

# 基于分治的物流路径规划算法

文 / 杨一凡

**摘要：**为提高物流配送中心路径规划的效率，缩短路径规划算法的求解时间，本文重点考察以旅行商问题（TSP）为基础的路径规划场景，提出了一种基于分治框架的路径规划方法，先将规模较大的TSP分解为一系列子问题，再使用基于Transformer的模型求解子问题，最后将子问题的解合并，以实现同一模型求解不同规模TSP，减少模型训练成本。本文比较了不同分治框架和不同子问题解法带来的差异。实验表明，通过3种改进的分治框架，机器学习模型能够应用于规模相差较大的TSP上，取得出色表现。

**关键词：**物流运输、路径规划、旅行商问题、分治、强化学习、注意力机制

## 一、引言

目前，物流公司使用的路径规划方案以传统启发式算法为主<sup>[1]</sup>。随着深度学习和强化学习技术的发展，越来越多的研究将机器学习模型应用于物流运输场景。这些研究发现，相比传统启发式算法，机器学习模型能够大幅提高路径规划效率<sup>[2]</sup>。但是，由于训练后的模型泛化能力有限，求解问题规模和训练问题规模无法存在太大差距，否则模型求解的准确度将会显著退化。而如果训练不同的模型以适应问题规模变化，其引入的硬件和时间成本将使算法不具有落地的可行性<sup>[3]</sup>。

为解决这一痛点，本文重点考察以TSP为基础的路径规划场景，提出了一种基于分治框架的TSP求解方法，先将大规模TSP分解为一系列相同规模的子问题，再使用基于Transformer<sup>[4]</sup>的模型求解子问题，然后将子问题的解合并，得到完整解。

TSP可被描述为：给定二维平面

上一组城市及其距离矩阵，求解访问每个城市一次并最终回到初始城市的最短距离（构成环路），优化目标即为环路总路径长度最短。

## 二、基于分治的 TSP 求解算法

分治法的思想由来已久，很多高效算法均建立在这个思想的基础之上，例如快速排序、归并排序、快速傅里叶变换等。只要确保划分与合并的正确性，分治法往往能够成为解决复杂问题的利器。将分治思想引入

TSP求解的第一个研究来自Karp。

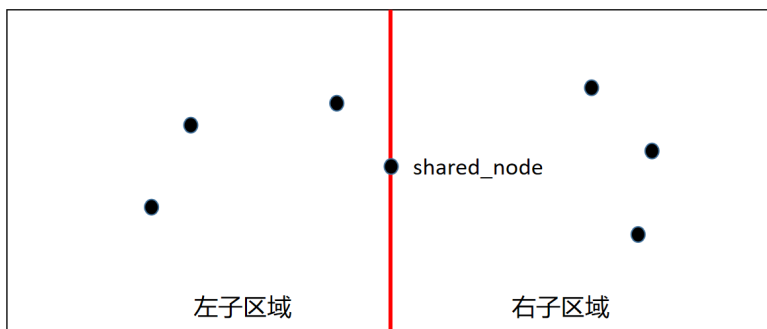
### 1.Karp 分治

Karp提出的<sup>[5]</sup>TSP分治算法非常简洁，以图1为例。平面上随机分布着7个物流站点，其中位于左右分界线上的物流中转站点称为shared\_node。步骤（1）~（4）为分割，（5）~（6）为合并。

