Harmony Search

和声搜索

1 背景

和声搜索 (Harmony Search, HS) 算法是2001年由韩国学者Geem ZW等人提出[1]

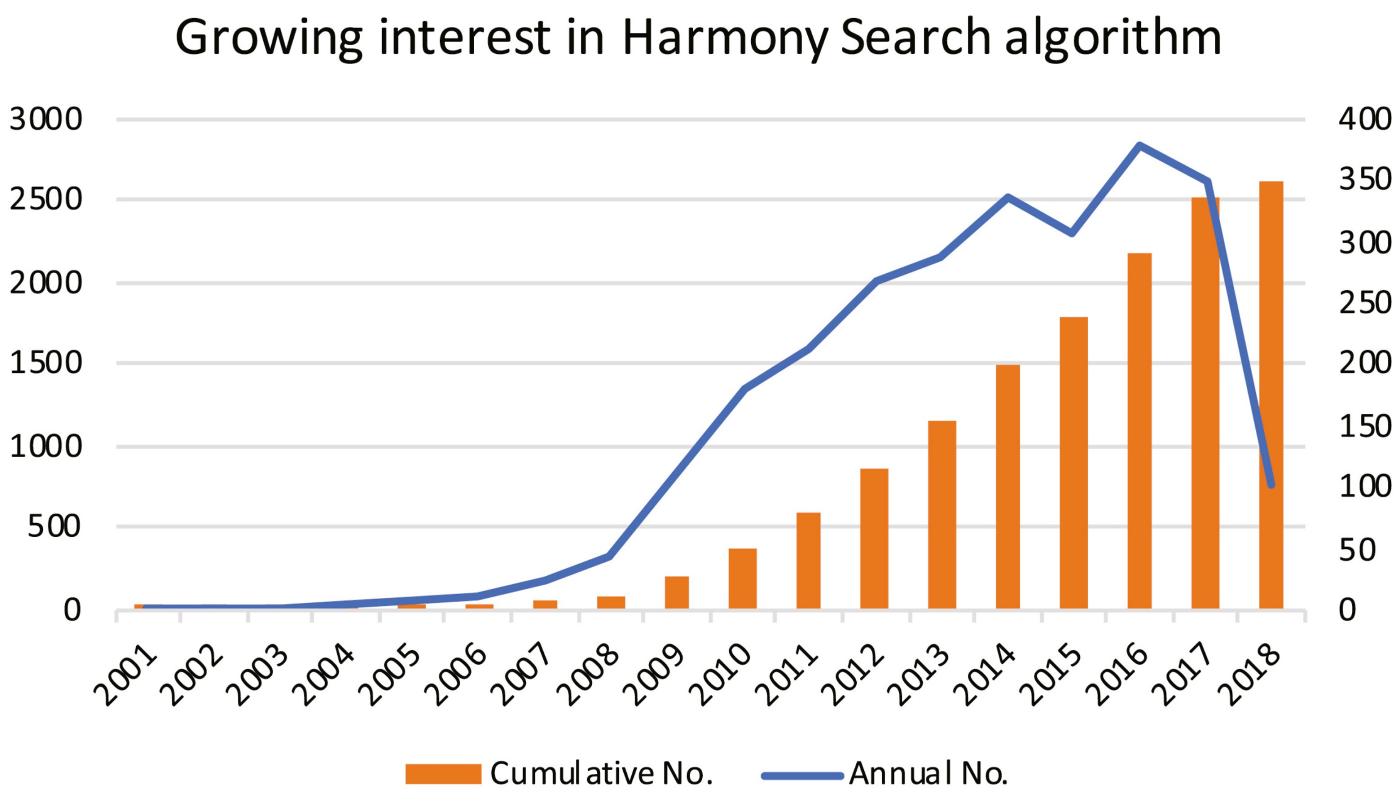
GA(遗传算法)	生物进化过程
SA(模拟退火算法)	物理退火过程
PSO(粒子群算法)	鸟群捕食行为

HS模拟了音乐演奏的原理。在音乐演奏中，乐师们凭借自己的记忆，通过反复调整乐队中各乐器的音调，最终达到一个美妙的和声状态。

[1] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation, 2001, 76(2):60−68.

