

基于改进粒子群算法的木材板材下料方法

黄秀玲, 陶泽, 尤华政, 李宸, 刘俊

(南京林业大学机械电子工程学院, 南京 210037)

摘要:木材板材在家具行业应用广泛,以绿色环保、节约能源为目的的木材板材优化下料已经成为研究的热点。木材板材下料优化问题属于二维矩形下料问题,是一种具有高度计算复杂性的问题。本研究主要针对单规格木材板材进行矩形零件下料问题,在木材板材长和宽都大于零件长和宽的情况下,通过建立二维下料的数学模型,采用标准粒子群算法、变邻域搜索算法、粒子群混合变邻域搜索算法分别进行求解,并以某企业的下料实例进行分析计算。首先,利用标准粒子群算法求解单规格板材下料问题;其次,利用变邻域搜索算法求解单规格板材下料问题。在获得局部最优解的基础上改变其邻域结构再进行局部搜索,找到另一个局部最优解,如此不断迭代,直到满足算法的终止条件,获得全局最优解;最后,利用粒子群变邻域搜索混合算法求解单规格板材下料问题。针对粒子群算法局部搜索能力较差、容易过早收敛的问题和具有较好包容性的特点,将变邻域搜索的思想融入粒子群算法中,使结果更加趋向全局最优。结果表明:粒子群变邻域搜索混合算法相比粒子群算法和变邻域算法效率都有显著提升,能显著提高该木材板材的利用率,增加企业经济效益。

关键词:木材板材;二维矩形下料问题;粒子群算法;变邻域搜索算法;粒子群混合变邻域搜索算法

中图分类号:TH164

文献标志码:A

文章编号:2096-1359(2024)01-0125-07

Research on cutting method of wood panels based on improved particle swarm algorithm

HUANG Xiuling, TAO Ze, YOU Huazheng, LI Chen, LIU Jun

(College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China)

Abstract: Wood-based panels is widely used in the furniture industry and the research on optimized cutting patterns for saving materials and energy has become a scientific research hotspot. The cutting patterns of wood panels have received increasing attention. At present, many furniture companies still use manual cutting pattern to produce furniture panel parts, which often takes a long time and causes serious waste of materials. Due to the diversity of customer needs, it is necessary to arrange a reasonable cutting plan to make full use of the wood-based panels, improving the economic efficiency of the enterprise and save natural resources. The optimization of the panel sawing pattern is a two-dimensional rectangular blanking problem with high computational complexity. This study mainly focused on the blanking of rectangular parts for single-size wood-based panels. In the case that the length and width of panels were greater than the length and width of parts, the standard particle swarm optimization (PSO) algorithm, variable neighborhood search algorithm and particle swarm optimization (PSO) hybrid variable neighborhood search algorithm were used to solve the problem by establishing a two-dimensional mathematical model of blanking. Cutting examples were analyzed and calculated based on the data collected from several furniture enterprises. Firstly, the standard particle swarm algorithm was used to solve the single-size panel blanking problem. Secondly, the variable neighborhood search algorithm was used to solve the single-size panel blanking problem. To obtain the local optimal solution, the neighborhood structure was changed, and then local search was performed to find new local optimal solution. This process was repeated until the termination condition of the algorithm was met. Finally, the particle swarm variable neighborhood search hybrid algorithm was used to solve the single-size plank blanking problem. Because the particle swarm algorithm has the drawbacks of poor localization ability, premature convergence and large tolerance, the idea of variable neighborhood search was integrated into the particle swarm algorithm to make the result closer to the global optimum. The results showed that, compared with the particle swarm optimization (PSO) and variable neighborhood search algorithm, the hybrid algorithm of the particle swarm optimization (PSO) significantly improved the utilization rate of wood-based

panels for the furniture enterprises, enhancing the economic benefits for these enterprises.

