

高级搜索

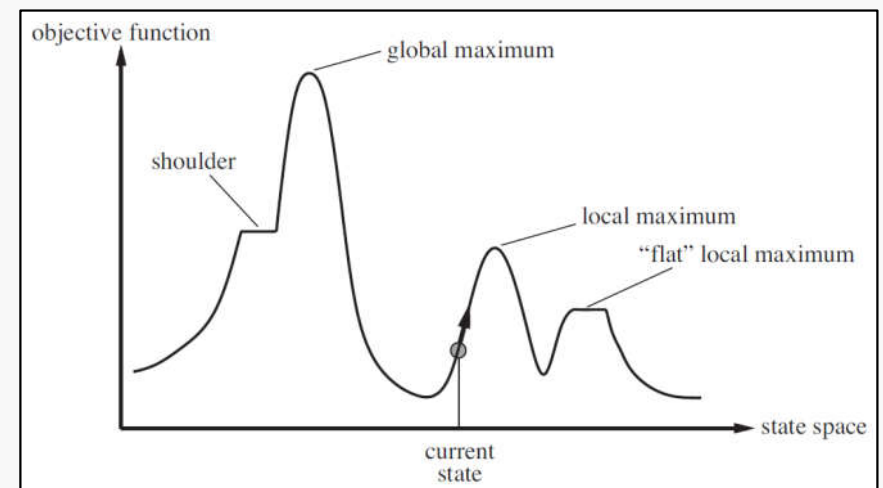
陈和港

2023. 04. 11

部分课件来源于王甲海老师课程助教团队

局部搜索

- 在工厂场地布局、作业车间调度等路径无关的问题中，我们不需要关注从初始状态到目标状态的路径，而需要关注解的状态。
 - 例如，在八皇后问题中，最终目标是皇后在棋盘上的布局，而皇后加入的先后次序并不重要。
- 局部搜索（local search）：不关心路径的算法
 - 从单个当前节点（而不是多条路径）出发，通常根据某种策略只移动到它的邻近状态。
 - 一般情况下不保留搜索路径。
- 最优化问题
 - 根据目标函数找到最佳状态。



目录

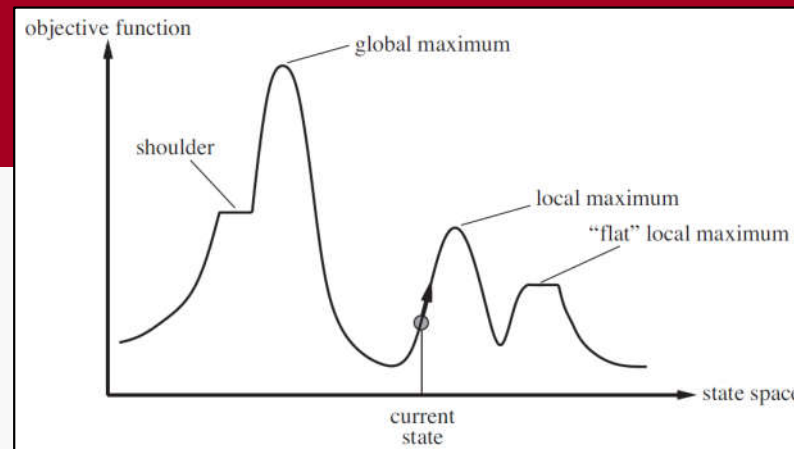
1. 爬山法

2. 模拟退火算法

3. 遗传算法

4. 遗传算法与黑白棋

爬山法



□ 爬山法又称贪婪局部搜索。

□ 不断向值增加的方向持续移动——登高：

- 从搜索空间中随机产生邻近的点，从中选择对应解更优的个体，替换原来的个体。

function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum

current \leftarrow MAKE-NODE(*problem*.INITIAL-STATE)

loop do

neighbor \leftarrow a highest-valued successor of *current*

if *neighbor*.VALUE \leq *current*.VALUE **then return** *current*.STATE

current \leftarrow *neighbor*

□ 局限：常常只能收敛到离初始位置比较近的局部最优解。

1 模拟退火算法

□ 模拟退火算法思想：有一定的概率接受更差的解

- 源于统计物理学，采用类似于物理退火的过程。
- 一开始处在高温状态（相当于算法随机搜索，大概率接受劣解）；
- 然后逐渐退火冷却（接受劣解概率变小直至为零，相当于爬山法）；
- 最终达到物理基态（相当于算法找到最优解）。