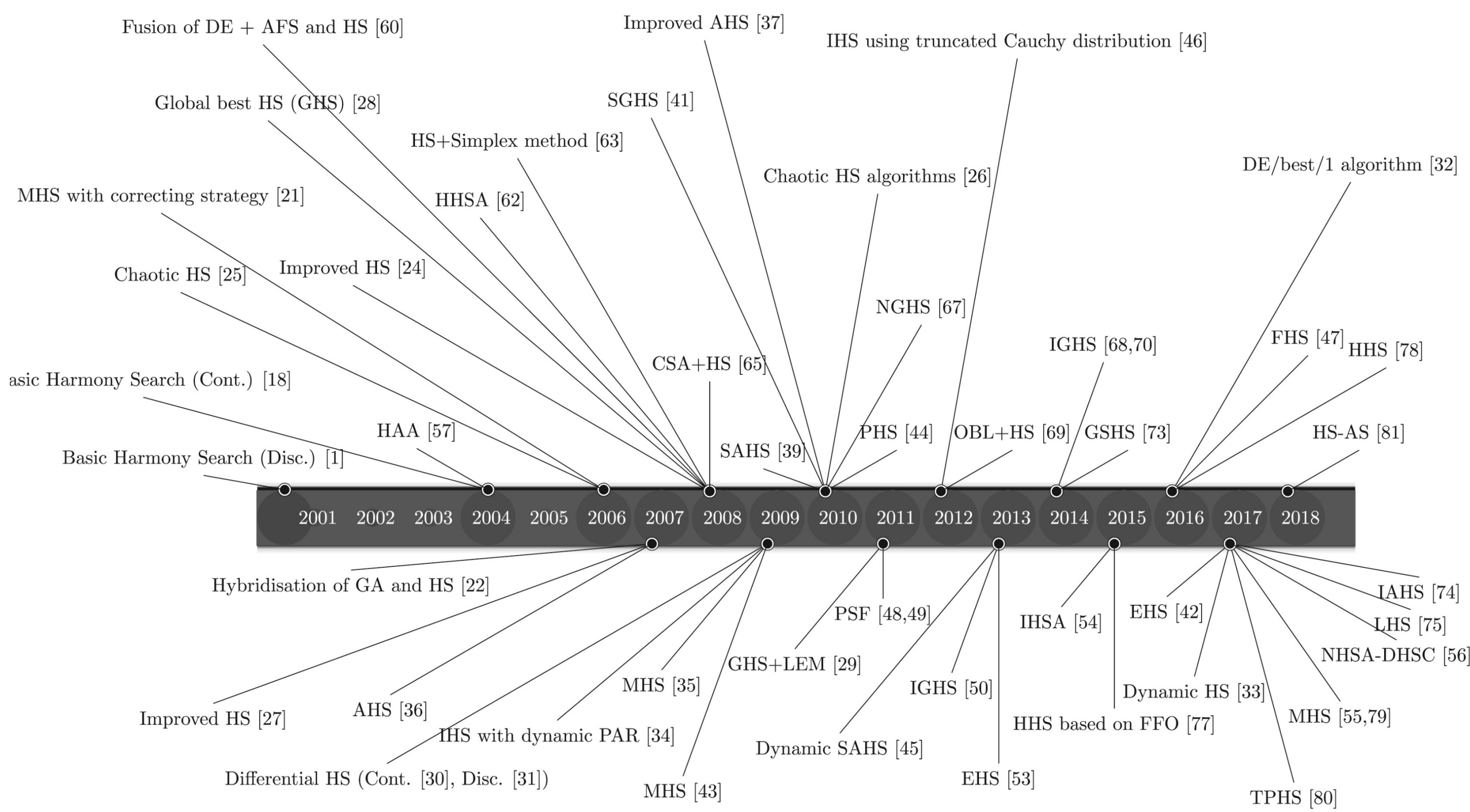
1 背景

发展趋势



[1] Zhang T, Geem Z W. Review of harmony search with respect to algorithm structure[J]. Swarm and Evolutionary Computation, 2019, 48: 31-43.

1 背景



[1] Zhang T, Geem Z W. Review of harmony search with respect to algorithm structure[J]. Swarm and Evolutionary Computation, 2019, 48: 31-43.