（1）比较当前物流覆盖区域横向跨度和纵向跨度孰大孰小（图中纵向小于横向）。

（2）将区域中的物流站点按照

图1 Karp 分割



跨度更大的方向的坐标排序（图中按照横坐标排序）。

(3) 找到中点，即为两个区域的shared\_node，以shared\_node为分界，沿着跨度更小的方向分割（图1中沿红线纵向分割）。

(4) 在分割后的两个子区域内重复1~3步骤（图中在分割成的左右两个子区域内递归调用此算法）。

合并时围绕shared\_node展开，考察shared\_node在两个子区域内4个邻接的物流站点。

(5) 分别计算可能的4种合并方案带来的收益。

① 连接1和3（图2中c），删去shared\_node与1和3的路径（图中a、b），收益为 $a+b-c$ ；

② 连接1和4，删去shared\_node与1和4的路径，计算收益；

③ 连接2和3，删去shared\_node与2和3的路径，计算收益；

④ 连接2和4，删去shared\_node与2和4的路径，计算收益。

(6) 采用收益最大的合并方案进行合并

图3与图4为将Karp的划分与合并算法整体应用于TSP实例<sup>[5]</sup>的情况。

尽管在Karp算法之后，Valenzuela等人提出了EDAC算法<sup>[6]</sup>，但Karp算法能够提供严格限制子问题所含站点数量的稳定分割，而EDAC算法则较为灵活，不能提供固定大小的子问题分割。由于机器学习模型的泛化能力所限，严格限制子问题规模的Karp算法更适宜应用在分治框架下。

## 2. 划分子问题

针对Karp算法的划分部分，本文提出了一种较为激进的改进方法，放弃基于shared\_node的划分方法，不再寻找物流中点，以图5为例。

(1) 比较当前物流覆盖区域横向跨度和纵向跨度孰大孰小（图中纵

图2 Karp 合并

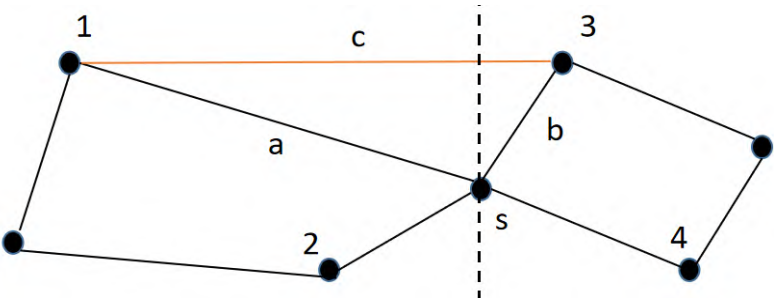


图3 问题实例划分并求解后

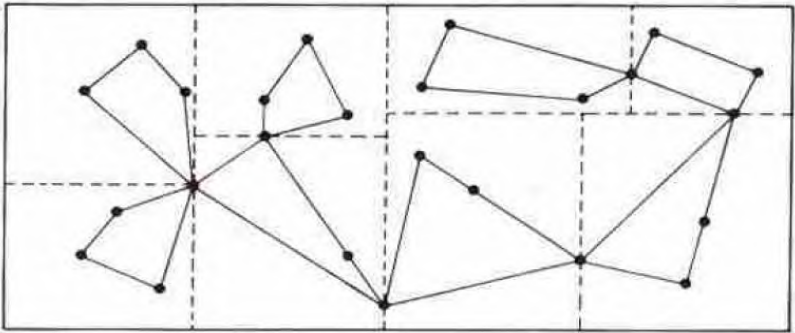


图4 问题实例合并后

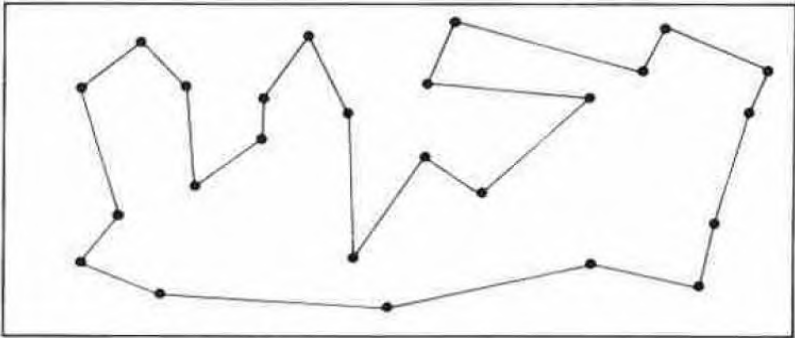
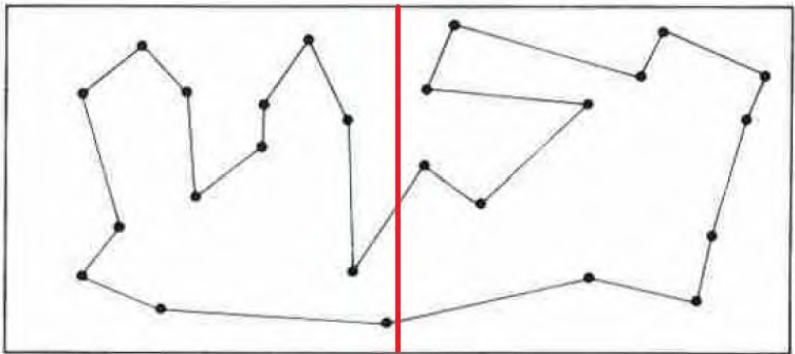