□ 算法的本质是通过温度来控制算法接受劣解的概率。

$$P = e^{\frac{\Delta E}{T}}$$

$$\Delta E = E_{new} - E_{old} \geq 0, P \geq 1。$$

$$\Delta E = E_{new} - E_{old} < 0, \text{ 有:}$$

$$\text{当 } T \rightarrow +\infty \text{ 时, } \frac{\Delta E}{T} \rightarrow 0 \Rightarrow P \rightarrow 1$$

$$\text{当 } T \rightarrow 0 \text{ 时, } \frac{\Delta E}{T} \rightarrow -\infty \Rightarrow P \rightarrow 0$$

1 模拟退火算法

□ 爬山法与模拟退火算法

function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum

current \leftarrow MAKE-NODE(*problem*.INITIAL-STATE)

loop do

neighbor \leftarrow a highest-valued successor of *current*

if *neighbor*.VALUE \leq *current*.VALUE **then return** *current*.STATE

current \leftarrow *neighbor*

function SIMULATED-ANNEALING(*problem*, *schedule*) **returns** a solution state

inputs: *problem*, a problem

schedule, a mapping from time to “temperature”

current \leftarrow MAKE-NODE(*problem*.INITIAL-STATE)

for $t = 1$ **to** ∞ **do**

$T \leftarrow$ *schedule*(t)

if $T = 0$ **then return** *current*

next \leftarrow a randomly selected successor of *current*

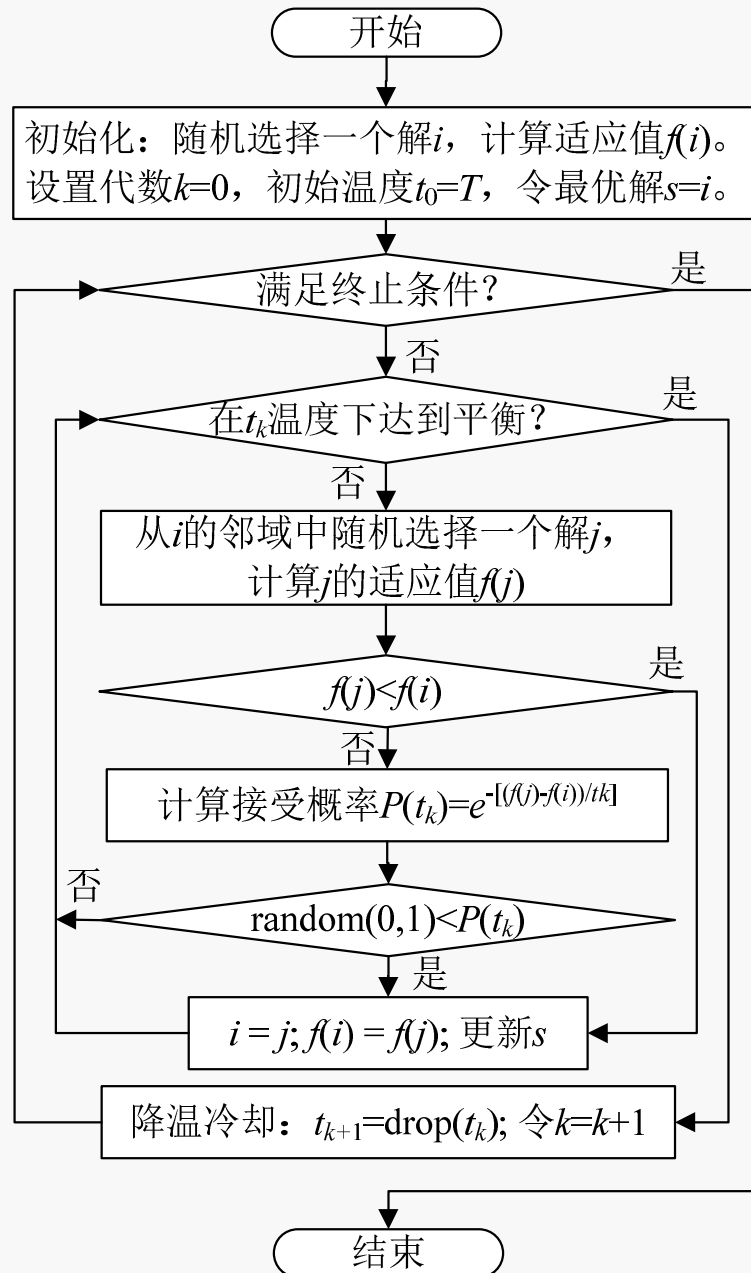
$\Delta E \leftarrow$ *next*.VALUE $-$ *current*.VALUE

