计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(2):168−175 [doi: 10.15888/j.cnki.csa.008330]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于深度强化学习的二维不规则多边形排样方法①

曾焕荣, 商慧亮

(复旦大学 工程与应用技术研究院, 上海 200433)
通信作者: 曾焕荣, E-mail: hrzeng18@fudan.edu.cn

摘　要: 本文将深度强化学习应用于二维不规则多边形的排样问题中, 使用质心到轮廓距离将多边形的形状特征映射到一维向量当中, 对于在随机产生的多边形中实现了 1% 以内的压缩损失. 给定多边形零件序列, 本文使用多任务的深度强化学习模型对不规则排样件的顺序以及旋转角度进行预测, 得到优于标准启发式算法 5%–10% 的排样效果, 并在足够次数的采样后得到优于优化后的遗传算法的结果, 能够在最短时间内得到一个较优的初始解, 具有一定的泛化能力.

关键词: 排样优化问题; 组合优化问题; 深度强化学习; 编码器-解码器结构; 行动家-评论家算法

引用格式: 曾焕荣,商慧亮.基于深度强化学习的二维不规则多边形排样方法.计算机系统应用,2022,31(2):168–175. http://www.c-s-a.org.cn/1003-3254/8330.html

Nesting Method of Two-dimensional Irregular Polygons Based on Deep Reinforcement Learning

ZENG Huan-Rong, SHANG Hui-Liang

(Academy for Engineering and Technology, Fudan University, Shanghai 200433, China)

Abstract: This study applies deep reinforcement learning to the nesting problem of two-dimensional irregular polygons. The shape characteristics of polygons are mapped into one-dimensional vectors according to the distances from the centroid to the contours. For randomly generated polygons, the compression losses are less than 1%. With a given sequence of the polygon items, this study employs a multi-task deep reinforcement learning model to predict the sequence and rotation angle of the irregular nesting items and obtains a nesting result 5%–10% higher than those of the traditional heuristic algorithms. A result better than that of the optimized genetic algorithm is also achieved under a sufficient sampling number. The model can deliver a better initial solution in the shortest time and, therefore, has a generalization ability.

Key words: nesting optimization problem; combinatorial optimization problem; deep reinforcement learning; encoder-decoder structure; actor-critic algorithm

排样问题 (nesting problem) 又称下料问题 (cutting stock problem), 是一种经典的带几何约束的组合优化问题, 主要涉及到数学、运筹学、信息与计算科学以及工程管理学等学科. 当排样对象限定在二维空间时, 该问题是指将多个零件互不重叠地摆放到板材当中, 且不超出板材限定的空间, 要求在所有零件完成摆放以后, 板材的空间利用率最大, 或者说浪费空间最小, 这两个指标只要得到微小的提升, 就能为企业节约大量的材料成本. 根据排样对象是否为规则形状, 二维排样问题又可分为二维矩形排样和二维不规则排样. 二维不规则排样问题由于在工业生产中的应用更为广泛, 因此相比二维矩形排样问题有着更大的研究意义, 但是由于二维不规则排样问题中零件的形状多变, 在特征提取、序列决策以及重叠判断等问题中有着更高的复杂度.

排样问题为经典的组合优化问题, 由于其 NP-Hard 的特性[1], 这类问题的解空间非常大, 时间复杂度随着

① 收稿时间: 2021-04-19; 修改时间: 2021-05-11; 采用时间: 2021-06-07; csa 在线出版时间: 2022-01-17

问题规模的增加迅速上升, 特别是涉及到几何计算时. 因此在大多数情况下, 排样算法主要是基于特定规则的启发式算法和以智能搜索为基础的元启发式算法为主. 但是近年来, 随着深度强化学习[2]研究热度的提升, 研究人员也开始将深度强化学习应用在组合优化类问题当中[3]. 深度学习的训练通常需要大量带标签的训练样本, 但对于组合优化问题来说, 获取大量有标记的数据是很困难的, 一种思路是使用启发式算法得到的结果作为数据标签, 但是这种方法无法得到比启发式算法更优的解; 另一种思路是使用强化学习算法, 由于无需使用有标记的训练数据, 且组合优化问题通常有着很明确的优化目标, 因此奖励函数的设计较容易. 同时, 许多组合优化问题的本质都是序列决策问题, 因此也非常适合使用强化学习方法. 深度强化学习求解组合优化问题的大致思路为: 首先将样本表示为可输入神经网络模型的形式, 通过大量样本对深度神经网络模型进行训练, 同时需要避免产生不符合约束条件的解, 训练完成后将模型作为求解器, 在测试阶段即可把对测试样例求解的过程转化为一次神经网络的前向传播过程.