图5 改进后的划分算法（舍弃 shared\_node）



向跨度小于横向)。

(2) 将区域中站点按照跨度更大的方向的坐标排序(图中按照横坐标排序)。

(3) 找到中间的两个站点,以两个站点之间的区域为界,沿着跨度更小的方向分割(图5中沿纵向分割)。

(4) 在分割后的两个子区域内重复1~3步骤(图5中是在分割后的左右两个子区域内递归调用此算法)。

### 3. 子问题求解

目前Transformer已经取代RNN成为自然语言处理任务的默认选项,基于Transformer构建的模型也在各个研究方向取得了突破性进展。

为此,本文采用基于Transformer注意力机制的预训练模型<sup>[7]</sup>求解子问题。该模型在Transformer的基础上,修改了编码、解码所需的参数和模型训练方式,成功将Transformer的注意力机制应用于TSP求解,极大缩短了TSP

求解时间。

在子问题求解步骤中,将所有节点的坐标矩阵作为输入,预训练模型能够输出节点的序列,即子TSP的解。

### 4. 合并子问题

针对Karp算法的合并环节,本文给出了三种改进后的合并算法,第一种较为保守,仍基于shared\_node进行合并,但扩大了合并时的搜索范围。第二和第三种较为激进,因为划分时就放弃使用shared\_node,故合并时也采用新的方法进行合并。第二种将各个子区域理解为城市节点组成一个新的TSP进行合并,第三种将各个子区域理解为树节点组成一个最小生成树问题(MST)进行合并。

#### (1) 扩大搜索范围的合并

Karp算法在合并时只允许搜索shared\_node周围的4个物流站点,在4个点之间寻找最优的合并方案,将搜索范围扩大将能够使算法具备更好的全局性。

图6 没有做交换的环路

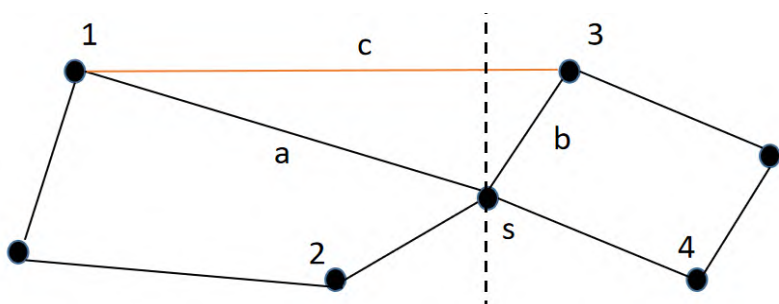
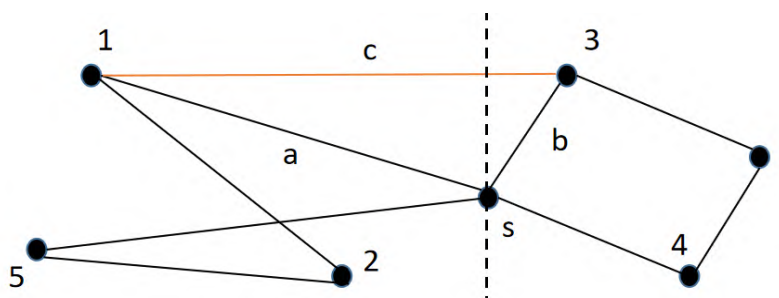


