一种任务分配问题的快速剪枝优化算法

马云红，井　哲，周德云

(西北工业大学 电子信息学院，陕西 西安　710072)

摘　要: 任务分配问题是运筹学中的一类规划问题，求解这类问题的比较经典的算法是匈牙利算法，但匈牙利算法在求解大规模任务分配时运算效率不高。文章提出了一种新的求解任务分配问题的方法——剪枝优化算法。算法通过逐步剔除已确定的部分分配方案对应代价矩阵元素，逐次降低分配问题的规模，从而实现快速求解全局任务分配问题。对于 n 个主体执行 n 个任务的分配问题，进行 $(n-1)$ 次操作就可以获得最优解。论文进行了相应的仿真，将文章提出的算法和匈牙利算法做了比较。仿真结果表明，该算法与传统匈牙利算法计算结果一致，但计算耗时远远小于匈牙利算法，即该算法大大提高了任务分配问题的求解速度。

关　键　词: 算法，任务分配，运筹学，剪枝优化算法，无人机

中图分类号: O22　　**文献标识码:** A　　**文章编号:** 1000-2758(2013)01-0040-04

在日常生活和工作中，我们经常会面临最佳任务分配问题: 比如说生产任务与生产设备之间的最佳分配; 一组人完成一组任务的任务分配等等。这类问题通常都可以归结为这样的描述: 设有 n 项任务需要派出 n 个主体去执行，已知不同主体完成不同的任务所要付出的成本或者代价不同，寻找最佳的分配方案，使得完成所有任务的总成本最小。这个问题是一个典型的运筹学范畴的标准任务指派问题，即指派不同的主体分别完成不同的任务，每个主体完成一项任务，使总的完成任务的代价最小。这类问题在其它领域也很常见，比如军事应用中的武器分配问题、生产流程中资源配置和设备安排问题等等。本文在分析这类问题的经典算法——匈牙利算法的基础上，提出了一种任务指派问题的快速求解方法 - 剪枝优化算法，该算法大大提高了求解任务指派问题，尤其是大规模任务指派问题的求解速度。

1　标准任务指派问题模型

任务指派问题的标准形式是: 以作战时的武力分配为例[2,3]: 设 n 个无人机要攻击 n 个目标，如果已经预先计算出第 i 架无人机攻击第 j 个作战目标的代价，要求确定无人机和目标的分配方案，以最小的代价完成总的作战任务，同时必须满足以下约束条件: 每个无人机都必须执行一项任务，每个任务都必须被执行。

我们记矩阵 $\boldsymbol{C}=(c_{ij})_{n\times n}$ 为指派问题的代价矩阵，c_{ij} 为第 i 架无人机攻击第 j 个作战目标的代价。以矩阵

$$\boldsymbol{X}=(x_{ij})_{n\times n}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}$$

表示任务分配的方法。元素 x_{ij} 的值为 0 或者 1。如果 $x_{ij}=1$ 表示安排第 i 架无人机攻击第 j 个作战目标; $x_{ij}=0$ 表示不安排第 i 架无人机攻击第 j 个作战目标。

这样，以最小代价完成作战任务的无人机任务分配问题的数学模型可以描述为

$$J^*_{X^*}(\boldsymbol{C},\boldsymbol{X})=\min_{X\in\Gamma} J(\boldsymbol{C},\boldsymbol{X}) \tag{1}$$

收稿日期: 2012-05-15　　**基金项目:** 西北工业大学 E 之星基金资助

作者简介: 马云红(1972—)，女，西北工业大学副教授，主要从事优化算法、飞行器任务规划和智能控制的研究。

$$J(C,X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{2}$$

约束条件为

$$\sum_{i=1}^{n} x_{ij} = 1,\quad j = 1,2,\cdots,n \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1,\quad i = 1,2,\cdots,n \tag{4}$$

$$x_{ij} = 0,1 \qquad i,j = 1,2,\cdots,n \tag{5}$$

式中,约束条件(3)表示每架无人机要执行一个任务;约束条件(4)表示每个任务被执行且被执行一次;约束条件(5)表示分配方案的元素值只能取 0 或者 1。且表示的含义如下:

$$x_{ij} = \begin{cases} 1 & V_i \to T_j \\ 0 & \text{其他} \end{cases} \quad i,j = 1,2,\cdots,n$$