深度强化学习在组合优化问题上的成功应用[4]印证了其在解决排样优化问题上的可行性. Hu 等人[5]首次将指针网络 (pointer network)[6]用于解决三维排样问题, 主要思想是用深度强化学习的方法来解决排样件的定序问题, 而定位问题以及排样件的摆放方向问题则通过传统启发式算法来解决; Duan 等人[7]对这种思路进一步改进, 提出了多任务的深度强化学习模型, 使用了监督学习的方法来预测箱子的摆放方向; Zhao 等人[8]为了解决在线装箱问题 (即模型在某时刻仅能得到下一个排样件的信息), 使用卷积神经网络与强化学习生成可行性掩码 (feasibility mask), 从而直接预测排样件的排样位置; Hu 等人[9]使用深度强化学习与 Seq2Seq 模型来解决二维及三维排样问题中装入顺序的依赖问题. 以上研究有一个共同特点, 即排样对象的形状都是规则的, 如二维排样问题中的矩形与三维排样问题中的立方体, 而工业生产中的排样对象更多的是不规则的, 将深度强化学习应用在不规则物体的排样问题中有着更好的研究意义与应用前景.

1 问题描述与求解框架

1.1 问题描述

本文在问题描述以及数学建模中所用到的变量定义如表 1 所示.

若有n个排样零件, 在给定的二维矩形排样空间$\mathbb{E}$中, 根据排样对象的几何特性在$\mathbb{E}$中搜寻空间子集, 其优化目标为:

$$\min L = x_{\max} - x_{\min} \tag{1}$$

$$\max p = \frac{\sum_{i=1}^{n} Area_i}{L_{\min} \times W} \tag{2}$$

即式 (1) 在固定矩形空间宽度的情况下, 使得排样后所围成的矩形长度 L 最小化; 式 (2) 使排样空间的利用率 p 最大化.

表 1 变量定义

变量符号	变量含义
i	零件序号
S_i	第i个零件所占用的空间
$Area_i$	第i个零件所占用的面积
W	排样空间的宽度
p	排样后的面积利用率
L	排样后多边形围成的矩形长度
$x_{\min}$	排样后多边形顶点的最小横坐标值
$x_{\max}$	排样后多边形顶点的最大横坐标值

同时, 以上目标函数需要满足以下约束:

$$\begin{cases} S_i \cap S_j = \emptyset, & i \neq j \\ S_i \in \mathbb{E} \end{cases} \tag{3}$$

即选取的空间子集不得超出$\mathbb{E}$所限制的空间, 且任意零件之间互不重叠. 出于排样任务的不同, 可能会有旋转等空间变换限制.

1.2 求解框架

排样优化问题是一个 NP-Hard 的离散组合优化问题, 研究人员提出了各种各样的算法来解决“组合爆炸”这一难题, 但现今并没有一个算法能够随着问题规模的增大而在多项式时间内能够求得最优解. 排样优化问题中的组合个数 T 可以通过式 (4) 进行计算:

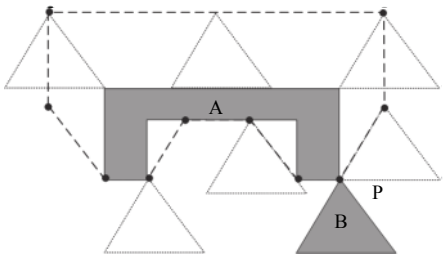
$$T(N, \theta) = \left(\frac{360}{\theta}\right)^N \times N! \tag{4}$$

其中, θ为物件允许旋转的角度, N为待排样的物件个数, 在$N = 15$时, 不考虑旋转的情况下, 其排列组合数就约有1.3×10^{12}种. 若考虑旋转的情况, 解空间则会迅速增大. 确定性算法如线性规划法只能在极小规模的排样问题中在可接受的计算时间内得到最优解, 中小规模的问题可以在启发式算法的基础上应用元启发式算法进行优化, 从而在可接受的时间内得到较为理想的解. 排样问题的求解框架可归纳总结为两大模块, 即定序算法以及定位算法两部分. 定序算法用于搜索一组最优的排样顺序, 必要时可以对形状进行旋转操作, 目

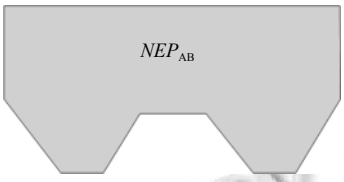
标是使调用定位算法解码后的板材利用率最大; 定位算法用于对搜索到的序列进行解码, 由算法中的定位规则确定零件在板材中的具体排放位置, 由此生成排样图, 并计算板材利用率. 二维不规则排样中, 常用的定序算法有随机法、基于特定排样规则的启发式算法、基于搜索的元启发式算法等.

1.3 重叠检测

由于排样过程中任意零件之间不得发生重叠, 本文在实验阶段使用临界多边形完成零件间的重叠检测. 临界多边形 (no-fit polygon, NFP)[10] 用于定义两个形状之间的重叠区域. 每个形状都有一个参考点. 假设有两个形状, 分别记为 A 和 B. 若把 A 的位置固定, A 和 B 之间的临界多边形是由 B 的参考点沿 A 的边缘滑动一周的轨迹所围成的闭合多边形, 记为NFP_{AB}, 在运动过程中, B 与 A 保持接触且不重叠. 图 1 给出了一个临界多边形的构建例子, 其中, 图 1(a) 为形状 B 的参考点 P 沿着形状 A 运行而形成的轨迹, 图 1(b) 为由运动轨迹生成的NFP_{AB}.



