

一种求解 TSP 初始化种群问题的方法

李志宾¹, 侯世旺², 程厚虎¹

LI Zhibin¹, HOU Shiwang², CHENG Houhu¹

1. 中北大学 机械与动力工程学院, 太原 030051

2. 怀化学院 商学院, 湖南 怀化 418000

1. School of Mechanical and Power Engineering, North University of China, Taiyuan 030051, China

2. School of Business Administration, Huaihua University, Huaihua, Hunan 418000, China

LI Zhibin, HOU Shiwang, CHENG Houhu. Method for initial population of TSP. Computer Engineering and Applications, 2016, 52(17): 172-176.

Abstract: In order to improve the computational efficiency and solution accuracy of the TSP problem, Field Triangular Probability Choosing Adaptive Neighbor-hood Algorithm (FTPCANA) is proposed to solve the problem of initial population construction. To make the neighbor-hood radius adapt to the distribution of the city, a neighbor-hood radius adaptive function based on Sigmoid function is designed. Besides, the next city is selected by using the triangular probability selection model in the neighbor-hood, which can avoid the blind and random selection of the next station. The optimization of the storage operation sequence is selected as the research problem of TSP. Based on Matlab simulation, the initial population generated by the proposed algorithm is compared to the initial population generated by the neighbor-hood method. The two initial populations, one of which is generated by proposed algorithm, and the other is randomly generated, are respectively calculated by genetic algorithm. It is proved that the algorithm can quickly generate high quality initial population, which greatly improves the computational efficiency and accuracy of solving the TSP problem.

Key words: traveling salesman problem; initial population; neighbor-hood method; triangular probability; adaptive function

摘要: 为提升求解 TSP 问题的计算效率和求解精度, 针对初始种群构造问题进行研究, 提出了域内三角概率选择自适应邻域算法。为使邻域半径能够适应城市的分布情况, 设计了一种基于 Sigmoid 函数的邻域半径自适应函数; 为了避免在邻域内盲目随机地选择下一站城市, 提出了在邻域内利用三角概率选择模型选择下一个城市。以自动化立体仓库安排出入库作业顺序优化作为 TSP 研究问题, 通过 Matlab 仿真计算, 将该算法和邻域法生成的初始种群进行对比分析, 并分别用该算法和随机生成的初始种群作为遗传算法的初始种群进行计算。证明了该算法可快速生成高质量的初始种群, 大大提升了求解 TSP 问题的计算效率和求解精度。

关键词: 旅行商问题; 初始种群; 邻域法; 三角概率; 自适应函数

文献标志码: A **中图分类号:** TP301 **doi:** 10.3778/j.issn.1002-8331.1603-0142

1 引言

旅行商问题 (Traveling Salesman Problem, TSP) 是一种典型的复杂组合优化问题, 并广泛存在于诸多领域, 例如自动化立体仓库的出入库优化问题^[1-2], 数控加工中的刀具行走轨迹问题^[3-4], 贴片剂贴装优化问题^[5-6]等。人们在求解这类问题时常采用遗传算法、模拟退火

算法、蚁群算法等智能算法进行求解^[7-8], 但存在计算效率低和最优解精度差的问题, 随着城市数量的增加, 计算量会成指数级增长, 在城市规模较大时此类缺陷会表现得更加突出。求解 TSP 问题时, 初始化种群质量直接影响了整个计算的寻优速度和最优解精度^[9]。

罗辞勇等人^[10]用邻域法生成遗传算法的初始种群,

基金项目: 山西省青年科技研究基金项目 (No. 2013021021-2); 教育部人文社会科学研究青年基金项目 (No. 13YJC630049)。

作者简介: 李志宾 (1989—), 男, 硕士研究生, 研究方向为自动化控制、机械设计; 侯世旺 (1978—), 通讯作者, 男, 博士, 副教授, 研究方向为生产物流系统优化、制造质量控制、工业工程, E-mail: houshiwang@163.com; 程厚虎 (1986—), 男, 硕士研究生, 研究方向为自动化控制、机械设计。

收稿日期: 2016-03-09 **修回日期:** 2016-05-06 **文章编号:** 1002-8331(2016)17-0172-05

讨论了邻域半径常数 β 的影响,但未对邻域法进行改进,初始种群生成质量和算法计算效率有进一步提升的空间。莫海芳等人^[11]利用邻域法和随机生成的方法分别生成部分初始种群,保证了初始种群的多样性,但未充分发挥邻域法的优势。汪金刚等人^[12]利用自适应邻域算法生成遗传算法的初始种群,取得了较好的效果,但随机选取邻域半径常数过于盲目,难以保证初始种群质量。同时人们常直接随机生成智能算法所需的初始化种群,由于随机生成的初始种群质量较低且具有随机性,这就进一步降低了计算效率和最优解精度。

