

顺序表示编码的和声退火混合算法求解 TSP

王玉亭¹, 孙 剑¹, 李俊青¹, 任立群², 潘全科¹

(1 聊城大学 计算机学院, 山东 聊城 252059; 2 聊城市人民医院 信息科, 山东 聊城 252000)

摘 要: 和声搜索是一种新的启发式优化算法, 针对和声搜索算法容易陷入局部最优的缺陷, 将其与模拟退火算法结合, 得到了一种混合算法, 为了保证和声的产生过程中不出现非法解, 算法采用了基于顺序表示的编码方法. 实验表明, 混合算法针对 TSP 问题具有较高的求解质量.

关键词: 旅行商问题; 和声搜索; 模拟退火; 顺序表示编码

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2010)10-0041-04

Ordinal Representation Coding Hybrid Heuristics Based on Harmony Search and Simulated Annealing Algorithm for TSP

WANG Yu-ting¹, SUN Jian¹, LI Jun-qing¹, REN Li-qun², PAN Quan-ke¹

(1 School of Computer Science, Liaocheng University, Liaocheng 252059, China;

2 Department of Information, Liaocheng People's Hospital, Liaocheng 252000, China)

Abstract: Harmony search(HS) is a new heuristic optimization method. But pure HS may produce premature and poor results. Based on the complementary strengths of HS and simulated annealing (SA) algorithm, a hybrid procedures are put forward by combining the HS and SA. In order to ensure there is no illegal solution in the process of generation of harmony, the newly designed hybrid algorithm use ordinal representation coding one harmony. The experiments show that within the framework of the newly designed hybrid algorithm, the NP-hard classic Traveling salesman problem can be efficiently solved with higher quality.

Key words: TSP; harmony search; simulated annealing algorithm; ordinal representation coding

1 引言

音乐创作中, 乐师们凭自己的记忆, 通过反复地调整乐队中各乐器的音调, 最终能达到一个美妙的和声状态. Z. W. Geem 等人^[1]受这一现象的启发, 将乐器的音调类比于优化问题中的决策变量 x_i^j ($i = 1, 2, \dots, n$), 将各乐器声调的和声类比于解向量 $\mathbf{X}^j = (x_1^j, x_2^j, \dots, x_n^j)$, 评价类比于目标函数 $f(\mathbf{X}^j)$, 和声记忆库类比于种群 $\{X^1, X^2, \dots, X^{HMS}\}$, 提出了一种新的启发式搜索算法——HS 算法. HS 算法从随机产生的和声记忆库出发, 通过对和声记忆库的思考、随机选择音调以及音调微调来产生候选解, 然

后通过对比候选解和记忆库中的最差解来更新和声记忆库. HS 算法原理简单、参数较少并且容易实现, 被成功地应用在多种优化问题中.

TSP 问题已经被证明是一个 NP-hard 问题^[2], 即 $P \neq NP$ 在的假设下, 找不到一个多项式时间算法来求得最优解. 对于 TSP 问题的研究有两个方向: 完全算法和近似算法. 完全算法能保证得到最优解, 但是运行时间是呈指数复杂度的, 因而难以适应大规模的实例; 近似算法则只要求找到近似解, 而在多项式时间内结束, 在 TSP 问题上的近似算法已经有许多算法, 取得一些效果^[3].

近年来发展起来的模拟退火 (Simulated An-

nealing, SA)算法和和声搜索(Harmony search, HS)算法等为解决这类复杂问题提供了新的思路 and 手段.然而,这些基本的局域搜索方法源于对某些自然过程的直接模拟,在求解问题时不可避免地存在着一定的缺陷.本文针对 TSP 问题的具体特征,提出了基于顺序表编码的和声退火混合算法,通过对 TSPLIB 中的几个实例进行仿真实验,表明算法具有良好的求解性能.

