Operations Research and Fuzziology 运筹与模糊学, 2022, 12(1), 16-25
Published Online February 2022 in Hans. http://www.hanspub.org/journal/orf
https://doi.org/10.12677/orf.2022.121002

Hans汉斯

公平性指派问题及其进化求解算法

崔允汀，何胜学*

上海理工大学管理学院，上海

收稿日期：2021年12月20日；录用日期：2022年1月21日；发布日期：2022年1月29日

摘　要

针对任务指派中指派结果可能存在工作负荷分布极不平衡的问题，建立了公平性指派问题优化模型，并设计了求解模型的进化算法。以指派后代理的工作负荷与负荷均值的差值大小度量工作负荷的不均衡状况，将上述差值的平方和作为工作负荷的公平性指标。基于经典线性指派问题，构建了公平指派问题的优化模型。将相关问题按是否具有固定均值分为两类，并对两类问题的特征和求解策略加以分析。为提高计算效率，针对第二类指派问题的模型特征，设计了一种可求解过程中保持解的可行性的改进遗传算法。通过多个算例计算，并与Lingo软件求解结果对比，验证了模型和算法的合理性和有效性。研究表明：1) 一般情况下，公平性指派问题是一类具有NP-hard特征的三次指派问题；2) 追求公平的指派结果可能导致整体的工作负荷量增加；3) 所设计的遗传算法可高效求解各种规模的公平指派问题，但往往只能给出局部最优解；4) 变异系数对求解结果有显著影响，过大或过小的变异系数值均不利于算法求得最佳解。

关键词

整数规划，组合优化，三次指派问题，遗传算法，NP-Hard

Fair Assignment Problem and Its Genetic Algorithm

Yunting Cui, Shengxue He*

Business School, University of Shanghai for Science and Technology, Shanghai

Received: Dec. 20th, 2021; accepted: Jan. 21st, 2022; published: Jan. 29th, 2022

Abstract

In order to solve the highly unbalanced working load distribution possibly shown after assignment, this paper formulated the optimization model of fair assignment problem and designed a

*通讯作者。

文章引用: 崔允汀, 何胜学. 公平性指派问题及其进化求解算法[J]. 运筹与模糊学, 2022, 12(1): 16-25.
DOI: 10.12677/orf.2022.121002

genetic algorithm to solve this model. In view that the difference between the working load of an agents and the average of working load reflects the unbalance state of working load, the sum of the square of the above differences was defined as the index of fairness of working load. Based on the classic linear assignment problem, this paper constructed the optimization model of fair assignment problem. The paper grouped the related problems into two categories according to whether the average is a constant and analyzed the properties and solution strategies of the two categories. To improve the efficiency of solving the problem in the second category, this paper designed a modified genetic algorithm that remains the feasibility of solutions obtained during the iterations. Through computing several examples and comparing the results with that of Lingo software, this paper verified the rationality and effectiveness of the new model and algorithm. This study shows the following: 1) In general, fair assignment problem is a type of cubic assignment problem with NP-hard property. 2) Pursuing a fair assignment result may lead to an increased total working load. 3) The designed genetic algorithm can solve various scales of fair assignment problem, but usually only give a local optimal solution. 4) The mutation coefficient has a noticeable impact on the final solution and a too big or too small mutation coefficient has negative influence on searching a better solution.

Keywords

Integer Programming, Combinational Optimization, Cubic Assignment Problem, Genetic Algorithm, NP-Hard

1. 引言

在给定工作负荷条件下，经典的指派问题指的是将 n 项任务指派给 n 个代理来完成，使得指派后总的工作负荷总量最小。指派问题不仅在数学规划的理论研究方面有重要价值，也在实践中得到了广泛应用。该问题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。但是现实中任务的指派不仅仅需要考虑最小化总的工作负荷，工作负荷在代理之间是否分配均衡，或言公平，也是需要重点关注的现实问题。公平的指派结果不仅可以消减代理之间由于工作负荷差异较大产生的不满情绪，促进团队合作和增强工作积极性，而且也为企业、组织和团体的长期健康发展和组织文化培育提供支持。与经典指派问题和与之关联的二次指派问题受到广泛关注和深入研究不同，公平性指派问题却少有问津。针对上述问题，本文将尝试定义指派问题的工作负荷公平性指标，构建相应的优化模型，并给出求解模型的一种有效启发式算法。