(a) 形状 B 沿着形状 A 运动



(b) 形成的临界多边形 NEP_{AB}

图 1 临界多边形构建过程

临界多边形的几何意义为:

(1) 若将 B 摆放以后, 其参考点位于NFP_{AB}以内, 则说明 A 与 B 之间有重叠的部分;

(2) 若参考点位于NFP_{AB}以外, 则 A 与 B 之间不重叠;

(3) 若参考点位于NFP_{AB}的边界, 则说明 A 与 B 相邻.

因此, NFP_{AB}的边界及其外部是可以放置 B 并避免与 A 发生重叠的可行区域. 使用了该方法以后, 重叠检测可以简化为判断参考点是否在临界多边形以内, 极大降低了排样过程中的几何运算量.

2 形状特征提取

在机器学习中, 常常需要用向量或矩阵来表示学习对象, 作为网络的输入. 向量或矩阵之间的欧氏距离也是衡量两个目标之间相似性的一个指标.

规则排样最大的特点是形状特征的表示通常较为简单, 如仅用长、宽两个值便可表示一个矩形. 而在二维不规则排样问题中, 特征点往往比较多, 不同的求解方案通常使用不同的几何表示, 而几何图形的表示法一定程度上决定了模型和算法的设计、计算精度以及计算时间. 形状特征的提取方法常常用于形状分类、目标检测等问题上, 常见的特征有链码、傅里叶描述子、形状上下文等[11]. 在这类问题中, 一个良好的特征表示通常在旋转、平移和仿射变换下是不变的. 但是在排样问题中, 由于图形旋转及仿射变换对其摆放位置的选择影响较大, 因此本文仅考虑平移不变性. 即形状相同, 但是旋转角度不同的两个图形以及经过仿射变换的图形可以看作不同的图形.

在排样问题当中, 形状的区域信息以及轮廓信息同等重要. 作为神经网络的输入, 特征向量不能有太多的冗余, 同时要保证对数据的表示要有一定的精度. 本文使用多边形质心到轮廓距离作为特征编码, 充分考虑到了形状的区域信息以及轮廓信息, 把形状特征嵌入到一维向量从而便于输入到神经网络中.

质心是多边形的几何中心, 可以通过对多边形轮廓线上均等采样点的坐标求均值得到. 虽然说质心的计算方法较为固定, 但是该质心-轮廓距离的计算方法一般只适用于凸多边形, 以及质心在图形内部的非凸多边形. 在排样问题当中, 经常会遇到一些较为复杂的多边形, 其质心位于多边形以外, 若简单地使用该质心计算其到多边形轮廓的距离, 对进一步的研究没有任何意义. 为了解决该问题, 可以将质心移至图形内部[12,13]. 考虑到现实排样问题中大多排样对象质心都在图形内部, 为了简化问题的运算, 本文主要考虑质心在多边形内部的情况.

欧式几何距离 (Euclidean distance) 是指在 n 维空间中的两个点之间的直线距离, 或者向量的自然长度, 即点到原点的距离. 在二维空间中, 点 a 与点 b 之间的欧式距离可用式 (5) 计算:

$$L(a,b)=\sqrt{(x_a-x_b)^2+(y_a-y_b)^2} \tag{5}$$

在求得多边形的质心以后, 可通过欧式距离计算公式获得质心到轮廓的距离, 主要思路为: 从以质心为原点发散N条射线, 相邻射线之间的距离为$\frac{2\pi}{N}$, 选定一

条起始射线, 取质心到交点的距离加入特征向量中, 若射线与形状之间有多个交点, 则取距离质心最远的交点并把该距离加入特征向量中, 如图 2 所示, 该图形的质心坐标$(x_c,y_c)=(5,5)$, 当向量维度为 10×1 时, 从该图形的质心引出 10 条射线, 获得质心到交点的距离加入到特征向量当中, 获得的特征向量为: $V=(1.50,0.85,1.58,0.85,1.50,0.85,1.58,1.58,0.85)^{\mathrm{T}}$.

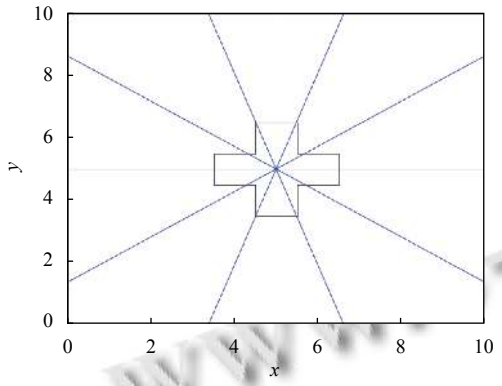


图 2 形状特征向量提取

为了验证形状提取特征的效果, 本文对特征向量进行形状重建, 并与原形状作对比. 本文随机生成了 3 030 个顶点个数在 3–8 个之间的多边形加入测试. 使用$N=180$对图形进行向量化表示, 并对该向量进行形状重建, 以评价该向量对图形的表示效果. 评价指标有两个, 分别为: (1) 面积覆盖率 (ACR), 用于评价重建后的形状对原有图形的覆盖情况; (2) 面积超出率 (AER), 用于表示重建后的形状超出原有图形部分的占比. 通过统计这两个指标不同范围内的形状个数, 可以评价出该向量对形状特征的提取效果. 结果如表 2 所示.

