基于改进多表达式编程算法的木材染色配方预测

摘 要：本研究通过使用计算机配色技术实现对普通木材的高精度染色，从而制造出视觉上与珍贵材无法区分的替代品，以降低对珍贵木材的依赖。在基因表达式编程的基础上，提出了多表达式编程（MEP）算法来预测染料配比，以优化染色后木板的光谱反射率。MEP 算法能够处理多染料之间复杂的交互关系，得到较为直观的函数关系式，对遗传算子概率进行自适应改进，并采用并行程序设计提高函数挖掘效率，在配色预测中得到结果为 0.313 的配方相对偏差结果，此外，我们将 MEP 算法的预测结果与传统神经网络和遗传算法进行了比较，结果表明 MEP 算法具有优异的预测性能。

关键词： 木材染色；基因表达式编程；多表达式编程；计算机配色；遗传算法；光谱反射率

Abstract: This study aims to achieve high-precision coloring of ordinary wood using computer coloration technology, creating substitutes that are visually indistinguishable from precious wood, in order to reduce dependence on precious wood. Based on gene expression programming, a multi-expression programming (MEP) algorithm is proposed to predict dye formulations and optimize the spectral reflectance of dyed wood boards. The MEP algorithm can handle complex interactions among multiple dyes and obtain more intuitive functional relationships. It improves the adaptive probability of genetic operators and employs parallel program design to enhance function mining efficiency. The predicted relative deviation of the formulation in color prediction using MEP algorithm is 0.313, which outperforms traditional neural networks and genetic algorithms.

Key words: Wood staining；Gene Expression Programming; Multi-Expression Programming; Computer Color Matching; Genetic Algorithms; Spectral reflectance.

随着全球气候变化、人口压力和非可持续的伐木行为影响下，珍贵木材的稀缺性日益显现。这种状况不仅对建筑、家具、装饰艺术等领域构成了巨大挑战，也引发了关于环境可持续性和保护生物多样性的重要讨论。这种压力的背景下，科学家和工程师已经转向使用更环保和可持续的方法，将普通木材转化为视觉上和触感上类似于珍贵木材的替代品，以此来减少对珍贵木材的依赖。在这种转变中，木材染配色技术的作用变得尤为重要。

木材染色要求具备高水平的技术精度和科学知识，我们的目标是通过计算机配色技术，准确地模拟珍贵木材的色彩，从而更精确地染色普通木材。通过这种方式，我们可以制造出在视觉上与珍贵木材无法区分的替代品，而不必支付高昂的环境代价。

近些年来，神经网络及深度学习已经被用于计算机配色的预测中。Chen M 通过结合 CNN、MLP 和 ResNet 三种神经网络模型对纺织物染色系统进行配色预测，在色差评估中取得较好的性能[1]；Furferi R 将 Kubelka-Munk 理论与人工神经网络结合预测织物反射率，从而提高了 Kubelka-Munk (KM) 理论上的性能[2]；Wang Q 为了使就胶囊颜色配置更加精确，利用粒子群算法优化 BP 神经网络对颜色进行预测，优势明显[3]；李文峰通过改进 elm 算法对木材染色配方进行预测，得到较好的配色效果[4]。另外，遗传编程算法也广泛应用于计算机配色中，如 Li H 将遗传算法与 BPNN 算法结合应用于牙科计算机配色[5]；应当注意，由遗传算法（GA）和遗传程序设计（GP）中发展而来的 GEP 算法因为其编码简单且可以处理复杂问题的优点已经被逐渐应用在预测问题中，GEP 吸收了遗传算法（GA）和遗传程序设计（GP）的优点，可以在较小的搜索空间得到更复杂的解，可以自由地改变基因长度和数量，从而更好地适应问题的复杂性。这种灵活性使得 GEP 在处理复杂问题时更具有优势。而 MEP 是在 GEP 的基础上进行改进的，使得单个染

色体可以包含多个表达式，在代码复用性、预测精度等方面要远好于 GEP。

本研究中，我们通过染色后木板的光谱反射率对三红染料浓度进行预测，三种染料对光谱反射率的影响是相互作用的，因而我们需要处理的不仅仅是一个目标变量，而是多个目标变量之间存在交互或交织关系的复杂情况。而多表达式编程算法可以有效地解耦这些交织的问题。

迄今为止，还没有学者应用 MEP 通过使用大量数据来预测木材染色染料配比，本研究旨在填补这一空白，并充分利用 MEP 优势解耦染料配比和光谱反射率之间的关系，得到两者之间的关系表达式，能够在获得珍贵材光谱反射率的基础上，使得配方预测仅用简洁的数学关系式就可得到。

1．数据采集

对于木材染色配方需要大量的数据训练来保证预测的准确性。在数据的获取上，我们用三种染料对木材进行染色，并获取染色单板的光谱反射率。

1.1 试验材料和设备：樟子松单板（将樟子松进行旋切，并加工成 40mm×70mm×1mm 尺寸大小的染色单板）、漂白剂（H_2O_2 溶液），活性红（X-3B）染料、活性黄（X-RG）染料和活性蓝（X-3G）染料，渗透剂（水性 JFC 溶液）、固色剂（无水碳酸钠）、促染剂（NaCl）；芬兰 SPECIM 高光谱分析仪、数显恒温水浴锅（上海力辰邦西仪器科技有限公司）、电子天平、烧杯、202-□型电热恒温干燥箱（天津泰斯特仪器公司）。

1.2 试验步骤：

1) 漂白：漂白处理可以除去木材中部分抽提物，从而提高上染率。在染色前对木材单板进行漂白操作，配置 500ml 浓度为 4%的 H_2O_2 漂白液，并加入各 0.5g 的 Na_2SiO_3 和 Na_3PO_4，将预制好的染色单板放入盛有漂白液的烧杯中，将烧杯放在温度为 65℃的恒温水浴锅中，设置浴比为 1：20，等待 2h 取出单板并用清水漂洗去表面残夜。