X 表示可行的分配矩阵,对应着一种分配方案,Γ 为所有可能的分配方案集合。$J(C,X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$ 表示 X 分配方式下的目标函数,$J^*_{X^*}(C,X) = \min_{X\in\Gamma}(C,X)$ 为全局最优目标函数值,它对应的 X^* 矩阵为最佳分配方案。

2　匈牙利算法

匈牙利算法[1]主要用于求解标准指派问题。关于标准指派问题的最优解具有这样的性质:若从代价矩阵 $(c_{ij})_{n\times n}$ 的一行(列)各元素中加上或减去一个数,得到新矩阵 $(c'_{ij})_{n\times n}$,以 $(c'_{ij})_{n\times n}$ 为代价矩阵求得的指派基础解集 x_{ij} 和原矩阵求解一致,只是代价值发生改变。

利用这个性质,匈牙利算法在求解最优分配方案的过程中,多次对原代价矩阵进行变换,使得原代价矩阵变换为含有很多 0 元素的新代价矩阵。如果在代价矩阵 $(c'_{ij})_{n\times n}$ 中可以找出 n 个独立的 0 元素(不同行不同列的 0 元素),其对应的目标函数中代价 $z' = 0$,即保证了总的代价最小,选中的独立 0 元素对应的原来的 $(c_{ij})_{n\times n}$ 的元素组合为原分配问题的最优分配解。

匈牙利算法的求解中需要进行多次代价矩阵的变化以及零元素的寻找、标记、选择及删除,且逻辑复杂,运算效率较低。本文在分析标准任务指派问题的特点的基础上,提出了一种求解标准指派问题的快速优化算法——"剪枝"降阶优化算法,简称为剪枝优化算法。

3　剪枝优化算法的理论与计算步骤

3.1　相关理论

相关的定义:

定义 1　$\Gamma_1 = \{X \mid x_{lk} = 1\}$,$\Gamma_2 = \{X \mid x_{lm} = 1\}$,很明显,$\Gamma_1$、$\Gamma_2$ 分别是一种确定了部分分配 $x_{lk} = 1$ 或 $x_{lm} = 1$ 的降阶指派问题。$\Gamma_1,\Gamma_2 \subset \Gamma$。

定义 2　$E = J^*_{X\in\Gamma}(C,X)$,为全局最优分配的目标函数值。

定义 3　$E_1 = \min_{X\in\Gamma_1} J(C_1,X)$,$E_2 = \min_{X\in\Gamma_2} J(C_2,X)$。

C_1 和 C_2 的计算如下(将已经确定的部分分配方案对应的代价元素值设为 0)

即令 $$C_1 = C_{ij}^{lk} = \begin{cases} 0 & i = l \quad \text{或者} \quad j = k \\ c_{ij} & \text{其它} \end{cases} \tag{6}$$

式中 l 为

$$l = \arg\max_i (c_{ik} - c_{im}) \times c_{ik},$$
$$i = 1,2,\cdots,n \quad k \neq m; k,m = 1,2,\cdots,n \tag{7}$$

这时 C_1 的 0 元素不再进行优化分配。很明显,E_1 变为一个求解 $(n-1)$ 个无人机对 $(n-1)$ 个目标的分配问题,可行空间缩小,这里称为剪枝。

同理令

$$C_2 = C_{ij}^{lm} = \begin{cases} 0 & i = l \quad \text{或者} \quad j = m \\ c_{ij} & \text{其它} \end{cases}$$

式中 l 为

$$l = \arg\max_i (c_{ik} - c_{im}) \times c_{ik} \tag{8}$$
$$i = 1,2,\cdots,n \quad k \neq m; k,m = 1,2,\cdots,n$$

