基于动态规划方法的期末复习提分设计

黄泽熙　　王泊锟

（电子科技大学　英才实验学院　611731）

摘　要：本文在动态规划基本理论的基础上，结合期末考试前复习的实际背景，将复习时间划分为：未结课、已结课两个部分，分别进行方案设计，从而获得个人最优复习方案和最佳提分表现.

关键词：　动态规划　　考前提分　复习方案设计　学习效率

A Design for Score Promotion in Final-exam Review Period Based on DP Theory

Huang Zexi　Wang Bokun

（Yingcai Experimental School　University of Electronic Science and Technology of China 611731）

Abstract: Based on the fundamental theory of dynamic programming (DP) and taking the actual background of the review period before final-exam into account, this article divides the time left for reviewing into two parts, namely, the phase before and the phase after the lecture period in a term, and designs solutions for each phase respectively, and thus works out the personal optimal reviewing approach and the best score promotion.

Key words: dynamic programming　score promotion before a test　review approach design　studying efficiency

1. 预备知识

动态规划是制定一系列相关决策时的一项有用的数学技术，它提供系统化的方法，即对于每个问题建立一个合适的递推关系公式，来寻求最优的决策组合.它利用详细统计找到决策的最优组合，在计算的效率上远超其他方法，例如与穷举法相比，它需要考虑的组合数会少若干个数量级.

能够利用动态规划解决的最优化问题应该具有以下几个基本特征：

1. 问题可分为几个阶段，每个阶段都有一个对应的策略决策.
2. 每个阶段都有一些与该阶段开始有关的状态.
3. 每个阶段的策略觉得结果都是从当前的状态变成下一阶段开始的状态.
4. 设计求解过程是为整个问题找到一个最优策略.
5. 已知目前阶段的状态，对于剩余阶段的最优策略与先前阶段所采用的策略无关.这称为动态规划的最优原理.
6. 求解过程从为最后阶段找到最优策略开始.
7. 如果知道$n+1$阶段的最优策略，就可以确定第n阶段的最优策略，因而得到递推关系.
8. 利用递推关系，求解过程从终点开始一步步逆序移动，每次都找到对应阶段的最优策略，知道找到最开始阶段的最优策略，这个最优策略就是整个问题的最优解决方案.

2. 符号的说明

N：阶段的数量.

n：当前阶段的标号$\left(n=1,2,\cdots,N\right)$.

s_n：第n阶段开始时的状态（初始状态）.

x_n：第n阶段的决策变量.

$p_n(s_n,x_n)$：第n阶段目标函数的效益值.

$f_n(s_n,x_n)$：从第n阶段到最后一阶段，目标函数的效益值.

x_n^*：给定s_n后，使目标函数$f_n(s_n,x_n)$最优的决策变量x_n.

$f_n^*(s_n)$：从第n阶段到最后一阶段，目标函数的效益值的最优值，即$f_n^*(s_n)=f_n(s_n,x_n^*)$.

$\eta_n(s_n,x_n)$：第n阶段的学习效率.

c_i：第i门课程的学分数.

e_i：第i门课程的分数.

B：学生的加权平均分.

M：学生的课程总数.

$g_i(t)$：在效率为 1 的情况下，复习第i门课程t天加权分数的提高值.

$G_i(t)$：在某效率下，复习第i门课程t天加权分数的提高值.

E_n：第n阶段的休息程度.

d_n：第n阶段的总天数.

N_n：第n阶段的休息天数.

F_n：第n阶段的学习状态.

3. 问题的提出

每当临近期末考试，如何合理分配自己的时间使得取得最好复习效果（获得最大的分数提升）便成为每位大学生都头疼的问题.这里以电子科技大学英才实验学院大学一年级 2014-2015 学年 2 学期期末考试复习计划为例进行讨论.

3.1. 复习时间的划分

在本次案例中，学生所需要进行期末考试的课程、考试时间对应的学分如表 1 所示.