2 HS 算法步骤

HS 算法的步骤如下^[4]:

Step 1: 初始化算法参数

HS 算法的参数有和声记忆库的大小(HMS)、和声记忆库思考概率(HMCR)、音调微调概率(PAR)以及创作的次数(NI),各项参数在第一步均要被初始化.

Step 2: 初始化和声记忆库

随机生成 HMS 个和声 $X^1, X^2, \dots, X^{HMS}$ 放入和声记忆库,这里和声记忆库可以类比于遗传算法中的基因池^[4].和声记忆库如下:

$$HM = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^{HMS} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_n^{HMS} \end{bmatrix} \quad (1)$$

Step 3: 创作一个新的和声

创作新的和声 $X' = (x'_1, x'_2, \dots, x'_n)$, 新和声的每一个音调 $x'_i (i = 1, 2, \dots, n)$ 通过以下三种机理产生:①考虑和声记忆库;②音调微调;③随机选择音调.具体操作如下:

首先,通过对和声记忆库思考和随机选取产生新和声的每一个音调 x'_i :

$$x'_i = \begin{cases} x_i^{U(1, HMS)}, & \text{if } Rnd < HMCR \\ x'_i \in \Omega_i, & \text{otherwise} \end{cases} \quad (2)$$

$i = 1, 2, \dots, n$

式中, $U(1, HMS)$ 表示 $[1, HMS]$ 上均匀分布的随机正整数, $x_i^{U(1, HMS)}$ 表示从和声记忆库的第 i 列分量随机选取一个分量; $x'_i \in \Omega_i$ 表示在第 i 分量的取值区间中随机取值; HMCR 为和声记忆库思考概率; Rnd 表示 $[0, 1]$ 上的均匀分布的随机数.

其次,若新和声的音调 x'_i 来自于和声记忆库,要对其进行音调微调,具体操作如下:

$$x'_i = \begin{cases} x'_i \pm Rnd1 * bw & \text{if } Rnd < PAR \\ x' & \text{otherwise} \end{cases} \quad (3)$$

式中, bw 为音调微调带宽; PAR 为和音调微调概率; Rnd 和 Rnd1 表示 $[0, 1]$ 上的均匀分布的随机数.

Step 4: 更新和声记忆库

通过对比新和声和记忆库中的最差解来更新和声记忆库,操作如下:

$$\text{if } f(X'_i) < f(X^{\text{worst}}) \text{ then } X^{\text{worst}} = X' \quad (4)$$

式中, X^{worst} 为和声记忆库中的最差解.

Step 5: 检查是否达到算法终止条件

重复 Step 3 和 Step 4, 直到创作(迭代)的次数达到 NI.

3 HSSA 混合算法

3.1 基本 HSSA 混合算法(BHSSA)

HS 收敛速度快,但收敛精度较低,SA 通用性强,易于实现,但计算时间长,效率低,并且 SA 采用的是串行搜索,容易陷入局部最优.本文结合两者的特点,将 HS 与 SA 相结合,充分发挥二者的优点,得到了一种基本 HSSA 混合算法(记为 BHSSA).BHSSA 混合算法流程图如图 1 所示,其执行过程是:先随机初始化和声记忆库,再通过对比和声记忆库的思考、随机选择音调以及音调微调来产生一个新和声,然后对新和声执行 SA 的抽样过程,最后通过对比新和声和记忆库中的最差解来更新和声记忆库.重复上述过程,开始新一轮的搜索.随着搜索过程的进行,算法逐渐定位在最优解附近区域并进行精细搜索.HS 算法使新和声不断向和声记忆库学习,SA 保证了和声的多样性,这些不同作用的优化操作,丰富了优化过程中的邻域结构,增强了全空间的搜索能力.

