

# 基于优化遗传算法的机器人靶全局路径规划研究

张朋 王宇

(南京润景丰创信息技术有限公司 江苏 南京 210000)

**摘要:** 针对传统遗传算法在机器人全局路径规划上存在的路径不平滑、搜索效率低等缺陷,以用于军事训练的机器人靶为研究对象,提出一种优化的遗传算法。根据障碍物分布比例随机生成靶场训练场地障碍物栅格地图模型,采用序号法进行染色体编码,构建基于路径长度和路径平滑度的适应度函数,根据先验知识指导生成初始种群,新增插入算子,重新定义选择、交叉和变异算子。从仿真结果来看,优化后的算法比传统算法在路径长度上稍逊,但在路径平滑度和搜索效率上得到很大的改善,更适用于真实场景下机器人靶的路径行走。

**关键词:** 机器人靶; 遗传算法; 插入算子; 路径规划

## 0 引言

在军事训练相关领域,机器人一般作为战术训练靶标,用于模拟歹徒或人质的运动,配合战术训练,相比传统的固定打靶,有很多优势,比如自由行走、智能对抗、战术配合等,可有效提高射击人员的搜索和临场反应能力。全局路径规划作为机器人靶的一项基本功能,合理的路径规划在战术训练中至关重要。机器人靶作为一种特殊的移动机器人,其全局路径规划的本质就是在包含障碍物的靶场训练场景中,规划出一条从出发点到终止点的最优无碰撞路径。

对于移动机器人的全局路径规划,大量学者对其进行研究,并提出了很多方法,如人工势场法、A×算法、D×算法、蚁群算法、模拟退火算法、神经网络算法和遗传算法等。参考文献[3]的舒红等人提出了一种基于多维信息素及模糊集的改进蚁群算法,可有效减少地图搜索空间,节约搜索时间,提高了对最优路径问题的优化性能与收敛速度。参考文献[4]的韩雪等人在传统人工势场法基础上,通过在斥力场增加与目标点相对位置因子来解决目标点振荡不可到达问题,通过判断局部极小来动态调整移动机器人的行走方向来摆脱局部极小点。参考文献[5]的张毅等人针对传统的遗传算法基础上,采用简单的一维编码替代复杂的二维编码,重新定义了交叉算子和变异算子,并将最短路径和免碰撞相结合作为适应度函数进行遗传优化,使得算法搜索效率得到很大提升,其不足之处就是尽管规划的路径距离可能最短,但未考虑路径的平滑程度。

本文在全局路径规划目标上综合考虑路径规划长度和平滑程度,根据障碍物分布比例随机生成靶场训练场地障碍物栅格地图模型,构建基于路径长度和路径平滑度的适应度函数,根据先验知识生成初始种群,新增插入算子,重新定义选择、交叉和变异算子,最后在 Matlab 中进行仿真分析,仿真结果显示优化后的遗传算法可有效提升算法进化效率和路径规划的质量。

## 1 遗传算法基本原理

遗传算法的本质是一种并行搜索算法,从一个初始种群开始进行选择、交叉、变异等操作得到下一代种群,按照适应度评价规则指导搜索过程,算法迭代演化后在最终满足

终止条件的基础上输出最优解。

下面是遗传算法的基本步骤:

Step1: 确定个体编码方式和适应度函数;

Step2: 通过先验知识或随机方式产生初始种群;

Step3: 根据适应度函数计算种群内个体的适应值;

Step4: 判断是否满足终止条件,若满足转至 Step6;

Step5: 对群体进行遗传操作(选择、交叉、变异等)

后产生下一代群体,转至 Step3;

Step6: 停止搜索,输出最优个体。

## 2 改进后的遗传算法

针对机器人靶在靶场训练场景下的全局路径规划问题,提出一种优化的遗传算法,提高算法搜索效率和路径规划质量。优化主要包含以下几部分:

- (1) 采用栅格法建立靶场区域环境模型;
- (2) 采用序号法进行染色体编码;
- (3) 构建基于路径长度和路径平滑度的适应度函数;
- (4) 利用先验知识来指导种群初始化;
- (5) 新增插入算子来修复种群中的个体栅格不连续的问题;
- (6) 选择算子采用轮盘赌策略;
- (7) 交叉算子采用等位基因互换策略;
- (8) 变异算子基于随机两个基因位进行插入算子操作。