if $\Delta E > 0$ **then** *current* \leftarrow *next*

else *current* \leftarrow *next* only with probability $e^{\Delta E/T}$

- 进一步地，我们还需考虑更多细节，例如，要求每个温度下达到平衡后，才进入下一个温度。

1 模拟退火算法



//功能: 模拟退火算法伪代码

//说明: 本例以求问题最小值为目标

//参数: T 为初始温度; L 为内层循环次数

procedure SA

//Initialization

Randomly generate a solution X_0 , and calculate its fitness value $f(X_0)$;

$X_{best} = X_0; k=0; t_k=T;$

while not stop

//The search loop under the temperature t_k

for $i=1$ to L //The loop times

Generate a new solution X_{new} based on the current solution X_k , and calculate its fitness value $f(X_{new})$.

if $f(X_{new}) < f(X_k)$

$X_k = X_{new};$

if $f(X_k) < f(X_{best})$ $X_{best} = X_k;$

continues;

end if

Calculate $P(t_k) = e^{-[f(X_{new})-f(X_k)]/tk};$

if random(0,1) < P

$X_k = X_{new};$

end if

end for

//Drop down the temperature

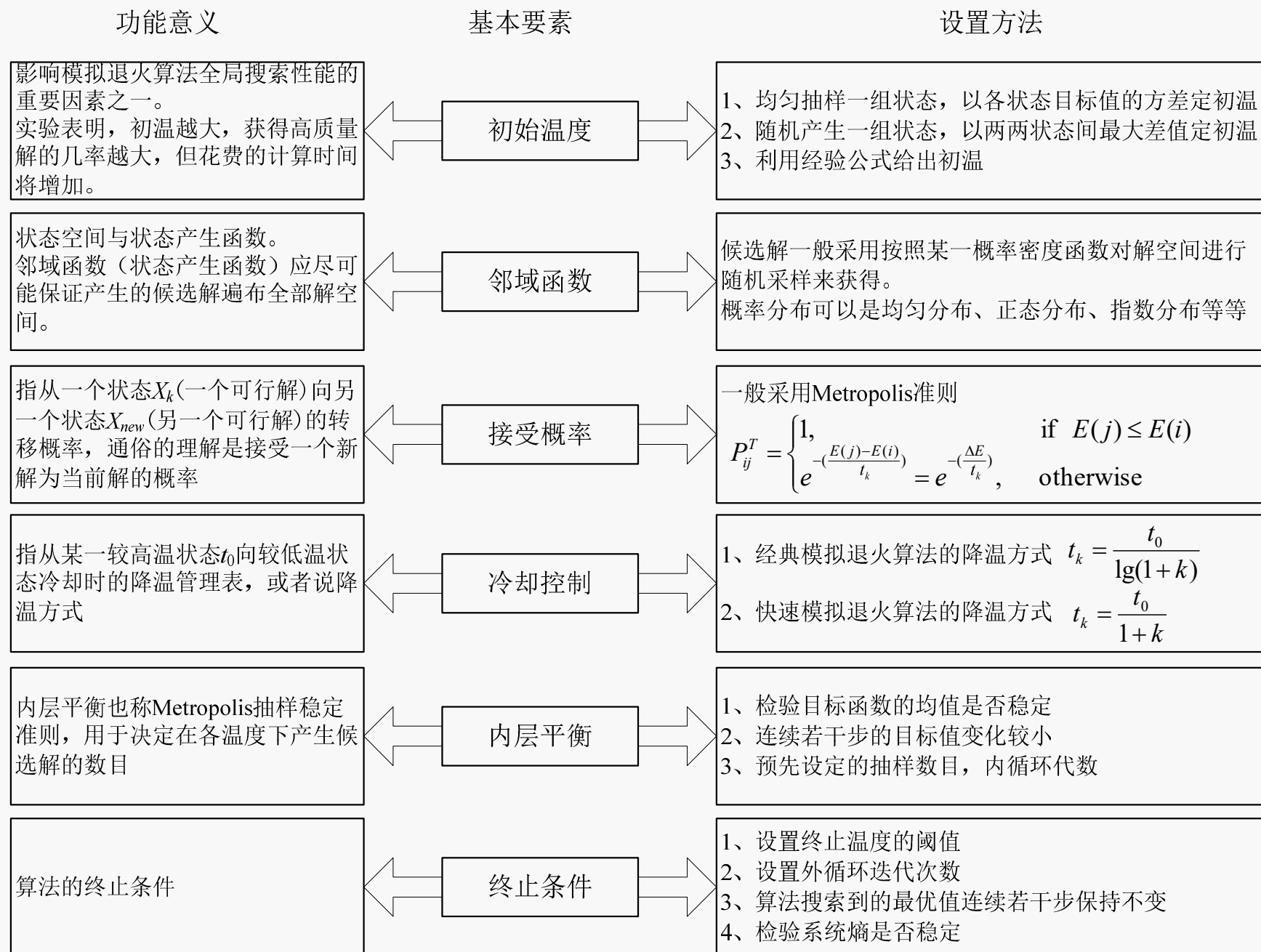
$t_{k+1} = \text{drop}(t_k); k=k+1;$

end while

print X_{best}

end procedure

1 模拟退火算法



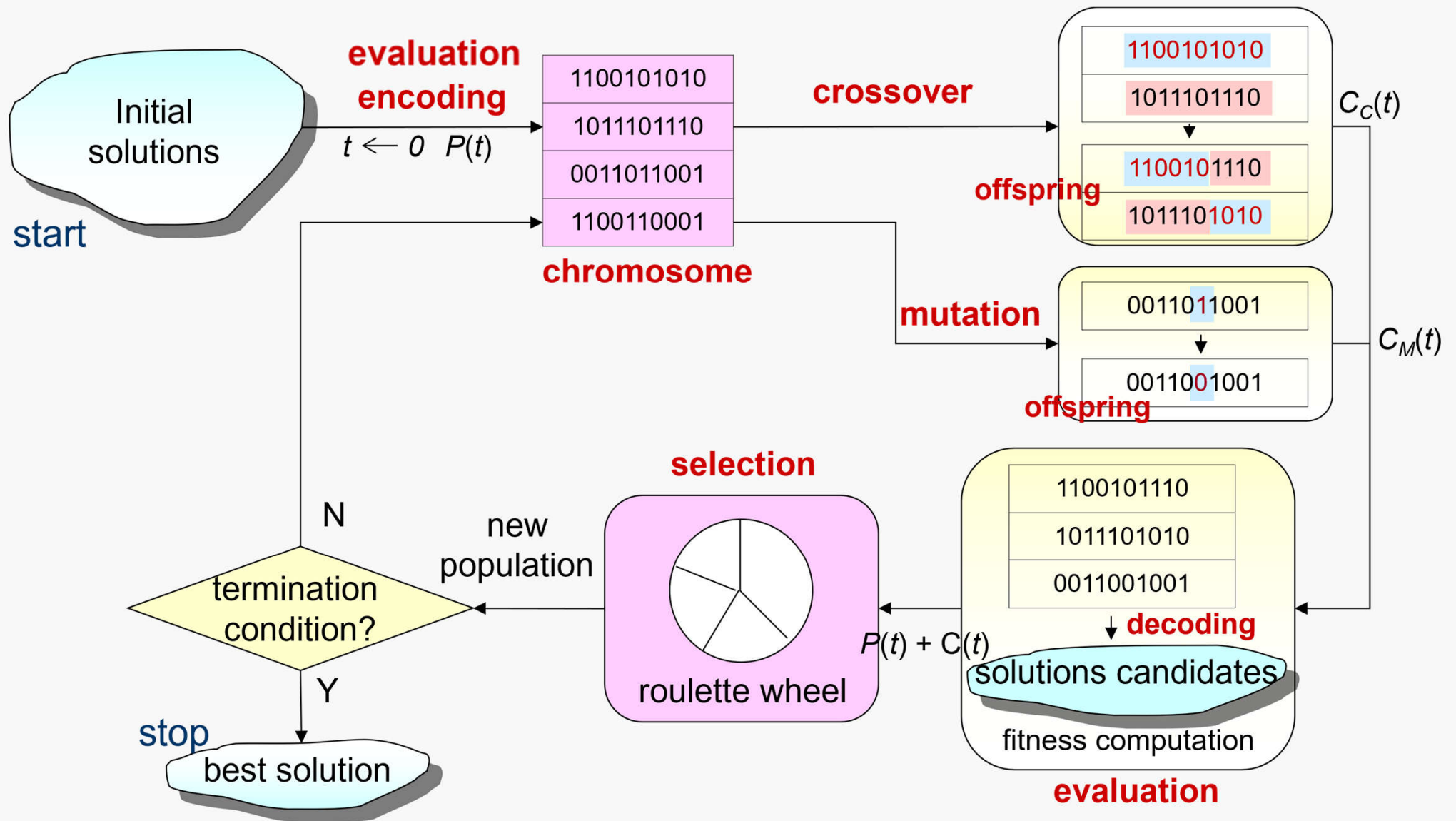
2 遗传算法

□ 遗传算法思想

- 源于进化生物学，模拟物竞天择的生物进化过程。
- 通过维护一个潜在解的群体执行了多方向的搜索，并支持这些方向上的信息构成和交换。
- 是以面为单位的搜索，比以点为单位的搜索，更能发现全局最优解。

1. 随机产生初始种群，其中的个体被**编码**为染色体；
2. 抽出父母双方的染色体，进行**交叉**，产生子代；
3. 对染色体进行**变异**，产生子代；
4. 评估每条染色体所对应个体的**适应度**；
