

公平性指派问题及均值逼近求解算法*

何胜学

(上海理工大学管理学院, 上海 200093)

摘要 为了平衡任务指派后代理之间的工作负荷, 建立了公平性指派优化模型, 并给出了一种求解问题全局最优解的数值方法. 将指派后各代理工作负荷与平均负荷的差的平方和作为工作负荷公平性的度量指标, 结合经典指派问题约束建立了公平性指派模型. 以工作负荷矩阵是否可以分解为两个特殊矩阵之和对问题进行了分类, 指出了一般情况下公平性指派问题属于一类特殊的三次指派问题. 通过有规律遍历工作负荷的可行取值范围, 将问题求解转化为一维搜索内嵌求解系列经典线性指派问题. 从理论上证明了均值逼近算法的合理性, 同时指出算法的时间复杂度为问题规模的四次多项式时间. 通过与商业优化软件 Lingo 的计算结果比较, 证实新方法不仅可以得到问题的全局最优解, 而且所需的计算时间很短.

关键词 整数规划, NP-hard 问题, 逼近算法, 组合优化, 指派问题.

MR(2000) 主题分类号 97M40, 97N80

DOI 10.12341/jssms21499

Fair Assignment Problem and Its Average Approximation Algorithm

HE Shengxue

(Business School, University of Shanghai for Science and Technology, Shanghai 20093)

Abstract To balance the assigned working loads among agents, this paper formulated a fair assignment optimization model and proposed a numerical method for obtaining the global optimal solution. By defining the sum of squares of differences between assigned working loads of agents and the average load as the index of measuring the fairness of working load, this paper constructed the fair assignment model combining the constraints of the classic assignment problem. This study classified the problem according to whether the working load matrix could be decomposed into two special matrices and pointed out that a fair assignment problem is in general a type of cubic assignment problem. By regularly going through the feasible range of working load, this paper transformed the solution process into a one-dimensional search process that needs to solve a series of classic linear assignment problems. This

* 国家自然科学基金资助项目 (71801153, 71871144), 上海市自然科学基金项目 (18ZR1426200) 资助课题.

收稿日期: 2021-09-10, 收到修改稿日期: 2022-05-02.

编委: 陈旭瑾.

study proved theoretically the soundness of the average approximation algorithm and pointed out the quartic polynomial time complexity concerning the scale of the problem. By comparing with the results of commercial optimization software-Lingo, the comparison verified that the new method obtains the global optimal solution and needs very short computational time.

Keywords Integer programming, NP-hard problem, approximation algorithm, combinational optimization, assignment.

1 引言

指派问题是一个经典的运筹学问题,在现实中也有大量的应用.将 n 项任务指派给 n 个代理来完成,存在大量的可行匹配结果,要在其中找到符合要求的最佳结果事实上是一件很具挑战性的工作.与经典的指派问题主要关注指派后代理的总工作负荷最小不同,本文将以最小化指派后各代理之间工作负荷的差异为目标和关注点.工作负荷在代理集合上的均匀分布不仅有助于任务的顺利下发实施,也会减少由于工作负荷差异带来的消极情绪,增加代理的工作积极性.从长远的角度看,公平的工作负荷分派也会增加企业、组织和团队的凝聚力,有利于组织文化的培养.与线性指派问题和二次指派问题(或言二次分配问题)得到广泛关注和深入研究形成鲜明对比,对三次指派问题的研究非常少,而作为一类特殊形式的三次指派问题—公平性指派问题很少被研究者涉及.这与该问题本身具有的实践价值和理论意义极不相符.为此,本文将尝试构建公平性指派问题的优化模型,并对问题的特征进行分析,给出一种多项式时间复杂度的求解算法.

三次指派 Q

下面简介指派问题的相关研究.首先介绍与问题解的性质分析相关的研究.文献[1]提出指派问题的纳什均衡解,并证明有限指派问题有且仅有纯纳什均衡解.在经典指派问题的最优解不唯一条件下,文献[2]提出了一个考虑个体理性的指派问题多重最优解的择优方法,从而保证了指派问题最优解的唯一性.

针对各类指派问题的求解,研究者提出了许多有效方法,主要包括:分支定界法、匈牙利算法、反点算法和粒子群算法等.其中,针对一个代理可以被指派多个任务的非平衡指派问题,以最小化指派后工作负荷的平方和为优化目标,文献[3]建立了相应的指派问题模型,并给出了利用线性化技术处理后模型的分支定界算法.针对一个代理可以被指派多个任务的非平衡指派问题,文献[4]给出了一种改进的匈牙利算法.文献[5]对匈牙利算法寻找独立的次序进行了改进,从而避免了匈牙利算法通常需要进行多次试分配的不足.考虑指派中可能存在的冲突约束,文献[6]给出了求解该问题的改进分支定界算法.文献[7]提出通过搜索多个经典线性指派问题可行解满足实际指派的多样化需要,并给出了相应的确定多个解的方法.文献[8]考虑了一类非确定型指派问题,针对人员有无工作数限制分情况加以讨论,并借鉴Floyd算法的负回路思想,提出了一种迭代算法.文献[9]研究了一事多人的情形下,如何给每个人指派工作,才能使总工期和总耗时最短,给出了一种向量标记算法.文献[10]针对从 n 个人中派出 k ($0 < k \leq \min\{m, n\}$)个人去完成 m 项任务中的 k 项任务,且使总效率最高类型的指派问题给出了反点算法.文献[11]讨论了从 n 个公司选取 k 个去投资 m 个项目中的 k 项的总收益最大的非均衡投资问题,提出了对应的反点算法.文献[12]提出了一种求解指派问题的新的离散粒子群算法.针对广义指派问题,即一人多事或一事多人的情

形, 文献 [13] 设计了一种改进的离散粒子群优化算法. 文献 [14] 提出了一种求解经典指派问题的改进粒子群算法.