为提升求解TSP问题的计算效率和求解精度,针对初始种群构造问题进行研究。为了避免在邻域内盲目随机地选择下一站城市,同时又要兼顾到下一站城市选择的多样性,提出了在邻域内利用三角概率选择模型选择下一个城市;在邻域半径设定时,结合Sigmoid函数的特点,设计了一种能适应城市分布情况的邻域半径自适应函数,使邻域半径取值更加精准。在以上基础上本文提出了域内三角概率选择自适应邻域算法(Field Triangular Probability Choosing Adaptive Neighbor-hood Algorithm,FTPCANA)。为验证FTPCANA算法的有效性,以自动化立体仓库安排出入库作业顺序优化作为TSP研究问题,应用Matlab软件进行仿真试验,首先将FTPCANA算法生成的初始种群与 $\beta=2$ 的邻域法(Neighbor-hood Method,NM)生成的初始种群进行对比;再次将FTPCANA算法生成的初始种群和随机生成的初始种群分别作为遗传算法(Genetic Algorithm,GA)初始种群,并对自动化立体仓安排出入库作业顺序数学模型进行求解,并将结果进行对比。

2 FTPCANA 算法设计

2.1 邻域法寻优原理

邻域法是在最近邻法基础上发展而来的,最近邻法是解决旅行商问题最自然的启发寻优方式。在最近邻法中当从某一城市离开时,下一站是未曾去过的距离最近的城市,当城市规模较大时这样产生的城市路线往往局限于局部最优而无法达到全局最优。邻域法则是在最近邻法的基础上通过邻域的设置提高了全局搜索性能,在邻域法中,离开某个城市时,其下一站并不是未曾到过的最近城市,而是在以半径 r 为内圆和以半径 βr 为外圆的环形区域中选择下一站城市。其原理如图1所示,假设当旅行商从 C_8 开始旅行时,距离 C_8 最近的城市为 C_3 ,此时以 C_8 到 C_3 之间的距离为内圆半径 r ,以 βr (β 为任意常数且 $\beta \geq 1$) 为外圆半径,以城市 C_8 为圆心形成圆环形区域(图1中阴影区域),该环形区域为 C_8 邻域。该邻域内包含的城市分别为 C_3 、 C_4 和 C_7 ,下一站城市则在这三个城市中随机选择。之后重复以上步骤,直到遍历每一个城市。这种方法即为邻域法。

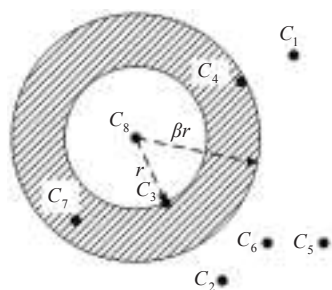


图1 邻域法原理图

2.2 FTPCANA 算法设计

邻域法相比较最近邻法的全局性有了一定的改善,但通过常数 β 设定的邻域半径无法适应城市的分布情况,这就导致邻域法最终不能跳出局部最优,并且在邻域内盲目随机地选取下一站城市显然降低了算法的效率。为了解决邻域法中存在的缺陷,提出了域内三角概率选择自适应邻域算法,针对邻域半径设定问题设计了一种基于Sigmoid函数的邻域半径自适应函数,针对邻域内城市的盲目选择引入了三角概率选择模型。

2.2.1 邻域半径自适应函数设计

邻域法中半径的取值非常关键,所期望的是邻域半径能够适应城市的分布情况,这就需要邻域半径呈现非线性变化,Sigmoid函数能更好地平衡线性和非线性之间的变化^[13],所以在这里采用单极性Sigmoid函数作为自适应函数,Sigmoid函数关系式如公式(1)所示,函数图像如图2,从图中可以看出 $f(x)$ 值域为 $(0,1)$,并且在接近极值处呈现非线性变化,而在中间阶段接近于线性变化。

