

基于员工熟练度差异的装配线人员配置的研究

梁志远

华南理工大学 广东广州 510640

【摘要】本研究通过构建创新的熟练度模型，分析员工熟练度差异对装配线配置的影响，并提出动态更新算法以提高配置的灵活性和准确性。研究显示，考虑熟练度差异的人员配置显著提升了生产效率和产品质量，为制造业效率优化提供了新视角和方法。未来研究可探索模型在不同环境的应用，并结合现代技术进一步提升效率。

【关键词】遗传算法；员工熟练度差异；装配线人员配置；生产效率优化

在制造业，装配线上员工的熟练度对生产效率和产品质量至关重要，但常被企业忽略。本研究强调合理的人员配置和技术及管理创新在提升效率中的作用，并提出一个基于熟练度的动态配置模型。同时，研究融合了装箱问题理论、操作熟练度理论和装配线人员配置理论，以优化生产过程，提高效率和质量。这些理论的应用对于提升企业竞争力在全球化和技术进步的今天显得尤为重要。

1 考虑员工熟练度的装配线人员配置模型构建

1.1 熟练度模型构建

由于人员具有技能多样性和水平差异的特点，这就增加了人员配置的灵活性，为了避免调度人员的主观意识太强，需要根据胜任力实时矩阵构建人员优化配置模型，使整条装配线的人员胜任力总和达到最大。

模型中， λ_j 表示第 i 个装配线作业人员相对于第 j 个装配工位的胜任力指数； x_j 表示第 i 个装配线作业人员配置于第 j 个装配工位；其中 $i=1, 2, \dots, n, j=1, 2, \dots, m$ 。

$$F = \max \sum_{i=1}^n \sum_{j=1}^m \lambda_j x_j \quad (1)$$

$$s.t. \sum_{i=1}^n x_j = 1 \quad (2)$$

$$\sum_{j=1}^m x_j = 1 \quad (3)$$

$$x_j = 1 \text{ 或 } 0, \forall i, j \quad (4)$$

$$\lambda_j = 0, x_j = 0, \forall i, j \quad (5)$$

本研究提出的装配线人员配置模型包括五个主要约束：①装配线人员胜任度总和最大化；②保证每个人员仅配置到一个工位；③每个工位仅配置一个人员；④决策变量限定为0或1；⑤确保每个岗位配置的人员能胜任其工

作。此外，研究通过熵权法确定指标权重，构建并归一化员工指标判断矩阵，计算各指标的熵值，进而确定熵权和各指标权重值。这一流程实现了指标权重的科学合理计算，为合理配置人员提供了有效的方法论。

1.2 熟练度模型的评价指标

采用专家评价法对模型准确度进行评估，该方法结合定量和定性分析，通过专家打分进行定量评价。首先，选定学习时间、完成数量、产品质量等评价指标，并为每个指标设定分值标准。然后，专家根据这些标准对指标进行评分并计算总分。最后，将评价结果与模型结果比较以衡量准确度。

本文以考虑员工熟练度差异的装配线的平衡为研究对象，从以下2个方面进行研究：

研究考虑人员熟练度因素的装配线平衡问题的数学模型及算法求解。传统的装配线主要通过作业分配达到负荷平衡，但由于不同员工对于同一工序中的生产效率存在较大的差异，简单的工作分配方式会导致效率低下的费用支出，并造成人员浪费。因此本文提出将人员可移动的因素纳入装配线平衡研究，通过高柔性的操作工来自动平衡混装线工作站的作业负荷波动，从而使得装配线能柔性地去满足市场所要求的节拍。

研究装配线的人员熟练度规划数学模型及算法求解。在获得作业分配/人员/班次的平衡方案后，对各人员所需的技能作出相应的培训/招聘规划，以确保混装线的平衡方案得以实施。

2 装箱理论在熟练度更新中的应用

2.1 算法的设计与实现

本研究提出了一种改进遗传算法（IGA）来解决装配线平衡问题中的工作站分配和成本控制。IGA通过增强初始种群多样性和创新工序分配方式，提供了新的解决方案。与

传统遗传算法相比，IGA在初始阶段引入种群多样性，并采用保留优先级关系的交叉方法，有效防止无效子代的产生，提高搜索效率。此外，算法采用大概率变异法调整工序序列，增强搜索能力，提升最优解出现的概率。在工序分配方面，IGA通过工人序列向量顺序读取工序序列向量，提高求解的灵活性和效率。这种方法不仅提升了装配线人员配置的精确度，还适用于复杂生产环境下的人员配置问题，具有重要的学术和实践价值。