除了经典的任务指派应用场景, 研究者也将相关理论应用于其他场景. 文献 [15] 建立了有资格限制的指派问题的数学模型, 并通过将效益矩阵转化为求解矩阵, 从而将有资格限制的指派问题化为传统的指派问题来求解. 针对指标间相关联的模糊多目标指派问题, 基于广义模糊测度和 Choquet 积分的模糊权重信息集结算子, 文献 [16] 给出了一种将模糊多目标指派问题转化为传统指派问题的算法. 针对移动通信中的拥堵, 文献 [17] 建立了以成本最小为目标的资源指派问题模型. 考虑乘客的步行距离, 以最小化分配到停机坪的飞机数为目标, 文献 [18] 建立了飞机的登机口分配模型, 并提出了相应的分支定界和定向搜索算法.

与单纯考虑支配后工作负荷总量最小不同, 部分研究也从代理偏好、多目标和结果满意度等角度对指派问题进行了深入分析. 针对有给定偏好产品集合的顾客, 在部分满足顾客需求条件下, 通过定义产品序列价值函数, 以最大化整体的价值为目标, 文献 [19] 建立了相应的指派模型, 并给出了一种近似求解方法. 针对目标体系衡量存在差异的多目标指派问题, 文献 [20] 给出了一种将综合评价法与匈牙利法相结合的求解方法. 文献 [21] 通过将指派结果按满意度排序, 并对排序后结果进行部分加权求和, 从而定义了指派结果的公平性度量指标. 以上述指标为目标文献 [21] 建立了对应指派问题模型, 并证实该问题具有 NP-hard 特征. 通过分析指派过程和结果相互影响的不同机理, 经典指派问题也被推广至二次分配和三次指派问题. 文献 [22] 首次提出了一般化的三次指派问题, 并给出了相关数学模型. 文献 [23] 讨论了介于经典的指派问题与三维指派问题之间的一类可分解为二阶段决策的特殊三维匹配问题, 并给出了一种多项式时间算法. 文献 [24] 总结了求解二次分配问题的方法, 并提及了三次指派问题发展历史与现状. 指出三次指派问题是一类极难的 NP-hard 问题, 但由于缺乏应用场景, 基本很少被关注.

与现有研究相比, 本研究的主要学术贡献包括: a) 明确提出了公平性指派问题, 并建立了相应的优化模型; b) 给出了一种多项式时间复杂度的全局精确解求解算法, 并从理论和数值分析两个角度证实了算法的有效性.

2 公平性指派问题

本节将给出公平性指派问题 (Fair Assignment Problem-FAP) 的数学模型、分类和基本性质.

2.1 公平指派的优化模型

指派问题是将 n 项任务分配给 n 个代理, 每个代理只能执行一项任务, 且所有任务必须被完成, 所有的任务不可分割, 分配的目标是平衡分配后代理之间的工作负荷. 令 I 和 J 分别表示代理和任务的集合. 令 $x_{i,j}$ 指示任务 $j \in J$ 是否被代理 $i \in I$ 执行, 如果被执行, 其值为 1; 否则, 其值为 0. 指派问题的基本约束如下

$$\sum_i x_{i,j} = 1, \quad \forall j \in J; \quad (1)$$

$$\sum_j x_{i,j} = 1, \quad \forall i \in I; \quad (2)$$

$$x_{i,j} \in \{0, 1\}, \quad \forall i \in I, j \in J. \quad (3)$$

式 (1) 的含义为任一任务只能被一个代理执行; 式 (2) 的含义为任一代理只能且必须执行一项任务; 式 (3) 是变量的取值范围约束.

令 $c_{i,j}$ 表示任务 j 被代理 i 执行时产生的工作负荷, 并假设所有工作负荷具有上下限 \underline{c} 和 \bar{c} , 及条件 $c_{i,j} \in [\underline{c}, \bar{c}]$, $i \in I, j \in J$ 成立. 令 x 表示由 $x_{i,j}, i \in I, j \in J$ 构成的决策向量. 在满足约束 (1)–(3) 条件下, 经典指派问题的目标是使最小化指派后总的工作负荷, 具体目标函数如下

$$f_1(x) = \sum_{i \in I} \sum_{j \in J} c_{i,j} x_{i,j}. \quad (4)$$

经典指派问题属于线性纯整数规划, 可以利用匈牙利算法在计算时间复杂度为 $O(n^3)$ 的条件下有效求解.

在给定 x 条件下, 任务分配之后代理之间的平均工作负荷 c_{av} 可按下式计算

$$c_{av} = \frac{\sum_{i \in I} \sum_{j \in J} c_{i,j} x_{i,j}}{n}. \quad (5)$$

平衡的工作负荷必然使得各个代理的工作负荷与平均工作负荷的绝对差值最小化. 因此可定义如下的工作负荷公平性指标 $f_2(x)$

$$f_2(x) = \sum_{i \in I} \sum_{j \in J} (c_{i,j} - c_{av})^2 x_{i,j}. \quad (6)$$

由最小化工作负荷指标 $f_2(x)$ 和约束式 (1)–(3) 构成公平性指派问题的优化模型 (Fair Assignment Model-FAM).

下面分析 FAM 的基本性质. 注意到对于满足式 (1)–(3) 的 0-1 变量 $x_{i,j}, i \in I, j \in J$ 而言, 下面的等式和约束条件隐含成立

$$x_{i,j}^2 = x_{i,j}; \quad (7)$$

$$x_{i,j} x_{i,k} = 0, \quad \forall j \neq k, i \in I; \quad (8)$$

$$x_{i,j} x_{k,j} = 0, \quad \forall i \neq k, j \in J; \quad (9)$$

$$x_{i,j} x_{k,l} \in \{0, 1\}, \quad \forall i \neq k, j \neq l; \quad (10)$$

$$x_{i,j} x_{k,l} x_{m,n} \in \{0, 1\}, \quad \forall i \neq k \neq m, j \neq l \neq n. \quad (11)$$

如果将式 (5) 中 c_{av} 的表达式代入式 (6), 并作代数展开, 易知 $f_2(x)$ 的展开式中将包含 $x_{i,j} x_{k,l} x_{m,n}, \forall i \neq k \neq m, j \neq l \neq n$ 形式的三次项, 因此可以说一般情况下, 公平性指派问题是一类特殊的三次指派问题 (Cubic Assignment Problem-CAP).