Keywords: wood panels; two-dimensional rectangular blanking problem; particle swarm algorithm; variable neighborhood search algorithm; particle swarm hybrid variable neighborhood search algorithm

在家具行业中,以绿色环保、节约能源为目的的优化木材板材下料问题已经成为研究热点,木材板材下料问题受重视程度日益提高。目前有很多企业生产家具产品时,对于木材板材仍使用人工下料,经常耗时较长且材料浪费严重。由于客户需求的多样性,需要合理安排下料方案,充分利用木材板材,提高企业经济效益和节约社会资源。本研究的木材板材下料优化问题属于二维矩形下料问题,是一种具有高度计算复杂性的问题^[1-4]。

20世纪60年代,Gilmore等^[5-6]提出了经典的“一刀切(guillotine cutting)”和“正交切割(non-guillotine cutting)”两种二维切割方式,通过线性规划和动态规划,解决了“一刀切(guillotine cutting)”问题范围内的下料算法,首次为解决下料问题提供了切实可行的方案和技术手段。但是由于当时计算辅助技术并不是很突出,所以仅停留在理论层面。

20世纪90年代后期,随着计算机技术的高速发展,计算复杂性理论不断成熟,大量的智能优化算法不断涌现,并在各行业中得到广泛应用^[7-8],如遗传算法(genetic algorithm)、模拟退火算法(simulated annealing)、灰狼优化(grey wolf optimization)、蜻蜓算法(dragonfly algorithm)、蚁群算法(ant colony algorithm)以及粒子群算法(particle swarm optimization algorithm)等。这些算法一般都是建立在生物智能或物理现象基础上的随机搜索算法,可以解决大规模的复杂下料问题,且通用性较强^[9-11]。笔者主要针对单规格木材板材下料问题,在木材板材长和宽都大于零件长和宽的情况下,采用粒子群算法、变邻域搜索算法、粒子群变邻域搜索混合算法求解单规格木材板材下料问题,并进行分析对比。

1 数学模型的建立

本研究所采用的原材料板材为 $L \times W$ 的矩形木材板材,待切割的毛坯零件一共有 K 种。在充分考虑多方面因素和约束的前提下,选择以剩余废料最少为优化目的,得出数学模型如下^[12]:

$$0 \leq X_k \leq L, (1 \leq k \leq C_j, 1 \leq j \leq N) \quad (1)$$

$$0 \leq Y_k \leq W, (1 \leq k \leq C_j, 1 \leq j \leq N) \quad (2)$$

$$X_k + [l_k P_k + w_k(1 - P_k)] \leq L, \quad (1 \leq k \leq C_j, 1 \leq j \leq N) \quad (3)$$

$$Y_k + [w_k P_k + l_k(1 - P_k)] \leq W, \quad (1 \leq k \leq C_j, 1 \leq j \leq N) \quad (4)$$

$$R_k \cap R_{k_j} = \emptyset, (1 \leq k \neq k_j \leq C_j, 1 \leq j \leq N) \quad (5)$$

$$\sum_{i=1}^k A_{ij} = C_j, (1 \leq j \leq N) \quad (6)$$

$$\sum_{j=1}^N A_{ij} T_j = n_i, (1 \leq i \leq K) \quad (7)$$

$$T_j \geq 1, (1 \leq j \leq N) \quad (8)$$

$$F = \min \left(LW \sum_{j=1}^N T_j - \sum_{j=1}^N \sum_{i=1}^K l_i w_i A_{ij} T_j \right) \quad (9)$$

式中: L 和 W 分别为原材料板材的长度和宽度; l 和 w 分别为零件板材的长度和宽度; i 和 K 分别为零件类型编号和集合, $i \in K$; N 为原材料板材的数量; C_j 为 j 张单板上排放零件数量; (X_k, Y_k) 为每个矩形零件对应左下角的坐标, $k \in C_j$; P_k 为矩形零件在原材料中的排列方式; R_k 为零件所在矩形的命名; n_i 为第 i 类零件的数量; A_{ij} 为在第 j 张单板方案中第 i 种零件的数量,且 $A_{ij} \leq n_i$; T_j 为第 j 种方式的切割次数, $j \in N, T_j \geq 1$; F 为目标函数。

式(1)确保排放在原材料板材上的所有毛坯零件 X 轴上的长度不会超出原材料板材的长度;式(2)确保排放在原材料板材上的所有毛坯零件 Y 轴上的长度不会超出原材料板材的长度;式(3)和式(4)为了保证零件不会超出原材料板材的边界,限制了长度和宽度;式(5)保证毛坯零件不会重叠排放在原材料板材上;式(6)和式(7)使所有毛坯零件的数量和配套要求得到满足;式(9)是目标函数,使得浪费的原材料板材的面积最小,也就是保证原材料板材的利用率能够达到最高。

2 算法求解

2.1 粒子群算法

粒子群算法也称作粒子群优化算法(particle swarm optimization, PSO),是1995年美国Kennedy等^[13]通过对鸟类群体行为进行模仿提出的一种进化算法(evolutionary algorithm, EA)。粒子群算法通常用于解决连续优化问题,将其应用于排样问题中的排列组合问题,需要对粒子群算法进行一定的调整^[14-15]。

