[引用格式] 赵伟康, 韩一娜, 张浩宇, 等. 多假设跟踪中的高效匈牙利算法研究[J]. 水下无人系统学报, 2018, 26(5): 444-448.

多假设跟踪中的高效匈牙利算法研究

赵伟康[1], 韩一娜[1], 张浩宇[1], 杨益新[1], 刘清宇[2]

(1. 西北工业大学 航海学院, 陕西 西安, 710072; 2. 海军研究院, 北京, 100073)

摘 要: 作为多假设跟踪技术中的一项核心算法, 匈牙利算法占用了多假设跟踪中大部分的运算时间。考虑到多假设跟踪中的指派问题具有其特殊性, 即其效率矩阵是稀疏的, 文中提出了一种对效率矩阵进行降维的处理方法, 给出了运算流程, 对比了该方法与传统匈牙利算法在处理较大效率矩阵时的耗时, 结果表明, 在确保与传统匈牙利算法结果一致的前提下, 该方法能够大幅度降低运算量。

关键词: 多假设跟踪; 匈牙利算法; 指派问题

中图分类号: TJ67; O224 **文献标识码:** A **文章编号:** 2096-3920(2018)05-0444-05

DOI: 10.11993/j.issn.2096-3920.2018.05.011

An Improved Hungarian Algorithm for Multiple Hypothesis Tracking

ZHAO Wei-kang[1], *HAN Yi-na*[1], *ZHANG Hao-yu*[1], *YANG Yi-xin*[1], *LIU Qing-yu*[2]

(1. School of Marine and Technology, Northwestern Polytechnical University, Xi'an 710072, China; 2. Naval Research Academy, Beijing 100073, China)

Abstract: Hungarian algorithm is a core algorithm in multiple hypothesis tracking technology, and it takes up most of the computation time in multiple hypothesis tracking(MHT). Considering that the assignment problem in the MHT has particularity, i.e., the efficiency matrix is sparse, this paper proposes a method for reducing the dimension of the efficiency matrix, and the computation flow is given. Comparison of the time consumption between the proposed method and traditional Hungarian algorithm in dealing with larger efficiency matrix shows that the proposed method greatly reduces the amount of computation while ensuring consistency with the results of the Hungarian algorithm.

Keywords: multiple hypothesis tracking(MHT); Hungarian algorithm; assignment problem

0 引言

在理想假设条件下, 多假设跟踪(multiple hypothesis tracking, MHT)算法被认为是处理数据关联的最优方法[1]。区别于其他所有数据关联技术, MHT 算法对所有可能的关联方案都维持一个假设, 并用一个树状结构来保存和管理这些假设, 该假设树随着扫描周期的推进进行横向和纵向的生长, 总体规模呈指数增长的趋势, 因此原始的 MHT 算法本质上是一种穷举寻优的方法。庞大的假设树为 MHT 算法带来了很大的运算量, 作为一种应用于工程实践中的算法, 需要抑制这种极快的问题规模增长, 目前成熟的 MHT 架构中包含了基于 Murty 算法[2]的 *K*-best 策略以及 *N*-scan 的枝剪策略[3], 前者用于控制假设树横向规模, 后者用来限制假设树深度。其中 *K*-best 策略用以保留同一个假设下概率最大的 *K* 个子假设而不是所有的子假设, 因此这就涉及到如何获得关联场景下概率最大分配的问题, 即 MHT 中的指派问题。匈牙利算法[4]目前在很多问题的求解中被应

收稿日期: 2016-11-19; 修回日期: 2016-12-18.
基金项目: 国家重点研发计划(2016YFC1400200); 国家自然科学基金面上项目(61671388).
作者简介: 赵伟康(1995-), 男, 在读硕士, 主要研究融合跟踪算法。

用[5-10]，也存在一些对匈牙利算法的改进策略。但 MHT 中的指派问题有其特殊性，即其效率矩阵是稀疏的，这提供了改进的空间。针对 MHT 中指派问题的特殊性，文中提出了一种先将原效率矩阵降阶并运行匈牙利算法然后再还原成原效率矩阵对应解的方法，在不破坏 MHT 架构的基础上可大幅度降低运算量。

1　指派问题与匈牙利算法

匈牙利算法最初由匈牙利数学家 Edmonds 于 1965 年提出，用于解决二分图匹配问题，后来该方法被推广应用到了带有权值的二分匹配问题，即指派问题中。指派问题指的是在一个矩阵中寻找若干个不冲突的元素，使得他们的和最大或最小，在求最大值的问题中该矩阵称为效率矩阵。指派问题和匈牙利算法是运筹学中的经典案例。