与一般 CAP 一样, 公平性指派问题的目标函数包含了所有的三次项和两次项, 但是公平性指派问题的目标函数可以通过代数运算, 化简为式 (6) 的形式. 一般 CAP 由于其目标函数各项系数的随意性, 不具有上述简化性质. 另外, 公平性指派问题和一般的三次指派问题的约束集合相同.

2.2 两类基本公平性指派问题

与线性指派问题和二次指派问题 (Quadratic Assignment Problem-QAP) 相比, 目前学术界鲜有对三次指派问题 (CAP) 的研究. 下面对两类基本的公平指派问题进行介绍. 为了便于理解和图示, 本小节将暂时令 $n = 3$.

第一类 FAP 是工作负荷均值固定的 FAP, 即 c_{av} 为定值. 假设任一给定代理完成任一任务由自身性质引发的工作负荷为一定值, 而任一给定任务无论由哪个代理执行由其自身性质引发的工作负荷也为一定值. 如图 1 所示, 代理 1, 2 和 3 对应的自身引发的工作负荷分别为 a_1 , a_2 和 a_3 ; 而不同任务由自身引发的工作负荷分别为 b_1 , b_2 和 b_3 . 一个具体指派的工作负荷 $c_{i,j}$ 满足 $c_{i,j} = a_i + b_j$. 对于 $n = 3$ 的情况, 工作负荷矩阵 $C_{3 \times 3}$ 可视为由如下的两个矩阵 $A_{3 \times 3}$ 和 $B_{3 \times 3}$ 的加和形成.

$$A_{3 \times 3} = \begin{bmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \end{bmatrix}, \quad (12)$$

$$B_{3 \times 3} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \end{bmatrix}, \quad (13)$$

$$C_{3 \times 3} = A_{3 \times 3} + B_{3 \times 3} = \begin{bmatrix} a_1 + b_1 & a_1 + b_2 & a_1 + b_3 \\ a_2 + b_1 & a_2 + b_2 & a_2 + b_3 \\ a_3 + b_1 & a_3 + b_2 & a_3 + b_3 \end{bmatrix}. \quad (14)$$

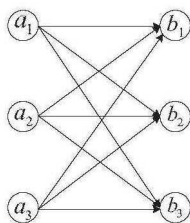


图 1 工作负荷均值固定的公平性指派问题
(Figure 1 FAP with fixed average of working load)

在上述假设条件下, 易知

$$c_{av} = \frac{\sum_{i=1}^n (a_i + b_i)}{n}. \quad (15)$$

如果将 (15) 代入式 (6), 易知对应的 FAP 退化为具体指派 (i, j) , $i \in I$, $j \in J$ 的工作负荷为定值 $(c_{i,j} - c_{av})^2$ 的经典线性指派问题, 于是可以利用匈牙利算法对其进行有效求解.

第二类公平性指派问题指的是工作负荷均值不固定的一般性公平指派问题. 当 $n = 3$ 时, 其对应的工作负荷矩阵 $C_{3 \times 3}$ 如式 (16) 所示

$$C_{3 \times 3} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}. \quad (16)$$

图 2 给出了 $n = 3$ 时工作负荷均值不固定的公平性指派问题二部图. 图 2 中, 带数字的圆圈代表序号为该数字的代理, 而带数字的小方块表示序号为该数字的任务. 在工作负荷均值不固定条件下, 对应的 FAM 为三次指派问题, 属于一类极难的 NP-hard 问题^[24]. 下文的讨论将以该类问题为主.

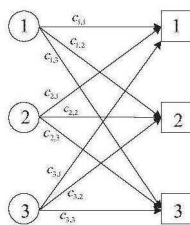


图2 工作负荷均值不固定的公平性指派问题
(Figure 2 FAP with a non-fixed average of working load)

3 均值逼近算法

本节将首先阐述算法的基本思想, 然后给出算法的具体执行步骤, 最后对算法的基本性质加以理论分析.

3.1 算法的基本思想

求解公平性指派问题的算法思想源于一个观察, 即当目标函数式 (6) 中的工作负荷均值 c_{av} 给定时, 问题转化为线性指派问题. 而对于线性指派问题, 可以利用匈牙利算法加以有效求解. 通过试算和理论分析, 在给定一个初始工作负荷均值后, 通过求解对应的线性指派问题可得到一个新的工作负荷均值; 利用新的工作负荷均值形成新的线性指派问题, 并对其加以求解; 重复上述求解过程可形成一个具有收敛特性的工作负荷均值序列. 而收敛的工作负荷均值对应的指派结果为问题的一个局部最优解. 考虑到工作负荷均值具有给定的取值范围, 通过在该取值范围上的一维搜索可以得到最终的全局最佳指派结果. 算法的核心思想是将复杂的三次指派问题转化为经典线性指派问题, 并通过一维搜索求解一系列经典线性指派问题不断逼近原问题的全局最优解.

3.2 算法步骤

基于前述基本思想构建的均值逼近算法的求解具体步骤如下

步骤 1 确定工作负荷均值 c_{av} 的取值范围 $[\underline{c}, \bar{c}]$, 其中取值的下限 $\underline{c} = \min\{c_{i,j}\}_{i \in I, j \in J}$, 上限 $\bar{c} = \max\{c_{i,j}\}_{i \in I, j \in J}$. 令子问题求解次数 $\tau = 0$. 则工作负荷均值的初始取值 $c_{av}(\tau) = \underline{c}$, 前一收敛均值 $c_{fore} = \underline{c} - 1$.

