

基于深度强化学习种群优化的演化式分拣调度算法*

曾德天, 曾增日, 詹俊[†]

(国防科技大学 计算机学院, 长沙 410073)

摘要: 机械制造中的产线分拣作业具有问题与数据的双重复杂性, 为了对分拣操作进行优化以提高生产效率, 设计了一套分拣作业的数据表示方法与一种基于种群优化的演化式算法, 同时整理并公开了一个真实的工业数据集。数据表示方法通过借鉴词袋模型对原始作业数据进行抽象表示; 演化式算法使用深度强化学习初始化遗传算法中的种群, 同时引入了精英保留策略以提高算法的优化能力。最后, 将提出的算法与其他算法在真实的工业数据集与旅行商问题数据集上进行了对比。结果表明, 该算法能找到更优的分拣顺序与访问路径, 验证了算法的有效性。

关键词: 遗传算法; 深度强化学习; 分拣作业调度; 顺序优化

中图分类号: TP301 **文献标志码:** A **文章编号:** 1001-3695(2022)03-016-0739-05

doi: 10.19734/j.issn.1001-3695.2021.08.0356

Evolutionary job scheduling algorithm based on population optimization by deep reinforcement learning

Zeng Detian, Zeng Zengri, Zhan Jun[†]

(College of Computer Science & Technology, National University of Defense Technology, Changsha 410073, China)

Abstract: The sorting operation of the production line in mechanical manufacturing has the double complexity of the problem and data. To optimize the sorting operation and improve production efficiency, this paper designed a method for data representation and an evolutionary algorithm based on population optimization. At the same time, this paper arranged and disclosed a real industrial data set. The method for data representation abstracted the original job data by referring to the bag-of-words model. The evolutionary algorithm used deep reinforcement learning to initialize the population in the genetic algorithm and introduced the elite retention strategy, which improved the optimization ability of the algorithm. Finally, it compared the proposed algorithm with other algorithms on the real industrial data set and travelling salesman problem data set. The results show that the proposed algorithm can find a better sorting sequence and the access path, which verifies the effectiveness of the algorithm.

Key words: genetic algorithm; deep reinforcement learning; sorting job scheduling; sequence optimization

工业分拣顺序优化 (optimization of industrial sorting sequence, OISS) 为制造业中真实的生产调度问题^[1]。OISS 是对分拣顺序进行优化的一项研究, 具有广泛的应用背景。由于 OISS 属于 NP-Hard 问题, 求解困难, 所以寻找高效优质的求解方法一直是相关领域的重要课题。

机械制造中的钢板分拣顺序优化作为典型的 OISS 问题, 可以将每个待分拣的钢板看做一个节点, 当需要求解最优钢板分拣顺序时, 此问题即被近似抽象成旅行商问题 (travelling salesman problem, TSP), 而如何在不同规模的输入数据上快速取得有效解是一项具有挑战性的任务^[2-4]。

传统的顺序优化主要使用元启发式算法^[5], 如遗传算法 (genetic algorithm, GA)^[6]、模拟退火 (simulated annealing, SA)^[7]、人工蜂群算法^[8]、灰狼算法^[9]等, 但始终无法取得令人满意的效果。Pettinger 等人^[10]则最早将强化学习 (reinforcement learning, RL) 与 GA 结合起来, 将 GA 中的种群作为 RL 的状态, 并通过传统 Q-学习算法选择不同的交叉和变异算子, 最终算法的表现优于传统 GA, 但仅限于 40 个节点以下的顺序优化。此方法带来了良好的启发, 之后的结合工作大部分遵循这一思路^[11-14]。而自 sequence-to-sequence 的代表网络 pointer network^[15]被提出以来, 神经网络求解顺序优化成为可能, 同时

RL 的引入进一步提升了神经网络的性能^[16]。受流行的 Transformer 结构影响, Kool 等人^[17]针对顺序优化将 Transformer 结构进行改进, 并将改进后的 Transformer 联合 RL 进行训练, 在 100 个节点左右的 TSP 问题上取得了最优结果。但当输入数据的规模进一步增大时, 深度网络的泛化性会变差, 传统的元启发式算法因维度恶化也易陷入局部最优, 故本文将探讨传统元启发式算法与深度强化学习 (deep reinforcement learning, DRL) 相结合共同求解顺序优化问题。

针对上述不同规模的钢板分拣顺序优化, 本文首先对场景中复杂的钢板数据进行了抽象表示, 并将 DRL 与 GA 相结合, 同时引入了 NSGA-II 的精英保留策略^[18], 最大限度提高了搜索算法的优化能力。而实际测试结果表明本文提出的算法在工业分拣顺序优化问题上可将生产效率最高提升 16.93%, 这也说明了生产制造中的 OISS 问题根据实际需求可以被进一步优化。

1 钢板分拣顺序优化问题描述

图 1 展示了机械制造生产线上的钢板顺序分拣流程。假设存在 $N \in N^*$ 块切割后的钢板需要被分拣, 而每块板中含有

收稿日期: 2021-08-17; **修回日期:** 2021-10-07 **基金项目:** 国家重点研究开发计划资助项目; 国家自然科学基金资助项目

作者简介: 曾德天 (1993-), 男, 湖北仙桃人, 博士研究生, 主要研究方向为人工智能与工业决策控制; 曾增日 (1989-), 男, 湖南娄底人, 博士研究生, 主要研究方向为人工智能与网络异常检测; 詹俊 (1988-), 男 (通信作者), 湖南益阳, 博士研究生, 主要研究方向为人工智能与工业异常检测 (zhanjun20@nudt.edu.cn)。

