

旅行商问题的一种高效近似算法及其排考应用

袁 志

(广州大学 华软软件学院, 广东 广州 510990)

摘 要: 要解决现实应用中的类似于小规模旅行商问题的组合优化问题, 提出了一种新的近似算法——群体竞争搜索算法(PCS). 使用了一种新的局部搜索算子 $2e/1p-opt$, 并按照“搜索-选优-变异-搜索”的迭代策略设计了群体竞争搜索算法. 与 CLK 算法进行了比较, 对于 280 城市以下的旅行商问题, 在计算速度和计算结果两方面都有较好表现. 将高校排考中的场次排序优化问题转化为 TSP 并用算法求解, 能在 1 秒内得到大幅改进的结果.

关键词: 旅行商问题; 局部搜索; 群体进化算法; 排考

中图分类号: TP3

文献标识码: A

文章编号: 1000-7180(2018)01-0056-05

An Efficient Approach for Small-scale TSP and its Application in Examination Timetabling

YUAN Zhi

(South China Institute of Software Engineering, Guangzhou University, Guangzhou 510990, China)

Abstract: To solve the practical combinatorial optimization problem similar to small-scale TSP, an efficient algorithm named Population Competitive Search(PCS) is presented. A new local search operator named $2e/1p-opt$ is used, and a algorithm is designed based on the iterative strategy of “search-select-mutate-search”. Compared with CLK on TSP instances with city numbers not more than 280, PCS has better performance in speed and results. PCS is applied for examination timetabling and achieves greatly improved results in 1 second.

Key words: travelling salesmen problem; local search; evolutionary algorithms; examination timetabling

1 引言

TSP 是 NP 难问题, 主要采用启发式算法和群体智能算法求近似解. 其中, Lin-kernigan 算法和链式 LK 算法^[1] (Chain Lin-Kernighan, CLK) 是公认的高效启发式搜索算法, 对小规模 TSP 实例能快速得到最优解. 自 CLK 算法提出后, 研究者大多转向研究大规模 TSP, 使用 Lin-kernigan 算法或链式 LK 算法进行启发式搜索, 然后用进化算法或群体智能算法提高全局优化能力^[2].

就我们所知, 目前还没有其他公开算法的速度和结果在小规模 TSP 上优于 CLK 算法. 但 CLK 实现难度较大, 当遇到应用变更时, 难于进行适应性修

改. 考虑到实际应用中的多数组合优化问题(如高校排考)类似于小规模旅行商问题, 研究既容易编程实现和变更, 又具有良好寻优能力的 TSP 算法仍然具有必要性.

排考是高校教务管理的重要工作, 主要任务是安排各门课程的考试时间. 文献[3]已经证明排考中的时间表问题是 NP 难问题. 通常, 考试场次是一个不大的数, 因此, 排考中的场次排序问题可以转化为小规模 TSP. 与 TSP 的不同之处在于, 场次排序问题是找一条最短路径而非最短环路.

本文研究一种快速且易于实现和变更的 TSP 算法, 将其用于高校排考优化. (1) 提出了一种新的启发式搜索算子 $2e/1p-opt$, 该算子用两边交换或

单点移动进行局部优化,当城市数 $n < 1000$ 时,大约在 $(12 \pm 6)n$ 次搜索后终止,得到的结果包含约 80% 的最优边。(2)我们发现,当 $2e/1p\text{-opt}$ 停止搜索时,对其结果实施一次两边交换后再次执行 $2e/1p\text{-opt}$,有很大的概率使结果得到进一步优化,基于这一观察,提出了“搜索-选优-变异-搜索”的迭代优化策略,按此策略设计了群体竞争搜索算法(PCS).在城市数在 280 以下的 TSPLIB 测试例上与 CLK 算法进行了比较,结果表明 PCS 算法在计算速度和计算结果上都有更好表现。(3)通过增加一个虚拟场次,使场次排序问题等价地转换为最短环路问题,然后将 PCS 应用于高校排考中的场次排序优化,在笔者所在高校近几年排考实践中,都能在 1 秒内得到较优结果。

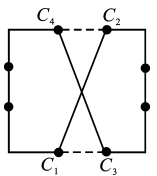
2 局部搜索算子 $2e/1p\text{-opt}$ 及其寻优能力

解 S 编码为城市号的序列,例如 $S = (0, 1, 2)$ 表示环路是 $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow c_0$,用 $\text{len}(S)$ 表示环路的长度,TSP 的目标是使 $\text{len}(S)$ 最小。

