

[文章编号] 1003-4684(2023)05-0075-07

求解旅行商问题的改进 k-opt 遗传算法

赵 涛,叶志伟,宗欣露,潘 虎

(湖北工业大学计算机学院,湖北 武汉 430068)

[摘 要] 为了增强遗传算法的局部搜索能力,加速算法运行效率,尽量避免算法陷入早熟问题,提出一种改进 k-opt 遗传算法求解旅行商问题。该算法利用改进的 k-opt 方法初始化获得较优种群,引入改进的交叉变异机制增强算法全局搜索能力,结合改进的 k-opt 方法强化算法局部搜索能力。实验结果表明,改进的 k-opt 遗传算法能有效平衡算法探索和开发能力,其求解的质量优且运行效率高。

[关键词] 旅行商问题;k-opt;遗传算法;局部搜索;组合优化问题

[中图分类号] TP391.9 **[文献标识码]** A

旅行商问题(travelling salesman problem, TSP)是组合优化领域的经典问题。其可描述为一个商人从任意城市出发,不重复不遗漏地访问每一个城市,最后返回出发地,其目标是找出一条包含所有城市的最短路径。TSP 问题是一个 NP 问题,对车辆路径、网络路由和车间调度等领域有重要参考价值。许多学者深入研究并提出了线性规划、动态规划、分支定界和 k-opt 等传统方法,这些方法在小规模 TSP 问题中能获得较优的求解方案。随着数据规模逐渐增大,上述方法在中大规模 TSP 问题求解时容易陷入局部最优解,且时间复杂度也随之增加。针对中大规模 TSP 问题,群智能算法如蚁群优化算法(ACO)^[1-2]、粒子群优化算法(PSO)^[3]、模拟退火算法(SA)^[4]和遗传算法(GA)^[5]等能在可期待的时间范围内获得可接受的解决方案,因此被更多的学者关注。

最近,谢聪^[6]提出一种改进离散蝴蝶优化算法,利用贪婪策略初始化种群,结合 2-opt 方法、4-opt 方法(双桥实验)和模拟退火提升算法寻优能力,但引入双桥实验使得算法时间复杂度很高。Karuna 和 Kusum^[7]在灰狼优化算法(grey wolf optimizer, GWO)中引入汉明距离使其适用于组合优化问题,利用 2-opt 方法增强局部寻优能力。Zhang^[8]在离散布谷鸟搜索算法中利用 k-means 算法将中大规模 TSP 问题划分为多个小规模 TSP 问题,以此来加速中大规模 TSP 问题的运行效率。但由于不同

规模城市之间缺乏交互能力,使得最终的求解效果较差。Zhang 和 Yang^[9]提出一种基于麻雀搜索算法(sparrow search algorithm, SSA)的离散麻雀搜索算法(discrete sparrow search algorithm, DSSA),利用麻雀的分工作业机制和全局扰动策略平衡算法的探索和开发能力,提高了算法跳出局部最优解的能力,同时利用 2-opt 方法增强局部搜索能力,取得了比离散灰狼算法更好的效果。k-opt 方法或其变形方法在上述算法均得到有效利用。

与此同时,许多处理连续型优化问题的新型算法在探索方面获得极大优势,而处理连续性问题算法难以完全适用于离散优化问题。因此,针对离散问题的局部优化算子被越来越多的学者关注,针对 TSP 问题的 k-opt 方法成为该问题在局部优化的首先方案。Osaba 等人^[10]动态化调整离散蝙蝠算法,其中 2-opt 和 3-opt 方法分别用于短距离和长距离优化,该操作能够有效地缓解算法陷入局部最优,从而根据种群个体潜力挖掘更优方案;Huang 等人^[11]同样在离散青蛙跳跃算法引入 2-opt 方法优化精英个体,在扰动因子的基础上强化算法搜索能力;Ahmet 等人^[12]在树种算法中加入大量交换、逆转和插入等扰动因子扩大全局探索能力,最终仍然通过 2-opt 方法局部搜索;Zhong 等人^[13]在离散鸽群优化算法中定义逆运算符、块插入运算符和交换运算符等扰动因子加强全局探索,采用模拟退火算法、2-opt 方法和 3-opt 方法作为局部搜索因子,既

[收稿日期] 2022-06-01

[第一作者] 赵 涛(1994-),男,湖北黄石人,湖北工业大学硕士研究生,研究方向为进化算法。

[通信作者] 叶志伟(1978-),男,湖北浠水人,工学博士,湖北工业大学教授,研究方向为机器学习和数据挖掘。

能保证种群多样性,又能确保每个个体方案的优质性。

上述引入 k-opt 方法的算法,其算法时间复杂度较高。针对 k-opt 方法的高时间复杂度问题,本文优化原始 k-opt 方法提出一种求解 TSP 问题的改进遗传算法。一方面,利用改进遗传算法的交叉变异机制(交换、逆转、插入)进行全局搜索,融合深度搜索、分层结构和禁忌表等策略强化全局搜索能力;另一方面,引入改进的 k-opt 方法增强局部搜索能力。算法的探索 and 开发能力在该机制下均得到增强。最后通过实验证明该算法能够高效地求解 TSP 问题,同时获得解的质量较佳。