### 2.1 算法假设

- (1) 忽略机器人靶的尺寸,将其视为一个质点进行处理;
- (2) 靶场区域内的障碍物信息是静态已知,可通过 1 个或多个栅格进行表示;
- (3) 机器人单次步进位置包仅括所在栅格处相邻的 8 个位置。

### 2.2 靶场环境建模

本文采用栅格法来构建靶场训练场景,为了方便验证不同障碍物分布情况下的算法可行性,障碍物按照占比随机分布在二维栅格平面中。栅格单元信息采用二进制编码格式,1 表示障碍物栅格,以灰色表示;0 表示自由栅格,以白色表示。如图 1 所示为障碍物占比为 0.2 时生成的一个随机栅格地图。其中水平方向为 X 轴,垂直方向为 Y 轴,栅

格区域为，按照从左到右、从下到上的顺序给每个栅格进行编号0~399。

根据栅格二维图可进一步得到栅格编号和实际坐标的函数映射关系，见公式(1)所示：

$$\begin{cases} x_i = \text{mod}(i, n) + 1 \\ y_i = \text{fix}(i, n) + 1 \end{cases} \quad (1)$$

其中：mod为取余操作，fix为向0靠近取整操作，n表示栅格的列数。

20	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399
19	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379
18	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359
17	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339
16	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
15	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
14	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
13	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259
12	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
11	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
10	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199
9	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
8	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
7	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139
6	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
4	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
3	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
2	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	坐标X																			

图1 20×20 栅格地图

### 2.3 染色体编码

染色体编码策略对后续的适应度函数和遗传算子的设计具有重要的影响，尤其是交叉和变异算子，其编码方式是决定遗传算法收敛速度的关键。

本文所设计的染色体为机器人在起始点和终止点之间的一条可通过路径，其路径可通过一连串的栅格序号(N<sub>o1</sub> → L → N<sub>oi</sub>)或一连串的坐标([x<sub>1</sub>, y<sub>1</sub>] → L → [x<sub>i</sub>, y<sub>i</sub>])来表示，这样就将机器人的全局路径规划问题转换为在起始点和终止点之间寻找一条最优的序列栅格。两种路径表示方式上序号法较为简洁，本文采用序号法对染色体进行编码，可有效节省存储空间，方便后续遗传算子的操作。此外，在评估自适应函数时，路径序号可以直接映射到直角坐标系中，方便计算路径长度和路径平滑程度。由于遗传算法采用的是并发随机搜索的方式，潜在的可通行路径长短不一，因此染色体的长度也是不一致的，为了防止长度无限放大，需要设置阈值来控制，若超过阈值则可直接舍弃该染色体。该阈值的设定与障碍物占比成正比关系，见公式(2)所示：

$$l_{\max} = \lambda_{\text{obs}} \times \text{size}(G_{m \times n}) \quad (2)$$

其中，l<sub>max</sub>表示染色体长度阈值，λ<sub>obs</sub>表示障碍物在栅格区域的占比，G<sub>m×n</sub>表示栅格数组。

### 2.4 适应度函数构建

适应度函数用来评价种群中每个染色体的优劣，对算

法的快速收敛和稳定性有着重要的影响。机器人路径规划一般都是追求最短路径，但最短路径并不一定是耗时最短且能耗最少的路径，最终路径中往往会存在路径不平滑、多次转弯的问题，本文综合考虑路径长度和路径平滑度，构建基于路径长度和路径平滑度的适应度函数。

#### (1) 路径长度适应度函数

将基于序号法编码的染色体个体通过坐标映射关系转化为直角坐标系下的坐标序列，通过坐标序列求解整个路径段的长度，具体见公式(3)：

$$L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (3)$$

其中，L表示该染色体路径段长度，n表示染色体基因的个数，(x<sub>i</sub>, y<sub>j</sub>)表示机器人当前位置坐标，(x<sub>i+1</sub>, y<sub>j+1</sub>)表示机器人后一位置的坐标。

