# 用遗传算法解决作业车间调度问题 算法

**摘要：**遗传算法（GA）是用于解决理论计算机科学中的优化问题的搜索算法。作业车间调度（JSS）问题是组合优化问题，其主要目标是找到具有最小制造时间的调度表，以处理m台机器上的j个作业。 JSS问题中的作业处理计划受到一些约束，例如一次只能在一台计算机上处​​理一项作业，必须按一定顺序处理作业的操作，并且必须抢占任何操作禁止的。在本文中，我们提出了一种新的遗传算法（GA）以解决JSS问题，该问题使用新的遗传表示（编码）来安排作业和机器分配。继之以遗传表示，一个初始种群是随机产生的。然后将遗传算法的交叉和变异算子应用于种群，以创建新的后代，直到达到某种终止标准为止。

到达。此外，进行了实验以证明所提出的遗传算法的性能和适用性。

**1.引言**

调度作业被认为是制造过程管理和规划中最关键的问题。调度与分配作业到生产资源有关，并指定顺序以优化某些目标功能。在制造环境中，调度取决于车间环境，例如作业车间，流程车间和开放式车间[1]。作业车间调度（JSS）问题可以解释为：作业集j包含n个作业，并且这些作业在从机器集m中选择的一台机器上进行处理。每个作业都由以下操作组成：O = O1，O2，...，On。每台机器可以一次处理一个作业的操作，并且当一项操作分配给一台机器时，该机器必须完成操作的处理而没有任何中断。此外，必须以特定的给定顺序处理作业的操作。问题是要找到在机器上进行操作的时间表，同时要牢记将最小化制造周期的优先约束。 Makespan（Cmax）是每个作业的所有操作完成的时间。

作业车间调度是一个众所周知的且强烈的NP难题[2]，并且已证明在计算上具有挑战性。 JSS问题分析为解决更复杂和现实的系统中面临的调度问题提供了相关的见识。因此，启发式方法更适合于车间调度[3]。对于一类组合问题，遗传算法被认为是最著名的优化技术[4，5]。

Holland [6]提出的遗传算法（GA）是基于进化过程的搜索算法。 GA在解决理论计算机科学中的优化问题方面众所周知。在遗传算法中，如果总体中的解得到适当编码并通过模仿遗传算法的过程，它就有能力为现实世界的问题发展解[7]。遗传算法的分析已于几年前开始，以了解遗传算法的工作原理以及如何使用它们来获得最佳解决方案[8]。在将GA应用于问题时，必须设计给定问题的合适表示（编码）。还需要一个适应度函数，该函数用于从总体中选择单个解决方案。在运行期间，选定的解决方案将进行交叉（复制）和变异操作以生成新的后代。这里的主要思想是，后代的表现将比其相应的父母更好。戴维斯[9]在1985年提出了第一个基于遗传算法的调度问题。此后，遗传算法被广泛用于解决调度问题。在这项研究中，我们提出了一种遗传算法来解决JSS问题。

本文的其余部分分为三个主要部分。在第2节中，讨论了GA和JSS问题领域的文献综述。在第3节中，通过详细介绍编码机制，适应度函数，选择标准，交叉和变异算子用于生成后代，提出了拟议的遗传算法。在第4节中进行了实验，以证明拟议的GA的性能，第5节中得出了结论。

**2.文献综述**

在文献中，已经采用了多种遗传算法来解决JSS问题。这些方法在使用的表示方案和GA，约束处理和追求的目标方面彼此不同。但是，所有方法都有一个共同点：需要知识才能生成具有竞争力的工作处理时间表。 1985年，Davis [9]首次使用遗传算法解决了JSS问题。 Jain和Meeran [10]提供了用于解决JSS问题的方法的综合综述。他们得出结论，在解决JSS问题的各种技术中，元启发式技术最适合JSS问题。启发式和GA的组合使用称为元启发式。随着1980年代中期GA的普及，不同的研究人员使用GA来解决JSS问题。

Li和Chen [11]定义了两行染色体结构，该结构基于程序的工作顺序和机器分布，以最大程度地减小JSS问题中的有效期。 Lee等。 [12]提出了一个框架，该框架利用遗传算法以及机器学习和启发式空间来找到作业释放顺序的顺序以及在每台单独的机器上分配作业的顺序。根据Onwubolu和Davendra [13]，遗传算法是用于优化JSS问题的最佳技术。而一些研究者，Bierwirth等。 [14]，Meeran和Morshed [15]认为优化GA的混合遗传算法性能远远优于标准遗传算法。

