

# 控制与决策

Control and Decision

## 求解旅行商问题的波动温控模拟退火算法

陈晟宗, 张纪会, 于守水, 郝为建

引用本文:

陈晟宗, 张纪会, 于守水, 郝为建. 求解旅行商问题的波动温控模拟退火算法[J]. *控制与决策*, 2023, 38(4): 911–920.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.0285>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 考虑随机旅行时间与二维装载约束的越库配送车辆路径优化

Optimization for two-dimensional loading constrained vehicle routing problem with cross-docking and stochastic travel time  
*控制与决策*. 2023, 38(3): 769–778 <https://doi.org/10.13195/j.kzyjc.2021.1605>

#### 基于自适应大规模邻域搜索算法的多车辆与多无人机协同配送方法

The cooperative delivery of multiple vehicles and multiple drones based on adaptive large neighborhood search  
*控制与决策*. 2023, 38(1): 201–210 <https://doi.org/10.13195/j.kzyjc.2021.2268>

#### 基于改进秃鹰搜索算法的同步优化特征选择

Simultaneous feature selection optimization based on improved bald eagle search algorithm  
*控制与决策*. 2022, 37(2): 445–454 <https://doi.org/10.13195/j.kzyjc.2020.1025>

#### 考虑学习效应的单人作业车间调度算法

Worker job shop scheduling algorithm considering learning effect  
*控制与决策*. 2022, 37(1): 37–46 <https://doi.org/10.13195/j.kzyjc.2020.0938>

#### 混合迭代贪婪算法求解准时生产分布式流水线调度问题

Hybrid iterated greedy algorithm for just in time distributed permutation flowshop scheduling problem  
*控制与决策*. 2022, 37(11): 3042–3051 <https://doi.org/10.13195/j.kzyjc.2021.0426>

# 求解旅行商问题的波动温控模拟退火算法

陈晟宗<sup>1</sup>, 张纪会<sup>1,2†</sup>, 于守水<sup>3</sup>, 郝为建<sup>3</sup>

(1. 青岛大学 自动化学院, 山东 青岛 266071; 2. 山东省工业控制技术重点实验室, 山东 青岛 266071;  
3. 青岛港国际股份有限公司, 山东 青岛 266011)

**摘要:** 针对传统模拟退火算法在求解旅行商问题时运行时间长, 易陷入局部最优, 且随着问题规模的增大缺陷愈发明显的问题, 对传统算法的内循环过程和退火机制进行改进, 使得内循环的搜索强度根据温度的变化自适应调整, 同时提出波动温度控制机制, 使得算法在保持温度幅值递减的总趋势下实现多次升温过程, 增强求解效果, 缩短求解时间, 并通过 TSPLIB 数据库提供的大量实例得以验证。

**关键词:** 模拟退火算法; 波动温控; 自适应内循环搜索; TSPLIB; 旅行商问题

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0285

引用格式: 陈晟宗, 张纪会, 于守水, 等. 求解旅行商问题的波动温控模拟退火算法[J]. 控制与决策, 2023, 38(4): 911-920.

## A simulated annealing algorithm with wave temperature control for the traveling salesman problem

CHEN Sheng-zong<sup>1</sup>, ZHANG Ji-hui<sup>1,2†</sup>, YU Shou-shui<sup>3</sup>, HAO Wei-jian<sup>3</sup>

(1. School of Automation, Qingdao University, Qingdao 266071, China; 2. Shandong Key Laboratory of Industrial Control Technology, Qingdao 266071, China; 3. Qingdao Port International Co., Ltd., Qingdao 266071, China)

**Abstract:** In view of the fact that the traditional simulated annealing algorithm has a long running time, easily been trapped in local optimum in solving the traveling salesman problem, especially the defects become more obvious with the increase of the problem scale, the inner loop and annealing mechanism of the traditional algorithm are improved, so that the search strength of the inner loop can be adjusted adaptively according to the change of temperature. At the same time, a fluctuation temperature control mechanism is proposed, so that the algorithm can realize multiple heating processes while maintaining the general trend of decreasing in temperature amplitude, which enhances the solution effect and shortens the solution time. It is verified by a large number of examples from the TSPLIB database.

**Keywords:** simulated annealing algorithm; wave temperature control; adaptive inner loop search; TSPLIB; the traveling salesman problem

## 0 引言

旅行商问题(the traveling salesman problem, TSP)是一个经典的组合优化问题,也是一个NP难问题,该问题广泛存在于许多领域,如电路布线、管道铺设、货物配送等问题经过简化处理后均可建模为TSP,因此,寻求TSP的有效求解方法具有重要的理论意义和实际价值.多年来研究人员为求解TSP提出了许多算法,文献[1-2]综述了求解TSP算法的研究进展,

将TSP的求解方法大致分为两类:精确算法和近似算法.精确算法包括分支定界法、动态规划法等,这类算法的优点是可以求出问题的精确解,缺点是通常只适合求解小规模问题.近似算法包括启发式算法和元启发式算法,近似算法的优点是求解速度快,但是求解易陷入局部最优,无法保证最终找到最优解.模拟退火算法是求解TSP的常用近似算法.由于在算法的迭代搜索过程中以Metropolis准则接受“劣解”,

收稿日期: 2022-02-23; 录用日期: 2022-06-24.

基金项目: 国家自然科学基金项目(61673228, 62072260); 青岛市科技局计划项目(21-1-2-16-zh).

责任编辑: 魏秀琨.

†通讯作者. E-mail: zhangjihui@qdu.edu.cn.

\*本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

理论上能够使得算法避免陷入局部最优从而得到全局最优解<sup>[3]</sup>,但是在实际求解过程中,很难确定合适的算法参数,从而导致收敛速度慢或早熟收敛等缺点,因此,国内外学者提出了许多改进方案,主要集中于初始温度的设置、初始解和新解的产生、降温方式等几方面。

针对初始温度的选取和初始解的产生,文献[4]将遗传算法与模拟退火算法相结合,给出了一种确定初始温度的方法.文献[5]在解空间内随机产生 $N$ 个可行解,并记录这 $N$ 个可行解对应目标函数的最大值 $E(X_{\max})$ 和最小值 $E(X_{\min})$ ,根据两者的差值和初始接受劣解的概率确定初始温度.文献[6]提出了用于生成初始可行路径的三角形TSP法,即反复添加一个城市与已有的城市相连接,使得添加该城市后,增加的两条路径减去与二者构成三角形的另一条路径的总距离最小,直至包括所有城市。

针对Metropolis接受准则,文献[7]使用最小生成树构造汉密尔顿回路对TSP的规模进行估计,以该回路的长度作为Boltzmann常数,目的是消除不同规模TSP的能量差值与当前温度值可能存在的数量级差异对Metropolis准则的影响。