下面按时间顺序对相关文献做一个简单介绍。文献[1]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。文献[2]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[3]提出了一种求解指派问题的新的离散粒子群算法。文献[4]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。文献[5]针对从 m 个人中派出 k ($0 < k \le \min\{m,n\}$) 个人去完成 n 项任务中的 k 项任务，且使总效率最高类的指派问题给出了一种反点求解方法。文献[6]对匈牙利算法寻找独立零的次序进行了改进，从而避免了匈牙利算法通常需要进行多次试分配的不足。

针对广义指派问题，即一人多事或一事多人的情形，文献[7]设计了一种改进的离散粒子群优化算法。文献[8]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[9]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。文献[10]讨论了从 m 个公司选取 k 个去投资 n 个项目中的 k 项的总收益最大的非均衡投资问题，提出了反点算法。在经典指派问题的最优解不唯一条件下，文献[11]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[12]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[13]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。文献[14]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[14]建立了对应指派问题模型，并证实该问题具有 NP-hard 特征。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[15]给出了一种改进的匈牙利算法。考虑指派中可能存在的冲突约束，文献[16]给出了求解该问题的改进分支定界算法。文献[17]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[18]提出了一种求解经典指派问题的改进粒子群算法。针对移动通信中的拥堵，文献[19]建立了以成本最小为目标的资源指派问题模型。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[20]建立了相应的指派模型，并给出了一种近似解求解方法。针对目标体系衡量存在差异的多目标指派问题，文献[21]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[22]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。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[23]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。

与现有研究相比，本研究的主要贡献在于：a) 提出了公平性指派问题，并定义了指派结果的公平性度量指标，构建了对应的优化模型；b) 对公平性指派问题进行了分类，对分类后的问题特征和求解策略进行了阐述；c) 设计了求解公平性指派问题的遗传算法，比较验证了算法的有效性，并对算法的参数进行了灵敏度分析。

2. 公平性指派问题

2.1. 优化模型

指派问题是指将 n 个任务分派给 n 个代理，每个代理需完成一项任务，任务不可分割。令 $i \in I$ 表示一个典型代理，I 为代理集合；$j \in J$ 表示一个典型任务，J 为任务集合。令 $x_{i,j}$ 表示第 i 个代理是否执行第 j 个任务，如果执行，其值为 1；否则，其值为 0。上述指派问题的基本约束如下：

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

$$\sum_{j \in J} x_{i,j} = 1, \forall i \in I, \tag{2}$$

$$x_{i,j} \in \{0,1\}, \forall i, j, \tag{3}$$

式(1)表示任意给定的一项任务会被指派给唯一代理去执行；式(2)表示任一代理仅会执行一项任务；式(3)

限定变量 $x_{i,j}$ 只能取值 0 或 1。令 x 为由所有 $x_{i,j}, i \in I, j \in J$ 构成的决策向量。为简化表述，令 X 表示满足约束式(1~3)的变量 x 的可行域。

一项任务 j 被指派给一个代理 i 的指派结果可被称为一个匹配(Matching)，记作 (i,j)。对应任一可行的匹配，有一个工作负荷。令 $c_{i,j}$ 为对应匹配 (i,j) 的工作负荷。则在给定 x 的条件下总的工作负荷可按下式计算：

$$z_1(x) = \sum_i \sum_j c_{i,j} x_{i,j}. \tag{4}$$

经典的线性指派问题以最小化总的工作负荷为优化目标，可表述为 $\min_{x \in X} z_1(x)$。利用匈牙利算法可对经典的线性指派问题加以有效求解，相应的算法计算时间复杂度为 $O(n^3)$。