1 旅行商问题

旅行商问题可表示为一个无向图 $G=(S,E)$ 。其中, $S=\{1,2,\dots,N\}$ 表示顶点集, E 为边集,边 $e_{ij} \in E, (i,j \in S, i \neq j)$ 。设 c_{ij} 为 e_{ij} 的距离,决策变量

$$x_{ij} = \begin{cases} 1, & e_{ij} \text{ 在当前解中} \\ 0, & \text{其他} \end{cases}$$

该问题的优化目标如下:

$$\begin{aligned} \min & \sum_{i=1}^N \sum_{j=1}^N x_{ij} c_{ij} \\ \text{s.t.} & \sum_{j \neq i} x_{ij} = \sum_{i \neq j} x_{ji} = 1, i \in S \\ & \sum_{i \neq j} x_{ij} \leq |V|-1, \forall C \subset S, x_{ij} \in \{0,1\}, i,j \in S \end{aligned}$$

其中, $|V|$ 表示集合 S 中的顶点个数。第一个约束条件表示每个顶点仅和两条边连接,第二个约束条件表示不会产生任何一条子回路。式(2)即本文需要优化的目标,该问题是一个典型的组合优化问题,当前没有确定的算法能在多项式时间范围内找到全局最优解。

2 k-opt 方法及其改进

针对 TSP 问题, Lin^[14] 在 1973 年提出 k-opt 方法优化当前路径。该方法简单有效,能在局部范围内根据当前解寻找更优解,但求解效率受数据规模影响较大。本文改进原始 k-opt 方法提高局部搜索运行效率。

2.1 原始 2-opt 和 3-opt 方法

图 1 为 2-opt 优化过程,该方法选择两个不相邻的点 i, j , 根据下式寻找更短路径。

$$D(i, i+1) + D(j, j+1) > D(i, j) + D(i+1, j+1)$$

其中 $D(i, j)$ 表示 i, j 两点之间的距离。若条件成立,则逆转序列 $L(i+1, j)$ 。其中 $L(i, j)$ 表示现有路径中起点为 i 终点为 j 的路径序列。通过对现有序列不断优化,最终构建一个更短序列。同理,图

2 为 3-opt 优化过程,该方法选择三个不相邻的点 i, j, k , 根据下式优化当前路径。若条件成立,则逆转序列 $L(j+1, k)$ 。

$$\begin{aligned} & D(i, i+1) + D(j, j+1) + D(k, k+1) > \\ & D(i, k) + D(i+1, j+1) + D(j, k+1) \end{aligned}$$

Helsgaun^[15] 系统地讨论了 k-opt 方法中不同 k 取值的优缺点。2-opt 和 3-opt 方法是基于贪心搜索的优化过程(图 1、2),因此以上方法也会继承贪心搜索的缺点,即最终导致陷入局部最优解。同时,2-opt 和 3-opt 方法的时间复杂度和数据规模息息相关。理论上,2-opt 方法逆转序列次数最多为 $O(n^2)$, 3-opt 方法逆转序列次数最多为 $O(n^3)$ 。

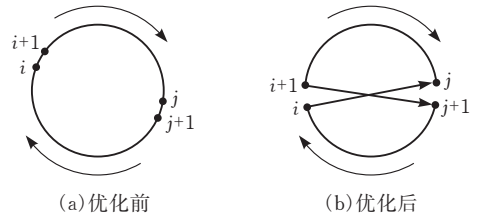


图 1 2-opt 方法优化过程

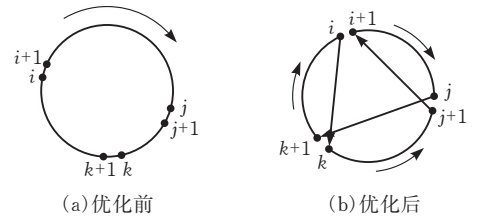


图 2 3-opt 方法优化过程

2.2 改进的 2-opt 和 3-opt 方法

为缓解 2-opt 和 3-opt 方法过早地陷入局部最优解,提出改进的 2-opt 和 3-opt 方法。在改进的 2-opt 方法中,根据当前节点 i 找出与自身不相邻的所有后续节点的集合 J , 构造集合如下:

$$\begin{aligned} \text{Diff } f_2 &= D(i, i+1) + \\ & D(j, j+1) - D(i, j) - D(i+1, j+1) \end{aligned}$$