针对降温机制,文献[8]提出了非常快速的模拟退火算法,使得求解效果有所提高.文献[9]提出改进的快速模拟退火算法,在一定程度上克服了传统模拟退火算法和文献[8]算法计算效率低的缺点.文献[10]提出了不受降温系数影响,只与迭代次数有关的降温方式,使得温度更新函数与退火时间的幂函数成反比,显著地提高了求解效率.文献[11]提出了自适应模拟退火算法,利用算法搜索过程收集的信息,决定升温或降温操作,该方法基本不受初始温度选择的影响,具有较强的鲁棒性.文献[12]分析模拟退火算法的多种降温方式和内循环次数对求解结果的影响,提出了多次淬火的模拟退火算法.文献[7]提出了自适应升温模拟退火算法,当求解过程陷入局部最优时,通过自适应升温因子使得温度升高,再重复降温过程。

通过实验发现,上述算法中,文献[8-9]对于求解大规模的TSP效果有限,易陷入局部最优;文献[10]的求解速度较快,前期解的质量迅速改善,但是后期易陷入局部最优;文献[11]依据算法搜索过程的反馈信息决定升温或降温操作,但是升温幅度对于求解较大规模的TSP效果有限;文献[7]和文献[12]设置了升温操作,但是升温过程均没有结合算法的搜索反馈

信息,求解较大规模问题的能力有限。

分析和总结以上算法的优缺点,针对传统模拟退火算法在求解旅行商问题时收敛速度慢,运行时间长,在有限时间内无法求得最优解,易陷入局部最优,且随着问题求解规模的增大缺陷愈发明显等问题,本文提出基于波动温度控制机制的自适应模拟退火算法.从5个方面进行了改进:初始温度的设定、邻域搜索机制、Metropolis准则优化、内循环次数、温度控制函数.算法的关键步骤是将相邻两次温度的比值作为内循环搜索次数变化的依据,同时创新地将改进的波动函数作为温控函数引入算法的退火过程,根据目标函数改进的次數和算法当前的迭代次数,自适应调整波动温控函数的系数,使得算法在保持温度幅值递减的总趋势下,可实现多次升温 and 降温,从而减小陷入局部最优的概率,获得满意的求解效果.通过TSPLIB数据库提供的数据进行实验,实验结果表明,改进的算法对于求解中小规模的TSP可在较短时间内求出最优解,对于大规模问题能够在较短时间内求得满意解,且在算法运行后期依然保持逼近最优解的能力。

## 1 传统模拟退火算法

模拟退火算法的思想来源于金属热处理过程中固体退火原理,由Metropolis等<sup>[13]</sup>于1953年提出,1983年,该算法被成功地引入至组合优化领域<sup>[14]</sup>.由于模拟退火算法在处理离散变量优化和非线性优化问题中具有其他算法不具备的优势,得到了众多学者的青睐.算法的基本流程如下。

step 1: 算法初始化: 设定初始参数,包括初始温度 $T_0$ 、内循环次数 $L$ 、外循环次数或终止温度、降温系数或降温幅度。

step 2: 在解空间随机产生一个初始解,记作 $X_{\text{old}}$ ,并对其按既定的规则进行扰动得到扰动后的解,记作 $X_{\text{new}}$ 。

step 3: 计算 $X_{\text{new}}$ 和 $X_{\text{old}}$ 的能量值 $E(X_{\text{new}})$ 和 $E(X_{\text{old}})$ (即目标函数值),根据解的能量值和当前温度 $T$ 选择性接受 $X_{\text{new}}$ .若接受 $X_{\text{new}}$ ,则将 $X_{\text{new}}$ 的值赋给 $X_{\text{old}}$ ,并返回至step 2,对此时的 $X_{\text{old}}$ 进行扰动操作;若不接受 $X_{\text{new}}$ ,则将此时的 $E(X_{\text{old}})$ 直接进行扰动并进行下一次判断.根据Metropolis准则依概率接受 $X_{\text{new}}$ ,接受概率 $p$ 为

$$p = \begin{cases} 1, & E(X_{\text{new}}) - E(X_{\text{old}}) < 0; \\ e^{-\frac{E(X_{\text{new}}) - E(X_{\text{old}})}{T}}, & \text{otherwise.} \end{cases}$$

step 4: 降温:常用的降温方式包括等差降温: $T_{k+1} = T_k - \Delta T$  ( $\Delta T$ 为固定值); 等比降温: $T_{k+1} = \alpha T_k$  ( $0 < \alpha < 1$ )等。

step 5: 判断是否达到外循环总次数或达到终止温度. 若是,则算法停止并输出当前最优解,否则返回至 step 2.

## 2 改进的自适应模拟退火算法

### 2.1 初始温度的设定

传统模拟退火算法通常将初始温度设定为一个足够大的值,以便能够获得更好的搜索效果,但是并不存在一个通用的设置方法,不合适的初始温度不但易造成时间的浪费且不利于解的搜索. 因此,对于初始温度的设定采用文献[4]的方法:根据 Metropolis 准则,设置初始温度为

$$T = -\frac{E(X_{\max}) - E(X_{\min})}{\ln p}$$

其中: $E(X_{\max})$ 和 $E(X_{\min})$ 分别为解空间内随机选取的 $N$ 个可行解所对应目标函数值的最大值和最小值, $p$ 的取值范围为(0.9, 0.99).

### 2.2 邻域搜索机制的改进

在传统模拟算法中,新解是对已接受的解进行扰动操作产生的,故扰动机制对于算法的性能至关重要. 扰动方式不宜过于简单,也不宜过于复杂. 过于简单的扰动方式不利于新解的搜寻,而一些相对复杂的扰动方式可通过几种简单的扰动适当组合得到. 故选取3个经典的扰动操作分别以不同的概率对旧解进行扰动.

1) 两城市交换法 (swap): 随机选择2个城市位置,将2个位置上的元素进行交换.

2) 城市随机插入法 (insertion): 随机选择2个位置,将第1个位置上的元素插入至第2个位置元素的后面.

3) 2-opt法 (reversion): 随机选择2个位置,将这2个位置间的元素进行逆序排列.

### 2.3 Metropolis 准则的改进

1) Metropolis 准则是影响是否接受扰动后得到新解的关键步骤,直接影响算法运行的时间和寻优能力.