2.1.1 编码

排列组合问题涉及对元素进行排列或组合,因此需要定义适合该问题的编码方式。常用的离散

式编码方法主要包括二进制编码、排列编码或 One-Hot 编码等方式。针对排样问题的顺序优化上选用基于位置的编码方式,即将每个可行解看作一个由各个元素位置决定的序列。排样过程中每个可行解可以表示为一个由 n 个元素位置决定的序列,其中第 i 个元素表示该物品在排样中的位置,具体的零件排序方式为:

$$X = (x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) \quad (10)$$

式中: x_i 表示第 i 个零件在排样中的位置, $i=1, 2, \dots, n$ 。整个序列 X 表示优化过程中一个完整的零件排列^[16]。这样的编码方式可以保证每个粒子的取值范围是固定的、不重复的,且可以直接映射到可行解的空间。

2.1.2 粒子速度

在排列优化问题中,由于粒子编码格式的特殊性,速度与位置的运算方式需要重新定义。本研究采用基于位置编码方式的离散式粒子群算法速度的计算方式^[17]。

定义速度 V 为位置 X_2 和位置 X_1 的差值,是一个 N 维向量:

$$V = X_2 - X_1 \quad (11)$$

式中,速度 V 中的每个元素 v_i 表示为 X_2 和 X_1 在序列上对应的元素的对比。当 X_1 的元素与 X_2 相同时速度 V 上的元素记录为 0,当 X_1 的元素与 X_2 不相同速度 V 上的元素记录为 X_2 的元素,具体可表示为:

$$v_i = \begin{cases} 0, & \text{if } x_{1,i} = x_{2,i} \\ x_{2,i}, & \text{else} \end{cases} \quad (12)$$

排列优化问题中,位置与速度的加法运算可表示为:

$$X = X_1 + V_1 \quad (13)$$

式中,进行加法运算后新产生的向量的每一个分量以序列 i 的顺序按照以下方式计算:当 V_1 的元素为 0 时, X 的元素记录为 X_1 的元素;当 V_1 的元素不为 0 时, X 的元素记录为 V_1 的元素,并将 X_1 中对应数值 $x_{1,i}$ 与数值 $v_{1,i}$ 的序列位置交换(由于编码的方式为位置编码,对于任意序列 X_1 ,当 $x_{1,i}$ 与 $v_{1,i}$ 不同时, X_1 中都应包含非零元素 $v_{1,i}$)。具体可表示为:

$$x_i = \begin{cases} x_{1,i}, & \text{if } v_{1,i} = 0 \\ v_{1,i}, \text{swap}(x_{1,i}, v_{1,i}) & \text{else} \end{cases} \quad (14)$$

粒子的运动方程具体描述如下:

$$V = c_1(X_{\text{pbest}} - X) + c_2(X_{\text{gbest}} - X) \quad (15)$$

式中,速度的数乘方式具有概率意义,它可以增强计算的随机性。在计算过程中生成一个随机数,通

过数值的比较确定速度值。

$$v_{2,i} = \begin{cases} v_{1,i}, & \text{if rand} < c \\ 0, & \text{else} \end{cases} \quad (16)$$

2.1.3 目标函数

矩形排样问题的目标是在一个给定规格的矩形板材上,按照某种排列顺序摆放零件,从而最大化板材的利用率。在粒子群算法中,将编码数值作为排样顺序,将位置编码下产生的排样利用率作为目标。

具体的,粒子群算法下,排样策略的全局目标如式(17)所示:

$$f = \max \left(\frac{\sum_{i=1}^N (S_i)}{W \times H^*} \right) \quad (17)$$

式中: f 表示为下料零件总面积在所用板材面积中占据的最大比例; i 为零件序号; N 为零件总数; S_i 为第 i 个零件所占据面积; W 为板材宽度; H^* 为该排样方案下 N 个零件排放完毕后,零件排样图的最大使用高度。

2.1.4 粒子群算法流程

本研究粒子群算法的流程如图 1 所示,具体流程如下:

- 1) 粒子群状态进行重置。
- 2) 对粒子的适应度值 $\text{Fit}[i]$ 进行计算。
- 3) 个体极值 $p_{\text{best}}(i)$ 比较,当 $\text{Fit}[i] > p_{\text{best}}(i)$ 时, $\text{Fit}[i]$ 替换掉 $p_{\text{best}}(i)$ 。
- 4) 全局极值 $g_{\text{best}}(i)$ 比较,当 $\text{Fit}[i] > g_{\text{best}}(i)$

