УДК 519.12.176

А.Н. Подоляка[1], О.Я. Никонов[2], В.А. Тимонин[2]

[1]*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков*
[2]*Харьковский национальный автомобильно-дорожный университет, Харьков*

ПОИСК СБАЛАНСИРОВАННЫХ РЕШЕНИЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Рассматривается алгоритм решения бикритериальной задачи о назначениях главный функционал, которой минимизирует время выполнения самой длинной задачи, а дополнительный минимизирует или максимизирует общее время выполнения задач.

Ключевые слова: *задача о назначениях, оптимальное решение, идеальное решение, многокритериальная задача.*

Введение

Рассматриваются задачи оптимизации назначения работ в системах с параллельно функционирующими объектами. В таких системах основная задача разбивается на множество подзадач решаемых на независимых параллельно работающих машинах, различающихся по своим техническим характеристикам. Общее время решения задачи в подобных системах определяется максимальным временем решения подзадачи. Математическая модель системы представлена бикретириальной задачей о назначениях главный функционал, которой минимизирует время решения самой длинной подзадачи, а дополнительный минимизирует или максимизирует общее время решения подзадач.

Основной материал

Постановка классической задачи о назначениях (ЗН). Необходимо найти паросочетание π^* минимального веса на множестве всех паросочетаний Р в матрице назначений $\beta_{M\times M}$, где М – порядок матрицы.

Целевая функция классической задачи о назначениях имеет вид:

$$F_{ap} = \min_{\pi\in P} \sum_{i,j\in\pi} \beta_{ij}\,, \tag{1}$$

где $w(\pi) = \sum_{i,j\in\pi} \beta_{ij}$ представляет вес паросочетания π; β_{ij} – элемент матрицы назначений, принадлежащий паросочетанию π. Вес паросочетания также называют длиной решения ЗН.

Оптимальное решение π^* – это паросочетание с минимальным весом.

$$w(\pi^*) = \min_{\pi\in P} w(\pi)\,. \tag{2}$$

Постановка сбалансированной задачи о назначениях (СбЗН). Пусть: Р* – множество всех оптимальных решений функционала F_{ap}, а P^b – функционала F_b, тогда постановка СбЗН имеет вид:

$$\begin{cases} F_b = \min\limits_{\pi\in P} \max\limits_{i,j\in\pi} \beta_{ij}; \\ F_{ap} = \min\limits_{\pi^b\in P^b} \sum\limits_{i,j\in\pi^b} \beta_{ij}. \end{cases} \tag{3}$$

F_b представляет собой функционал минимаксной задачи о назначениях. Он является главным и определяет ограничения на вес максимального элемента решения (его вес должен быть минимальным). Второй функционал является дополнительным, он требует, чтобы вес паросочетания, удовлетворяющего первому функционалу $w(\pi^b) \to \min$, был минимальным.

СбЗН имеет важное практическое приложение, например, в задачах оптимизации распределения задач в параллельных вычислительных системах (ПВС). Примером таких систем являются вычислительные кластеры. Функционал F_b минимизирует время решения самой длинной подзадачи, а значит, минимизирует время решения основной задачи. Функционал F_{ap} минимизирует общее время решения всех подзадач, т.е. дополнительный функционал отвечает за экономию ресурсов (вычислительных, энергетических) при решении основной задачи.

Постановка равномерно сбалансированной задачи о назначениях (РСбЗН). Отметим, что замена минимизации на максимизацию в функционале F_{ap} (3) сделает решения более равномерными, т.к. элементы решений будут стремиться к значению элемента найденного по функционалу F_b.

$$\begin{cases} F_b = \min\limits_{\pi\in P} \max\limits_{i,j\in\pi} \beta_{ij}; \\ F_{xap} = \max\limits_{\pi^b\in P^b} \sum\limits_{i,j\in\pi^b} \beta_{ij}. \end{cases} \tag{4}$$

Следует отметить, что РСбЗН может быть использована для поиска компромисса в алгоритмах решения многокритериальных задач.