其中,集合 $\text{Diff } f_2$ 表示所有后续节点与当前节点 i 发生序列交换后的差异。在集合 $\text{Diff } f_2$ 中找出最大值 \max_S_2 , 若条件 $\max_S_2 > 0$ 成立,则逆转 \max_S_2 对应的节点 j 和节点 i 之间的序列,即逆转当前序列 $L(i+1, j)$ 。在改进的 3-opt 方法中,根据当前节点 i, j 找出与自身不相邻的所有节点的集合 K , 构造集合 $\text{Diff } f_3$

$$\begin{aligned} \text{Diff } f_3 &= D(i, i+1) + D(j, j+1) + D(K, K+1) - \\ & D(i, K) - D(i+1, j+1) - D(j, K+1) \end{aligned}$$

与原始 2-opt 方法相比,改进后的 2-opt 算法逆转序列次数不超过 $O(n)$;改进后的 3-opt 方法逆转序列次数不超过 $O(n/2)$ 。其次,对于任意节点,改进后的 2-opt 方法最多逆转序列 1 次,而原始的 2-opt 算法最多逆转序列 $n-2$ 次。

3 改进的 k-opt 遗传算法

本文提出一种改进的 k-opt 遗传算法求解 TSP 问题。首先,在原始遗传算法中融入深度搜索、分层结构和禁忌表等机制强化遗传算法寻优能力。然后,利用改进的 k-opt 方法加速算法局部搜索运行效率和尽量避免算法陷入早熟问题(图 3)。

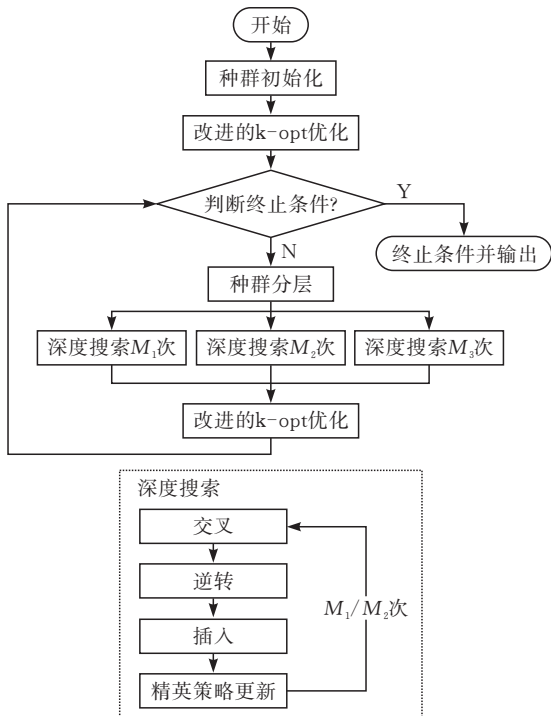


图 3 改进的 k-opt 遗传算法流程

在改进的遗传算法中,首先通过随机编码初始化种群个体,然后以距离作为适应度函数计算不同个体的适应度值。同时,采用改进的 k-opt 方法初始化,使种群个体获得较优解,加速算法的寻优能力。然后,算法采用交换、逆转和插入三种方式全局搜索。其中,交换是指将有序序列中随机两个节点的位置交换,逆转是指将有序序列中随机一个片段置换方向,插入是指将有序序列中的任意一个节点插入到与自身不相邻的位置。通过上述三种方式,种群个体在全局内随机搜索。同时,融合深度搜索、分层结构和禁忌表三种策略增强遗传算法的搜索能力。

首先,采用基于排序的分层结构(第一层群体、第二层群体和劣质群体)细化种群,该策略启发于灰狼优化算法^[16]:

$$X(i) = \begin{cases} \text{firstlayer} & p < 0.2N \\ \text{secondlayer} & 0.2N \leq p \leq 0.8N \\ \text{lastlayer} & p > 0.8N \end{cases}$$

其中, N 为城市规模大小, $p \in [1, N]$ 且为整数, $X(i)$ 表示种群中的第 i 个个体,通过分层结构将种群分为三个不同的分支。第一层群体包含较优求解

方案;第二层群体包含次优求解方案;劣质群体包含拟合效果较差的求解方案。对第一、二层群体进行基于深度搜索的全局探索。即对第一、二层个体分别独立搜索 $M1$ 、 $M2$ 次($M1$ 、 $M2$ 为超参数)。此外,重新初始化劣质群体,增大遗传算法的全局搜索能力。同时采用改进后的 k-opt 方法快速接近局部较优解。该分层结构使优质个体具有更高的频率被优化,同时也使群体有更大概率搜索到更优方案。最后,采用精英策略保留新生成的优质个体。