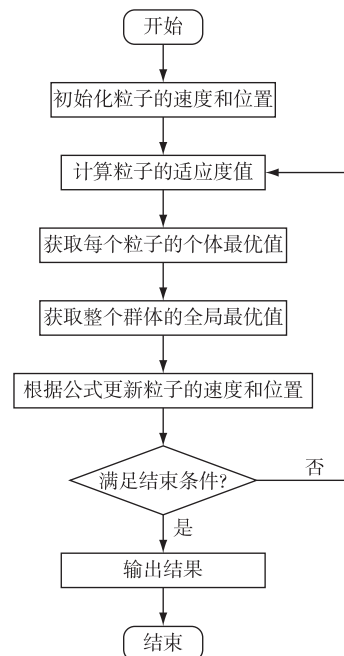


图 1 粒子群算法流程

Fig. 1 Flow chart of particle swarm optimization

时, $\text{Fit}[i]$ 替换掉 $g_{\text{best}}(i)$ 。

5) 根据式 (13) 和式 (15) 更新粒子的位置 X_i 和速度 V_i 。

6) 判断是否满足结束条件 (误差足够好或者达到最大循环次数), 若满足, 则算法结束并输出最优结果; 若不满足, 则返回 2)。

2.2 变邻域搜索算法

变邻域搜索算法 (variable neighborhood search, VNS) 是由 Hansen 和 Mladenovi 在 1997 年提出的一种新颖且高效的局部搜索算法^[18-19]。其基本思想是: 为了能够更好地获取局部最优解, 算法扩展了解的搜索范围, 并且在搜索的过程中系统化地改变其多个邻域结构^[20-21]。

本研究变邻域搜索算法的简要流程如图 2 所示, 算法流程如下:

1) 初始化。给定一个初始解 S , 并定义一系列邻域结构 $N_K, K=1, 2, \dots, K_{\max}$ 。

2) 令 $K=1$ 。

3) 把定义的初始解 S 作为当前解, 在当前邻域 N_K 内进行局部搜索获得该邻域的最优解 S_1 。

4) 若 S_1 优于 S , 就用 S_1 替换掉当前解 S , 跳回 2); 若不优于 S , 就令 $K=K+1$ 。

5) 若 $K > K_{\max}$, 就直接输出最优解, 结束; 若 $K < K_{\max}$, 就跳回 3)。

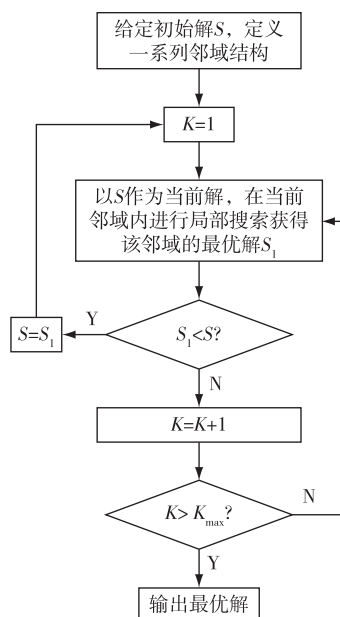


图2 VNS 算法流程

Fig. 2 Flow chart of VNS algorithm

变邻域搜索主要包括局部搜索 (local search) 和改变邻域 (neighborhood change)。其中, 局部搜索的作用是在同一个邻域结构中寻找更优的局部最优解, 改变邻域的作用是为整个搜索过程提供一

种迭代方法和停止准则。

本研究根据排样需要采用混合性邻域结构的变邻域搜索策略^[22]。其基本步骤如下:

1) 初始化。选择初始解 x , 邻域结构 $N(i), i=1, 2, \dots, k$, 当前最优解 $x_{\text{best}} = x$ 。

2) 如果满足算法终止条件, 那就输出 x_{best} ; 若不满足算法终止条件, 那就令 $i=1$ 。

3) 如果 $i=k+1$, 那么就跳转步骤 2); 否则, 在 x 的邻域结构 $N(i)$ 中随机选取可行解 x' , 在 x' 的邻域结构中搜索得到局部最优解 x'' , 若 x'' 比当前最优解 x_{best} 更优, 则 $x_{\text{best}} = x'', i=i+1$ 。循环这步操作。

