Reduction of the asymptotic complexity of the assignment problem

E A Morozov¹, V A Bartenev¹, G M Rudakova²,³, O V Korchevskaya² and E V Tueva²

¹ JSC “Academician M F Reshetnev Information satellite systems”, 52 Lenin street, Zheleznogorsk, Krasnoyarsk region, 662972, Russia
² Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russia
³ E-mail: gmrfait@gmail.com

Abstract. The assignment problem belongs to an extensive class of managerial decision-making tasks. One of the urgent tasks related to this class is the problem of effectively combining various streams (resources). The main purpose of the method modifications is to reduce the asymptotic complexity and to consider the specifics of the subject area. The article describes the solution of the assignment problem on a bipartite sparse graph. The main purpose of the method modifications is to reduce the asymptotic complexity and to consider the specifics of the subject area. A modification of the Hungarian method is proposed. The method is based on the successive shortest path algorithm and Dijkstra’s bidirectional search algorithm. The modification led to a decrease in the asymptotic complexity of the problem.

1. Introduction

The assignment problem belongs to an extensive class of managerial decision-making tasks. One of the urgent tasks related to this class is the problem of effectively combining various streams (resources). It involves finding the distribution of objects of one of the sets by a group of objects of another set and this distribution must correspond to the extremum of the objective function. There are various methods for solving assignment problems: the distribution method [1], the potential method [2], the transport problem in the network formulation [3], the method of resolving terms [1], the differential rent method [4], and others.

The classical algorithm for solving the assignment problem is a polynomial Hungarian method, developed in 1955 by H. Kuhn [5]. There are various modifications. For example, in [6, 7], the original algorithm was introduced, which made it possible to reduce the time complexity of the algorithm from $O(n^4)$ to $O(n^3)$.

The main purpose of the method modifications is to reduce the asymptotic complexity and to consider the specifics of the subject area.

In [8], an approach is presented in which new applications are dynamically added. In [9], the change of weight coefficients is also allowed, which is typical for transport and other tasks.

To not have to re-optimize the whole matrix when adding a new row or column, the algorithm should be incremental, i.e. adding a row or column should occur in $O(n^2)$ time.

In [10], a description of the solution of the assignment problem is presented, which is incremental.
in rows but not in columns. Adding a row is performed in \(O(n \times m)\) operations, where \(n\) is the number of rows, and \(m\) is the number of columns in the optimized part. This decision is taken as a basis.

2. Problem statement

Let there be two types of resource \(R = \{U; V\}\), where \(U_i(Z), i = 1,...,m\) and \(V_j(Z), j = 1,...,n\). Resources enter the system from outside at random times. When a new resource arrives \(U_i(Z) (V_j(Z))\) comparison is made with the existing ones from set \(V_j(Z) (U_i(Z))\) to combine with a certain degree of requirements \(Z\).

Let \(A\) be the set of found matches, each of which is associated with a weight. The weight of pairs missing in \(A\) will be considered equal to 0.

It is required to determine such as \(x_{ij}\), to maximize

\[
F_A = \sum_{(i,j) \in A} a_{ij} x_{ij}
\]

under restrictions

\[
\sum_{i \in U} x_{ij} \leq 1, \forall j \in V, \tag{2}
\]

\[
\sum_{j \in V} x_{ij} \leq 1, \forall i \in U, \tag{3}
\]

\[
x_{ij} \geq 0, \forall i \in U, j \in V. \tag{4}
\]

We assume that the power \(|U| = n\), a \(|V| = m\).

3. Problem statement

It is required to present algorithms for solving problem (1) – (4) in dynamics. That is, to determine the way to bring the solution \(x_{ij}\) to the optimal state by adding one vertex or removing one vertex without re-solving the entire problem.

Note that the solution to problem (1) – (4) even on a graph with fractions of equal power is not always a complete matching, moreover a complete matching does not always exist in sparse graphs. Figure 1 (a) shows an example of the matrix \(A\), in which there is a complete match with the value of the objective function \(F = 3\) and more optimal solution \(c F = 4\).

\[
\begin{pmatrix}
1 & 2 & 0 \\
0 & 1 & 2 \\
0 & 0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 & 2 \\
2 & 1 & 0 \\
2 & 2 & 1
\end{pmatrix}
\]

(a) (b)

Figure 1. Example of the matrix.

With a similar approach, the matrix \(A\) from example is converted to matrix \(B\), presented in the figure 1 (b). The following conversion is used: \(b_{ij} = 2P - a_{ij}, P \geq a_{ij}\). We present the problem (1) - (4) to the form suitable for solving by the classical Hungarian algorithm.