超参数 $M1$ 、 $M2$ 根据城市规模取两组不同的值。当城市规模不大于 100 时,降低深度搜索频率;当城市规模大于 100 时,加大深度搜索频率。其公式如下:

$$(M1, M2) = \begin{cases} (30, 10), & N \leq 100 \\ (300, 100), & N > 100 \end{cases}$$

此外,在每次搜索之前随机更换序列起点位置,但不改变城市序列顺序,其目的是动态化调整搜索起点。更进一步,在交换、逆转和插入三种搜索方式中,通过添加禁忌表提升搜索有效性。禁忌表的设计细节如下:

1) 在种群初始化时统计所有节点与自身最近临的 5 个节点,构建一个规模为 $(N, 5)$ 的禁忌表;

2) 对任意种群个体均初始化一个 $(1, N)$ 零矩阵禁忌表单 $Tabu$, 并判断个体相邻节点是否存在于禁忌表中。若其邻域节点存在于禁忌表,将禁忌表单中该节点的位置置 1, 表示该节点的邻域节点为较理想节点。

3) 随机从 $S(S = \{1, 2, \dots, N\})$ 或禁忌表单 $Tabu$ 中选择两个节点 (x_1, x_2) , 节点设计如下:

$$(x_1, x_2) = \begin{cases} (i, j) (R < 0.7) \cap (i, j \in S) \\ (i, j) (R \geq 0.7) \cap (i, j \in Tabu) \end{cases}$$

其中, R 是一个随机数且 $R \in [0, 1]$ 。当 (x_1, x_2) 来自禁忌表单 $Tabu$ 时,从 $Tabu$ 中随机选择两个位置为 0 的节点;当 (x_1, x_2) 来自 S 时,随机从 S 中选择两个不相邻的节点。

4 实验和结果分析

实验采用 MATLAB2021 编程,PC 内存为 64GB, Intel(R), Core(TM) i7-11800H CPU。为了验证该算法的有效性,实验采用多个公开 TSPLIB 实例进行测试,并将本算法和原始遗传算法(GA)、离散灰狼优化算法(DGWO)和离散麻雀搜索算法(DSSA)进行对比(表 1)。在求解 TSP 问题的众多算法中,DGWO 和 DSSA 是目前较新颖的工作,DGWO 的效果明显强于 GA、ACO、PSO 等经典算法,而 DSSA 获得比离散蝙蝠算法^[10]和离散青

蛙算法^[11]更优的处理方案^[9]。三种对比算法的超参数均为默认参数。由于算法所采用的种群规模和迭代次数均不一致,为了统一对比,本文所有算法的种群规模均为 50,分别对每个实例求解 10 轮,每轮迭代 100 次。评价指标为 10 次运行中的最优求解路径长度 Best 和算法平均运行时长 Time(单位:秒)。在当前公开的最优路径长度数据中,均只保留到整数,因此本实验中的最优求解路径长度也采用