$$f(x) = \frac{1}{1 + \exp(-\alpha x)} \quad (1)$$

其中 α 为正实数, α 取值的不同会对函数产生一定影响。

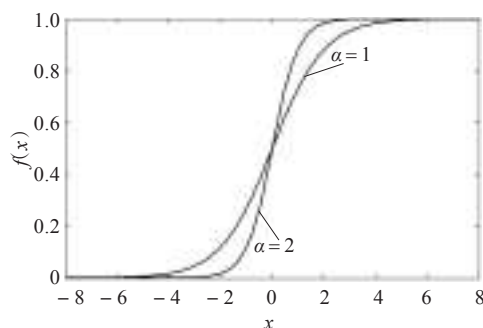


图2 单极性Sigmoid函数

针对求解旅行商问题,在这里以Sigmoid函数为基础构造自适应邻域半径函数,函数关系式如公式(2)所示。

$$r^{(t)} = \frac{1}{1 + \exp\left(\frac{r_{\text{avg}}^{(t)} - d_{\text{min}}}{d_{\text{avg}} - d_{\text{min}}}\right)} (r_{\text{max}}^{(t)} - r_{\text{min}}^{(t)}) + r_{\text{min}}^{(t)}, t = 1, 2, \dots, N \quad (2)$$

其中, N 表示城市最大数目, t 表示当前城市, $r_{\text{max}}^{(t)}$ 表示

未曾到过的城市与 t 之间的最大距离, r_{\min}^t 表示 t 与未曾到过的城市之间的最小距离, d_{\min} 表示所有城市之间的最小间距, d_{avg} 表示所有城市之间距离的平均值, r_{avg}^t 表示 t 与未曾到过的城市之间距离的平均值, r^t 表示城市 t 邻域的外圆半径。

2.2.2 邻域内三角概率选择模型

为了避免在邻域内盲目随机地选择下一站城市,将邻域内的城市按照与当前城市 t 之间的距离从小到大进行排序,邻域内所包含城市的数记为 n ,应用三角概率分布^[14]将已排序的城市赋予概率值 $P_i = 2(n+1-i)/(n(n+1))$,令 $P_0 = 0$,其中 i 表示邻域内城市的排列序号。这样,在邻域内距离城市 t 越近的城市被选中的概率越大,距离最近的城市被赋予的概率值为 $P_1 = 2/(n+1)$;距离城市 t 越远被选中的概率越小,距离最远的城市被赋予的概率值为 $P_n = 2/(n(n+1))$ 。通过产生一个随机常数 $k \in [0, nP_1 + (n-1)P_2 + \dots + P_n]$,当 $k \in [(i-1)P_1 + (i-2)P_2 + \dots + P_{i-1}, iP_1 + (i-1)P_2 + \dots + P_i]$ 时,选取邻域中的城市 t_i 为下一个城市。采用三角概率法选择下一站城市时,可以增大与城市 t 之间距离较近的城市选中概率,同时与城市 t 之间距离较远的城市被选中的概率依然存在,这样既增加了生成种群的多样性,又兼顾了局部最优的特性。

2.2.3 算法流程

FTPCANA 算法流程如下:

步骤1 对所有城市进行编码。

步骤2 随机选取起点城市。

步骤3 利用邻域半径自适应函数计算当前城市的邻域半径。

步骤4 确定邻域内的城市,并应用三角概率分布赋予邻域内每个城市概率。

步骤5 在概率区间内随机产生一个数字,依该数字所在城市概率区间选取下一个城市。

步骤6 计算未到城市数目,若大于1则步骤3,若等于1则步骤7。

步骤7 直接将最后一个城市作为下一站城市。

步骤8 输出遍历所有城市的路线。

3 出入库作业问题描述及数学模型

本文中研究的立体仓库为固定货架式立体仓库,当堆垛机进行单巷道作业时,堆垛机从出入库口进入巷道,依次到达待存取货位点进行货物存取,货物存取完毕后返回出入库口准备下一次作业。

对仓库调度系统作如下设定:货物总体积不超过周转箱容积,各货物存取时间相同;堆垛机水平、竖直方向运动独立,速度恒定,启停时间忽略;货格大小相同,长

度为 b ,宽度为 h ;每次任务结束堆垛机驻车位置为巷道口。

拣选作业时,由货物点由货位点 i 行走至货位点 j 路程为:

$$d_{ij} = |x_j - x_i| + |y_j - y_i| \quad (3)$$

堆垛机从出入库点 $(0,0)$ 到初始货位点 (x_s, y_s) ,以及从最后货位点 (x_e, y_e) 返回到 $(0,0)$ 的路程为:

$$d_{SE} = x_s + y_s + x_e + y_e \quad (4)$$

目标函数为遍历所有货位点的最小距离:

$$D_{\min} = \sum_W (|x_j - x_i| + |y_j - y_i|) + \sum_{SE} (x_s + y_s + x_e + y_e) \quad (5)$$

