

МЕТОДЫ И ТЕХНОЛОГИИ ПРИНЯТИЯ РЕШЕНИЙ

УДК 519.711.3

DOI:10.18287/2223-9537-2021-11-2-227-238

Оптимизация при ограничении числа проектных переменных

В.П. Офицеров¹, С.В. Смирнов²¹ *Московский авиационный институт (национальный исследовательский университет), Москва, Россия*² *Самарский федеральный исследовательский центр РАН, Институт проблем управления сложными системами РАН, Самара, Россия*

Аннотация

Изложен новый подход к постановке и решению оптимизационных задач линейного и нелинейного типа. От классической задачи линейного программирования оптимального распределения ограниченных ресурсов между заданными процессами, рассматриваемая постановка задачи отличается необходимостью выбора ограниченного числа процессов из некоторого конечного множества и распределения ресурсов по этим процессам. Целью является получение оптимального значения целевой функции по отношению к другим вариантам выбора числа процессов из этого же множества и распределению ресурсов между ними. Целевая функция может быть как линейной, так и нелинейной. Нелинейная функция должна обладать определёнными свойствами для корректной работы предложенного алгоритма поиска оптимального решения. Описываемый метод основан на развитии идей динамического программирования Беллмана. Приводятся доказательства оптимальности получаемых решений. В статье даётся оценка вычислительной сложности алгоритма и сравнение с классическими методами решения рассматриваемых задач. Охарактеризованы типы прикладных задач, решаемых с использованием предложенного метода. Компьютерные реализации описанного алгоритма могут использоваться в автоматизированных системах поддержки принятия решений.

Ключевые слова: оптимизация, динамическое программирование, линейное и нелинейное программирование, поддержка принятия решений.

Цитирование: Офицеров, В.П. Оптимизация при ограничении числа проектных переменных / В.П. Офицеров, С.В. Смирнов // Онтология проектирования. – 2021. – Т.11, №2(40). – С.227-238. – DOI:10.18287/2223-9537-2021-11-2-227-238.

Введение

Для производственных, транспортных, вычислительных и других систем важно эффективное функционирование подсистем планирования и принятия решений. Обычно на вход подобных подсистем поступают сведения о множестве ресурсов, таких как люди, материалы, оборудование, устройства, капитал, время, и множестве операций (поручений, требований), которые с помощью данных ресурсов должны быть выполнены; их выходом служит распределение ресурсов по операциям. Под улучшением характеристик рассматриваемых подсистем понимают повышение эффективности использования ресурсов, а именно уменьшение потребности в ресурсах без соответствующего изменения в объёме, стоимости и прибыли или увеличение прибыли при сохранении прежней потребности в ресурсах [1-3].

В статье представлен алгоритм решения одного класса задач комбинаторной оптимизации, возникающих при решении дискретных задач линейного и нелинейного программирования с концептуально важным дополнительным ограничением на число используемых в решении переменных [4-10]. Алгоритм является глубокой модификацией метода динамиче-

ского программирования [11, 12]. В отличие от классического метода определения функции Беллмана, в предложенном алгоритме на первом этапе находятся экстремальные значения целевой функции при использовании только одной переменной, определяемой в процессе поиска экстремума на каждом допустимом значении ресурсов. На последующих этапах используется рекурсия поиска экстремальных значений целевой функции путём пошагового увеличения числа переменных, используемых в решении.

Большинство задач комбинаторной оптимизации являются NP-трудными [8, 13, 14], что обуславливает сложность построения эффективных алгоритмов их решения. Для многих задач хорошо известные классические методы решения (динамическое программирование, методы ветвей и границ, «жадные» алгоритмы) оказываются неэффективными, что обуславливает необходимость разработки новых методов.

1 Постановка и решение задачи линейного программирования

Большое число задач можно сформулировать в терминах линейного программирования [4-6, 15]. Многие из них приобретают новое прикладное значение, если рассматривать задачу отыскания экстремума линейной функции n переменных при использовании в решении d^* переменных, где $d^* \leq n$. Математически задача формулируется так: найти

$$\max (c_1x_1 + \dots + c_nx_n) \quad (1)$$

при ограничениях

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &\leq b^*_1 \\ &\dots \\ a_{m1}x_1 + \dots + a_{mn}x_n &\leq b^*_m \\ a_{ji} > 0, \quad i = 1, \dots, n, j = 1, \dots, m, \end{aligned} \quad (2)$$

$$\sum_{i=1}^n \tilde{x}_i \leq d^*, \quad (3)$$

где

$$x_i \geq 0, \quad d^* \leq n, \quad \tilde{x}_i = \begin{cases} 1, & x_i > 0, \\ 0, & x_i = 0. \end{cases}$$

От традиционной постановки задачи линейного программирования [4, 5] рассматриваемую задачу отличает наличие ограничений (3), которые определяют задачу выбора не более d^* переменных (процессов), между которыми необходимо разделить ограниченные ресурсы, определяемые ограничениями (2) при условии нахождения максимума целевой функции (1).

Одним из примеров такого класса задач является выбор d^* типов изделий для производства. Пусть для создания некоторого типа изделий требуется m видов ресурсов и по условиям производства можно выпускать d^* типов изделий, но при этом есть возможность выбора типов из n возможных ($n \geq d^*$). Расход ресурса номера j при единичном выпуске изделия типа i обозначен через a_{ji} . Через c_i обозначен доход от продажи единицы i -го типа продукции, а через x_i – количество продукции i -го типа. Зависимость расходов и доходов от количества выпускаемых изделий i -го типа принята линейной, а производству доступно b_j единиц j -го ресурса. В этом случае задача выбора оптимальной номенклатуры типов изделий и объёмов их выпуска имеет вид (1)–(3), если требуется максимизировать доход.

При $d^* = n$ задача сводится к обычной задаче линейного программирования [4-6].