СбЗН и РСбЗН представляют собой одну и ту же задачу, т.к. математические постановки задач идентичны. Алгоритм решения у задач также будет одинаковым.

Алгоритм решения СбЗН. В основе алгоритма решения СбЗН лежит принцип приближения решений к идеальному решению задачи. В качестве этого решения выступает оптимальное решение классической ЗН.

Поиск идеального решения. Предположим, что СбЗН имеет решение, которое минимизирует оба функционала F_b и F_{ap} независимо друг от друга, т.е. $P^* \bigcup P^b \neq \otimes$. Тогда, элемент этого решения может быть вычислен с использованием любого оптимального решения классической ЗН:

$$\beta^{id} = \frac{w(\pi^*)}{M} = \frac{1}{M} \min_{\pi \in P} \sum_{i,j \in \pi} \beta_{ij} . \qquad (5)$$

Решение, которое минимизирует функционал классической ЗН и состоит из равных элементов назовем *идеальным*. Маловероятно, что такое решение существует. Его элементы можно рассматривать как математическое среднее относительно которого выполняется сравнение элементов реальных решений. Отметим, что математическое ожидание элементов каждого оптимального решения ЗН будет одинаковым, т.к. длина всех решений одинакова.

На основе идеального решения рассчитывается матрица выигрышей, которая отражает значение выигрыша или проигрыша при использовании того или иного элемента решения.

Необходимо отметить, что в некоторых системах, идеальное решение задачи может быть заранее известно, например, определяется спецификациями оборудования или задаваться вышестоящей организацией. В некоторых задачах идеальное решение может определяться на основе личного опыта или предпочтений лица принимающего решения. Понятно, что в этих случаях следует использовать заданное идеальное решение [3, 4].

Алгоритм вычисления матрицы выигрышей. Вычисление элементов матрицы выигрышей D выполняется по формуле:

$$D_{ij} = \begin{cases} \left(\beta_{ij} - \beta^{id}/M, 0\right) & \text{if } \left(\beta_{ij} - \beta^{id}/M\right) \geq 0; \\ \left(0, \beta^{id}/M - \beta_{ij}\right) & \text{if } \left(\beta_{ij} - \beta^{id}/M\right) < 0. \end{cases} \qquad (6)$$

Значение D_{ij} представляет собой вектор из двух полей: loss и gain, которые представляют собой соответственно значения недостатка и избытка при использовании элемента β_{ij}.

Для выигрышных элементов матрицы loss = 0 (нулевые потери), а для проигрышных выигрыш gain = 0 (нулевой выигрыш).

Поле loss является более важным. Будем считать, что его минимизация соответствует функционалу F_b СбЗН. Поле gain – дополнительное и отвечает за стоимость решения в целом. Будем считать, что максимизация gain соответствует функционалу F_{ap} СбЗН, а его минимизация – функционалу F_{xap} РСбЗН.

Замечание. Если элементы матрицы назначений целые числа, то элемент идеального решения, скорее всего, будет действительным числом. Чтобы оставаться в пространстве целых чисел можно все элементы матрицы D умножить на M. В этом случае, оптимальная перестановка не изменится, а ее вес будет равен $w(\pi^*)M$.

Чтобы найти оптимальное решение в матрице D необходимо решить бикритериальную задачу, имеющую множество упорядоченных по важности критериев (loss и gain). В нашем случае, ее постановка для СбЗН имеет вид:

$$\begin{cases} F_{loss} = \min_{\pi \in P} \sum_{i,j \in \pi} \left[D_{ij}.loss\right]; \\ F_{gain} = \max_{\pi^{loss} \in P^{loss}} \sum_{i,j \in \pi^{loss}} \left[D_{ij}.gain\right]. \end{cases} \qquad (7)$$

А для РСбЗН:

$$\begin{cases} F_{loss} = \min_{\pi \in P} \sum_{i,j \in \pi} \left[D_{ij}.loss\right]; \\ F_{xgain} = \max_{\pi^{loss} \in P^{loss}} \sum_{i,j \in \pi^{loss}} \left[-D_{ij}.gain\right]. \end{cases} \qquad (8)$$

Эта задача может быть сведена к однокритериальной, если для элементов D_{ij} определить необходимые арифметические операции и операторы сравнения.