图 1 漂白前后木材单板

Fig. 1 Wood veneer before and after bleaching

2) 染色：根据预先设定好的 800 组染料配比数据对木材进行染色，根据每一组染料浓度配置 500mL 的染液，加入 0.5mLJFC 溶液和 15g/L 的 NaCl，将樟子松单板置于烧杯中，将烧杯置于温度为 65℃的恒温水浴锅中加热 2h 后放入 20g/L 的无水碳酸钠固色，30min 后去除木板用清水漂洗表面残夜。

3) 干燥：将染色后的樟子松单板放入干燥箱中干燥至含水率为 6%-8%。

部分染色单板如图 2：

图 2 部分染色后单板

Fig. 2 Partially stained wood veneer

4) 光谱数据获取：将干燥后的染色单板用芬兰 SPECIM 高光谱分析仪拍照，并用专用软件提取各单板的光谱反射率，提取时，为了保证数据的准确性，随机选取单板的五个染色均匀的部位，提取波段在 400nm-700nm 的平均光谱反射率，每个波长间隔约为 2.6nm

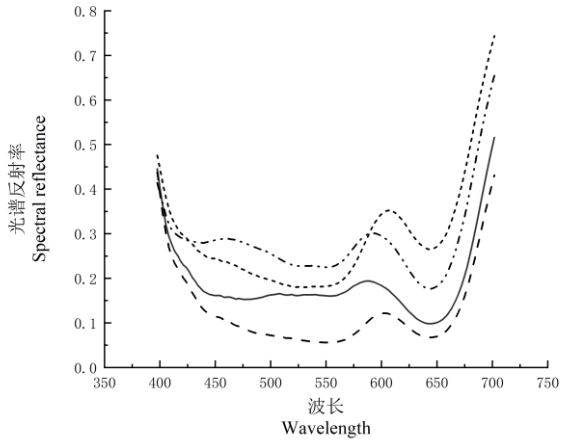

图 3 部分样本光谱反射率曲线

Fig. 3 Partial sample spectral reflectance curve

表 1 某染色单板部分波长光谱反射率数据展示（426.49nm 之后数据略）

Tab. 1 Partial wavelength spectral reflectance data for a stained wood veneer (data after 426.49nm omitted)

波长 Wavelength /nm	光谱反射率 Spectral reflectance
397.66	0.363
400.28	0.322
402.9	0.283
405.52	0.251
408.13	0.226
410.75	0.206
413.37	0.198
416	0.188
418.62	0.177
421.24	0.168
423.86	0.163
426.49	0.155

2．数据特征提取

特征提取在处理高维度数据时是一个关键的步骤，它有助于降低数据的维度，同时保留数据的主要结

构和信息，进而提高机器学习模型的性能。本研究测得光谱数据是 397nm-700nm 可见光波段之间的光谱反射率，数据集包含大量的特征，然而并非所有特征都对预测任务有作用。我们采用了随机森林模型评估每个特征的重要性，这是通过计算每个特征在所有决策树中降低的不纯度的平均值来实现的[6] 。如果一个特征在多个决策树中都能有效地降低不纯度，那么我们就认为这个特征非常重要。

特征重要性的计算基于决策树的分裂过程，对于光谱反射率的特征提取实则是回归问题，对于回归问题，假设我们有 m 棵树，对于每棵树 t 都有其节点分裂过程。特征 i 带来的平均平方误差减少量可以表示为：

$$MSER_{i(t)} = \Sigma \left(N_{node} / N_{total}\right) * \left(MSE_{parent} - MSE_{node}\right) \quad (1)$$

N_{node} 是在每个节点分裂后得到的样本数量； N_{total} 是总的样本数量； MSE_{parent} 是父节点的均方误差； MSE_{node} 是在每个节点分裂后得到的子节点的均方误差。 $MSER_{i(t)}$ 与特征 i 的重要性成正比关系。

我们将数据集分割为 80%的训练集和 20%的验证集，然后在训练集上训练了一个随机森林回归模型。模型训练完成后，我们获取了每个特征的重要性（百分比表示），并将特征按照重要性排序。

我们选取了重要性排名前 20 的波长反射率进行后续的模型建立，选择的波长如表 2：

表 2 特征提取后部分单板光谱反射率数据样本

Tab. 2 Partial sample spectral reflectance data of wood veneer after feature extraction

波长 Wave length/nm	光谱反射率 Spectral reflectance		
	染色单板 1 Stained veneer1	染色单板 2 Stained veneer2	染色单板 3 Stained veneer3
455.43(A)	0.103683191	0.09272542	0.10038679
447.52(B)	0.108366766	0.09744325	0.10712857
468.62(C)	0.08964023	0.07834418	0.08100179
546.26(D)	0.05238029	0.04378802	0.04114893
561.59(E)	0.05167837	0.04296519	0.04049571
473.9 (F)	0.084326707	0.0744829	0.07704929
450.16(G)	0.106992747	0.09571972	0.10341464
463.34(H)	0.097425299	0.08591981	0.09074321
444.89(I)	0.109337201	0.09887828	0.10975857
397.66(J)	0.362720435	0.35026645	0.43609464
542.91(K)	0.051983191	0.04348678	0.04202286
400.28(L)	0.321950043	0.30848253	0.37411821
460.7(M)	0.099037969	0.08770173	0.0919075
452.79(N)	0.105877816	0.09458296	0.10086071
556.25(O)	0.050687031	0.04198629	0.03976893
402.9 (P)	0.282613695	0.26939781	0.32783464
545.57(Q)	0.051237713	0.04275616	0.04091929
550.91(R)	0.050914249	0.04227326	0.04035679
532.25(S)	0.054378882	0.04620573	0.04564036
569.61(T)	0.054301493	0.0449264	0.04219857

3．模型描述

3.1 基因表达式编程

基因表达式编程（Gene Expression Programming, GEP）是一种遗传算法，它是由 Candida Ferreira 在 2001 年提出的。它在遗传编程（Genetic Programming）的基础上，采用了线性染色体结构和基因表达机制，从而改善了遗传算法的性能和效率[7] 。与传统的遗传规划 (GP) 不同，GEP 将基因型与表型分开，从而更有效地搜索解决方案空间。在本文中，我们基于 GEP 生成高度准确和可解释模型的潜力，将 GEP 用于木材染色领域中进行研究。

