УДК 519.8
ББК 22.18
М 74

Нагоев А.В.
Кандидат экономических наук, доцент кафедры математических методов и информационных технологий экономического факультета Адыгейского государственного университета, Майкоп, e-mail: navlad73@mail.ru
Блягоз З.У.
Кандидат физико-математических наук, профессор, зав. кафедрой математических методов и информационных технологий экономического факультета Адыгейского государственного университета, Майкоп, e-mail: blyagoz.zu@yandex.ru
Тешев В.А.
Кандидат физико-математических наук, доцент кафедры математических методов и информационных технологий, зам. декана экономического факультета Адыгейского государственного университета, Майкоп, e-mail: vateshev@mail.ru
Шелехова Л.В.
Доктор педагогических наук, доцент кафедры математических методов и информационных технологий экономического факультета Адыгейского государственного университета, Майкоп, e-mail: schelehova@rambler.ru

Модели управления затратами предприятия, представленные в виде задачи с булевыми переменными

(Рецензирована)

Аннотация
Рассмотрен модифицированный алгоритм оптимизации перераспределения материально-технических ресурсов, который сходен с методом неявного перебора, но значительно более прост. Повышение быстродействия алгоритма достигается за счет построения последовательности решений, для которых последовательность значений целевой функции не убывает.
Ключевые слова: *распределительная задача, метод Балаша, перераспределение ресурсов, оптимальность решения.*

Nagoev A.V.
Candidate of Economics, Associate Professor of Mathematical Methods and Information Technologies Department of Economics Faculty, Adyghe State University, Maikop, e-mail: navlad73@mail.ru
Blyagoz Z.U.
Candidate of Physics and Mathematics, Professor, Head of Mathematical Methods and Information Technologies Department of Economics Faculty, Adyghe State University, Maikop, e-mail: blyagoz.zu@yandex.ru
Teshev V.A.
Candidate of Physics and Mathematics, Associate Professor of Mathematical Methods and Information Technologies Department, Deputy Dean of the Economic Faculty, Adyghe State University, Maikop, e-mail: vateshev@mail.ru
Shelekhova L.V.
Doctor of Pedagogy, Associate Professor of Mathematical Methods and Information Technologies Department of Economics Faculty, Adyghe State University, Maikop, e-mail: schelehova_lv@mail.ru

Cost management model of the enterprise, presented as a problem with Boolean variables

Abstract
The paper deals with the modified algorithm of optimization of MTR redistribution, similar to the method of implicit search. Increase in speed of the algorithm is achieved by constructing a sequence of solutions with undiminishing sequence of values of the objective function.
Keywords: *distribution task, method of Balash, the redistribution of resources, optimality of the solution.*

Рассмотрим процессы перераспределения на предприятии неиспользуемых ресурсов. Совокупность последних делится на два множества: множество неиспользуемых ресурсов, по которым спрос не превышает наличия, и множество ресурсов, спрос по которым превышает предложение. Для каждого из этих множеств предложены различные стратегии и соответствующие модели перераспределения.

Для того чтобы построить модель, предположим, что есть два класса предприятий-заказчиков: те, которые находятся в непосредственной близости от предприятий-поставщиков, и те, которые удалены на значительное расстояние. С помощью систем связи компания информирует о потребностях аутсорсинговому предприятию в его автоматизированную информационную систему. Программа-диспетчер осуществляет контроль за закрепленными в ней удаленными пользователями. Доступ предприятия к автоматизированным информационным ресурсам организуется подачей с терминала компании заявки на перевозку материальных ресурсов, адресованной предприятию. Компьютерная программа по распределению заявок начинает работать, как только появляется хотя бы одна заявка.

Критерием при распределении k-го материального ресурса между предприятиями является плата за их перемещение от i-го предприятия к j-му предприятию. Физический смысл платы может быть самым разнообразным. Для учета платы в предприятии хранится матрица C, элементы которой определяются из следующих условий:

$C_{ij} = 0$, если материальный ресурс k, необходимый для j-го предприятия, находится там же;

$C_{ij} = C_{ij}$, если у i-го предприятия имеется в наличии заданный материальный ресурс k, который перевозится на j-ое предприятие;

$C_{ij} = \infty$, если i-ое предприятие не может предоставить материальный ресурс k на j-е предприятие.

Если j-му предприятию требуются материальные ресурсы из L различных предприятий, то в качестве элементов матрицы C_{ij}^1 берутся

$$C_{ij} = \sum_{l=1}^{L} C_{ij}^1.$$

Формальная постановка задачи. Пусть к моменту времени t на предприятии поступило m заявок от организаций, которые нуждаются в материальных ресурсах, и известны n организаций, у которых эти ресурсы есть. В заявке указан тип материальных ресурсов и время T_j, за которое осуществляется доставка материальных ресурсов j-му предприятию [1].