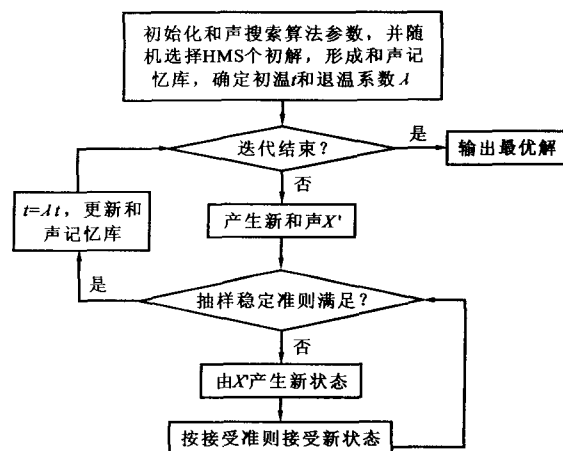


图 1 BHSSA 算法流程图

3.2 改进 HSSA 混合算法(IHSSA)

由基本 HSSA 混合算法流程可见,基本 HSSA

混合算法在每次迭代步中仅仅产生一个新和声,没有尽可能利用和声记忆库内的积累信息,为此本文提出了一种改进的 HSSA 混合算法(记为 IHSSA)。

IHSSA 混合算法在每个迭代步中由和声记忆库内的和声产生多个(N 个)新解,然后从和声库中旧解以及 N 个新解中重新选择个优异解进入和声记忆库内 HM,改进算法能够最大限度的利用和声记忆库里面的信息,提高搜索效率。IHSSA 混合算法流程图如图 2 所示。

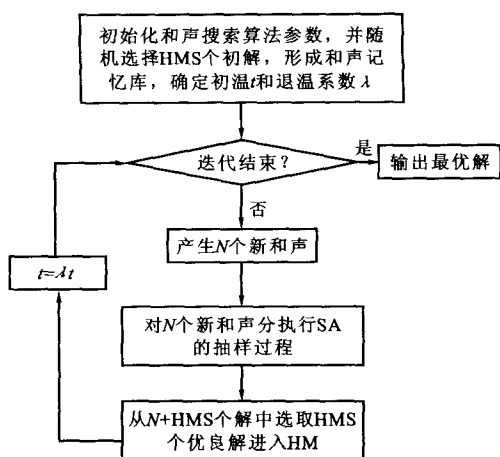


图 2 IHSSA 算法流程图

4 基于顺序表示编码的 HSSA 算法

如果采用基于路径表示(path representation)编码方法,即直接用城市在路线中的位置来表示一条路线,直接运用 HSSA 的方式来产生新和声,该和声可能不是可行解,所以本文采用基于顺序表示(ordinal representation)的编码方法。

基于顺序表示的编码方法是指将所有城市依次排列构成一个顺序表,对于一条旅程,按照旅行依次经过的顺序处理每个城市,每个城市在顺序表中的顺序就是一位编码的表示,每处理完一个城市,从顺序表中去掉该城市。

例如:对于四城市的 TSP 问题,顺序表为(c_1, c_2, c_3, c_4),一条旅程为(c_3, c_1, c_4, c_2),按照基于顺序表示的编码方法,该旅程的编码为(3,1,2,1)。具体编码过程如下。

Step 1:旅程(c_3, c_1, c_4, c_2)中的第一个城市 c_3 在顺序表的位置为 3,所以编码的第一个基因为 3,从顺序表去掉城市 c_3 ,顺序表变为(c_1, c_2, c_4);

Step 2:旅程(c_3, c_1, c_4, c_2)中的第二个城市 c_1 在目前顺序表的位置为 1,所以编码的第二个基因为 1,从顺序表去掉城市 c_1 ,顺序表变为(c_2, c_4);

Step 3:旅程(c_3, c_1, c_4, c_2)中的第三个城市 c_4 在目前顺序表的位置为 2,所以编码的第三个基因为 2,从顺序表去掉城市 c_4 ,顺序表变为(c_2);

Step 4:旅程(c_3, c_1, c_4, c_2)中的第四个城市 c_2 在目前顺序表的位置为 1,所以编码的第四个基因为 1,从顺序表去掉城市 c_2 ,顺序表变为空;