表 2 形状重建结果

AER	ACR						
	>99.5%	99%–99.5%	98%–99%	96%–98%	94%–96%	92%–94%	<92%
<0.1%	2 338	35	15	3	12	5	31
0.1%–0.5%	538	8	7	5	5	2	4
0.5%–1%	4	1	1	0	0	0	3
>1%	1	0	0	0	0	0	12

由此可见, 基于质心-轮廓距离的特征提取法能够基本实现 1% 以内的压缩损失, 足以表达形状的语义信息, 便于神经网络的训练. 此外, 本文对不同的 N 的重建效果也进行了测试, 测试结果发现重建效果也跟 N 的大小相关, 即 N 越大 (小于 360), 面积覆盖率就越大, 面积超出率就越小, 重建效果越好.

3 算法描述

3.1 编码器-解码器结构

本文在 Duan 等人[7] 提出的多任务三维装箱模型基础之上, 提出了一种融合注意力机制以及多任务的不规则多边形排样序列预测模型, 整体采用了基于编码器-解码器[14] 的结构, 由于输入零件的数目不定, 传统的神经网络难以处理不定长的输入. 一种解决思路是用 Seq2Seq 模型[15], 在编码阶段, 每一时刻输入一个零件信息, 在解码阶段将编码器的输出作为解码器的输入, 输出目标类的条件概率分布, 但是其输出目标类的长度是固定的, 对于排样问题此类的组合优化问题, 其输出的目标类数量完全取决于输入序列的长度, 而输入是一个可变的序列, 因此使用普通的 Seq2Seq 难以解决如排样问题这类的序列决策问题, 但是在基本的 Seq2Seq 模型中加入注意力机制可以很好地解决此类问题, 指针网络就是一种典型的使用此方法的模型, 用于保证输出只能从输入中选择这个先验信息. 对于传统的注意力模型, 在计算权重之后会对编码器的隐层进行加权, 求得加权后的向量. 而指针网络则在计算权重之后, 直接选择概率最大的编码器状态作为输出. 此外, 在本文中, 编码器与解码器均使用 LSTM[16] 网络结构以解决梯度消失的问题.

3.2 Actor-Critic 算法

Actor-Critic 算法源于策略梯度[17] 方法, 并在此基础上结合了基于值函数的方法. Actor-Critic 算法需要同时训练 Actor 和 Critic 两个神经网络, 分别负责学习策略和值函数:

Actor 网络也称策略网络, 用神经网络来表示策略函数. 根据输入信息学习动作集上的概率分布, 基于概率生成动作, 并根据 Critic 网络的评价调整策略, 网络输出是动作. 在本文中, Actor 网络的输入序列$x=(x_1,x_2,\cdots,x_n)$是多边形的特征向量序列, 输出$y=(y_1,y_2,\cdots,y_n)$为排样件的排样顺序以及方向. 策略函数$p_\theta(y|x)$表示给定输入序列x的情况下输出y的概率. 本文选择最短排样长度作为模型的奖励信息, 则 Actor 网络的作用就是增加能够获得最短排样长度的输出方案被选择的概率.

Critic 网络也称估值网络, 通过计算值函数来评估策略. 根据 Actor 网络的动作评价策略的价值, 并反馈给 Actor 网络, 网络输出是对目标函数的预测.

3.3 Actor 网络结构

本文主要使用深度强化学习的方法来解决排样问

题中的定序问题, 而该问题又包括两个子问题, 一是零件的排样顺序, 二是零件的旋转角度. Hu 等人[9] 在解决二维矩形排样问题时, 将经旋转后的矩形与原矩形看作为两个不同的形状并输入到 RNN 神经网络中, 若其中一个形状被选择, 则使用屏蔽机制将两个形状同时屏蔽. 但是不规则图形在排样中可选择的角度较多, 若使用该方法则会使得输入的形状数量成倍地增加. 因此, 本文在引言所述的三维装箱问题解决方案的基础之上, 设计了一种改进型的基于多任务的二维不规则排样定序算法, 可以在零件序号的选择的同时, 确定零件的旋转角度. 图 3 为本文 Actor 网络的架构图.