Требование целочисленности решения не несёт принципиальных изменений предлагаемого подхода, считается, что переменные x_i могут принимать только целые значения, когда это необходимо.

Используя приём динамического программирования [13, 14], рассматривается множество задач типа (1)–(3) при всевозможных значениях b_j и d ($0 \leq b_j \leq b^*_j, j = 1, \dots, m; 1 \leq d \leq d^*$).

Тогда при $d = 1$ решается задача отыскания максимума (1) по одной переменной при всех $0 \leq b_j \leq b_j^*, j = 1, \dots, m$, что можно записать так:

$$f_1(b_1, \dots, b_m) = \max_{1 \leq i \leq n} c_i x_i \quad (4)$$

при ограничениях для каждого выбираемого значения x_i и всевозможных значений b_j

$$x_i \geq 0, 0 \leq b_j \leq b_j^*, j = 1, \dots, m. \quad (5)$$

На практике каждое b_j , как правило, принимает конечное множество значений, образующее некоторую ε_j -сеть на $[0, b_j^*]$.

Найденные значения максимумов при использовании в решении одной переменной из n при всевозможных допустимых значениях выделяемых ресурсов $\Delta b_1, \dots, \Delta b_m$ обозначены

$$f_1(\Delta b_1, \dots, \Delta b_m), \quad (6)$$

где $0 \leq \Delta b_j \leq b_j, 0 \leq b_j \leq b_j^*, j = 1, \dots, m$. Здесь $\Delta b_j, j = 1, \dots, m$ – это всевозможные значения ресурсов, на которых найдены максимумы в соответствии с (4) и (5). Необходимо заметить, что для каждого набора $\Delta b_j, j = 1, \dots, m$ максимальное значение $f_1(\Delta b_1, \dots, \Delta b_m)$ может давать любая из n возможных переменных.

Для получения максимума в (1) при использовании в решении не более двух переменных, т.е. при $d = 2$, необходимо найти максимальные значения от одной переменной для всех возможных допустимых значений ресурса и просуммировать их с максимальными значениями целевой функции также от одной переменной, найденных на остающихся после выделения первой переменной ресурсах. Доказательство этого утверждения может быть получено следующим образом. В рамках динамического подхода рекуррентное выражение для поиска максимума при использовании двух переменных запишется так:

$$f_2(b_1, \dots, b_m) = \max [c_i x_i + f_1(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)], 1 \leq i \leq n, \quad (7)$$

при следующих ограничениях для каждого возможного значения x_i и значений b_j :

$$a_{ji} x_i \leq \Delta b_j, 0 \leq \Delta b_j \leq b_j, j = 1, \dots, m. \quad (8)$$

В (7) $f_1(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)$ – найденное на первом шаге алгоритма максимальное значение функции от одной переменной на конкретном значении выделенных ресурсов, которое складывается со значением $c_i x_i$, где i выбирается из множества $\{1, \dots, n\}$ и определяется также на конкретном (в момент выбора) наборе значений ресурсов $\Delta b_j, j = 1, \dots, m$.

Для получения максимума двух слагаемых, одно из которых имеет фиксированное значение при выбранных на текущий момент значениях $\Delta b_j, j = 1, \dots, m$, нужно выбрать переменную с номером i , которая даст максимальное значение $c_i x_i$ на выделенном ресурсе $\Delta b_j, j = 1, \dots, m$. На первом шаге алгоритма такое значение уже было определено и записано в (6). Вариант с возможным совпадением номеров переменных даёт значения $f_1(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)$ и $f_1(\Delta b_1, \dots, \Delta b_m)$. Следовательно, выражения (7) и (8) можно записать так:

$$f_2(b_1, \dots, b_m) = \max [f_1(\Delta b_1, \dots, \Delta b_m) + f_1(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)], \quad (9)$$

при выполнении ограничений:

$$0 \leq \Delta b_j \leq b_j, 0 \leq b_j \leq b_j^*, j = 1, \dots, m.$$

Максимальное значение целевой функции не более чем от двух переменных на ресурсах b_1, \dots, b_m получено перебором сумм максимальных значений двух функций, каждая из которых использует только одну из n возможных переменных, определённых на первом шаге алгоритма для всевозможных значений ресурсов.

Если вернуться к исходным обозначениям, выражение (9) после вычисления $f_2(b_1, \dots, b_m)$ можно записать так:

$$f_2(b_1, \dots, b_m) = \max (c_k x_k + c_l x_l),$$

где максимальное значение суммы получено после выполнения полного перебора вариантов возможных распределений ресурсов между двумя переменными. При этом на каждом варианте распределения ресурса выбирались две переменные, дающие максимальное значение целевой функции на выделенных им ресурсах. Найденные номера переменных, их значения и соответствующие им ресурсы обозначены так: номера - k и l , значения переменных - x_k и x_l , значение целевой функции - $c_k x_k + c_l x_l$, ограничения - $a_{jk} x_k \leq \Delta b_j$, $a_{jl} x_l \leq b_j - \Delta b_j$, $j = 1, \dots, m$.

Можно предположить, что существуют переменные с номерами y и z со значениями x_y и x_z , которые на тех же выделенных ресурсах дают значение больше значения $f_2(b_1, \dots, b_m)$. Это означает, что либо на ресурсе Δb_j , либо на ресурсе $b_j - \Delta b_j$ ($j = 1, \dots, m$) выполняется хотя бы одно из неравенств

$$\begin{aligned} c_y x_y &> c_k x_k, \\ c_z x_z &> c_l x_l, \end{aligned}$$

что невозможно в силу определения значений $c_k x_k$ и $c_l x_l$ как максимальных на этих ресурсах.

Аналогичные рассуждения можно провести для любого возможного распределения ресурсов между двумя любыми переменными. Это означает, что если в исходной постановке задачи не добавляется условие (3) выбора переменных, участвующих в решении задачи, т.е. если предложенным методом решается классическая задача линейного программирования с двумя переменными в целевой функции, то будет найдено глобальное оптимальное решение.