向下取整的方式保留到整数位。此外,计算不同算法最优求解路径长度和 TSPLIB 提供方案的差异,得到两者偏差的百分比 Gap(单位:%)。

$$\text{Gap}=\frac{\text{Best}-S_0}{S_0}\times 100\%$$

其中,Best 表示算法最优求解路径长度,S₀ 表示 TSPLIB 公开的最优方案。指标 Gap 反映算法逼近最优解的能力,其值越小越好。

表 1 实验结果对比

实例	S ₀	GA			DGWO			DSSA			Ours		
		Best	Gap	Time	Best	Gap	Time	Best	Gap	Time	Best	Gap	Time
a280	2579	2767	7.29	75	2648	2.675	401.5	2639	2.326	126.6	2620	1.59	88.1
bier127	118282	120465	1.846	25	118660	0.32	99.6	118911	0.532	24.8	118703	0.356	28.6
ch150	6528	6704	2.696	69	6594	1.011	136.7	6551	0.352	34.2	6552	0.368	28.8
d198	15780	16185	2.567	81	15961	1.147	194.7	15957	1.122	60.2	15819	0.247	49.6
eil101	629	682	8.426	18	646	2.703	47	647	2.862	14.3	642	2.067	25.0
gil262	2378	2532	6.476	193	2431	2.229	359.5	2438	2.523	112.7	2429	2.145	33.2
kroA150	26524	28368	6.952	85	26740	0.814	108.7	26568	0.166	33.8	26749	0.848	27.8
kroB150	26130	27205	4.114	68	26381	0.961	108.5	26288	0.605	33.8	26213	0.318	27.7
kroA200	29368	30473	3.763	144	29738	1.26	201	29600	0.79	62.6	29562	0.661	30.2
kroB200	29437	31192	5.962	126	29945	1.726	200.1	29866	1.457	62.3	29771	1.135	30.5
lin318	42029	45146	7.416	329	43035	2.394	545	43038	2.401	170.8	42609	1.38	36.3
linhp318	41345	45119	9.128	188	43028	4.071	550	43042	4.104	172.1	42754	3.408	36.8
pr107	44303	44459	0.352	161	44391	0.199	80	44337	0.077	15.2	44301	−0.005	29.5
pr152	73682	74255	0.778	78	73683	0.001	157	73702	0.027	33.1	74019	0.457	27.9
pr264	49135	52102	6.038	166	49372	0.482	508.9	49747	1.246	106.6	49203	0.138	35.5
pr299	48191	51520	6.908	485	49345	2.395	545.4	49277	2.254	146.8	48632	0.915	38.8
rat195	2323	2413	3.874	79	2382	2.54	193.9	2377	2.325	59	2362	1.679	30.2
ts225	126643	130001	2.652	137	127103	0.363	273.4	127229	0.463	79.1	126645	0.002	46.1
tsp225	3916	4102	4.75	107	3942	0.664	322	3951	0.894	83.1	3926	0.255	37.9
u159	42080	44032	4.639	88	42075	−0.012	125.1	42254	0.413	38.1	42075	−0.012	40.1
rand400	14722	15617	6.079	364	15281	3.797	931.3	15176	3.084	285.2	15065	2.33	43.8
d493	35002	36680	4.794	140	36143	3.26	1409.9	35994	2.834	454.3	35772	2	55
pcb442	50778	54137	6.615	78	52684	3.754	1674.5	52185	2.771	336	52170	2.741	75.4
pr439	107217	114258	6.567	123	109330	1.971	1150.3	109088	1.745	340	108413	1.115	50.1
rd400	15281	16298	6.655	157	15802	3.409	930.8	15700	2.742	280.3	15539	1.688	44.8
p654	34643	35738	3.161	557	34964	0.927	2516.3	34970	0.944	732.2	34826	0.528	196.3
rat575	6773	7274	7.397	170	7128	5.241	1971	7113	5.02	619.2	7045	4.016	58.8
rat783	8806	9507	7.96	430	9292	5.519	3860.9	9351	6.189	1239.4	9128	3.657	79.5
u574	36905	39710	7.601	275	38684	4.82	1989	38572	4.517	621	37963	2.867	59.0
u724	41910	44745	6.764	737	43779	4.46	3157	43763	4.421	1045.7	43333	3.395	73.4

表 1 为不同算法求解结果。与其他三种对比算法相比,本文提出的算法总体上获得了更好的解决方案。在 30 个 TSPLIB 实例中,本文算法获得 26 个更优的解决方案;在 pr107 和 u159 实例上,本文算法获得比 TSPLIB 提供的方案更好的处理结果;四种算法的平均偏差比为:Gap_{GA} : Gap_{DGWO} : Gap_{DSSA} : Gap_{Ours} = 3.788 : 1.539 : 1.447 : 1,本文算法提供的方案在指标 Gap 中获得了更低的偏差率,更接近公开方案。此外,在运行效率上本文算法

也同样优于其他三个算法。四种算法的平均运行时长比为:Time_{GA} : Time_{DGWO} : Time_{DSSA} : Time_{Ours} = 3.914 : 16.897 : 5.068 : 1,在求解问题效率上,本文算法得到大幅提升,即验证了改进 k-opt 方法的有效性,且原始遗传算法在效率上优于两种新型算法。图 4~图 6 展示了部分实例在本算法上求解的最优方案,观察发现明显不存在路径交叉的现象。