其中 W 表示货位点集合, SE 表示起始货位点和最后货位点。

4 Matlab 仿真分析

本文研究的自动化立体仓库为固定货架式立体仓库,单排货架分别由16层、75列货位格组成,货架及堆垛机具体参数如下: $b = 1\text{ m}$, $h = 1\text{ m}$ 。随机生成80条货单,各货位点的横、纵坐标如下: $\{(65,16)(56,12)(25,16)(35,15)(36,2)(4,15)(52,15)(25,8)(19,13)(41,8)(20,6)(34,2)(14,6)(5,9)(18,13)(14,13)(53,1)(14,16)(2,4)(54,3)(54,9)(36,8)(29,9)(69,3)(51,1)(67,3)(73,7)(67,12)(24,2)(58,16)(21,13)(46,16)(18,10)(67,3)(42,12)(53,2)(73,16)(17,6)(9,5)(23,12)(15,15)(21,8)(55,5)(28,1)(46,2)(5,4)(25,5)(22,4)(43,11)(57,9)(36,3)(2,10)(59,3)(68,4)(23,4)(58,12)(73,2)(72,3)(69,11)(39,7)(42,4)(46,16)(30,14)(71,15)(72,1)(45,3)(6,4)(1,15)(31,3)(60,7)(3,6)(51,2)(67,4)(59,14)(52,16)(19,13)(64,5)(37,7)(61,7)(66,3)\}$ 。

为验证该算法的有效性,从以下两方面进行研究:首先将FTPCANA算法与NM算法生成的初始种群对比;再次进行实例计算,分别用FTPCANA算法和随机生成的初始种群作为GA算法的初始种群(为便于书写,下文分别记为FTPCANA-GA算法和GA算法),采用Matlab R2012b进行自动化立体仓库安排出入库作业顺序优化仿真试验,并对试验结果进行分析。

4.1 FTPCANA 算法与NM 算法对比

分别将FTPCANA算法和 $\beta = 2$ 的NM算法连续生成50个种群规模为100的初始种群进行对比。初始种群数据对比见表1,FTPCANA算法随机生成的一个初始种群的出入库最优路线见图3,NM算法随机生成的一个初始种群的出入库最优路线见图4。

