第二章 禁忌搜索

清华大学数学科学系 邢文训

wxing@tsinghua.edu.cn

Tel: 62787945

Office:理科楼A416

Official hour:周四16: 00-17: 00

基本思想

- Glover在1986年提出这个概念
- 局部搜索算法的推广。
- 平用了禁忌技术,禁止重复前面的工作。
- 禁忌表记录下已经到达过的局部最优点或 达到局部最优的一些过程,利用禁忌表中 的信息不再或有选择地搜索这些点或过程, 以此来跳出局部最优点。

主要内容

- ■局部搜索
- 禁忌搜索
- ●技术问题
- ●应用实例——图节点着色和车间作业排序

2.1局部搜索

应用问题

$$\min f(x)$$

$$s.t. \ g(x) \ge 0,$$

 $x \in D.$

局部搜索算法:

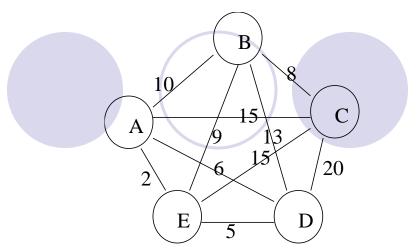
STEP1 选定一个初始可行解: x^0 ; 记录当前最优解: $x^{best} := x^0$, $T = N(x^{best})$;

STEP2 当 $T \setminus \{x^{best}\} = \emptyset$ 时,或满足其他停止运算准则时,输出计算结果,停止运算;

否则,从 T 中选一集合 S, 得到 S 中的最好解 x^{now} ; 若 $f(x^{now}) < f(x^{best})$, 则

 x^{best} := x^{now} , $T=N(x^{best})$;否则,T:=T-S;重复 STEP2。

五个城市的对称 TSP



初始解为 x^{best} ={ABCDE}, $f(x^{best})$ =45。固定A城市为起点。

情况 1: 全邻域搜索, 即 $S:=N(x^{best})$

第一循环: $N(x^{best}) = \{(ABCDE), (ACBDE), (ADCBE), (AECDB), (ABDCE), (ABEDC), (ABDCE), (ABDCE$

(ABCED)}, 对应目标函数值为: $f(x) = \{45, 43, 45, 60, 60, 59, 44\}$ 。 $x^{best} : = x^{now} = (ACBDE)$ 。

第二循环: $N(x^{best})$ ={(ACBDE), (ABCDE), (ADBCE), (AEBDC), (ACDBE), (ACEDB),

(ACBED)}, 对应目标函数值为: $f(x) = \{43, 45, 44, 59, 59, 58, 43\}$, x^{best} :

 $=x^{now}=(ACBDE)$ 。此时, $N(x^{best})$ —S为空集,于是所得解为(ADCBE),目标值为43。

情况2:一步随机搜索。

$$x^{best} = (ABCDE), f(x^{best}) = 45$$

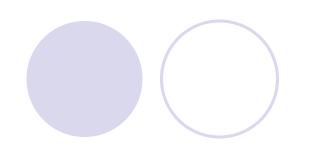
第一循环:由于采用 $N(x^{best})$ 中的一步随机搜索,可以不再计算 $N(x^{best})$

中每一点的值。若从中随机选一点,如 x^{now} =(ACBDE)。因 $f(x^{now})$ =43<45,

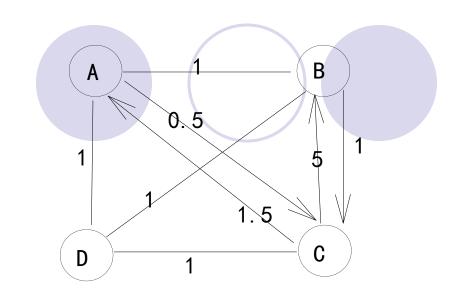
所以 x^{best} : =(ACBDE)。

第二循环: 若从 $N(x^{best})$ 中又随机选一点 x^{now} =(ADBCE),

 $f(x^{now})$ =44>43。 $N(x^{best})$ = $N(x^{best})$ -{ x^{now} }。最后得到的解为(ACBDE)。



局部最优性: 四城市非对称TSP



初始解为 x^{best} =(ABCD),并且假设城市A为起始点, $f(x^{best})$ =4。

邻域 $N(x^{best}) = \{(ABCD,4),(ACBD,7.5),(ADCB,8),(ABDC,4.5)\}$ 中,

局部最优解是(ABCD)。该算法终止时的解是局部最优解(ABCD)。

而全局最优解是 x^{best} =(ACDB), $f(x^{best})$ =3.5。

第一节小结

- 局部搜索算法的计算结果主要依赖起点的选取和 邻域的结构。
- 同一个起点,不同的邻域结构会得到不同的计算结果。
- 同一个邻域结构,不同的初始点会得到不同的计算结果。
- 为了得到好的解,可以比较不同的邻域结构和不同的初始点。
- 如果初始点的选择足够多,总可以计算出全局最优解。如sampling统计方法,参考[邢2019]。
 - ○[邢2019]**全局优化**,中国学科发展战略——数学优化,中国科学院,科学出版社,73-81,2020.