Step 5:编码结束,最终编码为(3,1,2,1)。

基于顺序表示的编码方法的解码过程和编码过程相反。

对于基于顺序表示的编码方法,容易证明下面定理。

定理 1 对于城市规模为 n 的 TSP 问题,基于顺序表示的编码方法表示一条旅程,编码中的第 i 个编码位 x_i 的取值范围为: $\{1, 2, \dots, n - i + 1\}$ 。

据定理 1,针对顺序表示的编码方法表示旅程, HSSA 算法中新和声的每一个音调 x'_i 的产生方法如下:

$$x'_i = \begin{cases} x_i^{U(1, HMS)} & \text{if } \text{Rnd} < \text{HMCR} \\ x'_i \in \{1, 2, \dots, n - i + 1\} & \text{otherwise} \end{cases} \quad (5)$$

式中, $U(1, HMS)$ 表示 $[1, HMS]$ 上均匀分布的随机正整数, $x_i^{U(1, HMS)}$ 表示从和声记忆库的第列分量随机选取一个分量; $x'_i \in \{1, 2, \dots, n - i\}$ 表示在第 i 分量的取值区间中随机取值; HMCR 为和声记忆库思考概率; Rnd 表示 $[0, 1]$ 上的均匀分布的随机数。

据定理 1,音调微调如下:

$$x'_i = \begin{cases} 1 + (x'_i \pm 1) \% (n - i + 1) & \text{if } \text{Rnd} < \text{PAR} \\ x'_i & \text{otherwise} \end{cases} \quad (6)$$

$i = 1, 2, \dots, n$

式中, PAR 为音调微调概率; Rnd 表示 $[0, 1]$ 上的均匀分布的随机数。

5 实验结果与分析

实验选用 gr24 问题和 oliver30 问题,比较了 SA^[5]、GA^[6]、PSO^[7]、IHSSA 的性能,所有算法均以 gcc 4.3.3 for MinGW 为编译环境,运行在 CPU 为 AMD Athlon 64 X2 2GHz、RAM 为 1GB、OS 为 Window XP 的 PC 机上。模拟退火算法的参数: $t = 1\ 000$, $\lambda = 0.95$, 抽样长度为 $n(n-1)/2$; 遗传算法参数为:交叉率为 0.7,变异率为 0.1,群体规模为 100; 粒子群算法参数为: $w = c_1 = c_2 = 0.9$, 群体

规模为 100; IHSSA 的参数为: HMS = 20、HMCR = 0.9、PAR = 0.2、 $t = 100$ 、 $\lambda = 0.95$ 、抽样长度为 $n(n-1)/2$, 每次迭代产生 15 个新和声. 每个问题随机计算 30 次, 各算法的进化曲线如图 3 和图 4 所示, 计算结果如表 1 和表 2 所示. 图中横坐标为迭代次数, 纵坐标为最短路径(取 30 次计算的平均值), 表中“偏差”指平均值相对问题最优值的偏差. 比较可知, IHSSA 算法相对于 SA、GA、PSO 算法有比较快的收敛速度和比较好的计算效果.