1 背景

乐理知识

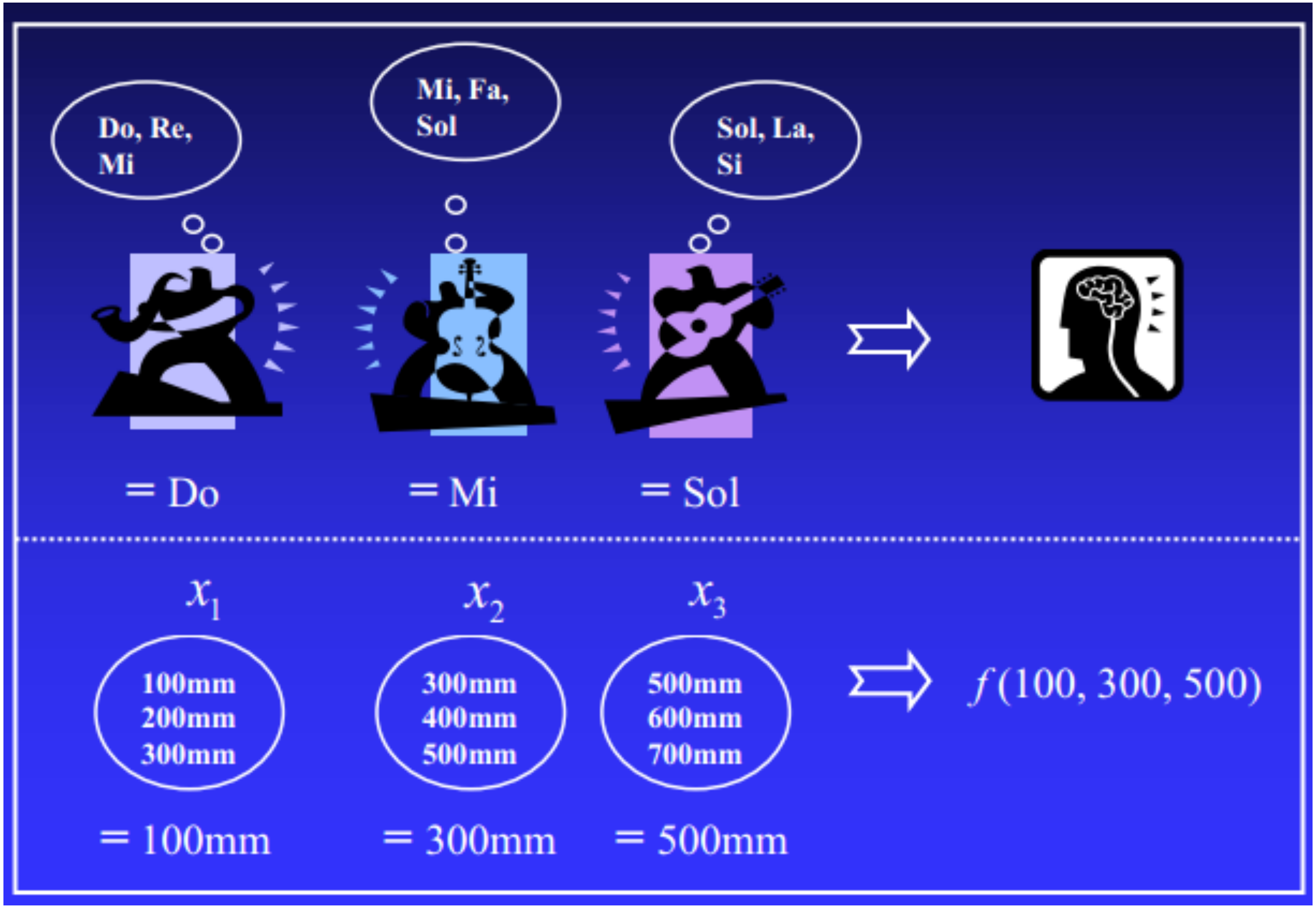
音调：声音频率的高低

音高：1234567这些音的高低

和声：由3个或者3个以上不同的音按一定的法则同时发声而构成的音响组合。

常见的和声：大三和弦(135)，主要用在大调的音乐中，比如国歌；小三和弦(1降35，613)，主要用在小调的音乐中，比如俄罗斯和日本的音乐；四六和弦，古典音乐中比较常用。四部和声，就是有4个声部的和声

1 背景



Do, Re,
Mi
Mi, Fa,
Sol
Sol, La,
Si
= Do
= Mi
= Sol
x_1
x_2
x_3
100mm
200mm
300mm
300mm
400mm
500mm
500mm
600mm
700mm
= 100mm
= 300mm
= 500mm
$f(100, 300, 500)$

2

算法介绍

音乐表演 ↔ 优化算法

音乐演奏→寻找由美学评价所决定的最佳状态

优化算法→寻找由目标函数值所决定的最优值

- 乐器 ↔ 决策变量
- 音调范围 ↔ 决策变量取值范围
- 和声 ↔ 解向量
- 审美(音乐效果评价) ↔ 目标函数
- 练习 ↔ 迭代
- 经验 ↔ 知识库