为了解决车间问题，实践中还使用了GA以外的其他技术，例如枚举技术，基于近似的方法和人工神经网络。威廉姆森等。文献[16]证明，只要一台机器上所有操作所需的合计准备时间接近三，就可以在多项式时间内确定计划周期为三的计划的存在。车间枚举方法的基本中心是攀爬和约束技术。 Lageweg等人提出的两个最常规的分支过程是“生成活动时间表”（GAS）和“解决基本冲突”（SEC）。 [17]。对于实际应用，标准GA可能不够灵活，当问题复杂时，这一点变得越来越明显。此外，Uckun等。 [18]得出结论，GA迅速倾向于可能的解决方案。

**3.拟议的遗传算法**

我们提议的GA的总体结构如下所述：

1.表示形式：群体中的每个染色体（溶液）都以一种格式表示，该格式将应用其他GA运算符；

2.选择：从种群中选择溶液进行繁殖；

3.产生后代：通过更改操作顺序（顺序交叉）和更改机器对操作的分配（均匀交叉）来获得新的解决方案；

4.突变：随机翻转时间表中的某些位；

5.健身功能：获得的每个日程表的完成时间

在计算完步骤4之后；

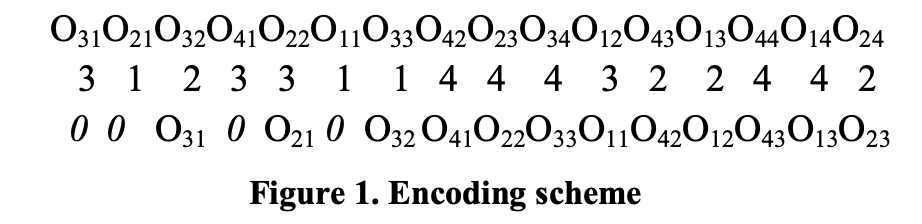
6.终止标准：固定世代数为

到达。 达到终止条件后，GA会返回最佳计划以及相应的有效期作为输出。

接下来说明我们用于解决JSS问题的上述六个步骤。

**3.1 编码方案**

编码是GA至关重要的要素。 使用GA寻找问题的最佳解决方案很大程度上取决于编码方案。 GA通常会处理字符串填充。 初始种群是随机产生的。 每个染色体代表JSS问题的解决方案（时间表）。 时间表中的作业分配给机器的操作由染色体的基因描述，并且染色体中出现的操作顺序表示操作序列。 在我们提出的遗传算法中，我们使用三行结构表示候选染色体（时间表）。 图1显示了4×4 JSS问题的三行结构。



在图1中，第一行代表作业的操作。例如，O31表示Job3（J3）的操作1，O21表示Job2（J2）的操作1，O33表示Job3（J3）的操作3，O43表示Job4（J4）的操作3，依此类推。第二行代表每个操作的机器分配。 3表示操作O31将转到机器编号3进行处理，1表示将机器1分配给操作O21，依此类推。第三行显示作业操作的操作依赖性。在此，0表示要执行O31，没有优先级限制退出。其他作业的首次操作也是如此。 O31表示操作O32的处理将在处理操作O31时开始。

**3.2。家长选择**

在表示进度表之后，下一步是从总体中选择进度表。选定的日程表称为父日程表。使用不同的技术来选择父时间表。为了避免过早收敛，我们从总体中随机选择两个计划。这意味着人口中的所有个人（时间表）都具有与父母相同的选择机会。 3.3。交叉算子

GA的交叉运算符用于交换作业的操作以及将特定的操作分配给两个选定父级中的计算机，以最大程度地缩短工期。在我们提出的算法中，我们将订单交叉用于两个父计划之间的洗牌。而统一的交叉算子用于对分配给操作的机器进行改组。

**3.3.1。订单交叉**

我们已经使用订单交叉[19]从父母的时间表中选出孩子的时间表。为了满足优先级约束，我们对顺序交叉运算符设置了2个限制。这些限制如下：

交叉点必须大于，其中J

代表作业总数，而O代表作业总数。

没有连续执行同一作业的操作。我们已经用一个简单的例子解释了订单转换。让两个选定的父时间表为：

P1：O31O21O32O41O22O11O33O42O23O34O12O43O13O44O14O24

P2：O11O21O12O41O22O31O13O42O23O14O32O43O33O44O34O24例如，选择任意两个交叉点j和k（j = 13和k = 16）。从父级P1获得的子字符串s（P1）包含P1的那些操作，这些操作占据j和k之间的位置，即s（P1）= O13O44O14O24。我们现在