Таким образом, доказано, что вычисленное значение $f_2(b_1, \dots, b_m)$ является глобальным максимумом на всех допустимых значениях b_1, \dots, b_m .

Продолжая процесс, можно получить рекуррентное уравнение относительно числа используемых в решении переменных:

$$f_d(b_1, \dots, b_m) = \max [f_1(\Delta b_1, \dots, \Delta b_m) + f_{d-1}(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)], \quad (10)$$

для всех $0 \leq \Delta b_j \leq b_j$, $0 \leq b_j \leq b_j^*$, $j = 1, \dots, m$, $d = 2, \dots, d^*$.

Используя метод математической индукции и свойства линейных операций, можно доказать, что для линейной целевой функции и линейных ограничений значение $f_d(b_1, \dots, b_m)$ является глобальным максимумом для всех допустимых b_1, \dots, b_m .

При использовании в решении только одной переменной ($d = 1$) значения $f_1(b_1, \dots, b_m)$ являются максимальными для допустимых значений b_1, \dots, b_m (в том числе и для значений $\Delta b_1, \dots, \Delta b_m$, используемых в рекурсивном выражении (10)). Вычисленное значение $f_2(b_1, \dots, b_m)$ является глобальным максимумом на всех допустимых значениях b_1, \dots, b_m . Можно предположить, что выражение (10) даёт максимальное значение при использовании в решении не более $(d - 1)$ переменных. Оно даёт максимальное значение и при использовании в решении не более d переменных. Действительно, в выражении (10) значения f_{d-1} по допущению являются максимальными на всех допустимых значениях ресурсов, а значения $f_1(\Delta b_1, \dots, \Delta b_m)$ являются максимальными в силу алгоритма их вычисления. Пусть значение максимума $f_d(b_1, \dots, b_m)$ получено суммированием

$$S_1(\Delta b_1, \dots, \Delta b_m) + f_{d-1}(b_1 - \Delta b_1, \dots, b_m - \Delta b_m),$$

где $S_1(\Delta b_1, \dots, \Delta b_m)$ - некоторое значение целевой функции при использовании в решении какой-то одной переменной из n возможных. По определению значение $f_1(\Delta b_1, \dots, \Delta b_m)$ является максимальным на ресурсе $\Delta b_1, \dots, \Delta b_m$. Это означает, что $S_1(\Delta b_1, \dots, \Delta b_m) = f_1(\Delta b_1, \dots, \Delta b_m)$, иначе значение максимума $f_d(b_1, \dots, b_m)$ не было бы получено. Таким образом, доказано, что вычисленное значение $f_d(b_1, \dots, b_m)$ является глобальным максимумом на всех допустимых значениях b_1, \dots, b_m .

Если на любом шаге d значение $f_d(b_1, \dots, b_m)$ получается при повторном использовании хотя бы одной из n переменных (т.е., некоторая переменная с номером i входит в оба слагаемых соотношения (10)), то

$$f_d(b_1, \dots, b_m) = f_{d-1}(b_1, \dots, b_m). \quad (11)$$

Действительно, пусть

$$f_d(b_1, \dots, b_m) = \max [f_1^d(\Delta b_1^d, \dots, \Delta b_m^d) + f_{d-1}(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)], \quad 0 \leq \Delta b_j \leq b_j,$$

а число использованных переменных для получения f_d не увеличилось. Это означает, что некоторая переменная x_i , которой соответствует $f_1^d(\Delta b_1^d, \dots, \Delta b_m^d)$, была использована на одном из предыдущих шагов при получении $f_{d-1}(b_1 - \Delta b_1, \dots, b_m - \Delta b_m)$. Согласно (4)-(10), значение целевой функции формируется таким образом, что будет справедливо выражение

$$f_d(b_1, \dots, b_m) = f_1^d(\Delta b_1^d, \dots, \Delta b_m^d) + f_1^k(\Delta b_1^k, \dots, \Delta b_m^k) + f_{d-2}(b_1 - \Delta b_1^d - \Delta b_1^k, \dots, b_m - \Delta b_m^d - \Delta b_m^k),$$

где $f_1^d(\Delta b_1^d, \dots, \Delta b_m^d)$ и $f_1^k(\Delta b_1^k, \dots, \Delta b_m^k)$ по предположению, получены при использовании одной и той же переменной i со значениями x_i^d и x_i^k .

В силу линейности функций и ограничений можно показать, что для x_i^d и x_i^k справедливо равенство

$$f_1^d(\Delta b_1^d, \dots, \Delta b_m^d) + f_1^k(\Delta b_1^k, \dots, \Delta b_m^k) = f_1(\Delta b_1^d + \Delta b_1^k, \dots, \Delta b_m^d + \Delta b_m^k).$$

Действительно, это соответствует свойству линейных операций $c_i x_i^d + c_i x_i^k = c_i x_i$, где $x_i = x_i^d + x_i^k$, и преобразованию ограничений для этой переменной

$$a_{ji} x_i^d \leq \Delta b_j^d, \quad a_{ji} x_i^k \leq \Delta b_j^k, \quad j = 1, \dots, m,$$

в ограничения

$$a_{ji} x_i^d + a_{ji} x_i^k \leq \Delta b_j^d + \Delta b_j^k, \quad j = 1, \dots, m,$$

или, что то же самое, в ограничения

$$a_{ji} x_i \leq \Delta b_j^d + \Delta b_j^k, \quad j = 1, \dots, m,$$

где значения $\Delta b_j^d + \Delta b_j^k$ для переменной с номером i рассмотрены на первом шаге алгоритма, и соответствующее значение $c_i x_i$ использовано для формирования значения $f_{d-1}(b_1, \dots, b_m)$ на шаге $(d-1)$. Отсюда следует, что найденное повторное включение переменной i в целевую функцию можно записать так

$$\begin{aligned} c_i(x_i^d + c_i x_i^k) + f_{d-2}(b_1 - \Delta b_1^d - \Delta b_1^k, \dots, b_m - \Delta b_m^d - \Delta b_m^k) = \\ = c_i x_i + f_{d-2}(b_1 - \Delta b_1^i, \dots, b_m - \Delta b_m^i) = f_{d-1}(b_1, \dots, b_m). \end{aligned}$$

Здесь $\Delta b_1^i = \Delta b_1^d + \Delta b_1^k, \dots, \Delta b_m^i = \Delta b_m^d + \Delta b_m^k$; $c_i x_i = c_i(x_i^d + x_i^k)$ - линейные ограничения для x_i^d и x_i^k складываются и дают ограничение для x_i .