2.3 粒子群混合变邻域搜索算法

通过分析粒子群与变邻域搜索算法结构及试验结果, 发现粒子群算法和变邻域搜索算法均有各自的优点和不足。

粒子群算法的局部搜索能力较弱, 搜索精度较低, 所以很容易造成过早收敛的情况, 但是变邻域搜索算法具有较好的局部搜索能力且搜索精度也较高。在运算过程中, 粒子群算法受历史最优解的影响较大, 经过不断的迭代都是在向最优解靠近, 缺少了解的多样性, 而变邻域搜索算法正好可以弥补这一缺点, 通过不断搜索不同邻域, 可以跳出局部最优的情况。

粒子群算法收敛速度较快, 全局搜索能力较好, 主要通过粒子不断向最优解靠近, 可以较快地在很大的解空间中找到具有最优解的大致范围。这正好可以弥补变邻域搜索算法收敛速度慢、搜索时间长的缺点, 可以给变邻域搜索的开始提供一个高质量的初始解。

为了克服粒子群算法可能产生早熟的现象, 考虑到粒子群算法中粒子每次更新都会受到当前解、个体历史最优解和种群历史最优解 3 个因素的影响, 本研究将粒子的更新过程采用变邻域搜索, 动态地改变邻域结构来拓展搜索的空间, 使算法最终能够跳出局部最优解, 搜索到全局最优解。

笔者设计的粒子群混合变邻域搜索算法的基本流程如下:

1) 对粒子群状态进行重置。

2) 对粒子的适应度值 $\text{Fit}[i]$ 进行计算。

3) 个体极值 $p_{\text{best}}(i)$ 比较, 当 $\text{Fit}[i] > p_{\text{best}}(i)$ 时, $\text{Fit}[i]$ 替换掉 $p_{\text{best}}(i)$ 。

4) 全局极值 $g_{\text{best}}(i)$ 比较, 当 $\text{Fit}[i] > g_{\text{best}}(i)$ 时, $\text{Fit}[i]$ 替换掉 $g_{\text{best}}(i)$ 。

5) 根据式 (14) 和式 (15) 更新粒子的位置 X_i 和速度 V_i 。

6)判断是否满足结束条件(误差足够好或者达到最大循环次数),若满足,则算法结束并输出最优结果;若不满足,则返回 2)。

7)对粒子进行调用变邻域搜索,主要的搜索策略是:顺序执行各个模块,一旦出现比当前更优秀的中间解,就用中间解代替当前解,继续调用这个模块进行搜索,直到搜索发现更差的中间解出现;否则执行下一模块直到结束。

8)判断是否满足结束条件 $K < K_{\max}$,如果满足终止条件,那么算法终止并输出最优结果;如果不满足终止条件,那么就转回 7)。

9)输出全局最优值,算法运行结束。

粒子群混合变邻域搜索算法流程图如图 3 所示。

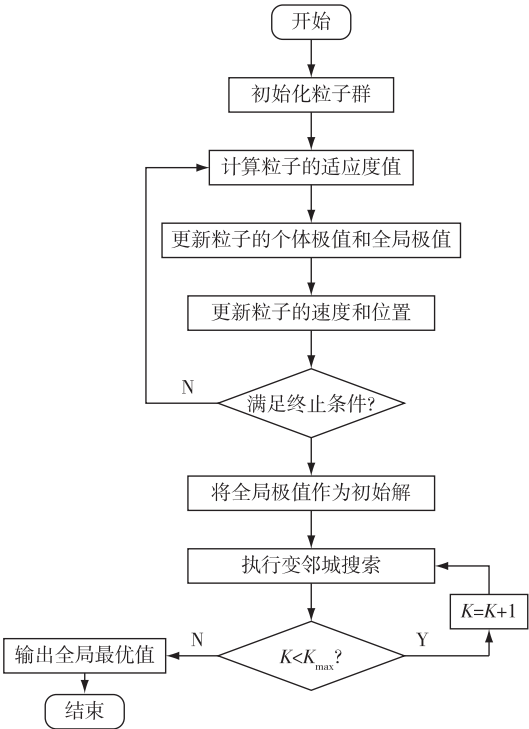


图 3 粒子群变邻域搜索混合算法流程
Fig. 3 Flow chart of particle swarm variable neighborhood search hybrid algorithm