	课程名				
	英语文献阅读	数学分析II（含常微分方程）	大学物理I	电路分析基础	离散数学
考试时间	7/13/2015	7/15/2015	7/8/2015	7/10/2015	7/2/2015
学分数	4	7	5.5	5.5	3

表 1 各科考试时间和学分表

在本学期中，考虑到各科的结课时间，可供学生进行期末复习的可分为两部分：结课前部分与结课后部分，分别对应的时间段为：6 月 22 日-6 月 28 日（结课前、共 7 天）、6 月 29 日-7 月 15 日（结课后、共 16 天）.在结课前的时间内，因为课程尚未结束，而且离期末考试还有一段时间，采取分科复习的方式（即在该时间段内，学生可任意选取课程复习），而在结课后的时间内，因为期末考试临近，采取考前突击复习的方式（即在该段时间内，学生只复习考试时间最近的课程）.因为部分课程的考试时间临近，为方便处理，将英语文献阅读与数学分析 II（含常微分方程）合为一个 11 个学分的课程，记为课程 1，并把课程 1 的考试时间视为 7 月 15 日；将大学物理 I 和电路分析基础合为一个 11 个学分的课程，记为课程 2，并把课程 2 的考试时间视为 7 月 10 日.离散数学记为课程 3，并把课程 2 的考试时间视为 7 月 2 日.

3.2. 提分的期望

学生各课程分数的提高的期望值与复习时间的关系，由学生本身对课程的掌握程度，课程对该学生的难易程度等诸多主观因素决定，因此，这里不便给出一个适用于所有学生的关系式，仅以某位学生为例给出关系，如表 2.对于其余同学而言，可以根据自己的实际情况给出对应的关系.

学习天数	课程分数提高值		
	课程1	课程2	课程3
1	2	1.5	6
2	2.5	3	10
3	4.5	5	13
4	5	6.5	15
5	6	7.5	15
6	7.5	8.5	15
7	10	9	15
8	13	9.5	15
9	14	10	15
10	15	10	15

表 2 分数提高值与学习天数关系

学习天数	课程分数加权提高值		
	课程1	课程2	课程3
1	0.88	0.66	0.72
2	1.1	1.32	1.2
3	1.98	2.2	1.56
4	2.2	2.86	1.8
5	2.64	3.3	1.8
6	3.3	3.74	1.8
7	4.4	3.96	1.8
8	5.72	4.18	1.8
9	6.16	4.4	1.8
10	6.6	4.4	1.8

表 3 加权平均后分数提高值与学习天数的关系

现在大多数大学采取加权平均分的形式来衡量学生的学习成果，电子科技大学也是如此.若记课程i学生获得的分数为e_i，学分数为c_i，学生的课程数为M，那么学生的加权平均分B可以记为

$$B=\frac{\sum_{i=1}^{M}c_ie_i}{\sum_{i=1}^{M}c_i} \tag{3.2.1}$$

根据式(3.2.1)，我们可以将分数提高值与学习天数的关系表转换为表 3 的形式.在以下讨论时，我们说的分数提高值均是只加权后的分数提高值.在结课前和结课后两个部分的时间内，提分值满足简单相加的关系，且考虑到两段复习时间有本质差异，假设两段时间内复习的提分值没有关系.

3.3. 效率的考量

学习效率，是在社会正常的教学条件下、学生正常的心理感受下，学习过程中单位时间内接受学习任务，处理学习问题，实现学习目标的程度.学习效率评价则是对学习效率的高低程度的测算和比较过程，是对学习效率的高低程度所做的肯定和否定判断.根据学习效率的测算和衡量所依据的认知理论，学习效率问题的研究尽管涉及到的因素很多，考虑到问题的实际情况只考虑如下两个方面：学习状态、学习时间.

在本问题中，学生的真实提分值不仅受表 3 的约束，而且还会受到效率的影响.若设表 3 中确定的复习第i门课程t天加权分数的提高值的关系式为$g_i(t)$，学生的效率为η，那么学生的真实提分值$G_i(t)$应满足

$$G_i(t)=\eta g_i(t) \tag{3.3.1}$$

在结课前，学生的作息及思想状态比较规律，可以认为学习效率是常数，取该部分时间的学习效率$\eta=1$.