这时 C_2 的 0 元素不再进行优化分配,很明显,E_2 也变为一个求解 $(n-1)$ 个无人机对 $(n-1)$ 个目标的分配问题,即:E_1 和 E_2 是 2 个低一阶指派问题的最优值。它们的代价矩阵分别为 C_1 和 C_2。

如果,求解分配方案使代价值最小,则有

$$J^*_{1\,X_1^*\in\Gamma_1}(C,X_1) + c_{lm} = J^*_{X\in\Gamma}(C,X) \tag{9}$$

即:在降阶后得到的 $(n-1)$ 阶的分配问题的值是确定的情况下,即

$$l = \arg\max_i (c_{ik} - c_{im}) \times c_{ik}$$

有

$$J^*_{1\,X_1^*\in\Gamma_1}(C,X_1) + c_{lm} \leqslant J^*_{1\,X_1^*\in\Gamma_1}(C,X_1) + c_{lj(j\neq m)} \tag{10}$$

即 $J^*_{1\,X_1^*\in\Gamma_1}(C,X_1) + c_{lm} = J^*_{X\in\Gamma}(C,X)$,$c_{lm}$ 为选中的分配方案,并确定了这部分最优分配方案的解,则将其

对应的行列剔除。即令

$$x_{lm} = 1, \begin{cases} x_{lj} = 0, j = 1,2,\cdots,n, j \neq m \\ x_{im} = 0, i = 1,2,\cdots,n, i \neq l \end{cases} \quad (11)$$

同理如果以代价值最大作为最优分配指标，则选择(8)式中 $x_{lk} = 1$ 为最优方案的部分解，并将其对应的行列元素剔除，于是就变成了降低一阶的最优方案求解问题。依次执行，直到代价矩阵只剩一个元素，全局最佳方案得解。

3.2 剪枝优化算法步骤

按照本文设计的剪枝优化方法进行代价最小的任务分配的具体步骤为:

1) 初始化 $x_{ij} = 1, i, j = 1,2,\cdots,n$。

2) 确定满足

$l = \arg\max_{i}(c_{ik} - c_{im}) \times c_{ik} \quad k \neq m; i,k,m = 1,2,\cdots,n$ 的行号 l; 选择 $x_{lm} = 1$，同时将对应的 c_{lm} 标记为 true，表示选中。

3) 剪枝。令 $\begin{cases} x_{lj} = 0, j = 1,2,\cdots,n, j \neq m \\ x_{im} = 0, i = 1,2,\cdots,n, i \neq l \end{cases}$。且将与 c_{lm} 同行同列的其它元素标记为 false，这个操作是进行剪枝的操作。此时，代价矩阵中的有效代价值矩阵规模缩小了一阶。

4) 对降阶后的代价矩阵中没有进行标记的矩阵中的元素重复进行步骤 2) 和步骤 3)，直至最后有效代价矩阵为 1×1 矩阵，即一个元素时，将该元素标记为 true，其对应的 X 阵元素值为 1，问题得解。

如果进行目标函数值最大的任务分配，则根据(8)式选择 $x_{lk} = 1$，进行剪枝优化。

经过分析可以看出，本文提出的算法求解($n \times n$)矩阵的分配问题时，只需循环进行($n - 1$)次就可以得到最优解。本文对提出的算法经过理论分析和推导，简化了计算中的逻辑判断，使得该方法逻辑简单，计算效率高，便于计算机实现。

3.3 剪枝优化算法实例解析

对于某一个典型的指派问题，假设各个主体完成各项任务的代价矩阵已经给出如下

$$\boldsymbol{C} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 2 \\ 3 & 1 & 7 \end{bmatrix}$$

采用文中的剪枝算法，经过 2 次剪枝就可得到最优分配解: 每步剪枝得到的解如下所示

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \text{第 1 次剪枝: } \boldsymbol{X} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

