第10章 近似算法

- 近似算法及其近似比
- 多机调度问题
  贪心的近似算法
  改进的贪心近似算法
- 货郎问题
  最邻近法
  最小生成树法
  最小权匹配法
- 背包问题
  一个简单的贪心算法
  多项式时间近似方案
  伪多项式时间算法与完全多项式时间近似方案

近似算法及其近似比

近似算法: A是一个多项式时间算法且对组合优化问题 Π 的每一个实例 I 输出一个可行解 σ. 记 $A(I)=c(\sigma)$, $c(\sigma)$ 是 σ 的值

近似算法的性能估计

当 Π 是最小化问题时, 记 $r_A(I)=A(I)/OPT(I)$

当 Π 是最大化问题时, 记 $r_A(I)=OPT(I)/A(I)$

最优化算法: 恒有 $A(I)=OPT(I)$, 即 $r_A(I)=1$.

A的近似比为 r (A是r–近似算法): 对每一个实例 I, $r_A(I) \le r$.

A具有常数近似比: r是一个常数.

可近似性分类

假设P≠NP，NP难的组合优化问题按可近似性可分成 3 类:

(1) 完全可近似的: 对任意小的$\varepsilon>0$, 存在$(1+\varepsilon)$-近似算法

背包问题

(2) 可近似的: 存在具有常数比的近似算法

最小顶点覆盖问题、多机调度问题

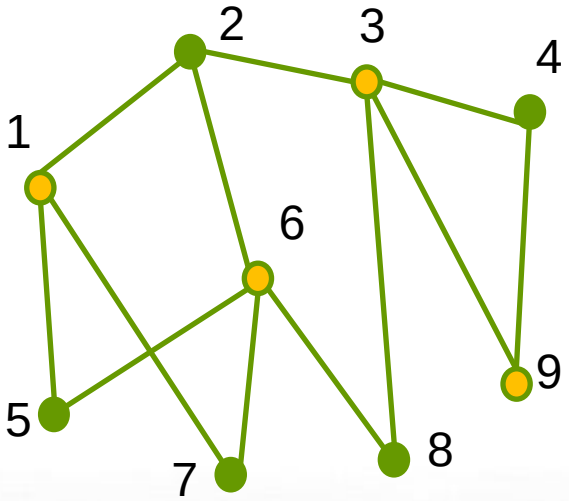
(3) 不可近似的: 不存在具有常数比的近似算法

货郎问题

最小顶点覆盖问题

问题: 任给图$G=<V, E>$, 求G 的顶点数最少的顶点覆盖.

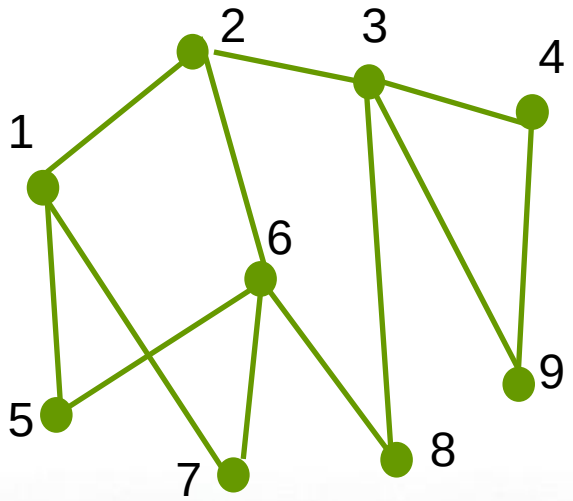
算法MVC: 开始时令$V'=\varnothing$. 任取一条边(u,v), 把u和v加入V'并删去u和v及其关联的边. 重复上述过程, 直至删去所有的边为止. V'为所求的顶点覆盖.



{1,2}

{1,2,3,4}

{1,2,3,4,5,6}



一个最优解：{1,3,6,9}， MVC的解：{1,2,3,4,5,6}

MVC算法分析

时间复杂度　$O(m)$, $m=|E|$.

近似比

(1) 记 $|V'| = 2k$, V'由 k 条互不关联的边的端点组成. 为了覆盖这 k 条边需要 k 个顶点, 从而 $\mathrm{OPT}(I) \geq k$. 于是

$$\mathrm{MVC}(I)/\mathrm{OPT}(I) \leq 2k/k = 2$$

(2) 设图G由 k 条互不关联的边构成, 显然

$$\mathrm{MVC}(I) = 2k, \quad \mathrm{OPT}(I) = k,$$

这表明 MVC 的近似比不会小于2, 上面估计的MVC的近似比已不可能再进一步改进.