Таким образом, показано отсутствие нарушения структуры целевой функции при попытке повторного включения в решение уже использованной переменной. Поэтому если на шаге d значение $f_d(b_1, \dots, b_m)$ увеличилось, то оно получено при использовании d переменных, так как в противном случае это увеличение произошло бы на одном из предыдущих шагов (имеется в виду минимальное число используемых переменных).

Для определения фактических номеров, числа и значений переменных, на которых достигается $f_d(b_1, \dots, b_m)$, можно использовать соотношение (10), двигаясь по шагам в обратном направлении и определяя:

- на шаге d - значение $f_1^d(\Delta b_1^d, \dots, \Delta b_m^d)$, равное $f_1(\Delta b_1, \dots, \Delta b_m)$ в (10) при получении $f_d(b_1, \dots, b_m)$, и соответствующие номер и значение переменной;
- на шаге $(d-1)$ - значение $f_1^{d-1}(\Delta b_1^{d-1}, \dots, \Delta b_m^{d-1})$, дающее $f_{d-1}(b_1 - \Delta b_1^d, \dots, b_m - \Delta b_m^d)$ и соответствующие номер и значение переменной.

Продолжая этот пошаговый процесс, можно получить конкретные номера переменных и их значения, определяющие максимальное значение целевой функции.

В классическом представлении функция Беллмана определяется последовательно по пронумерованным переменным [11]. Когда ресурс выделяется переменной i , то предполагается, что оставшийся ресурс оптимально распределён по уже рассмотренным $(i-1)$ упорядоченным по номерам переменным. Задача будет решена, когда будут получены результаты

для последней переменной в списке. В результате ограничение на число используемых в решении переменных не будет учтено. Чтобы его учесть при использовании классического рекуррентного уравнения Беллмана [13] придётся решить $C_n^{d^*}$ задач, последовательно перебирая варианты наборов d^* переменных из n возможных, и выбрать из этих решений наилучшее. Такой же подход (с перебором вариантов) можно использовать и при применении других традиционных методов решения задач линейного программирования с дополнительным ограничением на число используемых переменных.

Для иллюстрации работы предложенного метода рассмотрена задача:

$$\text{найти } \mathbf{max} (x_1 + 4x_2 + 6x_3 + 9x_4)$$

при ограничениях

$$\begin{aligned} x_1 + 3x_2 + 5x_3 + 7x_4 &\leq 10, \\ x_i &\in \{0, 1, 2, \dots\}, i = 1, \dots, 4; \sum_{i=1}^4 \tilde{x}_i \leq 2, \tilde{x}_i = \begin{cases} 1, & x_i > 0, \\ 0, & x_i = 0. \end{cases} \end{aligned}$$

Данные, получаемые в результате вычислений по (4)-(10) для возможных значений $b_1 = 1, 2, \dots, 10$, объединены в таблице 1. В примере используется только один тип ресурса, который обозначен как b_1 .

На первом шаге получены значения

$$f_1(b_1) = \mathbf{max}_{1 \leq i \leq 4} c_i x_i$$

для всех b_1 при выполнении ограничений для каждого возможного значения ресурса. Результаты вычислений f_1 показаны во второй строке таблицы. Каждый результат в отдельном столбце соответствует:

- значению ограничения на ресурс, показанному в первой строке таблицы;
- номеру i выбранной по алгоритму вычисления $f_1(b_1)$ переменной, показанному в третьей строке;
- значению переменной x_i , вычисленному при условии выполнения ограничения на ресурс, показанному в четвёртой строке.

На втором шаге алгоритма для каждого возможного значения ресурса вычисляется значение целевой функции от двух переменных по формуле (10), при этом используются результаты вычислений первого шага:

$$f_2(b_1) = \mathbf{max} [f_1(\Delta b_1) + f_1(b_1 - \Delta b_1)], 0 \leq \Delta b_1 \leq b_1,$$

при $a_i x_i \leq \Delta b_1; i = 1, 2, \dots, 4; b_1 \leq b_1^*, b_1^* = 10; b_1 \in \{1, 2, \dots, 10\}$.

Для ресурса, принимающего значения от 1 до 3, значения целевой функции получаются при использовании в решении только одной переменной и совпадают с результатами первого шага. Эти результаты показаны в пятой строке таблицы 1, обозначенной как « $f_2 = f_1^2 + f_1^1$ ».

При значении ресурса $b_1 = 4$ после просмотра возможных значений $f_1(\Delta b_1)$ при $\Delta b_1 \in \{1, 2, 3, 4\}$ получено максимальное значение $f_2(4) = f_1(1) + f_1(4 - 1) = 1 + 4 = 5$ при $x_1 = 1, x_2 = 1$. Эти результаты показаны в строках 5-8 таблицы 1 и столбце со значением $b_1 = 4$. В строке, обозначенной « $f_1^2 + f_1^1$ », показаны значения $f_1(\Delta b_1) + f_1(b_1 - \Delta b_1)$, на которых получен максимум; в строке, обозначенной « i, j », показаны номера переменных, формирующих оптимальное значение, а в строке 8, обозначенной как « x_i, x_j », значения соответствующих переменных.