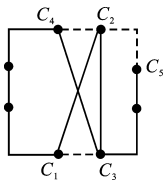
2.1 局部搜索算子 $2e/1p\text{-opt}$

局部搜索的思路是:依次检查 S 中的每一个位置,对于该位置上的城市 c_1 ,依次检查它的邻域(离它最近的 6 个城市,记为 $N(c_1)$)中的每一个城市 c_2 ,尝试用 c_2 替换 c_1 的后一个城市,变更后得到 S' ,如果 S' 优于 S ,则用 S' 替换 S .当连续检查 n 个位置都无法优化环路时停止搜索。

我们使用两种替换方法,第一种方法是两边互换,第二种方法是单点移动,合称为 $2e/1p\text{-opt}$,如图 1。



(a) 两边交换



(b) 单点移动

(---代表被删除的边,——代表新增的边)

图 1 两边交换或单点移动

图 1 中, c_1 是当前检查的城市, c_3 是 c_1 的后一个

城市, c_2 是 $N(c_1)$ 中的城市, c_4 和 c_5 分别是 c_2 的后一个城市和前一个城市. 两边交换通过转置 (reverse) c_2 至 c_3 之间的次序来实现,单点移动通过移动 (move) c_2 到 c_1 和 c_3 之间来实现。

例如: $S = (0, 1, 2, 3, 4, 5, 6)$ 且 $c_1 = 1, c_2 = 2, c_3 = 5$. 两边交换得 $S' = (0, 1, 5, 4, 3, 2, 6)$, 操作的结果是删除了 2 条边并新增了 2 条边,环路的长度增量按式(1)计算. 单点移动得 $S' = (0, 1, 5, 2, 3, 4, 6)$, 操作的结果是删除 3 条边并新增 3 条边,环路的长度增量按如下计算。

```
2e/1p-opt 伪代码
index1=0;
notImprove=0;
while notImprove<n;
    c1=S[index1];
    index3=(index1+1)%n;
    c3=S[index3];
    foreach(city c2 in NC[c1])
        index2=S.Indexof(c2);
        index4=(index2+1)%n;
        index5=(index2-1)% n;
        c4=S[index4];
        c5=S[index5];
        if len_2e < 0
            reverse(S, index2, index3)
            S.len+=len_2e
            notImprove=0;
        else if len_1p < 0
            move(S, index2, index3)
            S.len+=len_1p
            notImprove=0
        endfor
    notImprove++;
    index1=(index1+1)%n
endwhile
```

2.2 $2e/1p\text{-opt}$ 的寻优能力

从时间代价和解的质量两个方面来评价 $2e/1p\text{-opt}$ 的寻优能力。(1)时间代价. $2e/1p\text{-opt}$ 依次检查环路上的每一个位置,检查的位置数决定了时间代价。(2)解的质量. TSP 的很多测试例已经有公认的最佳解,文献[4]通过统计发现:Lin-Kernighan 算法得到的结果,一般含有约 80% 的最优边(最优解中所含的边),我们也用最优边比例来评价解的质量。

在 TSPLIB 上的测试例上运行 $2e/1p\text{-opt}$,对每个测试例产生 800 个解,当停止搜索时取两项数据:(1)累计检查的位置数(search locations);(2)解中所含的最优边比例(optimal percent),如表 1

所示.

表 1 2e/1p-opt 停止时检查的位置数和结果中的最优边比例

instance	ATT48	EIL51	KROA100	EIL101	PR144	A280	LIN318	Att532
Search locations	206	218	731	506	864	2 507	3 103	8 441
Optimal percent/%	88	85	84	81	76	79	78	84

我们发现,当城市数 $n<1000$ 时,2e/1p-opt 大约检查 $(12\pm6)n$ 个位置后停止,得到的解大约包含 80% 左右的最优边. 图 2 给出了 2e/1p-opt 在 EIL101, A280 和 Att532 上各得到的 100 个解中所含的最优边比例,可以看到,城市数越多,2e/1p-opt 得到的结果中,最优边的比例越向 80% 集中.