3.1.1 GEP 编码原理

GEP 的核心是个体的概念，个体由一个或多个基因组成。每个基因都由一个包含功能和末端的头部和一个仅包含末端的尾部组成[8] 。尾部的长度是根据头部功能的最大数量计算的，从而使表型表示具有更大的灵活性，头部基因包括函数和函数符号集（“+”、“-”、“×”、“/”、“ln”、“sin”等）和终端符号集（输入），尾部基因只包含终端符号集。基因的头部和尾部存在固定的关系，假设头部长度为 h，基因所包含得函数集中所有函数得最大操目数为 n，则尾部基因长度 e 为：

$$e = h \times (n-1) + 1 \tag{2}$$

为了增强算法的搜索能力和适应性，在基因尾部附加 DC 域[9] ，DC 域在基因编码中引入了额外的信息，在算法中我们采用随机 DC 域，DC 域的长度等于尾部基因长度，即：

$$DC_{length} = e \tag{3}$$

基因结构如图 4 示例所示:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
+	?	*	+	?	*	*	a	a	a	?	?	a	a	a	6	8	0	8	3	2	9	5

图 4 基因结构示例

Fig. 4 Example of Gene Structure

上图划分依次为头部基因、尾部基因和附加 DC 域。每个 GEP 基因都被解码成一个表达式树，终点“?”代表随机常数，从左到右，由 DC 域中的符号替换。假设随机 DC 域生成的数组为[$a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\ a_{10}$]，将其转化为表达式树如图 5。

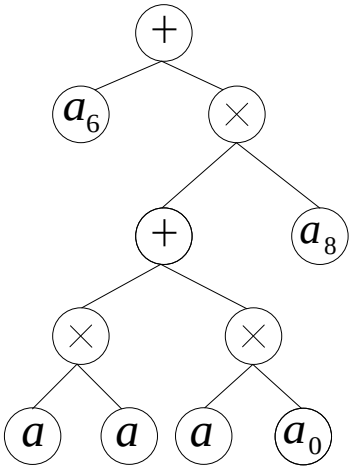

图 5 对应的表达式树

Fig. 5 Corresponding Expression Tree

其对应的表达式为 $(a^2+6\times a_0)\times a_8+a_6$。

3.1.2 GEP 的遗传算子

1）变异算子：将父代染色体作为变异对象，按照变异率随机选择基因片段进行变异，若变异发生在头部，则对应基因片段可以变异为任意函数符号集和终端符号集，若变异发生在尾部，则只能变异为终端符号集[10] 。

2）重组算子：重组又分为多点重组、两点重组和单点重组，单点重组是指在两个父代染色体上选择一个点，然后交换这个点后面的所有基因；两点重组是指在每个父代的染色体上选择两个点，然后交换这两个点之间的所有基因；在多点重组中，首先随机选择 N 个交叉点。然后，从第一个基因开始，子代会交替地从两个父代复制基因。每当我们到达一个交叉点时，就会改变我们正在复制基因的父代。多点重组示例如图 6 所示，选择 3 个交叉点，分别在 2、5、7：

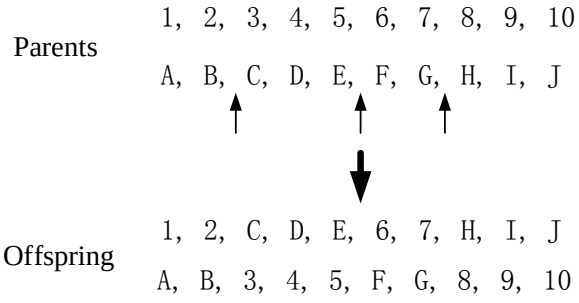

图 6 多点重组图例

Fig. 6 Multi-point Recombination Example

3）插串：包括 IS 插串和 RIS 插串，在 IS 插串中，随机选取一小段序列（通常是尾部的一部分），并插入到头部的某个位置。头部通常对应于表达式树的更高层次的节点，有助于改变头部的结构和行为；RIS 插串与 IS 插入类似，只是被选取的序列被插入到头部的开始位置（也就是根位置）。如图 7 示例，在 IS 中将 D，E 插入头部第二个位置，在 RIS 中将 D，E 插入跟位置。

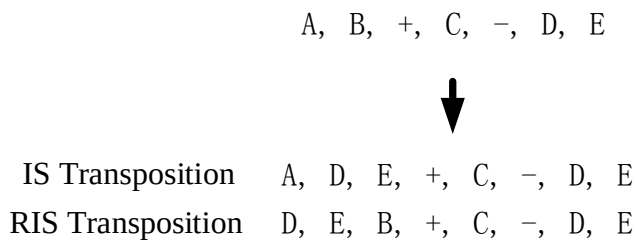

图 7 插串和跟插串实例

Fig. 7 Insertion and Root Insertion Examples

3.2 多基因表达式编程算法

为了更好地模拟光谱反射率和三种颜色浓度函数挖掘复杂的过程，引入多基因染色体的概念[11] 。在遗传算法中，染色体是解决方案的一种表示，而基因则是解决方案的组成部分。情况下，每个染色体表示一个可能的模型，其中每个基因对应模型中的一个颜色浓度。因此，我们的染色体是由三个基因组成，分别对应三种颜色的浓度。

图 8 多基因染色体

Fig. 8 Multi-gene Chromosome

利用多基因染色体的优点是，我们可以更好地捕捉到各种颜色浓度之间的相互关系，同时保持了模型的适应性。

3.3 多表达式编程算法

传统的 GEP 中，基因表达式的长度是固定的，这可能限制了算法在处理复杂问题时的表达能力。这种固定长度的限制可能导致性能下降。针对 GEP 所体现出的效率较低、精度不足的问题，在引入多基因表达式编程算法的基础上我们继续引入多表达式编程算法（Multi-expression programming, MEP）[12] 。MEP 的最大优势就是引入了多表达式的概念，使得每个个体可以拥有多条基因表达式，能够更好地适应不同问题的复杂性和特点[12] 。