При значении ресурса $b_1 = 5$ на втором шаге получено оптимальное значение при двух вариантах использования ресурса. Первым получается значение целевой функции равно 6 при $f_1(\Delta b_1 = 0) = 0$ и $f_1(5 - 0) = 6$ при $x_3 = 1$. Этот же результат получается при $f_1(\Delta b_1 = 2) = 2$ и $f_1(5 - 2) = 4$ при $x_1 = 2, x_2 = 1$:

$$f_2(5) = f_1(0) + f_1(5 - 0) = 6 = f_1(2) + f_1(5 - 2) = 2 + 4 = 6.$$

Так как в задаче не требуется обязательного использования двух переменных в решении, то в таблице записан первый результат. Продолжая процесс для значения ресурса $b_1 = 10$, значение целевой функции получается равное 13, когда $f_2(10) = f_1(1) + f_1(9) = 13$ при $x_1 = 1, x_2 = 3$.

В данном примере вычисление значений $f_2(b_1)$ для всех возможных значений $b_1 \in \{1, 2, \dots, 10\}$ было избыточным. Достаточно было сразу рассмотреть вариант с $b_1 = 10$, так как третьего шага в случае поиска оптимального решения при использовании трёх переменных нет. Если бы в задаче было условие использования в решении не менее трёх переменных из четырёх возможных, то в рекуррентных выражениях для поиска $f_3(b_1)$ следовало бы использовать $f_2(b_1)$ для всех возможных значений b_1 , хотя $f_3(b_1)$ опять можно было вычислять только для $b_1 = 10$ и $\Delta b_1 \in \{1, 2, \dots, 10\}$. Такое правило можно использовать и в общем случае на последнем шаге алгоритма.

Из таблицы 1 видно, что при использовании в решении одной переменной максимум целевой функции равен 12 при $x_2 = 3$, а максимум при использовании в решении двух переменных равен 13 при $x_1 = 1, x_2 = 3$.

Таблица 1 – Данные, получаемые при решении примера:
возможные значения ресурса b_1 ,
соответствующие им значения целевой функции от одной используемой переменной f_1 ,
значения номеров используемой переменной i , значения переменной x_i ,
значения целевой функции от не более двух используемых переменных $f_2 = f_1^2 + f_1^1$,
значения $f_1^2 + f_1^1$ при получении f_2 ,
 i, j - значения номеров используемых переменных i, j для, соответственно, f_1^2 и f_1^1 ,
значения переменных x_i, x_j в $f_2 = f_1^2 + f_1^1$

| | | | | | | | | | | | | | | | | | | | |
|---|-----------------------|----------|----------|----------|----------|----------|----------|----------|----------|------------|------|---|---|---|---|---|---|---|---|
| 1 | b_1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | | | | | | |
| 2 | f_1 | 1 | 2 | 4 | 4 | 6 | 8 | 9 | 9 | 12 | 12 | | | | | | | | |
| 3 | i | 1 | 1 | 2 | 2 | 3 | 2 | 4 | 4 | 2 | 2 | | | | | | | | |
| 4 | x_i | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 3 | 3 | | | | | | | | |
| 5 | $f_2 = f_1^2 + f_1^1$ | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 | | | | | | | | |
| 6 | $f_1^2 + f_1^1$ | 0+1 1 | 0+2 2 | 0+4 4 | 1+4 4 | 0+6 6 | 0+8 8 | 0+9 9 | 1+9 9 | 0+12 12 | 1+12 | | | | | | | | |
| 7 | i, j | | 1 | | 1 | | 2 | | 3 | | 2 | | 1 | 2 | | | | | |
| 8 | x_i, x_j | 0 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 3 | 1 | 3 |

При использовании рекуррентного соотношения (10) для определения последовательности $\{f_d(b_1, \dots, b_m)\}$ в общем случае считается, что b_j может принимать значения $0, \Delta_j, 2\Delta_j, \dots, K_j\Delta_j = b_j^*, j = 1, \dots, m$. Тогда в памяти вычислительной среды необходимо хранить минимум $\prod_{j=1}^m (K_j + 1)$ значений $f_{d-1}(b_1, \dots, b_m)$ для последующего получения $f_d(b_1, \dots, b_m)$.

Объём вычислений в общем случае можно оценить величиной $O(n + d^* \times K^m)$, где n - общее число переменных, d^* - ограничение на число переменных в решении, K - максимальное количество точек дискретизации среди всех ограничений, m - количество ограничений. Вычисления просты и используют результаты предыдущих шагов. При использовании для решения классического рекуррентного уравнения Беллмана объём вычислений оценивается как $O(C_n^{d^*} \times d^* \times K^m)$. При $d^* = n$ получаются оценки $O(n + n \times K^m)$ и $O(n \times K^m)$ соответственно. Таким образом, вычисления на основе уравнения Беллмана [11, 12] немного эффективнее, если в решении можно использовать все описанные в задаче переменные.

При решении задачи с большим числом ограничений m и больших числах K_j может потребоваться большой объём памяти, либо нахождение решения может быть достаточно долгим. В то же время число переменных n и d^* мало влияют на эффективность рассматриваемого алгоритма, поэтому возможности метода весьма широки при решении дискретных задач линейного программирования с большим числом переменных и с малым числом ограничений. Кроме того, количество точек дискретизации во многих задачах может выбираться кратно наибольшему общему делителю коэффициентов в каждом конкретном ограничении j , что может существенно уменьшить величины $K_j, j = 1, \dots, m$.

2 Нелинейный целевой функционал

Пусть вместо (1) рассматривается задача вида: найти

$$\mathbf{max} [c_1(x_1) + \dots + c_n(x_n)] \quad (12)$$

при ограничениях (2–3).

Используя идею динамического программирования, можно рассматривать множество задач типа (1)–(3), при всевозможных значениях b_j и d ($0 \leq b_j \leq b_j^*, j = 1, \dots, m; 1 \leq d \leq d^*$), но при этом имея ввиду, что функции $c_i(x_i)$ нелинейные.