本研究首先利用粒子群算法进行运算,求出的

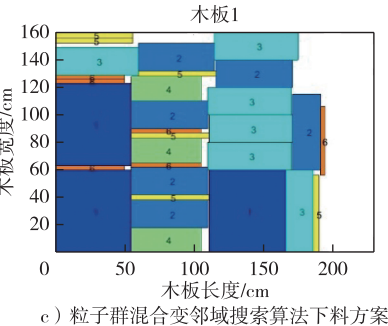
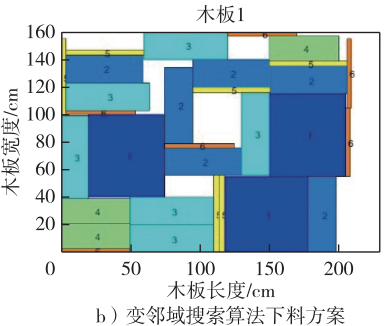
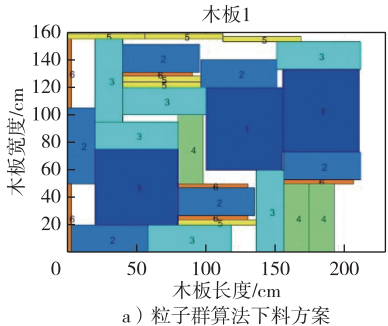


图 4 实例 1 的 3 种算法下料方案对比
Fig. 4 Comparison of cutting patterns of three algorithms in the example 1

最优解作为后续变邻域搜索算法的初始解,再用变邻域搜索算法继续进行求解。粒子群算法为后续的变邻域搜索提供了一个较好的初始解,并引导算法寻找更好的解;变邻域搜索是集中邻域搜索在搜索空间附近的较好解。从而提高了算法的全局搜索能力和局部搜索能力,平衡了算法在搜索过程中的普遍性和集中性。

3 实例分析

笔者根据某公司产品实际生产情况,选取 3 种不同需求的下料实例进行分析。

实例 1:根据某公司生产的木家具产品实际情况,选取下料板材长度为 230 cm、宽度为 160 cm 的单一规格的规则矩形板材,根据客户需求,需要做 3 套成品,一共有 6 种毛坯零件的需求。

毛坯零件的具体尺寸和数量如表 1 所示。

Table 1 Data of panel parts in the example 1			
序号	长度/cm	宽度/cm	数量
1	60	55	3
2	20	55	6
3	60	20	6
4	50	18	3
5	56	4	6
6	50	3	6

经过计算得到利用粒子群算法的利用率是 84.846%,利用变邻域搜索算法的利用率是 85.658%,粒子群混合变邻域搜索算法的利用率为 92.281%。3 种算法具体的下料方案如图 4 所示,下料都只消耗了 1 块板材,虽然下料的利用率可以接受,但是从下料的方案中可以看出,前两种下料方案仍有很大的改进空间。粒子群混合变邻域搜索算法相比粒子群算法和变邻域搜索算法的计算都有较大幅度的提升,且通过粒子群混合变邻域搜索算法生成的具体下料方案可以看到,并没有产生面积过大的废料,剩下的余料仍然可以投入后续生产使用。

实例 2:选取下料板材为单一规格的规则矩形木材板材,板材的长度为 140 cm,宽度为 90 cm,现在根据客户要求,需要切割出 9 种毛坯零件,通过变邻域搜索算法和粒子群算法计算出最优的下料方案。具体的毛坯零件的尺寸数量要求如表 2 所示。

表 2 实例 2 毛坯零件数据
Table 2 Panel parts data in the example 2

序号	长度/cm	宽度/cm	数量
1	24	14	1
2	38	22	5
3	50	32	2
4	60	20	5
5	42	36	2
6	48	14	2
7	26	18	1
8	42	24	2
9	30	16	3

实例 3:选取下料板材长度为 230 cm、宽度为 160 cm 的单一规格的规则矩形板材,根据客户需求,一共有 12 种毛坯零件的需求,具体的毛坯零件的尺寸和数量要求如表 3 所示。

表 3 实例 3 毛坯零件数据
Table 3 Panel parts data in the example 3

序号	长度/cm	宽度/cm	数量
1	100	15	2
2	54	30	1
3	46	40	2
4	42	42	1
5	110	15	1
6	36	16	1
7	60	24	1
8	88	26	2
9	50	40	3
10	58	18	2
11	116	20	1
12	34	28	1

对于实例 1、实例 2 和实例 3 同时用粒子群算法、变邻域搜索算法和粒子群混合变邻域搜索算法进行下料计算的结果对比如表 4 所示。

表 4 3 种算法结果对比
Table 4 Comparison of results of three algorithms

3 种算法	利用率/%		
	实例 1	实例 2	实例 3
粒子群算法	84.846	83.175	86.238
变邻域搜索算法	85.658	82.613	87.873
粒子群混合变邻域搜索算法	92.281	91.244	92.245

