

基于改进变邻域搜索的数控裁床路径优化



廖义辉 杨恩君 刘安东 俞立
浙江工业大学信息工程学院 杭州 310023
(yhliaochn@hotmail.com)

摘要 针对数控加工中平面多轮廓样片的空行程路径优化问题,文中提出了一种基于改进变邻域搜索(Modified Variable Neighborhood Search, MVNS)的元启发式方法。首先,将空行程路径优化问题转化为一类广义旅行商问题(Generalized Traveling Salesman Problem, GTSP)。其次,针对 GTSP 中的顺序序列问题,对传统的变邻域搜索中的局部搜索和抖动阶段进行了改进。在局部搜索中,设计了基于 2-opt 和插入算子的邻域结构,同时采用了一种增量计算方法,提高了求解质量和搜索效率;在抖动阶段中,结合遗传算法设计了分块和重组等算子,避免了过早地陷入局部最优。然后,利用禁忌搜索混合动态规划(Tabu Search with Dynamic Programming, TS-DP)算法排除重复的裁剪序列,并确定入刀点位置。最后,通过应用实例和对比实验,从求解精度和运行时间角度检验所提算法的有效性。对于服装样片的测试,所提算法相比服装 CAD 的精度值提升了 51% 以上,平均运行时间为 9.3s;对于 TSP 的测试,所提算法在多数算例上达到或超过对比算法的精度值;对于 GTSP 的测试,虽然所提算法在少数算例上达到或超过对比算法的精度值,但是平均误差与对比算法的差距不超过 1%,并且平均运行时间对比算法缩短了 73.7%。实验结果表明了该算法能同时兼顾求解精度和运行时间,具有一定的应用价值。

关键词: 数控裁床;空行程路径;广义旅行商问题;变邻域搜索;禁忌搜索

中图分类号 TP301.6

Path Optimization in CNC Cutting Machine Based on Modified Variable Neighborhood Search

LIAO Yi-hui, YANG En-jun, LIU An-dong and YU Li

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Abstract To solve the optimization problem of non-cutting path for multi-contour segments, a modified variable neighborhood search (MVNS) based metaheuristic algorithm is proposed for computer numerical control (CNC) processing systems. Firstly, this paper transfers the optimization problem into a generalized traveling salesman problem (GTSP). Secondly, for the sequential sequence problems in GTSP, it modifies the local search and shaking procedure in traditional variable neighborhood search. A 2-opt with insertion operator of neighborhood structure and an incremental calculation method are proposed for local search, which improve the solution quality and search efficiency. Combining genetic algorithm, some operators such as partition and reorganization are designed for shaking procedure, which avoid to prematurely fall into local optimum. Furthermore, a Tabu search with dynamic programming (TS-DP) algorithm is used to eliminate duplicate cutting sequences and determine the starting point of each segment. Finally, through the application examples and comparative experiments, the effectiveness of the proposed algorithm is tested from the perspective of solution accuracy and running time. In the test of cloth segments, the proposed algorithm improves the results of garment CAD about more than 51%, and the average running time is 9.3 s. In the test of TSP, the proposed algorithm achieves or exceeds the accuracy value of the comparison algorithm in most instances. In the test of GTSP, although the proposed algorithm achieves or exceeds the accuracy of the comparison algorithm in some instances, the difference of the average error between the proposed algorithm and the comparison algorithm is within 1%, and average running time of the proposed algorithm is 73.7% shorter than the comparison algorithm. Thus, the test results demonstrate that the proposed algorithm can simultaneously consider the solution accuracy and running time, which has a certain application value.

Keywords Computer numerical control cutting machine, Non-cutting path, Generalized traveling salesman problem, Variable neighborhood search, Tabu search

到稿日期:2019-08-08 返修日期:2019-10-11 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目: NSFC-浙江省两化融合联合基金(U1709213);浙江省自然科学基金(LY17F030019)

This work was supported by the NSFC-Zhejiang Joint Fund for the Intergration of Industrialization and Informatization (U1709213) and Natural Science Foundation of Zhejiang Province, China (LY17F030019).

通信作者:刘安东(lad@zjut.edu.cn)

1 引言

数控裁床是一种裁剪平面多轮廓样片的数控系统,相比传统的手工裁剪,极大地提高了制品的生产效率和原材料利用率^[1-2]。裁刀经过的总路径由样片的裁剪有效行程和不同样片之间的走刀空行程构成,其中有效行程为每个样片轮廓长度的总和,是固定的;而走刀空行程则随着裁刀走过的样片序列、样片的入刀点位置不同而发生变化。因此,裁剪路径优化问题即为优化空行程问题。通过对空行程的优化可以减短空行程长度,从而有效地缩短走刀时间,提高生产效率。