3.3.1 多表达式原理

根据前面构造的多基因表达式算法，我们将每个染色体表示为三个个线性的基因串，MEP 中的基因串由一系列的基因组成，其中每个基因片段可以是一个函数（有一个或多个参数）或者是一个终结符（没有参数）。为了从基因串中生成表达式，我们需要根据基因的类型（函数或终结符）来决定如何处理。如果一个基因是函数，我们需要获取其参数，这些参数可能在它后面的基因中。如果一个基因是终结符，我们可以直接将它视为一个表达式。

以下是一个具体的过程：

1) 从基因串的开始处，取出第一个基因。
2) 检查这个基因的类型。如果它是函数，根据函数的参数个数，从它后面的基因中取出对应数量的基因作为参数，然后将这个函数和参数组合成一个表达式。如果它是终结符，直接将它视为一个表达式。
3) 保存这个表达式。

4） 将当前位置移动到下一个未处理的基因，并回到步骤 2)。

当遍历完整个基因串后，就得到了所有的表达式。

3.4 多表达式编程算法的改进

传统的 MEP 中，变异等操作是完全随机的，没有考虑个体的适应度。这可能导致高适应度个体的关键特征被破坏，而低适应度个体没有足够的机会进行改进。因此，优化遗传算子操作的策略可以提高算法的性能。为在预测性能和效率上有更多的提升，继续引入两种优化和改进方法：

1) 自适应突变概率策略：根据个体的适应度动态地调整突变概率。适应度高的个体将具有较低的突变概率，以保留其优秀特征，而适应度较低的个体将具有较高的突变概率，以增加多样性和探索性。这种自适应性可以提高算法的收敛性和搜索能力。

在进行自适应突变概率策略改进时，首先对突变率设置一个范围为$[Rate_{\min}, Rate_{\max}]$,对单个染色体来说，其适应度为 f ，则该染色体的突变概率为

$$mutation\_rate = Rate_{\min} + (1-a)\times(Rate_{\max} - Rate_{\min}) \tag{4}$$

其中 a 的值为：

$$a = \frac{f_{chro} - \min(fitness)}{\max(fitness) - \min(fitness)} \tag{5}$$

$\max(fitness)$、$\min(fitness)$ 分别代表当前种群中的最大、最小适应度，f_{chro} 为当前染色体的适应度值。由式（3）可知突变概率和当前个体适应度存在线性负相关，对于适应度较差的个体将被赋予更高的突变率。

2）自适应重组概率策略：将根据种群所有个体的适应度的标准差自适应调整单点重组和两点重组概率，以标准差 d 来评估种群多样性，对适应度标准差设定区间范围$[T_{upper}, T_{lower}]$,则重组概率为：

$$P = \begin{cases} P_c \times 0.9 & d > T_{upper} \\ P_c & T_{lower} < d < T_{upper} \\ P_c \times 1.1 & d < T_{lower} \end{cases} \tag{6}$$

P_c 为初始给定重组概率，根据公式（6），当多样性较大或较小时，即能实行对重组概率的自适应调整。

3.5 改进后的算法描述

MEP 算法流程图如图 9 所示：

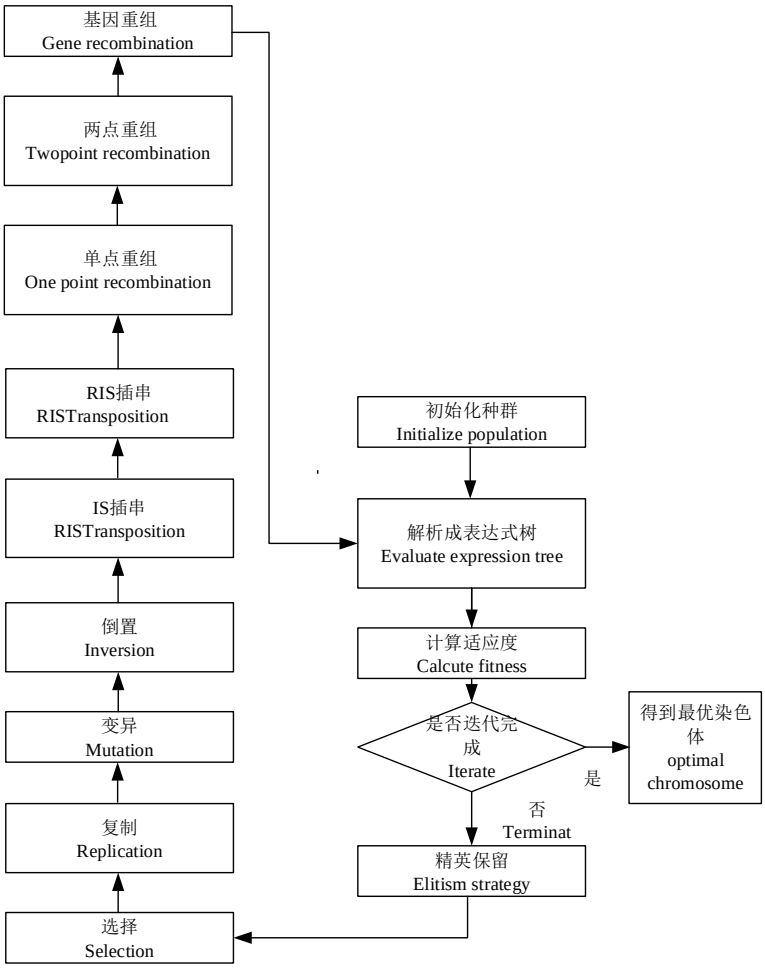

图 9 MEP 算法流程图

Fig. 9 MEP algorithm flowchart

步骤 1：按照设定好的种群数随机初始化染色体创建初始种群，将染色体基因设置为 3，对应每一种染料浓度。设置种群大小和迭代次数。

步骤 2：将每个染色体中的每个基因分别解码为表达式树（ET），并使用特定的适应度函数评估其适应度。这里我选用的适应度函数设计为均方差法（MSE），设第 i 个个体的均方差为：

$$E_i = \sum_{j=1}^{n} (P_{i,j} - T_j)^2 \tag{7}$$

其中 $P_{i,j}$ 为变量 y 关于求得函数 f 的预测值，T_j 是变量 y 的真实值，那么第 i 个个体的适应度函数 f_i 为：

$$f_i = 1000 \times \frac{1}{1+E_i} \tag{8}$$

由均方差的含义可知，f_i 的范围是 $(0,1000)$。

步骤 3：遍历每个染色体对染色体中的三个基因分别求适应度并取平均值，并找到适应度最高的染

色体，然后对种群进行遗传操作，主要有：

1）选择：本研究中选取轮盘赌法，它根据每个个体（染色体）的适应度值分配给它们在下一代中的复制份额。适应度更高的染色体有更大的机会被选中并复制到下一代。设第 i 个个体的适应度为 f_i，在轮盘赌方法下染色体被保留的概率为

$$p = \frac{f_i}{\sum f_i} \tag{9}$$

2）突变：将修改之后的自适应变异作为突变方式。

3）倒置：倒置操作也需要在基因级别执行，在染色体的头部（Head）区域和尾部区域(tail)分别随机选择两个位置，然后将这两个位置之间的基因序列进行倒置，即反转这部分基因序列。