图7 交换 shared\_node 与 1 后的环路



为了实现搜索范围的扩大,首先需要定义“交换”操作,此“交换”操作是对2-OPT算法<sup>[8]</sup>中“交换”操作的简化。

交换:改变相邻两个站点的访问顺序,其他站点的访问顺序不变。

例如,访问顺序为[1,2,3,4,5],交换3和4后,访问顺序变为[1,2,4,3,5]。不妨设3为shared\_node,交换前,与3相邻的两个点为2和4;交换后,与3相邻的两个点为4和5,这允许合并算法检查到shared\_node周边且不与shared\_node相邻的更多点。

下面给出改进后的合并算法:

①不作任何交换,分别计算可能的4种合并方案带来的收益(此时shared\_node与左侧区域的邻接点为1和2),找出最优合并方案,记为方案一。

a: 连接1和3(图6中路径c),删去shared\_node与1和3的路径(图中a、b),收益为 $a+b-c$ ;

b: 连接1和4,删去shared\_node与1和4的路径,计算收益;

c: 连接2和3,删去shared\_node与2和3的路径,计算收益;

d: 连接2和4,删去shared\_node与2和4的路径,计算收益。

②交换shared\_node与1(如图7),得到新的左侧访问顺序(此时shared\_node与左侧子区域的邻接站点为1和5),重复步骤1的计算,找出交换后的最优合并方案,记为方案二。

③交换shared\_node与2,得到新的左侧访问顺序,重复步骤1的计算,找出交换后的最优合并方案,记为方案三。

④交换shared\_node与3,得到新的右侧访问顺序,重复步骤1的计算,找出交换后的最优合并方案,记为方案四。

⑤交换shared\_node与4,得到

新的右侧访问顺序，重复步骤1的计算，找出交换后的最优合并方案，记为方案五。

⑥在5个方案中找出合并后总路径最短的合并方案，作为最终合并方案。

(2) 基于 TSP 的合并

放弃shared\_node后，该改进的合并算法分为三个步骤：

①将每个子区域看作一个物流站点，计算每个子区域的中心点作为站点坐标。

②求解子区域中心点构成的新TSP，将此TSP的解作为合并子区域的顺序。

③按照步骤2所得顺序合并子区域，合并子区域的算法由参考文献《Algorithm 1 Approximate and Merge Tiles》给出<sup>[9]</sup>。

(3) 基于 MST 的合并

与上一种改进不同，这种算法没有将子区域看作一个新的旅行商问题，而是看作一个新的MST进行求解：

①将每个子区域看作一个物流站点，计算每一个站点与相邻站点之间合并的代价，合并的代价矩阵即为

MST的边权矩阵。

②利用边权矩阵求解MST，由于子区域之间的邻接关系是稠密的，而Prim算法对边的数量不敏感，故采用Prim算法求解MST。

③以MST所包含的边为顺序，合并子区域。

两种合并方法的区别在于，上一种方法中，按照TSP的线性访问顺序，每一个子区域只能和周围的两个子区域进行合并，而此方法允许一个子区域和周围多个子区域合并，从而获得更大的搜索范围。

5. 方法应用

通过本文提出的改进后分治框架，基于机器学习的求解方法能够具备以下几个优势：

(1) 可用于求解超大规模 TSP

随着问题规模增长，传统算法的求解时间将急剧退化，带来难以接受的时间成本，使用基于分治的方法，能够将大规模问题分解为小规模问题，实现小模型求解大问题的效果，在成本可控的范围内求解超大规模TSP。

(2) 降低时间成本

对于一个问题实例，传统算法的

时间成本全部来自求解过程，而基于机器学习的算法通过将时间成本分摊转移至模型训练过程，利用机器学习“一次训练，多次使用”的特点，达到降低总时间成本的效果。

(3) 可推广

目前，机器学习模型之所以难以直接应用于物流领域，是因为泛化能力不足，在特定规模问题实例上训练的模型无法应用于规模相差过大的其他问题实例，而改进后的分治框架能够将一个大问题分解为多个既有训练模型能够求解的小问题，将模型动态配置到不同的问题规模上，助力机器学习模型推广到物流领域。

三、实验结果与分析

本文使用随机生成的数据集，应用市场上两个成熟的TSP求解器LKH<sup>[10]</sup>和Concorde<sup>[11]</sup>建立Baseline，并将基于分治的方法与之作比较，从而得出结论。

