами списка $\mathcal{L}$. Атрибут успеха ветвей указывает на следующий элемент списка. Последний элемент не имеет следующего, поэтому в этом случае значение success равно **none**. Процедура замены всех элементов в чем-то похожа на процедуру *Branch Reconstruction*. Заголовком списка $\mathcal{L}$ является Tree, т.е. первым элементом списка является Tree.suc.

B&B-List

```
Node ← Tree.suc
while Node ≠ none
    . . .
    Node ← Node.suc
return Node
```

Цикл выполняется до тех пор, пока не останется следующего элемента. Необходимые операции выполняются в строке 4. Переменная Node становится следующим элементом списка в строке 5. Вставить новую ветку в список несложно. Предположим, что это NewNode типа Branch, и он должен быть вставлен после узла, который находится в списке. Тогда необходимы две команды:

NewNode.suc $\leftarrow$ Node.suc

Node.suc $\leftarrow$ NewNode

Если ветки не хранятся как объекты, а описаны в длинных массивах, то использование атрибута success излишне, и вместо процедуры **B&B List** может применяться цикл **for**.

Самая большая техническая проблема B&B с точки зрения информатики — это управление памятью. Поскольку веток создается в огромном количестве, исходные ветки должны время от времени удаляться из списка, а занимаемая ими память должна быть освобождена. Это своего рода сборка мусора. Это можно сделать в три основных этапа. В первом атрибуте Descendant всех элементов списка присваивается значение **false**. На втором основном шаге атрибут Descendant изменяется на **true** тогда и только тогда, когда ветвь имеет неизвестных потомков. На третьем шаге удаляются ненужные ветки. Предполагается, что существует процедура Out, которая удаляет ветвь как параметр, удаляет ее и освобождает часть памяти.

```
Garbage-Collection
Node ← Tree.suc
while Node ≠ none
      Node.Descendant ← False
      Node ← Node.suc
Node ← Tree.suc
while Node ≠ none
      do if not Node.Decomposition and Node.Bound > ẑ
      then Pont ← Node.Parent
      while Pont ≠ none do
          Pont.Descendant ← True
          Pont ← Pont.Parent
       Node ← Node.suc
Node ← Tree.suc
while Node ≠ none do
       Pont ← Node.suc
       if (not Node.Descendant and Node.Decomposition) or
Node.Bound ≤ ẑ
      then Out(Node)
       Node ← Pont
return ???
```

The use of information obtained from other sources

Использование информации, полученной из других источников

Этот метод можно ускорить, используя информацию, предоставляемую дополнительными алгоритмическими инструментами.

Применение эвристических методов

Целью применения эвристических методов является получение допустимых решений. С теоретической точки зрения решить, существует ли какое-либо допустимое решение, тоже NP-трудно. С другой стороны, эвристика может дать допустимые решения в случае большинства численных задач. Применяемые методы зависят от характера рассматриваемой задачи, т.е. для чисто двоичных, ограниченных целочисленных и смешанных целочисленных задач могут потребоваться различные методы. Например, для чисто целочисленных задач могут хорошо работать локальный поиск и множители Лагранжа. Множители Лагранжа также обеспечивают верхнюю границу (в случае максимизации) оптимального значения.

Если допустимое решение известно, то сразу можно не принимать во внимание ветви, основанные на их границах. См. Строку 15 алгоритма Branch and Bound. Там автоматически удаляются ветви с недостаточно хорошими границами. В случае чисто бинарной задачи явное ограничение целевой функции также может дать множество последствий.

Предварительная обработка
Предварительная обработка означает получение информации о переменных и ограничениях на основе алгебраических ограничений и целостности.
Например, если суммировать два ограничения задачи (36), то неравенство

$$6x_1 \leq 15$$

получается, что $x_1 \leq 2$.

Пусть

$$g_i(x) \leq b_i \tag{39}$$

быть одним из ограничений задачи (14)-(16).
Многие тесты могут быть основаны на следующих двух простых наблюдениях:

1. Если максимальное значение левой части (39) $x \in \mathcal{X}$ не больше правой части 39), то ограничение является избыточным.

Пусть

$$g_i(x) \leq b_i \tag{39}$$

быть одним из ограничений задачи (14)-(16).
Многие тесты могут быть основаны на следующих двух простых наблюдениях:

1. Если максимальное значение левой части (39) $x \in \mathcal{X}$ не больше правой части 39), то ограничение является избыточным.
2. Если минимальное значение левой части (39), если $x \in \mathcal{X}$ больше правой части (39), то это невозможно чтобы удовлетворить ограничению, т.е. проблема (14) - (16) не имеет допустимого решения.

Если при некотором дополнительном ограничении второе наблюдение верно, то рассматриваемое ограничение может быть исключено. Типичным примером является то, что определенные переменные должны иметь максимально / минимально возможное значение. Таким образом можно исправить переменную или уменьшить ее диапазон.

Релаксацию Лагранжа можно также использовать для исправления некоторых переменных. Предположим, что оптимальное значение задачи (22) и (16) равно $\nu(L(\lambda \mid x_j = \delta))$ при дополнительном условии, что x_j должен принимать значение δ. Если $\hat{z}$ — значение целевой функции известного допустимого решения и $\hat{z} > \nu(L(\lambda \mid x_j = \delta))$, тогда x_j не может принимать значение δ. Дальнейшие методы предполагают, что проблема LP-релаксации решена, и на основе оптимальных двойных цен пытаются зафиксировать значения переменных.