图 1 表示效率矩阵是方阵以及非方阵时的指派问题，可以看出，当问题规模上升后，指派问题的复杂度增长很快。

2	3	**1**
4	2	3
1	**5**	3

2	3	1	**4**
4	2	3	1
1	**5**	3	4

图 1　方阵和非方阵中的最大指派方案

Fig. 1　Maximum assignment schemes for square and non-square arrays

匈牙利算法的基本步骤如下[5]。

步骤 1：获得指派问题的效率矩阵 $\boldsymbol{A}_0(n\times n)$；

步骤 2：首先从效率矩阵每行减去该行最小的元素，再从每列减去该列最小的元素，得到每行每列都至少有 1 个 0 元素的矩阵 $\boldsymbol{A}_1$；

步骤 3：寻找最少的直线覆盖 $\boldsymbol{A}_1$ 中的 0 元素得到 $\boldsymbol{A}_2$，如果最少直线数等于 n，转入步骤 5，否则转入步骤 4；

步骤 4：$\boldsymbol{A}_2$ 中未被覆盖的元素减去这些元素中最小的元素，同时在直线的交点加上该元素，得到矩阵取代 $\boldsymbol{A}_1$ 转入步骤 3；

步骤 5：从 0 元素最少的行或列开始指派，直到各行各列都存在指派，得到最优指派方案 $\boldsymbol{E}$。

对于效率矩阵不是 n 阶方阵时的情况，需要向矩阵中加 0 补足成 1 个方阵。匈牙利算法的主要运算集中在步骤 3 与步骤 4，在效率矩阵较大的时候需要反复循环步骤 3 与步骤 4，且算法何时跳出循环进入步骤 5 是不可预知的，因此描述其算法复杂度时使用的是最差的情况，此时的时间复杂度达到了 $o(n^3)$，这表明匈牙利算法的运算量受问题规模影响很大。匈牙利算法相当于提供了式(1)表示的函数，在效率矩阵具有一般性时该算法是很高效的

$$(\boldsymbol{E},s)=\mathrm{Hung}(\boldsymbol{A}) \tag{1}$$

式中：$\boldsymbol{A}$ 表示效率矩阵；$\boldsymbol{E}$ 表示指派方案，是与 $\boldsymbol{A}$ 同阶的逻辑矩阵，其中逻辑 1 表示 $\boldsymbol{A}$ 矩阵中对应位置的元素被算法选中，0 表示没有选中；s 为方案 $\boldsymbol{E}$ 对应的总效率。一般来说，认为在对效率矩阵没有任何先验认识的条件下，匈牙利算法是解决指派问题的精确且高效的方法。

2　MHT 指派问题

MHT 算法一般包含假设生成、假设组合和枝剪、假设管理等过程，指派问题发生在假设的组合与枝剪过程，此时 MHT 需要获取当前测量值所有关联情况的假设，而将这些假设全部列举出来是不现实的。针对这一点，指派问题的效率矩阵提供了一种直观表示所有假设的方法，它依赖于 MHT 中的几条基本假设：1) 同一个测量只能关联于当前的 1 个轨迹，或成为杂波(虚警)，或成为新轨迹的起点；2) 每个活动的轨迹在每个周期最多只关联于 1 个测量，或被判断为漏报；3) 所有杂波和新轨迹起点间没有关联性。

以上 3 个假设保证了 MHT 数据关联问题为指派问题，MHT 中习惯使用后验的概率密度作为指派问题效率矩阵的关联效率，因此，一般 MHT 问题的效率矩阵具有如图 2 的形式[1]。

	T_1	T_2	FA_1	FA_2	FA_3	NT_1	NT_2	NT_3
y_k^1	l_{11}	l_{11}	$\lg\beta_{FA}$	×	×	$\lg\beta_{NT}$	×	×
y_k^2	l_{21}	l_{22}	×	$\lg\beta_{FA}$	×	×	$\lg\beta_{NT}$	×
y_k^3	l_{31}	l_{32}	×	×	$\lg\beta_{FA}$	×	×	$\lg\beta_{NT}$

图 2　常见多假设跟踪效率矩阵形式

Fig. 2　A common efficiency matrix in multiple hypothesis tracking(MHT)