1. LKH 与 Concorde 比较

表1中TSP-20表示包含20个城市节点的TSP，依此类推。

在求解时间上，可以从表1的“Time”看出，LKH的求解时间对问

表1 LKH 与 Concorde 比较

		TSP-20	TSP-50	TSP-100	TSP-200	TSP-300	TSP-500
LKH	Time(s)	0.68	3.26	4.72	5.51	6.42	8.39
	Gap(%)	0.00	0.00	0.00	0.00	0.01	0.06
Concorde	Time(s)	0.01	0.05	0.24	1.46	4.96	20.3
	Gap(%)	0.00	0.00	0.00	0.00	0.00	0.00

表2 预训练模型表现数据

		TSP-20	TSP-50	TSP-100	TSP-200	TSP-300	TSP-500	TSP-1K	TSP-2K	TSP-3K	TSP-5K
TSP-20	Gap (%)	0.30	4.53	15.87	30.04	37.94	50.06	67.33	86.40	98.12	114.05
TSP-50	Gap (%)	0.71	1.85	4.70	13.45	20.30	30.01	45.89	62.78	73.21	88.22
TSP-100	Gap (%)	7.09	4.23	4.30	8.20	13.05	21.33	34.32	50.33	60.19	74.10



题规模增长不敏感,适用于求解较大规模问题以缩短实验数据采集的时间成本。而Concorde则在小规模TSP的求解时间有极其出色的表现,当问题规模超过500后,其求解时间退化明显。

在求解质量上,表1的“Gap”表明,LKH会随着问题规模的增大而与Concorde之间出现差距,在较大规模TSP中,LKH解的质量将比Concorde差。

2. 预训练模型表现

从表2中得出,TSP20、TSP 50、TSP 100三种模型在求解各自对应的问题规模时,Gap分别为0.3%、1.85%、4.3%,符合Kool[7]给出的实验结果。

然而,受制于泛化能力,随着问题规模的增长,模型的表现急剧退化(如图8,大规模问题中模型Gap均超过100%),可见无法直接将小规模模型应用于大规模TSP。

3. 三个系列

(1) 求解质量

实验设计7种求解方案,分别是:未改进的分治框架+Concorde求解(Concorde-div)、保守改进的分治框架+Concorde求解(Concorde-div-conservative)、未改进的分治框架+预训练模型求解(model-div)、基于TSP改进的分治框架+Concorde求解(Concorde-div-TSP)、基于MST改进的分治框架+Concorde求解(Concorde-div-MST)、基于TSP改进的分治框架+预训练模型求解(model-div-TSP)、基于MST改进的分治框架+预训练模型求解(model-div-MST)。

求解质量如图9、图10、图11,在Concorde-div中,由于子问题求解使用Concorde求解器,故其Gap全部由分治框架带来,可用于说明分治框架的表现。而model-div中,Gap由