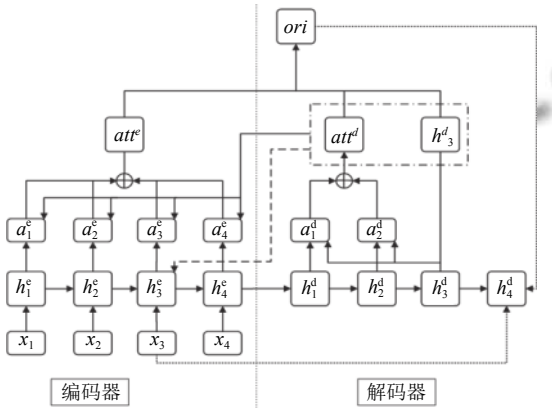


图 3 Actor 网络架构图

由于零件序列的决策受已排列零件的影响较大, 为了充分利用解码器所产生的序列信息, 在编码阶段以及解码阶段均使用了注意力机制, 在解码阶段, 在t时刻之前产生的零件权重可用式 (6) 计算:

$$a_j^d = Softmax(v_1^{\mathrm{T}}\tanh(W_1h_j^d + W_2h_t^d)), j \in \{1,\cdots,t-1\} \quad (6)$$

通过把t时刻前产生的零件隐层向量与其权重进行加权求和, 可以得到解码器在t时刻的权重向量:

$$att_t^d = \sum_{j=1}^{t-1} a_j^d \times h_j^d \quad (7)$$

在零件序号的确定上, 本文运用指向机制, 使用通过式 (7) 得到的加权向量可以用来计算“指针”所指向量的概率分布. 在$t-1$时刻, 解码器输出零件序号s_{t-1}与旋转角度ori_{t-1}后形成旋转后的零件图形, 经形状特征提取后作为y_t输入到解码器网络中, 得到h_j^d, 将其与att_t^d进行向量拼接以后得到的新向量可以用于预测零件被选择的概率. 在t时刻, 零件序号的选择概率如式 (8)、式 (9) 所示:

$$u_j^t = v_2^{\mathrm{T}}\tanh(W_3h_j^e + W_4[att_t^d : h_t^d]), j \in (1,\cdots,n) \quad (8)$$

$$p(x_j|y_1,y_2,\cdots,y_{t-1},y_t) = Softmax(u_j^t), j \in (1,\cdots,n) \quad (9)$$

其中, h_j^e表示编码器中第j个形状的隐层向量, h_t^d表示解码器在t时刻的隐层向量, v^{T}为可学习的注意力向量, W为可学习的注意力矩阵.

经过式 (9) 的计算后可以得到t时刻每一个零件被选择的概率. 由于在排样问题中一般不允许已选择的零件再次被选择, 因此, 可以运用屏蔽机制, 通过将决策序列中出现过的零件的概率置为 0, 确保模型只会指向未被选择过的零件, 具体如式 (10) 所示:

$$u_j^t = \begin{cases} v_2^{\mathrm{T}}\tanh(W_3h_j^e + W_4[att_t^d : h_t^d]), t \neq \pi(j), \pi(j) \in (1,\cdots,t-1) \\ 0, \text{ 其他} \end{cases} \quad (10)$$

其中, $\pi(j)$表示j号零件被选择的时间, 如果没有被选择过, 则该值为 0.

式 (9) 所得概率可以视为每个零件的注意力权重, 使用该权重对编码器中零件的隐层向量进行加权求和, 可得到编码器在t时刻的注意力加权向量, 如式 (11)、式 (12) 所示:

$$a_j^e = p(x_j|y_1,y_2,\cdots,y_{t-1},y_t), j \in \{1,\cdots,n\} \quad (11)$$

$$att_t^e = \sum_{j=1}^{n} a_j^e \times h_j^e \quad (12)$$

通过把att_t^d、att_t^e、h_t^d进行向量拼接, 我们可以得到t时刻的语义向量, 本文使用该向量进行零件旋转角度的预测, 如式 (13) 所示:

$$p(ori_i) = Softmax(\sigma(W_5[att_t^d : att_t^e : h_t^d] + b)) \quad (13)$$

其中, σ为激活函数, W与b均为可学习的参数. 若零件允许旋转的最小角度为θ, 则其输出有$\frac{360}{\theta}$个类, 第i类输出代表零件旋转$\theta \times (i-1)$度.

编码器-解码器模型中, 编码器负责处理输入的排样件信息. 一个排样实例中的元素应该是无序的, 而在编码器中零件的形状信息是按顺序输入到神经网络的, 会对神经网络的求解造成一定的影响[16]. 因此, 本文在原有模型的基础之上, 加入 glimpse 机制. 这一操作可以在计算时间不明显增加的同时, 能够较好地消除输入顺序对输出结果的影响. 则零件被选择的概率可以通过以下公式得到:

$$w_j^t = v_3^{\mathrm{T}}\tanh(W_6h_j^e + W_7att_t^e), j \in (1,\cdots,n) \quad (14)$$

$$p(x_j|y_1,y_2,\cdots,y_t) = Softmax(w_j^t), j \in (1,\cdots,n) \quad (15)$$

使用式 (14)、式 (15) 替换式 (8)、式 (9) 即可使

glimpse 机制生效.

3.4 Critic 网络结构

在组合优化类问题强化学习的训练过程中, 智能体与环境交互以后获得可获得一个预测序列以及其奖励 (reward), 在二维排样问题中, 奖励可以是板材面积利用率或者排样后多边形围成的矩形长度. 此时需要一个基准值 (baseline) 对此预测序列的效果进行估计, 然后用这个估计值代替真实的奖励值形成策略梯度, 再用这个梯度来进行网络的更新.