空行程的长短取决于样片裁剪序列和每个样片的入刀点,是一个两级优化问题。因此,该问题可转化为一类广义旅行商问题,从而可采用智能算法进行求解,如模拟退火算法^[3]、遗传算法^[4]和禁忌搜索算法^[5]等。近年来,为了提高GTSP中求解算法的精度和运行时间,不少学者对传统的智能算法进行了改进。Ardalan等^[6]提出了一种帝国发展机制,解决了传统帝国竞争算法(Imperialist Competitive Algorithm, ICA)易陷入局部最优的不足。Wang等^[7]提出了一种改进的量子进化算法(Improved Quantum Evolutionary Algorithm, IQEA),可以有效地提高解的精度。Smith等^[8]提出了一种自适应大邻域搜索算法,加快了传统大邻域搜索算法的运行时间。同时,一些学者提出了新颖的算法,如线性时间算法^[9]、空间驱动算法^[10]和人工蜂群算法^[11]等,求解效果较好。上述算法在多个算例中均表现出了一定的优越性,但是离最优值仍然存在一些差距,求解精度和运行时间有提高的空间。

目前,在不同类型的优化问题中变邻域搜索(Variable Neighborhood Search, VNS)算法的求解精度与运行时间均取得了不错的效果。VNS最早由Hansen等^[12]提出,是一种有效的局部搜索算法。Xu等^[13]利用VNS求解路径规划问题,运行时间较短。Ranjbar等^[14]通过VNS解决人才调度问题,优化效果较好。Palubeckis等^[15]将VNS和模拟退火算法相结合,在大规模问题中求解精度较高。此外,VNS已经成功地应用于求解各种类型的TSP^[16-18]、生产规划问题^[19]、离散有序中值问题^[20]等。虽然传统的VNS的求解精度相对较高,但是也存在易陷入局部最优、在大规模问题中运行时间较长等问题。

为了同时兼顾求解精度和运行时间,本文提出了一种基于改进变邻域搜索的元启发式方法,用于求解空行程路径优化问题。首先建立数学模型将数控裁床路径优化问题转化为一类广义旅行商问题,通过MVNS算法求解样片的裁剪序列;然后采用TS-DP算法得到样片的入刀点。本文通过服装样片应用实例和标准算例验证了所提算法能有效地优化样片的空行程,且与IQEA等算法相比具有一定的优越性。

2 问题描述

待裁剪的样片排样图由若干个独立样片构成,样片外轮廓的形状并不规则,包括直线、圆弧和曲线。确定样片的排样图后,每个样片外轮廓的特征点可以用几何信息代替^[4]。数

控裁床裁剪时,裁刀由裁床原点出发,选择一个样片的入刀点,如图1中样片上的小圆点所示。裁刀裁完一个样片后,回到入刀点,即完成了一个样片的裁剪。此时,该样片的入刀点即变为出刀点,裁刀从该点移动到另外一个样片,直到完成所有样片的裁剪,最后回到裁床原点,图1虚线所示路径为一个完整的空行程。

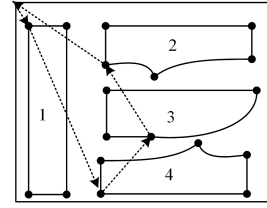


图1 空行程路径示意图

Fig. 1 Diagram of non-cutting path

学上把上述问题归结为一类GTSP^[6],为此,本文定义如下6个变量: m 表示一个数控裁床需要裁剪的样片总数; V_i 表示第 i 个样片的序号; r 表示数控裁床原点; r_i 表示第 i 个样片的入刀点; r_{ij} 表示第 i 个样片的第 j 个轮廓点; n_i 表示第 i 个样片的轮廓点总数。因此,裁床样片空行程数学模型表述如下:假定裁剪序列为 $V = \{r, V_1, V_2, \dots, V_m\}$, V 对应的入刀点集为 $R = \{r, r_1, r_2, \dots, r_m\}$,则空行程长度为^[7]:

$$L(V) = d(r, r_1) + \sum_{i=1}^{m-1} d(r_i, r_{i+1}) + d(r_m, r) \quad (1)$$

其中, $d(r, r_1)$ 和 $d(r_m, r)$ 分别表示裁床原点到第1个样片和最后1个样片入刀点的距离; $\sum_{i=1}^{m-1} d(r_i, r_{i+1})$ 表示第1个样片的入刀点到第 $m-1$ 个样片的入刀点的总距离; $d(r_i, r_j)$ 表示第 i 个样片入刀点到第 j 个样片入刀点的距离,且 $d(r_i, r_j) = d(r_j, r_i)$ 。

本文的目的是针对平面多轮廓样片的空行程路径优化问题,提出了一种基于改进VNS的元启发式方法,使得式(1)中空行程长度 $L(V)$ 在尽可能短的求解时间内最小化,并获得样片入刀点位置。

$$L_{\min} = \min(L(V)) \quad (2)$$

3 变邻域搜索算法

空行程路径优化问题实际上是优化裁刀的裁剪序列 V 和确定每个样片的入刀点位置 R ,其中样片的裁剪序列对空行程路径的影响更为显著。为此,本文采用MVNS算法优化样片的裁剪序列,再通过TS-DP算法排除重复的裁剪序列,确定每个样片入刀点,最终得到优化后的空行程。