传统模拟退火算法 Metropolis 准则接受概率的表达式为 $p = e^{-\frac{E(X_{\text{new}}) - E(X_{\text{old}})}{T}}$ ,对于不同规模的 TSP, $E(X_{\text{new}}) - E(X_{\text{old}})$ 的值与当前的 $T$ 值可能存在数量级差异的情况,从而导致 $p$ 值过大(趋近于1)或过小(趋近于0),从而影响算法的运行时间和寻优能力. 文

献[7]的做法为令

$$p = e^{-\frac{E(X_{\text{new}}) - E(X_{\text{old}})}{aT}}$$

其中 $a$ 为连接所有城市的最小生成树的长度,这种方法求得的 $a$ 值可被证明不超过该 TSP 最优解的2倍. 该方法在一定程度上降低了 $E(X_{\text{new}}) - E(X_{\text{old}})$ 的值与当前 $T$ 值出现数量级差异的可能,但是在求解较大规模 TSP 时,其可行解能量之差可能仍然远大于最优解能量值,导致该方法优势不明显,且随着问题规模的增大,求解最小生成树的时间也随之增加. 鉴于此,令

$$p = e^{-\frac{E(X_{\text{new}}) - E(X_{\text{old}})}{ST}}$$

其中: $S = e^{-\frac{E(X_{\text{average}}) - E(X_{\min})}{T_0}}$ , $E(X_{\text{average}})$ 和 $E(X_{\min})$ 为从解空间内随机选取的 $N$ 个可行解对应目标函数的平均值和最小值, $T_0$ 为初始温度. 通过 $S$ 调节 Metropolis 接受准则,可在一定程度上消除 $E(X_{\text{new}}) - E(X_{\text{old}})$ 的值与当前 $T$ 值可能出现的数量级差异,从而使得 $p$ 值更加灵敏.

2) 在执行 Metropolis 准则时,设置2个计数器,分别为改进计数器 $G$ 和接受计数器 $J$ ,当 $E(X_{\text{new}}) - E(X_{\text{old}}) < 0$ 时,改进计数器 $G$ 的值增加1,接受计数器 $J$ 也增加1;当 $E(X_{\text{new}}) - E(X_{\text{old}}) \geq 0$ 且接受了 $X_{\text{new}}$ 时,只对接受计数器 $J$ 的值增加1.

### 2.4 温度控制方式的改进

求解的效率和质量是模拟退火算法中2个相互矛盾的指标,退火环节则是平衡这二者的关键,合适的温度控制方式能够降低求解陷入局部最优的概率,因此,设计了全新的温控方式,将改进的波动函数 $y = \frac{A(\cos \omega x)^2 + \varphi}{Cx + D}$ 作为温度控制函数引入退火过程,代替传统算法中的等差或等比降温.

如图1所示,实线为波动函数 $y = \cos x$ 的图像,虚线为改进的波动函数 $y = \frac{5(\cos 0.5x)^2 + 1}{0.1x + 1}$ 的图像(为清晰起见,作局部放大处理). 改进的波动函数不但保留了原波动函数的相关性质,还具有整体幅值递减的特征,且函数与 $x$ 轴无交点,即函数无0点, $x$ 轴为函数趋向于无穷方向的渐近线. 通过调整 $A$ 、 $\omega$ 、 $\varphi$ 、 $C$ 和 $D$ 的值,可对函数图像进行变换.

1) 改变 $A$ 的值,即改变振幅的大小,函数波峰与波谷的相对距离将发生变化.

2) 改变 $\omega$ 的值,相邻两波峰(或波谷)的横向距离将发生变化. 在改进算法中,令 $x$ 轴的数值等于外循环代数,即温度的变化次数. 为避免多数相邻两代的温度值未处于相邻两波峰(或波谷)内进而影响后续

搜索,令 $\omega \leq \pi$ ,经实验分析, $\omega$ 的值位于(0.1, 1]间较为合适。

3)  $\varphi$ 的作用为控制波峰、波谷的高度和相对距离,但是 $\varphi$ 对波谷的高度影响更大, $\varphi$ 越大,波谷远离 $x$ 轴的程度越大于波峰。

4) 改变 $C$ 的值,将影响 $\frac{A}{C}$ 和 $\frac{\varphi}{C}$ 的相对大小。 $C$ 值越大,波峰与波谷的相对距离越小,函数整体越趋近于 $x$ 轴,反之,波峰与波谷的相对距离则越大,函数整体越远离 $x$ 轴。

5) 改变 $D$  ( $D > 0$ ) 的值,函数图像整体向左平移。 $D$ 值越大,在 $x$ 轴正半轴,函数波峰与波谷的相对距离则越小,函数整体越趋近于 $x$ 轴。

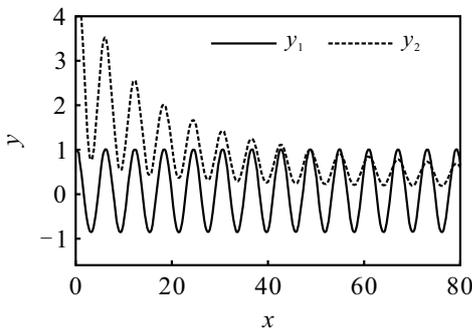


图1 改进的波动函数和原波动函数对比

综上所述,对于改进的波动函数,其基本特性是:当分子的值越大时,分母值则相对越小,整体函数值便越大,使得波峰波谷均有所增大;反之,当分子的值越小时,分母值相对越大,函数图像越接近 $x$ 轴,且波峰波谷均呈降低趋势。改变 $\omega$ 的值,函数图像的横向疏密程度将发生变化。

事实上,将改进的波动函数作为温控函数,令 $x = k$ ,若 $A$ 、 $\omega$ 、 $\varphi$ 、 $C$ 和 $D$ 的值固定,则降温过程将严格按函数曲线变化,且退火过程没有结合当前解的搜索情况,从而导致降温过程的自适应性不足,故对 $A$ 、 $\omega$ 、 $\varphi$ 、 $C$ 和 $D$ 的值进行动态调整,使其结合当前解的搜索情况从而更具智能性。

对于 $A$ ,若其值固定,则函数的振幅依次递减,使得算法的升温幅度无法突破在此之前的升温幅度。鉴于此,对于 $A$ 的设置应使得振幅能够摆脱依次递减的趋势,由于函数的自变量为当前外循环次数,函数的定义域为(0,  $K$ ],结合当前解的搜索情况,令

$$A = \frac{T_0^{\frac{K-k}{K}} L_k}{1 + G_k}.$$

其中: $T_0 = -\frac{E(X_{\max}) - E(X_{\min})}{\ln p}$ ,  $K$ 为外循环总次数, $L_k$ 为第 $k$ 次迭代的内循环次数, $G_k$ 为第 $k$ 次迭代的内循环中解改进的次数。

对于 $\omega$ ,在算法搜索后期,当求得问题的最优解或高度满意解后,剧烈的升温和降温过程将增加算法的运行时间,故相邻两波峰(或波谷)的距离在算法运行后期应适当增大,使得温度曲线趋于平缓,鉴于 $\omega$ 的值宜位于(0.1, 1]间,且 $\frac{1 + G_k}{1 + J_k} \in (0.1, 1]$ ,故令 $\omega = \frac{1 + G_k}{1 + J_k}$ ,其中 $J_k$ 为第 $k$ 次迭代的内循环中解被接受的次数(对 $G_k$ 和 $J_k$ 的值加1是为防止分子或分母等于0)。

对于 $C$ ,由于已对 $A$ 的值进行设置, $C$ 值过大或过小将影响 $A$ 值设置的效果,经实验分析, $C$ 的取值在(0.1, 0.5)间较为合适。