Hu 等人[5] 在其模型中使用了一种类似记忆重放的方法来更新基准值, 首先使用启发式算法对每个样本s_i都获取一个预测序列o_i, 并计算出其奖励值为$b(s_i)$的初始值, 之后的训练过程中通过以下方式更新基准值:

$$b(s_i) \leftarrow b(s_i) + \alpha(reward(o_i|s_i) - b(s_i)) \tag{16}$$

其中, $reward$为对o_i使用如启发式算法这类的传统方法后求解得到的值, 但是若在大规模的训练集上使用传统方法进行基准值的求解, 无疑会造成大量时间与资源的浪费. 另外一种方法是使用 Critic 网络来预估输入序列的基准值, 训练好的 Critic 网络能够较好地预估基准值, 在节约了使用传统方法计算基准值时间的同时, 降低了梯度方差, 显著地提升了模型的性能[18].

本文同样使用编码器-解码器结构作为 Critic 网络. 其中, 编码器结构与 Actor 网络一致, 将零件的特征向量x输入映射到隐层向量h中, 并将该隐层向量输入到 LSTM 网络中, 随后, 编码器的隐层向量被送往解码器的 LSTM 处理块 (processing blocks) 中, 若有m个处理块, 则对编码器中的隐层向量进行m次运算, 并运用 glimpse 机制消除输入序列间的依赖关系. 最后, 在得到最后一个处理块的输出以后, 输入到层数分别为l和 1 的两个全连接层当中, 将最后一个全连接层的输出作为对基准值$b(s_i)$的预测, 即s_i序列预期获得的奖励值.

3.5 训练

3.5.1 探索与利用

若模型在对零件序列进行预测时, 为了短期利益仅根据已掌握的信息做决策, 即仅局限于已知的最优动作, 选择当前概率最大的零件, 则有可能因为没有环境中获得足够的信息而学习不到全局最优解. 为了更好地对环境进行探索, 模型在进行序列决策的时候需要采取一些与当前策略不同的决策. 在训练过程中, 模型根据ε-greedy策略来进行序列决策, 即有ε的概率使用贪心策略以及$1-\varepsilon$的概率使用随机策略。具体操作为: 模型在 [0, 1] 区间内随机采样一个实数, 当该实数小于ε时, 则选择概率最大的决策; 当实数大于等于ε时, 则根据各决策的概率大小来选择决策.

3.5.2 损失定义

本文 Actor-Critic 框架使用回合更新的 REINFORCE 策略梯度法进行训练, 基于整个决策序列来训练网络优化策略函数. 网络的损失函数包含了两个损失, 分别为 Actor 网络的损失$L_{\theta|x}$以及 Critic 网络的损失$L_{\varphi|x}$. $L_{\theta|x}$可以通过以下公式进行计算.

$$L_{\theta|x} = E_{y\sim p_\theta(y|x)}(reward(y|x) - b_\varphi(x)) \tag{17}$$

式 (17) 中的数学期望无法直接计算, 通常构造多个排样序列$x_1, x_2, \cdots, x_B$并根据蒙特卡洛方法采样每个实例对应的排样序列, 其中$y \sim p_\theta(\cdot\ |x_i)$, 则式 (17) 的损失可以转化为:

$$L_{\theta|x} = \frac{1}{B}\sum_{i=1}^{B}(reward_\theta(y_i|\ x_i) - b_\varphi(x_i)) \tag{18}$$

评论家网络采用随机梯度下降的方法训练网络参数, 其目标函数为均方误差表示, 如式 (19) 所示:

$$L_{\varphi|x} = \frac{1}{B}\sum_{i=1}^{B}\|b_\varphi(x_i) - reward(y_i|\ x_i)\|_2^2 \tag{19}$$

3.5.3 算法流程

综合上述分析, 可以将本文模型的训练算法流程总结为算法 1.

算法 1. Actor-Critic 训练算法

输入: 训练集X, 训练步数T, 批样本容量B
输出: 返回网络参数θ

初始化网络参数
for step = 1 to T do
  $x_i \sim sample(X)$ for i = 1, 2, ···, B
  for i=1 to B do
    for i=1 to N do
      $x_{i,t} \sim \varepsilon-greedy(p_\theta(\cdot|y_{i,1}, y_{i,2}, \cdots, y_{i,t-1}))$
      $ori_{i,t} \sim \varepsilon-greedy(p_\theta(\cdot|y_{i,1}, y_{i,2}, \cdots, y_{i,t-1}))$
      $y_{i,t} \leftarrow (x_{i,t}, ori_{i,t})$
    $b_i \leftarrow b_\varphi(x_i)$
    end for
  end for
  $\nabla_\theta L_{\theta|x} \leftarrow \frac{1}{B}\sum_{i=1}^{B}(reward(y_i|x_i) - b_\varphi(x_i))\nabla_\theta \log p_\theta(y_i|x_i)$
  $L_{\phi|x} = \frac{1}{B}\sum_{i=1}^{B}\|b_\phi(x_i) - reward(y_i|\ x_i)\|_2^2$
  $\theta \leftarrow ADAM(\theta, \nabla_\theta L_{\theta|x})$
  $\phi \leftarrow ADAM(\phi, \nabla_\varphi L_{\phi|x})$
end for
return θ