4）插串

5）**重组**：根据改进后的自适应重组概率实现单点重组和两点重组

6）**基因重组**：基因重组算子是专门针对多基因染色体进行的操作，对于多基因染色体，随机选择一个基因，然后互换两个染色体中处于相同位置上的基因。

步骤 4：为了保证进化结束的种群性能要好于或至少等于前代，选择精英保留策略，将上一代中适应度最优个体替换掉当代适应度最差个体，

如此循环，直到迭代结束，输出最优染色体。

3.5 基于并行的多表达式编程（MEP）方法

在处理大规模遗传编程问题时，并行计算可以大大提高效率和可扩展性。在本文中，我们介绍了一种基于并行的多表达式编程（MEP）方法，该方法可以有效地利用多处理器进行遗传编程计算。

并行 MEP 首先通过将总体划分为几个子种群来利用多个处理器。这些子种群被分配到不同的处理器上，然后并行地进行遗传操作和适应度评估。这种方法的关键优点是，可以大大减少单个处理器需要处理的数据量，从而提高计算速度。

我们使用 Python 的多进程库来实现并行处理，这允许我们在每个子种群上并行执行计算密集型的遗传操作。子种群的适应度评估也并行进行，这进一步加快了计算速度。

3.6 模型参数设置

首先对 GEP 的参数和 MEP 的参数进行设置，因为最终选择的输入为 20 维，输出为 3 维，考虑到输入输入数据较多，但是头部基因长度选择过大会严重降低 GEP 运行效率，设置过短无法充分挖掘函数关系；对于改进后的 MEP，因为引入多表达式编程概念，可以充分挖掘每个基因携带的表达式树，但也带来大量计算，所以相比于 GEP，在进行大量验证的基础上可以减小种群规模来提高算法的效率；在设置自适应重组适应度标准差区间时，多次运行程序，求出种群的平均标准差，将区间设置为标准差平均值的 75%-125%，设置的 GEP 和 MEP 参数如下表 3：

表 3 GEP 和 MEP 参数设置
Tab. 3 GEP and MEP Parameter Settings

参数 Parameters	GEP	多进程 MEP/MEP
种群规模 Population size	500	100
单染色体基因数目 Number of genes in a single chromosome	3	3
头部基因长度 Length of the head gene	20	20
迭代次数 Number of iterations	100	100
函数集 Function set	$+$、$-$、$/$、$\times$、sin、cos、ln、exp、x^2、$sqrt$、$\frac{1}{x}$	
变异概率 Mutation probability	0.3	自适应
倒置概率 Inversion probability	0.1	0.1
IS 插串概率 IS transposition probability	0.2	0.2
RIS 插串概率 RIS transposition probability	0.2	0.2
单点重组概率 Single-point recombination	0.3	自适应
两点重组概率 Two-point recombination probability	0.3	自适应
自适应变异概率区间 Adaptive mutation probability interval		[0.2,0.6]
自适应重组初始概率 Initial probability for adaptive recombination		0.3
自适应重组适应度标准差区间 Standard deviation range for adaptive recombination fitness		[1.35，2.25]

将数据划分为训练集和测试集，测试集随机选取 10 组配方进行验证。

4．模型性能及结果分析.

在 python 环境中分别用 GEP 和 MEP 按照设置的程序对数据进行函数挖掘，在我们的实验中，GEP 和 MEP 都以适应度函数的高低作为选择和进化染色体的主要依据。我们观察到，在相同的迭代次数，且 MEP 种群规模远小于 GEP 的情况下，MEP 方法比 GEP 方法得到的染色体具有更高的适应度和更小的均方误差，这意味着 MEP 方法产生的模型能够更好地拟合训练数据。

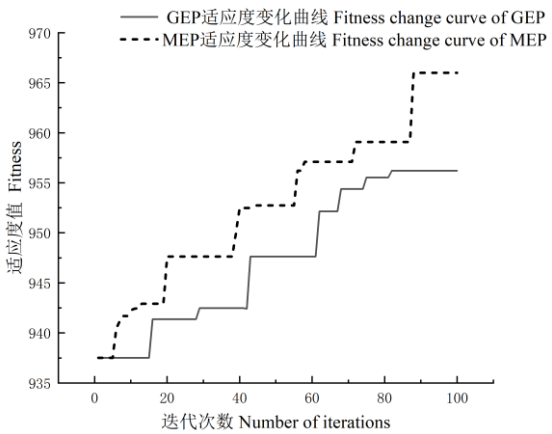

图 10 适应度寻优曲线

Fig. 10 Fitness optimization curve

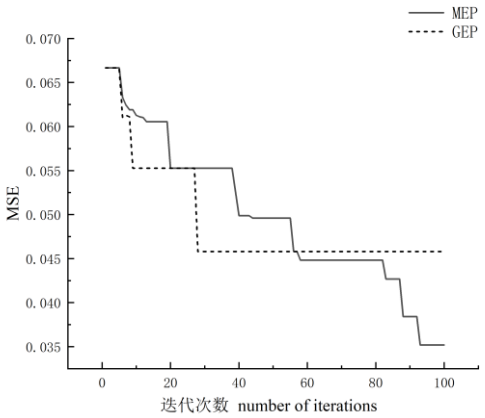

图 11 均方误差寻优曲线

Fig. 11 Mean Squared Error optimization curve

MEP 模型需要大量的计算资源来遍历可能的函数和操作符组合，并且必须对每一个可能的解都进行评估，这在大型或复杂的问题上会非常耗时。

对此我们分别使用单进程和多进程的方式实施 MEP，采用 24 核处理器运行多线程 MEP 模型，得到的适应度寻优曲线如图 12 和运行时间对比图 13，得到的结果揭示了多进程方法在处理大规模问题时具有显著的优势。我们观察到在适应度函数性能指标上，多进程的 MEP 与单进程的 MEP 表现相近，说明并行处理并没有牺牲模型的准确性。与单进程相比，多进程的 MEP 在计算时间上大大减少，表明了并行处理能够显著提高算法的运行效率。