An example of the original task graph is presented in figure 2 (a).
Let us complement each vertex of the graph with its double belonging to the opposite lobe. Connect the top and its twins with weight $2P$ (figure 2 (b)). Then we connect each double with the other doubles in the same way that the vertices that generated them are connected (figure 2 (c)). The resulting graph has an adjacency matrix, shown in Figure 2 (d). Let’s call it $C$.

We formulate the problem with the matrix $C$.

It is required to determine such $y_{ij} \in \{0,1\}, i = 1..(n + m), j = 1..(n + m)$, to minimize

$$F_C = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij}$$

under restrictions

$$\sum_{i=1}^{n+m} y_{ij} = 1, \forall j = 1..(n + m),$$

$$\sum_{j=1}^{n+m} y_{ij} = 1, \forall i = 1..(n + m),$$

$$y_{ij} \geq 0, i = 1..(n + m), j = 1..(n + m).$$

Obviously, regardless of the value of the matrix $A$, the problem with the matrix $C$ has a full matching on the main diagonal, that is, the upper estimate of the minimum value of $F_C$ is $F_C = 2P(n + m)$.

4. Modified Hungarian method

Lemma 1. If $y_{ij}$ is the solution of the problem (5)-(8), then

$$\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} = \sum_{i=1}^{n+m} \sum_{j=1}^{i-1} c_{ij} y_{ij}.$$

Proof.

From the opposite. Suppose for definiteness that

$$\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} < \sum_{i=1}^{n+m} \sum_{j=1}^{i-1} c_{ij} y_{ij}$$

For the opposite case, the reasoning is similar. Build a set

$$y_{ij}', y_{ij}' = \begin{cases} y_{ij}, & \text{if } i \geq j \\ y_{ji}, & \text{if } i < j \end{cases}$$

Every $y_{ij} = 1$ above the main diagonal belongs area $B$, and below – to area $B^T$, because the rest
$c_{ij} = \infty$, and value $F_C$ is finite. Therefore, every $y_{ij} = 1$, for which $i > j$ is $i \leq n, j > n$. Thus, for cols $i \leq n$ set $y'_{ij}$ matches with $y_{ij}$, and for rows $i > n$ set $y'_{ij}$ matches with cols of solution $y_{ij}$, $j > n$. Therefore, restriction (7) is performed for set $y'_{ij}$. Set $y'_{ij}$ symmetrical with respect to the main diagonal, then restriction (6) is performed.

By assumption

$$
\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} < \sum_{i=1}^{n+m} \sum_{j=1}^{n-1} c_{ij} y_{ij} 
$$

(12)

$$
F_C' = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} + 2P \sum_{i=1}^{n+m} y_{ii} + \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} < \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} + 2P \sum_{i=1}^{n+m} y_{ii} + \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} = F_C
$$

(13)

Set $y'_{ij}$ gives a smaller value of the objective function than the solution $y_{ij}$. We received a contradiction; therefore, the assumption is not true.

Interconnection of problem solving (1) – (4) and (5) – (8) determined by Lemma 2 and Lemma 3.

Lemma 2. Let a set $y_{ij}$ that is problem solving be defined, (5) – (8), then set

$$
x_{ij} = y_{i,j+n}, i = 1..n, j = 1..m
$$

is the problem solving (1) – (4).

Proof.

By Lemma 1

$$
F_C = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} = 2 \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij} y_{ij} + 2P \sum_{i=1}^{n+m} y_{ii}
$$

(15)

Every $y_{ij} = 1$ above the main diagonal belongs to area B.

$$
F_C = 2 \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} y_{ij,n} + 2P \sum_{i=1}^{n+m} y_{ii} = 2P \left[ 2 \sum_{i=1}^{n+m} \sum_{j=i+1}^{n+m} y_{ij} + \sum_{i=1}^{n+m} y_{ii} \right] - 2 \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_{ij}
$$

(16)

By restrictions (6 - 7):

$$
2 \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} y_{ij} = n + m .
$$

(17)

In this way,

$$
F_C = 2P(n + m) - 2F_A.
$$

(18)

Different solutions $y_{ij}$ give all possible solutions $x_{ij}$, so minimal $F_C$ is approached with maximal $F_A$.

Lemma 3. Let a set $x_{ij}$ that is problem solving be defined, (1) – (4), then set

$$
y_{ij} = \begin{cases} 
  x_{i,j-n}, \text{if } i \leq n, j > n \\
  x_{i,j-n}, \text{if } i > n, j \leq n \\
  1 - \sum_{k=1}^{m} x_{ik}, \text{if } i = j, i \leq n \\
  1 - \sum_{k=1}^{n} x_{kj}, \text{if } i = j, j \leq n \\
  0, \text{ in other ways}
\end{cases}
$$

(19)

is the solving of problem (5) – (8).
Proof.