Арифметические операции и операции сравнения. Пусть вектора a(l,g) и b(l,g)- элементы матрицы выигрышней D, а l и g представляют значения выигрыша и проигрыша.

a(0,0) – нейтральный элемент.

–a(l,g) = a(–l,–g) – обратный элемент.

Сложение:

a(l,g) + b(l,g) = c(a.l + b.l, a.g + c.g); (9)

a(l,g) +(–a(l,g)) = a(0,0).

Вычитание:

c(l,g) – b(l,g) = a(c.l – b.l, c.g – b.g) = a(l,g). (10)

Равенство:

a(l,g) == b(l,g)if (a.l == b.l AND a.g == c.g); (11)

a(l,g) != b(l,g), if (a.l != b.l OR a.g != c.g). (12)

Больше СбЗН:

a(l,g) > b(l,g)

if (a.l > b.l) OR ((a.l == b.l) AND (a.g > c.g)). (13)

Больше РСбЗН:

a(l,g) > b(l,g)

if (a.l > b.l) OR ((a.l == b.l) AND (a.g < c.g)). (14)

Меньше:

a(l,g) < b(l,g)

if (a.l < b.l) OR ((a.l == b.l) AND (a.g < c.g)). (15)

Меньше РСбЗН:

a(l,g) < b(l,g)

if (a.l < b.l) OR ((a.l == b.l) AND (a.g > c.g)). (16)

Элементы матрицы выигрышней D реализуются на языке программирования в виде класса с соответствующим набором арифметических операций и операторов сравнения.

Свертка функционала. Будем считать, что оптимизации функционалов F_{loss} и F_{gain} (3), (4) соответствует минимизации суммы значений D_{ij}, т.к. выполнение (3), (4) гарантируется арифметическими операциями и операторами сравнения элементов D_{ij}. Указанное допущение, позволит нам перейти от бикритериальной задачи (7), (8) к однокритериальной векторной задаче.

$$F_{lg} = \min_{\pi\in P} \sum_{i,j\in\pi} D_{ij} . \quad (17)$$

Т.е. процедура свертки функционала СбЗН имеет вид.

$$\begin{cases} F_b = \min_{\pi\in P} \max_{i,j\in\pi} \beta_{ij} \\ F_{ap} = \min_{\pi^b\in P^b} \sum_{i,j\in\pi^b} \beta_{ij} \end{cases} \Rightarrow \begin{cases} F_{loss} = \min_{\pi\in P} \sum_{i,j\in\pi} \left[D_{ij}.loss \right] \\ F_{gain} = \min_{\pi^{loss}\in P^{loss}} \sum_{i,j\in\pi^{loss}} \left[D_{ij}.gain \right] \end{cases} \Rightarrow$$

$$\Rightarrow F_{lg} = \min_{\pi\in P} \sum_{i,j\in\pi} D_{ij}.$$

Таким образом, алгоритм решения СбЗН состоит из следующих этапов:

1. Решение классической задачи о назначениях и вычисление веса элемента идеального решения β^{id} по формуле (6).
2. Вычисление матрицы выигрышей D_{ij} по формуле (6).
3. Определение оптимального решения D* для матрицы D_{ij}.
4. Определение решения СбЗН на основе паросочетания D*.