2.2 禁忌搜索

改进局部搜索:邻域结构2-opt

$$D = (d_{ij}) = \begin{vmatrix} 0 & 1 & 0.5 & 1 \\ 1 & 0 & 1 & 1 \\ 1.5 & 5 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{vmatrix}$$

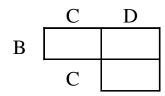
第一步:

解的形式

A B C D

 $f(x^0) = 4$

禁忌对象及长度



候选集

对换 评价值

C,D	4.5★
В,С	7.5
B,D	8

f(ABDC)=4.5, f(ACBD)=7.5, f(ADCB)=8. 选择使得评价值最优的对换CD, 得到新的迭代点(ABDC).

第二步:

解的形式

禁忌对象及长度

В

候选集

对换 评	价值
------	----

В,С	3.5★
B,D	4.5
C,D	4.5T

$f(x^1) = 4.5$

В

第三步:

禁忌对象及长度

,	候.	选	集

对换 评价值

	C	D
В	3	
	С	2

D

3

	-	-			
			\sim		
			\boldsymbol{C}		

 C

В,С	4.5T
B,D	7.5★
C,D	8T

解的形式

A C D B

 $f(x^2) = 3.5$

第四步:

解的形式

禁忌对象及长度

候选集

A C	В	D
-----	---	---

 $f(x_3) = 7.5$

$$\begin{array}{c|cccc} C & D \\ \hline B & 2 & 3 \\ \hline C & 1 \\ \hline \end{array}$$

对换 评价值

- 4 4/ +	* 1 1/ 1 1111
B,D	3.5T
В,С	4.5T
C.D	4.5T

问题

选择禁忌的对象?

禁忌的长度如何选取?

如果禁忌长度为2: CD和BC交换被允许.

第四步

解的形式

A C B D

 $f(x^3) = 7.5$

第五步:

禁忌对象及长度

候选集

对换 评价值

B,D	3.5T
В,С	4.5T
C,D	4.5★

解的形式

A D B C

 $f(x^4) = 4.5$

禁忌对象及长度

$$\begin{array}{c|cc} & C & D \\ B & 0 & 1 \\ \hline & C & 2 \\ \end{array}$$

候选集

对换评价值

B , D	4.5 T
C , D	7.5 T
B , C	8★

- 禁忌的长度如何选取?
 - ○禁忌长度短会造成循环,也就可能在一个局部最优 解附近循环。
 - ○禁忌长度长会造成算法的记忆存储量增加,使得算法计算时间增加,同时可能造成算法无法继续计算下去。
- ●候选集合如何选取?
- •是否有评价值的其他替代形式?
- ●如何利用更多的信息?
- ●终止原则怎样给出?

禁忌搜索算法

STEP 1 给予禁忌表(tabu list) $H = \emptyset$ 并选定一个初始解 x^{now} ;

STEP 2 满足停止规则时,停止计算,输出结果;否则,在 x^{now} 的

邻域 $N(x^{now})$ 中选出满足不受禁忌或解禁元素组成的候选集

 $Can_N(x^{now})$; 在 $Can_N(x^{now})$ 中选一个评价值最佳的解

 x^{next} , x^{now} : $=x^{next}$;更新历史记录 H,重复 Step 2。

定义 集合 C 称为相对邻域映射 N: $x \in C \rightarrow 2^C$ 是连通的,若对 C 中

的任意两点 x,y,存在互异的 $x=x_1,x_2,\cdots,x_l=y$, 使得

 $N(x_i) \cap \{x_{i+1}\} \neq \emptyset, i = 1, 2, \dots, l-1$.