Будем считать, что информация, требуемая для реализации заявки организации, может поступать на предприятие и запоминается там в любой момент, а непосредственное решение задачи возможно только во временные интервалы:

$$T_{i,d} = t_{i,d}^1 - t_{i,d}^{11}, \quad d = 1,2,\ldots, f_i,$$

где $t_{i,d}^1$ – момент начала d-го интервала i-го предприятия; $t_{i,d}^{11}$ – момент окончания d-го интервала i-го предприятия; f_i – число интервалов у i-го предприятия.

Временные интервалы определяют расписание поставки материальных ресурсов i-го предприятия, причем $T_{i,d} >> T_j$. Так как мы рассматриваем оперативное выполнение заявок предприятий, то будем считать, что реализация заявки предприятия, начавшаяся в некоторый интервал $T_{i,d}$, должна быть завершена в течение этого же интервала. Таким образом, задача о распределении материальных ресурсов предприятия состоит в такой реализации заявок предприятий, при которой выполнялись бы соотношения:

$$b_i = T_{i,d}\text{, если } t \le t^1_{i,d}\text{,} \quad \text{и} \quad b_i = t^{11}_{i,d} - t\text{, если } t^{11}_{i,d} < t < t^1_{i,d}\text{,}$$

где b_i – время, необходимое для реализации заявок предприятий, а суммарные затраты на их перемещение были минимальными.

При этом одна и та же заявка предприятия не должна обслуживаться несколькими предприятиями. После того как время *d*-го интервала *i*-го предприятия реализовано, распределяется $(d+1)$ интервал. Следовательно, мы пришли к распределительной задаче следующего вида: найти вектор

$$x = \{x_{i,j}\}, \quad i = 1,2,\dots,n; \quad j = 1,2,\dots,m,$$

минимизирующий функцию

$$z = \sum_{i\in I}\sum_{j\in J} C_{ij} * x_{ij} \tag{1}$$

при условиях

$$\sum_{j\in J} x_{ij}T_j \le b_i, \quad \forall j \in J, \tag{2}$$

$$\sum_{i\in I} x_{ij} = 1, \quad \forall j \in J, \tag{3}$$

$$x_{ij} = 1{,}0 \quad \forall i \in I, \quad \forall j \in J. \tag{4}$$

Вектор *x*, удовлетворяющий ограничениям (3) и (4), назовем решением; вектор *x*, удовлетворяющий ограничениям (2), (3) и (4), – допустимым решением; допустимое решение, минимизирующее функцию (1), – оптимальным решением.

Как только заявка предприятия закрепляется за соответствующим предприятием, имеющим запрашиваемый материальный ресурс, предприятие-заказчик получает сообщение о времени $t_{i,d,k}$ возможного перемещения ресурса.

Математическая модель прикрепления предприятий-заказчиков к предприятиям, имеющим тот или иной материальный ресурс, относится к классу задач целочисленного программирования с булевыми переменными (0-1) и для их решения можно применить существующие уже алгоритмы [2].

Специфика задачи, отраженная в уравнениях (2)-(4), позволяет построить алгоритм, сходный с методом Балаша, но более простой и эффективный. В этом алгоритме число решений задачи оказывается равным n^m.

Основная идея алгоритма составит в построении последовательности решений $X = \{x_q\},\ q = 1, Q$, для которых последовательность значений целевой функции $Z = \{z_q\}$ не убывает. При этом первое полученное решение является допустимым и будет оптимальным решением задачи. Последовательности *X* можно поставить в соответствие граф *G*, имеющий форму дерева такую, что узел *q* графа *G* представляет собой решение x_q, каждую дугу (q, p) связывает два решения x_q и x_p. Корнем дерева x_0 является решение с минимальным значением целевой функции. Известен алгоритм направленного перебора решений *X*, основанный на запоминании всех промежуточных решений, что необходимо, так как заранее неизвестная часть может потребоваться в дальнейшем для построения дерева решений. Такой алгоритм требует большого объема памяти, поэтому он был модифицирован таким образом, чтобы на каждом шаге запоминалось только два решения, необходимых для построения следующего шага, – продолжение 1 и продолжение 2.

Продолжение 1 строится из узла x_p, для которого

$$z_p = \min_q \{z_q\},$$

а *продолжение 2* – из узла, от которого образовано решение x_p, причем значение целевой функции продолжения 2 имеет минимальное приращение относительно z_p. Если на некотором шаге построения дерева решений надо вернуться назад к пройденному узлу, то по номеру узла определяется узел, из которого он получен, и строятся новые два решения.