При $d = 1$ решается задача отыскания максимума (12) при использовании одной из n переменных при всех $0 \leq b_j \leq b_j^*, j = 1, \dots, m$:

$$f_1(b_1, \dots, b_m) = \mathbf{max}_{1 \leq i \leq n} c_i(x_i) \quad (13)$$

при ограничениях (2)-(3).

Далее используется рекуррентное соотношение (10).

По определению значение $f_1(b_1, \dots, b_m)$ является максимумом целевой функции при распределении ресурсов b_1, \dots, b_m для одной из n переменных.

Пусть хотя бы одна функция $c_i(x_i)$ удовлетворяет следующему условию для значений аргумента x_i равных x и y :

$$c_i(x) + c_i(y) > c_i(x + y). \quad (14)$$

Тогда при поиске максимума по соотношениям (13) и (10) может возникнуть ситуация, когда вычисленное по (10) значение $f_d(b_1, \dots, b_m)$ будет получено за счёт «постепенной» выдачи ресурсов i -му процессу. Однако, при выполнении условия (14) фактическое значение $f_d(b_1, \dots, b_m)$ не будет максимальным значением на выделенных ресурсах. Чтобы избежать этого, следует ограничиться задачами, в которых для всех $i \in \{1, \dots, n\}$ выполняется условие:

$$c_i(x) + c_i(y) \leq c_i(x + y). \quad (15)$$

При $d = 1$ значение $f_d(b_1, \dots, b_m)$ максимально по определению. Пусть при условии (15) выражение (10) даёт максимальное значение функционала при $d = k$. Можно показать, что при $d = k + 1$ соотношение (10) даст фактическое максимальное значение для всех $0 \leq b_j \leq b_j^*, j = 1, \dots, m$.

Линейность ограничений и условие (15) исключают «повторное» распределение ресурса i -му процессу. Следовательно, в (10) при $d = k + 1$ достигается искомый максимум, а из метода математической индукции следует, что максимум достигается при любом $d \leq n$.

В качестве примера рассмотрен поиск максимума целевой функции вида

$$\begin{aligned} & (x_1)^2 + 2(x_2)^4 + 3(x_3)^2 \\ & \text{при } x_1 + 2x_2 + 4x_3 \leq 5, \\ & x_i \in \{0, 1, 2, \dots\}, i = 1, 2, 3; \sum_{i=1}^3 \check{x}_i \leq 2, \check{x}_i = \begin{cases} 1, x_i > 0, \\ 0, x_i = 0. \end{cases} \end{aligned}$$

Следуя (13) и (10), вычисляются $f_1(1) = 1, f_1(2) = 4, f_1(3) = 9, f_1(4) = 32, f_1(5) = 32$.

Далее, так как условие (15) выполняется для всех i , получаются согласно (10) $f_2(1) = 1, f_2(2) = 4, f_2(3) = 9, f_2(4) = 32, f_2(5) = 33$.

Важно отметить, что, например, все функции вида

$$a_0 + a_1x + a_2(x_2)^2 + \dots + a_n(x_n)^n,$$

где $a_i \geq 0, x \geq 0, n > 0$ удовлетворяют условию (15), т.е. класс задач, к которым применим разработанный метод, достаточно широк.

Заключение

В статье изложен новый подход к постановке и решению оптимизационных задач линейного и нелинейного типов. От классической задачи линейного программирования оптимального распределения ограниченных ресурсов между заданными процессами рассматриваемая постановка задачи отличается необходимостью выбора ограниченного числа процессов из некоторого конечного множества. Ресурсы распределяются по этим процессам так, что достигается оптимальное значение целевой функции по отношению к другим вариантам состава проектных переменных с таким же числом процессов из этого же множества. Предложенный метод решения основан на развитии идей динамического программирования. Приведены доказательства оптимальности получаемых решений.

Обоснована оценка вычислительной сложности решения поставленных задач. Приведено сравнение описываемого подхода с классическими методами решения рассматриваемых задач. Показано, что классические методы можно применять путём сведения исходной постановки задачи к $C_n^{d^*}$ задачам без дополнительных ограничений на число используемых в решении процессов, где d^* - число процессов в решении, выбираемых из n возможных. После решения $C_n^{d^*}$ задач необходимо выбрать из них оптимальное.

Рассмотренные в статье математические постановки задач и их решения имеют ряд актуальных практических приложений. Например, можно сформулировать задачу для оптимального размещения ограниченной номенклатуры грузов на ограниченных складских площадях [2], или задачу выбора ограниченного числа типов ракет-носителей из возможного набора типов для оптимального выполнения программ космических исследований [16] и другие подобные [17].

При решении практических задач приходится сталкиваться с многокритериальной оптимизацией. Если многокритериальный функционал качества каким-либо способом приводится к одному критерию [18-21], то затем можно применять предложенный метод решения.

Список источников

- [1] *Singiresu, S.R.* Engineering optimization: theory and practice / S.R. Singiresu - New York: Wiley, 2009. - 813 p.
- [2] *Офицеров, В.П.* Моделирование и оптимизация снабженческой деятельности для крупных компаний / В.П. Офицеров, С.В. Смирнов // Проблемы управления и моделирования в сложных системах: Труды V международной конф. (17-21 июня 2003 г., Самара, Россия). – Самара: СамНЦ РАН, 2003. – С. 197–205.
- [3] *Стрекаловский, А.С.* Линейные и квадратично-линейные задачи двухуровневой оптимизации / А.С. Стрекаловский, А.В. Орлов. — Новосибирск: Изд-во СО РАН, 2019. – 261 с.
- [4] *Гасс, С.* Линейное программирование / С. Гасс. — М.: Гос. изд-во физико-математической литературы, 2015. — 304 с.
- [5] *Юдин, Д.Б.* Задачи и методы линейного программирования: Конечные методы / Д.Б. Юдин, Е.Г. Гольштейн. – М.: URSS, 2010. – 264 с.
- [6] *Гераськин, М.И.* Линейное программирование / М.И. Гераськин, Л.С. Клентак. – Самара: Изд-во СГАУ, 2014. – 104 с.