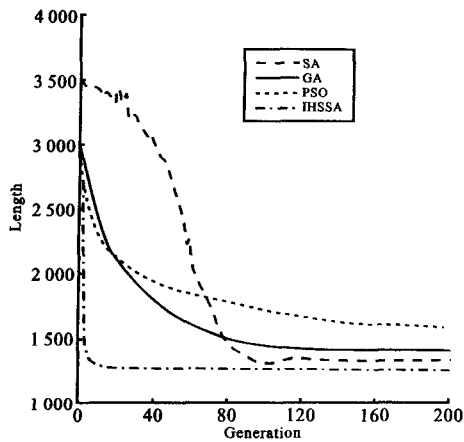


图 3 各算法优化 gr24 的收敛曲线

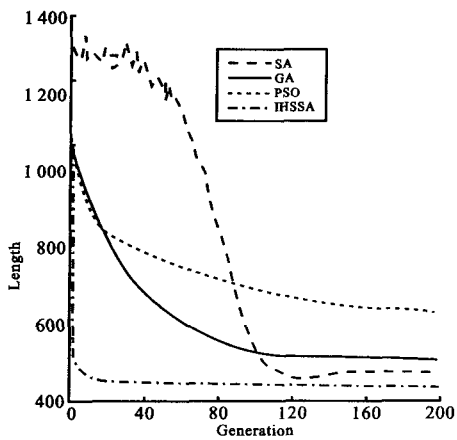


图 4 各算法优化 oliver30 问题的收敛曲线

表 1 各算法优化 gr24 问题的计算结果

gr24	SA	GA	PSO	IHSSA
平均值	1 354.2	1 420.0	1 360.9	1 272.6
偏差/%	6.46	11.64	6.99	0.05
最小值	1 272	1 290	1 289	1 272
标准差	62.86	92.73	42.80	2.03

将 IHSSA 算法针对 Oliver30 问题随机运行 20 次的结果与文献[8]比较, 见表 3. 比较可知, IHSSA 算法较文献[8]中的算法有比较好的计算结果.

表 2 各算法优化 oliver30 问题的计算结果

oliver30	SA	GA	PSO	IHSSA
平均值	469.6	505.6	483.4	422.2
偏差/%	0.118 0	0.203 8	0.150 9	0.53
最小值	423.949 0	423.740 6	424.691 6	423.740 6
标准差	45.45	37.84	23.43	2.36

表 3 IHSSA 算法与文献[8]中结果比较

oliver30	文献[8]	IHSSA
平均值	426.685 6	425.733 2
最优解	423.740 5	423.740 5
最差解	430.711 2	430.608 4

6 结束语

最近发展起来的智能优化算法——和声搜索算法与遗传算法相比, 收敛速度快, 但收敛精度较低, 针对这一弱点进行了改进, 提出了和声搜索-模拟退火混合算法来解决 TSP 问题. 实验结果表明和声搜索-模拟退火混合算法具有较高的计算性能. 下一步的主要研究工作是: (1) 研究算法参数对混合算法性能的影响, 寻找各参数与求解问题之间的联系, 设计出适合于求解问题的参数; (2) 设计更高效的编码方式以及和声产生方式, 进一步提高算法的时间性能.

参考文献:

[1] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search[J]. Simulation, 2001, 76(2):60-68.

[2] 杨理云. 用模拟退火算法求解旅行商问题[J]. 微电子学与计算机, 2007, 24(5):199-202.

[3] Guo T, Michalewicz Z. Inver-over operator for the tsp [C]//Proceedings of the 5th International Conference on Parallel Problem Solving from Nature Conf. Berlin: Springer-Verlag, 1998:803-812.

[4] Geem Z W, Kim J H, Loganathan G V. Harmonysearch optimization: application to pipe network design[J]. International Journal of Model Simulation, 2002, 22(2):25-133.

[5] 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社, 2001:32-33.

[6] Davis L. Applying adaptive algorithms to epistatic domains [C]//Proceedings of the International Joint Conference on Artificial Intelligence. USA:Chicago, 1985:62-164.

(下转第 49 页)

总线数量的变化对系统性能影响也很大.图4显示了当总线数量发生变化时,数据移动次数(右)以及周期数(左)随之变化的情况.1、2、3、4种情况分别代表总线数为4、6、7、8.