图8 预训练模型表现变化趋势

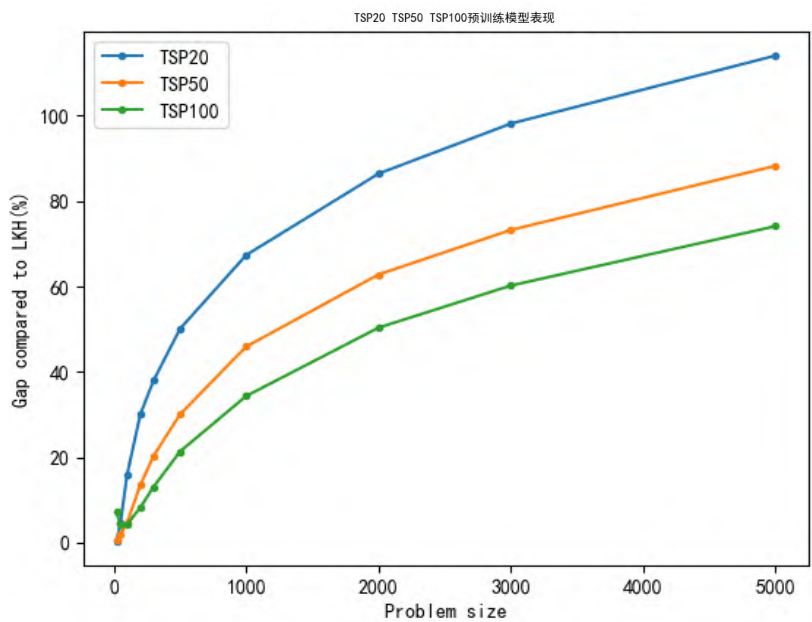


图9 TSP20 系列求解质量

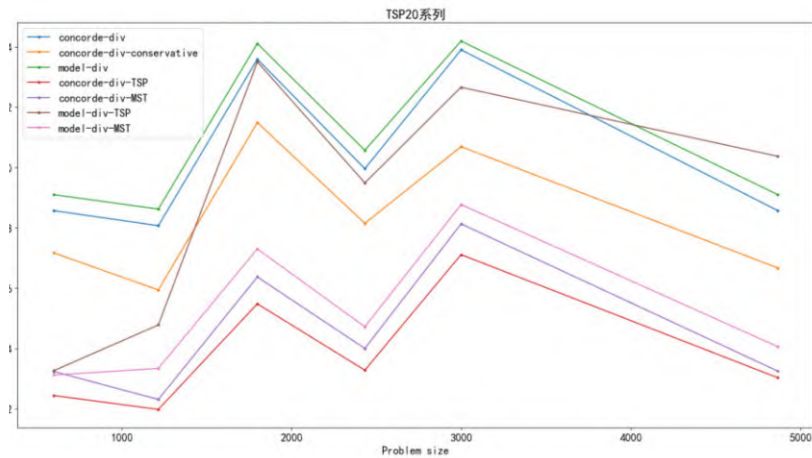


图10 TSP50 系列求解质量

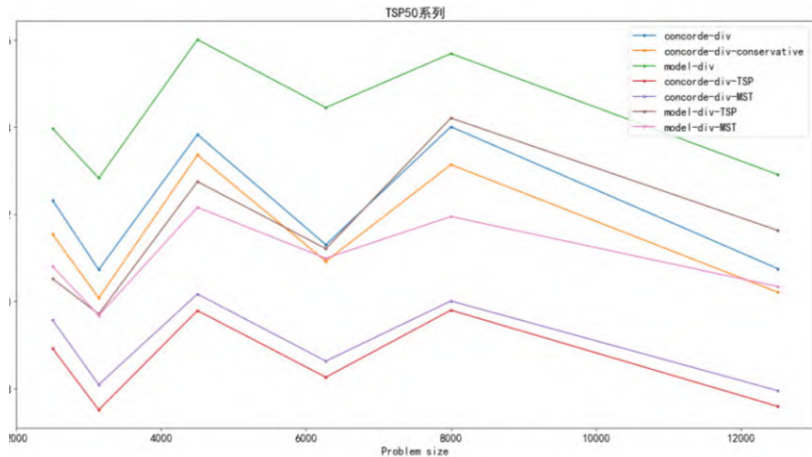
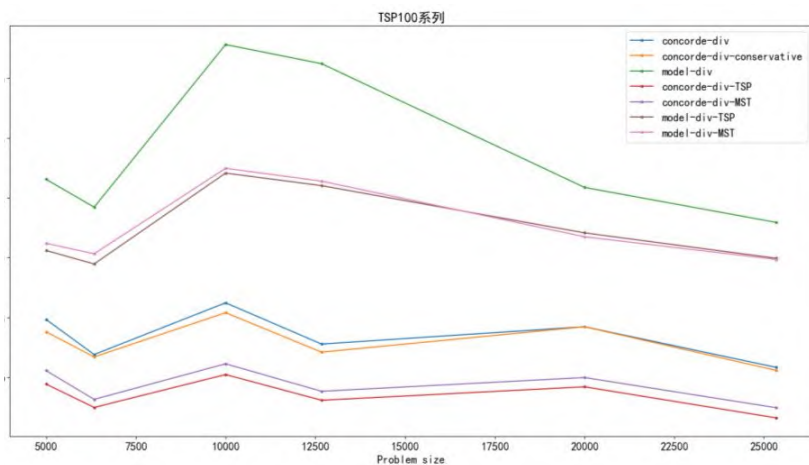