Алгоритм решения задачи. Для описания алгоритма используются обозначения, принятые в исходной задаче. Алгоритм состоит из начального шага, на котором находится решение x_0, и трех повторяющихся шагов: допустимости, ветвления и выбора.

Начальный шаг. Для каждого $\forall j \in J$ определяем $i_j \in I$, для которого выполняется соотношение:

$$C(i(j), j) = \min_i \{c_{ij}\}.$$

Если это соотношение выполняется для нескольких индексов i_j, то выбираем наибольший индекс $i_j^1 = \max\{i_j\}$. Затем полагаем $x(i(j), j) = 1$ и $x_{ij} = 0$ для всех $i \in I$ $(i \neq i_j)$. Этому решению соответствует x_0 – корень дерева *G*.

Шаг допустимости. Решение x_q, полученное на предшествующем шаге, подвергается проверке 1 и проверке 2.

Проверка 1. Вычисляем выражение

$$b_i^1 = b_i - \sum_{j \in J} x_{ij}^1 T_j, \quad \forall i \in I.$$

Если $b_i \geq 0$ $(\forall i \in I)$, то решение x_q удовлетворяет этой проверке и является оптимальным решением, решение на этом заканчивается. Если $b_i < 0$ для некоторого $i \in I$, то образуем множества

$$M_q = \{i | b_i^1 < 0\} \quad \text{и} \quad N_i = \{j | x_{ij}^q = 1\}, \quad \forall i \in M_q,$$

где M_q – множество ограничений (2), не удовлетворяющих проверке; N_i – множество индексов *j*, попавших в эти ограничения. После этого переходим к проверке 2.

Проверка 2. Для каждого $i \in M_q$ и $j \in N_i$ устанавливаем, существует ли индекс $s \neq i$ такой, что $c_{sj} > c_{ij}$, и образуем множества

$$J_i = \{j | j \in N_i, i \in M_q\}, \quad s \neq i$$

такие, что $c_{si} > c_{ij}$ или $s \neq i$ такое, что $c_{si} = c_{ij}$. Затем вычисляем величину

$$f_i = \sum_{j \in J_i} T_j.$$

Если $f_1 < -b_1^{-1}$ по крайней мере для одного индекса $i \in M_q$, то считаем, что решение не прошло проверку, узел *q* вычеркиваем и переходим к шагу выбора. Если $f_1 \geq -b_1^{-1}$ $(\forall i \in M_q)$, то вычисляем величины

$$g_{ij} = c_{si} - c_{ij} \quad \forall\, (i \in M_q,\ j \in J_i,\ s \in I)$$

и образуем множества

$$s_{ij} = \{s | g_s^{ij} > 0\ s \neq i \ \text{ и } g_s^{ij} = 0\ s < 1\},$$

$$J(i^i) = \left\{ i^1 \middle| a_{ij}^{i1} = \min_{s \in S_{ij}} g_{ij}^s \right\}.$$

Множество $J\left(i^1\right)$ содержит номера ограничений i^1, в которые необходимо переместить *j*-ые компоненты решения, чтобы выполнялось *i*-ое ограничение в решении x_q.

Шаг ветвления. Строятся два новых решения: продолжение 1 и продолжение 2. Для каждого $i \in M_q$ строим все решения $x\left(i^1 r\right)$, удовлетворяющие следующим условиям:

1. $x_{ij}^{i1r} = x_{ij}^q \quad \left(\forall j \in \left(J - J_i\right) \text{ и } \forall i \in J\right)$;

2. Для каждого $j \in J_i \quad x_{wj}^{i1r} = 1$ либо для $w = i$, либо для *w*, равного элементу множества $I\left(i^1\right)$, соответствующего рассматриваемому элементу j;

3. $\sum_{j \in J} x_{ij}^{ilr} \le b_i$.

4. Не имеется никакой избыточности, т.е. для двух решений $x\left(i^1 v\right)$ и $x\left(i^1 e\right)$ выполняются следующие условия: если определить для каждого решения x_{ir} множество

$$E_{ir} = \left\{ i \middle| j \in J_i, \; x_{ij}^{i1r} = 0 \right\},$$

то выполнено как условие $E_{iv} \not\subset E_{ie}$, так и условие $E_{ie} \not\subset E_{iv}$. Количество комбинаций P_i, с помощью которых можно выполнить *i*-ое невыполненное ограничение (2), удовлетворяет условию:

$$P_i = \sum_{\beta=1}^{R_i} c_{Ri,\beta}, \quad i \in M_q, \tag{5}$$

где R_i – число элементов множества J_i; $c_{Ri,\beta}$ – число сочетаний из R_i элементов по β элементов.