2

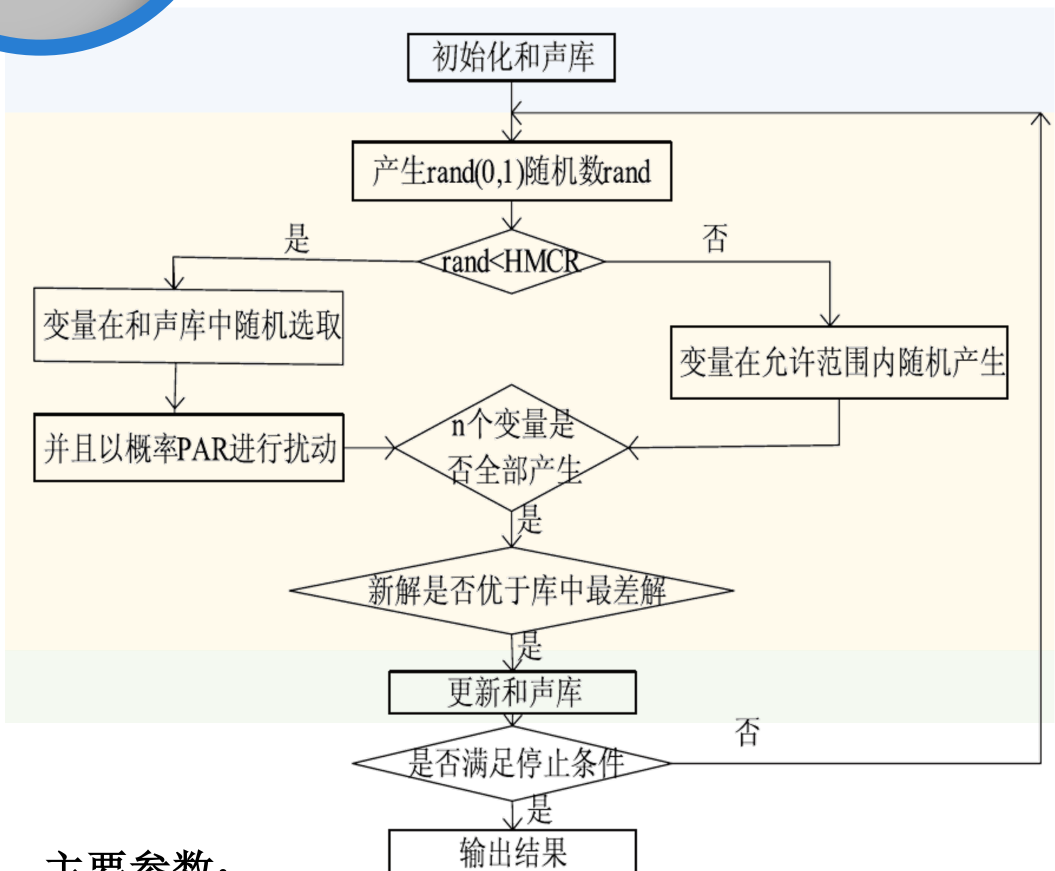
问题介绍

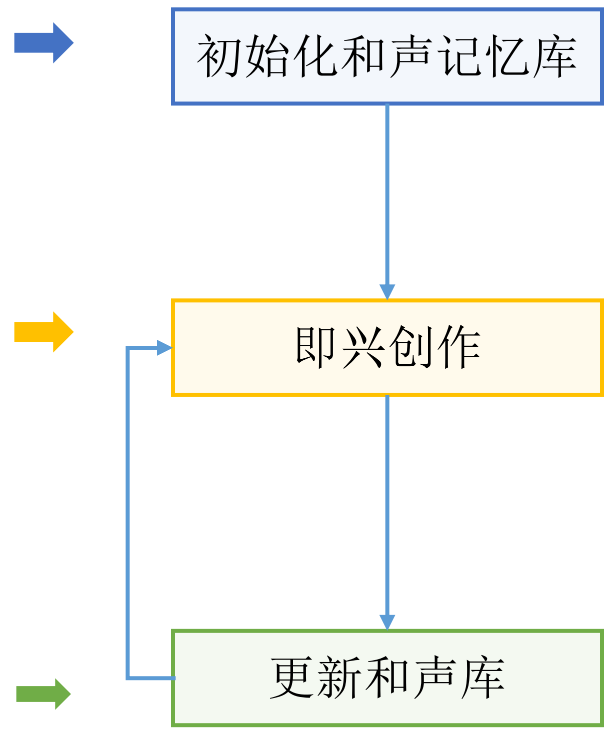
优化问题：

$$\min f(X)$$
$$s.t.\ x(i) \in D_i, i = 1,2, \dots, n$$

$f(x)$	目标函数
$X = \{x(1), \dots, x(n)\}$	解向量
$x(i)$	第i个决策变量
$D_i = [x_{min}(i), x_{max}(i)]$	连续决策变量的取值范围；
	离散决策变量则为$D_i = \big(x(i,1), x(i,2), \dots, x(i,K)\big)$
n	决策变量个数
K	离散决策变量可能值的个数

2 算法介绍





主要参数：

- 和声记忆库大小(Harmony memory size, HMS)
- 记忆库取值概率 (Harmony memory considering rate, P_{CR})
- 微调概率 (Pitch adjusting rate, P_{AR})
- 音调微调带宽dB
- 创作的次数$Tmax$