图11 TSP100 系列求解质量



两部分构成，预训练模型（model）和分治框架。其中，sub-Gap计算了每个子区域上model的Gap，可用于说明model的表现。由此可得，Concorde-div的Gap与model-div的sub-Gap共同构成了model-div的div-Gap。

三个系列之间横向来看，随着规模增大，分治框架表现变好，分治框架带来的Gap：TSP100系列 < TSP50系列 < TSP20系列；而model表现变差，model带来的Gap：TSP100系列 > TSP50系列 > TSP20系列。

随着规模增大，model表现变差主要是由于分割成的子区域数量增

大，每个子区域内的城市分布愈发偏离均匀分布，超出了模型训练时的预期。针对这个问题，可以考虑在训练模型时采用分割后区域与均匀分布区域混合作为训练数据，而非只有均匀分布的训练数据。

TSP20、TSP 50、TSP 100三个系列各自内部纵向来看，随着规模增大，分治框架和model表现都较为稳定，没有随问题规模增长而退化。值得注意的是，d=6、d=7、d=8三个能够严格划分为对应子问题的实例，相对于不能够严格划分的问题实例，Gap有小幅改进。可见，针对问题的规模，选择能够使得严格划

分均匀的子问题大小将会得到最佳表现。

应用保守改进算法后（Concorde-div-conservative），分治框架本身带来的Gap平均缩小了1%左右。

应用基于TSP的改进算法后（XXX-div-TSP），分治框架本身带来的Gap平均缩小了4%左右。

应用基于MST的改进算法后（XXX-div-MST），分治框架本身带来的Gap平均缩小了3.5%左右。

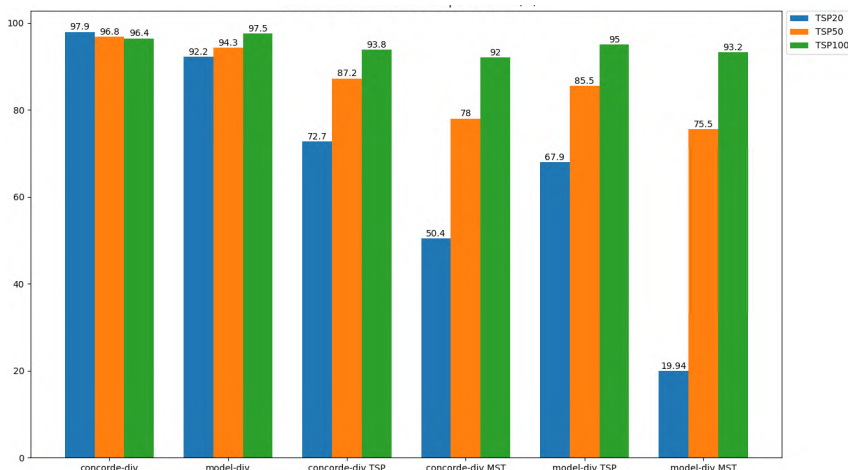
在TSP20、TSP 50等规模较小的系列中，基于MST的改进算法比基于TSP的改进算法更优；在TSP100这样规模更大的系列中，基于TSP的改进算法则优于基于MST的改进算法。

## （2）求解时间

图12三个系列中，使用分治框架后，求解时间相比LKH的Baseline有明显改进。

在使用分治框架的方法内部比较，求解器采用Concorde还是model需要视划分后的子问题规模而定。在TSP20、TSP50系列中，内层使用Concorde求解的分治方法比使用model更快，即若划分后的子问题规模较小，则使用Concorde最快。而若问题规模过于庞大，以TSP100