决定效率矩阵规模的是本次扫描的测量集 $\{y_k^i, i=1,2,\cdots,n\}$，其中 k 表示第 k 次扫描，i 表示测量的编号，共 n 个测量值。T_1 与 T_2 表示当前活动的轨迹；FA_1，FA_2 和 FA_3 表示与测量集对应的虚警

假设; NT_1, NT_2和NT_3表示与测量集对应的新轨迹假设。矩阵中的元素

$$l_{ij}=\lg\left\{\left[P_D^j p_{k|k-1}^j(y_k^i)\right]\Big/\left(1-P_D^j P_G^j\right)\right\} \tag{2}$$

表示对应测量与假设关联的概率密度的对数; “×”表示禁止指派, 可设定为负无穷大以保证不会被指派, 相同规模的效率矩阵中, “×”越多的矩阵越容易在匈牙利算法下得到结果; $\lg\beta_{FA}$与$\lg\beta_{NT}$分别表示杂波空间密度的对数以及新轨迹数空间密度的对数, 这 2 个密度在一般的跟踪情景中都是全局的, 因此用常量表示。在 MHT 的效率矩阵中任何一个行列没有冲突的指派方法都可以成为合法的假设, 因此该矩阵实际上就可以表示所有的合法假设。这样每种指派方案的总效率就成为假设的得分, 该得分是对假设树进行枝剪的依据, 所以 MHT 需要寻找所有总效率最大的指派方案, 以保证在枝剪中使得那些发生概率很高的假设可以保留下来。MHT 中的最优假设寻找问题就被建模成一个标准的指派问题。

因此可以总结出, MHT 的指派问题对应的效率矩阵是由 1 个完整矩阵和 2 个对角阵组成。另外考虑到 MHT 问题中每个周期的测量数量往往远大于活动的轨迹个数, 因此效率矩阵的第 1 个部分常常是 1 个行数大于列数的矩阵。

3　MHT 指派问题改进方法

3.1　问题的数学化

将效率矩阵从 MHT 情景中剥离出来, 重新表示成如下更为普遍的形式:

$$\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1m} & b_1 & \times & \cdots & \times & c_1 & \times & \cdots & \times \\
a_{21} & a_{22} & \cdots & a_{2m} & \times & b_2 & \cdots & \times & \times & c_2 & \cdots & \times \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nm} & \times & \times & \cdots & b_n & \times & \times & \cdots & c_n
\end{bmatrix}$$

图 3　MHT 效率矩阵

Fig. 3　Efficiency matrix in MHT

图 3 中表示的矩阵最终是一个 $n\times(m+2n)$的矩阵, 称为原始效率矩阵 $\boldsymbol{P}$, 将矩阵 $\boldsymbol{P}$ 划分成 $n\times m$ 的矩阵 $\boldsymbol{A}$, n 阶对角阵 $\boldsymbol{B}$ 和 $\boldsymbol{C}$, 即 $\boldsymbol{P}=[\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]$。显然将矩阵 $\boldsymbol{P}$ 直接输入到式(1)表示的标准匈牙利算法可以获得最大效率和对应的指派方案, 但匈牙利算法在该类型的效率矩阵中产生了很多无意义的运算。主要原因是矩阵 $\boldsymbol{P}$ 中大部分的信息集中在矩阵 $\boldsymbol{A}$ 中, 而矩阵 $\boldsymbol{P}$ 的规模常常远大于矩阵 $\boldsymbol{A}$, 考虑到匈牙利算法较大的运算复杂度, 直接对矩阵 $\boldsymbol{P}$ 运行匈牙利算法效率很低。

3.2　改进的方法

首先对该效率矩阵定义一个数列

$$Q=\{\max(b_i,c_i)\},i=1,2,\cdots,n \tag{3}$$

式中: $\boldsymbol{Q}$ 表示矩阵[$\boldsymbol{B}$,$\boldsymbol{C}$]中每一行的最大值。如果仅仅考虑对矩阵 $\boldsymbol{A}$ 的指派, 则相当于矩阵 $\boldsymbol{A}$ 中每一个元素都拥有一个备用方案, 每一个指派既可以指派矩阵 $\boldsymbol{A}$ 中的该元素也可以指派该元素的备用值。那么, 那些矩阵 $\boldsymbol{A}$ 中比备用值小的元素是不会被选中的, 因为总可以通过将其替换成备用值来获得更大的效率。基于这一点认识, 首先可以对矩阵 $\boldsymbol{A}$ 中比备用值小的元素替换成其相应的备用值, 替换过的矩阵称为 $\boldsymbol{A'}$。为此同时定义一个逻辑矩阵 $\boldsymbol{R}$, 用以记录被替换过的元素。

$$\begin{aligned}
A'(i,j)&=\begin{cases}Q(i), A(i,j)<Q(i)\\ A(i,j), A(i,j)\geqslant Q(i)'\end{cases}\\
R(i,j)&=\begin{cases}1, A(i,j)<Q(i)\\ 0, A(i,j)\geqslant Q(i)'\end{cases}\\
i&=1,2,\cdots,n; j=1,2,\cdots,m
\end{aligned} \tag{4}$$