在P2中删除操作s（P1），获得：

P` = OOOOOOOOOOOO。 2 11 21 12 41 22 31 42 23 32 43 33 34

最后，我们在P`中插入子字符串s（P），结果是

新时间表：

七月八月

21

因此，在订单交叉之后，我们获得一个调度，该调度在j和k之间具有来自P1的对应操作，而在其他位置，则是列表P`的操作，即

2

作为P2和s（P1）的元素之差获得。订单交叉操作后产生的后代是：

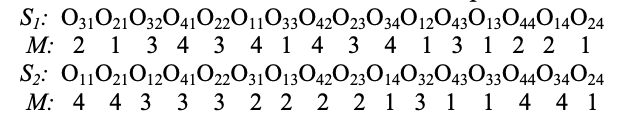
OS1: O31O21O32O41O22O11O42O23O12O13O14O43O33O44O34O24

OS2: O11O21O12O41O22O31O42O23O32O33O34O43O13O44O14O24

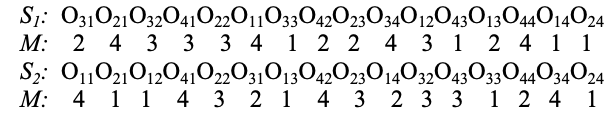
产生的后代时间表表明，满足了操作的优先约束。 每个操作都在执行其依赖操作之后执行。

**3.3.2。 均匀交叉**

在JSS问题中，预先将机器分配给作业的特定操作。 我们使用统一分频器来更改作业操作之间的机器分配，以最大程度地缩短工期。 在统一转换中，当两个时间表中的同一作业操作具有不同的机器分配时，则将其中的机器编号改组。 以下示例说明了用于机器改组的均匀交叉的工作。 令在订单交叉操作之后获得的两个时间表是：



经过统一交叉后，时间表S1和S2中为每个工序分配的新机器将是：



**3.4变异算子**

Lee等。 [20]在他们提出的遗传算法中使用了优先保留移位突变（PPS）算子。 我们还使用了PPS运算符，几乎没有进行修改。 以下步骤显示了我们修改后的PPS运算符的工作方式，以及如何统一统一交叉运算符后获得的时间表。

从时间表中选择任何位置i。

找到先前操作的位置（po）

步骤1中选择的操作

在所选位置i和范围内选择位置x

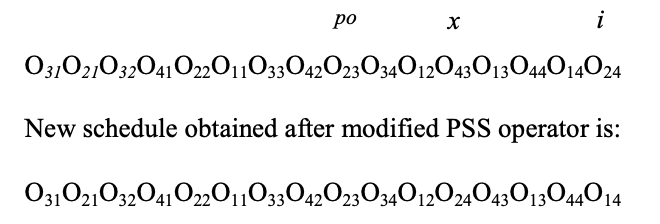
po。

将位置i上的操作移至位置x。

已将操作移到位置x或更高位置

行动领先一步。

假设选择位置16th的操作作为变速操作。 它的位置是第9位。

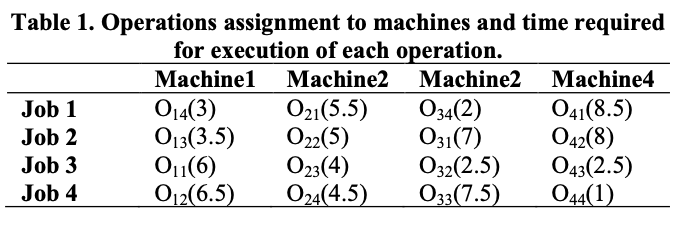


**3.5。 健身功能**

JSS问题的主要目标是找到具有最低制造期限值的计划。 可以从每台机器上的操作处理顺序和操作处理开始时间的信息中计算出制造跨度。 我们选择了makepan作为适应度函数。 由于我们希望找到具有较低制造期值的进度表，因此，遗传进化将更喜欢具有较低适应性值的进度表。 在每一代中，将评估所有选定的计划，并记录最佳个人。