与关注指派后整体的工作负荷大小不同，本文将目光转向指派后工作负荷在代理群体中的分布，力图平衡各代理的工作负荷，从而实现公平的任务指派。下面首先定义指派结果的公平性指标。指派后代理群体的平均工作负荷可按下式计算：

$$c_{av}(x) = \frac{\sum_i \sum_j c_{i,j} x_{i,j}}{n}. \tag{5}$$

平均工作负荷又称工作负荷均值。各代理的实际工作负荷与均值的差异体现了指派结果的公平性。最理想的公平状态是各代理的实际工作负荷相等，即等于工作负荷均值。基于上述观察，可定义工作负荷的公平性指标如下：

$$z_2(x) = \sum_i \sum_j \left(c_{ij} - c_{av}(x)\right)^2 x_{i,j}, \tag{6}$$

式(6)中，$c_{av}(x)$ 由式(5)计算得到。显然，$z_2(x)$ 的值越小，指派的工作负荷结果就越公平。理想条件下，$z_2(x)$ 的值最小可取 0。以最小化 $z_2(x)$ 为目标，以式(1~3)为约束，得到公平性指派问题的优化模型为：

$$\min_{x \in X} z_2(x). \tag{7}$$

2.2. 分类与性质

观察式(6)可知，如果 $c_{av}(x)$ 不因 x 而改变，是一个常数，则公平性指派模型(7)就会退化为一个线性的指派问题。因此我们将上述具有固定均值的公平性指派问题称为第一类公平性指派问题。下面给出该类问题的一种具体形式。

定理 1：假设下述条件 $c_{i,j} = a_i + b_j, \forall i \in I, j \in J$ 成立，那么就有 $c_{av} = \left(\sum_i a_i + \sum_j b_j\right)/n$。

证明：

$$c_{av} = \frac{\sum_i \sum_j (a_i + b_j) x_{i,j}}{n} = \frac{\sum_i \sum_j (a_i x_{i,j} + b_j x_{i,j})}{n} = \frac{\sum_i a_i \left(\sum_j x_{i,j}\right) + \sum_j b_j \left(\sum_i x_{i,j}\right)}{n} = \frac{\sum_i a_i + \sum_j b_j}{n}。\ \square$$

定理 1 表明，当一个匹配 (i,j) 的工作负荷 $c_{i,j}$ 等于与代理 i 独立相关的工作负荷 a_i 和与任务 j 独立相关的工作负荷 b_j 之和时，无论如何指派，得到的工作负荷均值为一定值。第一类公平性指派问题属于线性指派问题，因此可以利用匈牙利算法直接进行求解。

除了上述具有固定均值的公平性指派问题，大多数情况下 $c_{av}(x)$ 会随着 x 的改变而改变。我们将具有可变工作负荷均值的公平性指派问题称为第二类公平性指派问题。由约束(1~3)，易知下面的等式与取值约束成立：

$$x_{i,j}^2 = x_{i,j}, \forall i \in I, j \in J \tag{8.1}$$

$$x_{i,j}x_{i,k}=0,\forall j\neq k,i\in I \tag{8.2}$$

$$x_{i,j}x_{k,j}=0,\forall i\neq k,j\in J \tag{8.3}$$

$$x_{i,j}x_{k,l}\in\{0,1\},\forall i\neq k,j\neq l \tag{8.4}$$

$$x_{i,j}x_{k,l}x_{m,n}\in\{0,1\},\forall i\neq k\neq m,j\neq l\neq n \tag{8.5}$$

将式(5)的$c_{av}(x)$表达式带入式(6)，然后将式(6)做代数展开，易知得到的展开式中一定含有(8.5)形式的三次项。因此，可知一般情况下，问题(7)属于三次指派问题。三次指派问题是一类极难的 NP-hard 问题，目前还没有有效方法可以求得规模较大的该类问题的精确最优解。对该类问题常见的求解策略是设计具有针对性的启发式方法。

3. 进化算法的设计

本节将对遗传算法(Genetic Algorithm-GA)的具体的操作进行设计，以期使之适用于三次指派问题的求解。设计的算法在执行过程中保持了解的可行性。在各种算法操作(如交叉与变异)中始终保持解的可行性将可以避免大量的冗余计算。