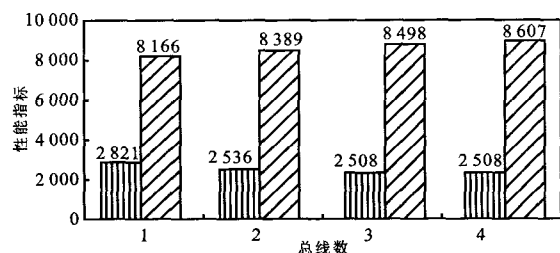


图4 不同总线数目下性能图

从图4可看出,随着总线数的减少,数据移动次数也随之减少,这主要是因为省去了数据在不同总线间的移动.随着总线的增加,算法所需要的周期数呈减少趋势,因为在数据传输中等待几率降低.

功能单元的数目达到某一数目时再增加功能单元的数量也不会带来系统性能的提高,因为数据运算不是均分在各个功能单元中,当一部分功能单元可以满足调度要求时,其余的功能单元将闲置.基于以上各项性能的分析比较得到3.2节中所述系统结构.

4.2 与通用DSP性能比较

为考察TTA架构下Intra 4×4 算法实现的效率,以TI公司的C5402作为参照.C5402 DSP具有先进的改造型哈佛结构,40位算术逻辑单元(ALU),1个 17×17 乘法器和1个40位专用加法器.

将Intra 4×4 算法输入到CCS仿真器中,得到运行周期数为15469.

根据设定,在TTA中每个加法执行需要两个周期,但在流水操作下可以认为,TTA可以像

C5402一样,一个周期给出加法的结果.考虑到TTA中有两个加法器,因此,执行Intra 4×4 算法时TTA处理器的所需要的周期数为C5402的32.3% ($2536 \times 2 / 15469 = 0.328$),即效率调高三倍.

5 结束语

基于TTA的处理器设计时,可以根据具体的目标应用对系统进行裁剪,避免了系统资源的冗余而节约了面积,且由于TTA架构在数据调度上的灵活性,系统有可以实现很高的效率;系统设计的起点为对目标应用的高级语言描述,并可生成硬件描述语言,减少了设计所需时间.鉴于多媒体应用中所表现出的数据量大、操作繁琐等特点,以TTA架构进行处理器设计将是一种更为理想的选择.

参考文献:

- [1] 陈光化,翟海华,石旭利,等. H.264编码器中 $1/4$ 像素精度插值算法的VLSI实现[J]. 微电子学与计算机, 2008,25(2):176-180.
- [2] 李霞,黄玄,周莉,等. AVS帧内预测算法分析及VLSI实现[J]. 微电子学与计算机, 2009,26(1):29-33.
- [3] 李世平,蒋刚毅. 快速帧内预测模式选择新方法[J]. 电子学报, 2006,34(1):141-146.
- [4] Hoogerbrugger J. Code generation for transport triggered architectures[D]. The Netherlands: Delft University of Technology, 1996.
- [5] Bojun Meng, Oscar C Au. Fast intra-prediction mode selection for 4×4 Blocks in H.264[C]//IEEE International Conference on Acoustics, Speech, and Signal Processing. China, Hong Kong, 2003:389-392.

作者简介:

赵丽丽 女,(1984-).研究方向为系统芯片设计.

谢 憬 硕士研究生.研究方向为SoC项目的开发与研究.

(上接第44页)

- [7] 黄岚,王康平,周春光,等. 粒子群优化算法求解旅行商问题[J]. 吉林大学学报:理学版, 2003, 41(4):477-480.
- [8] 曹平,陈盼,刘世华. 改进的粒子群算法在旅行商问题中的应[J]. 计算机工程, 2008,34(11):217-221.

作者简介:

王玉亭 男,(1980-),硕士,讲师.研究方向为软件工程、智

能优化.

孙 剑 女,(1979-),硕士,讲师.研究方向为计算机网络、人工智能.

李俊青 男,(1976-),硕士,副教授.研究方向为P2P网络、智能信息处理.

任立群 男,(1975-),工程师.研究方向为数据库.

潘全科 男,(1971-),博士,教授.研究方向为优化调度理论与方法、智能优化算法及应用.