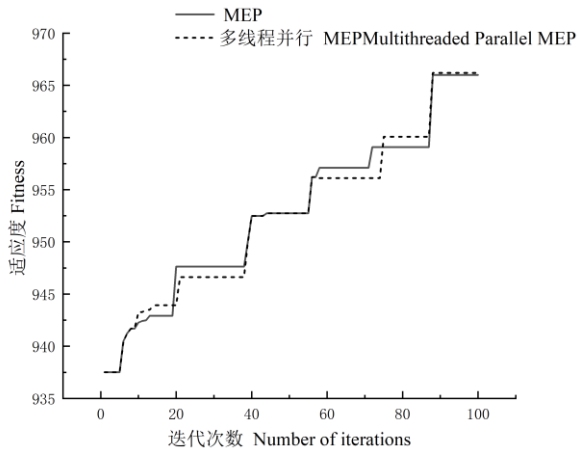

图 12 MEP/多线程并行 MEP 适应度寻优曲线

Fig. 12 Fitness optimization curve of MEP/Multithreaded Parallel MEP

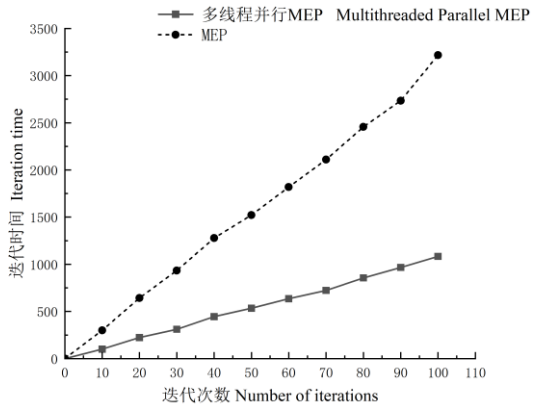

图 13 MEP/多线程并行 MEP 迭代时间曲线

Fig. 13 Iteration time curve of MEP/Multithreaded Parallel MEP

GEP 和 MEP 寻优结果对比：

表 4 GEP 和 MEP 寻优最佳染色体

Tab. 4 Optimal Chromosome Results for GEP and MEP Optimization

模型 Model	最优染色体 Best chromosome
G E P	+-+(sqrt)++K**(sin)(inv)(sqrt)(sin)(ln)(inv)*(inv)+(sin)/IARSQDSBPNHGOGKNTETJQ10115219814215121941261410311619 3
	(inv)(exp)+/(sin)O(ln)(ln)B(exp)E(inv)*/(inv)(sin)+RTOOTROGTDCFFCQAEK?N?T11206162011375031861613116131114
	(sqrt)+KC+/(X2)(sin)//(inv)*(sqrt)K(exp)(ln)/(inv)(inv)(inv)BTCDMJBJMCPQPBKAEKBMF11811172162320086115134181481 7
M E P	(X2)+G--S+(sqrt)(sqrt)(inv)(inv)(ln)(ln)(inv)(sqrt)(X2)+G--SHDHTDDARLJKSAILDEFID12131413191614114709831622010161119
	/(sqrt)(inv)(ln)(sin)(inv)D(X2)+-(inv)(ln)(ln)TK(exp)(sin)++(sqrt)HLOLGHFFA?GJJPGO?RBRO3161218814408202374176128791 6
	(X2)(sin)-(sqrt)C(sin)*(X2)(X2)-+(X2)-(inv)-(X2)+(ln)(sqrt)OKBMD?H?EHNIC?LPDCF?QO3101511201981171623143420181161 7

将染色体解码得到三种染料浓度的函数关系式：

GEP：

$$y_1=\sqrt{\sin(\frac{P}{N})\times\ln(I)}-(\frac{1}{A})\times R\times S+\sin(\frac{1}{Q})+\frac{1}{D+S}+\sqrt{\sin(B)}+K$$

$$y_2=\frac{1}{e^{\frac{O}{\ln(B)}+\sin(\ln(e^E))}}$$

$$y_3=\sqrt{K+C}$$

MEP：

$$y_1=(\frac{1}{\ln((H-D)^2)}+\sqrt{\ln(G)}-\sqrt{H-T+S}-S+G)^2$$

$$y_2=\sin(D)\times\sqrt{\ln(\frac{1}{(\frac{1}{\ln(e^{\sin(\sqrt{o}+H+L)})}+\frac{1}{T}-\ln(K))^2})}$$

$$y_3=\sin(\sqrt{\sin((B^4-M-D+\ln(33.38))^2\times(\sqrt{\frac{1}{H}}+O-K)^2-C))^2}$$

将由 GEP 和 MEP 得到的表达式在测试集上验证，采用的评价指标为配方相对偏差，配方平均相对偏差公式为：$r=\sum_{i=1}^{3}\frac{|y_i-p_i|}{y_i}$，$y_i(i=1,2,3)$ 为测试样本真实配比，$p_i(i=1,2,3)$ 为测试样本预测染料配比。得到预测值如表 5：