4 实验与结果分析

4.1 数据集准备

本文介绍的基于机器学习的算法性能和数据集有较强的关联性, 为了能够合理比较本文所介绍算法的性能, 本文参考了目前流行的二维排样问题研究的数据集, 分别生成了用于训练和测试的多边形, 其中训练集 10 000 组, 测试集 300 组, 每组又分为 10、15、20 个多边形 3 种情况, 每种数量的数据集又分可旋转 (R) 与不可旋转 (NR) 两种情况. 多边形的顶点数量在 [3, 8] 之间, 面积在 [50, 300] 之间. 由于本文主要考虑质心在多边形内部的情况, 因此当生成的多边形质心在多边形外部时, 则将其丢弃. 为了加速训练过程, 本文在数据集生成后进行数据的预处理, 即计算每个图形的特征向量, 以及每一组多边形的 NFP 并进行本地缓存.

4.2 实验设置

排样宽度为 80, 最小旋转角度为 90 度, 优化目标为排样后多边形围成的矩形长度 L. 模型使用 Adam 优化器[19] 训练 300 个 epoch 完成, 并在测试集上进行测试, 训练过程中采用梯度截断防止梯度爆炸的产生. 其中, 神经网络模型的训练在 NVIDIA GeForce RTX 2080Ti GPU 上完成, 重叠检测、奖励值计算等操作以及传统方法如启发式算法及遗传算法的计算在 Intel Xeon E5-2667 v4 CPU 上完成.

为了验证本文模型的效果, 本文与随机法、启发式算法以及经典的遗传算法[20] 进行实验对比. 其中启发式算法对零件分别按特定规则进行排序 (如面积、长度等), 结果取其各种排列方式的最优值; 遗传算法具体的参数如表 3 所示, 其中变异包括了交叉与旋转两种情况.

表 3 遗传算法参数

参数	参数值
种群大小	30
交叉概率	0.5
旋转概率	0.5
变异概率	0.1
算法终止代数	40

在确定定序算法以后, 本文使用左下填充定位法 (bottom left fill) 作为排样的定位算法[21]. 按定序算法生成的排样顺序, 将零件逐个尽可能地排到底部, 再向左进行平移, 使其尽可能靠近最左侧. 并在所有空间与已排样件依次进行重合试排, 尽可能地将未被利用的空余空间填满, 从而减少中空的区域, 提高了整体的排样利用率.

4.3 实验结果与分析

实验阶段使用不同方法对各测试集进行排样, 取其排样长度的均值为实验结果, 如表 4 所示. 其中, 本文算法与随机法均采用了“多次采样”的策略, 取其中的最优排样长度为实验结果. 由于遗传算法的限制是每换一组输入序列都要重新花费时间来进行迭代计算, 本文的目标是使用机器学习方法设计一个通用的求解模型, 能够从数据中学习到高维特征, 对新的输入也能在最短的时间预测出较优的解决方案, 减少运行遗传算法所需的多余计算时间. 为突出本文算法相较于遗传算法在运行时间上的优势, 本文算法将采样次数设置为 100, 耗费时间约为遗传算法进行 3 次迭代所用时间. 由实验结果可以观察到, 在运行时间大幅减少的前提下, 本文算法仍能够得到比其它算法更优的解, 这在一定程度上验证了本文算法的可行性. 由于本文在解码器中加入了注意力机制, 模型能够根据已排列零件的信息对下一个零件的序号以及方向进行预测, 相比遗传算法, 能够使得新排入的零件尽可能地贴合已排列的零件, 面积利用率更高. 同时, 为了优化排样长度, 模型需要同时考虑尚未排列的零件信息, 做到两者之间的平衡.

表 4 排样长度均值实验结果

数据集	本文算法	遗传算法	启发式算法	随机法
10-R	45.22	45.26	46.29	53.41
15-R	50.62	50.81	52.07	59.94
20-R	63.08	63.17	66.24	71.44
10-NR	45.78	45.86	46.95	53.76
15-NR	50.87	50.93	53.60	60.21
20-NR	63.23	63.47	66.79	73.24

此外, 无论是深度强化学习法还是遗传算法均有一定的排样优化空间. 排样效果在一定程度上受旋转角度约束的影响, 理论上最小旋转角度越小, 能够得到最优排样的可能性就越高. 但是旋转角度过多会使网络的训练变得十分复杂, 且遗传算法也非常难以收敛于最优解. 为了简化问题复杂度, 本文将最小旋转角度仅限制在 90°, 即一个零件仅有 4 个方向可用于排样. 虽然对旋转角度进行了限制, 但是通过本文方法可以迅速获得较优的初始解, 随后可以使用收缩法[22] 对该解进行优化.