图 7~图 9 为四种算法在三个实例上的迭代收敛曲线。以图 7 中 a280 实例拟合曲线为例,通过对

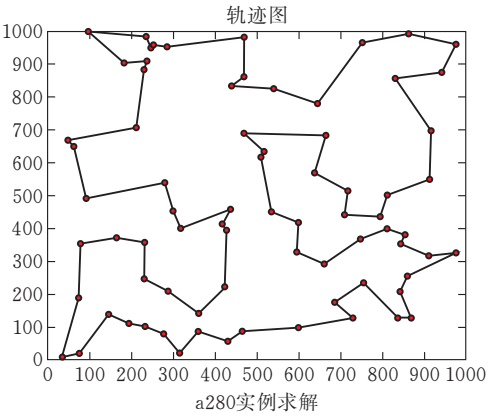


图 4 a280 最优路线

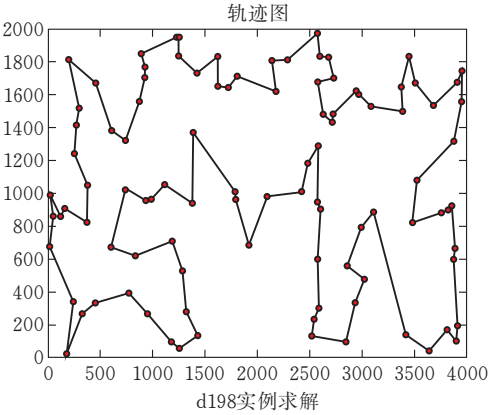


图 5 d198 最优路线

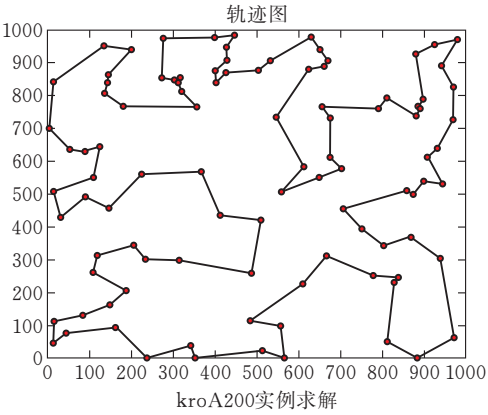


图 6 kroA200 最优路线

比四个算法的拟合曲线发现,原始遗传算法存在早熟问题,即算法过快收敛于局部最优解;而改进的算法即使在中期仍然能搜索出更优解。同时,改进的遗传算法前期强化种群全局搜索能力,后期加强种群局部搜索能力,故在整个搜索过程中能较好地平衡算法的探索 and 开发能力。与 DGWO 和 DSSA 相比,本文算法的前期寻优能力存在微弱的差距,但整体寻优能力存在优势。其根本原因在于:本文通过改进 k-opt 方法在前期增强算法探索能力,同时结合多种策略在当前解基础上添加扰动因子,在提高算法求解质量的同时能加快算法运行效率。

在上述对比实验的基础上,本文将原始 2-opt

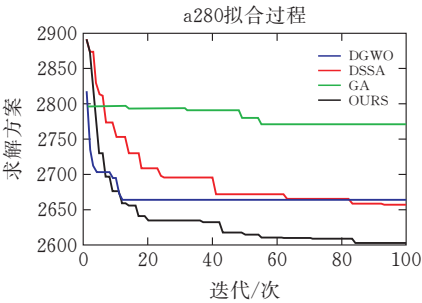


图 7 d198 迭代收敛曲线

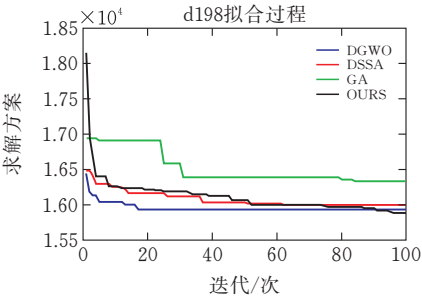


图 8 a280 迭代收敛曲线

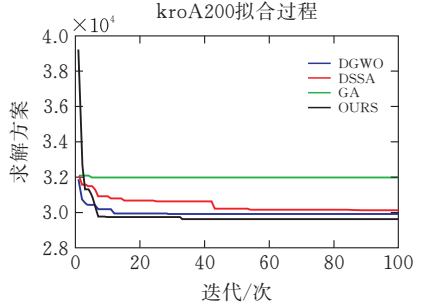


图 9 kroA200 迭代收敛曲线

方法和改进的 2-opt 方法进行对比。两种不同方法均采用改进的遗传算法作为基本框架,指标为 10 次运行中的最优求解路径长度 Best、算法最优求解路径长度和 TSPLIB 提供方案的差异 Gap、10 次平均求解路径长度 Avg、10 次最优方案的方差 Std 和算法平均运行时长 Time(单位:s)。实验在 14 个小型实例数据上进行测试,其他参数均采用上述方案。分析求解结果,在指标 Best 中,两个不同方法在 7 个实例上获得相同的最佳求解方案,原始 2-opt 方法获得 3 个更优的方案,改进的 2-opt 方法获得 4 个更优的方案;在指标 Avg 和指标 Std 中,原始 2-opt 方法分别获得 6 个更优的方案,改进的 2-opt 方法分别获得 7 个更优的方案;在指标 Time 中,改进的 2-opt 方法明显消耗更少的时间求解问题,两种方法的平均效率为 $\text{Time}_{2\text{-opt}} : \text{Time}_{2\text{-optimproved}} = 11.3 : 1$ 。综上所述,改进后的 2-opt 方法既降低了算法时间复杂度,也减缓了原始 2-opt 算法过早陷入局部最优解问题。