5. 遵照适应度越高，选择概率越大的原则，**选择**出新的种群。
6. 如果种群已足够好，则输出其中最优的个体，否则返回第2步。

2 遗传算法



2 遗传算法

procedure: Simple GA

input: GA parameters

output: best solution

begin

$t \leftarrow 0;$

// t : generation number

initialize $P(t)$ by **encoding routine**;

// $P(t)$: population of chromosomes

fitness $eval(P)$ by **decoding routine**;

while (not termination condition) do

crossover $P(t)$ to yield $C(t)$;

// $C(t)$: offspring

mutation $P(t)$ to yield $C(t)$;

 fitness $eval(C)$ by **decoding routine**;

select $P(t+1)$ from $P(t)$ and $C(t)$;

$t \leftarrow t+1$;

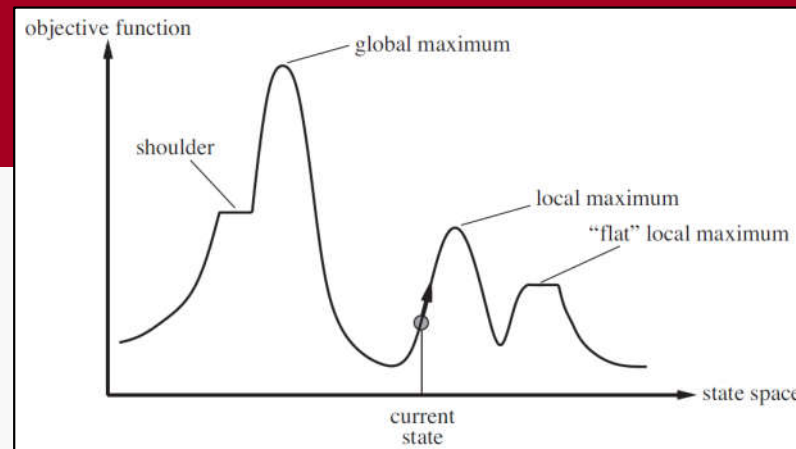
end

output best solution;