步骤 2 确定算法的迭代步长 δ . 将矩阵 C 的所有元素置入一个集合, 并按照大小升序排列, 得到满足条件 $c_m < c_{m+1}$, $m = 1, 2, \dots, M-1$ 的有序集合 $C_M = \{c_1, c_2, \dots, c_M\}$. 算法的迭代步长 δ 可按下式计算

$$\delta = \max \left\{ \frac{\min_{m \leq M-1} \{(c_{m+1} - c_m)\}}{n}, \Delta \right\}, \quad (17)$$

式中, Δ 为一个根据问题事先设定的正常数, 用于避免由于 C_M 中可能出现两个非常接近的元素从而使得到的步长过小.

步骤 3 调用匈牙利算法求解子问题. 首先, 令求解子问题的初始工作负荷均值 $c_{initial} = c_{av}(\tau)$. 然后以式 (18) 为目标函数, 式 (1)–(3) 为约束, 得到一个线性指派问题.

$$\min f(x) = \sum_{i \in I} \sum_{j \in J} (c_{i,j} - c_{av}(\tau))^2 x_{i,j}. \quad (18)$$

调用匈牙利算法求解上述线性指派问题, 得到相应的最佳 $x^*(\tau)$ 和目标函数值 f_τ^* . 利用 $x^*(\tau)$, 根据式 (5) 计算得到对应的工作负荷均值 $c_{av}^*(\tau)$. 如果 $|c_{av}^*(\tau) - c_{av}(\tau)| \leq \gamma$, 转下一步; 否则, 令 $c_{av}(\tau) := c_{av}^*(\tau)$, 继续求解更新后的线性指派问题. 这里 γ 是给定正常数, 用于判断工作负荷均值是否已经收敛.

步骤 4 确定新的工作负荷均值. 如果 $c_{av}^*(\tau) \neq c_{fore}$, 令 $c_{fore} := c_{av}^*(\tau)$ 和 $c_{av}(\tau) := c_{av}^*(\tau) + \delta$; 否则, 令 $c_{av}(\tau) := c_{initial} + \delta$.

步骤 5 终止判断. 如果 $c_{av}(\tau) \leq \bar{c}$, 转步骤 3; 否则, 终止算法, 输出对应 $f^* = \min\{f_\tau^*\}$ 的最优解 x^* .

3.3 算法的性质定理

后续的证明需要利用下面的两个引理.

引理 1 给定集合 $C_K := c_1, c_2, \dots, c_K$. 令函数 $z(c_{av}) = \sum_{1 \leq k \leq K} (c_k - c_{av})^2$. 则问题 $\min z(c_{av})$ 的最优解为 $c_{av}^* = \sum_{1 \leq k \leq K} (\frac{c_k}{K})$.

引理 2 假设数列 $\{s_k\}$, $k = 1, 2, \dots$ 单调递增 (或递减). 数列 $\{s_k\}$ 收敛的充分必要条件为该数列有界.

上述两个引理为数值优化和分析的基本结论, 因此本文省略对其证明. 下面分析与本文算法相关的 3 个基本定理.

定理 1 任意给定一个均值, 均值逼近算法会单向收敛于问题的一个局部最优解.

证 假设给定工作负荷均值 c_1 , 令函数 $z(x, c_{av}) = \sum_{i \in I} \sum_{j \in J} (c_{i,j} - c_{av}(\tau))^2 x_{i,j}$, 而 X 为由约束式 (1-3) 构成的 $x_{i,j}$, $i \in I, j \in J$ 的可行域. 则算法需要求解的线性指派问题为 $\min_{x \in X} z(x, c_1)$. 令 c_2 为对应问题 $\min_{x \in X} z(x, c_1)$ 的最优解的工作负荷均值. 假设 $c_1 < c_2$, 如图 3 所示. 对于任一指派 (i, j) , 其对应的工作负荷 $c_{i,j}$ 在均值为 c_{av} 时的目标函数项为 $z_{i,j}(c_{av}) = (c_{i,j} - c_{av})^2$. 图 3 给出了均值分为 c_1 和 c_2 时, 目标函数项 $z_{i,j}(c_{av})$ 的变化曲线 $z(c_1)$ 和 $z(c_2)$. 从图 3 可知, 当 $c_{i,j} < \frac{(c_1+c_2)}{2}$ 时, 对应的目标函数项 $z_{i,j}(c_{av})$ 的值在工作负荷均值从 c_1 增大到 c_2 时, 也会随之增大; 而当 $c_{i,j} > \frac{(c_1+c_2)}{2}$ 时, $z_{i,j}(c_{av})$ 的值会降低. 基于上述观察可知, 在工作负荷均值从 c_1 增大到 c_2 后, 对应问题 $\min_{x \in X} z(x, c_2)$ 的最优解的实际选取的工作负荷会趋向于选取比当负荷均值为 c_1 时的值更大的值, 因此会有 $c_2 \leq c_3$. 这里 c_3 为对应问题 $\min_{x \in X} z(x, c_2)$ 的最优解的工作负荷均值. 上述观察如果成立, 会形成一个单调递增的序列 $\{c_1, c_2, \dots\}$. 易知, 上述序列的元素的取值范围为 $[\underline{c}, \bar{c}]$. 根据引理 2, 可知上述序列必收敛. 下面对上述观察加以严格证明, 并说明收敛的点对应问题的一个局部最优解.