最优性必要条件: 在禁忌搜索算法中, 若解区域相对邻域映射 N是连通的,则可以构造禁忌算法, 使得算法求得全局最优解。

算法的构造如下:从任何初始解x开始,禁忌表中记录所有一步 (除x外)可达的所有点,即邻域中的所有点,再以这些点为起 点,一步步地记录所有可行的路线。禁忌的对象是不重复已记 录的道路。很明显,这样的算法一定可以到达任何一点y。

2.3 技术问题

- 变化因素与禁忌
 - ○解的简单变化x→y
 - ●禁忌y→x
 - ○向量分量的变化

$$(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) \to (x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n)$$

- ●禁忌y_i →x_i
- ○目标值变化

$$H(a) = \left\{ x \in D | f(x) = a \right\}$$

■禁忌b→a

禁忌对象为简单的解变化。禁忌表最多只记忆四个被禁的解,即禁忌长度为4。

从 2-opt 邻域 $N(x^{now})$ 中选出最佳的五个解组成候选集 $Can_N(x^{now})$;初始

 $Can_N(x^{now}) = \{(ACBDE;43), (ABCDE;45), (ADCBE;45), (ABCED;44), (ABEDC;59)\},\$

 $x^{next} = (ACBDE).$

从候选集中选最好的一个。

第二步

$$x^{now} = (ACBDE), f(x^{now}) = 43, H = \{(ABCDE; 45), (ACBDE; 43)\},$$

 $Can_N(x^{now}) = \{(ACBDE; 43), (ACBED; 43), (ADBCE; 44), (ABCDE; 45), (ACEDB, 58)\},\$

从候选集中选最好的一个。由于(ACBDE)受禁,所以选 x^{next} =(ACBED)。

第三步

$$x^{now} = (ACBED)$$
, $f(x^{now}) = 43$,

 $H = \{(ABCDE; 45), (ACBDE; 43), (ACBED; 43)\},\$

 $Can_N(x^{now}) = \{(ACBED; 43), (ACBDE; 43), (ABCED; 44), \}$

(AEBCD;45), (ADBEC;58)}.

由于. (ACBDE;43),(ACBED;43) 受禁,所以选 x^{next} =(ABCED)。 第四步

 $x^{now} = (ABCED)$, $f(x^{now}) = 44$,

 $H = \{(ABCDE;45), (ACBDE;43), (ACBED;43), (ABCED;44)\},\$

 $Can_N(x^{now}) = \{(ACBED; 43), (AECBD; 44), (ABCDE; 45), (ABCED; 44), (ABDEC; 58)\}.$

 x^{next} =(AECBD)。此时 H 已达 4 个解,新选入的解替代最早被禁的解。

第五步

$$x^{now} = (AECBD), \quad f(x^{now}) = 44,$$

$$H = \{(ACBDE;43), (ACBED;43), (ABCED;44), (AECBD;44)\},\$$

$$Can_N(x^{now}) = \{(AEDBC; 43), (ABCED; 44), (AECBD, 44), (AECDB; 44)$$

(AEBCD;45)}.
$$x^{next} = (AEDBC)_{\circ}$$

禁忌TSP城市间顺序的对换变化

H 只记忆三对对换时。若(X,Y)在禁忌表中,表示禁忌 X 与 Y 或 Y 与 X 的交换。从 2-opt 邻 域 $N(x^{now})$ — $\{x^{now}\}$ 中选出最佳的五个状态对应的交换对组成候选集 $Can_N(x^{now})$,因为

受禁的是交换对,因此不考虑没有变化的 x^{now} 。初始解 $x^{now} = x^0 = (ABCDE)$, $f(x^0) = 45$ 。第一步

$$x^{now} = (ABCDE)$$
, $f(x^{now}) = 45$, $H = \emptyset$, $Can_N(x^{now}) = \{(ACBDE; 43),$

(ADCBE;45), (AECDB;60), (ABEDC;59), (ABCED;44)}, x^{next} =(ACBDE).