3.1 初始化

算法的解可以用一个二维整数矩阵表示,包含了样片序号和入刀点序号。裁床原点可以看作只含一个轮廓点的样片,则数量为 m 的待裁剪样片用 $0, 1, 2, \dots, m$ 进行编号,每个样片的轮廓点用 $1, 2, \dots, n_i$ 进行编号。因此,该矩阵第1行表示裁剪序列,第2行对应位置表示该样片入刀点在每个样片轮廓点集内的序号。当算法初始化时,裁剪序列和样片入刀点序号均随机产生。以6个样片为例,算法解的表示方式如图2所示。

2	3	0	5	4	1	6
3	5	1	3	2	4	2

图2 解的表示方式

Fig.2 Solution representation

3.2 局部搜索

在传统的局部搜索中,每个邻域结构搜索范围较小,使得邻域解不多,导致求解精度不高。同时,每次搜索时都将重新计算每个邻域解的路径长度,当测试算例规模较大时求解时间代价较大^[16]。因此,为了提高求解精度,本文提出了一种基于2-opt及插入算子的邻域结构和增量计算方法,使得在每个邻域结构中增大对解的搜索范围,并缩短运行时间。

在局部搜索中采用变邻域下降(Variable Neighborhood Descent, VND)框架^[14],通过邻域结构中的算子有序地选取样片序号并对其进行变换。对于一个裁剪序列,2-opt算子随机选择两个样片,翻转这两个样片间的序列;插入算子将被选中的一个样片序号插入到另一个被选中的样片序号后。以2-opt算子构成的第1个邻域结构为例,定义一个随机的裁剪序列 $\beta = \{r, \beta_1, \beta_2, \dots, \beta_m\}$,用整数集可以表示为 $\{0, 1, 2, \dots, m\}$ 。定义该邻域结构为 N_1 ,即 $\{\beta_{(0,1)}, \beta_{(0,2)}, \beta_{(0,3)}, \dots, \beta_{(m-1,m)}\}$,如式(3)所示。其中 $\beta_{(i,j)}$ 表示对 β 中第 i 个序号和第 j 个序号之间的序列片段进行翻转,保持其他序列片段不变,若执行 $\beta_{(i,j)}$ 后的路径长度优于当前最短路径,则更新当前最短路径。传统VNS中邻域结构一般会产生 $(m+1)$ 个邻域解^[16],本文通过式(3)可以产生 $(m+1) \times m/2$ 个邻域解,尽可能增大搜索范围以提高求解质量。针对邻域结构 N_2 中的插入算子,遍历方式与 N_1 相同,仅操作算子不同。

$$\begin{aligned}
 \beta_{(0,1)} &= (\beta_2, r, \beta_1, \beta_3, \dots, \beta_m) \\
 \beta_{(0,2)} &= (\beta_2, \beta_1, r, \dots, \beta_m) \\
 &\dots \\
 \beta_{(m-1,m)} &= (r, \beta_1, \beta_2, \beta_3, \dots, \beta_m, \beta_{m-1})
 \end{aligned}
 \tag{3}$$

然而,因为空行程是一个闭合回路,若将裁剪序列的初始序号设置为固定值如数字0,则图3中的序列A,B均可以转化为序列D。此外,鉴于空行程长度与路径方向无关,序列C可以通过改变路径方向转化为序列D。因此,式(3)必然会产生重复的裁剪序列而造成多余的计算,故本文做了进一步的优化,优化方法如下:在 N_1 中,对重复序列的筛选比较耗时,采取折中方式搜索式(3)中的奇数行或偶数行,从而将范围变为 $(m+1) \times m/4$;在 N_2 中,仅 $\beta_{(i,i+1)}$ 未对原序列做出改变,而其余序列并未重复,故可以不执行 $\beta_{(i,i+1)}$,从而将范围变为 $m \times (m-1)/2, m > 3$ 。因此,优化后的搜索范围仍然比传统VNS的搜索范围要大很多。

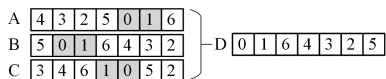


图3 裁剪序列

Fig.3 Cutting sequences

为了进一步缩短运行时间,本文采用了一种增量计算方法^[21],仅计算部分样片间的长度而无需重新计算新路径长度。以2-opt算子为例,假定原序列为 α ,变换后的序列为 β ,

变换位置分别为 i 和 j ,2-opt算子如图4所示。

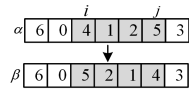


图4 2-opt算子

Fig.4 2-opt operator

显然,图4所示序列中的翻转片段和头尾处的样片序列距离值并未改变,只需计算样片序号 i 和 j 附近的样片入刀点距离,即可获得新路径的距离增量,最终通过距离增量来判断新路径的长短。