在结课后，学生如果持续不断地复习将会导致学习效率的下降，为了量化这一过程，我们将引入休息程度的概念.学生在结课后的学习过程共可根据复习课程的不同分为若干个阶段.设在第n阶段中休息的天数为N_n，那么第i阶段的休息程度E_n可以记为

$$E_n = \frac{1}{3}N_{n-1} + \frac{2}{3}N_n \tag{3.3.2}$$

考虑到每个阶段内学习的连续性，我们假设每阶段休息时间 N_n 满足

$$N_n \le \frac{1}{2}d_n \tag{3.3.3}$$

式中 d_n 为该阶段总时间.

在不考虑其他次要因素的情况下，由式(3.3.2)所决定的休息程度将直接决定学生的学习状态 F_n，其关系如表 4.

休息程度 E_n/天数	学习状态感受	学习状态量值
$[0,1)$	一般	1
$[1,2)$	较好	2
$[2,+\infty)$	很好	3

表 4 学习状态与休息程度的关系

第 n 阶段的学习效率则由学习状态 F_n 和各阶段总时间 d_n 确定

$$\eta_n = 0.6 + \frac{F_n}{d_n} \tag{3.3.4}$$

这样，我们就确定了结课前与结课后学生的真实提分值.

4. 方案的设计

4.1. 结课前方案的设计

结课前的时间段是指 6 月 22 日-6 月 28 日，考虑到本段时间的实际情况，用于每天均有课程安排，不可能拿出整天来休息，因此不考虑休息天数.方案设计的目标是：运用动态规划方法，将这 7 天分给课程 1、课程 2、课程 3，从而实现提高分数 f 的最大值.

解决此问题需要做出三个决策：即分配给课程 1、课程 2、课程 3 各自的复习天数.而这三门课程可以视为动态规划的不考虑先后顺序的三个阶段.决策变量 x_n 是指分配给每一科目（每个阶段）的天数.而状态变量 s_n 是指可用于下一门课程分配的天数，容易得到下面的式子

$$s_n = s_{n-1} - x_n (n = 1,2,3) \tag{4.1.1}$$

式中， s_n 的可能取值为 $1,2,\cdots,7$.

注意到每阶段的效益值就由本阶段的状态和本阶段的决策决定，这属于动态规划求解问题的范畴，因此通过动态规划理论来解决这一问题.

4.1.1. 递推关系的确定

对于整个过程而言，目标函数的效益值为各个阶段效益值 p_n 之和（即分数提高量的总和），于是可得递推关系

$$f_n(s_n, x_n) = p_n(s_n, x_n) + Max\sum_{i=n+1}^{3} p_i(x_i) \tag{4.1.2}$$

式中

$$Max\sum_{i=n+1}^{3} p_i(x_i) = f_{n+1}^*(x_n) \tag{4.1.3}$$

阶段之间的约束条件满足

$$\sum_{i=n}^{3} x_i = s_n \tag{4.1.4}$$

在每一阶段取定最佳值之后，式(4.1.2)就可写为

$$f_n^*(s_n) = Max\{p_n(x_n) + f_{n+1}^*(s_n - x_n)\} \tag{4.1.5}$$

这样，我们就利用以上递推公式通过逆序法计算最优解.

4.1.2. 第 3 阶段的最优解

当 $n=3$ 时，由于没有后续过程，于是直接有

$$f_3^*(s_3) = Max\{p_3(x_3)\} \tag{4.1.6}$$

注意到表 3 中所给的提分关系式确定的是坐标平面内离散的点，因此我们通过穷举得到不同状态变量 s_n 下的最优解及最优决策变量如表 5.

s_3	x_3^*	$f_3^*(s_3)$
0	0	0.72
1	1	1.2
2	2	1.56
3	3	1.8
4	4	1.8
5	5	1.8
6	6	1.8
7	7	1.8

表 5 第 3 阶段最优解的情况列表

4.1.3. 第 2 阶段的最优解

接下来，讨论第 2 阶段的最优解.根据递推公式

$$f_2^*(s_2) = Max\{p_2(x_2) + f_3^*(s_2 - x_2)\} \tag{4.1.7}$$

从第 2 阶段$(n=2)$向后移动，讨论不同的状态变量s_2对应的最优决策，给出表 6，其中

$f_3^*(s_2 - x_2)$可以通过查第三阶段的表格得到.