由于H为空集,从候选集中选最好的一个,它是B与C的对换构成。

第二步

$$x^{now} = (ACBDE), \quad f(x^{now}) = 43, \quad H = \{(B, C)\},$$

 $Can_N(x^{now}) = \{(ACBDE, 43), (ABCDE, 45), (ADBCE, 44), (ACEDB, 58), (ACBED, 43)\},$

选 x^{next} =(ACBED)。它是D和E对换。

第三步

$$x^{now} = (ACBED)$$
, $f(x^{now}) = 43$, $H = \{(B, C), (D, E)\}$,

 $Can_N(x^{now}) = \{(ACBDE; 43), (ABCED; 44), (AEBCD; 45), (ADBEC; 58), (ACEBD; 58)\}$

由于受禁,所以选 x^{next} =(AEBCD)。

禁忌还应该考虑方向的变化。可以考虑x与y交换和y与x交换都被禁忌。如禁忌B和C对换时,可以考虑禁忌C与B对换和B与C对换,这是双向禁忌。

反方向禁忌避免上一步已经变化的两个元素再变化回去。

正向禁忌的出发点是已经选择正向的变化,为了考虑更大范围的变化,不应该马上再考虑这种变化。

目标值变化的禁忌: H 只记忆三个目标值,从 2-opt 邻域 $N(x^{now})$ 中选出最佳

的五个元素为候选集 $Can_N(x^{now})$;在 $Can_N(x^{now})$ 中选一个目标值最佳

的解
$$x^{next}$$
。初始解 $x^{now} = x^0 = (ABCDE)$, $f(x^0) = 45$

第一步

$$x^{now} = (ABCDE), \quad f(x^{now}) = 45, \quad H = \{45\},$$

$$Can_N(x^{now}) = \{(ABCDE; 45), (ACBDE; 43), (ADCBE; 45), (ACBDE; 45), (ACDE; 45), (ACBDE; 45), (ACDE; 45), ($$

(ABEDC;59),(ABCED;44)},

$$x^{next} = (ACBDE).$$

从候选集中选最好的一个。

第二步 $x^{now} = (ACBDE), \quad f(x^{now}) = 43, \quad H = \{45,43\},$

$$Can_N(x^{now}) = \{ (ACBDE; 43), (ACBED; 43), (ADBCE; 44), (ABCDE; 45), (ACEDB, 58) \},$$

由于函数值 43 受禁,选候选集中不受禁的最佳函数值 44 的状态。

● 禁忌的范围较第一种情况要广泛。

 $x^{next} = (ADBCE)_{\circ}$

- 禁忌范围小——搜索的解个体仔细,但可能搜索的点较多。
- 禁忌范围大 ——一次关联的解个体较多,可能错失全局最优。

- 禁忌长度的确定
 - ○禁忌长度是被禁对象不允许选取的迭代次数。
 - \bigcirc 一般是给被禁对象x一个数(禁忌长度)t,tabu(x)=t,每迭代一步,该项指标做运算tabu(x)=t-1,直到tabu(x)=0时解禁。
 - ○*t*为常数,如 t=10,n,n^{1/2}
 - ○t在一个区间内变化。一个禁忌对象变化越大,被禁的长度越大(可能跳出局部最优)。
 - ○t所在区间的上下界也动态变化,如某些解重复出现,则下界增加。

●特赦规则(aspiration criteria)

$$D = (d_{ij}) = \begin{bmatrix} 0 & 2 & 10 & 10 & 1 \\ 1 & 0 & 2 & 10 & 10 \\ 10 & 1 & 0 & 2 & 10 \\ 10 & 10 & 1 & 0 & 2 \\ 2 & 10 & 10 & 1 & 0 \end{bmatrix}$$

受禁的对象: 2-opt的城市位置顺序对换

目前的一个解为(AEDBC), f(AEDBC)=24

被禁的对换包括BC。

若交换BC,目标值5。这个目标值好于前面的任何一个最佳候选解,

解禁BC对换。这就是一种特赦规则。

特赦规则

- 基于评价值的规则
 - ○可以得到一个更好的解
- 基于最小错误的规则
 - 候选集中所有的对象都被禁忌,为了得到更好的解,从候 选集的所有元素中选一个评价值最小的状态解禁。
- 基于影响力的规则。
 - ○选择对目标值的影响大的变化对象解禁。
 - ○背包问题: 当需要特赦时,占包最大的物品可以特赦出包。

- 候选集合的确定
 - ○由邻居组成,全部或部分邻居,可以随机选取。
- 呼价函数
- 直接评价函数:主要包含以目标函数的运 算而得到评价值的方法。
- ●间接评价函数:尽量反映目标函数的特性。

间接评价函数举例:约束生产计划批量问题 CLPS(capacited lot-size planning and scheduling)

$$\min \sum_{i=1}^{n} \sum_{t=1}^{I} (p_i x_{it} + s_i y_{it} + h_i I_{it})$$

$$s.t I_{it-1} + x_{it} - I_{it} = d_{it}, i = 1, 2, \dots, n; t = 1, 2, \dots, T,$$

$$\sum_{i=1}^{n} a_i x_{it} \le c_t, \ t = 1, 2, \dots, T,$$

$$y_{it} = \begin{cases} 1, \ \exists \ x_{it} > 0, \\ 0, \ \ \ \ \ \ \ \ \ \ \end{cases}$$

$$x_{it}, I_{it} \ge 0, i = 1, 2, \dots, n; t = 1, 2, \dots, T.$$

原问题等价转换成可利用禁忌搜索的组合优化问题:

$$f(Y) = \min \sum_{i=1}^{n} \sum_{t=1}^{T} (p_i x_{it} + s_i y_{it} + h_i I_{it})$$

$$s.t I_{it-1} + x_{it} - I_{it} = d_{it}, i = 1, 2, \dots, n; t = 1, 2, \dots, T,$$