根据染色体路径长度构建路径长度适应度函数，具体见公式(4)：

$$fit_1 = L^{-1} \quad (4)$$

#### (2) 路径平滑度适应度函数

由于机器人单次步进位置包仅括所在栅格处相邻的8个位置，因此可以采用当前位置点与前后两个位置点构成的角度来评估路径的平滑程度，夹角求解见公式(5)：

$$\begin{cases} \vec{a} = (x_{i+1} - x_i, y_{i+1} - y_i) \\ \vec{b} = (x_{i+2} - x_{i+1}, y_{i+2} - y_{i+1}) \\ \theta = \arccos(\vec{a} \times \vec{b}) / (|\vec{a}| \times |\vec{b}|) \end{cases} \quad (5)$$

其中， $\vec{a}$ 表示当前位置点指向前一个位置点的向量， $\vec{b}$ 表示当前位置点指向后一个位置点的向量，θ表示两个向量之间的夹角。

三点构成的角度分布如图2所示，大致可以分为四种情况：平角180°、钝角135°、直角90°和锐角45°。平滑度最好的是平角，其余依次是钝角、直角和锐角。

按照这四种角度分布情况，按照幂次等次分别设置各

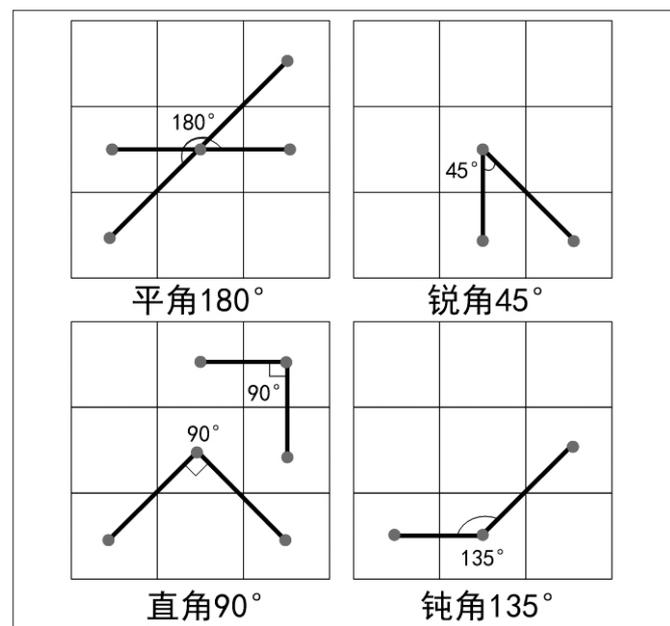


图2 三点构成的角度分布图

自的惩罚因子, 该路径平滑度自适应函数由一系列的惩罚因子求和得出, 具体见公式 (6) :

$$\left\{ \begin{array}{l} \rho_i = \begin{cases} c^0 & \theta = \pi \\ c^1 & \theta = 0.75\pi \\ c^2 & \theta = 0.5\pi \\ c^3 & \theta = 0.25\pi \end{cases} \\ fit_2 = \sum_{i=1}^{n-2} \rho_i \end{array} \right. \quad (6)$$

其中,  $\rho_i$  表示角度惩罚因子,  $C$  表示惩罚指数。

根据以上两部分, 选取合适的权重系数  $W_1$  和  $W_2$  构建基于路径长度和路径平滑度的适应度函数  $fit$ , 见公式 (7) :

$$fit = w_1 \cdot fit_1 + w_2 \cdot fit_2 \quad (7)$$

其中,  $W_1$  表示染色体路径长度权重,  $W_2$  表示染色体路径平滑度权重。

### 2.5 种群初始化

由于靶场区域环境建模之后, 才会进行种群初始化, 因此二维栅格地图信息是已知的。传统的遗传算法通过规避障碍物栅格, 在自由栅格中随机搜索的方式生成初始种群, 操作虽然简单, 但在随机过程中会产生大量不可行路径个体, 影响后续的算法运行效率。