s_2 \ x_2	$f_2^*(s_2) = Max\{p_2(x_2) + f_3^*(s_2 - x_2)\}$								$f_2^*(s_2)$	x_2^*
	0	1	2	3	4	5	6	7		
1	0.72	0.66							0.72	0
2	1.2	1.38	1.32						1.38	1
3	1.56	2.1	2.04	2.2					2.2	3
4	1.8	2.22	2.52	2.92	2.68				2.92	3
5	1.8	2.46	2.88	3.4	3.4	3.3			3.4	3 or 4
6	1.8	2.46	3.12	4.02	3.88	4.02	3.5		4.02	5
7	1.8	2.46	3.12	4.5	4.24	4.5	4.22	3.74	4.5	5

表 6 第 2 阶段最优解的情况列表

4.1.4. 第 1 阶段的最优解

最后考虑$n=1$，即第 1 阶段的情况.由递推公式

$$f_1^*(s_1) = Max\{p_1(x_1) + f_2^*(s_1 - x_1)\} \tag{4.1.8}$$

可得第 1 阶段的最优解如表 7.

s_1 \ x_1	$f_1^*(s_1) = Max\{p_1(x_1) + f_2^*(s_1 - x_1)\}$								$f_1^*(s_1)$	x_1^*
	0	1	2	3	4	5	6	7		
1	4.5	4.9	4.5	4.9	4.4	4.02	4.02	4.4	4.9	1 or 3

表 7 第 1 阶段最优解的情况列表

根据上述求解过程，我们可以得到的最终最佳方案为：

方案一：课程 1：学习 1 天；课程 2：学习 5 天；课程 3：学习 1 天

方案二：课程 1：学习 3 天；课程 2：学习 3 天；课程 3：学习 1 天

总的提分表现为 4.9 分.

考虑到实际情况下，在总提分表现相同时，显然各科均衡提高的方案更具有优越性，因此方案二更胜一筹.

4.2. 结课后方案的设计

在结课后的时间段内，我们按照复习课程具体把整段时间划分为 3 个阶段：第 1 阶段（6 月 29-7 月 2 日，共 3 天）、第 2 阶段（7 月 3 日-7 月 10 日，共 8 天）、第 3 阶段（7 月 11 日

-7 月 15 日，共 5 天），对应应当做出的决策就是在各个阶段内的时间.与结课前类似，目标函数定义为加权平均分的提高量 f .每阶段的初始状态 s_n 为上一阶段休息的天数 x_{n-1} （第一阶段中，上一阶段休息的天数为 0），决策变量为本阶段休息的天数 x_n ，即有

$$s_n = x_{n-1} \tag{4.2.1}$$

这样，本阶段的效益值就由本阶段的状态和本阶段的决策决定，符合动态规划的要求，因此我们同样通过动态规划来解决问题 2.

4.2.1. 递推关系的确定

对于整个过程而言，目标函数的效益值为各个阶段效益值 p_n 之和（即分数提高量的总和），效益值的最优值即最大值，于是可得递推关系

$$f_n^*(s_n) = Max\{p_n(s_n, x_n^*) + f_{n+1}^*(x_n^*)\} \tag{4.2.2}$$

即

$$f_n^*(s_n) = p_n(s_n, x_n^*) + f_{n+1}^*(x_n^*) \tag{4.2.3}$$

而对于第 n 阶段的提高分数，则由本阶段的效率 η_n 、休息天数 x_n 和所复习的科目性质 $g_i(t)$ 决定.根据前述有

$$p_n(s_n, x_n) = \eta_n(s_n, x_n) g_i(d_n - x_n) \tag{4.2.4}$$

而 η_n 与的取值可以通过前面的计算和表格得到，这样，我们就确定了递推关系式，接下来通过逆序法讨论整个过程的最优解.

4.2.2. 第 3 阶段的最优解

在第 3 阶段，考生复习课程 1，由于这是最后一阶段，故有

$$f_3(s_3, x_3) = p_3(s_3, x_3) \tag{4.2.5}$$

式中 $s_3 = 0,1,2,3,4$ ， $x_3 = 0,1,2$.