Оценка эффективности алгоритма решения СбЗН. Сложность алгоритма, представленного в работе, зависит от сложности вычисления значений D_{ij} и сложности алгоритма решения классической однокритериальной ЗН. Сложность вычисления D_{ij} по формуле (6), (10) определяется их количеством и равна $O(n^2)$. Наиболее эффективным методом решения ЗН является алгоритм Куна, сложность которого равна $O(n^3)$. Поэтому сложность алгоритма решения СбЗН определяется сложностью решения классической ЗН и равна $O(n^3)$. Исходя из этого, можно сделать вывод, что данный алгоритм решения СбЗН является предельно эффективным.

Выводы

Алгоритмы решения минимаксной ЗН основаны на решении последовательности ЗН, каждая из которых накладывает ограничения на значения матрицы β_{MxM}. Например алгоритм, рассматриваемый в работе [1] использует дихотомический поиск. Следует отметить, что алгоритм решения СбЗН является более эффективным с точки зрения вычислительной сложности.

В отличие от однокритериальных задач, например, классической ЗН, решения многокритериальных задач противоречивы, поскольку, невозможно с абсолютной уверенностью утверждать, что то или иное решение, строго оптимально. Поэтому, ключевой характеристикой решений является их равномерность, которая отражает баланс критериального выигрыша и проигрыша. Удовлетворение этого принципа приводит к увеличению длины решений, а значит, потере их качества. Можно сказать, что дополнительный функционал СбЗН компенсирует увеличение длины решения при удовлетворении принципа равномерности. Поэтому, алгоритм может быть использован в качестве одного из методов решения многокритериальной задачи о назначениях.

Следует также отметить, что ключевой проблемой предложенного подхода при решении многокритериальных задач является процедура выбора идеального решения. Поэтому, основное внимание дальнейших исследований будет направлено на разработку интерактивной процедуры построения идеальных решений, которая позволит находить множество решений, различающихся по своим характеристикам. Это позволит лицу принимающему решения на основе личного опыта и предпочтений осуществить выбор эффективного решения из этого множества, или построить новое решение, если будет необходимо.

Список литературы

1. *Панішев А.В. Дихотомічний пошук рішення мінімаксної задачі про призначення / А.В. Панішев, О.О. Подоляка, С.В. Черницук // Вісник ЖІТІ. – 1998. – № 7. – С. 195-201.*
2. *Панішев А.В. Одне з узагальнень задачі про призначення з обмеженнями / А.В. Панішев, О.О. Подоляка // Вісник ЖІТІ. – 1999. – № 11. – С. 139-144.*
3. *Штойер Р. Многокритериальная оптимизация: теория, вычисления и приложения / Р. Штойер. – М.: Радио и связь, 1992. – 360 с.*
4. *Хоменюк В.В. Элементы теории многокритериальной оптимизации / В.В. Хоменюк. – М.: Наука, 1983. – С. 8-25.*

Поступила в редколлегию 24.01.2011

Рецензент: д-р техн. наук, проф. А.В. Панишев, Житомирский государственный технологический университет, Житомир.

ПОШУК ЗБАЛАНСОВАНИХ РОЗВ'ЯЗКІВ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

О.М. Подоляка, О.Я. Ніконов, В.О. Тімонін

Розглядається алгоритм розв'язання бікрітеріальної задачі про призначення головний функціонал, якої мінімізує час виконання найдовшої задачі, а додатковий мінімізує або максимізує загальний час виконання задач.

Ключові слова: *задача про призначення, оптимальний розв'язок, ідеальний розв'язок, багатокритеріальна задача.*

THE SEARCH OF BALANCED SOLUTION OF THE ASSIGNMENT PROBLEM

A.N. Podolyaka, O.Y. Nikonov, V.A. Timonin

An algorithm of decision the bicriterial assignment problem, which main functional minimizes the performance time of the longest task, but an additional minimizes or maximizes the total performance time of all problems.

Keywords: *assignment problem, optimal decision, ideal decision, multicriterion problem.*