若干被切割后的零件,最终零件需要通过机械手臂被放置到物料框。出于工业标准化流程的要求,框中存在一定的码垛规则,即一个框中最多只能装载 $T \in N^*$ 种不同类型的零件,相同类型的零件被码放的层数不超过 $L \in N^*$ 层。一旦码垛规则不被满足,则需要清框处理,而频繁的清框需要反复调度 AGV 搬运,耗费额外的人力物力。一般智能工厂在规划时,需尽量减少无效的物料搬运^[19],因此可以通过调整 N 块钢板的分拣顺序以降低清框频率。钢板分拣顺序优化可以被抽象为一个带约束的优化问题,其中相关符号的定义见表 1。优化的目标及约束见式(1),即最小化 $F(x)$,其中两个约束对应于码垛规则(本文中 $T=16, L=10$)。

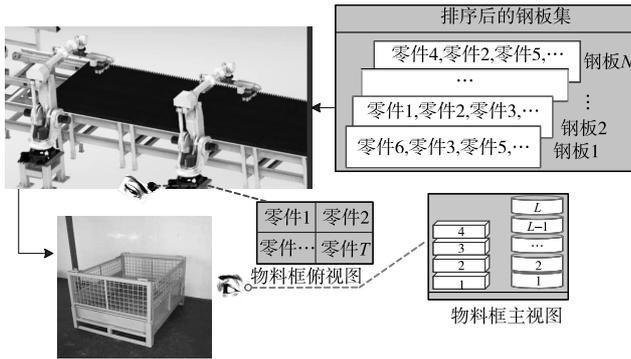


图 1 钢板分拣顺序优化示意图
Fig. 1 Structure diagram of scheduling optimization problem about sorting sequence

表 1 相关符号定义表
Tab. 1 Definition of related symbols

符号	定义
x	带分拣顺序的钢板数据,记为 $\{ \dots \rightarrow P_{i-1} \rightarrow P_j \rightarrow \dots \rightarrow P_{j-1} \rightarrow P_j \rightarrow \dots \}$, P 代表钢板,其中正整数下标 $i, j \in [1, N]$,且 $i \neq j$ 。
V_θ	当前物料框中任一区域零件的堆叠层数, $V_\theta \in N^*$ 。
θ	当前物料框中所有零件的种类数, $\theta \in N^*$ 。
F	目标函数。输入为 x ,根据码垛规则对给定分拣顺序的钢板集进行分拣,输出为总清框次数。其数学定义见式(1),为递归形式。初始化时 $F(x)$ 值为 0,每次清框被触发时, $F(x)$ 加 1。

$$\min: F(x; V_\theta, \theta) = \begin{cases} 0 & \theta = 0 \wedge V_\theta = 0 \\ F(x; V_\theta, \theta) + 1 & \theta > T \vee V_\theta > L \end{cases}$$

$$\text{s. t. } \begin{cases} V_\theta \leq L \\ \theta \leq T \end{cases} \quad (1)$$

由于零件种类繁多,故在钢板数据的抽象表示上本文借鉴了自然语言处理中的词袋模型^[20]。首先为所有的零件创建一个固定顺序的零件字典,共 $D \in N^*$ 项,之后创建一个对应字典中零件顺序的 D 维全零向量。当表示某一块钢板时,遍历钢板中所含的零件,若某一类型零件的数量为 $k \in N$,则将向量中对应位置上的数加 k 。最后的 D 维向量作为此钢板的原始表示数据。

为便于研究,本文约定机械手的分拣规则为:按字典中的零件种类顺序依次将钢板上某一类型的零件全部分拣完再处理下一类型零件。这一分拣规则契合零件布局规则,即钢板套料时倾向于将相同类型的零件排列在一起以达到最大化钢板的空间利用率,同时有助于减少机械臂分拣时地反复运动。图 2 中展示了典型的真实钢板零件分布,每个零件使用一种颜色的线条包围,可以看到相同形状的零件被尽量排布在一起。

2 算法设计

DRL 拥有较好的泛化性且网络的前向计算耗时较少,为了加速 GA 收敛同时取得更好的优化结果,本文将 DRL 与 GA

相结合,提出了一种新的演化式算法。

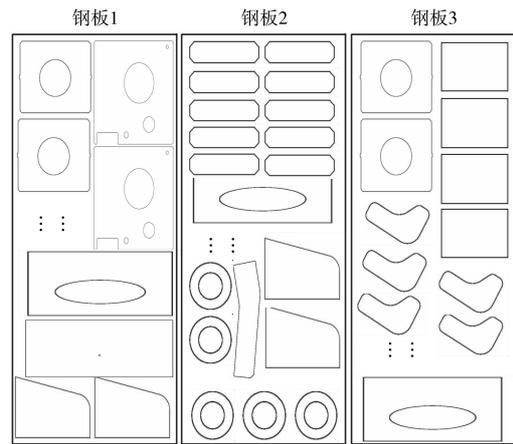


图 2 真实的钢板零件分布示意图
Fig. 2 Real distribution of parts in steel plate

演化式算法首先使用 RL 对策略网络进行训练,得出参数化的决策网络,之后将原始的钢板数据集抽象编码并输入训练完毕的决策网络,得到一个初始解(即钢板优化顺序),并将初始解补充到 GA 的初始化种群中,使得 GA 在初始时具有较高质量的解,从而减少探索阶段的时间开销。使得 GA 可以注入更多精力在开发阶段,从而加速整个过程收敛的同时提高结果的可靠性。最后,将最终种群中适应度值最小的个体作为最佳钢板分拣顺序。