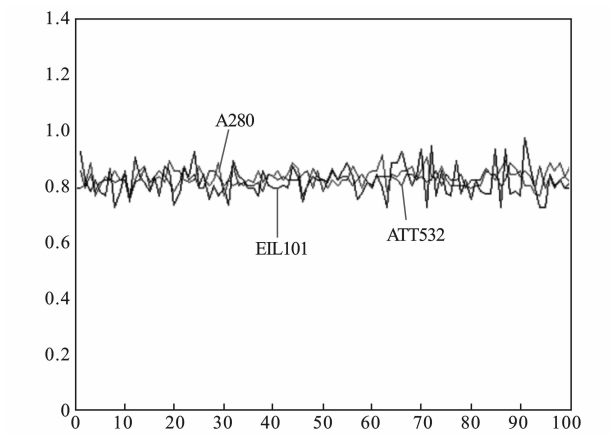


图 2 2e/1p-opt 的 100 搜索结果的最优边比例

3 以 2e/1p-opt 为核心的群体竞争搜索算法

3.1 搜索-选优-变异-搜索的迭代原理

以往研究者使用启发式搜索算法时,一旦搜索停止,就再次生成随机的初始解,然后再次进行启发式搜索,多次重启后取最好的结果作为最终解. 我们发现了另一种更为有效的方法,与重启策略每次都从头开始不同,我们利用了前期的搜索结果,每次都在更好的基础上进行深入搜索.

我们发现,当 2e/1p-opt 停止搜索时,对所得到的局部最优环路,执行一次两边交换(2e-switch),然后再执行 2e/1p-opt,可能使局部环路得到较大改善. 图 3 中,(a)是 2e/1p-opt 得到的一个结果,(b)是一次 2e-switch 的结果,(c)和(d)是再次执行 2e/1p-opt 的结果,环路得到了进一步优化.

显然,不是每次 2e-switch 后再执行 2e/1p-opt 都能得到更好的结果. 为此,我们对不同的边尝试 2e-switch 和 2e/1p-opt 操作,得到更好的结果后,再

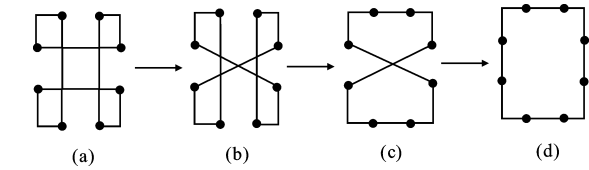


图 3 2e/1p-opt after 2e-switch

次尝试 2e-switch 和 2e/1p-opt 操作. 这一策略称为“搜索-选优-变异-搜索”. 为了扩大选优的范围,我们用群体搜索来替代单个搜索,提出了群体竞争搜索算法.

2e-switch 伪代码

```
select index1 from S randomly
select index2 form S randomly except index1
reverse(S, index1, index2)
S, len+=len_2e
```

3.2 群体竞争搜索算法思想和流程

受自交植物进化规律的启发,我们提出群体竞争搜索算法(PCS). 自交植物(如豌豆)通过闭花传粉来繁殖,不与其他植株杂交,种群进化主要依赖于基因突变. 在自然选择作用下,优良个体获得繁衍能力,劣质个体被淘汰,使种群朝着有利方向进化. 受资源限制,一个种群总是维持稳定的规模,单纯依靠基因突变则进化速度过慢,我们加入人工因素:(1)用局部搜索操作加快优化速度;(2)选最优且互异的个体作为下一代的父本;(3)在个体繁衍时,对子代个体实施 1 次随机的两边交换,实现基因变异.

与遗传算法相比,群体竞争搜索的特异性在于:(1)用搜索优化取代杂交操作;(2)用最优且互异的若干个体作为父代个体,取代概率性选择;(3)用固定尺度的变异替代概率性变异.

用图 4 来说明群体竞争搜索的原理,图 4 中的横坐标是问题的解,纵坐标是解的适应值. 如果一些解经过搜索后得到相同的局部最优解,则将它们集中在一个区域,称为“同底区域”,同底区域中的谷底是局部最优解. 个体经过搜索后都会到达谷底,选择

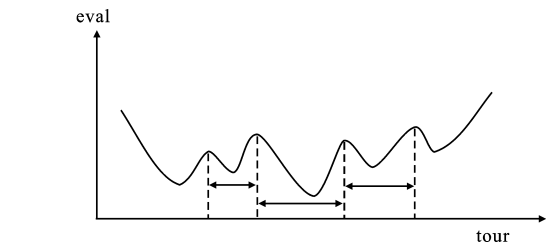


图 4 同底区域

最优的若干个谷底作为父代个体,父代个体经过克隆变异后得到若干子代个体,一部分子代个体将到达新的同底区域,经再次搜索到达新的谷底,通过迭代不断逼近最优解.

PCS 算法

- Step1:初始化算子.种群规模 p ,选择比例 θ (θ 值建议为 0.2),随机生成若干个初始个体.
- Step2:循环直至满足终止条件.
- Step2.1 搜索算子.对每一个个体执行 $2e/1p$ -opt.
- Step2.2 选择算子.按适应值对个体排序,保留 p 个排名靠前且互异的个体,删除其余个体.如果互异个体少于 p ,则随机新生成若干个个体加入种群.
- Step2.3 克隆变异算子.竞争中被保留的个体,分别产生 $(1-\theta)/\theta$ 个克隆个体,对每一个克隆个体进行 1 次随机的两边交换.
- Step3:输出群体中的最优解.