对于 $\omega$ ,由于其值控制波峰、波谷的高度和相对距离, $\omega$ 过小使得波峰与波谷间的距离过大从而会影响搜索时间, $\omega$ 过大则影响 $A$ 值设置的效果,经实验观察, $\omega$ 的取值为 $nT_0$ 较为合适,其中 $n \in [0.5, 2]$ 。

对于 $D$ ,为避免对已设置的参数值造成过多的影响,同时发挥出 $D$ 对波动函数的作用,令 $D = 1$ 。

综上所述,令当前温度为

$$T_k = \frac{T_0^{\frac{K-k}{K}} L_k \cos^2\left(\frac{1 + G_k}{1 + J_k} k\right) + nT_0}{Ck + 1}.$$

## 2.5 内循环次数的改进

传统模拟退火算法中,每个温度值所对应的内循环次数是固定的,但是TSP的解空间可能存在若干邻域对最优解或满意解贡献较大,对这样的邻域应增加搜索次数以增大搜索强度。于是,对内循环改进为 $L_{k+1} = \left\lceil \frac{T_k}{T_{k+1}} \right\rceil L_k$ , $\lceil \cdot \rceil$ 为取整操作。

以上对于温度和内循环设置的目的是令升温和降温的幅度与速度根据当前解的搜索情况自适应调整。在算法搜索前期,解改进的次数较多,则 $\frac{L_k}{1 + G_k}$ 对 $T_k$ 的影响也相对较小,且 $T_0^{\frac{K-k}{K}}$ 随着 $k$ 的增加而递减,在一定程度上抵消了 $\frac{L_k}{1 + G_k}$ 的变化对于 $T_k$ 的影响,使得 $T_k$ 的变化不会过于剧烈。随着算法的进行,解改进的次数总体呈降低趋势,则 $\frac{L_k}{1 + G_k}$ 的值对 $T_k$ 的影响也随之增强,在算法搜索后期,随着 $k$ 的增加, $T_0^{\frac{K-k}{K}}$ 对 $T_k$ 的影响趋于消失,此时温度的升降更多地取决于 $\frac{L_k}{1 + G_k}$ 的变化,即当新解改进的次数逐渐减少时,整体温度将逐渐波动升高,根据Metropolis准则,降低了收敛于局部最优的概率,进而增强全局寻优的能力。

当温控函数曲线处于从波峰到波谷的区间时,根据Metropolis准则,此时接受劣解的能力逐渐减弱,令

$L_{k+1} = \left[ \frac{T_k}{T_{k+1}} \right] L_k$ , 可适当增大内循环搜索次数, 从而增强寻优能力. 当温控函数曲线处于从波谷到波峰的区间时, 根据Metropolis准则, 此时接受能力强, 搜索效率高, 令  $L_{k+1} = \left[ \frac{T_k}{T_{k+1}} \right] L_k$ , 可适当减少内循环搜索次数, 从而节约算法的运行时间.

由于解改进的次数与  $\frac{1+G_k}{1+J_k}$  的值正相关, 当  $\frac{1+G_k}{1+J_k}$  的值相对较大时, 即解改进的次数较多时, 算法升温 and 降温的次数将更加集中, 大部分搜索工作在这一阶段完成. 当  $\frac{1+G_k}{1+J_k}$  的值相对较小时, 即解改进的次数相对减少时, 温度变化将趋于平缓, 解的接受能力将趋于稳定, 由于  $L_{k+1} = \left[ \frac{T_k}{T_{k+1}} \right] L_k$ , 内循环次数的变化也趋于稳定, 此时算法的搜索能力将趋近于传统模拟退火算法, 在此基础上, 根据  $\frac{T_0^{K-k}}{1+G_k} L_k$  进行升温, 从而增强了全局寻优能力, 直至找到最优解或满意解后,  $G_k$  的值不再发生变化,  $T_k$  的值逐渐波动降低, 又因  $L_{k+1} = \left[ \frac{T_k}{T_{k+1}} \right] L_k$ , 内循环次数在算法搜索后期解不再改进后也逐渐减小, 从而缩短了算法的运行时间.

### 3 实验和结果分析

为检验改进算法的有效性, 使用TSPLIB数据库中不同规模的TSP实例进行仿真实验. 实验安排如下.

- 1) 采用不同的实例测试组合扰动方式的效果.
- 2) 采用不同的实例测试改进的Metropolis接受准则的效果.
- 3) 测试温度控制函数中参数取值对求解效果的影响.
- 4) 将所提出算法与文献[7-11]中使用不同降温方式的算法的求解性能进行比较, 采用不同的实例测试波动温控机制的效果.
- 5) 将所提出算法与其他文献中的近似算法的求解数据进行比较, 进一步表明所提出算法的有效性.

实验环境为Matlab 2018 b, Windows 10操作系统, Intel(R) Core(TM) i5-9 500 CPU处理器, 8 GB内存.

#### 3.1 实验 1

2-opt法是一种高效的局部搜索方法, 故适当增加2-opt法的所占比例. 为验证该方法的合理性, 选取4种不同比例的扰动方式进行测试, 以St 70、Tsp 225和Dsj 1 000三个算例为例, 各进行30次实验, 取其平均值进行对比, 实验结果如表1所示.

表 1 不同扰动比例求解TSP所得最优解代数和所用时间

测试算例(温度变化总次数 $K$ )	pSwap : pInsertion : pReversion							
	1 : 1 : 1		1 : 2 : 1		1 : 1 : 2		2 : 1 : 1	
	代数	时间	代数	时间	代数	时间	代数	时间
St 70 (500)	403	108.6	397	92.2	209	56.4	363	73.8
Tsp 225 (500)	534	239.1	522	304	38.0	118.7	487	159.8
Dsj 1 000 (1 000)	1 165	679.4	895	635	17.8	504.1	1 087	473.8

由表1可见, 当pSwap : pInsertion : pReversion = 1 : 1 : 2时, 求解效果较好, 2-opt法占较大比例有利于算法求解的思想得以验证, 故所提出算法按pSwap : pInsertion : pReversion = 1 : 1 : 2对解进行扰动.

图2为采用不同扰动方式求解A 280实例时目标函数的变化情况.

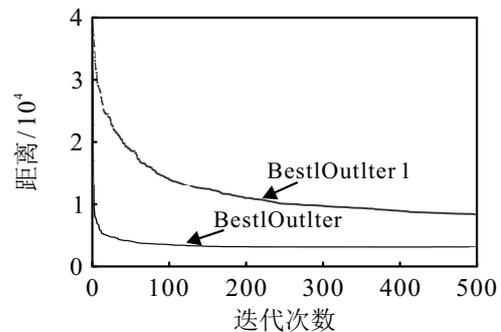


图 2 2种扰动方式对求解的影响

由图2可见, 在求解A 280算例时, 单纯使用2交换法对旧解进行扰动(BestOutlier 1曲线)的求解效果明显弱于用3种扰动方式对旧解进行扰动(BestOutlier曲线)的求解效果(pSwap = 0.25, pInsertion = 0.25, pReversion = 0.5), 进一步表明使用多种方式对解进行扰动有助于提高算法的搜索效果. 对其他算例进行实验, 多方式扰动同样有效.