另外, 如果矩阵 $\boldsymbol{R}$ 中某一列全部是 1, 意味着该列中所有的元素都比备用值小, 则他们全部都不可能被最优指派选中, 可以从矩阵 $\boldsymbol{A'}$ 和矩阵 $\boldsymbol{R}$ 中删除这些对应列。事实上, 在 MHT 环境中, 如果测量出现了漏报的情况, 很容易导致这种现象, 将这些行删除能有效降低矩阵 $\boldsymbol{A'}$ 的规模, 从而减少基准匈牙利算法的运算量。记经过删除的矩阵 $\boldsymbol{A'}$ 的大小为 $n\times m'$。

对矩阵 $\boldsymbol{A'}$ 运行匈牙利算法并不能得到最优指派, 因为若选择了 $\boldsymbol{A'}$ 中被替换过的元素, 实际上该元素并不会与该列的其他元素冲突, 而对于 $\boldsymbol{A'}$ 来说该列的其他元素却不会被选中了。

图 4 中, 直接对 $\boldsymbol{A'}$ 运行匈牙利算法, 结果显然是 10+7, 然后第 3 行取备用值为 3, 总效率为 20。但是, 如果第 1 行选择小一点的备用值 6, 可以不占用第 2 列, 那么第 3 行就不需要选择很小的备用值, 而是选择 6, 这样获得 10+6+6=22 的总效率。这一效应是不能忽视的, 因为 $\boldsymbol{A'}$ 的最优指派中总无法避免通过将其中的一个或若干个值选为较差的备用值可以获得相对更优解的可能。显然可以通过划去矩阵 $\boldsymbol{A'}$ 中的若干行然后重新

运行匈牙利算法，并保留每种情况的总效率，最优的指派必然是包含在这些情况中的。然而划去的方案随着规模指数增长，这样做可能会导致比直接对原始效率矩阵求解更大的计算量。

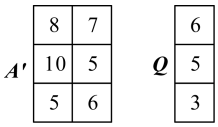

图 4　上述方法的一种特例

Fig. 4　A special case of the above method

考虑到对于一个 $n\times m'(n>m')$ 的矩阵 $\boldsymbol{A}'$，对其运行匈牙利算法会有 m' 个值被选中，而前面介绍的划去若干行的方法中至多需要划去 $m'-1$ 行才能保证遍历了所有有可能成为最优指派的情况。因此，文中提出一种向矩阵 $\boldsymbol{A}'$ 添加一个增广矩阵 $n\times(m'-1)$ 的方法，不需要遍历所有的情况就能获得最优指派。对于图 4 的情形，如果将矩阵 $\boldsymbol{Q}$ 并入 $\boldsymbol{A}'$ 中，对一个 3×3 的矩阵运行匈牙利算法就能获得最优指派，加入的矩阵 $\boldsymbol{Q}$ 相当于遍历了所有不划去和划去一行的情况，同理，如果向 $\boldsymbol{A}'$ 增广 $m'-1$ 个矩阵 $\boldsymbol{Q}$ 就能遍历所有至多划去 $m'-1$ 行的情况。因此，该方法需要将原始效率矩阵 $\boldsymbol{P}$ 处理成图 4 的形式，称为矩阵 $\boldsymbol{A}''$，$\boldsymbol{A}''$ 是一个 $n\times(2m'-1)$ 的矩阵，它的规模远小于原始效率矩阵 $\boldsymbol{P}$。如图 5 所示。

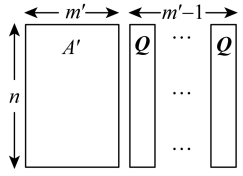

图 5　矩阵 $\boldsymbol{A}''$ 一般形式

Fig. 5　General form of matrix *A*”