结论：MVC是2-近似算法.

近似算法的分析

近似算法的分析：算法的运行时间与近似比.

分析近似比

(1) 估计最优解的值(建立最优值与近似解的值之间的关系）

(2) 构造使算法产生最坏的解的实例. 如果这个解的值与最优值的比达到或者可以任意的接近得到的近似比(这样的实例称作紧实例), 那么说明这个近似比已经是最好的、不可改进的了; 否则说明还有进一步的研究余地.

研究问题本身的可近似性

即在P≠NP(或其他更强)的假设下, 该问题近似算法的近似比的下界.

多机调度问题

多机调度问题:

任给有穷的作业集 A 和 m 台相同的机器, 作业 a 的处理时间为正整数 $t(a)$, 每一项作业可以在任一台机器上处理. 如何把作业分配给机器才能使完成所有作业的时间最短? 即, 如何把 A 划分成 m 个不相交的子集 A_i 使得

$$\max\left\{\sum_{a\in A_i} t(a) \mid i=1,2,\cdots,m\right\}$$

最小?

贪心法

- 负载: 分配给一台机器的作业的处理时间之和.
- 贪心法 G-MPS

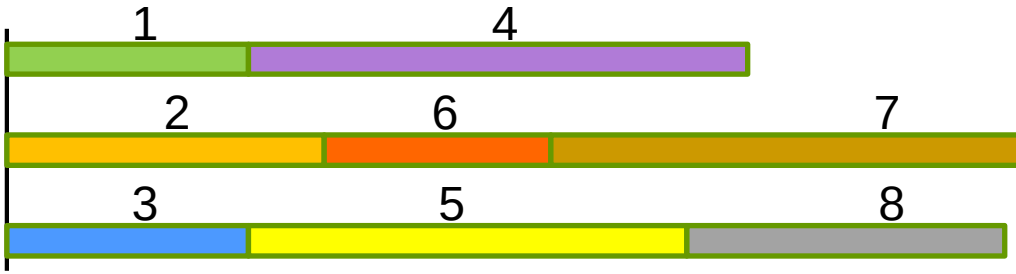
  按输入的顺序分配作业

  把每一项作业分配给当前负载最小的机器

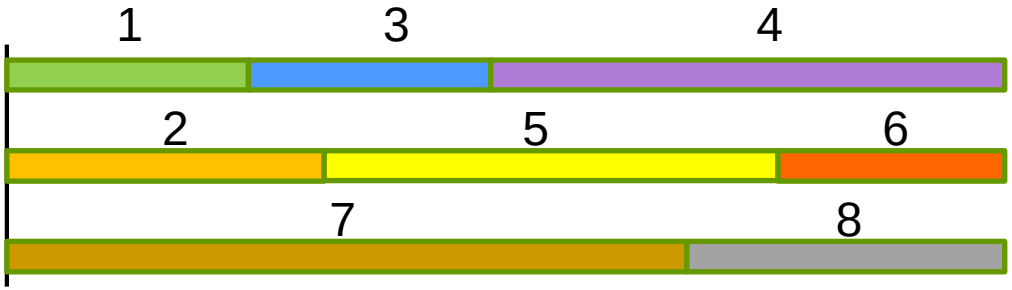
  如果当前负载最小的机器有 2 台或 2 台以上，则分配给其中的任意一台（比如标号最小的一台）

实例

例如， 3 台机器， 8 项作业，处理时间为 3, 4, 3, 6, 5, 3, 8, 4



G-MPS的解
完成时间 15



最优解
完成时间 12

算法给出的分配方案是 {1,4}, {2,6,7}, {3,5,8},
负载分别为 3+6=9, 4+3+8=15, 3+5+4=12
最优的分配方案是 {1,3,4}, {2,5,6}, {7,8},
负载分别为 3+3+6=12, 4+5+3=12, 8+4=12