3.3 抖动阶段

在传统的抖动阶段,随机选取一个邻域算子对操作对象进行处理,导致改变的程度过小,容易陷入局部最优^[16]。因此,为了避免算法过早陷入局部最优,本文在抖动阶段设计分块、重组操作,增加对解的改变程度;同时,引入遗传算法中的交叉与变异机制,进一步增大全局搜索范围。

首先,将裁剪序列随机地分成 n 块连续的序列片段,然后按块打乱顺序后重新组合在一起。以6个样片为例,分块与重组操作如图5所示。

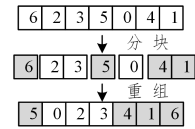


图5 分块与重组

Fig.5 Partition and reorganization

然后,产生一个 $[0, 1]$ 范围内的随机概率 p ,当交叉概率 p_c 小于 p 时,则执行交叉操作。在交叉过程中,会产生一个随机解 x_p ,对当前代获取的解 x_g 与 x_p 进行交叉操作。以6个样片为例,具体操作过程如图6所示。在每次交叉时随机产生一个概率数组 r ,其元素 r_i 是 $[0, 1]$ 范围内的一个随机概率, $i=1, 2, \dots, m+1$ 。通过均匀交叉获取 x_p 中 $r_i < Y$ 的样片序号,其中 Y 为设定的阈值,选中的序号集定义为 α ;通过顺序交叉依次选取 x_g 中与 α 不相同的样片序号,选中的序号集定义为 β ,进而将 α 和 β 重新组合成新的裁剪序列。

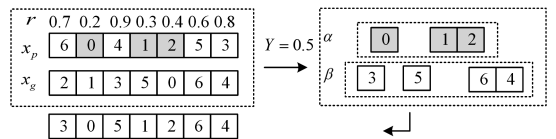


图6 交叉操作

Fig.6 Crossover operation

最后,若 $[0, 1]$ 范围内的一个随机概率 δ 大于变异概率 p_m ,则不执行变异操作;否则对上述操作后的样片序列执行变异操作,变异算子从3-opt和交换算子中随机选择1个。其中,3-opt随机选取两个不相连的样片序列片段,分别将这2个片段间的序列进行翻转;交换算子交换两个样片的样片序号位置。

此外,算法的后期比前期更易陷入局部最优,不同于标准遗传算法的多个个体,MVNS的个体只有一个。为保证前期优秀个体的稳定性,本文对 p_c 和 p_m 进行动态调整,设计了一

种线性递增的概率公式:

$$p_c = p_{c \min} + \frac{g^* (p_{c \max} - p_{c \min})}{z} \quad (4)$$

$$p_m = p_{m \min} + \frac{g^* (p_{m \max} - p_{m \min})}{z} \quad (5)$$

其中, $p_{c \max}$ 和 $p_{c \min}$ 分别表示 p_c 的上下限, $p_{m \max}$ 和 $p_{m \min}$ 分别表示 p_m 的上下限, g 和 z 分别表示 MVNS 算法中的当前迭代数和最大迭代数。

3.4 TS-DP 算法

经过 MVNS 算法优化裁剪序列后, 采用动态规划算法确定每个样片入刀点。由于动态规划是一种精确算法, 求解过程比较耗时, 因此可以先通过禁忌搜索排除相同的裁剪序列以避免重复的计算。因此, 本文设计了一种 TS-DP 算法。

在禁忌搜索中, 首先设置一个禁忌表, 将经 MVNS 算法获得的裁剪序列放入该表, 若下次迭代中获得的裁剪序列已存在于表内, 则不执行动态规划算法, 继续 MVNS 搜索; 否则通过动态规划算法确定每个样片的入刀点。其中对禁忌表内的裁剪序列并不是永远禁止的, 任何一个表内的裁剪序列迭代 T 次后或已重复 t 次后将表内去除。

通过禁忌搜索的筛选, 若允许执行动态规划算法, 则其示意图如图 7 所示。图 7 中, 样片数 $m=4$, 裁床原点 r 既表示裁刀的初始位置(起点 s), 也表示裁刀加工完毕后回到的终点位置(终点 e), 则节点集数 $k=6$ 。求解最短空行程的过程为^[7]: 首先, 确定第 $k-1$ 段所有轮廓点到达终点 e 的最短空行程长度; 其次, 确定第 $k-2$ 段中所有轮廓点到终点 e 的最短空行程长度, 该步骤需要上一步所形成的信息。反复执行上述操作直到确定起点 s 到终点 e 的最短空行程长度。