算法介绍

算法步骤

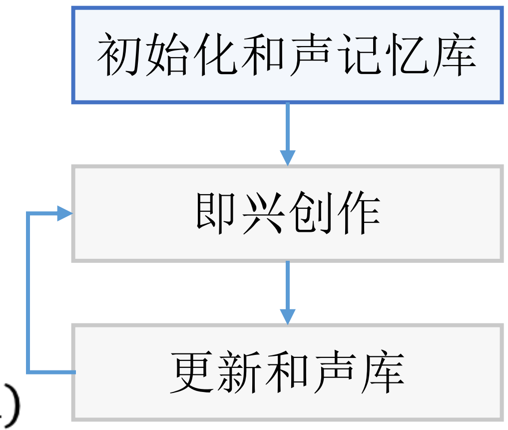
- Step1 初始化HMS个和声向量组成和声记忆库

$$HM = \{X_1, X_2, \ldots, X_{HMS}\}$$

$$X_i = \{x_i(1), x_i(2), \ldots, x_i(n)\}$$

连续型：$x_i(j) = x_{min}(j) + (x_{max}(j) - x_{min}(j)) * rand(0,1)$

离散型：$x_i(j) = x_i(j,k),\ k = rand(0,K)$



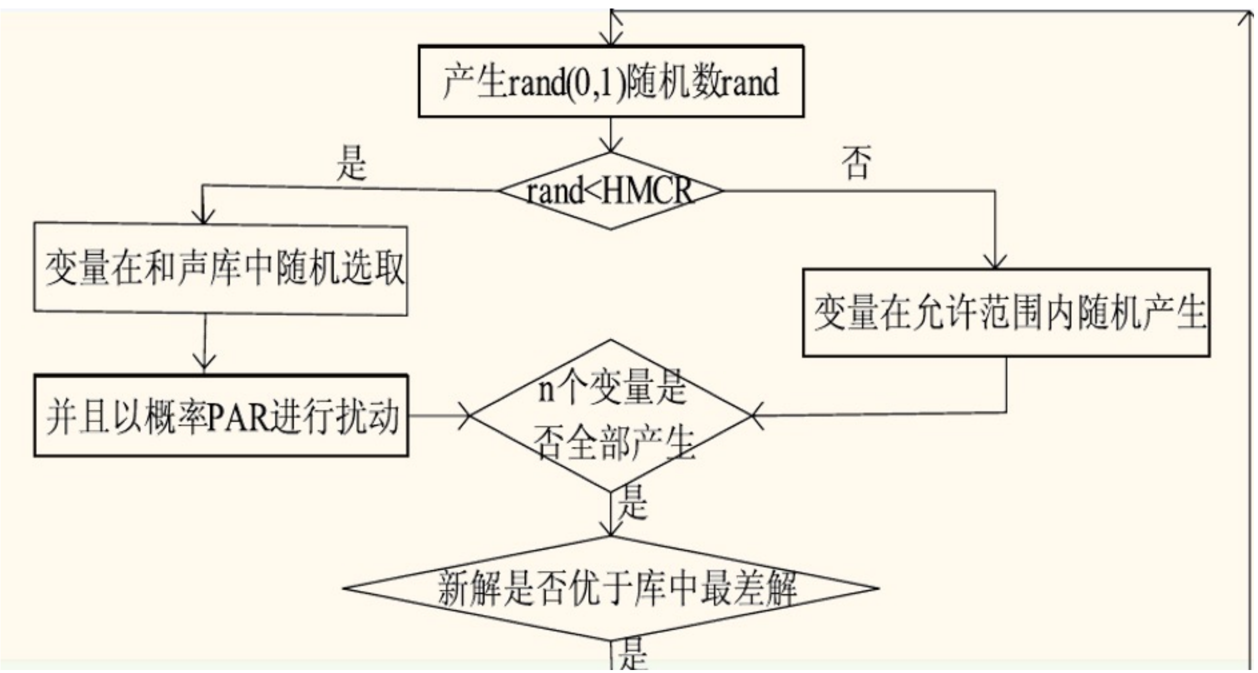
随机初始化

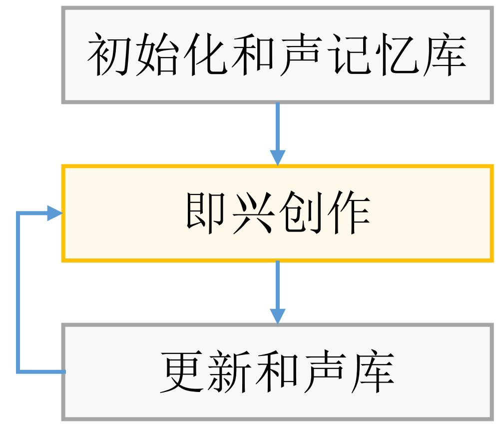
为产生性能较好的初始解，也可以利用**规则/知识**初始化

2 算法介绍

算法步骤

➢Step2 即兴创作产生一个新的和声向量



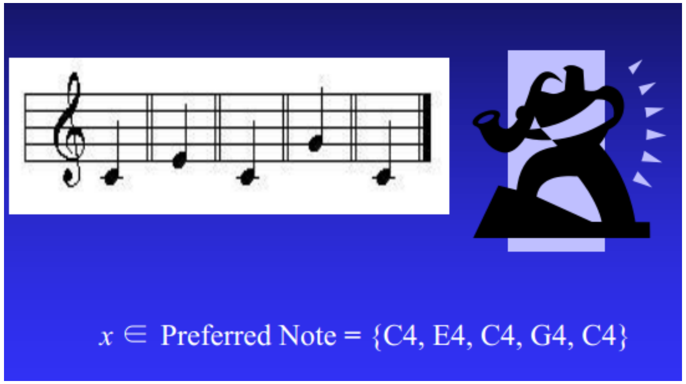


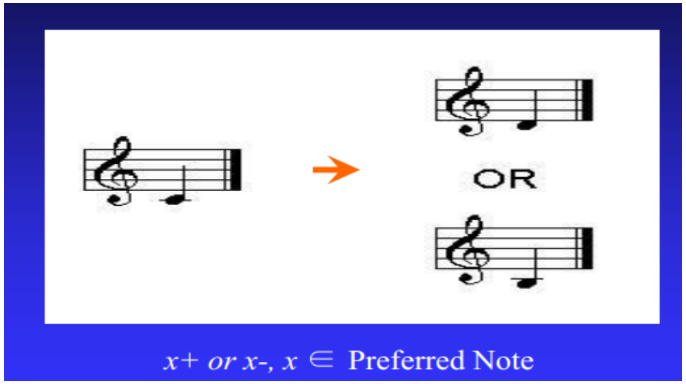
2

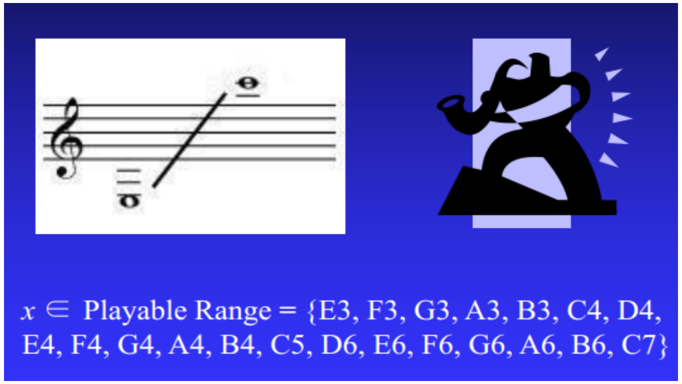
算法介绍

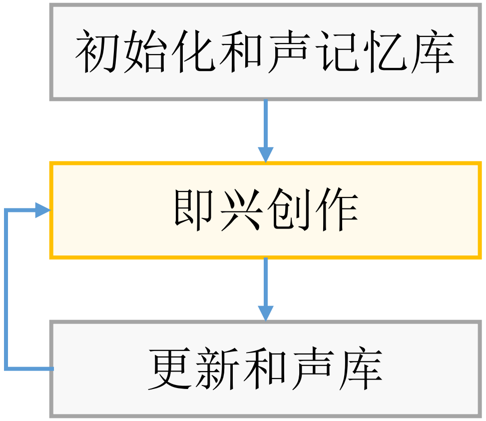
算法步骤

➢Step2 即兴创作产生一个新的和声向量









2 算法介绍

算法步骤

➢Step2 即兴创作产生一个新的和声向量

- ➢ 学习和声记忆库

  $x_{new}(j) = x_a(j)$

- ➢ 音调微调

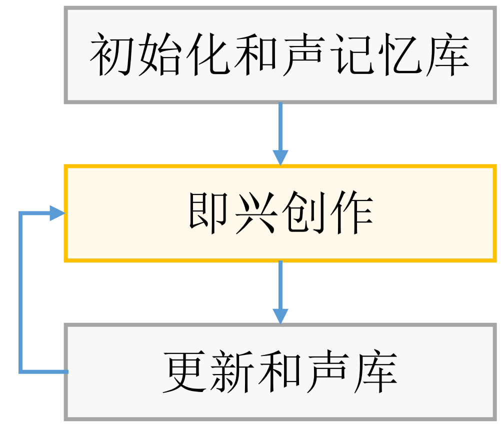
  连续型:$x_{new}(j) = x_{new}(j) \pm rand(0,1) * dB$