贪心法的性能

定理 对多机调度问题的每一个有 m 台机器的实例 I，

$$G-MPS(I)\le\left(2-\frac{1}{m}\right)OPT(I)$$

证 两个事实：

(1) $$OPT(I)\ge\frac{1}{m}\sum_{a\in A}t(a)$$

m个机器的最大负载不小于平均负载

(2) $$OPT(I)\ge\max_{a\in A}t(a)$$

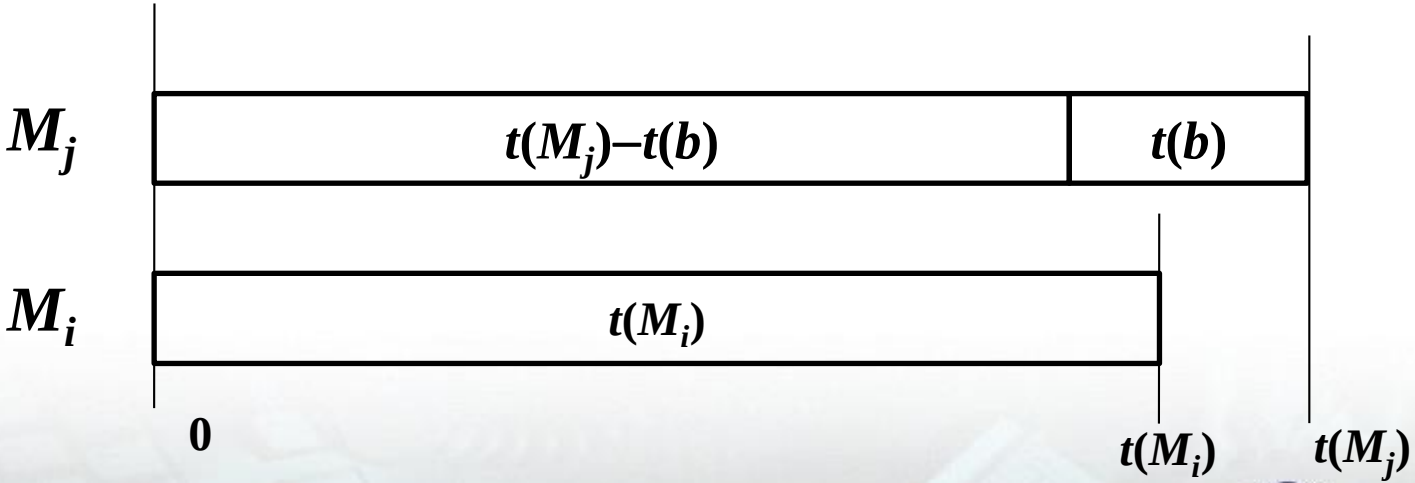
m个机器的最大负载不小于单个任务负载

证 明

设机器 M_j 的负载最大, 记作 $t(M_j)$. 又设 b 是最后被分配给机器 M_j 的作业. 根据算法, 在考虑分配 b 时 M_j 的负载最小, 故

$$t(M_j)-t(b)\le\frac{1}{m}\left(\sum_{a\in A}t(a)-t(b)\right)$$

$$t(M_j)\le\frac{1}{m}\left(\sum_{a\in A}t(a)-t(b)\right)+t(b)$$



证 明

于是

$$G-MPS(I)=t(M_j)$$

$$\le \frac{1}{m}\left(\sum_{a\in A} t(a)-t(b)\right)+t(b)$$

$$=\frac{1}{m}\sum_{a\in A} t(a)+\left(1-\frac{1}{m}\right)t(b)$$

$$\le OPT(I)+\left(1-\frac{1}{m}\right)OPT(I) \quad \text{事实(1)和(2)}$$

$$=\left(2-\frac{1}{m}\right)OPT(I).$$

紧实例

m台机器， $m(m-1)+1$ 项作业，

前 $m(m-1)$ 项作业的处理时间都为 1，最后一项作业的处理时间为m.

算法把前 $m(m-1)$项 作业平均地分配给 m 台机器, 每台 $m-1$ 项，最后一项任意地分配给一台机器.