3.1. 解的表达与关键算子

应用 GA 首先需确定一个合理的解的表达方式。在 GA 中，一个可行解也被称为一个个体。本文采取的指派问题可行解表达方式为$g=(j_1,j_2,j_3,\cdots,j_k,\cdots,j_n)$。这里 g 代表一个个体，而 j_k 代表指派给第 k 个代理的任务的序号。在算法中为了区分不同的个体，用 g_m 表示个体 m，其具体的表示式为 $g_m=(j_{1,m},j_{2,m},j_{3,m},\cdots,j_{k,m},\cdots,j_{n,m})$，其中 $j_{k,m}$ 表示个体 m 的第 k 个代理所指派的任务的序号。

GA 中个体 g 的适应值 f_g 定义为目标函数值的相反数，即 $f_g=-z_2(x)$ 满足 $x_{i,j_i}=1,\forall i\in I$。如无特别说明，算法中提到的随机分布均为相应的给定区间或集合上的均匀分布。在利用锦标赛法从群体中选择个体时，令函数 Tournament(G, k)表示先从群体集合 G 中随机取出 k 个个体，然后通过比较 k 个个体的适应值确定适应值最高的个体。而函数 Random(n)表示从 1 到 n 的自然数中随机取出一个数。

下面给出 GA 中交叉算子的具体操作：

步骤 0：设当前种群为 G，锦标赛规模为 s，而 $g_m=(j_{1,m},j_{2,m},j_{3,m},\cdots,j_{k,m},\cdots,j_{n,m})$ 为有待决定其具体组成的新个体。令 $j_{k,m}=null$，$\forall k$，这里 $null$ 表示对应的量待定。令集合 $G(m)=\{j_{k,m}\}_k$。

步骤 1：选取两个个体：$g_1=\text{Tournament}(G,s)$ 和 $g_2=\text{Tournament}(G,s)$。

步骤 2：确定两个交叉点：先计算两个随机位置 $n_1=\text{Random}(n)$ 和 $n_2=\text{Random}(n)$；然后确定两个具体的交叉点 $p_1=\min\{n_1,n_2\}$ 和 $p_2=\max\{n_1,n_2\}$。

步骤 3：对所有 $k\in\{1,2,3,\cdots,n\}$ 执行：如果($k<p_1$)或($k>p_1$)，令 $j_{k,m}:=j_{k,1}$。

步骤 4：执行如下循环迭代，完成新可行个体生成：

for k=1:n {

if ($j_{k,2}\notin G(m)$) {

for s=1:n { if ($j_{s,m}==null$) { $j_{s,m}=j_{k,2}$; $G(m):=G(m)\cup\{j_{s,m}\}$;break;}}

}

}

相对于交叉算子，变异算子操作比较简单。设变异系数为 β，而当前需对其执行变异操作的个体为 $g_m=(j_{1,m},j_{2,m},j_{3,m},\cdots,j_{k,m},\cdots,j_{n,m})$。变异的具体操作如下：

for k = 1:n {if ($\text{Random}(n)<\beta$) {令 $p=\text{Random}(n)$; $j_{k,m}:=j_{p,m}$; $j_{p,m}:=j_{k,m}$;}}

3.2. 算法步骤

基于上述的概念定义和算子，本文设计的具体 GA 算法执行流程如下：

步骤 0：初始化。设定种群规模$|G|$、锦标赛规模 s、复制比例ϵ、变异系数β、最大迭代次数$N_{\max}$。令当前迭代次数$\tau := 0$。

步骤 1：生成初始种群。生成一个数字 1 到 n 的随机排列，作为一个个体。按种群规模$|G|$，重复个体生成操作，形成初始种群$G^{(0)}$。

步骤 2：复制。利用锦标赛法，从当前种群选取$\text{Round}(\epsilon|G|)$个个体作为新一代种群的复制个体。这里函数$\text{Round}(\epsilon|G|)$表示取最接近$\epsilon|G|$的整数。

步骤 3：交叉和变异。利用上文介绍的交叉算子得到一个新个体，然后对其实施变异操作，最后得到一个可行的新个体。重复上述操作$\lfloor |G| - \text{Round}(\epsilon|G|) - 1 \rfloor$次，得到一个新个体集合。