$$\sum_{i=1}^{n} a_{i} x_{it} \leq c_{t}, \ t = 1, 2, \dots, T,$$

$$x_{it} \le My_{it}, i = 1, 2, \dots, n; t = 1, 2, \dots, T$$

$$x_{it}, I_{it} \ge 0, i = 1, 2, \dots, n; t = 1, 2, \dots, T.$$

当没有资源约束时,CLPS模型存在一个最优解满足 $I_{it-1}x_{it}$ =0,由此对应Y \in {0,1}^{nT}唯一一个解

$$x_{i\tau_1} = \begin{cases} \sum_{t=\tau_1}^{\tau_2} d_{it}, y_{i\tau_1} = y_{i\tau_2+1} = 1, y_{it} = 0, \tau_1 < t \le \tau_2, \\ 0, & \sharp \dot{\Xi}. \end{cases}$$

从最后一个时段T到第一个时段反向按能力约束逐个时段验证解是否满足能力约束。当不满足时,将超出资源约束的部分后(前)移一个时段加工。如此修正,若最后(第一个)时段资源不足时,则认为无可行解,此时目标值记为充分大的一个数。若可行,则按修正后的解,用修正后的解计算目标函数值。

如何评价这样的"间接评价函数"?

- 理论:对给定的Y,若无能力约束的解为约束问题的可行解,间接评价函数值与真实评价值相同。
- 真实评价函数*f*(*Y*)是一个线性规划问题,多项式时间可解,但禁忌搜索算法每次调用求最优解算法的累计计算时间较大。
- 间接评价函数节省计算时间。
- 评价这样的想法:以实际应用问题的数值 计算效率评价。

记忆频率信息

- 解集合、被禁有序对象组、目标值集合等的出现频率。 这些信息有助于监控算法的计算效能,指导算法的改 进。
 - 静态的频率信息:如解、对换或目标值在计算中出现的 频率。记录它们在计算中出现的次数,出现的次数与总 的迭代数的比率,从一个状态出发再回到该状态的迭代 次数等。
 - 动态的频率信息:如从一个解、对换或目标值到另一个解、对换或目标值的变化趋势。通常需要记录的信息有:
 - 一个序列的长度,即序列中元素个数;
 - 从序列中的一个元素出发,再回到该序列该元素的迭代次数;
 - 一个序列的平均目标(评价)值,从序列中一个元素到另一个元素目标(评价)值的变化情况;
 - 该序列出现的频率。

终止规则

- ○确定步数终止。给定一个充分大的数K,总的迭代次数不超过K步。
- ○频率控制原则。当某一个解、目标值或元素序列的频率超过一个给定的标准时,如果算法不做改进,只会造成频率的增加,此时的继续计算无法不会得到更好的解,因此,终止计算。
- ○目标控制原则。在禁忌搜索算法中,提倡记忆当前最优解。 如果在一个给定的步数内,当前最优值没变,同第二规则 相同的观点,停止运算。
- ○目标值偏离程度原则。对一些问题可以简单地计算出它们的下界(目标为极小),对给定的误差ε,当计算值同下界之差在误差范围内时,终止计算。

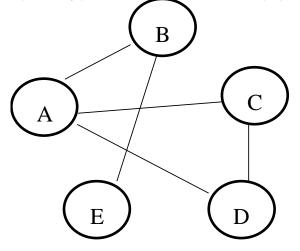
2.4 应用实例——图节点着色和车间作业排序

● 图节点着色问题

给 定 一 个 无 向 图 $G=(V,\ E)$, 其 中 V 是 节 点 集 $V=\{1,2,...,n\}$, E 是 边 集 $E=\{(i,j)|i,j\in V\}\ ,\ (i,j)$ 表示有连接 i,j 的一个边。若 $V_i\subset V$ 、 $V=\bigcup_{i=1}^k V_i$ 且 V_i 内

部的任何两个节点没有 E 中的边直接相连,则称 (V_1,V_2,\cdots,V_k) 为 V 的一个划分。图

的节点着色问题可以描述为: 求一个最小的 k, 使得 (V_1,V_2,\dots,V_k) 为 V 的一个划分。



它的一个划分是:

$$V_1 = \{A, E\}, V_2 = \{B, C\}, V_3 = \{D\}_{\circ}$$

$$V_1 = \{A\}, V_2 = \{B\}, V_3 = \{C\}, V_4 = \{D\}, V_5 = \{E\}$$

也是一个划分。

求解图节点覆盖问题的两步

•给定一个常数k,考虑目标函数:

$$f(V_1, V_2, \dots, V_k) = |E(V_1)| + |E(V_2)| + \dots + |E(V_k)|$$

 $|E(V_i)|$ 为 V_i 中直接连接两个节点的边数。 (V_1,V_2,\cdots,V_k) 为

V 的一个划分的充分必要条件是 $f(V_1, V_2, \dots, V_k) = 0$ 。

第二步的主要工作是:对不满足 $f(V_1,V_2,\dots,V_k)=0$ 的 (V_1,V_2,\dots,V_k) ,增

加子集的个数 k; 从满足 $f(V_1, V_2, \dots, V_k) = 0$ 的划分中选择最小的划分数 k。

技术问题

解的形式

$$S = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}, \quad 1 \le i_j \le k, \quad 1 \le j \le n$$