3.3 实验

选用 TSPLIB 中的小规模测试例($n<100$)进行实验,做 2 项对照:一是对照 PCS 算法与 CLK 算法的收敛速度;二是在 PCS 算法中分别采用 2 -opt 和 $2e/1p$ -opt 两种局部搜索操作,在相同的计算时间里对照计算结果.其中,CLK 算法直接采用 Concorde 上的源代码,PCS 用 C# 来编程实现.

(1)PCS 与 CLK 比较

PCS 运行 100 次,每次使其收敛到最优解,取平均收敛时间(秒).结果表明:PCS 能 100%收敛到最优解,且速度与 CLK 算法相近甚至更快,如表 1 所示.

表 1 PCS 与 CLK 比较

Instance /optimal	CLK		PCS	
	Result.	Time	Result	Time
ATT48/33 522	33 522	0.22	33 522	0.015
EIL51/426	426	0.12	426	0.016
KROD100/21 294	21 294	0.33	21 294	0.078
EIL101/629	629	0.24	629	0.125
PR144/58 537	58 537	1.43	58 537	0.397
A280/2 579	2579	6.54	2 579	1.268

(2)采用 2 -opt 和 $2e/1p$ -opt 的 PCS 算法之间的比较

在 PCS 算法中,分别采用 2 -opt 和 $2e/1p$ -opt 两种局部搜索操作,各运行 100 次,每次允许运行的最长时间设置为 10 秒,比较平均解(Result)、收敛到最优解时的平均迭代次数(Iterations)、运行时间.结果表明, $n<100$ 时,PCS with 2 -opt 同样能得到最优解,但收敛速度明显差于 $2e/1p$ -opt;当 $n>$

100 时,PCS with 2 -opt 多数情况下得不到最优解,如表 2 所示.

表 2 两种局部搜索的 PCS 比较

Instance	PCS with 2 -opt			PCSwith $2e/1p$ -opt		
	Result	Iterations	Time	Result	Iterations	Time
ATT48	33 522	15	0.062	33 522	4	0.015
EIL51	426	31	0.14	426	5	0.016
KROD100	21 294	33	0.22	21 294	10	0.078
EIL101	630.1	185	1.61	629	18	0.125
PR144	58 537	168	1.43	58 537	26	0.397
A280	2602	263	6.54	2 579	38	1.268

通过 PCS 的流程与算子设计的说明,以及实验对照,可以看到 PCS 在计算速度、计算结果、可编程性和易变更性上都有明显优势.

4 PCS 在排考中的应用

排考的输入是学生选课数据,如果两门课程含有相同的学生则称它们为互斥课程,否则称它们为相容课程.排考的约束条件和优化目标是:(1)一个考试场次中的所有课程彼此必须相容;(2)总场次尽量少;(3)学生连续考试的人次尽量少.

4.1 工作基础

计算机辅助排考一般分两个阶段来进行.第一阶段进行场次分配,将课程分成不同的场次;第二阶段进行场次排序.

(1)场次分配

场次分配问题可表述成图的最小着色问题,贪心算法是一种常用的图着色算法^[5].首先生成一个空场次,从待分配课程中找到相斥课程最多的一门课程,分配进该场次,然后不断寻找与该场次相容但与待分配课程相斥最多的课程,分配进该场次,当再也找不到与该场次相容的课程时,生成一个新的空场次,进行下一轮分配.场次分配的结果用图 5 表示,其中的边表示课程相斥,边的权值是相同学生数.

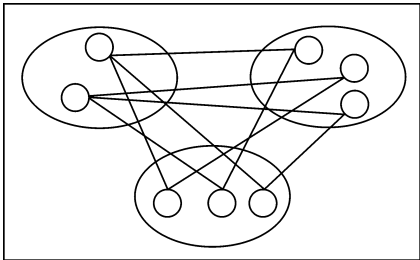


图 5 将课程分配到考试场次

贪心算法所得结果,后分配的课程一定与先分配的场次互斥,但先分配的课程可能与后分配的场次相容,虽然可能导致总场次少量增加,但是为报考人员提供了调整的空间.研究者还提出了用遗传算法进行场次分配^[6],在场次数上的表现与贪心算法相差不大.