步骤 4：合成新的种群。将步骤 2 和 3 得到的所有可行个体和当前种群中的最优个体组合，形成新一代的种群$G^{(\tau+1)}$。

步骤 5：终止判断。如果$\tau + 1 < N_{\max}$，转步骤 2；否则，终止算法，输出当前最优可行解。

4. 算例分析

本节将通过具体算例来验证模型与方法的有效性。利用 Java 程序语言实现了上节给出的 GA 算法，数值实验是在 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。算法的基本参数设定如下：种群规模$|G| = 100$、锦标赛规模$s = 5$、复制比例$\epsilon = 0.1$、变异系数$\beta = 0.015$、最大迭代次数$N_{\max} = 200$。本节结束部分将对上述参数取值进行灵敏度分析。

首先，分析一个$n = 10$的简单算例。对应的工作负荷矩阵$C_{10\times10}$如下：

$$C_{10\times10} = \begin{bmatrix} 30 & 53 & 26 & 63 & 49 & 59 & 28 & 23 & 49 & 32 \\ 20 & 26 & 25 & 69 & 42 & 25 & 24 & 46 & 23 & 40 \\ 62 & 28 & 44 & 47 & 66 & 21 & 41 & 62 & 57 & 58 \\ 59 & 64 & 67 & 66 & 49 & 25 & 20 & 39 & 55 & 61 \\ 32 & 28 & 63 & 66 & 60 & 34 & 38 & 59 & 20 & 49 \\ 34 & 67 & 44 & 62 & 21 & 50 & 26 & 45 & 32 & 46 \\ 37 & 23 & 55 & 23 & 66 & 40 & 49 & 38 & 58 & 47 \\ 43 & 33 & 28 & 32 & 23 & 21 & 61 & 53 & 62 & 64 \\ 26 & 60 & 36 & 45 & 43 & 47 & 32 & 68 & 35 & 41 \\ 64 & 51 & 51 & 31 & 36 & 56 & 33 & 33 & 47 & 68 \end{bmatrix}。$$

由于问题规模较小，200 次迭代的计算耗时小于 0.001 秒。对问题进行 3 次求解，算法的 3 次运算迭代表现在图 1 中给出。其中两次计算的最终结果相同，具体为：最终的$c_{av} = 47.4$，目标函数值$z_2(x) = 56.4$，而求得的最佳个体为(9, 8, 4, 5, 10, 3, 7, 1, 6, 2)。而另一次的计算结果为：最终的$c_{av} = 47.5$，目标函数值$z_2(x) = 68.5$，而具体的最佳个体为(9, 8, 3, 5, 10, 6, 7, 1, 4, 2)。从图 1 的数据可以看出 GA 算法具有较高的可靠性和一致的收敛特征。

为了进一步验证算法的有效性，表 1 中给出了 6 个不同规模的指派问题。表 1 中$[\underline{c}, \overline{c}]$表示对应工作负荷矩阵中工作负荷的上下限，而一个匹配的具体的工作负荷是随机从区间$[\underline{c}, \overline{c}]$中取出的一个整数。问题对应的工作负荷矩阵$C_{20\times20}$在表 1 之后给出，其它负荷矩阵限于篇幅，在此略去。

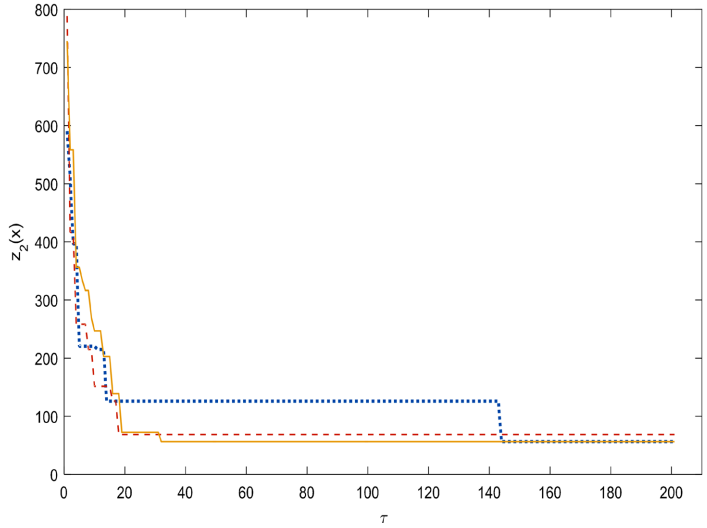

Figure 1. Performance of 3 times of implementation of algorithm
图 1. 算法 3 次计算的迭代表现