2.1 深度强化学习

钢板的原始向量数据较为稀疏,将其通过一个 $M \in N^*$ 维的全连接层,得到一个 M 维的稠密向量作为嵌入表示,之后决策网络的输入均为钢板的嵌入表示。嵌入层与决策网络进行联合训练,其中决策网络选用了 pointer network 与 Transformer。pointer network 是一个典型的 encode-decode 架构,如图 3 所示。encode 和 decode 分别使用了单层的 LSTM 网络;同时对注意力机制进行了改造,使得每次 encode 的输出实际上是每个输入钢板此次可能被选择的概率向量(图中红色箭头指向拥有最大概率的钢板;黑色圆点代表此前已被选中的钢板,故当前时刻概率为负无穷),其维度和 encode 输入的序列长度一致,解决了输出的向量长度固定问题。pointer network 的注意力机制可简述为如下公式:

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i) \quad (2)$$

$$p(C_i | C_1, \dots, C_{i-1}, P_{all}) = \frac{e^{u_j^i}}{\sum_{k=1}^N e^{u_k^i}} \quad k \in [1, N] \wedge k \in N^* \quad (3)$$

其中: e_j 是 encode 在时间序列 j 次的隐藏层输出, d_i 是 decode 在时间序列 i 次的隐藏状态输出, $u^i = [u_1^i, u_2^i, \dots, u_N^i]$,对 u^i 做式(3)所代表的归一化操作可得到所有钢板被选中的概率向量; C_i 为 i 时刻选择的钢板编号, P_{all} 代表所有的钢板数据,其中 v^T, W_1, W_2 均为可训练的固定尺寸参数。而使用 Transformer 结构代替 LSTM 组成的 encode-decode 框架拥有更加强大的表征与泛化能力,具体的网络结构细节见文献[17]。

DRL 训练时使用了 Actor-Critic 框架^[21],并使用 pointer network 或 Transformer 网络作为 Actor 进行决策,同时将指数滑动平均作为 Critic 指导 Actor 进行参数更新。通常强化学习可以被抽象成一个马尔可夫决策过程,使用四元组 $\langle s, a, r, p \rangle$ 表示。其中 s 表示状态(state), a 表示动作(action), r 代表环境给予的奖励(reward), p 代表状态转移概率(probability)。在钢板分拣排序问题中, a 定义为选择的某一钢板, s 定义为所有的钢板数据与之前所有决策时刻所选择的钢板编号, r 初值为 0。当分拣 a 所对应的钢板时,清框次数加 1 则 r 减 1。记决策策

略为 $\pi, p = \pi(s, a)$ 是 a 在 s 下被选择的概率,一个 s 下所有 a 的概率和为 1。

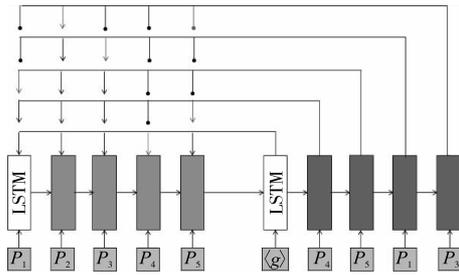


图 3 Pointer network 结构图
Fig. 3 Structure of pointer network

Actor 的损失函数计算如式(4)所示, n 为训练时样本的数量, $A_{\pi}(s, a)$ 为优势函数, 其定义见式(6)。式(7)定义了策略 π 下的动作价值函数 $Q_{\pi}(s, a)$, 为当前状态 s 下采取动作 a 后获得的奖励 r 加上 γ 倍衰减的下一个状态 s' 的值函数 $V_{\pi}(s')$ 。而式(8)则定义了状态的值函数, 由此可知 $V_{\pi}(s)$ 为 $Q_{\pi}(s)$ 的期望值。Critic 的损失函数计算如式(5)所示, 括号内的项为时间差分误差, 对误差求平方和即为 Critic 的损失值。

$$L_{\pi} = -\frac{1}{n} \sum_{i=1}^n A_{\pi}(s, a) \log \pi(s, a) \quad (4)$$

$$L_{critic} = \frac{1}{n} \sum_{i=1}^n (r + \gamma V_{\pi}(s') - V_{\pi}(s))^2 \quad (5)$$

$$A_{\pi}(s, a) = Q_{\pi}(s, a) - V_{\pi}(s) = r + \gamma V_{\pi}(s') - V_{\pi}(s) \quad (6)$$

$$Q_{\pi}(s, a) = r_s^a + \gamma V_{\pi}(s') \quad (7)$$

$$V_{\pi}(s) = E_{\pi}[r + \gamma V_{\pi}(s')] \quad (8)$$

2.2 带精英保留策略的 GA

使用 GA 对钢板分拣顺序进行优化时, 首先需要将问题的解编码为 GA 种群中的个体。在随机初始化种群时, 任意挑选某一顺序作为个体的初始值, 如 $[P_1 \rightarrow P_2 \rightarrow \dots \rightarrow P_N]$, 其中 P 代表对应钢板的 D 维向量, 适应度函数为 $F(x)$ 。

选择算子: 选择算子使用轮盘赌操作, 以个体适应度值的倒数作为其被选择的概率, 选出一批 $F(x)$ 较小的个体作为进化的父体与母体。

交叉算子: 首先随机选择起始下标 $start$ 与终点下标 end , 将父体 $x_f[start: end]$ 中的顺序片段保留传递至下一代。同时将母体 x_m 中所包含的 $x_f[start: end]$ 基因全部删除, 并在 x_m 的 $start$ 位置处插入 $x_f[start: end]$ 所含的顺序片段, 从而形成新的钢板顺序。

变异算子: 使用单点交叉变异, 在染色体 x 中随机选择两个不同的位置 i 和 j , 并将对应位置上的钢板交换。

本文在 GA 中引入了 NSGA-II 的精英保留策略。将父代种群与搜索产生的子代种群合并, 根据个体的适应度进行排序, 选取前 $S \in N^*$ 个个体作为下一代种群。精英保留策略的引入有利于保持优良的个体, 提高种群的整体进化水平。