表1 FTPCANA 和NM 初始种群对比

算法	d_{\min}/m	d_{avg}/m	t/s
NM	459.7	604.1	4.5
FTPCANA	361.2	459.0	5.1

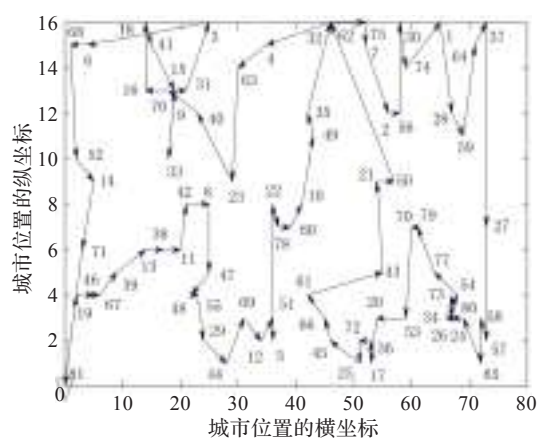


图3 FTPCANA 出入库最优路线图

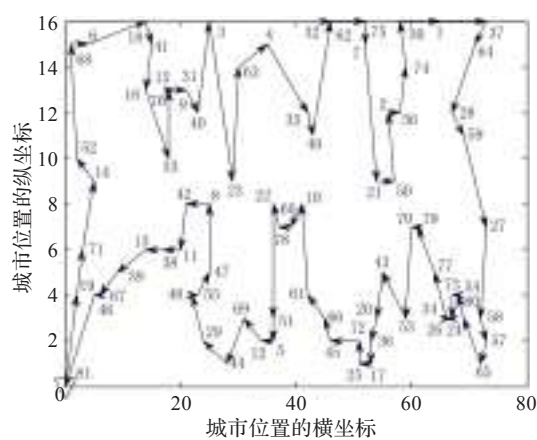


图5 FTPCANA-GA 出入库最优路线图

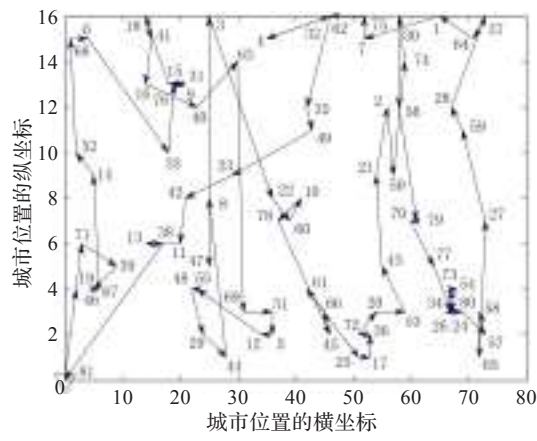


图4 NM 出入库最优路线图

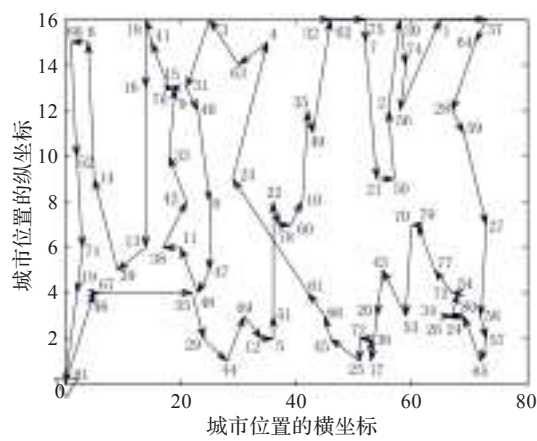


图6 GA 出入库最优路线图

表1中 d_{\min} 表示最优路线的平均路程, d_{avg} 表示初始种群中所有路线的平均路程, t 表示生成初始种群平均计算时间。由表1可得,FTPCANA相比NM初始种群最优路线平均路程减少21.4%,初始种群所有路线平均路程减少24.0%,计算时间稍有变长,但由于两种算法计算时间很短,故该影响可以忽略。

图3、图4中横坐标表示立体仓库巷道货架的列数、纵坐标表示层数,从图中可以看出,图3中交叉点明显少于图4,并且更加接近理想路线。

4.2 实例计算对比

分别用FTPCANA算法生成的初始种群和随机生成的初始种群作为GA算法的初始种群(即FTPCANA-GA和GA),对以上自动化立体仓安排出入库作业顺序优化数学模型进行求解,表2是FTPCANA-GA和GA连续计算50的数据结果对比,图5、图6为FTPCANA-GA和GA任意一次生成的最优路线,图7为FTPCANA-GA和GA任意一次迭代计算500次的优化计算过程。其中GA算法变异概率 $P_m=0.05$,交叉概率 $P_c=0.9$,最大迭代次数 $M=500$,种群规模 $N=100$ 。

表2 FTPCANA-GA 和GA 优化结果对比			
算法	n	d_{\min}/m	t/s
GA	—	303.8	173.1
FTPCANA-GA	100	256.0	146.8

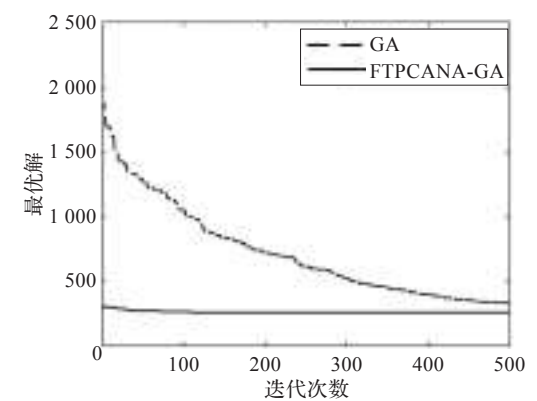


图7 FTPCANA-GA 和GA 优化计算过程

表2中 n 表示收敛迭代次数,“—”表示500次迭代计算中未收敛, d_{\min} 表示平均最优路线路程, t 表示收敛时的计算时间。由表2可知,FTPCANA-GA的平均收敛迭代计算次数为100,而GA在500次迭代计算内没有完全收敛,FTPCANA-GA算法100次迭代计算相比GA算法500次迭代计算的平均最优路线路程减少15.7%,并且FTPCANA-GA寻优时间相比GA寻优时间相比至少减少15%。这就表明FTPCANA生成的初始种群大大提高了GA算法的计算效率和最优解精度。