end

总结：“袋鼠跳”类比

每一个解就是一只袋鼠，我们希望它们不断的向着更高处跳去，直到跳到最高的山峰。



□ 爬山法

在爬山法中，袋鼠最有希望到达最靠近它出发点的山顶，但不能保证该山顶是珠穆朗玛峰，或者是一个非常高的山峰。因为一路上它只顾上坡，没有下坡。

□ 模拟退火算法

在模拟退火中，袋鼠喝醉了，而且随机地大跳跃了很长时间。运气好的话，它从一个山峰跳过山谷，到了另外一个更高的山峰上。但最后，它渐渐清醒了并朝着它所在的峰顶跳去。

总结：“袋鼠跳” 类比

□ 遗传算法

在遗传算法中，有很多袋鼠，它们降落到喜马拉雅山脉的任意地方。这些袋鼠并不知道它们的任务是寻找珠穆朗玛峰。但每过几年，就在一些海拔高度较低的地方射杀一些袋鼠，并希望存活下来的袋鼠是多产的，在它们所处的地方生儿育女。

从前，有一大群袋鼠，它们被莫名其妙的零散地遗弃于喜马拉雅山脉。于是只好在那里艰苦的生活。海拔低的地方弥漫着一种无色无味的毒气，海拔越高毒气越稀薄。可是可怜的袋鼠们对此全然不觉，还是习惯于活蹦乱跳。于是，不断有袋鼠死于海拔较低的地方，而越是在海拔高的袋鼠越是能活得更久，也越有机会生儿育女。就这样经过许多年，这些袋鼠们竟然都不自觉地聚拢到了一个个的山峰上。事实上，在所有的袋鼠中，只有聚拢到珠穆朗玛峰的袋鼠才能被带回美丽的澳洲。

遗传算法与黑白棋

□ 黑白棋

黑白棋由黑、白两种棋子组成，各 32 枚，棋盘是 64 个 8×8 的正方格，棋子要落在方格内。黑白棋的具体规则如下：

- 1) 在每一场棋局开始时，棋盘的正中间有交叉放置的黑、白棋各两枚，执黑棋一方先落子。
- 2) 在空的方格新落下己方棋子，同时使对手一枚及以上棋子翻转，为一次合法的落子。
- 3) 对手棋子被翻转的条件：己方新落下的棋子与棋盘上已有的同色棋子间，被夹住的所有对手棋子(和己方棋子中间无间隙)，夹住的方向可以是任何方向。
- 4) 每次落子可以翻转所有满足条件(3)的对手棋子，可以翻转的对手棋子必须被翻转。
- 5) 若棋盘上没有位置能实现翻转对方的棋子，则本轮由对手选择落子的位置，直到己方有合法的落子位置为止。
- 6) 如果一方有合法的落子位置，则必须选择落子，不可以选择放弃。
- 7) 棋局持续下去，直到棋盘填满或者双方都无合法棋步可下。
- 8) 当棋盘被下满时，游戏结束；当棋盘上只有一方的棋子时，游戏结束。游戏结束后，棋盘上剩余棋子多的一方赢得比赛。

遗传算法与黑白棋

□ 遗传算法

黑白棋的棋力强弱在于估值函数以及搜索算法，遗传算法主要通过优化估值函数来提升AI的棋力。

- 1) 位置价值：由于规则的限制，棋盘上不同的位置存在优劣。
- 2) 行动力：落子方可以选择下棋的位置数 A 减去另一方棋的位置个数 B 。如果 $A > B$ ，我们认为当前走棋一方是占优的，否则便是处于劣势的
- 3) 稳定子：稳定子指的是无论对方怎么走，我方都永远不会被翻转的棋子。
- 4) 棋子数：棋子数当前黑棋或者白棋的个数。
- 5) 奇偶性：经过研究，黑白棋被认为是后手优势的。所以我们认为，如果没有某方弃权，最后落下棋子的一方将会是谁，那么谁就会获得有利地位。黑棋在这方面有利时，取值为1.否则为-1.