此外, 本文将排样空间的宽度W固定为 80, 以便于模型的训练, 为了把模型推广到其他高度, 可以使用缩放的思路, 即对于其他排样宽度W', 将多边形同比缩放

到W'/W倍后再进行特征提取, 接下来便可以使用预训练后的模型进行多边形的排样.

5 结论与展望

本文为不规则多边形的排样问题设计了一种基于Actor-Critic 算法与编解码结构的多任务深度强化学习模型. 通过质心到轮廓的距离提取多边形的形状特征, 并将该特征映射到定长的一维向量中, 使得神经网络能够学习到多边形的语义信息, 并对排样顺序、旋转角度进行预测. 由于本文中特征提取是基于有损的方法, 本文方法缺点在于无法处理复杂的图形排样, 在算力允许的条件下, 未来可以考虑使用无损的形状特征来处理复杂图形. 此外, 如果更换数据集, 则可能需要重新对模型进行训练, 但是通过预训练的方法, 可以使得网络能够适应新的数据集, 因此本文模型具有一定的泛化能力. 通过与传统排样算法的对比, 本文在最佳排样长度、运算时间等指标均有一定优势, 能够在最短时间生成合理的排样图, 并为大规模排样的解决提供了可能性, 具有实际的研究与应用前景.

参考文献

1 Bennell JA, Oliveira JF. A tutorial in irregular shape packing problems. Journal of the Operational Research Society, 2009, 60(1): S93–S105. [doi: 10.1057/jors.2008.169]

2 Mnih V, Kavukcuoglu K, Silver D, *et al*. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533. [doi: 10.1038/nature14236]

3 李凯文, 张涛, 王锐, 等. 基于深度强化学习的组合优化研究进展. 自动化学报, 2021, 47(11): 2521–2537. [doi: 10.16383/j.aas.c200551.]

4 Bello I, Pham H, Le QV, *et al*. Neural combinatorial optimization with reinforcement learning. arXiv: 1611.09940v3, 2017.

5 Hu HY, Zhang XD, Yan XW, *et al*. Solving a new 3D bin packing problem with deep reinforcement learning method. arXiv: 1708.05930, 2017.

6 Vinyals O, Fortunato M, Jaitly N. Pointer networks. arXiv: 1506.03134v2, 2017.

7 Duan L, Hu HY, Qian Y, *et al*. A multi-task selected learning approach for solving 3D flexible bin packing problem. Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems. Montreal: AAMAS, 2019. 1386–1394.

8 Zhao H, She QJ, Zhu CY, *et al*. Online 3D bin packing with constrained deep reinforcement learning. Proceedings of the 35th Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2021. 741–749.

9 Hu RZ, Xu JZ, Chen B, *et al*. TAP-Net: Transport-and-pack using reinforcement learning. ACM Transactions on Graphics, 2020, 39(6): 232.

10 Albano A, Sapuppo G. Optimal allocation of two-dimensional irregular shapes using heuristic search methods. IEEE Transactions on Systems, Man, and Cybernetics, 1980, 10(5): 242–248. [doi: 10.1109/TSMC.1980.4308483]

11 Kurnianggoro L, Wahyono, Jo KH. A survey of 2D shape representation: Methods, evaluations, and future research directions. Neurocomputing, 2018, 300: 1–16. [doi: 10.1016/j.neucom.2018.02.093]

12 陈涛, 艾廷华. 多边形骨架线与形心自动搜寻算法研究. 武汉大学学报·信息科学版, 2004, 29(5): 443–446, 455. [doi: 10.13203/j.whugis2004.05.015]

13 朱钰, 王伟, 章传银. 伪形心多边形形心距离计算方法. 测绘科学, 2018, 43(2): 6–9, 44. [doi: 10.16251/j.cnki.1009-2307.2018.02.002]

14 Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. arXiv: 1409.3215, 2014.

15 Vinyals O, Bengio S, Kudlur M. Order matters: Sequence to sequence for sets. arXiv: 1511.06391v4, 2016.

16 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

17 Peters J, Bagnell JA. Policy gradient methods. In: Sammut C, Webb GI, eds. Encyclopedia of Machine Learning and Data Mining. Boston: Springer, 2017. [doi: 10.1007/978-1-4899-7687-1_646]

18 Ilyas A, Engstrom L, Santurkar S, *et al*. A closer look at deep policy gradients. arXiv: 1811.02553, 2020.

19 Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv: 1412.6980v8, 2015.

20 Babu AR, Babu NR. A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. Computer-Aided Design, 2001, 33(12): 879–891. [doi: 10.1016/S0010-4485(00)00112-3]

21 Chazelle B. The bottom-left bin-packing heuristic: An efficient implementation. IEEE Transactions on Computers, 1983, C-32(8): 697–707. [doi: 10.1109/TC.1983.1676307]

22 Egeblad J, Nielsen BK, Odgaard A. Fast neighborhood search for two-and three-dimensional nesting problems. European Journal of Operational Research, 2007, 183(3): 1249–1266. [doi: 10.1016/j.ejor.2005.11.063]