表 2 原始 k-opt 方法及其改进方法对比

实例	S ₀	2-opt					2-opt-improved				
		Best	Gap	Time	Avg	Std	Best	Gap	Time	Avg	Std
rand50	5553	5555	0.036	17.8	5555	0	5555	0.036	3.6	5555	0
rand75	7054	7053	−0.014	46.9	7083.7	29.5	7053	−0.014	4.3	7067.6	13.6
rand100	7891	7891	0	82.2	7961.7	60.8	7916	0.317	5.3	7971.4	30.7
eil51	426	431	1.174	22.6	432.6	1.7	428	0.469	3.6	430	1.8
eil76	538	552	2.602	47.7	553.5	1.6	546	1.487	4.3	553.3	3.5
kroA100	21282	21285	0.014	75.1	21469.6	137.3	21285	0.014	5.3	21373.5	42
kroB100	22141	22139	−0.009	61.6	22312.2	127.3	22139	−0.009	5.3	22359.9	182.2
kroC100	20749	20750	0.005	61.6	20941.4	133.7	20750	0.005	5.3	20864.8	78.8
kroD100	21294	21384	0.423	61.3	21467.7	51.6	21442	0.695	5.3	21513.2	49.6
kroE100	22068	22140	0.326	60.9	22165.9	66.5	22162	0.426	5.2	22222.6	43.6
pr76	108159	108444	0.264	40.6	108764.9	380.6	108159	0	4.4	108707.5	422
rat99	1211	1219	0.661	67	1224.7	5.5	1219	0.661	5.1	1231.5	7.3
rd100	7910	7911	0.013	68.3	7942.3	28.1	7910	0	5.2	7970	32.8
st70	675	677	0.296	34.1	680.1	4.6	677	0.296	4.2	679.7	3.7

5 总结

本文提出一种求解旅行商问题的改进 k-opt 遗传算法。该算法改进原始遗传算法以增强算法探索能力,同时融合改进的 k-opt 方法强化局部搜索,既能加速算法运行效率,又能减缓算法陷入局部最优解。具体地,采用交换、逆转和插入三种策略使得种群个体在更广阔的范围搜索;添加禁忌表有效增加搜索;随机置换序列顺序;利用深度搜索和分层结构优化当前解。实验采用多个公开数据集进行测试,通过对比离散灰狼算法(DGWO)、离散麻雀搜索算法(DSSA)和原始遗传算法(GA),该算法在求解 TSP 问题时收敛精度和收敛效率均有良好的表现。虽然本文在求解 TSP 问题时获得了较优解和较快运行效率,但将 TSP 问题迁移到超高规模(如 TSP-art)时,该算法的求解方案和已知最优路径还存在一定差距。如何在超高规模问题中求解 TSP 问题将成为下一阶段的研究目标。

[参 考 文 献]

[1] 冯志雨,游晓明,刘升.分层递进的改进聚类蚁群算法解决 TSP 问题[J].计算机科学与探索,2019,13(08): 1280-1294.

[2] 赵鑫,杨雄飞,钱育蓉.改进的蚁群优化算法求解旅行商问题[J].计算机工程与设计,2022,43 (04):962-968.

[3] 郭文忠,陈国龙.求解 TSP 问题的模糊自适应粒子群算法[J].计算机科学,2006(06):161-162 .

[4] 何锦福,符强,王豪东.求解 TSP 问题的改进模拟退火算法[J].计算机时代,2019(07):47-50.

[5] 张立毅,高杨,费腾,等.求解旅行商问题的搜寻者遗传算法[J].数学的实践与认识,2019,49(07):115-122.

[6] 谢聪.求解 TSP 问题的改进离散蝴蝶优化算法[J].数学的实践与认识,2020,50(01):173-182.

[7] PANWAR K, DEEP K. Discrete Grey Wolf Optimizer for symmetric travelling salesman problem [J]. Applied Soft Computing, 2021, 105: 107298.

[8] ZHANG Z, YANG J. A Discrete Cuckoo Search Algorithm for Traveling Salesman Problem and Its Application in Cutting Path Optimization[J]. Computers & Industrial Engineering, 2022, 169: 108157.

[9] ZHANG Z, HAN Y. Discrete sparrow search algorithm for symmetric traveling salesman problem[J]. Applied Soft Computing, 2022, 118: 108469.

[10] OSABA E, YANG X S, DIAZ F, et al. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems[J]. Engineering Applications of Artificial Intelligence, 2016, 48: 59-71.

[11] HUANG Y, SHEN X N, YOU X. A discrete shuffled frog-leaping algorithm based on heuristic information for traveling salesman problem[J]. Applied Soft Computing, 2021, 102: 107085.

[12] CINAR A C, KORKMAZ S, KIRAN M S. A discrete tree-seed algorithm for solving symmetric traveling salesman problem[J]. Engineering Science and Technology, an International Journal, 2020, 23 (04): 879-890.