遗传算法通过随机选择特征生成个体，确保初始种群的多样性。种群大小和结构由问题特性和计算资源决定。算法的关键在于交叉和变异，通过结合个体特征和基因序列的变化产生新的解决方案，增加多样性。通过计算种群适应度评估个体性能，高分个体优先用于后续操作，促进向优解的进化。

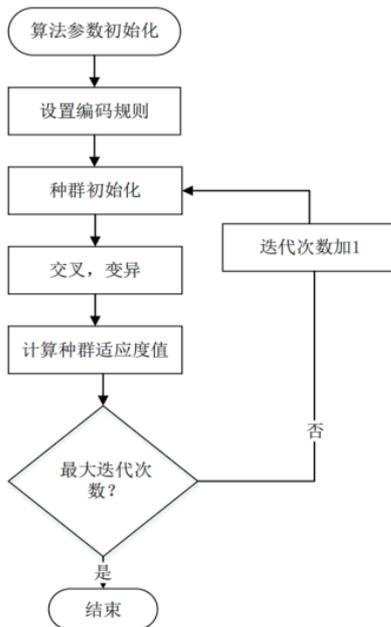


图1 遗传算法设计步骤流程图

2.2 实验步骤

实验通过模拟和分析了工人的熟练度与任务分配之间的关系，以及如何通过合理的任务分配提高生产效率。首先，实验对工人的熟练度进行了初始化，这是基于他们的经验年数、评估测试得分和培训时长，通过加权平均的方法计算出每个工人的熟练度。紧接着，任务分配考虑了员工的熟练度、任务难度、任务偏好和任务优先级。这种综合考虑能够实现任务的合理分配。

在工人完成任务后，他们的熟练度会根据任务的难度、完成质量和当前熟练度进行更新。这一过程依赖于一个基于加权因子的算法。为了评估任务分配和熟练度更新策略

的效果，实验采用了多种性能指标，如生产效率、生产周期时间和员工满意度等。

实验中还设置了多种参数，包括学习率、任务分配策略和实验周期，以评估它们对算法性能的影响。通过这些步骤，实验展示了如何通过动态调整工人的熟练度并合理分配任务来提高生产线的效率。这种方法的优势在于能够根据实际情况灵活调整，同时确保任务分配的公平性和效率性。

表1 学习率表

学习率	生产效率	生产周期时间	员工满意度
0.1	75%	10.5 hrs	7.8/10
0.5	82%	9.5 hrs	8.2/10
0.9	88%	8.3 hrs	6.9/10

表2 分配策略表

分配策略	生产效率	生产周期时间	员工满意度
熟练度分配	85%	9.0 hrs	8.5/10
任务队列分配	80%	10.2 hrs	7.5/10

表3 试验周期表

实验周期	生产效率	生产周期时间	员工满意度
10个任务/周期	83%	9.7 hrs	8.0/10
20个任务/周期	81%	10.0 hrs	7.7/10

实验结果显示，合理的学习率、任务分配策略和实验周期的选择对提高生产效率和员工满意度至关重要。

3 仿真模拟

在这个实验中，研究模拟了工人在不同工作站上的工作情况，并记录了他们的熟练度得分。

仿真实验代码

```

np.random.seed(42)

def simulate_scores(worker, workstation):
    base_score = np.random.randint(60, 90)
    if worker == 'W1' and workstation == 'WS1':
        return base_score + np.random.randint(5, 10)
    elif worker == 'W2' and workstation == 'WS2':
        return base_score + np.random.randint(5, 10)
    else:
        return base_score

simulated_scores = []
for worker in df['Worker'].unique():
    for ws in df['Workstation'].unique():
        score = simulate_scores(worker, ws)
        simulated_scores.append([worker, ws, score])
sim_df = pd.DataFrame(simulated_scores,

```