在给出了上述的遗传算子定义后, GA 在钢板分拣顺序优化上的步骤如算法 1 所示。在随机生成初始种群后, 不断的执行选择、交叉、变异等遗传操作, 直到执行完最大迭代周期数或者最优适应度值达到收敛状态(其中收敛状态定义为当种群中的最优个体连续 5 次不再优化), 则结束搜索过程, 否则继续执行遗传进化操作。算法对应的流程如图 4 所示。

2.3 整体算法运行步骤

算法 1 演化式算法在顺序优化上的流程

输入: 变异概率 R_m ; 种群数量 S 。

输出: 最终种群中适应度值最小的个体 x_{best} 。

a) 使用训练完毕的决策网络(训练部分见 2.1 节)初始化种群中的个体。

- b) 随机初始化种群中剩余的 $S-1$ 个个体。
- c) 对种群中的所有个体计算其适应度值。
- d) $iteration = 0$ 。
- e) while $iteration <$ 最大迭代周期 do
- f) 选择算子(见 2.2 节)。
- g) 交叉算子(见 2.2 节), 产生新个体 New_x 。
- h) 加入 New_x 到种群, 同时删除适应度值最大的个体。
- i) if $R_m >$ $random()$ then
- j) 变异算子(见 2.2 节), 产生新个体 New_x' 。
- k) 加入 New_x' 到种群, 同时删除适应度值最大的个体。
- l) end if
- m) $iteration + = 1$ 。
- n) if 状态是否收敛 then
- o) Break;
- p) end if
- q) end while
- r) 输出 x_{best} 。

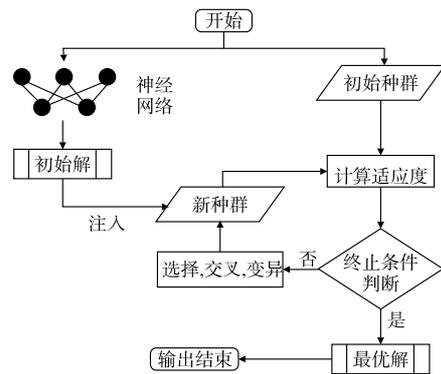


图 4 算法流程图
Fig. 4 Flowchart of the algorithm

3 实验分析与性能评价

3.1 实验细节

实验数据来源于某工程机械制造企业的自动化产线。产线上的零件种类共有 226 种, 即 $D = 226$ 。每块钢板从零件库中挑选若干零件的形状进行激光切割与分拣。本文对某一时间段内的钢板数据进行了收集, 整理了共 70 份真实的钢板数据, 包含了所有种类的零件。同时, 为了对实验数据做进一步的补充, 通过随机抽取零件, 生成了一批模拟钢板数据(100 份)。通过同时使用模拟数据与真实数据, 验证提出算法的泛化性, 相关数据的获取链接为: <https://github.com/tiantianhuanle/Job-Scheduling>。本次实验环境为: Intel Core i7-8565U@1.8 GHz 处理器, 8 GB 内存, Windows 10 操作系统, Python 3.8 编程。

3.2 不同方法的优化效果

表 2 和 3 分别展示了不同方法在求解不同规模的真实验证集与模拟验证集(通过随机抽取 N 个钢板组成一条样本数据)上的效果, 样本数据的 N 分别取 10, 20, 30, 50, 60, 数据量均为 2 048 条。求解指标 $F(x)$ 的值越小越好, 相关方法的参数设置如表 4 所示, 其中演化式算法中的参数与 GA 及 Transformer + RL 中的参数设置相同。

以表 2 为例, 当验证数据的 $N = 10$ 时, 随机顺序(目前产线的方法, 作为基准方法)产生的平均清框次数为 6.91, 而演化式算法给出解的平均清框次数为最优的 5.74, 表中的百分数为对应方法相比基准方法的下降幅度。同样的情况也发生在其他输入尺寸的验证数据集上, 从表中可以看出两个趋势:

a) PointerNet 给出解的平均清框值均高于 Transformer, 说明 PointerNet 的优化性能要弱于 Transformer。

b) 当验证数据的输入尺寸较小时, 元启发式算法同样取

得了较好的表现;而当数据输入尺寸逐步增大 (>10) 时, DRL 开始优于元启发式算法。但演化式算法在所有尺寸上均取得了最优表现, 这表明 GA 中引入 DRL 在实际的大规模 OSSI 求解上具有优势。

表 2 不同方法在不同尺寸的真实验证集下给出的清框频率值
Tab.2 Object values given by different methods in different size on real validation dataset

方法	尺寸				
	10	20	30	50	60
随机顺序	6.91	14.21	21.54	35.57	42.94
模拟退火	6.56	13.78	21.17	35.41	42.19
GA	5.06% ↓	3.03% ↓	1.72% ↓	0.45% ↓	1.75% ↓
	6.08	13.05	20.34	34.43	41.71
PointerNet + RL	12.01% ↓	8.16% ↓	5.57% ↓	3.20% ↓	2.86% ↓
	6.51	13.34	20.59	34.47	41.81
Transformer + RL	5.79% ↓	6.12% ↓	4.41% ↓	3.09% ↓	2.63% ↓
	6.29	12.89	19.78	32.82	39.64
演化式算法	8.97% ↓	9.29% ↓	8.17% ↓	7.73% ↓	7.69% ↓
	5.74	11.71	18.80	32.56	39.56
	16.93% ↓	17.59% ↓	12.72% ↓	8.46% ↓	7.87% ↓