(2)场次排序

排序是对所有的场次形成考试时间表.文献[7]给出了一种贪心算法:首先取一个冲突值最小的场次放入解 S ,然后每次从剩余场次中取出一个场次直到所有场次都取完.每一个新增到 S 中的场次,要求它放置在 S 中的最合适情况下,能使连考人次最少.贪心算法的结果离全局最优相差很远.

4.2 PCS 用于考试场次排序优化

场次排序的输入是场次集合 C 和冲突矩阵 D ,其中, $C = \{c_1, c_2, \dots, c_n\}$, $D = \{d_{i,j}\}$.与 TSP 的不同之处在于,场次排序问题是找一条最短路径而非最短环路.

我们增加一个虚拟场次 c_{n+1} ,并将虚拟场次与其他所有场次的冲突值设为 0,然后对 $C = \{c_1, c_2, \dots, c_{n,n+1}\}$ 求最短环路,得到结果后,从 c_{n+1} 处解开环路,得到的路径即为所求.经过处理,排序优化问题等价地转换为 TSP 问题.

以笔者所在高校作为实验对象,该校有 13 500 学生,每学期约有 200 门左右的课程,学生平均每学期修读 5 门课程.用贪心算法进行场次分配,每学期大概形成 23 到 33 个考试场次.

分别用 PCS 算法和贪心算法进行场次排序,PCS 总能在 1 秒内得到比贪心算法好 20% 以上的结果.由于篇幅的限制,不能完整给出每个学期的场次排序问题的输入和输出.从 2016 年春学期取 10 个场次的冲突矩阵作为场次排序问题的输入.

从表 3 可以看出,贪心算法得到的排序串为“1, 6, 2, 5, 9, 7, 10, 4, 8, 3”,连考人次 1 681. CRA4EBC 得到的排序串为“5, 2, 10, 9, 6, 1, 4, 8, 3, 7”,连考人次 1 274.

5 结束语

为解决现实应用中的类似于小规模旅行商问题的组合优化问题,提出了一种新的旅行商问题算法——群体竞争搜索算法(PCS).使用了一种新的局部搜索算子 $2e/1p-opt$,并按照“搜索-选优-变异-搜索”的迭代策略设计了群体竞争搜索算法.与 CLK 算法进行了比较,对于 280 城市以下的旅行商

表 3 10 个场次的关联矩阵

D_{ij}	1	2	3	4	5	6	7	8	9	10
1	—	—	—	—	—	—	—	—	—	—
2	302	—	—	—	—	—	—	—	—	—
3	895	866	—	—	—	—	—	—	—	—
4	74	209	539	—	—	—	—	—	—	—
5	124	139	567	71	—	—	—	—	—	—
6	57	88	360	64	42	—	—	—	—	—
7	111	66	406	51	66	128	—	—	—	—
8	166	271	532	87	105	208	100	—	—	—
9	137	281	514	109	55	131	68	222	—	—
10	434	838	1 337	420	495	404	235	928	1 161	—

问题,在计算速度和计算结果两方面都有较好表现.将高校排考中的场次排序优化问题转化为 TSP 并用算法求解,能在 1 秒内得到大幅改进的结果.很多组合优化问题都可以转换为小规模旅行商问题来求解,对于这一类问题,可望在保持本文算法基本流程的前提下,设计出相应的优化算法.

参考文献:

[1] Martin O C, Otto S W, Felten E W. Large-step markov chains for the traveling salesman problem [J]. Complex Systems, 1991, 5(3): 299-326.

[2] 程毕芸,鲁海燕,徐向平,等. 求解旅行商问题的改进局部搜索混沌离散粒子群优化算法[J]. 计算机应用, 2016, 36(1): 138-142.

[3] Schaert A. A survey of automated timetabling[J]. Artificial Intelligence Review, 1999, 13(2): 87-127.

[4] 邹鹏,周哲,江贺,等. 球旅行商问题的局部循环搜索算法的运行时间和性能分布分析[J]. 计算机学报, 2008, 29(1): 92-99.

[5] 崔丽. 高校学分制环境下排考问题的改进图算法[J]. 计算机系统应用, 2015, 24(3): 220-225.

[6] 李红婵,朱颢东. 采用十进制免疫遗传算法求解高校排课问题. [J]. 系统工程理论与实践, 2012, 32(9): 2031-2036.

[7] 蔡木生. 高校自动排考算法的设计与实现[J]. 计算机工程与应, 2010, 46(24): 69-72.

作者简介:

袁 志 男, (1971-), 副教授. 研究方向为网络与信息系统. E-mail: mynameisyz@163. com.