表 5 GEP 和 MEP 的预测结果

Tab. 5 Predicted Results for GEP and MEP

样本编号 Sample ID	真实配方 Actual Formulation			GEP 预测配方 GEP Predicted Formulation				MEP 预测配方 MEP Predicted Formulation			
	活性红 Red Dye	活性黄 Yellow Dye	活性蓝 Blue Dye	活性红 Red Dye	活性黄 Yellow Dye	活性蓝 Blue Dye	配方相对偏差 Relative Deviation	活性红 Red Dye	活性黄 Yellow Dye	活性蓝 Blue Dye	配方相对偏差 Relative Deviation
1	0.14	0.38	0.05	0.154	0.430	0.044	0.117	0.163	0.460	0.086	0.365
2	0.27	0.23	0.1	0.245	0.358	0.095	0.233	0.256	0.234	0.179	0.286
3	0.25	0.65	0.05	0.238	0.820	0.045	0.137	0.289	0.765	0.064	0.204
4	0.19	0.52	0.03	0.153	0.670	0.039	0.261	0.158	0.340	0.036	0.238
5	0.04	0.21	0.04	0.056	0.271	0.041	0.238	0.046	0.231	0.029	0.172
6	0.13	0.11	0.02	0.201	0.180	0.014	0.494	0.174	0.140	0.018	0.237
7	0.3	0.6	0.1	0.380	0.602	0.145	0.240	0.534	0.768	0.111	0.390
8	0.14	0.38	0.03	0.167	0.343	0.035	0.152	0.157	0.356	0.052	0.306
9	0.23	0.1	0.12	0.410	0.140	0.134	0.433	0.330	0.056	0.144	0.358
10	0.03	0.06	0.01	0.029	0.140	0.009	0.489	0.043	0.066	0.022	0.578

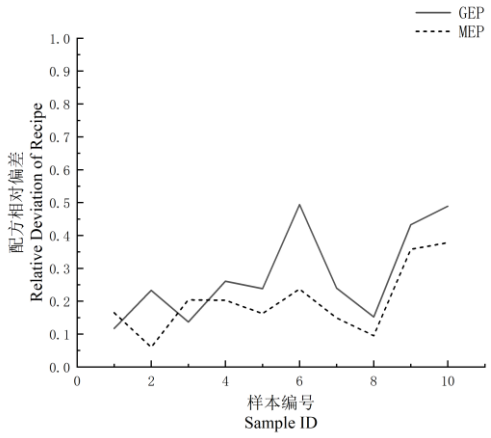

图 14 GEP、MEP 模型预测配方相对偏差

Fig. 14 Relative Deviation of Recipe Prediction by GEP and MEP Models.

表 6 GEP 模型和 MEP 模型平均配方相对偏差对比

Tab. 6 Comparison of Mean Relative Deviation for GEP and MEP Models

模型 Model	GEP	MEP
配方相对偏差均值	0.489	0.333

Mean Relative Deviation of Formulation

根据 GEP 模型和 MEP 模型的配方预测结果，绘制配方相对偏差曲线图，如图 14 所示。由曲线可以看出，与 GEP 预测结果相比，MEP 预测效果更好，对大部分测试样本得到的配方相对偏差更小，表 6 为测试样本的平均相差对偏差，即 MEP 模型预测结果与实际配方更为接近。这表明 MEP 算法在处理复杂的配方预测问题上具有优势。相对于 GEP 模型，MEP 模型能够更准确地挖掘出配方中各个成分之间的复杂交互关系，从而提高了预测的精度和准确性。

为进一步验证 MEP 在染色配方预测中的优势，选择选择传统的遗传算法和神经网络模型包括遗传编程（GP）、反向传播神经网络（BP 神经网络）、径向基网络（RBF 神经网络）对相同数据建立模型并进行预测，其中在 SVR 中根据人工经验寻优设置三个参数惩罚参数 C、损失函数 epsilon、核系数 gamma 设置的范围为[0,100]；BP 神经网络中设置迭代次数为 1000 次，加入两个全连接层；RBF 神经网络基函数设置为 RBF 函数，L2 范数设为 2，宽度参数设置为 0.5；随机森林模型中设置决策树数量为 100，决策树的最大深度为 15。得到的预测结果如表 7、表 8：

表 7 随机森林和 SVR 模型的预测结果

Tab. 7 Predicted Results for GP and SVR Models

样本编号 Sample ID	随机森林 Random Forest			配方相对偏差 Relative Deviation	SVR			配方相对偏差 Relative Deviation
	活性红 Reactive Red Dye	活性黄 Reactive Yellow Dye	活性蓝 Reactive Blue Dye		活性红 Reactive Red Dye	活性黄 Reactive Yellow Dye	活性蓝 Reactive Blue Dye	
1	0.189	0.232	0.054	0.274	0.197	0.110	0.069	0.498
2	0.149	0.337	0.043	0.492	0.260	0.575	0.041	0.709
3	0.160	0.422	0.049	0.242	0.143	0.414	0.021	0.455
4	0.176	0.248	0.044	0.348	0.170	0.187	0.038	0.340
5	0.221	0.231	0.065	1.748	0.058	0.267	0.061	0.413
6	0.218	0.297	0.054	1.365	0.209	0.176	0.038	0.710
7	0.174	0.431	0.077	0.311	0.193	0.489	0.104	0.193
8	0.147	0.386	0.044	0.176	0.129	0.403	0.037	0.122
9	0.207	0.178	0.074	0.421	0.270	0.134	0.132	0.204
10	0.054	0.144	0.010	0.743	0.054	0.163	0.017	1.059

表 8 BP 神经网络和 RBF 神经网络模型的预测结果

Tab. 8 Predicted Results for BP Neural Network and RBF Neural Network Models

样本编号 Sample ID	BP 神经网络			配方相对偏差 Relative Deviation	RBF 神经网络			配方相对偏差 Relative Deviation
	活性红 Reactive Red Dye	活性黄 Reactive Yellow Dye	活性蓝 Reactive Blue Dye		活性红 Reactive Red Dye	活性黄 Reactive Yellow Dye	活性蓝 Reactive Blue Dye	
1	0.173	0.540	0.063	0.306	0.219	0.174	0.043	0.412
2	0.302	0.345	0.200	0.540	0.272	0.394	0.039	0.444
3	0.432	0.812	0.045	0.359	0.160	0.633	0.038	0.211
4	0.189	0.552	0.036	0.089	0.202	0.196	0.034	0.270
5	0.002	0.331	0.034	0.558	0.051	0.525	0.033	0.653