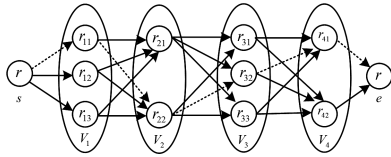


图 7 动态规划算法

Fig. 7 Dynamic programming algorithm

3.5 总体算法流程

本文采用 MVNS 结合 TS-DP 算法求解空行程路径优化问题, 其算法流程如图 8 所示。

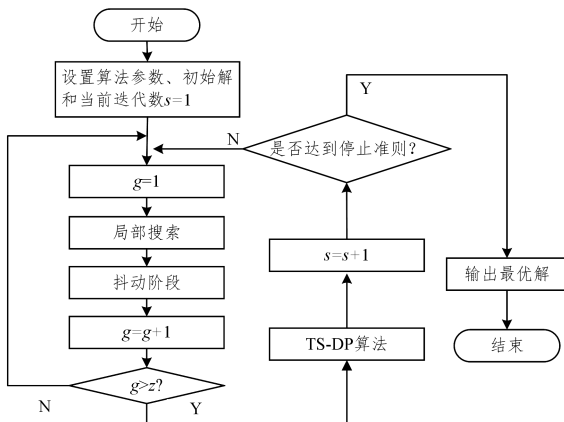


图 8 MVNS 结合 TS-DP 算法流程图

Fig. 8 Flow diagram of MVNS with TS-DP algorithm

图 8 中有 3 个停止条件, 达到任何一个即停止搜索: 当前迭代数 s 超过最大迭代数 s_{\max} ; 算法搜索时间超过限定的时间 t_{\max} (单位为 s); 当前代最优解与上一代最优解之比连续 h 次大于设定的比值 θ 。

4 实验结果与分析

算法采用混合编程的方式进行仿真实验, 即利用 Visual Studio 2015 软件设计 C++ 执行 MVNS 算法, 通过 MATLAB 2016a 软件执行其余的算法。实验平台使用 Win7 操作系统的 PC 机, 处理器为 1.5 GHz AMD A4-5000 APU with Radeon(TM) HD Graphics, 内存为 4GB。

实验中的主要参数设置如下: 最大迭代数 $s_{\max} = 10$, MVNS 算法中的最大迭代数 $z = 20$, 局部搜索中的邻域结构数量 $k_{\max} = 2$, 抖动阶段的分块数 $n = 5$, 交叉概率上下限 $p_{c \max} = 0.4$, $p_{c \min} = 0.1$, 变异概率上下限 $p_{m \max} = 0.3$, $p_{m \min} = 0.01$, 运行时间上限 $t_{\max} = 60$, 次数上限值 $h = 4$, 设定比值 $\theta = 0.9$ 。

4.1 应用实例

选取 10 份由博克服装 CAD 软件生成的排样图, 独立运行 10 次。表 1 列出了本文算法针对 10 份排样图的优化结果, 优化率表示本文算法获取的空行程长度相比于 CAD 软件获取的空行程长度的优化百分比。表 1 中, 时间为求解每份排样图空行程的平均运行时间。

表 1 不同排样图的优化结果

Table 1 Optimization results of different layout drawings

排样图	样片数	CAD 软件/mm	MVNS/mm	优化率/%	时间/s
1	11	3914	1912	51.15	0.5
2	30	17974	7052	60.76	3.2
3	35	29523	10560	64.23	5.8
4	66	48854	14184	70.97	9.3
5	70	29524	10568	64.20	5.8
6	72	32645	13655	58.17	9.1
7	82	26837	18012	67.11	6.2
8	105	43425	15379	64.59	11.7
9	140	57087	21383	62.57	18.9
10	222	88854	55609	62.58	22.1
平均值	—	—	—	62.63	9.3

针对表 1 中的数据, 主要从优化率和运行时间两方面进行分析。在优化率方面, 目前大多数服装 CAD 软件采用的算法是把排样顺序作为裁剪顺序, 通过最近邻算法确定入刀点的位置, 虽然花费的时间较短, 通常在 10s 以内, 但是裁剪效果并不理想。对于样片数量为 11~222 的排样图, 使用本文算法时的优化率为 51%~71%; 当样片数低于 100 时, 空行程长度缩减了 2002~34670mm, 优化率为 51%~71%; 当样片数高于 100 时, 空行程长度缩减了 27712~55609mm, 优化率均高于 60%。这说明文本算法对大规模样片的裁剪效果更加突出。在时间方面, 10 组数据的平均运行时间为 9.26s, 样片数量最多的排样图运行时间为 22.1s, 当样片数低于 100 时其运行时间低于 10s, 当样片数高于 100 时其运行时间为 11.7~22.1s, 因此运行时间较短。在图 9 所示的空行程优化图中样片数为 30, 粗实线表示裁刀空行程轨迹。