对矩阵 $\boldsymbol{A}''$ 运行匈牙利算法，获得一个 $n\times(2m'-1)$ 的表示指派方案的逻辑矩阵

$$\boldsymbol{E}=\mathrm{Hung}(\boldsymbol{A}'') \tag{5}$$

矩阵 $\boldsymbol{E}$ 的前半部分表示对矩阵 $\boldsymbol{A}'$ 的指派，这些被选中的元素中被替换成备选值的元素需要重新校准到矩阵[$\boldsymbol{B}$,$\boldsymbol{C}$]的相应位置，而没有获得指派的行也需要在矩阵[$\boldsymbol{B}$,$\boldsymbol{C}$]中选择备用值。因此该方法的最后一步是根据矩阵 $\boldsymbol{E}$ 和 $\boldsymbol{R}$ 调整部分指派的位置，获得最终与矩阵 $\boldsymbol{P}$ 同阶的结果矩阵 $\boldsymbol{E}'$。

因此该方法包含了如下的步骤：

步骤 1：获得原始效率矩阵 $\boldsymbol{P}$；

步骤 2：将 $\boldsymbol{P}$ 拆分成 $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ 3 个矩阵；

步骤 3：按照式(3)的规则获得矩阵 $\boldsymbol{Q}$；

步骤 4：以式(4)的规则获得矩阵 $\boldsymbol{R}$ 和 $\boldsymbol{A}'$，移除 $\boldsymbol{R}$ 中全为 1 的列以及 $\boldsymbol{A}'$ 中对应的列；

步骤 5：对 $\boldsymbol{A}'$ 添加若干列获得增广矩阵 $\boldsymbol{A}''$；

步骤 6：对 $\boldsymbol{A}''$ 运行标准的匈牙利算法，获得对应的指派方案 $\boldsymbol{E}$；

步骤 7：为 $\boldsymbol{E}$ 中没有获得指派的行补充备选指派，形成对应于 $\boldsymbol{P}$ 的最终结果 $\boldsymbol{E}'$。

为了与上述步骤对照，图 6 给出一个具有代表意义的原始效率矩阵经过每个步骤，获得最终结果的过程。除了步骤 6 的其他步骤都是逻辑判断和矩阵的拆分组合，附加了这些步骤可以让输入匈牙利算法的矩阵从 $\boldsymbol{P}$ 减小到 $\boldsymbol{A}''$ 的规模，所以能达到减少运算量的目的。

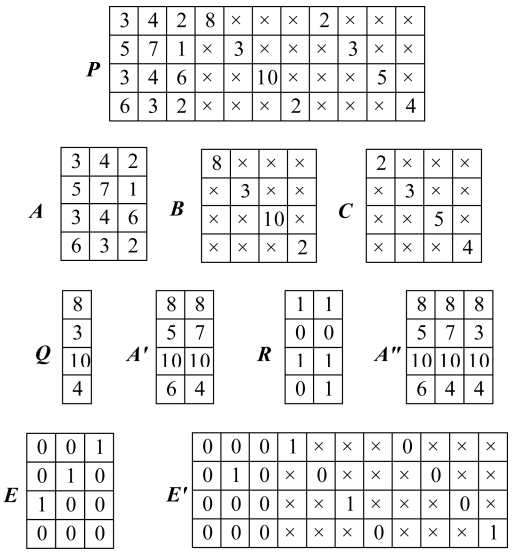

图 6　基于实例的运算流程

Fig. 6　Instance-based computation flow

4　性能分析与比较

4.1　理论分析

匈牙利算法是一类迭代算法，因此其运算量并不稳定，公认的 n 阶方阵的匈牙利算法的时间复杂度在最差的情况下是 $o(n^3)$，对于效率矩阵非方阵的情况，常见的方法是补零使其成为一个方阵[9]，因此对于一个 $n\times m(n>m)$ 的效率矩阵，

时间复杂度最差依旧是 $o(n^3)$，只是这时很难达到最差条件的，实际的运算时间远小于该值。对于这类矩阵存在一些改进的方法[10]，但是相比较补零的方法没有很明显的优势。影响 MHT 中指派问题规模的因素是当前的测量数 n 和当前活动的轨迹数 m，形成的效率矩阵 $\boldsymbol{P}$ 是 $n\times(2n+m)$的，理论上直接运行匈牙利算法的时间复杂度最差为 $o((2n+m)^3)$ 。而该方法将需要输入匈牙利算法的矩阵减小到 $n\times(2m'-1)$，在 n 远大于 m 和 m' 的假设下时间复杂度是 $o(n^3)$，m' 与 m 之间的关系取决于 MHT 中观测数据的漏报概率，漏报概率越大 m' 相对 m 来说越小。因此 n 与 m 之间相差越大，文中方法在时间复杂度上优越性越明显。