Общее количество решений X_α, с помощью которых можно выполнить невыполненные ограничения (1) для всех $i \in M_q$, равно

$$P = \prod_{i \in M_q} P_i. \tag{6}$$

Для экономии объема памяти на этапе получения P_i комбинаций было предложено запоминать не сами комбинации, а соответствующие им приращения коэффициентов матрицы *C:*

$$\Delta C_{i\Theta} \sum_{i \in J_i} g_{ij}^{i1}, \quad i \in M_q, \quad \Theta = 1, 2, \ldots, P_i.$$

На этапе перебора решений $X_\alpha = \{x_h\}$, $h = 1, P$, для каждого полученного решения x_h определяется приращение целевой функции

$$\Delta z_h = \sum_{i \in M_q} \Delta c_{i\Theta}$$

и сравнивается с приращением целевой функции, полученным для предыдущего решения. Если $\Delta z_h \le \Delta z_{h-1}$, то решение *x* запоминается. Если $\Delta z_h \le \Delta z_{h-1}$, то формируется следующее решение. После того, как будут просмотрены все решения X_α, в памяти останется решение $x_{q+1} \subset X_\alpha$, имеющее минимальное приращение целевой функции

$$\Delta z_{q+1} = \min_h \{\Delta z_h\}.$$

Полученное решение x_{q+1} представляет собой вектор

$$x_{q+1} = \left\{\alpha_1, \alpha_2, \ldots, \alpha_i\right\}, \quad i \in M_q,$$

компоненты которого α_i есть порядковые номера допустимых комбинаций. Само решение получается после повторного обращения к программе получения P_i комбинаций путем выделения комбинации, которой соответствую x_{q+1}. Решение x_{q+1} является продолжением 1. Продолжение 2 находится аналогичным образом, только в качестве узла, из которого оно строится, берется узел x_γ, из которого получено решение x_q, а приращение целевой функции определяется относительно последнего решения, полученного из x_γ.

Шаг выбора. В результате просмотра всех узлов, зафиксированных к рассматриваемому моменту, определяется и исключается из Z узел, с которым связана минимальная стоимость

$$z_\beta = \min_q \{z_q\}.$$

Далее процесс построения дерева решений будет продолжаться из узла x_β, поэтому переходим к шагу допустимости. Если существует несколько z_β, удовлетворяющих этому условию, то выбирается узел, имеющий наибольший индекс. Если $z = 0$, то никакого допустимого решения не существует и алгоритм заканчивается.

Для исследования эффективности модифицированного алгоритма и оценки объема оперативной памяти, необходимой для работы, была составлена программа алгоритма на языке Турбо-Паскаль. Исходные данные для решения задачи, т.е. матрица затрат C и коэффициенты T_i и b_i в ограничениях (3-5), были получены с помощью датчика случайных чисел с равномерным законом распределения. Результаты решения приведены в таблице 2, где $z_{\text{опт}}$ – значение функции стоимости, соответствующее оптимальному решению; P_1 – число решений, хранящихся в памяти; $P_{\text{опт}}$ – номер оптимального решения (P_1 и $P_{\text{опт}}$ мало отличаются друг от друга); $T_{\text{реш}}$ – время решения задачи.

Таблица 1

Решение распределительной задачи методом модифицированного алгоритма

n	m	$z_{\text{опт}}$	P_1	$P_{\text{опт}}$	$T_{\text{реш}}$	Число узлов
3	27	139	3	2	1	22
3	52	256	3	2	1	20
4	30	139	3	2	1	390
5	33	142	9	7	12	728
5	65	213	5	4	1	36
6	75	227	17	15	20	8118
7	42	137	74	66	45	6197
7	82	234	7	5	10	10923

В столбце «Число узлов» для сравнения дано общее количество решений, просмотренных в процессе построения дерева решений (оно равно числу узлов, которые надо хранить в памяти при немодифицированном алгоритме).

Анализ результатов, приведенных в таблице 1, показывает, что модифицированный алгоритм обладает хорошей сходимостью к оптимальному решению и дает значительный выигрыш в памяти, что особенно важно для больших размерностей задачи.

Примечания:

1. Сирота А.А. Компьютерное моделирование и оценка эффективности сложных систем. М.: Техносфера, 2006. 280 с.
2. Советов Б.Я., Яковлев С.А. Моделирование систем. М.: Высшая школа, 1998. 343 с.

References:

1. Sirota A.A. Computer modeling and assessment of effectiveness of complex systems. M.: Technosfera, 2006. 280 pp.
2. Sovetov B.Ya., Yakovlev S.A. Modeling of systems. M.: Vysshaya Shkola, 1998. 343 pp.