本文采用先验知识来指导种群初始化, 起始点和终止点之间通过行递增的方式随机选择自由栅格, 使得路径朝向目标点初始化, 再通过插入算子来修复相邻栅格之间存在的不可通行问题, 最终输出一条从起始点到终止点之间可通行的路径。具体算法步骤如下:

Step1: 根据起始点和终止点确定栅格行数 (包括起始点和终止点所在的行) ;

Step2: 以起始点所在行开始, 以行递增的方式, 在每一行随机选择一个自由栅格;

Step3: 对 Step2 产生的一系列栅格序号, 从左往后依次评估相邻栅格之间的连通性, 若不连通, 则引入插入算子, 若插入操作失败, 则直接舍弃该染色体, 并转至 Step2;

Step4: 输出一条从起始点到终止点可通行的路径, 判断个体数量是否超出初始种群阈值, 若没有, 转至 Step2, 否则种群初始化结束。

### 2.6 遗传算子定义

#### 2.6.1 插入算子

插入算子的作用是修补路径栅格中不连续的问题, 采用公式 (8) 来判断染色体中相邻栅格序号、是否连续。

$$\Delta = \max\{\text{abs}(x_{i+1} - x_i), \text{abs}(y_{i+1} - y_i)\} \quad (8)$$

其中,  $\Delta$  表示相邻栅格  $\rho_i$ 、 $\rho_{i+1}$  在  $X$  和  $Y$  方向最大差值, 若  $\Delta=1$ , 表示  $\rho_i$ 、 $\rho_{i+1}$  是连续的, 否则不连续。

若相邻栅格不连续, 则引入插入算子。该插入算子引入一个中点栅格概念, 具体定义见公式 (9) :

$$\left\{ \begin{array}{l} x_m = \text{floor}\left[\frac{1}{2}(x_i + x_{i+1})\right] \\ y_m = \text{floor}\left[\frac{1}{2}(y_i + y_{i+1})\right] \end{array} \right. \quad (9)$$

其中, floor 为向负方向取整操作。

插入算子的算法步骤如下所示:

Step1: 在两个不连续栅格中取中点栅格, 判断中点栅格是否为自由栅格, 若是自由栅格直接插入, 并转至 Step3, 若是障碍物栅格, 转至 Step2;

Step2: 按照上、下、左、右、左上、右上、左下、右下的顺序来选择相邻自由栅格, 若成功选取到, 判断该栅格是否已存在目前路径中, 若不在, 进行插入操作, 转至 Step3, 若遍历完 8 个位置, 依旧没有满足条件的栅格, 则表示插入失败, 直接舍弃该染色体;

Step3: 继续判断新插入的栅格和新插入栅格的前一个栅格是否连续, 若不连续, 转至 Step1, 若连续, 向后继续寻找不连续栅格, 重复前三个步骤直至路径中所有栅格都是连续的;

#### 2.6.2 选择算子

采用适应值比例选择法策略进行选择算子的设计, 主要目的是使得上一代种群中适应度大的个体以较大概率被选择出来, 并且还要保持群体规模不变。具体算法步骤如下所示:

Step1: 求解种群中每个染色体的适应度值  $fit_i$  ;

Step2: 求解种群中所有染色体适应度值累加和  $FIT = \sum fit_i$  ;

Step3: 求解种群中每个染色体被选中的概率  $\gamma_i = fit_i / FIT$  ;

Step4: 对种群中的染色体进行概率求和排序得到概率中间累加值  $\eta = \sum \gamma_j$  ;

Step5: 在 0~1 之间产生随机数  $N$ , 并与  $\eta_i$  比较, 寻找  $\eta_i \geq n$  的个体;

Step6: 重复 Step5, 直至选出与上一代群体规模一致的染色体。

#### 2.6.3 交叉算子

交叉操作是遗传算法中染色体基因重组的基本机制, 一般都是用在染色体选择操作之后。传统算法一般选用单点交叉, 对进行交叉的两条染色体随机选取交叉点, 这里的交叉点是指两条染色体除起始和终止点以外的相同栅格序号, 这样可以保证交叉后生成的新染色体路径也是连续的。本文对传统交叉算子进行改良设计, 通过分组随机抽取交叉的方式, 并对交叉后代与父代的适应度值进行比较, 选择最优的一代, 进而可以保证算法朝最优的方向前进。具体算法步骤如下所示:

Step1: 将种群中的染色体按照适应度值高低排序, 分成两部分: 高适应度值组和低适应度值组;

Step2: 从高低两组中分别随机抽取一个染色体进行交叉操作;

Step3: 计算交叉后的染色体适应度值, 并与父代染色体适应度值进行比较, 淘汰适应度值低的一代;

Step4: 重复 Step2, 直至交叉的数量满足算法设计的阈值要求。

#### 2.6.4 变异算子

变异操作是指对染色体内部基因进行突变, 扩大了算

法的搜索空间，避免陷入算法局部最优，对种群个体多样性有很重要的作用。传统算法的随机变异可能会导致变异前可通行路径在变异后无法通行的问题。本文优化了传统变异算子，通过随机选择两个基因位并进行插入算子的修复操作，保证了变异后的染色体路径依旧是可通行的，具体算法步骤如下所示：

Step1：染色体中随机选取两个基因位，丢弃之前两个基因位之间的栅格序号；

Step2：通过插入算子重新修补两个基因位的不连续问题，若修补失败，转至 Step3，若修补成功，转至 Step4；

Step3：判断修补次数是否超过算法设定阈值，若没有，返回至 Step1，若超过，直接保留父代染色体；

Step4：输出一条变异后的染色体。

### 3 实验结果与分析

利用 Matlab 仿真平台对算法的可行性进行仿真测试，证明了优化后的遗传算法在机器人靶全局路径规划上的可行性。

#### 3.1 参数选取

算法参数主要包括环境栅格大小  $G_{m \times n}$ 、障碍物占比、起始点序号、终止点序号、种群规模、遗传迭代次数、交叉概率、变异概率、路径长度比重、路径平滑度比重、变异插入次数阈值。

其中，算法参数取值见表 1 所示。

#### 3.2 算法仿真

为了验证算法的通用性，障碍物在地图上的分布也是采取随机分布的策略。下面就不同障碍物占比下对比传统遗传算法和优化后的遗传算法。

图 3 ~ 图 6 为两种算法在不同障碍物占比时的全局路径规划图。其中，黑色圆圈代表起始点，红色方块代表终止点，绿色虚线路径为传统遗传算法规划出的路径，蓝色实线路径为优化后的遗传算法规划的路径。