表 3 不同方法在不同尺寸的模拟验证集下给出的清框频率值
Tab.3 Object values given by different methods in different size on simulated validation dataset

方法	尺寸				
	10	20	30	50	60
随机顺序	5.53	11.25	16.94	28.34	34.19
模拟退火	5.20	10.86	16.56	28.07	33.47
GA	5.97% ↓	3.47% ↓	2.24% ↓	0.95% ↓	2.11% ↓
	4.76	10.19	15.78	27.15	32.82
PointerNet + RL	13.92% ↓	9.42% ↓	6.85% ↓	4.20% ↓	4.01% ↓
	4.98	10.19	15.38	25.83	31.03
Transformer + RL	9.95% ↓	9.42% ↓	9.21% ↓	8.86% ↓	9.24% ↓
	4.92	10.04	15.11	25.41	30.58
演化式算法	11.03% ↓	10.76% ↓	10.80% ↓	10.34% ↓	10.56% ↓
	4.38	9.39	14.69	25.20	30.43
	20.80% ↓	16.53% ↓	13.28% ↓	11.08% ↓	11.00% ↓

表 4 不同方法参数设置表
Tab.4 Parameter setting of different methods

方法	参数	大小
GA	种群数	50
	变异概率	0.1
	迭代周期	30
模拟退火	初始温度	20
	最低温度	1
	迭代周期	30
	同一温度重复次数	5
	训练周期	350
PointerNet + RL	Batch_size	512
	嵌入维度 M	128
	词典长度 D	226
	LSTM 单元	1 层
	隐藏层维度	128
	训练周期	350
	Batch_size	512
Transformer + RL	嵌入维度 M	128
	词典长度 D	226
	Transformer 单元	1 层
	隐藏层维度	128

表 3 模拟数据集中亦与之类似, 结合了 Transformer + RL 与 GA 的演化式算法在所有的 N 上均有最佳的表现, 尤其是验证数据的规模逐步增大 (>30) 时, 元启发式算法的表现开始恶化, 而演化式算法依然鲁棒, 证明了 DRL 初始化 GA 种群在寻优过程中所带来的优势。此外, 演化式算法相比单独的 Trans-

former + RL 具有更强的优化表现, 说明 GA 也进一步提升了演化式算法的优化性能, 在演化式算法中两个模块相辅相成。

3.3 消融实验

为了探究演化式算法中的 DRL 部分, 在图 5(a) 中分别展示了深度网络 PointerNet 与 Transformer 在 RL 训练中的 loss 下降曲线。从图中可以看出 PointerNet + RL 在初始阶段 loss 值较为波动, 而在中后期趋于收敛。Transformer + RL 的这一趋势则更加明显, 在 2k 步左右其 loss 达到了最大值; 5k 步到 12.5k 步间 loss 下降明显, 之后趋于收敛。图 5(b) 中分别展示了 PointerNet 与 Transformer 在训练数据上的平均 $F(x)$ 值变化趋势。可以看到随着训练的进行, 两个网络的平均 $F(x)$ 值均在不断优化。图 5(c) 中分别展示了 PointerNet 与 Transformer 在同一真实验证集数据 ($N = 10$) 上的平均 $F(x)$ 值变化曲线。由图可知, 随着网络的优化, 两个网络的平均 $F(x)$ 值均在不断优化, PointerNet 的平均 $F(x)$ 值起始时为 7.08, 最终稳定在 6.51 左右; 而 Transformer 起始时为 6.8, 最终稳定在 6.29 左右。对比之下, 同样说明了 Transformer 网络拥有更加优秀的表现。

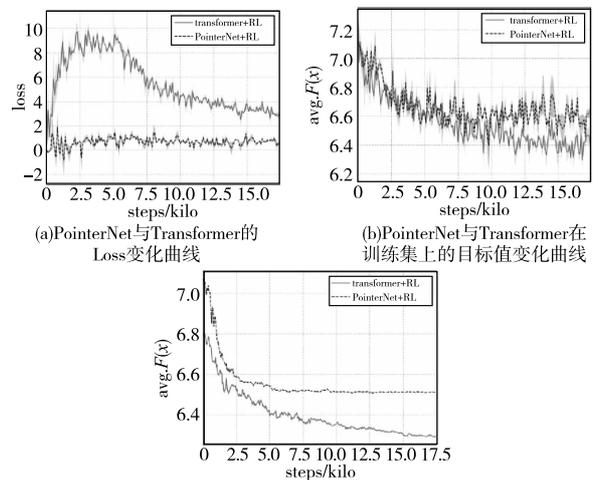


图 5 DRL 训练中各项指标变化曲线
Fig.5 Change curve of various indexes in DRL training

综上所述, 从图中三个指标的变化中可以看出, 随着训练的进行, DRL 的决策网络在不断地收敛, 而在验证集上的表现也验证了决策网络的泛化能力。

为了研究演化式算法中的 GA 部分, 在图 6 中绘制了 GA 中变异概率与种群数量两参数变化时, GA 在真实验证集 ($N = 10$) 上平均 $F(x)$ 值的变化曲线。变异概率取五组, 分别为 0.1, 0.3, 0.5, 0.7, 0.9。从图中可以看出, 随着种群数量的增大, 平均 $F(x)$ 值在不断地降低。而当种群的数量大于 10 以后, 红色曲线 (变异概率为 0.1) 基本上取得了最优的平均 $F(x)$ 值 (见电子版), 说明 OSSI 求解时 GA 的变异概率并不需要过大, 设为 0.1 附近较为合适。

此外, 从图 6 中可知当种群数量固定时, 由变异概率的改变引起的目标值变化幅度不超过 0.05。而当变异概率固定时, 由种群数量的改变引起的目标值变化幅度远超 0.05, 这说明了相比变异概率, 种群数量对寻优过程具有更强的影响力, 在超参数的设置中应更加关注种群数量。