  离散型 $x_{new}(j) = x_{new}(j, k \pm 1)$

- ➢ 随机选择音调

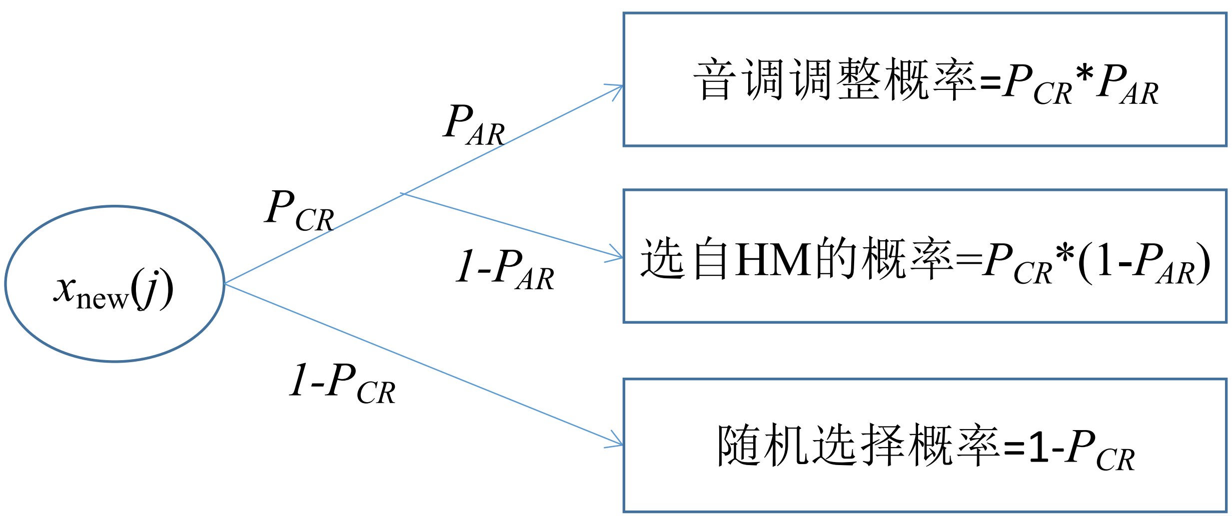
  连续型: $x_{new}(j) = x_{min}(j) + (x_{max}(j) - x_{min}(j)) * rand(0,1)$

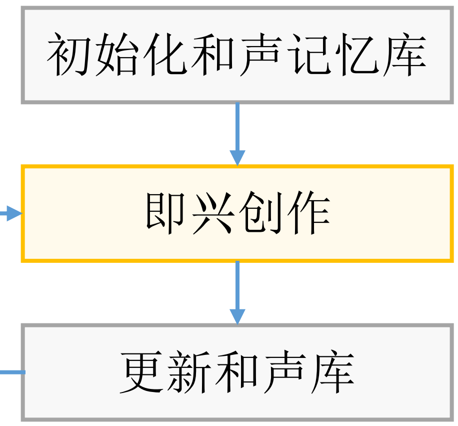
  离散型: $x_{new}(j) = x_i(j,k),\ k = rand(0,K)$



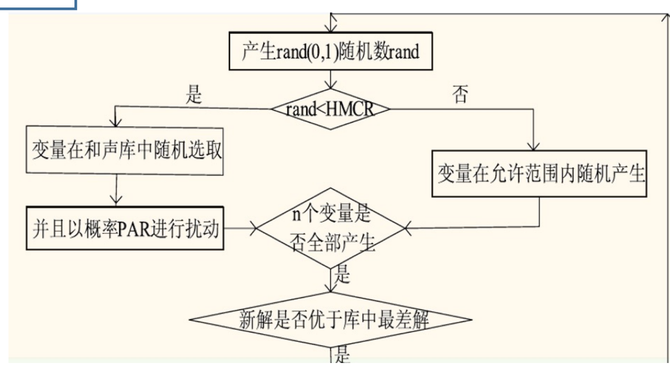
2 算法介绍

算法步骤





- 记忆库取值概率 (Harmony memory considering rate, P_{CR})
- 微调概率 (Pitch adjusting rate, P_{AR})



2

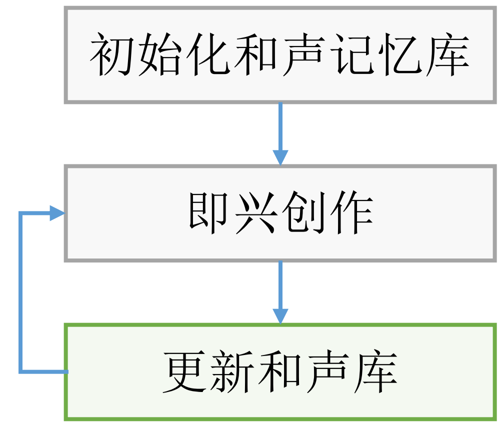
算法介绍

算法步骤

➢Step3 更新记忆库

$$X_{\text{Worst}} = X_{\text{new}} \quad \text{if } f(X_{\text{new}}) < f(X_{\text{Worst}})$$

对Step2 中的新解进行评估，如果优于HM 中的函数值最差的一个，则将新解更新至HM 中



2

算法介绍

GA 与HS 搜索机制的相似点

1. 基础HS和GA都是随机初始化
2. 都有适者生存机制，会淘汰效果较差的解
3. HS有类似于GA的交叉与变异机制

2

算法介绍

GA 与HS 搜索机制的差异比较

	GA	HS
复制	父代个体中挑选出成对的染色体	从和声库里挑选和声
交叉	任意两条染色体进行交叉：通常进行单点或双点交叉	HMS组解进行交叉采用多点交叉模式
变异	交叉后进行变异	变量有1-P_{CR}的概率逃脱和声记忆库
微调	无微调机制	选自HMS中的变量有P_{AR}的概率进行微调

2

算法介绍

算法优点

1. 通用框架，不依赖问题信息
2. 原理简单，易于实现
3. 群体搜索，具有记忆个体最优解的能力
4. 协同搜索,具有利用个体局部信息和群体全局信息指导算法进一步搜索的能力
5. 易于与其他算法混合,构造出具有更优性能的算法

3 算法示例 (1)

$$\text{Min } f(x) = (x(1)-2)^2 + (x(2)-3)^2 + (x(3)-1)^2 + 3,\ x(j) \in (0, 6)$$

初始化和声记忆库 (HMS = 3)

$X_1 = \{1.5, 2.1, 1\}$ 4.06

$X_2 = \{1.7, 3, 4.4\}$ 14.65

$X_3 = \{5.5, 3.8, 2.6\}$ 18.45

3

算法示例 (1)

$$\text{Min } f(x) = (x(1)-2)^2 + (x(2)-3)^4 + (x(3)-1)^2 + 3,\ x(j) \in (0, 6)$$