由图5、图6可以明显看出,图5中的路线图是没有交叉点的哈密尔顿(Hamilton)回路^[15],而图6中的交叉

点还相对较多,可见采用FTPCANA算法生成的初始种群效果明显优于随机生成的初始种群。

从图7可知,FTPCANA生成的初始种群的最优解已经非常接近全局最优,并且与GA算法经过500次迭代计算的最优解质量相当。而在GA算法中由于初始种群质量较差,导致计算效率低下、最优解精度较低。

5 结束语

本文通过设计一种基于Sigmoid函数的邻域半径自适应函数和在邻域内利用三角概率选择模型选择城市,提出了域内三角概率选择自适应邻域算法,并用该算法生成求解TSP问题的高质量初始化种群。通过Matlab仿真试验,以自动化立体仓库安排出入库作业顺序优化问题为例进行研究,试验表明该算法极大地提高了旅行商问题的初始化种群质量,明显提升了求解TSP问题智能算法的最优解精度和计算效率,对解决不同领域的TSP具有很广泛的推广意义。

参考文献:

- [1] 曾强,张泽斌,杨龙飞.有容量限制的自动化立体仓库堆垛机路径规划优化方法[J].机械设计与制造,2015,1(1):172-176.
- [2] 方彦军,唐猛.自动小车存取系统复合作业三维空间路径优化[J].计算机集成制造系统,2014,21(3):702-708.
- [3] 张礼兵,吴婷,袁根福,等.基于遗传算法的激光打孔路径优化[J].机电工程,2007,24(6):77-79.
- [4] 刘会霞,王霄,蔡兰.分层实体制造激光头切割路径的建模与优化[J].中国激光,2004,31(9):1137-1142.
- [5] 殷旅江,高亮,李登桥,等.改进元胞遗传算法的转塔式贴片机贴装优化[J].华中科技大学学报,2015,43(3):113-117.
- [6] 陈铁梅,罗家祥,杜娟,等.带扰动和变异因子的改进禁忌搜索算法求解贴片机贴装过程优化[J].控制与决策,2013,28(3):363-368.
- [7] 王剑文,戴光明,谢柏桥,等.求解TSP问题算法综述[J].计算机工程与科学,2008,30(2):72-74.
- [8] 周康,强小利,同小军,等.求解TSP算法[J].计算机工程与应用,2007,43(29):43-47.
- [9] 高海昌,冯博琴,朱利.智能优化算法求解TSP问题[J].控制与决策,2006,21(3):241-247.
- [10] 罗辞勇,卢斌,刘飞.一种求解TSP初始化种群问题的邻域法[J].重庆大学学报,2009,32(11):1311-1315.
- [11] 莫海芳,康立山.求解TSP的混合遗传算法[J].计算机工程与应用,2007,43(18):40-41.
- [12] 汪金刚,罗辞勇.求解TSP问题的自适应邻域遗传算法[J].计算机工程与应用,2010,46(27):20-24.
- [13] Mennon A, Mehrotra K, Mohan C K, et al. Characterization of a class of sigmoid functions with applications to neural networks[J]. Neural Networks, 1996, 9(5): 819-835.
- [14] 朱光宇.模因内三角概率选择混合蛙跳算法[J].计算机集成制造系统,2009,15(10):1979-1985.
- [15] Johnson D S, Mcgeoch L A. Local search in combinatorial optimization[M]//Local search in combinatorial optimization. London: John Wiley and Sons, 1997.
- [13] Levy R, Manning C D. Is it harder to parse Chinese, or the Chinese treebank?[C]//Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL). Stroudsburg: ACL, 2003: 439-446.
- [14] Klein D, Manning C D. Accurate unlexicalized parsing[C]//Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL). Stroudsburg: ACL, 2003: 423-430.
- [15] Chang P, Tseng H, Jurafsky D, et al. Discriminative reordering with Chinese grammatical relations features[C]//Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation. Boulder: ACL, 2009: 51-59.
- [16] Sheng Cai, Utiyama M, Sumita E, et al. Dependency-based pre-ordering for Chinese-English machine translation[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore: ACL, 2014: 155-160.
- [17] Kazemi A, Toral A, Way A, et al. Dependency-based reordering model for constituent pairs in hierarchical SMT[C]//Proceedings of EAMT. Antalya: EAMT, 2015: 43-50.
- [18] Li P, Liu Y, Sun M, et al. A neural reordering model for phrase-based translation[C]//Proceedings of the 25th International Conference on Computational Linguistics. Dublin: Coling, 2014: 1897-1907.
- [19] Chen D, Manning C D. A fast and accurate dependency parser using neural networks[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha: ACL, 2014: 740-750.

(上接165页)