令 $C_1 = \{c_{i,j}\}_1$ 表示对应问题 $\min_{x \in X} z(x, c_1)$ 的最优解的实际选取的工作负荷构成的集合. 则 c_2 为集合 C_1 中所有元素的均值. 根据引理 1, $\min_{x \in X} z(x, c_1)$ 的最优目标值大于 $\sum_{c_{i,j} \in C_1} (c_{i,j} - c_2)^2$. 因为 $C_1 = \{c_{i,j}\}_1$ 也是问题 $\min_{x \in X} z(x, c_2)$ 的一个可行解对应的工作负荷集合, 因此问题 $\min_{x \in X} z(x, c_2)$ 的最优目标值一定小于等于 $\sum_{c_{i,j} \in C_1} (c_{i,j} - c_2)^2$. 上述的逻辑推理可以继续, 从而对于上述的单调递增的序列 $\{c_1, c_2, \dots\}$ 而言, 其对应的目标函数值会呈现出单调下降的特征. 综上, 当序列 $\{c_1, c_2, \dots\}$ 收敛时, 对应的线性指派问题的最优解对应原问题的一个局部最优解.

上面的分析针对的是工作负荷均值增大的情景. 对于工作负荷均值减少的情景, 具体分析也是类似的, 在此不再赘述.

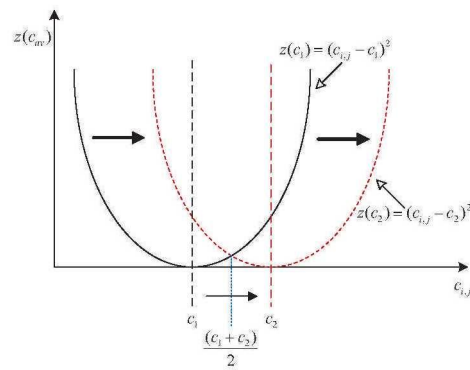


图 3 均值变化对子问题求解的影响

(Figure 3 Impact of average on solving sub-problem)

定理 2 在均值的可行取值范围内, 均值逼近算法求得的局部最优解中包含了全局最优解.

证 由定理 1 可知任意给定一个可行的工作负荷均值, 算法均会收敛于与之相关的一个局部最优解. 如果令算法的迭代步长 $\delta = \frac{\min_{m < (M-1)} \{(c_{m+1} - c_m)\}}{n}$, 则从一个可行指派变为另一个可行指派, 对应的工作负荷均值的变化一定大于等于上述的迭代步长 δ . 算法从均值的可行取值范围的下界 \underline{c} 出发, 当均值的取值大于等于上界 \bar{c} 时结束, 除了求解子问题时出现的均值收敛性增大, 均值的增大幅度为 δ . 从上述分析可知, 算法事实上遍历了均值的可行域, 因此定理的结论成立. 当然算法中取 $\delta = \max\{\frac{\min_{m < (M-1)} \{(c_{m+1} - c_m)\}}{n}, \Delta\}$ 是为了避免很小的均值增大幅度, 从而加速求解速度. 上述的步长权益取值方法并不影响上述的推理结论.

定理 3 将工作负荷矩阵 C 的所有元素置入一个集合, 并按照大小升序排列, 得到满足条件 $c_m < c_{m+1}$, $m = 1, 2, \dots, (M-1)$ 的有序集合 $C_M := c_1, c_2, \dots, c_M$. 令 $\mu = \frac{\min_{m < (M-1)} \{(c_{m+1} - c_m)\}}{n}$ 和 $n_\mu = \lceil \frac{(\bar{c} - \underline{c})}{\mu} \rceil$. 其中 $\lceil r \rceil$ 表示取大于实数 r 且最靠近 r 的整数. 均值逼近算法的计算时间复杂度为 $O(n^3 \times n_\mu)$.

证 算法求解线性指派问题利用的匈牙利算法的时间复杂度为 $O(n^3)$. 由迭代步长的定义可知, 给定一个均值搜索其对应局部最优解的求解次数不大于 n_μ . 而每次搜索局部最优解的计算中可以通过与相邻均值的计算结果比较, 只进行一次求解线性指派问题的计算. 因此, 定理 3 的结论成立.

4 算例分析

首先分析一个具有 7 个代理的简单算例. 具体的工作负荷矩阵 $C_{7 \times 7}$ 如下所示.

$$C_{7 \times 7} = \begin{bmatrix} 9 & 11 & 3 & 6 & 6 & 7 & 4 \\ 10 & 9 & 11 & 5 & 6 & 3 & 6 \\ 8 & 10 & 5 & 6 & 4 & 7 & 1 \\ 6 & 8 & 10 & 4 & 9 & 5 & 8 \\ 11 & 1 & 9 & 8 & 7 & 11 & 8 \\ 4 & 1 & 3 & 2 & 7 & 0 & 3 \\ 4 & 6 & 2 & 9 & 8 & 5 & 4 \end{bmatrix}.$$

从 $C_{7 \times 7}$ 易知 $[\underline{c}, \bar{c}]$ 为 $[0, 11]$, $\delta = \frac{\min_{m < (M-1)} \{(c_{m+1} - c_m)\}}{n} = \frac{1}{7}$. 利用 Java 程序语言实现均值逼

近算法, 在 NetBeans IDE 12.3 开发环境下运行, 采用的计算机处理器为 Intel®Core i7-1065G7 CPU. 由于问题规模较小, 计算的耗时小于 0.001 秒. 计算得到的指派结果为 $[6, 2, 1, 7, 3, 5, 4]$. 这里指派结果中的有序数字表示对应位置代理所需执行的任务序号, 如第一个数字 6 表示代理 1 需执行的任务序号为 6, 结果中的第 2 个数字 2 表示代理 2 需执行的任务序号为 2, 依次类推, 直到第 7 个数字 4 表示代理 7 需执行任务 4. 计算所得的最佳目标函数值为 4.8571, 对应的工作负荷均值为 8.1429. 将算法执行过程中需要增加单位步长的均值点和对应的目标函数值, 共计 40 组数据, 绘制在图 4 中. 从图 4 可以看出, 如果将均值视为变量, 问题是一个具有多个局部最优解的问题; 而且对每个局部最优解而言, 都有一个关联的收敛邻域, 在该邻域内目标函数均会收敛于该局部最优解. 所有局部最优解的上述关联邻域会形成一个对均值的可行取值范围的一个完整划分. 上述的分析与上文的理论分析一致, 进一步印证了理论分析的正确性. 我们也利用商业优化软件 Lingo 对上述问题进行了求解, 得到了相同的计算结果, 计算耗时 2 秒.