#### 3.2 实验 2

以Berlin 52、A 280、Dsj 1 000实例测试用S替代Boltzmann常数对Metropolis接受准则的影响, 分别与原Metropolis准则和文献[6]中改进Metropolis准则进行对比. 求解Berlin 52算例目标函数的收敛曲线如图3所示, 不同Metropolis准则的求解结果如表2所示.

图3为用改进Metropolis准则后的算法与文献[6]中所使用改进Metropolis准则的算法以及传统模拟退火算法对Berlin 52问题进行求解的结果对比. BestOutlier 1曲线为原Metropolis准则的求解效果, BestOutlier 2曲线为文献[6]中Metropolis准则的

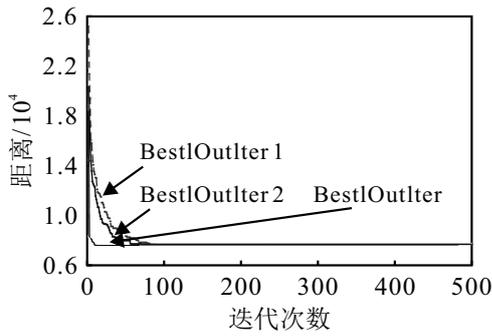


图3 3种不同Metropolis准则的求解效果

表2 不同准则求解TSP出现最优解的代数和时间

测试算例(温度变化总次数K)	SA Metropolis准则		文献[6]准则		本文改进准则	
	代数k	t/s	代数k	t/s	代数k	t/s
St 70 (500)	82	54.2	73	56.3	12	40.6
Tsp 225 (500)	422	158.6	309	162.5	198	113.7
Dsj 1000 (1000)	1279	497.2	1308	536.5	454	397.2

求解效果, BestlOutlter 曲线为使用改进的 Metropolis 准则后的求解效果. 由图3可见, 改进后准则的接受效果更好, 得到满意解的速度快, 在 500 次外循环过程中, 最优解出现在第 12 代, 比文献 [6] 的算法与原传统算法均更早求得最优解, 表 2 数据也反映了改进 Metropolis 准则的优势.

### 3.3 实验 3

以求解 Berlin 52、Tsp 225、A 280 算例为例, 测试温度控制函数中的  $C$ 、 $n$  和  $\omega$  值对算法性能的影响, 实验结果如图 4 所示.

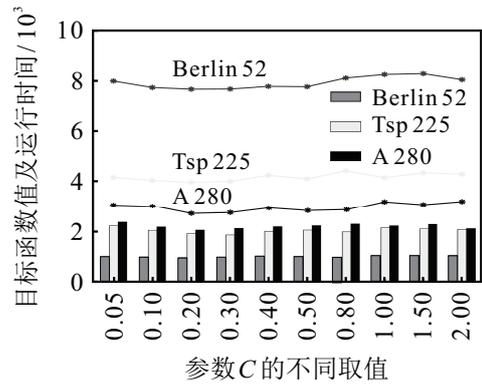
如图 4 所示, 折线图和柱状图分别表示当其余实验参数均相同时, 不同参数取值 ( $C, n, \omega$ ) 所求得的 30 次实验的平均目标函数值和平均运行时间 (为清晰起见, 所有的求解时间值均放大了 15 倍). 由图 4 可见, 当  $C \in (0.1, 0.5), n \in [0.5, 2], \omega \in (0.1, 1]$  时求解效果较好. 对其余算例进行相同的测试, 取值区间也均基本适用, 在此不作一一展示.

图 4(d) 为求解 Tsp 225 算例时,  $\omega = \frac{1 + G_k}{1 + J_k}$  的变化趋势, 由图 4(d) 可见, 随着迭代的进行,  $\omega$  整体呈减小趋势, 进一步为温控函数设计的合理性提供了佐证.

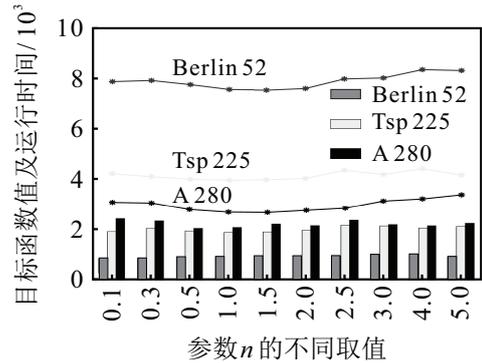
### 3.4 实验 4

测试引入波动温控机制后算法的运行效果, 其中求解各算例使用的参数如表 3 所示.

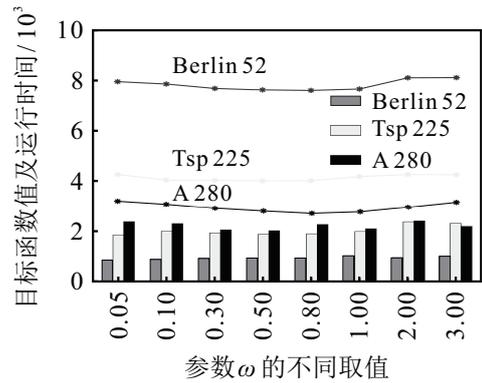
使用引入波动温控机制的自适应模拟退火算法, 按照表 3 参数对 15 个不同规模的 TSP 进行求解.



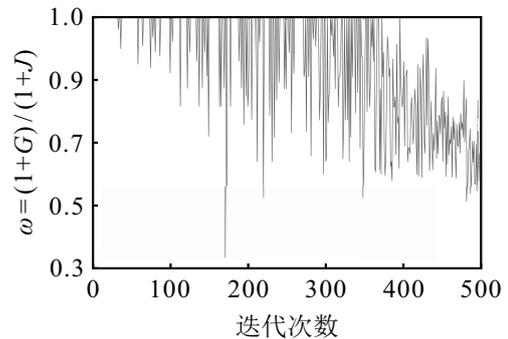
(a) 不同 C 值的实验结果



(b) 不同 n 值的实验结果



(c) 不同 omega 值的实验结果



(d)  $\omega$  的变化趋势

图 4 不同参数值的实验结果和  $\omega$  的变化

主要对比 30 次实验各算法求得的目标函数值信息和运行时间信息以及各算法 1 min 的求解率 (测试算例的最优值/1 min 截止时求得的目标函数值). 由实验数据可以看出, 引入波动温控的模拟退火算法对 15 个算例的求解结果均等于或最接近于最优值. 求