- [7] **Кормен, Т.** Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. М.: ООО «И.Д. Вильямс», 2013.- 1328 с.
- [8] **Martelo, S.** Knapsack problems / S. Martelo, P. Toth. — London: Wiley, 1990. — 306 p.
- [9] **Kellerer, H.** Knapsack Problems / H. Kellerer, U. Pferschy, D. Pisinger — Springer Science+Business Media, 2004. — 548 p.
- [10] **Bretthauer, K.M.** The nonlinear knapsack problem – algorithms and applications / K.M. Bretthauer, B. Shetty // European Journal of Operational Research — 2002. — Vol. 138, Iss. 3. — P. 459–472.
- [11] **Беллман, Р.** Прикладные задачи динамического программирования / Р. Беллман, С. Дрейфус. – М.: Наука, 1965. – 460 с.
- [12] **Bellman, R.E.** Dynamic Programming / R.E. Bellman. - Princeton, NJ: Princeton University Press, 2010. – 392 p.
- [13] **Пападимитриу, Х.** Комбинаторная оптимизация: Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. - М.: Мир, 1985.- 510 с.
- [14] **Гэри, М.** Вычислительные машины и трудно решаемые задачи / М. Гэри, Д. Джонсон. — М.: Мир, 1982. – 416 с.
- [15] **Струченков, В.И.** Методы оптимизации в прикладных задачах / В.И. Струченков. – М.: СОЛОН–Пресс, 2009. – 320 с.
- [16] **Офицеров, В.П.** Об оптимальном выполнении программы космических исследований заданным числом типов ракет-носителей / В.П. Офицеров // АН СССР. Космические исследования. – 1980. – Т. 18, № 4. – С. 550–555.
- [17] **Офицеров, В.П.** Об одном подходе к автоматизации и информатизации процесса составления программ обучения / В.П. Офицеров, М.В. Офицеров, О.А. Бочарова // Вестник РУДН. Серия «Информатизация образования». – 2012. – № 4. – С. 105–113.
- [18] **Пиявский, С.А.** Метод универсальных коэффициентов при принятии многокритериальных решений / С.А. Пиявский // Онтология проектирования. – 2018. – Т. 8, №3(29). – С. 449-468. – DOI: 10.18287/2223-9537-2018-8-3-449-468.
- [19] **Пиявский, С.А.** Формулы для вычисления универсальных коэффициентов при принятии многокритериальных решений / С.А. Пиявский // Онтология проектирования. – 2019. – Т. 9, №2(32). – С. 282-298 – DOI: 10.18287/2223-9537-2019-9-2-282-298.
- [20] **Ерасов, И.В.** Об одном кардинальном алгоритме обработки экспертной информации на основе метода анализа иерархий / И.В. Ерасов, В.П. Офицеров // Информатизация образования и науки. – 2013. – № 4 (20). – С. 153–161.
- [21] **Саати, Т.** Принятие решений при зависимостях и обратных связях: Аналитические сети / Т. Саати – М.: Изд-во ЛКИ, 2008. – 360 с.

Сведения об авторах



Офицеров Владимир Петрович 1953 г. рождения. Окончил Куйбышевский авиационный институт им. С.П. Королёва в 1975 г., к.т.н. (1982). Доцент кафедры «Вычислительные машины, системы и сети» Московского авиационного института (национального исследовательского университета). В списке научных трудов более 100 работ в области прикладной математики, информационных технологий, прогнозирования развития сложных социально-технических систем. Author ID (РИНЦ): 618798. ofitserovvp@mai.ru

Смирнов Сергей Викторович, 1952 г. рождения. Окончил Куйбышевский авиационный институт им. С.П. Королёва в 1975 г., д.т.н. (2002). Главный научный сотрудник Института проблем управления сложными системами Самарского федерального исследовательского центра РАН. Член Российской ассоциации искусственного интеллекта, Международной ассоциации по онтологиям и их приложениям (IAOA). В списке научных трудов более 170 работ в области прикладной математики, компьютерного моделирования сложных систем, создания интеллектуальных систем поддержки принятия решений. ORCID: 0000-0002-3332-5261; Author ID (РИНЦ): 17628; Author ID (Scopus): 57205017287; Researcher ID (WoS): AAB-2013-2021. smirnov@iccs.ru



Поступила в редакцию 15.10.2020, после рецензирования 05.05.21. Принята к публикации 05.06.2021.

Optimization while limiting the number of design variables

V.P. Ofitserov¹, S.V. Smirnov²

¹ *Moscow aviation institute (National Research University), Moscow, Russia*

² *Samara Federal Research Scientific Center RAS, Institute for the Control of Complex Systems RAS, Samara, Russia*

Abstract

A new approach to the formulation and solution of optimization problems of linear and nonlinear type is stated in this article. The problem statement under consideration differs from the classical linear programming problem of the optimal distribution of limited resources between given processes by the need to choose a limited number of processes from a certain finite set and allocate resources over these processes. The goal is to obtain the optimal value of the objective function in relation to other options for choosing the number of processes from the same set and the distribution of resources between them. The objective function can be either linear or non-linear. A nonlinear function must have certain properties for the correct operation of the proposed algorithm for finding the optimal solution. The described method is based on the development of Bellman's ideas of dynamic programming. The proofs of the optimality of the obtained solutions are provided. The article gives an estimate of the computational complexity of the algorithm and a comparison with classical methods for solving the problems under consideration. The types of applied problems solved using the proposed method are characterized. Computer implementations of the described algorithm can be used in automated decision support systems.