解的分量变化

$$\begin{pmatrix} x \\ i_x \end{pmatrix} \rightarrow \begin{pmatrix} x \\ i_x \end{pmatrix}$$

禁忌

$$\begin{pmatrix} x \\ i_x \end{pmatrix} \leftarrow \begin{pmatrix} x \\ i_x \end{pmatrix}$$

特赦规则: 若x*是当前最优解,当一个受禁的邻居x满足 $f(x) \le f(x*)-1$ 时,则受禁的变化特赦。

Hertz 和 Werra 程序的框架如下

己知:

划分集合数 k,解的形式 S,目标函数 f,解的变化,邻域映射 N(S),禁忌长度,目标值下界 f^* ,两个目标值没有改进的最大允许迭代次数 nbmax。

开始

- ——任选一个初始解;
- ——nbiter:=0; (*当前的迭代步*)
- ——bestiter:=0; (*当前最优解所在的迭代步*)
- ——bestsol:=**S**; (*当前的最优解*)
- ——令 z=f(S), A(z):=z-1; (特赦值)
- ——初始化禁忌表 H;
- 当 (f(S)>f*) 且 (nbiter-bestiter<nbmax) 时, 计算
 - ---nbiter:=nbiter+1;
 - ——产生 N(S)的一个候选集 V^* ,要求候选元素 x 是非禁忌的或是特赦的 f(x)≤A(f(S));
 - 一在 V*中选目标值最优的解 S*;
 - 一若 f(S*) ≤A(f(S)), 则 A(f(S*)):=f(S*)-1; 否则若 f(S)≤A(f(S*)), 则 A(f(S*)):=f(S)-1;
 - 一更新禁忌表;
 - 一若 f(S*)<f(bestsol),则 bestsol:=S*; bestiter:=nbiter; (清空,从头开始)
 - -S:=S*

继续;

结束。

输出: 计算中的最优解。

计算结果

随机产生实例的参数为:无向图共有1000个节点,边集的密度是50%(大约是250000个边),禁忌表长度|H|=7,候选解个500数。最终对上面规模实例的计算结果是平均可用87种颜色划分。概率分析的理论结果是85中颜色。

Hertz A, de Werra D. The tabu search metaheuristic: how we use it. Annals of Mathematics and Artificial Intelligence, 1990, 1:111-121.

车间作业调度问题 (Job Shop Scheduling)

n个工件(job)(这是一个统称), J_1, J_2, \cdots, J_n 在m台机器(machine)(也

是统称) M_1, M_2, \cdots, M_m 上加工。每一个工件 J_i 有 n_i 个工序

(operation) $O_{i1}, O_{i2}, \cdots, O_{in_i}$,第 O_{ij} 工序的加工时间为 p_{ij} 。必须按工序顺

序进行加工且每一工序必须一次加工完成(无抢占, no preemption)。一台机器在任何时刻最多只能加工一个产品,一个工件不能同时在两台机器上加工。如何在上面的条件下使最后一个完工的工件加工时间最短?

开放车间调度

- m 机器, n 工件, 加工完所有工件最小费用
- Open shop scheduling: 高考阅卷点阅卷。
- 每题工件,阅题小组机器,不受题号顺序 限制。
- ●每组安排产品的加工顺序?