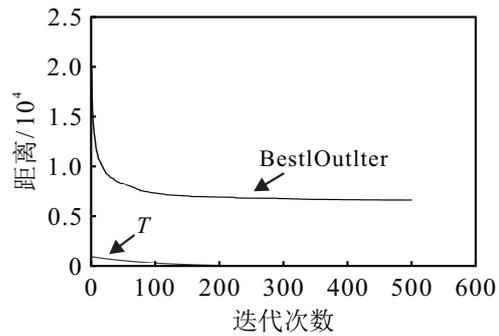
表3 实验相关参数取值

测试算例	$N$ 值	初始接受概率 $p$	pSwap	pReversion	pInsertion	$C$ 值	$n$ 值	初始内循环次数 $L_0$	温度变化次数 $K$
Burma 14	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Eil 51	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Berlin 52	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Eil 76	100	0.9	0.25	0.5	0.25	0.2	1	200	500
St 70	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Eil 101	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Bier 127	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Tsp 225	100	0.9	0.25	0.5	0.25	0.2	1	200	500
A 280	100	0.9	0.25	0.5	0.25	0.2	1	200	500
Att 532	100	0.9	0.25	0.5	0.25	0.3	1	200	500
Rat 783	100	0.95	0.25	0.5	0.25	0.3	2	500	1000
Dsj 1000	100	0.95	0.25	0.5	0.25	0.3	2	500	1000
R1 1304	100	0.95	0.25	0.5	0.25	0.3	2	500	1000
F1 1577	100	0.95	0.25	0.5	0.25	0.3	2	500	1000
D2 103	100	0.95	0.25	0.5	0.25	0.3	2	500	1000

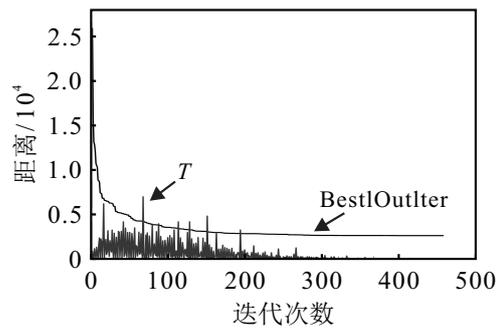
解所用的平均时间也较短,且改进算法的求解更加稳定.波动程度更小,1 min 求解率也均优于其他算法.为检验波动温控机制和自适应调整内循环搜索次数的求解效果,将求解收敛趋势曲线和温度变化曲线放在同一图内进行对比,如图5(a)和图5(b)所示.

图5(a)和图5(b)为传统模拟退火算法和引入波动温控机制的模拟退火算法求解A 280算例目标函数的收敛趋势以及温度变化曲线,其中 BestlOutlter 曲线为目标函数的变化曲线,  $T$  曲线为温度变化曲线.两种算法均从相同的初始温度1 000开始(传统模拟退火算法降温系数取  $\alpha = 0.99$ ),并施加相同的扰动寻优方式和接受准则.由5(a)和图5(b)可见,传统模拟退火算法求解A 280算例时,不但前期求解速度慢,且算法后期陷入了局部最优.用波动温控的算法求解时(此时  $D = 1$ ),在算法搜索前期,随着温度的逐渐波动升高,改进算法的求解质量迅速改善,在算法中后期,当求解进入局部最优状态或接近收敛时,经过几次较大的升温过程可增强算法的搜索能力,使得求解过程增大了跳出局部最优的概率,最终求得最优解.

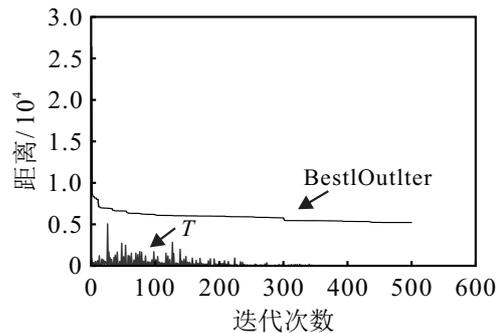
由图5(b)可见,在150代前,经过几次较大的升温过程,目标函数值几次跳出局部最优,迅速收敛至最优解附近.为进一步测试算法在不同时期的升温幅度对求解质量的影响,在图5(b)实验的基础上,通过调整  $D$  值,使得温控函数曲线向左平移,从而避开前期高温时段,对比实验结果如图5(c)所示.避开了高温时段的降温过程的求解效率与求解能力均弱于图



(a) SA 求解A 280算例



(b) 波动温控SA 求解A 280算例( $D=1$ )

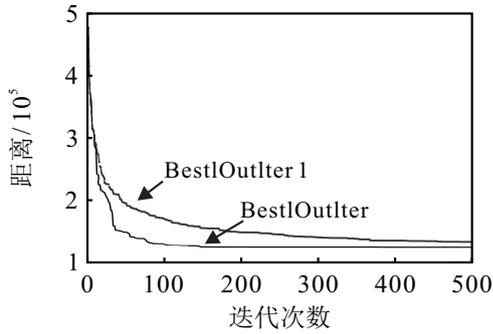


(c)  $D=150$ 时求解A 280算例

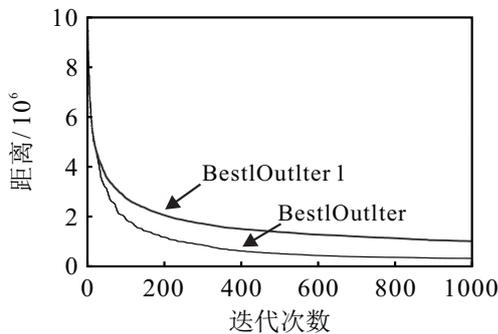
图5 SA和波动温控算法求解A 280算例

5(b)实验的降温过程,从而进一步表明前期较大的几次升温过程有利于增强算法的搜索能力.图6(a)~图6(c)为传统模拟退火算法和波动降温的模拟退火算法分别求解Bier 127、R1 1 304和D2 103算例的目标函数曲线,两种算法均从相同的初始温度开始求解,并施加相同的扰动寻优方式和接受准则.

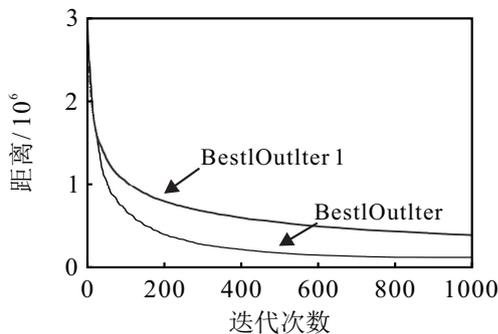
图6中:BestlOutlter 1 曲线为传统模拟退火算法求解效果,BestlOutlter 曲线为改进算法的求解效果.对于Bier 127算例,同样进行500次的温度变化,对于R1 1 304和D2 103算例,同样进行1 000次的温度变化,由图6可见,改进算法的求解质量明显优于传统模拟退火算法,且算法的搜索能力更强.