为了比较不同的初始化方法对后续 GA 搜索过程的影响, 本文亦对四种初始化种群的策略进行了研究。这四种策略分别为随机初始化种群、使用训练完毕的 PointerNet 初始化种群、使用训练完毕的 Transformer 初始化种群、以及使用 SA 算法初始化种群。图 7 展示了 GA 在四种初始化策略下在真实与模拟验证集 ($N = 10$) 上的优化表现与平均时间开销。折线

代表不同策略的优化表现。由图可知,传统 GA 的平均 $F(x)$ 值为 6.081 与 4.76,使用训练完毕的 Transformer 初始化 GA 种群策略达到了最优的平均 $F(x)$ 值 5.74 与 4.38;相比之下,提升了 0.341 与 0.38 次清框(约 5.6%与 8.0%)。

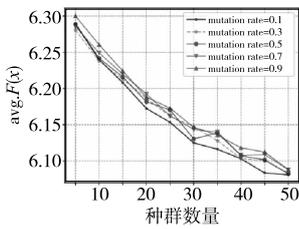


图 6 GA 中不同种群与变异概率时目标值变化图
Fig.6 Curve of object value with different population size and mutation probability in GA

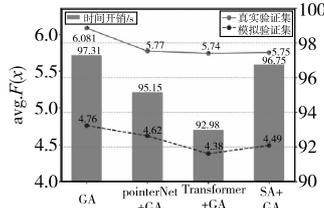


图 7 四种策略在验证集上的优化表现与时间开销
Fig.7 Performance of the four strategies on the real validation set

随机初始化的 GA 在验证集上的整体平均时间开销为 97.31 s,而使用训练完毕的 DRL 初始化种群后 GA 在整体时间开销上具有更好的表现,其中使用 Transformer 初始化种群后 GA 在时间开销上最低,为 92.98 s。而使用 SA 算法初始化 GA 种群也能产生次优表现,但此时 SA 算法的时间开销达到了 1 226 s,极为耗时。

这说明了当 GA 中的初始种群拥有较好的解时有助于加速搜索过程的收敛,而不同初始化策略的对比实验再次证明了 OSSI 求解时使用 DRL 初始化种群个体能带来最大的清框值减少。

3.4 计算效率

表 5 展示了单独 GA 与演化式算法在求解不同规模验证数据集上的整体时间开销(运行 10 次取平均),单位为秒。由表中的数据可知,在钢板集的尺寸较大(大于 50)时,演化式算法相比于传统的 GA,其时间开销优势明显,这对于大规模生产中实时性要求高的场景而言极具吸引力。究其原因,当求解数据规模较小时,GA 在较短的时间内亦可寻得质量不错的解;然而,当求解数据的规模逐步增大时,由于维度恶化,其所需的时间开销亦明显增大;而演化式算法由于 DRL 的加持,使得 GA 专注于开发阶段,反而降低了整体的时间开销。

表 5 GA 与演化式算法在不同尺寸数据上的时间开销
Tab.5 Time cost of GA and evolutionary algorithm on data with different size

方法	尺寸				
	10	20	30	50	60
GA	97.31	187.66	300.98	559.61	723.38
演化式算法	104.66	203.51	312.13	497.92	642.93
	-7.46%	-8.45%	-3.70%	11.02%	11.12%

4 公开数据集测试

为进一步验证演化式算法的泛化性,文中在 TSP 公开数据集^[17](每个节点 10 万条测试数据)上对其进行了测试,并将其与其他较为先进的算法进行了对比分析,测试结果如表 6 所示,方法的参数设置见相关文献与表 4。

表中的值为平均的路径长度,括号内为求解的时间开销;其中开源求解器 OR-Tools 运行时间过长,未纳入统计。从表中可以看出演化式算法仍取得了最优的结果,尤其是在 100 个节点上,这进一步验证了本文算法的优化能力;但相比单独的 DRL 方法(PointerNet + RL 或 Transformer + RL),由于后续 GA 的继续搜索,演化式算法的运行时间变长,这是以时间复杂度的增加为代价换取优化质量的提升。

表 6 不同方法在不同尺寸的 TSP 数据集上的表现
Tab.6 Performance of different methods on TSP data sets of different sizes

方法	节点数		
	20	50	100
LKH3 ^[22]	3.84(18 s)	5.70(5 m)	7.76(21 m)
OR-Tools ^[23]	3.85	5.80	7.99
Gurobi ^[24]	3.84(7 s)	5.70(2 m)	7.76(17 m)
PointerNet + RL	3.98(0.6 s)	5.95(3.8 s)	8.30(5 s)
Transformer + RL	3.84(0.5 s)	5.73(2.6 s)	7.94(3.8 s)
演化式算法	3.84(12 s)	5.70(43 s)	7.75(2 m)

5 结束语

对分拣作业进行调度有助于优化企业生产流程,提高生产效率。本文针对机械制造产线中的钢板分拣优化提出了一套数据表示方法与一种演化式算法。数据表示方法通过类比词袋模型对钢板数据进行抽象表示;演化式算法则通过 DRL 与 GA 联合优化目标函数。从上述的实验中可得如下结论:a)本文提出的数据表示方法对 OISS 问题是有效的;b)在结论 b)的基础上,本文提出的演化式算法在真实的 OISS 验证集上将平均清框次数最高降低了 16.93%,证明了算法的优良性能;c)本文提出的演化式算法在时间开销上相比传统的 GA 也具有一定优势。本文研究的局限在于神经网络需要完成离线训练后才能进行使用,而这一步耗时过长,下一步将重点研究缩短网络的训练耗时;同时将针对生产线的上下游进行分析,构建场景的数字孪生系统,整合调度策略,实现生产效率最大化。