#### 3.3 结果分析

表 2 为不同障碍物占比下两种算法路径长度和转角和

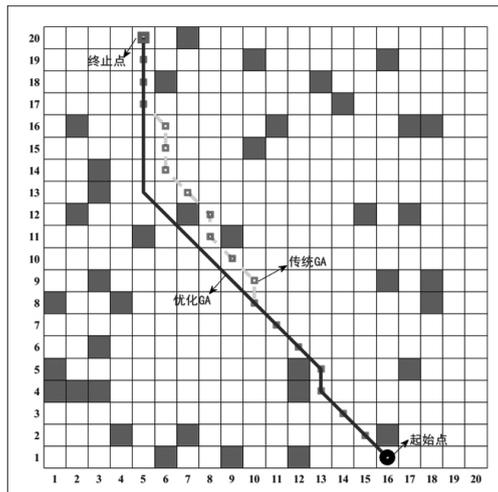


图 3 障碍物占比 0.1 时路径规划图

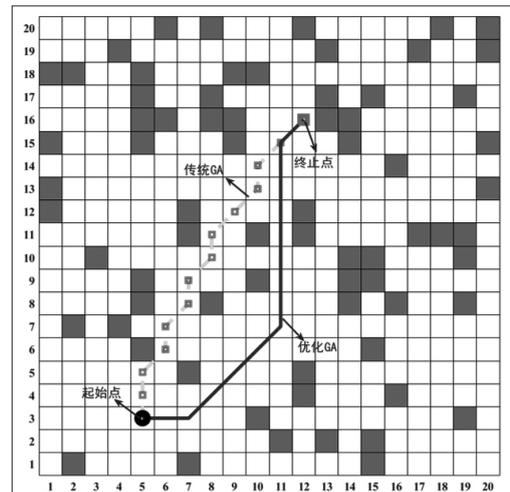


图 4 障碍物占比 0.2 时路径规划图

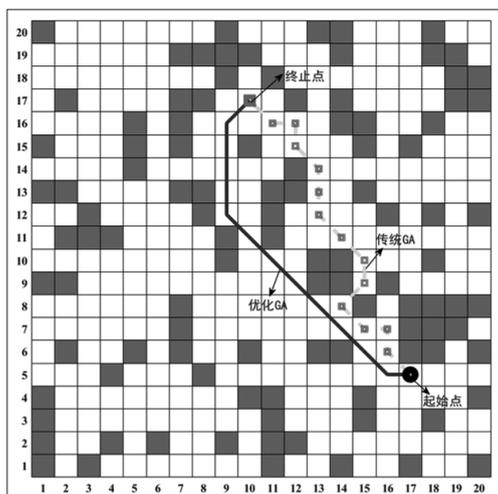


图 5 障碍物占比 0.3 时路径规划图

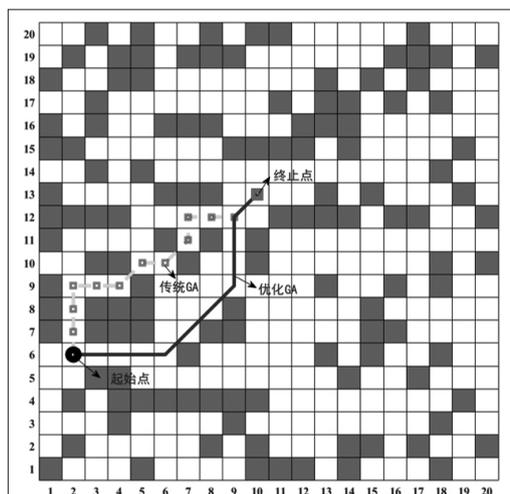


图 6 障碍物占比 0.4 时路径规划图

表 1 优化的遗传算法参数取值表

算法参数	取值
栅格大小 $G_{m \times n}$	$20 \times 20$
障碍物占比 $\lambda_{obs}$	(0.1 0.4)
起始点序号 $p_{start}$	地图随机选取
终止点序号 $p_{end}$	地图随机选取
种群规模 NP	30
遗传迭代次数 T	100
交叉概率 $P_c$	0.8
变异概率	0.2
路径长度比重 $W_1$	1
路径平滑度比重 $W_2$	10
变异插入次数阈值 k	10

对比，表 3 为不同障碍物占比下两种算法路径搜索效率。结合图 2 ~ 图 4，可以看出传统算法路径长度上稍优于优化后的算法，但路径并不平滑，存在多次转弯，且算法收敛速度也低于优化后的算法，这种情况下机器人实际运行时会显得不流畅；优化后的算法规划的路径平滑度得到了很大的改

(下转第 52 页)

与碟簧盒连接；两个丝杠分别拧入一根调节螺母内；去掉托架前面板，增大调整空间，并在两根传动轴轴承盒卡槽处用螺丝固定两个加强筋板，用于保证托架的强度。

2.2.3 创新点

改造后的创新点包括：

(1) 用连杆代替定滑轮，解决了定滑频繁抱死的现象；  
 (2) 采用丝杠代替钢丝绳，丝杠的强度远大于钢丝绳，解决了钢丝绳断股、断裂的现象；

(3) 增大下传动轴调整耳，使其完全覆盖在碟簧盒上，避免油泥进入碟簧盒影响其正常工作；

(4) 除去托架前面板，并在两根传动轴轴承盒卡槽处用螺栓固定加强板，既保证了托架的结构强度，又增加了调节螺母转动空间；

(5) 将调节螺母放在两根调整丝杠中间，调节螺母的位置方便人工操作，降低了劳动强度。

2.3 项目实施效果及经济效益分析

2.3.1 该项目总投资

改造9台平式托架调整装置，共投资9.3元。

2.3.2 节时增效

本项目实施后，大大提高了检修效率，年节约检修停机20小时，按150吨/小时产量、每吨钢盈利80元计算，

则年效益计算为：

$$150 \text{ 吨} / \text{小时} \times 20 \text{ 小时} \times 80 \text{ 元} / \text{吨} = 24 \text{ 万元}。$$