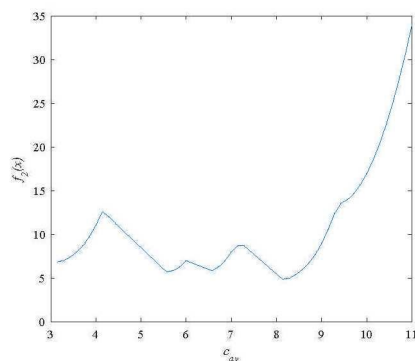


图 4 随工作负荷均值增加, 目标函数值的变化

(Figure 4 Variation of objective function values with the increasing average of working load)

为了进一步证实算法的有效性, 下面对 6 个不同规模的问题进行计算和比较. 表 1 给出了 6 个问题的参数, 表 1 中 n 为任务或代理的数目, $[c, \bar{c}]$ 表示工作负荷的取值范围, δ 为按照 $\frac{\min_{m \leq (M-1)} \{(c_{m+1} - c_m)\}}{n}$ 计算得到的步长. 具体的工作负荷是在给定区间 $[c, \bar{c}]$ 上服从均匀分布的随机整数, 而设定参数 $\Delta = 1$.

表 1 不同的指派问题

(Table 1 Different assignment problems)

问题	n	$[c, \bar{c}]$	δ
1	10	$[20, 40]$	0.1
2	15	$[20, 60]$	0.07
3	20	$[30, 90]$	0.05
4	25	$[30, 90]$	0.04
5	30	$[10, 100]$	0.034
6	35	$[20, 70]$	0.029

对应上面给出的 6 个问题, 表 2 给出了分别利用均值逼近算法 (Average Approximation Algorithm-3A) 和商业软件 Lingo 自带求解器得到的结果. 由于 Lingo 在超长计算时间 (超过数小时) 条件下仍然无法针对问题 4、5 和 6 给出一个可行解, 因此表 2 中没有相关的计算信息. 对前三个问题, 两种方式均可求得相同的最优解. 但是 Lingo 的计算时间从 15 秒迅速增加到 35 分 37 秒, 而 3A 法的计算时间均小于千分之一秒 (Netbean 软件显示的计算时间为 0 秒, 即小于机器可以识别的 0.001 秒). 对于 6 个问题, 3A 法需要求解的线性指派子问题的次数增加有限, 从 50 到 198; 而 Lingo 自带求解器显示的迭代次数从 16 万次增加到了 1450 多万次. 上述比较证实了均值逼近算法的有效性和实用性. 为便于读者验证分析, 在表 3 后给出了问题 2 对应的具体工作负荷矩阵 $C_{15 \times 15}$.

表 2 不同方法的计算结果比较
(Table 2 Comparison of results from different methods)

方 法	问 题	指派结果	负荷 均值	目标 函数值	计算 时间 (s)	子问题求解次数 (求解器迭代次数)
3A	1	[6, 3, 7, 5, 8, 9, 2, 1, 4, 10]	22.7	8.1	< 0.001	50
	2	[1, 5, 4, 11, 15, 6, 8, 7, 12, 10, 3, 14, 2, 13, 9]	39.8	42.4	< 0.001	100
	3	[12, 18, 10, 1, 8, 2, 15, 14, 19, 17, 6, 20, 4, 3, 7, 11, 5, 16, 9, 13]	39.3	46.2	< 0.001	123
	4	[11, 18, 25, 14, 9, 7, 3, 12, 8, 15, 6, 2, 13, 10, 20, 22, 5, 17, 24, 4, 19, 21, 16, 1, 23]	65.76	58.56	< 0.001	172
	5	[30, 21, 8, 17, 3, 26, 4, 1, 2, 16, 13, 19, 24, 9, 15, 29, 7, 12, 25, 28, 18, 27, 22, 20, 14, 5, 23, 11, 6, 10]	84.97	128.97	< 0.001	198
	6	[35, 12, 25, 17, 3, 32, 29, 28, 16, 27, 1, 14, 31, 6, 9, 5, 19, 8, 26, 30, 20, 4, 2, 11, 21, 18, 24, 13, 33, 22, 15, 23, 34, 7, 10]	56.6	38.40	< 0.001	131
Lingo	1	[6, 3, 7, 5, 8, 9, 2, 1, 4, 10]	22.7	8.1	15	168383
	2	[1, 5, 4, 11, 15, 6, 8, 7, 12, 10, 3, 14, 2, 13, 9]	39.8	42.4	411	3883007
	3	[12, 18, 10, 1, 8, 2, 15, 14, 19, 17, 6, 20, 4, 3, 7, 11, 5, 16, 9, 13]	39.3	46.2	2137	14518548