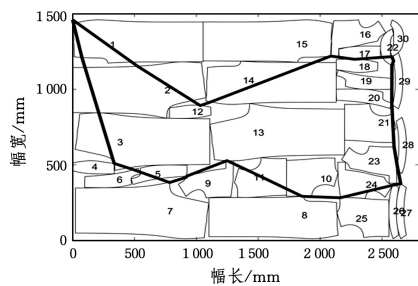


图9 空行程优化结果

Fig.9 Optimization result of non-cutting path

由于服装样片的平均轮廓点数为25~70,轮廓点数量较多,使得动态规划的遍历次数较多,导致运行时间稍长。因此,针对大规模样片,在误差允许的情况下,可以通过减少对轮廓点的遍历次数来缩短运行时间,选取表1中样片数大于100的排样图,结果如表2所列。由表2可知,当轮廓点数缩减为原来的约1/4~1/9时,优化率依旧保持在60%以上,而求解时间从原来的11~22s缩短到7~13s。因此,可以根据实际情况,牺牲少量的精度来缩短求解时间。

表3 3种算法对TSP基准算例的测试结果

Table 3 Test results of three algorithms for TSP benchmark instances

No.	Instances	No. of cities	Optimal	IVNS		GSOA		MVNS	
				value	error/%	value	error/%	value	error/%
1	berlin52	52	7542	7544.36	0.03	—	—	7544.36	0.03
2	st70	70	675	677.11	0.31	682.6	1.13	677.17	0.32
3	eil76	76	538	545.39	1.37	558.8	3.86	544.36	1.18
4	kroA100	100	21282	21618.2	1.58	—	—	21282	0.00
5	lin105	105	14379	14383	0.03	14383.3	0.03	14442	0.44
6	pr136	136	96772	97979.11	1.25	100023.5	3.36	98730	2.02
7	pr152	152	73682	73847.6	0.22	74286.2	0.82	74155	0.64
8	rat195	195	2323	2450.14	5.47	2327.9	0.21	2391.4	2.94
9	kroA200	200	29368	30300.56	3.18	30434.1	3.63	29616	0.84
10	rd400	400	15281	16155.91	5.73	15281	0.00	15857	3.77
Average					1.92		1.63		1.22

对表3中3种算法的测试结果进行分析可知,MVNS在算例kroA100中达到最优值,误差为0;其余算例中,MVNS的误差均在4%以内。对比MVNS和IVNS的优化值结果,MVNS在6个算例上均优于IVNS;在其余4个算例中,两者差距不超过1%,平均误差分别为1.92%和1.22%,可以看出MVNS对VNS的改进效果更优。对比MVNS和GSOA的优化值结果,MVNS在5个算例上获得了比IVNS更优的

表2 大规模排样图的优化结果

Table 2 Optimization results of large-scale layout drawings

样片数	处理前			处理后		
	轮廓点数	优化率/%	时间/s	轮廓点数	优化率/%	时间/s
105	3500	64.59	11.7	731	63.08	7.18
140	5196	62.54	18.9	984	61.73	11.7
222	11965	62.58	22.1	1314	61.10	12.1

4.2 对比实验

本文对GTSP的处理是先用MVNS算法搜索裁剪序列,该过程可以看作对TSP的求解,因此通过对TSP的测试可以评估MVNS算法的性能,选取标准TSPLIB库中常用的10组测试算例进行测试。通过每种算法所求得长度优化值value与目前网站公布的最优值Optimum获取误差error^[2]:

$$error = \frac{value - Optimum}{Optimum} \times 100 \quad (6)$$

本文算法独立运行10次,将优化值与改进的变邻域搜索算法(Improved VNS, IVNS)^[16]和一种无监督学习算法GSOA^[22]进行比较,结果如表3所列,其中加粗的数据表示3种算法在每个算例中的最优结果,原文中未给出优化值的算例不显示结果。

结果;在其余3个算例中两者差距不超过4%,平均误差分别为1.63%和1.22%,结合IVNS的平均误差1.92%可以看出,MVNS具有较高的求解精度。

为了评估本文方法的整体效果,将MVNS与TS-DP混合,选取标准GTSP库常用的11组算例进行测试。本文算法独立运行10次,将其优化值与近年来的有效算法ICA^[6]、IQEA^[7]进行比较,结果如表4所列。

表4 3种方法对GTSP基准算例的测试结果

Table 4 Test results of three methods for GTSP benchmark instances

No.	Instances	Optimal	ICA		IQEA		MVNS	
			value	error/%	value	error/%	value	error/%
1	11eil51	174	174	0.00	—	—	174	0.00
2	20kroB100	10328	10328	0.00	10328	0.00	10327	-0.01
3	30kroA150	11018	11018	0.00	11018	0.00	11107	0.81
4	40kroA200	13111	13116	0.04	13112	0.01	13356	1.87
5	53gil262	1013	1013	0.00	1013	0.00	1042.2	2.88
6	46pr226	64007	64007	0.00	—	—	64007	0.00
7	53pr264	29549	29549	0.00	29549	0.00	29827	0.94
8	60pr299	22612	22615	0.013	22615	0.01	23992	1.68
9	84fl417	9651	9651	0.00	9651	0.00	9660	0.09
10	88pr439	60100.4	60100.4	0.00	—	—	60896	1.32
Average				0.01		0.01		0.96