Keywords: *optimization, dynamic programming, linear and non-linear programming, decision support.*

Citation: *Ofitserov VP, Smirnov SV. Optimization while limiting the number of design variables [In Russian]. Ontology of designing. 2021; 11(2): 227-238. DOI: 10.18287/2223-9537-11-2-227-238.*

List of tables

Table 1 - The data obtained by solving the example:

- possible values of the resource b_1 ,
- the corresponding values of the objective function from one used variable f_1 ,
- values of numbers of the used variable i ,
- the values of the variable x_i ,
- the values of the objective function from no more than two used variables $f_2 = f_1^2 + f_1^1$,
- values $f_1^2 + f_1^1$ when obtaining f_2 ,
- i, j - value of numbers of used variables i for, respectively f_1^2 and f_1^1 ,
- the values of the variables x_i, x_j in $f_2 = f_1^2 + f_1^1$

References

- [1] *Singiresu SR*. Engineering optimization: theory and practice. New York: Wiley, 2009. 813 p.
- [2] *Ofitserov VP, Smirnov SV*. Modeling and optimization of supplying activity for the large companies [In Russian]. Proc. of V Int. conf. "Complex systems: Control and Modeling Problems" (June 17-21, 2003, Samara, Russia). Samara: Samara Scientific Center RAS, 2003: 197–205.
- [3] *Strekalovsky AS, Orlov AV*. Linear and quadratic-linear problems of two-level optimization [In Russian]. Novosibirsk: Publishing house Siberian Branch of RAS, 2019: 261 p.
- [4] *Gus S*. Linear programming [In Russian]. Moscow: State publishing house of physical and mathematical literature, 2015: 304 p.
- [5] *Yudin DB, Golstein EG*. Tasks and methods of linear programming: Final methods [In Russian]. Moscow: URSS, 2010: 264 p.
- [6] *Geraskin MI, Klentak LS*. Linear programming [In Russian] Samara: SGAU publishing house, 2014: 104 p.
- [7] *Kormen T, Leyzerson Ch, Rivest R, Stein K*. Algorithms: creation and analysis [In Russian]. Moscow: LLC "I.D. Williams", 2013:- 1328 p.
- [8] *Martelo S, Toth P*. Knapsack problems. London: Wiley, 1990: 306 p.
- [9] *Kellerer H, Pferschy U, Pisinger D*. Knapsack Problems. Springer Science+Business Media, 2004: 548 p.

- [10] **Brethauer KM, Shetty B.** The nonlinear knapsack problem - algorithms and applications. *European Journal of Operational Research*. 2002; 138(3): 459-472.
 - [11] **Bellman R, Dreyfus S.** Applied problems of dynamic programming [In Russian]. Moscow: Publishing house "Nauka", 1965: - 460 p.
 - [12] **Bellman RE.** Dynamic Programming. Princeton, NJ: Princeton University Press, 2010: 392 p.
 - [13] **Papadimitriu H, Stayglits K.** Combinatory optimization: Algorithms and complexity [In Russian]. Moscow: Publishing house "Mir", 1985: 510 p.
 - [14] **Gary M, Johnson D.** Computers and hardly solvable tasks [In Russian]. Moscow: Publ. house "Mir", 1982: 416 p.
 - [15] **Struchenkov VI.** Optimization methods in applied tasks [In Russian]. Moscow: SOLON PRESS, 2009: 320 p.
 - [16] **Ofitserov VP.** About optimum program execution of space researches the set number of types of carrier rockets [In Russian]. *Space researches*. 1980; 18(4): 550–555.
 - [17] **Ofitserov VP, Ofitserov MV, Bocharova OA.** About one approach to automation and informatization of process of drawing up programs of training [In Russian]. *RUDN Bulletin. Education Informatiz. series*. 2012; 4: 105–113.
 - [18] **Piyavsky SA.** Method of universal coefficients for the multi-criterial decision making [In Russian]. *Ontology of designing*. 2018; 8(3): 449-468. - DOI: 10.18287/2223-9537-2018-8-3-449-468.
 - [19] **Piyavsky SA.** Forms for calculation of universal coefficients when adopting multiple critical decisions [In Russian]. *Ontology of designing*. 2019; 9(2): 282-298. - DOI: 10.18287/2223-9537-2019-9-2-282-298.
 - [20] **Erasov IV, Ofitserov VP.** About one cardinal algorithm of processing of expert information on the basis of a method of the analysis of hierarchies [In Russian]. *Informatization of science and education*. 2013; 4: 153–161.
 - [21] **Saati T.** Decision-making at dependences and feedback: Analytical networks [In Russian]. Moscow: Publishing house LKI, 2008: 360 p.
-

About the authors

Vladimir Petrovich Ofitserov (b. 1953) graduated from the Korolyov aviation Institute (Kuibyshev-city) in 1975, PhD (1982). He is an Associate professor of the department "Computers, systems and networks" of the Moscow aviation Institute (National Research University). He is a co-author of about 100 scientific articles and abstracts in applied mathematics, information technology, forecasting the development of complex socio-technical systems. Author ID (RSCI): 618798. ofitserovvp@mai.ru

Sergey Victorovich Smirnov (b. 1952) graduated from the Korolyov aviation Institute (Kuibyshev-city) in 1975, D. Sc. Eng. (2002). Chief Researcher at Institute for the Control of Complex Systems of Russian Academy of Sciences, professor at Povolzhskiy State University of Telecommunication and Informatics. He is a member of Russian Association of Artificial Intelligence and International Association for Ontology and its Applications. He is a co-author of more than 170 works in the field of applied mathematics, computer modeling of complex systems, creation of intelligent decision support systems. ORCID: 0000-0002-3332-5261; Author ID (RSCI): 17628; Author ID (Scopus): 57205017287; Researcher ID (WoS): AAB-2013-2021. smirnov@iccs.ru

Received October 15, 2020. Revised May 05, 2021. Accepted June 05, 2021.