在表 4 中,通过粒子群算法和变邻域搜索算法

计算结果的对比可以看出,变邻域搜索算法计算出的木材板材的利用率与粒子群算法相近,甚至有时会超过粒子群算法,说明变邻域搜索算法也同样适用于木材板材下料问题的求解。但没有达到预期中的理想效果,仍有需要改进的地方。

通过粒子群算法与粒子群混合变邻域搜索计算结果的对比可以看出,实例 1 利用率从 84.846% 提升到 92.281%,实例 2 利用率从 83.175% 提升到 91.244%,实例 3 从 86.238% 提升到 92.245%,由改进后的粒子群算法计算出的木材板材利用率明显比粒子群算法高出很多。粒子群混合变邻域搜索算法不会受到规格数量的影响,计算出的利用率可以达到 90% 以上,同时也不会产生面积较大的废料。这也充分证明了本研究提出的将变邻域搜索算法的思想加入粒子群算法中,能够有效解决粒子群算法容易早熟缺陷、局部搜索能力差的问题。

综上所述,本研究 3 种算法中,最优异且适用的算法为粒子群混合变邻域搜索算法,适合在家具等相关行业木材板材实际生产中进行下料计算,解决了木材板材加工过程中的原材料利用率不高的难题,给家具等相关行业带来了实际的经济效益。

4 结 语

本研究主要根据某公司生产的木家具产品将其转化为单规格的二维矩形下料问题,并将其运用到该企业的生产实际中。在利用离散粒子群进行求解过程中,通过动态改变邻域结构来改善粒子群算法容易陷入早熟的情况。该算法避免了粒子群算法易陷入局部最优解的问题,给变邻域搜索算法提供了质量较高的初始解,加快了算法的收敛速度,使得结果趋于全局最优。为验证算法的可靠性,根据公司生产实例,设计了 3 组基于利用率的试验分析,通过 3 种不同的算法对矩形板材进行切割计算,以实例 3 为例,在选取长度为 230 cm、宽度为 160 cm 的单一规格的规则矩形板材后,改进后的粒子群算法的利用率由 86.238% 提升为 92.245%。实验结果表明,粒子群混合变邻域搜索算法可以有效地节约木材板材资源,提升企业经济效益。

参考文献(References):

[1] 李勇. 近似算法在排样优化中的应用[D]. 武汉: 华中科技大学, 2005.
LI Y. Approximate algorithm applied in stock cutting problems [D]. Wuhan: Huazhong University of Science and Technology, 2005.