即兴创作

$$X_{\text{new}} = \{-, -, -\}$$

***rand*(0, 1)<*P*$_{CR}$, *rand*(0, 1)>*P*$_{AR}$**

$x_{\text{new}}(1) = x_2(1) = 1.7$

***rand*(0, 1)>*P*$_{CR}$**

$x_{\text{new}}(3) = 0+rand(0, 1) * 6$
$= 0.3 * 6 = 1.8$

***rand*(0, 1)<*P*$_{CR}$, *rand*(0, 1)<*P*$_{AR}$**

$x_{\text{new}}(2) = x_1(2) = 2.1$

$x_{\text{new}}(2) = x_{\text{new}}(2) + rand(0, 1)*\text{dB} = 2.1 + 0.5 * 1 = 2.6$

$$X_{\text{new}} = \{1.7, 2.6, 1.8\}$$ 3.89

3 算法示例 (1)

$$\text{Min } f(X) = (x(1)-2)^2 + (x(2)-3)^4 + (x(3)-1)^2 + 3,\ x(j) \in (0, 6)$$

更新记忆库

$$X_W = X_3$$

$$f(X_{new}) = 3.89 < f(X_W) = 18.45$$

$$X_3 = X_{new}$$

旧记忆库

$X_1 = \{1.5, 2.1, 1\}$ 4.06

$X_2 = \{1.7, 3, 4.4\}$ 14.65

$X_3 = \{5.5, 3.8, 2.6\}$ 18.45

新记忆库

$X_1 = \{1.5, 2.1, 1\}$

$X_2 = \{1.7, 3, 4.4\}$

$X_3 = \{1.7, 2.6, 1.8\}$

3 算法示例 (2)

流水线车间调度问题 (FSP)

- n个工件需要在m台机器上加工，每个工件访问机器的顺序相同(置换流水线：每台机器加工工件的顺序也相同)
- 确定工件的加工顺序，使调度目标最优

$$n\text{ jobs} \rightarrow \mathrm{M}_1 \rightarrow \mathrm{M}_2 \rightarrow \cdots \rightarrow \mathrm{M}_m$$

HS如何求解组合优化问题？

连续 ⟷ 离散

3

算法示例 (2)

- 设计连续域到离散域的映射

和声向量 ⟷ 调度解

{0.18, 0.61, 0.52} ⟷ {2, 3, 1}

- 设计离散化算法操作

和声向量

{2, 3, 1}

← 离散记忆库选择

← 离散音调调整

← 离散随机选择

3

算法示例 (2)

连续域到离散域的映射

- Largest position value (LPV)规则

$$X_i = \{0.4, 0.6, 0.3, 0.5, 0.1\}$$

第j维	1	2	3	4	5
$x_i(j)$	0.4	0.6	0.3	0.5	0.1

工件序列	2	4	1	3	5
$x_i(j)$非增排序	0.6	0.5	0.4	0.3	0.1

$$\pi_i = \{2, 4, 1, 3, 5\}$$

3

算法示例 (2)

设计离散化算法操作

- 记忆库选择

If $x_a(j)$ is not included in $x_{\text{new}}()$

$x_{\text{new}}(j) = x_a(j)$;

Else

randomly select a job that is not included in $x_{\text{new}}()$;

End If

- 随机选择

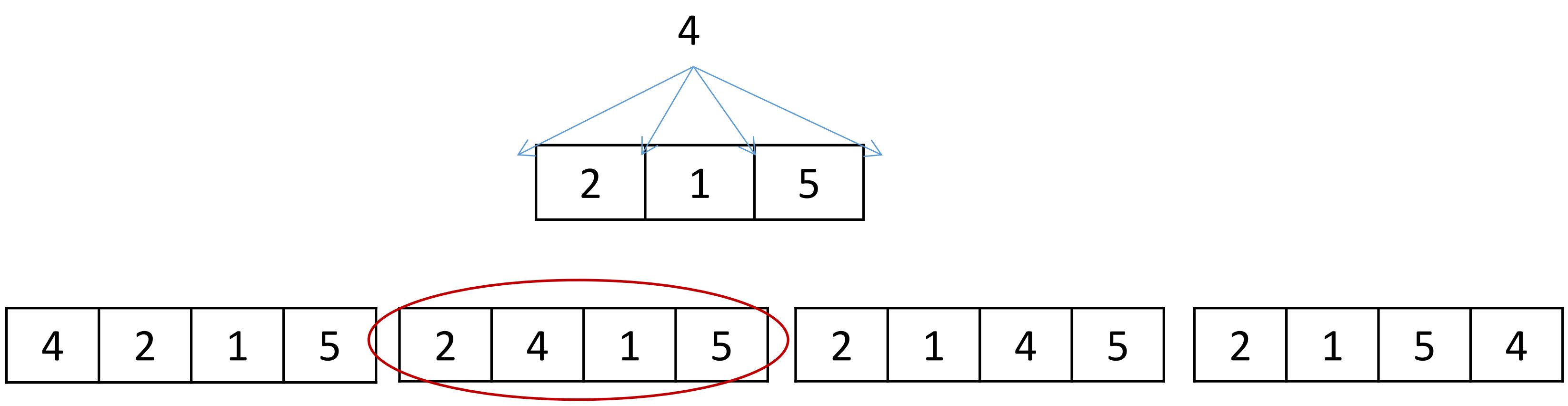
randomly select a job that is not included in $x_{\text{new}}()$

算法示例 (2)

设计离散化算法操作

- 音调调整

将第j位的工件依次插入到当前序列的第1到j位，并选择使目标值最小的序列



3

算法示例 (2)