As shown in lemma 1, $\forall y_{ij}, i, j = 1..n + m, F_C = 2P(n + m) - 2F_A$. Let’s show that a set constructed by formulas (10) gives exactly the same value.

$$F'_C = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij}y_{ij} = \sum_{i=1}^{n+m} \sum_{j=i+1}^{n+m} c_{ij}y_{ij} + 2P\sum_{i=1}^{n} y_{ii} + 2P \sum_{j=n+1}^{n+m} y_{jj} + \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c_{ij}y_{ij} =$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}x_{ij} + 2P\left(n - \sum_{k=1}^{m} \sum_{j=1}^{m} x_{jk}\right) + 2P\left(m - \sum_{k=1}^{n} \sum_{j=1}^{m} x_{kj}\right) + \sum_{j=1}^{n+m} \sum_{i=1}^{n+m} c_{ij}x_{ij} =$$

$$= 2P \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}x_{ij} + 2P(n + m) - 2P \left[2\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij}\right] = 2P(n + m) - 2F_A = F_C$$

(20)

It is easy to verify that solution (10) satisfies constraints (6) – (8).

We now turn to the description of the incremental step.

Suppose that we have problem (5) – (8) with matrix $C$ and its solution $y_{ij}$.

It is required to add to the matrix $C$ column and row with number $n+m+1$. Column is a new vertex in the graph, and the line is its twin. The arguments for the opposite case are similar. To obtain a solution to the new problem with the extended matrix, it suffices to construct an alternating chain of minimal cost from the vertex of the corresponding column $n+m+1$ to row $n+m+1$. The trivial solution is a chain consisting of one edge $(n+m+1, n+m+1)$ with weight $2P$.

On the real vertices of the graph, we perform a direct search, and on their twins - the reverse. In order for these paths to connect, they need to go through one of the edges $(i, i)$, since only these edges connect the real vertices and their counterparts. The search stops as soon as it visits the top-double.

Theorem 1. The above algorithm correctly performs the incremental step for problem (1) – (4).

Proof.

Using Lemma 3 on the existing solution of problem (1) – (4), we can construct a solution to problem (5) – (8). The correctness of the algorithm for solving problem (5) – (8) follows from the correctness of the Successive Shortest Path algorithm [13, 10] and the Dijkstra bidirectional algorithm [11]. Having obtained a solution to problem (5) – (8) with an added vertex using Lemma 2, we can also obtain a new solution to problem (1) – (4).

We now consider the decrement step.

Suppose you want to delete the column and row $n+m$ from matrix $C$. By rearranging rows and columns without breaking symmetry, you can ensure that any column $j=n+1,\ldots, n+m$ is on the last position.

5. Conclusion

Testing has shown that the average number of elements of an alternating chain when performing an incremental step is of the order $O(n \times (1 - \sigma))$, where $n$ – the number of vertices in the extensible part of the graph, and $\sigma$ - average sparsity of graph. The average number of operations required to perform an incremental step is of the order $O(n \times m \times (1 - \sigma)^2)$, where $m$ – the number of vertices in the fraction of the graph opposite extensible.

Thus, the solution of the assignment problem by the modified Hungarian method led to a decrease in the asymptotic complexity of the problem.

References

[1] Korobov P N 2002 Mathematical programming and modeling of economic processes (St. Petersburg)
[2] Paley I A 2008 Linear Programming (Moscow: Eksmo)
[3] Ford L R 1965 Streams in networks (Moscow: Mir)
[4] Boborykin V A 1986 Mathematical methods for solving transport problems (St. Petersburg:
[5] Kuhn H W 1955 The Hungarian method for the assignment problem Naval Research Logistics Quarterly 2 83–97
[6] Edmonds J and Karp R M 1972 Theoretical improvements in algorithmic efficiency for network flow problems JACM Journal of the Association for Computing Machinery 19 pp 248-264
[7] Hamdy A T 2001 Operations Research: An Introduction (Moscow: World)
[8] Toroslu I H and Ucoluk G 2007 Incremental assignment problem Information Sciences 6
[9] Ayorkor Korsah G, Stentz A and Bernardine Dias M 2007 The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs Tech. Report CMU-RI-TR-07-27
[10] Kuznetsov V A and Karavaev A M 2007 Optimization on graphs (algorithms and implementation) (Petrozavodsk: PetrSU)
[11] Danzig J 1966 Linear programming of its generalization and application (Moscow: Progress)
[12] Dijkstra E W 1959 A note on two problems in connexion with graphs Numerische Mathematik 1 269-71
[13] Engquist M A 1982 Successive Shortest Path algorithm for the Assignment Problem INFOR 20 370-84