- [2] DOOSTMOHAMMADI M, AKARTUNALI K. Valid inequalities for two-period relaxations of big-bucket lot-sizing problems: zero setup case[J]. *European Journal of Operational Research*, 2018, 267(1): 86–95. DOI: 10.1016/j.ejor.2017.11.014.
- [3] SONG X, BENNELL J A. Column generation and sequential heuristic procedure for solving an irregular shape cutting stock problem[J]. *Journal of the Operational Research Society*, 2014, 65(7): 1037–1052. DOI: 10.1057/jors.2013.44.
- [4] WÄSCHER G, HAUBNER H, SCHUMANN H. An improved topology of cutting and packing problems[J]. *European Journal of Operational Research*, 2007, 183(3): 1109–1130. DOI: 10.1016/j.ejor.2005.12.047.
- [5] GILMORE P C, GOMORY R E. A linear programming approach to the cutting-stock problem[J]. *Operations Research*, 1961, 9(6): 849–859. DOI: 10.1287/opre.9.6.849.
- [6] GILMORE P C, GOMORY R E. Multistage cutting stock problems of two and more dimensions[J]. *Operations Research*, 1965, 13(1): 94–120. DOI: 10.1287/opre.13.1.94.
- [7] 李琼琼, 布升强, 杨家富. 生物群体智能算法在移动机器人路径规划中的应用研究综述[J]. *世界科技研究与发展*, 2021, 43(5): 535–546. DOI: 10.16507/j.issn.1006–6055.2021.04.004.
- LI Q Q, BU S Q, YANG J F. Research review on biological swarm intelligence algorithm in mobile robot path planning[J]. *World Sci-Tech R & D*, 2021, 43(5): 535–546.
- [8] 张晓, 刘英, 李玉荣, 等. 基于卷积神经网络的经济林产品检测与分选研究进展[J]. *世界林业研究*, 2021, 34(5): 81–86. DOI: 10.13348/j.cnki.sjlyyj.2021.0048.y.
- ZHANG X, LIU Y, LI Y R, et al. Research on detection and sorting of products from non-wood forest based on convolution neural network[J]. *World Forestry Research*, 2021, 34(5): 81–86.
- [9] ALVAREZ-VALDES R, PARREÑO F, TAMARIT J M. A branch-and-cut algorithm for the pallet loading problem[J]. *Computers & Operations Research*, 2005, 32(11): 3007–3029. DOI: 10.1016/j.cor.2004.04.010.
- [10] ALVAREZ-VALDES R, PARREÑO F, TAMARIT J M. A tabu search algorithm for a two-dimensional non-guillotine cutting problem[J]. *European Journal of Operational Research*, 2007, 183(3): 1167–1182. DOI: 10.1016/j.ejor.2005.11.068.
- [11] JAKOBS S. On genetic algorithms for the packing of polygons[J]. *European Journal of Operational Research*, 1996, 88(1): 165–181. DOI: 10.1016/0377–2217(94)00166–9.
- [12] 陈燕, 郑欣亮, 鲁淑飞, 等. 考虑切割成本的矩形件优化下料算法[J]. *计算机集成制造系统*, 2022, 28(12): 3859–3868. DOI: 10.13196/j.cims.2022.12.013.
- CHEN Y, ZHENG X L, LU S F, et al. Considering cutting costs optimization of rectangular pieces of a blanking algorithm [J]. *Journal of Computer Integrated Manufacturing System*, 2022, 28(12): 3859–3868.
- [13] KENNEDY J, EBERHART R. Particle swarm optimization[C]// *Proceedings of ICNN'95-International Conference on Neural Networks*. November 27–December 1, 1995, Perth, WA, Australia: IEEE, 2002: 1942–1948. DOI: 10.1109/ICNN.1995.488968.
- [14] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization[J]. *Swarm Intelligence*, 2007, 1(1): 33–57. DOI: 10.1007/s11721–007–0002–0.
- [15] YE M Y, YAN X A, JIA M P. Rolling bearing fault diagnosis based on VMD-MPE and PSO-SVM[J]. *Entropy*, 2021, 23(6): 762. DOI: 10.3390/e23060762.
- [16] PARSOPOULOS K E. Parallel cooperative micro-particle swarm optimization: a master-slave model[J]. *Applied Soft Computing*, 2012, 12(11): 3552–3579. DOI: 10.1016/j.asoc.2012.07.013.
- [17] 钟一文, 杨建刚, 宁正元. 求解 TSP 问题的离散粒子群优化算法[J]. *系统工程理论与实践*, 2006, 26(6): 88–94. DOI: 10.3321/j.issn: 1000–6788.2006.06.015.
- ZHONG Y W, YANG J G, NING Z Y. Discrete particle swarm optimization algorithm for TSP problem[J]. *Systems Engineering-Theory & Practice*, 2006, 26(6): 88–94.
- [18] COELLO C C A, REYES-SIERRA M. Multi-objective particle swarm optimizers: a survey of the state-of-the-art [J]. *International Journal of Computational Intelligence Research*, 2006, 2(3): 287–308. DOI: 10.5019/j.ijcir.2006.68.
- [19] MLADENOVIC N, HANSEN P. Variable neighborhood search [J]. *Computers & Operations Research*, 1997, 24(11): 1097–1100. DOI: 10.1016/S0305–0548(97)00031–2.
- [20] BOUFFARD V, FERLAND J A. Improving simulated annealing with variable neighborhood search to solve the resource-constrained scheduling problem [J]. *Journal of Scheduling*, 2007, 10(6): 375–386. DOI: 10.1007/s10951–007–0043–7.
- [21] RIBEIRO C C, SOUZA M C. Variable neighborhood search for the degree-constrained minimum spanning tree problem[J]. *Discrete Applied Mathematics*, 2002, 118(1/2): 43–54. DOI: 10.1016/S0166–218X(01)00255–4.
- [22] 董伟. 变邻域搜索算法研究及在组合优化中的应用[D]. 阜新: 辽宁工程技术大学, 2011.
- DONG W. Research of variable neighborhood search and application in combinatorial optimization[D]. Fuxin: Liaoning Technical University, 2011.

(责任编辑 田亚玲)