(a) Bier 127算例



(b) R1 1 304算例



(c) D2 103算例

图6 SA与波动温控算法求解对比

3.5 实验5

对比波动温控模拟退火算法与其他文献中不同算法求解TSP的实验数据.结果如表4所示(“-”为文献中未提及的数据,所提出算法的求解数据均加粗处理).

表4 不同算法求解数据对比

测试算例(最优解)	测试算法	算法所求最优解	算法运行时间/s
Burma 14 (30.878 5)	MFACA <sup>[15]</sup>	30.878 5	—
	ACO <sup>[16]</sup>	30.878 5	36.2
	DGSO <sup>[17]</sup>	30.878 5	—
	ATRSA <sup>[7]</sup>	30.878 5	38.2
	SA	30.878 5	34.1
	<b>本文算法</b>	<b>30.878 5</b>	<b>33.6</b>
Eil 51 (426)	IGSA <sup>[18]</sup>	460	102
	ACO <sup>[16]</sup>	438.74	—
	MMAS <sup>[6]</sup>	436.63	—
	AADC <sup>[19]</sup>	428.74	—
	ACADCG <sup>[20]</sup>	428.19	—
	DWPA <sup>[21]</sup>	428.87	—
	ATRSA <sup>[7]</sup>	426	62.4
	ICA <sup>[22]</sup>	428.87	—
	DBA <sup>[17]</sup>	436.48	520
	IACO <sup>[16]</sup>	431.05	126
	FWA <sup>[20]</sup>	453.26	703
	CFWA <sup>[20]</sup>	428.87	735
Berlin 52 (7 542)	RBIFWA <sup>[20]</sup>	426.13	307
	ISA <sup>[7]</sup>	428.87	—
	SA	436.71	68.3
	<b>本文算法</b>	<b>428.19</b>	<b>63.9</b>
	DBA <sup>[17]</sup>	7 878.71	476
Eil 76 (538)	IACO <sup>[16]</sup>	7 544.365 9	708
	FWA <sup>[20]</sup>	7 932.54	328
	CFWA <sup>[20]</sup>	7 544.365 9	634
	RBIFWA <sup>[20]</sup>	7 544.365 9	307
	SA	7 544.365 9	62.7
	<b>本文算法</b>	<b>7 544.365 9</b>	<b>64.7</b>
St 70 (675)	IGSA <sup>[18]</sup>	548	172
	ACO <sup>[16]</sup>	555.61	—
	MMAS <sup>[6]</sup>	552.26	—
	RIFA <sup>[23]</sup>	544.43	—
	ACADCG <sup>[20]</sup>	543.41	—
St 70 (675)	ATRSA <sup>[7]</sup>	538	87.4
	ICA <sup>[22]</sup>	544.37	—
	SA	553.32	77.4
	<b>本文算法</b>	<b>544.37</b>	<b>78.8</b>
	ACO <sup>[16]</sup>	687.46	—
	MMAS <sup>[6]</sup>	685.13	—
	AADC <sup>[19]</sup>	682.31	—
	ACADCG <sup>[20]</sup>	681.25	—
	ATRSA <sup>[7]</sup>	677.11	—
	DBA <sup>[17]</sup>	703.96	615
	IACO <sup>[16]</sup>	677.11	421
	FWA <sup>[20]</sup>	724.75	132
CFWA <sup>[20]</sup>	677.11	726	
RBIFWA <sup>[20]</sup>	675	126	
ISA <sup>[7]</sup>	677.11	88.5	
SA	682.57	75.9	
<b>本文算法</b>	<b>677.11</b>	<b>74.5</b>	

表4 (续)

测试算例(最优解)	测试算法	算法所求最优解	算法运行时间/s
Eil 101 (626)	ACO <sup>[16]</sup>	675	92.3
	SA	671	84.7
	<b>本文算法</b>	<b>630</b>	<b>87.6</b>
Bier 127 (118 282)	ACO <sup>[16]</sup>	126 321	—
	SA	121 145	92.3
	<b>本文算法</b>	<b>118 475</b>	<b>106.9</b>
Tsp 225 (3 916)	ACO <sup>[16]</sup>	4 032	—
	IGSA <sup>[18]</sup>	4 068	146.9
	ATRSA <sup>[7]</sup>	3 972	—
	SA	4 120	111.5
	<b>本文算法</b>	<b>3 925</b>	<b>132.3</b>
A 280 (2 579)	SA	3 030	166.7
	<b>本文算法</b>	<b>2 663</b>	<b>147.1</b>
Att 532 (87 686)	SA	96 157	197.7
	<b>本文算法</b>	<b>89 212.319 2</b>	<b>190.5</b>
Rat 783 (8 806)	SA	12 729	367.2
	<b>本文算法</b>	<b>9 295</b>	<b>396.8</b>
Dsj 1 000 (18 660 188)	SA	30 268 416	494.7
	<b>本文算法</b>	<b>19 963 508</b>	<b>512.7</b>
R1 1 304 (252 948)	SA	924 750	865.1
	<b>本文算法</b>	<b>282 653</b>	<b>870.4</b>
F1 1 577 (22 249)	SA	53 160	1 123.9
	<b>本文算法</b>	<b>26 854</b>	<b>1 179.6</b>
D2 103 (80 450)	SA	426 655	1 289.3
	<b>本文算法</b>	<b>88 722</b>	<b>1 425.7</b>

分析对比数据可得出,由于引入了升温机制,使得改进的自适应波动温控模拟退火算法因温度变化导致每次内循环次数不同,在求解问题时,相比于传统的模拟退火算法时间略有延长,但可获得更好的搜索效果,尤其在求解大规模的TSP时,这种现象更为明显,相比于其他温度控制方式,所提出算法优势也十分明显,求解绝大多数问题均能够找到最优解或高度满意解,且改进的模拟退火算法对于求解大规模问题所用时间较短,尤其是在算法搜索前期,求解速度快,质量高,求得的最终解与TSPLIB数据库中的最优参考值相比误差亦较小。

#### 4 结论

改进的自适应模拟退火算法在求解TSP时均能够取得较好的效果,尤其是对于较大规模的问题,在求解时,能够迅速收敛至最优值附近,这是使用其他温度控制方式的模拟退火算法所不具备的性能。传