[13] ZHONG Y, WANG L, LIN M, et al. Discrete pigeon-inspired optimization algorithm with Metropolis acceptance criterion for large-scale traveling salesman problem[J]. Swarm and Evolutionary Computation, 2019, 48: 134-144.

[14] LIN S, KERNIGHAN B W. An effective heuristic algorithm for the traveling-salesman problem[J]. Operations research, 1973, 21(02): 498-516.

[15] HELSGAUN K. General k-opt submoves for the Lin-Kernighan TSP heuristic[J]. Mathematical Programming Computation, 2009, 1(02): 119-163.

[16] 吴虎胜,张凤鸣,李浩,等. 求解 TSP 问题的离散狼群算法[J]. 控制与决策, 2015, 30(10): 1861-1867.

An Improved Kopt Genetic Algorithm for Travelling Salesman Problem

ZHAO Tao, YE Zhiwei, ZONG Xinlu, PAN Hu

(School of Computer Science , Hubei Univ. of Tech., Wuhan 430068, China)

Abstract: The traveling salesman problem is a classical combinatorial optimization problem, and the genetic algorithm has a good global search ability to obtain the shortest path with better quality when dealing with this problem. However, the local search ability of the genetic algorithm is weak. In order to enhance the local search ability of the genetic algorithm, accelerate the operation efficiency of the algorithm, and slow down the phenomenon of the algorithm falling into premature maturity, an improved kopt genetic algorithm for solving the traveling salesman problem is proposed in this paper. The algorithm uses the improved kopt method to initialize to obtain a better population, introduces an improved crossover mutation mechanism to enhance the global search ability of the algorithm, and combines the improved kopt method to strengthen the local search ability. The experimental results show that the improved kopt genetic algorithm can effectively balance the algorithm's exploration and exploitation abilities, and its solution has good quality and high operation efficiency.

Keywords: traveling salesman problem; kopt; genetic algorithm; local search; combinatorial optimization problem

[责任编辑: 张岩芳]

(上接第 33 页)

Sorting Model of Camellia Fruit Shells and Tea Seeds Based on ResNet18 Network

DONG Geng¹, WANG Yanqing^{1,2}, SUN Jiwei¹, DUAN Yufei^{1,2}

(1 Agricultural Machinery Engin. Research and Design Institute, Hubei Univ. of Tech., Wuhan 430068, China ;

2 Hubei Agricultural Machinery Equipment Intelligent

Hubei Agricultural Machinery and Equipment Intelligent Engineering , Wuhan 430068,China)

Abstract: This paper proposes a sorting model based on the improved ResNet18 model for the sorting of camellia fruit shells and tea seeds, which is a difficult problem for the industrialization of camellia fruit. The experimental sample images were acquired and processed by a camellia fruit sorting machine, and the model was further optimized by comparing the effects of different batch sizes, learning rates, activation functions, and optimizers. The experiments show that the improved ResNet18 model has a significant reduction in the loss function compared with the unimproved one, and the accuracy of the validation set is improved, with the average accuracy of the validation set increasing from 97.03% before to 97.21% currently, while the accuracy of the final test set increases from 97.25% to 97.75%, an improvement of 0.5 percentage points, and the higher accuracy can meet the needs of camellia fruit sorting.

Keywords: deep learning; camellia fruit; sorting; ResNet18 model

[责任编辑: 闫 品]

(上接第 57 页)

Parallel Compound Current Control Strategy for LCL SAPF

WU Ke^{1,2}, JIANG Yunhao^{1,2}, BU Jianyi^{1,2}, FAN Chao³, DENG Wei³, XU Anfei^{1,2}, YUAN Lei^{1,2}

(1 Hubei Collaborative Innovation Center for High-efficiency Utilization of Solar Energy,

Hubei Univ. of Tech., Wuhan, 430068, China ;

2 Xiangyang Industrial Institute of Hubei Univ. of Tech., Xiangyan 441100, China ;

3 The Second Construction Co., Ltd. of China Construction Third Engin. Bureau, Wuhan 430074, China)

Abstract: The traditional controller has poor dynamic performance and low steady-state tracking accuracy, which cannot adjust the harmonic current in real time. In this paper, a shunt compound current control strategy based on an LCL active power filter is proposed. The fuzzy PI controller is combined with the fast repetition controller to improve the dynamic response speed and steady-state compensation accuracy of the system. The detailed design scheme is also given. In order to improve the voltage regulation speed of the dc-side capacitor, fuzzy PI control is designed for capacitor voltage control. Finally, the effectiveness and superiority of the proposed control strategy are verified by simulation.

Keywords: LCL shunt active power filter; combined current control; adaptive fuzzy pi control; fast repetition control; parallel control

[责任编辑: 张岩芳]