对表 4 中 3 种方法的测试结果进行分析可知, MVNS 在 2 个算例(11eil51 和 46pr226)中达到最优值, 误差为 0; 在算例 20kroB100 中的优化值 10 327 优于最优值 10 328, 误差为 -0.01%; 其余算例的误差均在 3% 以内。对比 MVNS 和 ICA 的优化值结果, MVNS 有 1 个算例(20kroB100) 优于 ICA; 在 2 个算例(11eil51 和 46pr226)上达到与 ICA 相同的值; 在其余 7 个算例中, 两者差距均未超过 3%。对比 MVNS 与 IQEA 的优化值结果, MVNS 有 1 个算例(20kroB100) 优于 IQEA, 有 1 个算例(46pr226)达到与 IQEA 相同的值; 在其余 5 个算例中, 两者差距均未超过 3%。此外, 三者的平均误差分别为 0.01%, 0.01% 和 0.96%。因此, 可以看出, 本文方法的求解精度与近年来的有效算法的平均误差值不超过 1%。

为了进一步评估本文方法的时间性能, 将本文方法与上述对比算法中效果最好的 IQEA 进行对比。由于时间对比实验需要保证测试环境一致, 且运行时间也与设计的代码有一定关系, 目前主要存在两种评估方式: 1) 复现对比算法, 但是其难以保证与原作者的代码一致, 处理起来比较困难; 2) 将本文方法置于对比算法的实验环境下进行测试, 与对比算法文献中给出的结果进行对比, 此方式更有说服力, 因此本文采用此方式。本文方法在文献[7]中的实验环境下运行, 对各个算例进行 30 次独立运算, 而 IQEA 的实验结果与原文献保持一致, 运行时间的对比结果如表 5 所列, 其中 *time* 和 *aver* 分别表示求解每个算例的平均运行时间和平均路径长度。*dev* 表示本文方法的 *time* 值相比于 IQEA 的 *time* 值所缩短的百分比。

表 5 运行时间对比结果

Table 5 Results of running time comparison

No.	Instances	IQEA		MVNS		
		<i>aver</i>	<i>time/s</i>	<i>aver</i>	<i>time/s</i>	<i>dev/%</i>
1	20kroB100	10416	43	10401	11.5	73.3
2	30kroA150	11182	50	11278	12.2	75.6
3	40kroA200	13222	66	13456	14.7	77.7
4	53gil262	1052	38	1066	14.2	62.6
5	53pr264	30018	78	30133	15.7	79.9
6	60pr299	22832	80	24176	17.1	78.6
7	84fl417	9801	78	9771	25.0	67.9
Average			67		15.8	73.7

对表 5 中两种方法的测试结果进行分析可知, 从运行时间 *time* 来看, 本文方法优于 IQEA; 在 7 个算例中, IQEA 的运行时间为 38~80 s, 而本文方法的运行时间为 11.5~25 s, 平均运行时间相比于 IQEA 缩短了 73.7%, 再结合表 1 与表 2 的测试结果可知, 本文算法的运行时间较短。从平均路径长度 *aver* 来看, 本文方法在 3 个算例中(20kroB100 和 84fl417) 优于 IQEA, 其余算例中两者求得的结果较为相近, 再结合表 3 与表 4 的测试结果可知, 本文方法的求解精度较高。因此, 本文能够同时兼顾算法的求解精度与运行时间。

结束语 本文针对空行程路径优化问题, 提出了一种基于改进 VNS 的元启发式方法。该方法以最短空行程长度为

主要目标, 同时考虑算法的运行时间, 将空行程路径优化问题转化为一类 GTSP, 对传统 VNS 中的局部搜索和抖动阶段进行改进以优化样片裁剪序列, 并设计 TS-DP 算法确定入刀点位置。对实际服装样片数据和标准问题进行测试, 结果表明了本文方法的有效性。

然而, 本文方法仍存在一些不足, 值得进一步探究: 1) 除了外轮廓的空行程问题, 可以继续探索样片的内轮廓空行程问题; 2) 在入刀点的优化阶段, 可以对 TS-DP 算法进行改进或者寻求其他搜索算法来替换该算法, 从而进一步提高本文方法的求解精度和运行时间; 3) 鉴于变邻域搜索算法的通用性, 今后将进一步拓展该算法的应用领域。