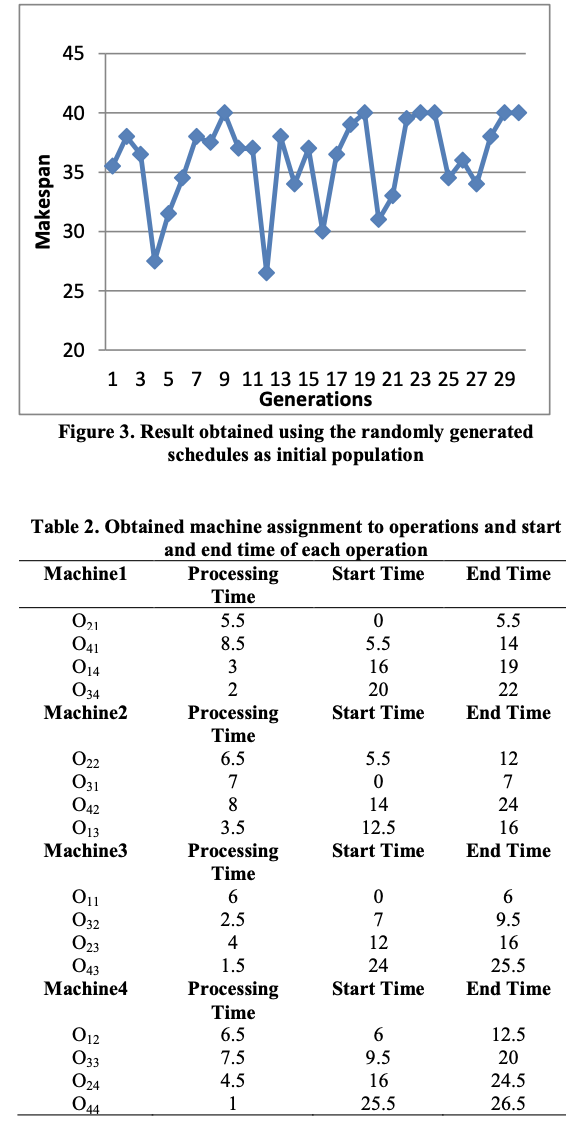
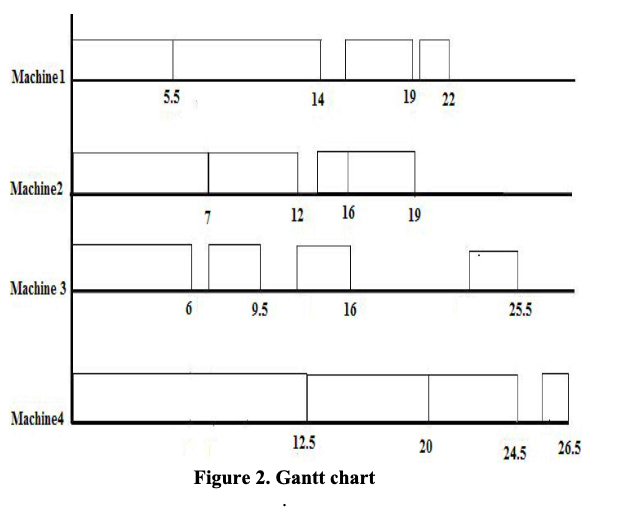
**4.结果与讨论**

我们已经将提出的遗传算法应用于4×4 JSS问题，以检验所采用方法的适用性和有效性。 表1给出了每个作业的操作处理顺序和每个操作的处理时间。



我们考虑的初始种群为50，并且在交叉率为0.8且突变率为0.2的条件下运行GA 30次。 表2中的结果表明，我们提出的遗传算法可以提供与其他方法一样好的结果。 从表2还可以清楚地看出，最后处理的作业是计算机4上的O44。因此，此JSS问题的makepan值为26.5。

表2中的结果还提供了有关每台机器的工作顺序，每项操作的启动和完成时间的信息。 例如，在机器1上，O21在时间0开始，在5.5结束。 然后，操作O41在时间5.5开始，并在时间14结束，依此类推。 加工路线的甘特图如图2所示。每一代后获得的制造时间如图3所示。从图3可以清楚地看出，制造时间的最佳值出现在第12代。



**5**

本文提出了一种用于作业车间调度（JSS）问题的遗传算法（GA）。 计算结果表明，提出的遗传算法是有效的。 遗传算法和JSS问题为演化计算提供了框架，并为组合优化问题提供了见识。 尽管GA的缺点是找到更好的解决方案需要花费大量时间，但它为演化计算提供了灵活的框架。 将来，我们希望了解拟议的遗传算法将如何应用于较大的问题，以查看其在重大问题上的表现。