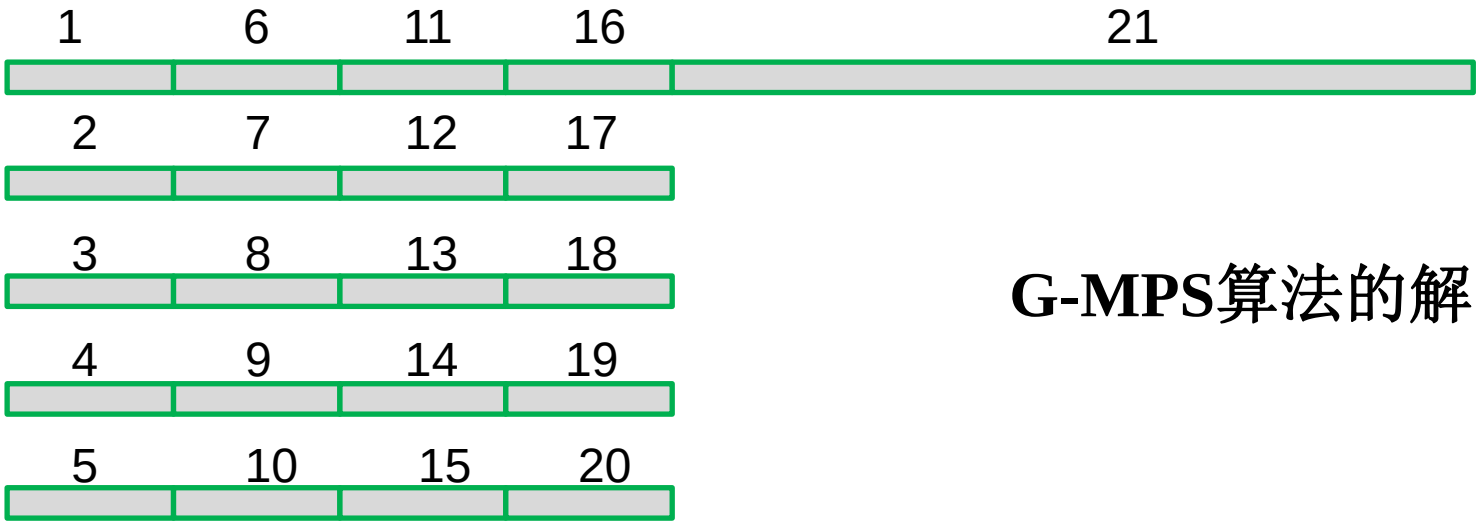
$$\text{G-MPS}(I) = 2m-1.$$

最优分配方案是把前 $m(m-1)$ 项作业平均地分配给 $m-1$ 台机器, 每台 m 项, 最后一项分配给留下的机器,

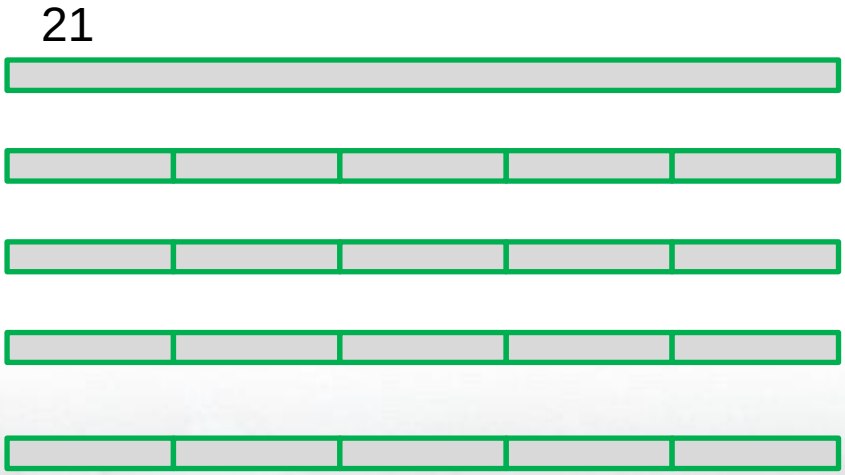
$$\text{OPT}(I) = m.$$

G-MPS是2-近似算法

m=5的紧实例



G-MPS算法的解



最优解

改进的贪心近似算法

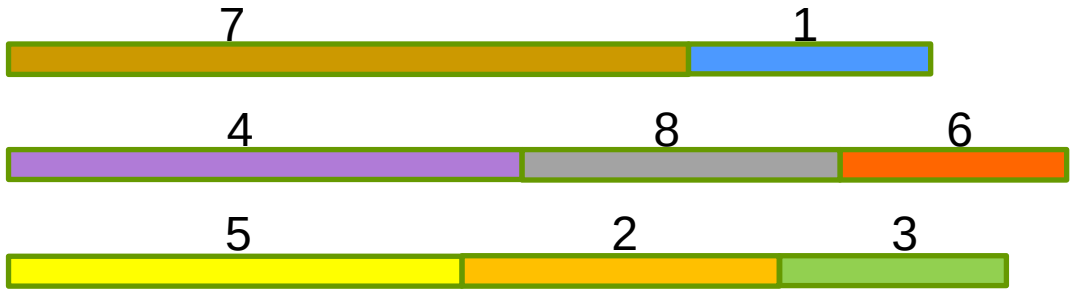
递降贪心法DG-MPS：首先按处理时间从大到小重新排列作业, 然后运用G-MPS.