参考文献

- [1] SHAHZADEH A, KHOSRAVI A, ROBINETTE T, et al. Smooth Path Planning Using Biclothoid Fillets for High Speed CNC Machines[J]. International Journal of Machine Tools and Manufacture, 2018, 132: 36-49.
- [2] CHENTSOV A G, CHENTSOV P A, PETUNIN A A, et al. Model of Megalopolises in the Tool Path Optimisation for CNC Plate Cutting Machines[J]. International Journal of Production Research, 2018, 56(14): 4819-4830.
- [3] SHERIF S U, JAWAHAR N, BALAMURALI M. Sequential Optimization Approach for Nesting and Cutting Sequence in Laser Cutting [J]. Journal of Manufacturing Systems, 2014, 33(4): 624-638.
- [4] KARUPPANAN B R C, SARAVANAN M. Optimized Sequencing of CNC Milling Toolpath Segments Using Metaheuristic Algorithms[J]. Journal of Mechanical Science and Technology, 2019, 33(2): 791-800.
- [5] ZHANG C, HAN F, ZHANG W. A Cutting Sequence Optimization Method Based on Tabu Search Algorithm for Complex Parts Machining[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2019, 233(3): 745-755.
- [6] ARDALAN Z, KARIMI S, POURSAZBI O, et al. A Novel Imperialist Competitive Algorithm for Generalized Traveling Salesman Problems[J]. Applied Soft Computing, 2015, 26: 546-555.
- [7] WANG Z, YANG W B, WANG W L, et al. Path Optimization for Multi-contour Based on Quantum Evolutionary Algorithm [J]. Computer Integrated Manufacturing Systems, 2017, 23(10): 2128-2135.
- [8] SMITH S L, IMESON F. GLNS: An Effective Large Neighborhood Search Heuristic for the Generalized Traveling Salesman Problem[J]. Computers & Operations Research, 2017, 87: 1-19.
- [9] CHENTSOV A, KHACHAY M, KHACHAY D. Linear Time Algorithm for Precedence Constrained Asymmetric Generalized Traveling Salesman Problem [C] // 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016. Troyes: Elsevier, 2016: 651-655.
- [10] ZIA M, CAKIR Z, SEKER D. Spatial Transformation of Equality-Generalized Travelling Salesman Problem to Travelling

- Salesman Problem[J]. ISPRS International Journal of Geo-Information, 2018, 7(3): 115-131.
- [11] VENKATESH P, SINGH A. An Artificial Bee Colony Algorithm with Variable Degree of Perturbation for the Generalized Covering Traveling Salesman Problem[J]. Applied Soft Computing, 2019, 78: 481-895.
- [12] HANSEN P, MLADENOVIĆ N. Variable Neighborhood Search: Principles and Applications[J]. European Journal of Operational Research, 2001, 130(3): 449-467.
- [13] XU Z, CAI Y. Variable Neighborhood Search for Consistent Vehicle Routing Problem[J]. Expert Systems with Applications, 2018, 113: 66-76.
- [14] RANJBAR M, AHMAD K. A Generalized Variable Neighborhood Search Algorithm for the Talent Scheduling Problem[J]. Computers & Industrial Engineering, 2018, 126: 673-680.
- [15] PALUBECKIS G, TOMKEVIČIUS A, OSTREIKA A. Hybridizing Simulated Annealing with Variable Neighborhood Search for Bipartite Graph Crossing Minimization[J]. Applied Mathematics and Computation, 2019, 348: 84-101.
- [16] HORE S, CHATTERJEE A, DEWANJI A. Improving Variable Neighborhood Search to Solve the Traveling Salesman Problem [J]. Applied Soft Computing, 2018, 68: 83-91.
- [17] WANG X, GOLDEN B, WASIL E. A Steiner Zone Variable Neighborhood Search Heuristic for the Close-Enough Traveling Salesman Problem [J]. Computers & Operations Research, 2019, 101: 200-219.
- [18] HERRÁN A, COLMENAR J M, DUARTE A. A Variable Neighborhood Search Approach for the Hamiltonian P-median Problem[J]. Applied Soft Computing, 2019, 80: 603-616.
- [19] QIU Y, WANG L, XU X, et al. A Variable Neighborhood Search Heuristic Algorithm for Production Routing Problems[J]. Applied Soft Computing, 2018, 66: 311-318.
- [20] OLENDER P, OGRYCZAK W. A Revised Variable Neighborhood Search for the Discrete Ordered Median Problem[J]. European Journal of Operational Research, 2019, 274(2): 445-465.
- [21] SUN Q, JIN Y, HE K, et al. Hybrid Evolutionary Algorithm for Solving Mixed Capacitated General Routing Problem[J]. Computer Science, 2018, 45(4): 76-82.
- [22] FAIGL J. GSOA: Growing Self-Organizing Array-Unsupervised Learning for the Close-enough Traveling Salesman Problem and Other Routing Problems[J]. Neurocomputing, 2018, 312: 120-134.



LIAO Yi-hui, born in 1994, postgraduate. His main research interests include intelligent algorithms and path optimization.



LIU An-dong, born in 1985, Ph.D, associate professor. His main research interests include distributed predictive control and robot control.