Table 1. Different assignment problem
表 1. 不同的指派问题

问题	1	2	3	4	5	6
n	10	15	20	25	30	35
$[\underline{c},\bar{c}]$	[20, 40]	[20, 60]	[30, 90]	[30, 90]	[10, 100]	[20, 70]

$$C_{20\times 20}=\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}$$

对应表 1 给出的 6 个问题，表 2 给出了分别利用 GA 算法和商业优化软件 Lingo 自带求解器计算的结果。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。从表 2 可知，Lingo 的计算时间从 15 秒迅速增加到 35 分 37 秒，而 GA 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，即小于机器可识别的 0.001 秒)。在迭代次数列，GA 算法总的迭代次数设为 200，小括号内的数值为算法收敛到最佳解时的迭代次数；对于 Lingo 软件而言，该列显示了 Lingo 自带求解器求解的迭代次数。从表 2 中的迭代次数数据可知，GA 算法实际需要的收敛迭代次数小于 181 次，而 Lingo 自带求解器求解所需的迭代次数从 16 万次增加到了 1450 多万次。

从 6 个问题求解的指派结果看，两种方法均得到了问题 1 的相同最优解；对于问题 2 和 3，GA 算法仅给出了较好的局部最优解，Lingo 给出了全局最优解；对问题 4 到 6，GA 算法仍可在短时间内给出了较好的局部最优解，而 Lingo 在经过数小时计算也无法给出了问题的一个可行解。表 2 中的数据也表明以追求指派结果的公平性为目的时，可能会造成指派后整体的工作负荷量较大，例如 GA 对问题 3 和 5 的计算，得到的工作负荷均值均靠近工作负荷取值范围的上界。因此，实际指派时，有必要在目标中加入代表工作负荷总量的加权项。考虑到上述转变并不会增加问题求解的难度，因此本文不再对此做进一步分析。上述各种比较证实了 GA 算法在处理实际问题时的高效性和具有处理大规模问题的能力。

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

方法	问题	指派结果	c_{av}	$z_2(x)$	计算时间(s)	迭代次数
GA	1	(6, 3, 7, 5, 8, 9, 2, 1, 4, 10)	22.7	8.1	<0.001	200 (81)
	2	(6, 2, 5, 10, 15, 8, 12, 7, 1, 4, 3, 13, 9, 14, 11)	43.6	57.6	<0.001	200 (37)
	3	(7, 8, 13, 11, 20, 9, 16, 6, 12, 2, 17, 18, 19, 10, 5, 15, 4, 14, 3, 1)	80.8	107.2	<0.001	200 (178)
	4	(8, 2, 25, 11, 22, 7, 3, 12, 20, 15, 6, 9, 13, 1, 18, 16, 5, 17, 24, 21, 19, 4, 14, 10, 23)	66.6	120.0	<0.001	200 (171)
	5	(30, 25, 8, 17, 29, 26, 4, 1, 22, 27, 7, 16, 24, 19, 3, 6, 10, 12, 13, 28, 11, 18, 23, 21, 14, 5, 9, 2, 20, 15)	81.7	346.3	<0.001	200 (145)
	6	(3, 5, 9, 29, 1, 27, 24, 12, 34, 17, 8, 2, 31, 35, 20, 25, 28, 6, 33, 10, 15, 22, 18, 4, 23, 14, 19, 16, 11, 30, 7, 26, 21, 13, 32)	52.83	130.97	<0.001	200 (181)
Lingo	1	(6, 3, 7, 5, 8, 9, 2, 1, 4, 10)	22.7	8.1	15	168,383
	2	(1, 5, 4, 11, 15, 6, 8, 7, 12, 10, 3, 14, 2, 13, 9)	39.8	42.4	411	3,883,007
	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	14,518,548