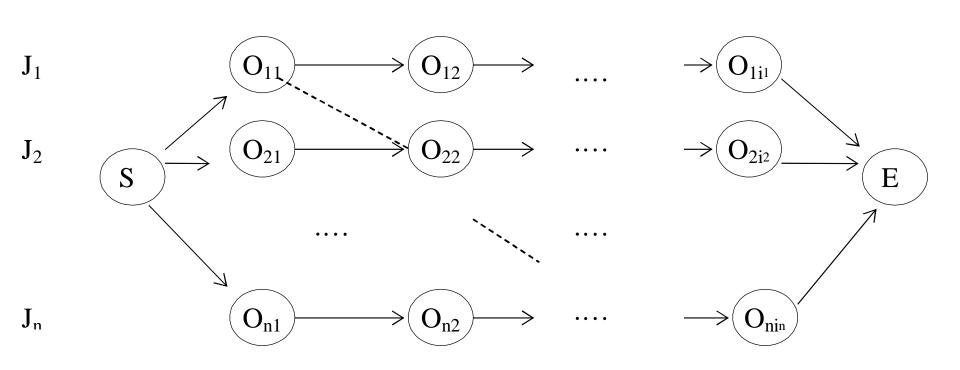
流水车间调度

- Flow shop scheduling: 一个集团公司飞机生产。
- 工件:不同机型(出口、海军、民用)。
- 机器:各生产车间(钳工、钣金、组装、试飞等)。
- 零件生产、部分组装(机头、机身、机翼),整体组装(发动机、设备等),试飞。每个产品有相同的工序,每道工序在限定机器加工,必须按工序加工。
- 如何安排产品的生产顺序?

(工件) 车间作业调度

- Job shop scheduling: 一个研究所本科生、硕士生、博士生、博士后培养。
- 工作(件):本科生、硕士生、博士生、博士后培养。
- 工序。本科生:课程学习->毕业设计;研究生:课程学习->毕业论文;博士生:课程学习->论文发表->毕业论文。
- 机器 (教师) 分工合作: 课程、硕导、博导。
- 必须按工序顺序加工,但不同产品的工序可能不同,每个工序只能在限定机器加工。
- 关键: 每台机器上的工件加工顺序。

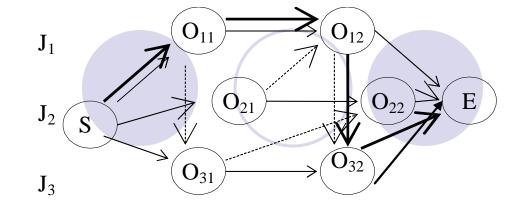
车间作业调度:图论模型(实线:工序,虚线对应同一台机器加工顺序)



常用结论

- 车间作业调度问题在两个工件时是存在多项式时间的最优算法。
- 当工件数超过2时,问题的复杂性是NP难。
- 非常直观,问题的一个解对应*m*台机器上的一个加工排序。
- 邻域构造: 选一台机器上的两个工序交换加工顺序。
 - ○如一台机器上原有的加工序为(a,b,c,d,e),交换ac序后,新的加工序为(c,b,a,d,e)。这样的邻域映射使得每个解有个邻居。
 - ○可以研究若干个位置交换的情形。一种特殊的情况是将一个工序移 到另一个位置加工。如在一台机器上原有的加工序是(abcdef),现将f 移到第一个位置加工,加工序则为(fabcde)。
- 关键路(critical path)法。
 - ○基本思想是:在上面的方法中,一些交换对目标值没有影响,于是 在计算中,这些交换浪费计算时间,应抓住最长的、加工中没有时 间空闲的一条路(即关键路),交换同在这条路上且同在一台机器 上加工的两个加工工件的位置。

关键路的理论



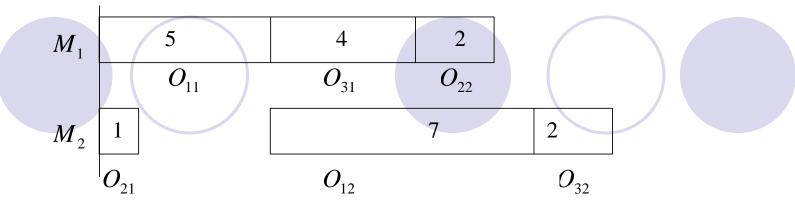
 O_{32}

 O_{11}, O_{22}, O_{31} 在一台机器上加工, O_{12}, O_{21}, O_{32} 在另一台机器上加工。假设各

工序的加工时间为: $p_{11} = 5, p_{22} = 2, p_{31} = 4$, $p_{12} = 7, p_{21} = 1, p_{32} = 2$ 。

 O_{12}

关键路是: $O_{11} \rightarrow O_{12} \rightarrow O_{32}$ 长度为14



交换 O_{31} , O_{22} 的加工位置对最长完工时间没有影响,但交换关键路上的一个工件,如交换 O_{12} , O_{32}

$$O_{11} \to O_{31} \to O_{32} \to O_{12}$$

长度为18

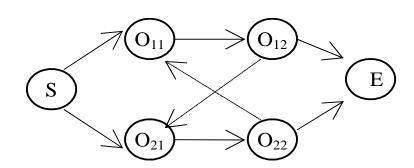
- 定义 在关键路上,满足下列条件的相邻节点集称为块(block):
 - 由关键路上的相邻节点组成,至少包含两个工序;
 - 集合中的所有工序在一台机器上加工;
 - ○增加一个工序后,不满足(1)或(2)。
- **定理** 若一解的关键路不包含块,则一定是最优解。
 - ○证明:关键路上不包含块,同一台机器的相邻工序由虚线相连,由此可知,任意相邻的两个工序由一条实线连接。这条关键路长度是某一个工件的所有工序加工时间和,是车间作业完工时间的一个下界。因此定理结论成立。
- **定理** 若y和y'是车间作业调度的两个可行解,且y和y'对应的有向图为S和S'。若y'改进y,则一定满足下面两条件之一:
 - ○至少有一个工序,它在y的一个块B中且不是B块中的第一个工序,但在y'中它在B的其他工序之前加工;
 - ○至少有一个工序,它在y的一个块B中且不是B块中的最后一个工序,但在y'中它在B的其他工序之后加工。