第 2 次剪枝 $\boldsymbol{X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$。

4 算法验证与比较

本文对提出的算法进行了大量的仿真验证，并与传统的匈牙利算法进行了仿真比较，比较结果列在如下的表 1 中。

表 1 算法的求解时间与优化结果比较

c_{ij}	匈牙利算法/s	剪枝优化算法/s	速度提高/倍数
5 × 5	0.039 273	0.004 278	9.180 2
8 × 8	0.039 650	0.004 985	7.953 9
12 × 12	0.039 859	0.005 310	7.506 4
20 × 20	0.042 913	0.006 285	6.827 8
30 × 30	0.148 499	0.024 527	6.054 5
50 × 50	0.182 077	0.061 419	2.964 4

以上表格中，5 × 5 表示代价矩阵的维数，用以解决 5 个任务，5 个作战无人机的任务指派问题，同理，50 × 50 表示代价矩阵的维数，用以解决 50 个任务，50 个作战无人机的任务指派问题。数据表明，本文算法的计算速度明显高于匈牙利算法。仿真结果见图 1。

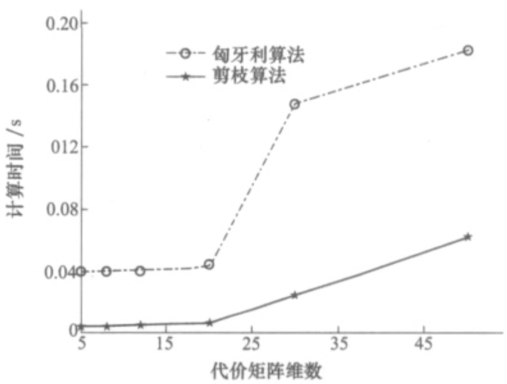

图 1 不同算法求解不同规模指派问题的计算时间比较

5　结　论

1）本文提出的剪枝优化方法，是通过逐步获得部分最优解的方法，将可行解的范围逐步缩小，并最后获得问题的全局最优解。论文对提出的算法进行了理论分析和证明，该算法求解过程简单，计算逻辑清晰，便于计算机实现。仿真结果显示，算法明显提高了标准指派问题的求解速度。

2）需要说明的是: 本文算法的适应范围是用于求解代价矩阵中元素均为非负值的问题。如果代价矩阵中既有正数又有负数，则首先需要找出最小的负数，再把每一个代价元素都增加相同的正值以保证矩阵的所有元素值均为非负数，这时再根据文中的算法进行求解。根据匈牙利算法的证明，对矩阵元素统一增加一个正数不影响问题的最优解。

参考文献:

[1] Winston Wayne L. Operation Research Application and Algorithms. Beijing，Tsinghua University Press，2011
[2] Wei Kangy，Andrew Sparks. Task Assignment in the Cooperative Control of Multiple UAVs. AIAA-2003-5583
[3] Corey Schumacher，Phillip R Chandler. Path Elongation for UAV Task Assignment. AIAA-2003-5585

A Faster Pruning Optimization Algorithm for Task Assignment

Ma Yunhong[1]，Jing Zhe[1]，Zhou Deyun[1]
(Department of Electronics Engineering，Northwestern Polytechnical University，Xi´an 710072，China)

Abstract: To our knowledge，the Hungary algorithm is not satisfactory for solving a large scale task assignment problem in the fields of operation research because it requires several changes in cost matrix and seeks，selects and deletes the labels of a zero element. Hence，we propose what we believe to be a faster pruning optimization algorithm to solve the problem. We reduce the task assignment scale by pruning the elements relative to the partial optimal resolution in each operation; the pruning optimization algorithm can thus obtain the optimal solution with only $(n-1)$ times of operation for a n to n task assignment. We simulate the pruning optimization algorithm and compare it with the Hungary algorithm. The simulation results and their comparison given in Table 1 and Fig. 1 show preliminarily that our pruning optimization algorithm obtains the same calculation results as the Hungary algorithm，it takes far less calculation time than the Hungary algorithm，thus quickening the speed for solving a task assignment problem.

Key words: algorithms，task assignment; operation research，pruning optimization，unmanned aerial vehicles (UAV)