$$C_{15 \times 15} = \begin{bmatrix} 49 & 56 & 53 & 82 & 35 & 53 & 76 & 86 & 46 & 32 & 33 & 41 & 53 & 43 & 49 & 73 & 83 & 43 & 83 & 36 \\ 78 & 56 & 46 & 38 & 39 & 73 & 69 & 80 & 66 & 54 & 82 & 39 & 34 & 49 & 71 & 38 & 65 & 39 & 37 & 44 \\ 60 & 63 & 62 & 51 & 71 & 86 & 73 & 61 & 37 & 37 & 35 & 67 & 85 & 75 & 76 & 71 & 86 & 64 & 32 & 72 \\ 40 & 67 & 72 & 74 & 39 & 49 & 49 & 69 & 85 & 73 & 81 & 56 & 64 & 38 & 32 & 52 & 88 & 76 & 52 & 36 \\ 57 & 69 & 63 & 82 & 68 & 31 & 55 & 40 & 82 & 40 & 38 & 60 & 86 & 83 & 88 & 63 & 84 & 59 & 47 & 80 \\ 83 & 38 & 66 & 77 & 85 & 31 & 54 & 67 & 83 & 35 & 31 & 61 & 46 & 58 & 68 & 41 & 76 & 46 & 47 & 76 \\ 72 & 32 & 39 & 46 & 88 & 51 & 34 & 88 & 70 & 79 & 79 & 84 & 81 & 61 & 38 & 83 & 30 & 81 & 52 & 37 \\ 59 & 64 & 40 & 81 & 82 & 84 & 59 & 52 & 84 & 39 & 64 & 74 & 36 & 40 & 39 & 34 & 87 & 47 & 66 & 83 \\ 77 & 46 & 58 & 68 & 45 & 78 & 50 & 70 & 44 & 50 & 71 & 80 & 74 & 57 & 39 & 72 & 69 & 31 & 38 & 55 \\ 86 & 81 & 75 & 60 & 78 & 65 & 50 & 55 & 81 & 64 & 74 & 50 & 71 & 78 & 51 & 49 & 43 & 72 & 67 & 68 \\ 45 & 83 & 51 & 42 & 50 & 37 & 63 & 32 & 44 & 83 & 87 & 72 & 88 & 40 & 31 & 88 & 84 & 51 & 60 & 83 \\ 77 & 46 & 67 & 56 & 51 & 88 & 75 & 85 & 60 & 83 & 57 & 46 & 49 & 68 & 53 & 85 & 65 & 80 & 78 & 40 \\ 33 & 40 & 48 & 40 & 44 & 78 & 33 & 42 & 62 & 63 & 76 & 89 & 36 & 67 & 41 & 65 & 83 & 62 & 82 & 65 \\ 72 & 63 & 39 & 84 & 33 & 76 & 61 & 82 & 73 & 79 & 34 & 38 & 38 & 86 & 56 & 33 & 62 & 33 & 83 & 45 \\ 65 & 76 & 64 & 51 & 77 & 36 & 39 & 80 & 58 & 53 & 79 & 48 & 71 & 30 & 33 & 65 & 57 & 83 & 55 & 60 \\ 52 & 60 & 56 & 44 & 37 & 74 & 73 & 67 & 71 & 63 & 38 & 44 & 83 & 67 & 79 & 35 & 32 & 88 & 48 & 67 \\ 59 & 78 & 31 & 83 & 42 & 75 & 71 & 53 & 88 & 84 & 86 & 46 & 49 & 31 & 46 & 73 & 59 & 84 & 89 & 46 \\ 55 & 56 & 59 & 77 & 57 & 75 & 61 & 56 & 43 & 75 & 88 & 66 & 88 & 79 & 70 & 40 & 85 & 70 & 58 & 80 \\ 38 & 86 & 79 & 55 & 58 & 34 & 77 & 30 & 39 & 36 & 44 & 82 & 51 & 70 & 48 & 63 & 41 & 82 & 51 & 71 \\ 81 & 48 & 75 & 87 & 38 & 86 & 63 & 56 & 62 & 57 & 60 & 73 & 38 & 67 & 58 & 31 & 38 & 86 & 31 & 80 \end{bmatrix}.$$

为了进一步验证算法对较大规模公平性指派问题的处理能力, 表 3 给出了 6 个具有较多代理与任务的指派问题计算结果. 相关的工作负荷取值范围由 $[\underline{c}, \bar{c}]$ 给出, 具体的工作负荷在上述范围内按照均匀分布随机生成. 参数 Δ 的值仍取 1. 从表 3 给出的计算时间可知, 算法可以快速的处理较大规模的公平性指派问题.

表 3 较大规模的指派问题
(Table 3 Relatively large-scale assignment problems)

问题	n	$[\underline{c}, \bar{c}]$	δ	负荷均值	目标函数值	计算时间 (s)	子问题求解次数
1	100	[10, 50]	0.1	21.03	6.91	< 0.001	40
2	150	[20, 60]	0.07	47.06	8.46	< 0.001	49
3	200	[30, 90]	0.05	74.03	5.82	1	59
4	250	[30, 100]	0.04	44.03	7.74	2	69
5	300	[50, 200]	0.034	125.2	63.64	7	150
6	350	[100, 300]	0.029	175.34	98.86	12	198

5 结 论

通过研究, 本文得到的主要结论包括: a) 除特殊情况, 公平性指派问题是一类特殊形式的三次指派问题; b) 均值逼近算法可以有效遍历可行域, 得到问题的全局最优解; c) 均值逼近算法具有问题规模的四次多项式的计算时间复杂度; d) 通过与商业优化软件 Lingo 计算结果的比较, 说明新方法的计算效率高, 适用于实际规模问题的求解.

本研究仅是对公平性指派问题的初步探索, 未来还需要对优化模型的各种变化形式, 如在目标中考虑最小化总的工作负荷, 进行深入研究, 也可以针对模型尝试各种其他的求解算法, 如遗传算法等. 另外, 也需要对公平性指派问题在各个领域的应用进行具有针对性分析.