一般而言启发式算法的求解效果不仅与其具体的操作规则有关，也会受到设定的具体参数值的影响。因此，有必要对相关的算法参数进行灵敏度分析。以表 1 中给出的问题 3 为例，对 GA 相关的参数——种群规模 $|G|$、锦标赛规模 s、复制比例 ϵ、变异系数 β、最大迭代次数 $N_{\max}$ 进行了有关灵敏度计算，结果发现：除变异系数 β 外，其它参数的不同取值(如 $|G|$ 从 20 到 200 的范围内取值，s 从 3 到 30 的范围内取值，ϵ 在区间[0.1, 0.5]上取值，而 $N_{\max}$ 从 200 到 500 的范围内取值)，对算法的最终结果影响不大。变异系数 β 取不同值时对算法的最终结果有较为明显的影响，图 2 给出了当 β 分别取 0.005、0.015 和 0.05

时，算法的迭代计算表现。我们发现 β 的值过小或过大均不利于算法求得较好的指派结果，而当 β 的值靠近 0.015 时，算法的表现最好。上述结果说明在实际应用 GA 求解公平性指派问题时应对其变异系数进行必要的试算，从而确定 β 较理想的取值。

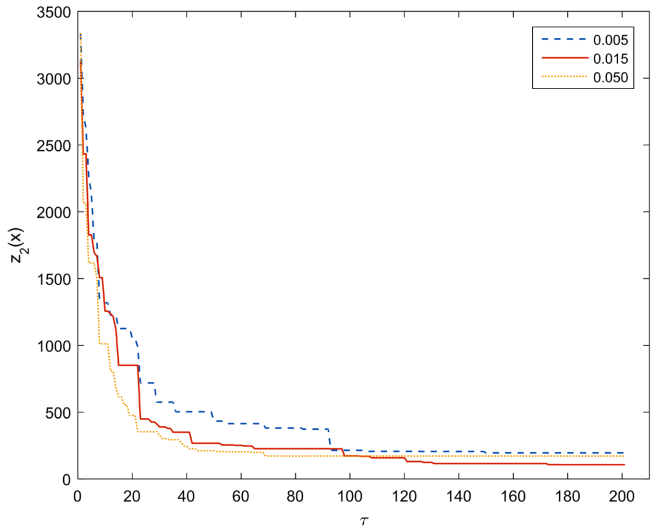

Figure 2. Influence of mutation coefficient β on the results of algorithm
图 2. 变异系数 β 对算法计算结果的影响

5. 结论

本文得到的主要研究结论包括：a) 以各个代理的工作负荷与负荷均值的差的平方和作为度量指派后工作负荷公平性的指标是合理的，以上述指标为优化对象建立的公平性指派问题模型在一般情况下属于具有 NP-hard 特征的三次指派问题；b) 与商业优化软件 Lingo 的计算结果相比，本文所设计的遗传算法具有计算耗时少，且可有效处理各种规模问题的优势，但是也存在一般情况下仅能得到局部最优解的缺点；c) 计算结果显示以公平性为指派唯一目标时，可能产生过大的总工作负荷量，因此实际应用时需将公平性和效率一并考虑；d) 算法的灵敏度分析显示，除变异系数外，算法的计算效果受其他参数具体取值变化的影响甚微，而过大过小的变异系数对算法获得较好解均有负面影响。

以本研究为基础，可以进一步深入研究的方向包括：分析指派中追求效率和公平性两者之间的相互影响；尝试其他启发式方法，进一步优化算法获取最优解的能力；结合实践，进一步丰富和改进问题的约束和优化目标。

基金项目

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)。

参考文献

[1] 黄龙生, 徐光辉. 有资格限制的指派问题的求解方法[J]. 运筹与管理, 2005, 14(1): 28-31.

[2] 李岩, 郭强. 非确定性指派问题的求解算法[J]. 计算机工程与应用, 2009, 45(15): 61-63, 66.

[3] 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[J]. 计算机应用研究, 2009, 26(11): 4091-4093, 4097.