邻域结构

- 邻域N1:设y'为一个可行解,若y'将y的关键路中一个块中的一个作业前移到最前或后移到最后位置加工,则称y'为y的一个邻居。所有这样的移动组成的集合为y的邻域。
- 邻域N2:设y'为一个可行解,若y'是y的关键路中一个块中的作业的前移或后移的所有可能位置加工,则称y'为y的一个邻居。所有这样的移动组成的集合为y的邻域。
- 注: 在上面的邻域定义中,我们强调y'为一个可行解

$$M_1: O_{11} \to O_{22} \qquad M_2: O_{12} \to O_{21}$$

 $M_1(new): O_{22} \to O_{11}$



具体实现

定理 车间作业调度的可行解集合相对N2是连通的, 即以任何一个可行解为起点,可以通过N2达到一个 全局最优解。

多功能机器车间作业排序问题。在他们的模型中,工序 O_{ij} 可由一个机器

集合 $G_{ij} \subset \{M_1, M_2, \cdots, M_m\}$ 中的任何一个机器加工。极端的情况

 $|G_{ij}|=1$ 为常规的车间作业排序问题。他们产生 53 组随机数据进行数值

计算。数据生成的原则是:以已有典型数据为基础,以多功能系数分组随机产生数据文件,所产生的数据分别用 edata、rdata 和 vdata 标记。典型数据的主要参数是:机器数分别为 5、6、10 和 15,工件数分别为 6、10、

15、20 和 30,且每一个工件的工序数都相同,满足 $n_i = m, i = 1, 2, \dots, n$ 。

计算结果

- 多功能机器车间作业排序问题。在他们的模型中,工序可由
 - 一个机器集合中的任何一个机器加工。极端的情况(只能在
 - 一台机器上加工)为常规的车间作业排序问题。

数据分别用 edata、rdata 和 vdata 标记。机器数分别为 5、6、10 和 15,工件数分别为 6、10、15、20 和 30,且每一个工件的工序数都相同,满足 $n_i = m, i = 1, 2, \dots, n$ 。

夕计	能参数
タリ	化纱奴

	$ G_{ij} $ ave	$ G_{ij} $ max		
Edata	1.15	$2(m \le 6) \ 3(m \ge 10)$		
Rdata	2	3		
Vdata	$\frac{1}{2}m$	$\frac{4}{3}m$		

其中, $|G_{ij}|$ ave 表示所有 $|G_{ij}|$ 的平均数, $|G_{ij}|$ max 表示所有 $|G_{ij}|$ 的最大数。

参考文献: Hurink J, Jurisch B, Thole M. Tabu search for the job-shop scheduling problem with multi-purpose machines. OR Spektrum, 1994, 15:205~215

N1 和 N2 为邻域结构,禁忌对象选择为:禁忌上一步位置变化的复原,如一个块为(abcde),c原在的位置是 3,经过位置移动后,禁忌回到 3。禁忌表的长度为 30,总的迭代长度分为 1000 和 5000 次两种。停止规则为:在迭代总长达到之前,若邻域中的所有可行解都被禁忌、目前解的目标值等于下界或出现循环三种情况之一出现则停止计算。

功能类型	相对误差	N1-1000	N2-1000	N1-5000	N2-5000
edata	平均	5.2	5.3	4.8	4.5
	最大	24.0	22.8	23.4	19.8
rdata	平均	2.8	2.9	2.3	2.3
	最大	13.4	14.5	12.0	10.7
vdata	平均	0.5	0.5	0.4	0.4
	最大	3.2	3.1	1.9	2.1

邻域N1和N2对计算结果的影响不明显,总体的平均效果,禁忌搜索算法相当好,所得目标值同已知最好下界(另一篇文章的已知结果)的相对差的平均效果在0.4%到5.2%之间;随着机器功能的增加,相对误差逐渐变小;迭代次数1000和5000的差别不是很大,也就是在迭代总次数限制在1000次时,计算效果同迭代5000次基本相同。因此可以采用1000次的迭代而节省计算时间。