例如对上述紧实例得到最优解.

对另一个实例：先重新排序 8, 6, 5, 4, 4, 3, 3, 3;

3台机器的负载分别为 8+3=11, 6+4+3=13, 5+4+3=12.

比G-MPS的结果好.



DG-MPS的解
完成时间13

分析：DG-MPS增加排序时间$O(n\log n)$, 仍然是多项式时间.

近似比

定理 对多机调度问题的每一个有 m 台机器的实例 I,

$$\text{DG-MPS}(I) \le \left(\frac{3}{2}-\frac{1}{2m}\right)\text{OPT}(I)$$

证 设作业按处理时间从大到小排列为 $a_1, a_2, \ldots, a_n$, 仍考虑负载最大的机器 M_j 和最后分配给 M_j 的作业 a_i.

(1) M_j 只有一个作业, 则 $i = 1$, 必为最优解.

(2) M_j 有 2 个或 2个以上作业, 则 $i \ge m+1$ ⇨ $n \ge m+1$

$$\text{OPT}(I) = \max\left\{\sum_{a\in A_i} t(a) \mid i = 1,2,\cdots,m\right\} \ge t(a_m)+t(a_{m+1}) \ge 2\, t(a_i)$$

证明

$$\mathrm{DG-MPS}(I)=t(M_j)$$

$$\le \frac{1}{m}\left(\sum_{k=1}^{n} t(a_k)-t(a_i)\right)+t(a_i)$$

$$=\frac{1}{m}\sum_{k=1}^{n} t(a_k)+\left(1-\frac{1}{m}\right)t(a_i)$$

$$\le \mathrm{OPT}(I)+\left(1-\frac{1}{m}\right)\cdot\frac{1}{2}\mathrm{OPT}(I)$$

$$=\left(\frac{3}{2}-\frac{1}{2m}\right)\mathrm{OPT}(I)$$

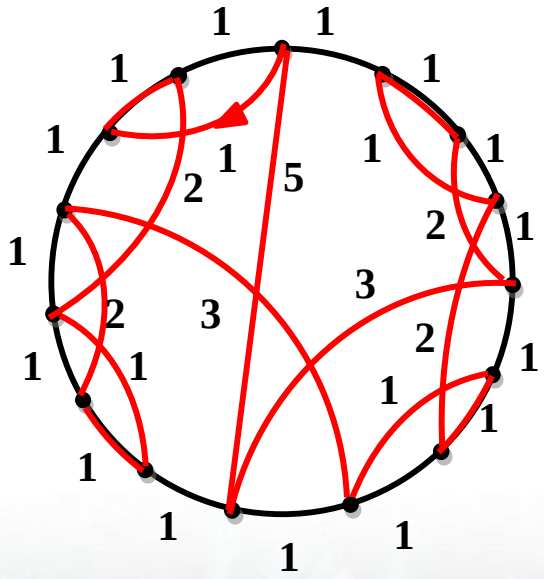
货郎问题

本节考虑满足三角不等式的货郎问题

最邻近法NN：从任意一个城市开始，在每一步取离当前所在城市最近的尚未到过的城市作为下一个城市．若这样的城市不止一个， 则任取其中的一个．直至走遍所有城市, 最后回到开始出发的城市.

一个NN性能很坏的实例 I,

$NN(I)=27$，$OPT(I)=15$



最邻近法的性能

定理 对于货郎问题所有满足三角不等式的 n 个城市的实例 I, 总有

$$\mathrm{NN}(I) \le \frac{1}{2}\left(\lceil \log_2 n \rceil + 1\right)\mathrm{OPT}(I).$$

而且, 对于每一个充分大的 n, 存在满足三角不等式的 n 个城市的实例 I 使得

$$\mathrm{NN}(I) > \frac{1}{3}\left(\log_2(n+1) + \frac{4}{3}\right)\mathrm{OPT}(I).$$

结论：NN算法不是常数近似比的算法，不能实际应用.

问题：对于货郎问题（或者满足三角不等式的子问题）是否存在常数近似比的算法？