[4] Hahn, P.H., Zhu, Y. and Guignardm Smith, J. (2010) Exact Solution of Emerging Quadratic assignment Problems. *International Transactions in Operational Research*, **17**, 525-552. https://doi.org/10.1111/j.1475-3995.2010.00763.x

[5] 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[J]. 运筹学学报, 2011, 15(3): 124-128.

[6] 周莉, 张维华, 徐射雕. 求解指派问题的一次性分配算法[J]. 计算机工程与应用, 2011, 47(18): 135-138, 152.

[7] 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[J]. 科技通报, 2013, 29(8): 130-132.

[8] 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[J]. 运筹与管理, 2013, 22(6): 34-38.

[9] 徐屹嵩, 王应明. 指派问题的纳什均衡解[J]. 运筹与管理, 2013, 22(4): 101-105, 110.

[10] 王立柱, 刘阳, 石洋, 孙军. 非均衡投资收益极大指派问题[J]. 沈阳师范大学学报(自然科学版), 2014, 32(3): 364-368.

[11] 徐屹嵩, 王应明. 指派问题的多重最优解的择优方法[J]. 运筹学学报, 2014, 18(2): 96-102.

[12] 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[J]. 系统工程理论与实践, 2017, 37(8): 2162-2170.

[13] Karsu, Ö. and Azizoglu, M. (2019) An Exact Algorithm for the Minimum Squared Load Assignment Problem *Computers and Operations Research*, **106**, 76-90. https://doi.org/10.1016/j.cor.2019.02.011

[14] Lesca, J., Minoux, M. and Perny, P. (2019) The Fair OWA One-to-One Assignment Problem: NP-Hardness and Polynomial Time Special Cases. *Algorithmica*, **81**, 98-123. https://doi.org/10.1007/s00453-018-0434-5

[15] Rabbani, Q., Khan, A. and Quddoos, A. (2019) Modified Hungarian Method for Unbalanced Assignment Problem with Multiple Jobs. *Applied Mathematics and Computation*, **361**, 493-498. https://doi.org/10.1016/j.amc.2019.05.041

[16] Öncan, T., Suvak, Z., Akyüz, M.H. and Kuban Altınel, İ. (2019) Assignment Problem with Conflicts. *Computers and Operations Research*, **111**, 214-229. https://doi.org/10.1016/j.cor.2019.07.001

[17] Malaguti, E. and Durán, R.M. (2019) Computing *K* Different Solutions to the Assignment Problem. *Computers & Industrial Engineering*, **135**, 528-536. https://doi.org/10.1016/j.cie.2019.06.025

[18] El-Ashmawa, W.H. and Ali, A.F. (2020) A Modified Salp Swarm Algorithm for Task Assignment Problem. *Applied Soft Computing Journal*, **94**, Article ID: 106445. https://doi.org/10.1016/j.asoc.2020.106445

[19] Liu, Y., Yang,Y., Wang, E., Liu, W., Luan, D., Sun, X., *et al*. (2020) A Fair Task Assignment Strategy for Minimizing Cost in Mobile Crowdsensing. *Proceedings of the* 26*th International Conference on Parallel and Distributed Systems*, Hong Kong (China), 2-4 December 2020, 44-53. https://doi.org/10.1109/ICPADS51040.2020.00016

[20] Gao, G., Ning, L., Ting, H., Xu, Y., Zhang, Y. and Zou, Y. (2020) Approximation Algorithms for the Partial Assignment Problem. *Theoretical Computer Science*, **838**, 231-237. https://doi.org/10.1016/j.tcs.2020.07.041

[21] 高原, 李仁传, 张合勇, 刘辉. 考虑目标差异的多目标指派问题研究[J]. 海军工程大学学报, 2020, 32(5): 102-106, 112.

[22] 林浩, 林澜. 一类二阶段指派问题[J]. 运筹与管理, 2021, 30(2): 97-101.

[23] Karsu, Ö., Azizoglu, M. and Alanli, K. (2021) Exact and Heuristic Solution Approaches for the Airport Gate Assignment Problem. *Omega*, 103, Article ID: 102422. https://doi.org/10.1016/j.omega.2021.102422