统模拟退火算法在求解大规模的TSP时,易长时间陷入局部最优,求解所需的时间也较长。此外,改进的模拟退火算法与其他文献中的算法相比也有较大优势。

本文的不足之处在于所需调整的参数较多,经多次实验将主要的参数取值确定在一定范围内,这也在一定程度上增加了算法运行的不确定性,存在进一步改进的空间。

#### 参考文献(References)

- [1] 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.  
(Gao H C, Feng B Q, Zhu L. Reviews of the meta-heuristic algorithms for TSP[J]. Control and Decision, 2006, 21(3): 241-247.)
- [2] Baraglia R, Hidalgo J I, Perego R. A hybrid heuristic for the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 2001, 5(6): 613-622.
- [3] Bohachevsky I O, Johnson M E, Stein M L. Generalized simulated annealing for function optimization[J]. Technometrics, 1986, 28(3): 209-217.
- [4] Lin F T, Kao C Y, Hsu C C. Applying the genetic approach to simulated annealing in solving some NP-hard problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1993, 23(6): 1752-1767.
- [5] 王子才, 张彤, 王宏伟. 基于混沌变量的模拟退火优化方法[J]. 控制与决策, 1999, 14(4): 381-384.  
(Wang Z C, Zhang T, Wang H W. Simulated annealing algorithm based on chaotic variable[J]. Control and Decision, 1999, 14(4): 381-384.)
- [6] 庞永明, 钟才明, 程凯. 基于聚类集成的蚁群优化与受限解空间的TSP算法[J]. 中国科学技术大学学报, 2016, 46(9): 780-787.  
(Pang Y M, Zhong C M, Cheng K. A TSP algorithm based on clustering ensemble ACO and restricted solution space[J]. Journal of University of Science and Technology of China, 2016, 46(9): 780-787.)
- [7] 陈科胜, 鲜思东, 郭鹏. 求解旅行商问题的自适应升温模拟退火算法[J]. 控制理论与应用, 2021, 38(2): 245-254.  
(Chen K S, Xian S D, Guo P. Adaptive temperature rising simulated annealing algorithm for traveling salesman problem[J]. Control Theory & Applications, 2021, 38(2): 245-254.)
- [8] Ingber L. Very fast simulated re-annealing[J]. Mathematical and Computer Modelling, 1989, 12(8): 967-973.
- [9] 陈华根, 李丽华, 许惠平, 等. 改进的非常快速模拟退火算法[J]. 同济大学学报: 自然科学版, 2006, 34(8): 1121-1125.

- (Chen H G, Li L H, Xu H P, et al. Modified very fast simulated annealing algorithm[J]. Journal of Tongji University: Natural Science, 2006, 34(8): 1121-1125.)
- [10] 杨若黎, 顾基发. 一种高效的模拟退火全局优化算法[J]. 系统工程理论与实践, 1997, 17(5): 29-35.  
(Yang R L, Gu J F. An efficient simulated annealing algorithm for global optimization[J]. Systems Engineering—Theory & Practice, 1997, 17(5): 29-35.)
- [11] 庞哈利, 郑秉霖, 徐心和. 一种自适应的模拟退火算法[J]. 控制与决策, 1999, 14(5): 477-480.  
(Pang H L, Zheng B L, Xu X H. An adaptive simulated annealing algorithm[J]. Control and Decision, 1999, 14(5): 477-480.)
- [12] 范晔, 周泓. 作业排序模拟退火算法影响因素分析和一种多次淬火模拟退火法[J]. 系统工程理论方法应用, 2003, 12(1): 72-76.  
(Fan Y, Zhou H. On effective factors of simulated annealing algorithm on job-shop scheduling and an algorithm with repetitious quenching[J]. Systems Engineering—Theory Methodology Application, 2003, 12(1): 72-76.)
- [13] Metropolis N, Rosenbluth A W, Rosenbluth M N, et al. Equation of state calculations by fast computing machines[J]. The Journal of Chemical Physics, 1953, 21(6): 1087-1092.
- [14] Kirkpatrick S, Gelatt C D Jr, Vecchi M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.
- [15] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240-1245.  
(Wu Q H, Zhang J H, Xu X H. An ant colony algorithm with mutation features[J]. Journal of Computer Research and Development, 1999, 36(10): 1240-1245.)
- [16] 屈稳太, 丁伟. 一种改进的蚁群算法及其在TSP中的应用[J]. 系统工程理论与实践, 2006, 26(5): 93-98.  
(Qu W T, Ding W. An improved ant colony algorithm and application in the TSP[J]. Systems Engineering—Theory & Practice, 2006, 26(5): 93-98.)
- [17] 张长胜, 孙吉贵, 欧阳丹彤. 一种自适应离散粒子群算法及其应用研究[J]. 电子学报, 2009, 37(2): 299-304.  
(Zhang C S, Sun J G, Ouyang D T. A self-adaptive discrete particle swarm optimization algorithm[J]. Acta Electronica Sinica, 2009, 37(2): 299-304.)
- [18] 张雁翔, 祁育仙. 改进遗传模拟退火算法求解TSP[J]. 智能计算机与应用, 2017, 7(3): 52-54.  
(Zhang Y X, Qi Y X. An improved genetic simulated annealing algorithm to solve TSP[J]. Intelligent Computer and Applications, 2017, 7(3): 52-54.)
- [19] 孟祥萍, 片兆宇, 沈中玉, 等. 基于方向协调的蚁群算法[J]. 控制与决策, 2013, 28(5): 782-786.  
(Meng X P, Pian Z Y, Shen Z Y, et al. Ant algorithm based on direction-coordinating[J]. Control and Decision, 2013, 28(5): 782-786.)
- [20] 孙力娟, 王良俊, 王汝传. 改进的蚁群算法及其在TSP中的应用研究[J]. 通信学报, 2004, 25(10): 111-116.  
(Sun L J, Wang L J, Wang R C. Research of using an improved ant colony algorithm to solve TSP[J]. Journal of China Institute of Communications, 2004, 25(10): 111-116.)
- [21] 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.  
(Wu H S, Zhang F M, Li H, et al. Discrete wolf pack algorithm for traveling salesman problem[J]. Control and Decision, 2015, 30(10): 1861-1867.)
- [22] 张鑫龙, 陈秀万, 肖汉, 等. 一种求解旅行商问题的新型帝国竞争算法[J]. 控制与决策, 2016, 31(4): 586-592.  
(Zhang X L, Chen X W, Xiao H, et al. A new imperialist competitive algorithm for solving TSP problem[J]. Control and Decision, 2016, 31(4): 586-592.)
- [23] 吴俊斌, 吴晟, 吴兴蛟. 一种用于求解TSP问题的随机最佳插入烟花算法[J]. 计算机工程与科学, 2020, 42(11): 2080-2087.  
(Wu J B, Wu S, Wu X J. A randomized best insertion fireworks algorithm for solving TSP problem[J]. Computer Engineering & Science, 2020, 42(11): 2080-2087.)

## 作者简介

陈晟宗(1994—), 男, 硕士生, 从事智能优化算法的研究, E-mail: 839749584@qq.com;

张纪会(1969—), 男, 教授, 博士生导师, 从事复杂系统建模、智能优化算法、物流系统工程等研究, E-mail: zhangjihui@qdu.edu.cn;

于守水(1980—), 男, 高级工程师, 从事智慧物流、企业安全与信息化等研究, E-mail: yuss@qdport.com;

郝为建(1982—), 男, 工程师, 从事智慧物流、智能系统等研究, E-mail: haowj.ajb@qdport.com.

(责任编辑: 魏冰)