6	0.156	0.087	0.015	0.220	0.153	0.170	0.016	0.314
7	0.290	0.582	0.234	0.468	0.375	0.362	0.081	0.280
8	0.198	0.402	0.026	0.202	0.180	0.396	0.044	0.267
9	0.243	0.238	0.156	0.579	0.239	0.110	0.179	0.211
10	0.050	0.078	0.016	0.522	0.054	0.073	0.007	0.436

表 9 各模型配方平均相对偏差对比

Tab. 9 Comparison of Average Relative Deviation for Various Models in Formulation

模型 Model	MEP	随机森林 Random Forest	BP 神经网络 BP Network	RBF 神经网络 RBF Network	支持向量回归 SVR
配方相对偏差 Relative Deviation	0.313	0.613	0.384	0.350	0.470

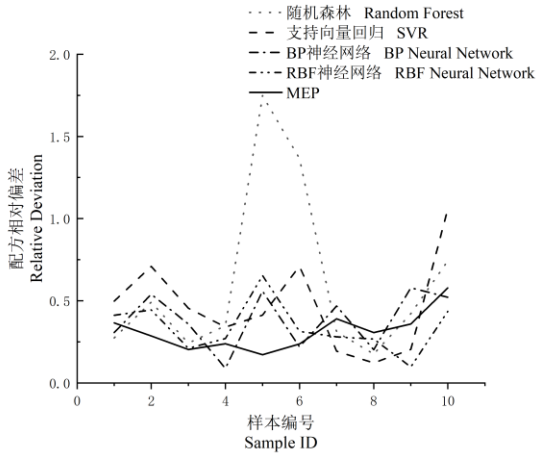

图 15 各模型在测试样本预测值曲线

Fig. 15 Prediction Curves of Various Models

各模型在测试样本上的平均配方相对偏差如表 9，各模型在测试样本预测值曲线如图 15，通过对表 9 中各模型在测试样本上的平均配方相对偏差进行比较，可以明显看出多表达式编程（MEP）在木材染色配方中具有较高的精度。相对于其他模型，MEP 模型能够更准确地预测染料配比，同时，配方相对偏差稳定在一个较小的范围内，说明预测效果比较稳定，表明 MEP 模型对于不同的木材样本都能够保持较高的预测准确性，并且不会因为样本的变化而引入较大的偏差。

5．讨论

5.1 多表达式编程（MEP）在木材染色配方中的优势

MEP 允许使用多个表达式来表示染色配方，每个表达式可以描述不同的染色参数和配方变量。生成的表达式通常具有良好的可解释性，可以清晰地展示染色参数和变量之间的关系。这使得染色工艺的调整和优化变得更加容易和直观。

5.2 多表达式编程（MEP）存在的问题

MEP 涉及使用多个表达式来表示染色配方，这增加了算法的复杂性。选择合适的表达式、确定适当的染色参数和变量之间的关系，并进行优化，需要专业知识和经验。

在参数选择和优化中 MEP 需要选择和优化多个表达式，以找到最佳的染色配方。这涉及到选择适当的染色体参数、确定合适的优化目标和约束条件，并进行搜索和调整。这个过程可能会比较耗时和困难。

MEP 的成功依赖于准确的输入数据和模型。需要准确收集和测量染色参数、变量和目标，以获得可靠的模型。此外，模型的误差和不确定性可能会影响优化结果的准确性和可靠性，对于得到的函数模型会存

在泛化性不强的缺点。

6.结论

多表达式编程在对木材染色浓度函数挖掘中，多表达式编程在对木材染色浓度函数的挖掘中显示了其出色的性能。它有效地整合了多个简单表达式，形成了一个复杂的模型，这个模型能够准确地描绘了染色浓度与各种影响因素之间的关系。通过使用这种方法，我们可以更精准地预测不同情况下的染色浓度，从而在实际应用中优化染色过程，提高生产效率和产品质量。

参考文献

[1] Chen M, Tsang H S, Tsang K T, et al. An Hybrid Model CMR-Color of Automatic Color Matching Prediction for Textiles Dyeing and Printing Neural Computing for Advanced Applications: Second International Conference, NCAA 2021, Guangzhou, China, August 27-30, 2021, Proceedings 2. Springer Singapore, 2021: 603-618.

[2] Furferi R, Governi L, Volpe Y. Color matching of fabric blends: Hybrid Kubelka-Munk+ artificial neural network based method[J]. Journal of Electronic Imaging, 2016, 25(6): 061402-061402.

[3] Wang Q, Shuai L, Dai X, et al. PSO-BP neural network-based gelatin hollow capsule color matching model[C]//2020 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2020: 94-99.

[4] 李文峰. 基于极限学习机的杉木单板染色智能配色模型研究[D].东北林业大学,2022.

[5] Li H, Lai L, Chen L, et al. The prediction in computer color matching of dentistry based on GA+ BP neural network[J]. Computational and mathematical methods in medicine, 2015, 2015.

[6] 张维,张浩晨.一种基于最优集成随机森林的小样本数据特征提取方法[J].西北工业大学学报,2022,40(06):1261-1268.张志伟,尹文亭,李彦鹏.基于基因表达式编程的煤矿地表变形预测模型的研究与实现[J].河南科技,2023,42(01):34-39.

[7] 石康乐,何朗,朱四荣,彭斯俊,黄樟灿,谭华.基于基因表达式编程的 FRP 蠕变分析与预测[J].武汉理工大学学报,2022,44(07):1-9.

[8] 贾晓莉. 基因表达式编程的改进及在雾霾浓度预测中的应用研究[D].西安建筑科技大学,2021.

[9] 湛航,何朗,黄樟灿,李华峰,张蔷,谈庆.改进的基于层次距离的基因表达式编程特征选择分类算法[J].计算机应用,2021,41(09):2658-2667.

[10] 朱明放,任艳玲,张建斌.多表达式编程的函数发现问题研究[J].计算机工程与设计,2011,32(03):1134-1137.

[11] 代术成,唐常杰,朱明放,陈瑜,乔少杰,向勇,李太勇.基于多表达式基因编程的复杂函数挖掘算法[J].四川大学学报(工程科学版),2008(06):121-126.

[12] 邓薇. GEP 和 MEP 的新型解码评估技术及融合[D].长沙理工大学,2013.