设计离散化算法操作

$$\mathrm{HM} = \begin{bmatrix} 2 & 1 & 4 & 3 \\ 1 & 4 & 2 & 3 \\ 3 & 2 & 4 & 1 \end{bmatrix}$$

- $j = 1$ 记忆库选择：$x_{\text{new}}(1) = x_2(1) = 1$
- $j = 2$ 记忆库选择：$x_{\text{new}}(2) = x_3(2) = 2$
  音调调整：$\{2\ 1\}, \{1\ 2\} \rightarrow \{2\ 1\}$
- $j = 3$ 随机选择：$\{3, 4\} \rightarrow x_{\text{new}}(3),\ x_{\text{new}}(3) = 3$
- $j = 4$ 记忆库选择：$x_{\text{new}}(4) = x_1(4) = 3$
  $\{4\} \rightarrow x_{\text{new}}(4), x_{\text{new}}(4) = 4$

$$X_{\text{new}} = \{2, 1, 3, 4\}$$

3

算法改进

算法改进的三个方向：

- 算法参数动态调整
- 基于问题性质的搜索策略
- 和其他算法协同优化

3 算法改进

➢ 算法参数动态调整

即兴次数

$$\mathrm{PAR}(\mathrm{Imp}) = \mathrm{PAR}_{\min} + \frac{\mathrm{Imp}}{\mathrm{MaxImp}} \times (\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min}),$$

$$\mathrm{PAR}(\mathrm{Imp}) = \mathrm{PAR}_{\min} + \frac{f_{\mathrm{Imp},\max} - f_{\mathrm{mean}}}{f_{\mathrm{Imp},\max} - f_{\mathrm{Imp},\min}} \times (\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})$$

[1] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems[J]. Applied mathematics and computation, 2007, 188(2): 1567-1579.
[2] dos Santos Coelho L, de Andrade Bernert D L. An improved harmony search algorithm for synchronization of discrete-time chaotic systems[J]. Chaos, Solitons & Fractals, 2009, 41(5): 2526-2532.
[3] Lee J H, Yoon Y S. Modified harmony search algorithm and neural networks for concrete mix proportion design[J]. Journal of Computing in Civil Engineering, 2009, 23(1): 57-61.

3 算法改进

➢ 策略搜索

✓ 用前向或后向策略产生初始解

- 前向：按活动时间对活动进行升序排列，依次进行调度
- 后向：按活动时间对活动进行降序排列，依次进行调度

✓ Opposition-Based Learning 产生新解

Algorithm 3. Opposition-based initialization

for $i = 1$ to HMS **do**
  for $j = 1$ to D **do**
    $x_{i,j} = L_j + rand(0,1) \cdot (U_j - L_j)$
    $ox_{i,j} = L_j + U_j - x_{i,j}$ //opposition-based learning
  end for
end for

[1] Ponz-Tienda J L, Salcedo-Bernal A, Pellicer E, et al. Improved adaptive harmony search algorithm for the resource leveling problem with minimal lags[J]. Automation in Construction, 2017, 77: 82-92.[2] dos Santos Coelho L, de Andrade Bernert D L. An improved harmony search algorithm for synchronization of discrete-time chaotic systems[J]. Chaos, Solitons & Fractals, 2009, 41(5): 2526-2532.
[2] Xiang W, An M, Li Y, et al. An improved global-best harmony search algorithm for faster optimization[J]. Expert Systems with Applications, 2014, 41(13): 5788-5803.

3 算法改进

- 算法协同

$$
\begin{aligned}
x_{\text{new,i}} &\leftarrow x_{\text{new,i}} + y_i \times \left(x_i^U - x_i^L\right) \\
y_i &= T \times \operatorname{sign}(\mathrm{rnd} - 0.5) \times \left(\left(1 + \frac{1}{\mathrm{T}}\right)^{|2\mathrm{rnd}-1|} - 1\right)
\end{aligned}
$$

模拟退火

$$
\begin{aligned}
x_{\text{new,i}} &\leftarrow x_{\text{new,i}} + \mathrm{rnd}\left([0,1]\right) \times \left(\mathrm{x}_{\mathrm{i}}^{\mathrm{j}} - \mathrm{x}_{\mathrm{i}}^{\mathrm{k}}\right), i = 1, \ldots, \mathrm{N} \text{ and} \\
\mathrm{j}, \mathrm{k} &= \mathrm{rnd}\left(\{1, 2, \ldots, \mathrm{HMS}\}\right)
\end{aligned}
$$

差分进化

[1] Tian Y H, Bo Y M, Gao M F. Parameters choice criteria in harmony annealing for function optimization[J]. Computer Simulation, 2005, 22(4): 70-74.[2] dos Santos Coelho L, de Andrade Bernert D L. An improved harmony search algorithm for synchronization of discrete-time chaotic systems[J]. Chaos, Solitons & Fractals, 2009, 41(5): 2526-2532.
[2] Roy G G, Panigrahi B K, Chakraborty P, et al. On optimal feature selection using modified harmony search for power quality disturbance classification[C]//2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE, 2009: 1355-1360.

4

参考文献

1. Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search. Simulation, 2001, 76(2): 60-68.
2. Geem Z W, Lee K S, Park Y J. Application of harmony search to vehicle routing. American Journal of Applied Sciences, 2005, 2: 1552–1557.
3. Wang L, Pan Q K, Tasgetiren M F. Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms. Expert Systems with Applications, 2010, 37(12): 7929-7936.
4. Wang L, Pan Q K, Tasgetiren M F. A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem. Computers & Industrial Engineering, 2011, 61(1): 76-83.
5. Gao K, Pan Q, Li J. Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion. The International Journal of Advanced Manufacturing Technology, 2011, 56(5-8): 683-692.
6. **Sörensen K. Metaheuristics—the metaphor exposed. International Transactions in Operational Research, 2015, 22(1): 3-18.**