参考文献:

- [1] 张贻弓,吴耀华.可合流的自动分拣系统订单排序优化[J].山东大学学报:工学版,2008,15(5):67-71.(Zhang Yigong, Wu Yao-hua. Order sequencing optimization of confluent automatic sorting system[J]. Journal of Shandong University: Engineering Edition, 2008,15(5):67-71.)
- [2] Karlin A R, Klein N, Gharan S O. A (slightly) improved approximation algorithm for metric TSP[C]//Proc of the 53rd Annual ACM SIGACT Symposium on Theory of Computing. 2021:32-45.
- [3] Wang Pengcheng, Peng Wei, Zhang Wenxin, et al. Joint channel and power allocation algorithm for flying Ad hoc networks based on Bayesian optimization [C]//Proc of International Conference on Advanced Information Networking and Applications. Cham: Springer, 2021:320-331.
- [4] Duan Chen, Peng Wei, Wang Baosheng. CO-BPG: a centralized optimizer for routing in BGP-based data center networks[C]//Proc of International Conference on Advanced Information Networking and Applications. Cham: Springer,2021:307-319.
- [5] 江璞玉,刘均,周奇,等.大规模黑箱优化问题元启发式求解方法研究进展[J].中国舰船研究,2021,16(4):1-18.(Jiang Puyu, Liu Jun, Zhou Qi, et al. Research progress of meta heuristic solution methods for large-scale black box optimization problems[J]. Chinese Ship Research,2021,16(4):1-18.)
- [6] 张冠宇,尚文利,张博文,等.一种结合遗传算法的工控协议模糊测试方法[J].计算机应用研究,2021,38(3):680-684.(Zhang Guanyu, Shang Wenli, Zhang Bowen, et al. A fuzzy test method for industrial control protocol combined with genetic algorithm[J]. Application Research of Computers,2021,38(3):680-684.)
- [7] 丁旻钧天,曹怀虎.融合模拟退火的随机森林房价评估算法[J].计算机应用研究,2020,37(3):784-788.(Ding Minjuntian, Cao Huaihu. Stochastic forest house price evaluation algorithm combined with simulated annealing[J]. Application Research of Computers, 2020,37(3):784-788.)
- [8] 王守金,程小桐,宋晓宇.引入高斯分布缩放因子的人工蜂群算法[J].计算机工程与设计,2019,40(9):2507-2512,2536.(Wang Shoujin, Chen Xiaotong, Song Xiaoyu. Multi colony artificial bee colony algorithm with variable sub population size[J]. Computer Engineering & Design,2019,40(9):2507-2512,2536.)