4.2　仿真与结果

n 表示一次扫描获得的量测数，m 表示 huod2 的轨迹数，在最理想的跟踪环境下效率矩阵的每一列会存在一个值远大于其他值，而且这些值不会出现在同一行，此时匈牙利算法的意义并不明显。仿真时考虑最不理想的情况，即量测中没有十分明显与轨迹关联的值，矩阵 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 中的元素为均匀分布的随机数，由此组成的原始效率矩阵 $\boldsymbol{P}$ 是对 2 个方法最不友好的。

为了更加直观地表示 2 种方法的运算量，采用控制变量的思想，在同一台计算机上用同样的编程语言使用 2 种方法解决同一个指派问题，对比两者的运算时间。

由表 1 可见，在 m 和 n 比较接近时，改进方法效率提升了近 1 倍，在 m 明显小于 n 时，改进方法的优势更加明显，能大幅下降输入匈牙利算法的矩阵规模，效率上升了数十倍。

表 **1**　不同规模问题的计算时间对比

Table 1　Computation time comparison for different scale problems

n/行	m/列	传统方法耗时/s	改进方法耗时/s
200	200	0.218 8	0.109 4
200	100	0.203 1	0.046 9
400	400	2.046 9	1.062 5
400	200	1.671 9	0.468 8
400	100	1.562 5	0.046 9
600	600	8.031 3	3.765 6
600	300	7.765 6	1.937 5
600	150	6.343 8	0.171 9

5　结束语

文中提出了一种适合在 MHT 中使用的改进的匈牙利算法，特点是运用在 MHT 中能保证获得与匈牙利算法完全一致的指派结果，而运算复杂度较后者有很大的降低。该方法嵌入到 MHT 算法中表现稳定。

参考文献：

[1] 翟海涛. 多假设跟踪算法研究及其应用[J]. 信息化研究. 2010, 36(8): 25-27.
Zhai Hai-tao. Reseach on Multiple Hypothesis Tracking Algorithm and Its Application[J]. Informatization Research, 2010, 36(8): 25-27.

[2] Murty K G. An Algorithm for Ranking All the Assignments in Order of Increasing of Cost[J]. Operations Research, 1968, 16: 682-687.

[3] Miller M L, Stone H S, Cox I J. Optimizing Murty's Ranked Assignment Method[J]. NEC Research Institute, Technical Report, 1997, 33(3): 851-862.

[4] 钱颂迪. 运筹学[M]. 北京: 清华大学出版社, 1998.

[5] Chin-Jung Huang. Integrate the Hungarian Method and Genetic Algorithm to Solve the Shortest Distance Problem[C]//2012 Third International Conference on Digital Manufacturing & Automation. Guilin, China: IEEE, 2012: 496-499.

[6] Patel R R, Desai T T, Patel S J. Scheduling of Jobs based on Hungarian Method in Cloud Computing[C]//2017 International Conference on Inventive Communication and Computational Technologies(ICICCT). Coimbatore, India: IEEE, 2017: 6-9.

[7] Yan Chao-bo, Zhao Qian-chuan. Advances in Assignment Problem and Comparison of Algorithns[C]//27th Chinese Control Conference. Kunming, China: IEEE, 2008: 607-611.

[8] 张新辉. 任务多于人数的指派问题[J]. 运筹与管理. 1997, 6(3): 20-25.
Zhang Xin-hui. Assignment Problem with Tasks More than the Number of Persons[J]. Operations Research and Management Science, 1997, 6(3): 20-25.

[9] 李延鹏, 钱彦岭, 李岳. 基于改进匈牙利算法的多技能人员调度方案[J]. 国防科技大学学报, 2016, 38(2): 144-149.
Li Yan-peng, Qian Yan-ling, Li Yue. Multi-skilled Labor Allocating Method Based on Improved Hungary Algorithm[J]. Journal of National University of Defense Technology, 2016, 38(2): 144-149.

[10] 马晓娜.“人少任务多”型指派问题的一种新算法[J]. 重庆工商大学学报(自然科学版), 2014, 31(12): 68-75.
Ma Xiao-na. A New Algorithm for Assignment Problems with “Tasks More Than the Number of Persons”[J]. Journal of Chongqing Technology and Business University: Natural Science Edition, 2014, 31(12): 68-75.

(责任编辑: 陈　曦)