图12 TSP20、TSP50、TSP100 系列求解时间



本文主要着眼分治框架的改进，给出了保守改进、基于TSP的改进和基于MST的改进3种改进算法，提出了一种基于分治的不同规模的TSP求解方法。

系列为例，需要划分为100及以上的子问题大小，则使用model求解更快。

四、总结

物流领域中，尽管已经有研究将机器学习模型应用于路径规划，但由于在小规模问题实例上训练的模型无法直接泛化到大规模问题上，且直接

在大规模问题实例上训练模型的效率低成本高，目前机器学习模型无法应用于问题规模不固定的实例中。本文认为，基于分治的求解方法成为一种值得尝试的思路，但现有的分治算法具有较强的局部性，特别是在合并时算法的搜索范围极为有限。本文主要着眼分治框架的改进，给出了保守改进、基于TSP的改进和基于MST的改

进3种改进算法，提出了一种基于分治的不同规模的TSP求解方法。实验结果表明，通过分治框架，机器学习模型能够应用于不同规模TSP上，取得出色表现的同时极大缩短路径规划的求解时间。

（作者单位为浙江大学计算机科学与技术学院）

参考文献:

[1]李凯文,张涛,王锐,覃伟健,贺惠晖,黄鸿.基于深度强化学习的组合优化研究进展[J].自动化学报,2021,47(11):2521-2537.

[2]王扬,陈智斌,吴兆蕊,高远.强化学习求解组合最优化问题的研究综述[J].计算机科学与探索,2022,16(02):261-279.ENGEL P A. Impact Wear of Materials[M]. 2nd ed. New York: Elsevier, 1986.

[3]Bengio, Andrea and Antoine. "Machine Learning for Combinatorial Optimization: A Methodological Tour d'Horizon ". Eur.J.Oper.Res. 290(2021):405-421.

[4]Vaswani, Ashish, et al. "Attention is all you need. " Advances in neural information processing systems 30(2017).

[5]Richard M. Karp, (1977) Probabilistic Analysis of Partitioning Algorithms for the Traveling -Salesman Problem in the Plane. Mathematics of Operations Research 2(3):209-224.

[6]Valenzuela, Christine L., and Antonia J. Jones. "Evolutionary divide and conquer(I): A novel genetic approach to the TSP." Evolutionary Computation 1, no.4(1993):313-333.

[7]Kool, Wouter, Herke Van Hoof, and Max Welling. "Attention, learn to solve routing problems!." arXivpreprintarXiv:1803.08475 (2018).

[8]Lawler, Eugene L., and David E. Wood. "Branch-and-bound methods: A survey." Operations research 14.4 (1966): 699-719.

[9]Helsgaun, Keld. An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic. Diss. Roskilde University. Department of Computer Science, 2006.

[10]Lin Shen, and Brian W. Kernighan. "An effective heuristic algorithm for the traveling - salesman problem." Operations research 21.2 (1973): 498-516.

[11]Rezoug, Abdellah, Mohamed Bader-El-Den, and Dalila Boughaci. "Guided genetic algorithm for the multidimensional knapsack problem." Memetic Computing 10.1 (2018): 29-42.

本刊重点关注的物流展会

展会、交流名称	日期	展会地点	电话	联系人
第三十四届国际制冷、空调、供暖、通风及食品冷冻加工展览会	2023年4月7-9日	上海新国际博览中心	010-64934668	薛龙云、康琨
第二十三届中国零售业博览会	2023年4月19-21日	重庆国际博览中心	010-68013731	赵怡舟
2023第14届中国（广州）国际物流装备与技术展览会	2023年5月18-20日	广州·中国进出口商品交易会展馆D区	020-89617180	钟健龙
2023亚洲物流双年展（国际物流、交通运输及远程信息处理博览会）	2023年6月14-16日	上海新国际博览中心	021-20205583	陈倩
2023 LogiMAT China 国家内部物流解决方案及流程管理展览会	2023年6月14-16日	上海新国际博览中心	021-50816029	黄超
2023上海国际快递物流产业博览会	2023年7月5日-7日	上海新国际博览中心	021-67651295	华涛
名称:2023第10届郑州物流展览会暨第8届郑州展览会	2023年7月3日-5日	郑州国际会展中心	0371-86680039	吕腾飞
2023 CeMAT ASIA第23届亚洲国际物流技术与运输系统展览会	2023年10月24日-27日	上海新国际博览中心	021-2055 7119	丁牟海一