- [12] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. *Knowledge-Based Systems*, 2016, 96(3):120-133.
- [13] Yang Xinshu. Firefly algorithms for multimodal optimization [C]//Proc of International Symposium on Stochastic Algorithms. Berlin: Springer, 2009:169-178.
- [14] Kennedy J, Eberhart R. particle swarm optimization [C]//Proc of International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 1995.
- [15] 艾欣,李一铮,王坤宇,等.基于混沌模拟退火粒子群优化算法的电动汽车充电站选址与定容[J]. *电力自动化设备*, 2018, 38(9):9-14. (Ai Xin, Li Yizheng, Wang Kunyu, et al. Locating and sizing of electric vehicle charging station based on chaotic simulated annealing particle swarm optimization algorithm [J]. *Electric Power Automation Equipment*, 2018, 38(9):9-14.)
- [16] 麻秀范,王皓,李颖,等.基于变权 Voronoi 图和混合粒子群算法的电动汽车充电站规划[J]. *电工技术学报*, 2017, 32(19):160-169. (Ma Xiufan, Wang Hao, Li Ying, et al. Optimal planning of charging stations for electric vehicle based on weight-changed voronoi diagram and hybrid particle swarm optimization algorithm [J]. *Transactions of China Electrotechnical Society*, 2017, 32(19):160-169.)
- [17] 陈婷,卫志农,吴霜,等.考虑电动汽车充电站选址定容的配电网规划[J]. *电力系统及其自动化学报*, 2013, 25(3):1-7. (Chen Ting, Wei Zhinong, Wu Shuang, et al. Distribution network planning by considering siting and sizing of electric vehicle charging stations [J]. *Proceedings of the CSU-EPSSA*, 2013, 25(3):1-7.)
- [18] 官嘉玉,江修波,刘丽军.考虑电动汽车随机充电行为及低碳效益的充电站规划[J]. *电力科学与技术学报*, 2019, 34(4):93-100. (Guan Jiayu, Jiang Xiubo, Liu Lijun. Optimal allocation of charging stations considering stochastic charging behavior and low carbon efficiency of electric vehicles [J]. *Journal of Electric Power Science and Technology*, 2019, 34(4):93-100.)
- [19] 程宏波,肖永乐,王勋,等.考虑低碳收益的电动汽车充电站选址规划[J]. *中国电力*, 2016, 49(7):118-121, 172. (Cheng Hongbo, Xiao Yongle, Wang Xun, et al. Site Planning of EV Charging Station by Considering Low Carbon Emission [J]. *Electric Power*, 2016, 49(7):118-121, 172.)
- [20] 段豪翔,吕林,向月.计及分时充电电价激励的电动汽车充电站与配电网协同规划[J]. *电力系统及其自动化学报*, 2017, 29(1):103-110. (Duan Haoxiang, Lyu Lin, Xiang Yue. Coordinated planning for electric vehicle charging station and distribution network considering time-of-use charging price [J]. *Proceedings of the CSU-EPSSA*, 2017, 29(1):103-110.)
- [21] 所丽,唐巍,白牧可,等.考虑削峰填谷的配电网集中型充电站选址定容规划[J]. *中国电机工程学报*, 2014, 34(7):1052-1060. (Suo Li, Tang Wei, Bai Muke, et al. Locating and sizing of centralized charging stations in distribution network considering load shifting [J]. *Proceedings of the CSEE*, 2014, 34(7):1052-1060.)
- [22] 陈光,毛召磊,李济沅,等.计及碳排放的电动汽车充电站多目标规划[J]. *电力系统自动化*, 2014, 38(17):49-53, 136. (Chen Guang, Mao Zhaolei, Li Jiyuan, et al. Multi-objective optimal planning of electric vehicle charging stations considering carbon emission [J]. *Automation of Electric Power Systems*, 2014(17):49-53, 136.)
- [23] 刘柏良,黄学良,李军,等.含分布式电源及电动汽车充电站的配电网多目标规划研究[J]. *电网技术*, 2015, 39(2):450-456. (Liu Boliang, Huang Xueliang, Li Jun, et al. Multi-objective planning of distribution network containing distributed generation and electric vehicle charging stations [J]. *Power System Technology*, 2015, 39(2):450-456.)
- [24] 侯萍,王洪涛,张浩,等.用于组织和产品碳足迹的中国电力温室气体排放因子[J]. *中国环境科学*, 2012, 32(6):961-967. (Hou Ping, Wang Hongtao, Zhang Hao, et al. GreenHouse gas emission factors of Chinese power grids for organization and product carbon footprint [J]. *China Environmental Science*, 2012, 32(6):961-967.)
- [25] 电动汽车充电基础设施发展指南(2015-2020年)[J]. *汽车与社会*, 2015(35):22-31. (Development guidelines for electric vehicle charging infrastructure (2015 - 2020) [J]. *Auto & Society*, 2015(35):22-31.)
- [26] 国家能源局.国家能源局关于2017年度全国电力价格情况监管通报(风电部分) [R/OL]. (2018-10-09). http://www.nea.gov.cn/2018-10/09/c_137519800.htm. (National Energy Administration. The National Energy Administration's 2017 national electricity price regulatory bulletin (wind power part) [R/OL]. (2018-10-09). http://www.nea.gov.cn/2018-10/09/c_137519800.htm.)
- (上接第 页)
- [9] 姚远远,叶春明.求解作业车间调度问题的改进混合灰狼优化算法[J]. *计算机应用研究*, 2018, 35(5):1310-1314. (Yao Yuanyuan, Ye Chunming. Improved hybrid grey wolf optimization algorithm for job shop scheduling problem [J]. *Application Research of Computers*, 2018, 35(5):1310-1314.)
- [10] Pettinger J E, Everson R M. Controlling genetic algorithms with reinforcement learning [C]//Proc of the 4th Annual Conference on Genetic and Evolutionary Computation. 2002:692-692.
- [11] Zhu Min, Yi Sheng, Yang Chunling, et al. Research on RLGA-based Hardware Evolution Optimization Technology [C]//Proc of the 15th IEEE Conference on Industrial Electronics and Applications. Piscataway, NJ: IEEE Press, 2020:188-193.
- [12] Chen Qiong, Huang Mengxing, Xu Qiannan, et al. Reinforcement learning-based genetic algorithm in optimizing multidimensional data discretization scheme [J]. *Mathematical Problems in Engineering*, 2020, 2020(2):article ID 1698323.
- [13] Bora T C, Mariani V C, dos Santos Coelho L. Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm [J]. *Applied Thermal Engineering*, 2019, 146:688-700.
- [14] Lu Zhichao, Whalen I, Boddeti V, et al. Nsga-net: neural architecture search using multi-objective genetic algorithm [C]//Proc of Genetic and Evolutionary Computation Conference. 2019:419-427.
- [15] Vinyals O, Fortunato M, Jaitly N. Pointer networks [EB/OL]. (2015) [2021-10-25]. <https://arxiv.org/abs/1506.03134>.
- [16] Ma Qiang, Ge Suwen, He Danyang, et al. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning [EB/OL]. (2019) [2021-10-25]. <https://arxiv.org/abs/1911.04936>.
- [17] Kool W, Van Hoof H, Welling M. Attention, learn to solve routing problems! [EB/OL]. (2018) [2021-10-25]. <https://arxiv.org/abs/1803.08475>
- [18] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(2):182-197.
- [19] 缪学勤.智能工厂与装备制造业转型升级[J]. *自动化仪表*, 2014, 35(3):1-6. (Miao Xueqin. Transformation and upgrading of intelligent factory and equipment manufacturing industry [J]. *Automation Instrumentation*, 2014, 35(3):1-6.)
- [20] Dehkordi R A, Khosravi H. Vehicle type recognition based on dimension estimation and bag of word classification [J]. *Journal of AI and Data Mining*, 2020, 8(3):427-438.
- [21] Xi Lei, Wu Junnan, Xu Yanchun, et al. Automatic generation control based on multiple neural networks with actor-critic strategy [J]. *IEEE Trans on Neural Networks and Learning Systems*, 2020, 32(6):2483-2493.
- [22] Helsgaun K. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems [EB/OL]. (2017) [2021-10-25]. http://akira.ruc.dk/~keld/research/LKH/LKH-3_REPORT.pdf.
- [23] Crainic T G, Roy J. OR tools for tactical freight transportation planning [J]. *European Journal of Operational Research*, 1988, 33(3):290-297.
- [24] Gurobi optimizer reference manual [EB/OL]. (2018) [2021-10-25]. <http://www.gurobi.com>.