参 考 文 献

- [1] 徐屹嵩, 王应明. 指派问题的纳什均衡解. 运筹与管理, 2013, **22**(4): 101–105, 110.
(Xu Y S, Wang Y M. The Nash equilibrium in the assignment problem. *Operations Research and Management Science*, 2013, **22**(4): 101–105, 110.)
- [2] 徐屹嵩, 王应明. 指派问题的多重最优解的择优方法. 运筹学学报, 2014, **18**(2): 96–102.
(Xu Y S, Wang Y M. Aranking method for the assignment problem with multiple optimal solutions. *Operations Research Transactions*, 2014, **18**(2): 96–102.)
- [3] Karsu Ö, Azizoglu M. An exact algorithm for the minimum squared load assignment problem. *Computers and Operations Research*, 2019, **106**: 76–90.
- [4] Rabbani Q, Khan A, Quddoos A. Modified Hungarian method for unbalanced assignment problem with multiple jobs. *Applied Mathematics and Computation*, 2019, **361**: 493–498.
- [5] 周莉, 张维华, 徐射雕. 求解指派问题的一次性分配算法. 计算机工程与应用, 2011, **47**(18): 135–138, 152.
(Zhou L, Zhang W H, Xu S D. One-time assignment algorithm to solve assignment problem. *Computer Engineering and Applications*, 2011, **47**(18): 135–138, 152.)
- [6] Öncan T, Suvak Z, Akyüz M H, et al. Assignment problem with conflicts. *Computers and Operations Research*, 2019, **111**: 214–229.
- [7] Malaguti E, Durán R M. Computing k different solutions to the assignment problem. *Computers & Industrial Engineering*, 2019, **135**: 528–536.
- [8] 李岩, 郭强. 非确定性指派问题的求解算法. 计算机工程与应用, 2009, **45**(15): 61–63, 66.
(Li Y, Guo Q. Algorithm for uncertainty assignment problem. *Computer Engineering and Applications*, 2009, **45**(15): 61–63, 66.)
- [9] 付晓薇, 郭强, 马芹芹. 一类非确定型多目标指派问题及其算法研究. 运筹与管理, 2013, **22**(6): 34–38.
(Fu X W, Guo Q, Ma Q Q. A multiple objective uncertainty assignment problem and the study of its algorithm. *Operations Research and Management Science*, 2013, **22**(6): 34–38.)
- [10] 王立柱, 刘阳. 分配小于人数和任务数的指派问题的反点算法. 运筹学学报, 2011, **15**(3): 124–128.
(Wang L Z, Liu Y. Reverse point algorithm of assignment problem on assignment less than jobs and persons. *Operations Research Transactions*, 2011, **15**(3): 124–128.)
- [11] 王立柱, 刘阳, 石洋, 等. 非均衡投资收益极大指派问题. 沈阳师范大学学报 (自然科学版), 2014, **33**(3): 364–368.
(Wang L Z, Liu Y, Shi Y, et al. Maximum of unbalanced investment income assignment problem. *Journal of Shenyang Normal University (Natural Science Edition)*, 2014, **33**(3): 364–368.)
- [12] 孙晓雅, 林焰. 一种新的离散粒子群算法在指派问题中的应用. 计算机应用研究, 2009, **26**(11): 4091–4093, 4097.
(Sun X Y, Lin Y. Using new DPSO algorithm to solve assignment problem. *Application Research of Computers*, 2009, **26**(11): 4091–4093, 4097.)
- [13] 王一川, 单甘霖, 童俊. 改进离散粒子群优化算法求解广义指派问题. 科技通报, 2013, **29**(8): 130–132.
(Wang Y C, Shan G L, Tong J. Solving generalized assignment problem based on improved discrete particle swarm optimization. *Bulletin of Science and Technology*, 2013, **29**(8): 130–132.)
- [14] El-Ashmawa W H, Ali A F. A modified salp swarm algorithm for task assignment problem. *Applied Soft Computing Journal*, 2020, **94**: 106445.
- [15] 黄龙生, 徐光辉. 有资格限制的指派问题的求解方法. 运筹与管理, 2005, **14**(1): 28–31.
(Huang L S, Xu G H. Solution of assignment problem of restriction of qualification. *Operations Research and Management Science*, 2005, **14**(1): 28–31.)
- [16] 陈岩, 李庭, 鲍博. 基于 Choquet 积分的指标关联模糊多目标指派问题. 系统工程理论与实践, 2017, **37**(8): 2162–2170.
(Chen Y, Li T, Bao B. Choquet integral based interacting multiobjective assignment problem with fuzzy information. *Systems Engineering — Theory & Practice*, 2017, **37**(8): 2162–2170.)

- [17] Liu Y J, Yang Y J, Wang E, et al. A fair task assignment strategy for minimizing cost in mobile crowdsensing. Proc. of the 26th International Conference on Parallel and Distributed Systems, Hong Kong: IEEE Press, 2020, 44–53.
- [18] Karsu Ö, Azizoglu M, Alanli K. Exact and heuristic solution approaches for the airport gate assignment problem. *Omega*, 2021, **103**(6): Article ID: 102422.
- [19] Gao G C, Ning L, Ting H F, et al. Approximation algorithms for the partial assignment Problem. *Theoretical Computer Science*, 2020, **838**(34): 231–237.
- [20] 高原, 李仁传, 张合勇, 等. 考虑目标差异的多目标指派问题研究. 海军工程大学学报, 2020, **32**(5): 102–106, 112.
(Gao Y, Li R C, Zhang H Y, et al. Research on multi-objective assignment problem by considering target difference. *Journal of Naval University of Engineering*, 2020, **32**(5): 102–106, 112.)
- [21] Lesca J, Minoux M, Perny P. The fair OWA one-to-one assignment problem: NP-hardness and polynomial time special cases. *Algorithmica*, 2019, **81**(1): 98–123.
- [22] Lawler E L. The quadratic assignment problem. *Management Science*, 1963, **9**(4): 586–599.
- [23] 林浩, 林澜. 一类二阶段指派问题. 运筹与管理, 2021, **30**(2): 97–101.
(Lin H, Lin L. A type of two-stage assignment problem. *Operations Research and Management Science*, 2021, **30**(2): 97–101.)
- [24] Hahn P M, Zhu Y, Guignard M, et al. Exact solution of emerging quadratic assignment problems. *International Transactions in Operational Research*, 2010, **17**(5): 525–552.