2.3.3 备材效益

项目未实施前，平均每年更换调整器等设备的维护费用约为3.5万元；改造后，设备维护费用暂无投入。

2.3.4 年效益计算

$$24 + 3.5 - 9.3 = 18.2 \text{ 万元}。$$

3 结语

通过对裙板辊和轧机托架的改进优化，降低了裙板电机轴承的工作温度，延长了电机轴承的使用寿命，加强了轧机托架调整机构的结构强度，保证了轧制精度。以上改进降低了设备的故障率，提高了生产线的工作效率，保证了产品质量，具有很高的推广价值。

参考文献：

[1] 周晓峰. 故障诊断技术在棒材精轧机上的应用. 冶金设备管理与维修 [J]. 2015, 04(02): 6-8.  
 [2] 吴斌. 棒材轧机精轧系统工艺优化. 山东冶金 [J]. 2008, 02(01): 2-3.  
 [3] 贾会华. 棒材精轧机传动系统改造. 科技研究 [J]. 2010, 03(02): 10-11.

(上接第49页)

表2 不同障碍物占比下两种算法路径长度和转角和

障碍物占比	传统 GA 算法		优化 GA 算法	
	长度	转角和	长度	转角和
$\lambda_{obs}=0.1$	23.55	405°	23.55	135°
$\lambda_{obs}=0.2$	15.98	405°	17.07	135°
$\lambda_{obs}=0.3$	16.89	585°	16.33	135°
$\lambda_{obs}=0.4$	13.24	405°	14.07	135°

表3 不同障碍物占比下两种算法路径搜索效率

障碍物占比	传统 GA 算法		优化 GA 算法	
	迭代次数	搜索时间	迭代次数	搜索时间
$\lambda_{obs}=0.1$	15	0.22s	10	0.15s
$\lambda_{obs}=0.2$	25	0.42s	18	0.38s
$\lambda_{obs}=0.3$	40	0.78s	32	0.62s
$\lambda_{obs}=0.4$	80	1.64s	76	1.2s

善，算法收敛更快，搜索效率更高。

4 结语

本文实现了基于优化遗传算法的机器人靶全局路径规划研究，根据障碍物分布比例随机生成靶场训练场地障碍物栅格地图模型，构建基于路径长度和路径平滑度的适应度函数，根据先验知识指导生成初始种群，新增插入算子，重新定义

选择、交叉和变异算子。从仿真结果来看，优化后的算法比传统算法在路径长度上稍逊，但在路径平滑度、搜索效率上得到很大的改善，更适用于真实场景下的机器人靶路径行走。在后期研究中，需要进一步优化算法参数，将算法移植到真实的机器人靶控制系统中指导机器人路径行走。

参考文献：

[1] 于正亮, 肖曦. 外军无人训练靶标发展研究 [J]. 国防科技, 2014, 35(4): 99-103.  
 [2] 朱大奇, 颜明重. 移动机器人路径规划技术综述 [J]. 控制与决策, 2010, 25(7): 961-967.  
 [3] 舒红, 封硕, 谢步庆. 轮式机器人路径规划的改进蚁群算法 [J]. 制造业自动化, 2020, 42(1): 64-69.  
 [4] 韩雪, 吴金文, 石瑶. 移动机器人实时避障策略研究及实例仿真 [J]. 工业设计, 2017, (9): 121-122, 129.  
 [5] 张毅, 代恩灿, 罗元. 基于改进遗传算法的移动机器人路径规划 [J]. 计算机测量与控制, 2016, 24(1): 313-316.  
 [6] 刘志海, 薛媛, 周晨, 等. 基于遗传算法的机器人路径规划的种群初始化改进 [J]. 机床与液压, 2019, 47(21): 5-8.

作者简介：张朋（1989.11-），男，江苏盐城人，硕士研究生，中级工程师，研究方向：移动机器人导航控制；王宇（1984.10-），男，江苏徐州人，硕士研究生，中级工程师，研究方向：移动机器人SLAM算法研究。