注意到问题中的提分函数、效率函数均是不连续的，且定义域由有限个点组成，因此我们通过穷举的方式得到最优解.在本阶段中，穷举得表 8.

s_3	x_3	$\eta_3(s_3, x_3)$	$p_3(s_3, x_3)$	$f_3(s_3, x_3)$
0	0	0.8	2.112	2.112
0	1	0.8	1.76	1.76
0	2	1	1.98	1.98

1	0	0.8	2.112	2.112
1	1	1	2.2	2.2
1	2	1	1.98	1.98
2	0	0.8	2.112	2.112
2	1	1	2.2	2.2
2	2	1.2	2.376	2.376
3	0	1	2.64	2.64
3	1	1	2.2	2.2
3	2	1.2	2.376	2.376
4	0	1	2.64	2.64
4	1	1.2	2.64	2.64
4	2	1.2	2.376	2.376

表 8 第 3 阶段目标函数的函数值列表

从表中容易注意到当 s_3 取不同值时，x_3 的对应取值 x_3^*，使 $f_3(s_3,x_3)$ 最优，即

$$f_3^*(s_3)=\begin{cases}2.112, s_3=0, x_3^*=0\\ 2.2, s_3=1, x_3^*=1\\ 2.376, s_3=2, x_3^*=2\\ 2.64, s_3=3, x_3^*=0\\ 2.64, s_3=4, x_3^*=0 \ or\ 1\end{cases} \tag{4.2.6}$$

4.2.3. 第 2 阶段的最优解

在第 2 阶段，考生复习课程 2，根据递推关系有

$$f_2(s_2,x_2)=p_2(s_2,x_2)+f_3(x_2) \tag{4.2.7}$$

式中 $s_2=0,1$，$x_2=0,1,2,3,4$.

与第 3 阶段类似，通过穷举可得表 9.

s_2	x_2	$\eta_2(s_2,x_2)$	$p_2(s_2,x_2)$	$f_3(x_2)$	$f_2(s_2,x_2)$
0	0	0.725	3.0305	2.112	5.1425
0	1	0.725	2.871	2.2	5.071
0	2	0.85	3.179	2.376	5.555
0	3	0.975	3.2175	2.64	5.8575
0	4	0.975	2.7885	2.64	5.4285
1	0	0.725	3.0305	2.112	5.1425
1	1	0.85	3.366	2.2	5.566
1	2	0.85	3.179	2.376	5.555
1	3	0.975	3.2175	2.64	5.8575
1	4	0.975	2.7885	2.64	5.4285

表 9 第 2 阶段目标函数的函数值列表

得到 $f_2(s_2,x_2)$ 的最优值

$$f_2^*(s_2)=\begin{cases}5.8575, s_2=0, x_2^*=3, x_3^*=0\\5.8575, s_2=1, x_2^*=3, x_3^*=0\end{cases} \tag{4.2.8}$$

4.2.4. 第 1 阶段的最优解

在第 1 阶段，考生复习课程 3，根据递推关系有

$$f_1(s_1,x_1)=p_1(s_1,x_1)+f_1(x_1) \tag{4.2.9}$$

式中 $s_1=0$，这是因为在第 1 阶段开始时，由于第一阶段前的第一部分没有安排休息时间，

$x_1=0,1$.

与之前阶段类似，穷举可得表 10.

s_1	x_1	$\eta_1(s_1,x_1)$	$p_1(s_1,x_1)$	$f_2(x_1)$	$f_1(s_1,x_1)$
0	0	0.933333	1.456	5.8575	7.3135
0	1	0.933333	1.12	5.8575	6.9775

表 10 第 1 阶段目标函数的函数值列表

得到 $f_1(s_1,x_1)$ 的最优值

$$f_1^*(s_1)=7.3175, s_1=0, x_1^*=0, x_2^*=3, x_3^*=0 \tag{4.2.10}$$

根据上述求解过程，我们可以得到的最终最佳方案为:
方案一：第 1 阶段：休息 0 天；第 2 阶段：休息 3 天；第三阶段：休息 0 天
总的提分表现为 7.3175 分.

5. 结论

本文通过动态规划设计了期末复习提分的分案，得到的结果合理，对学生的复习具有指导价值.与解决离散问题的穷举法相比，动态规划算法具有显著的优越性.在本文所提出的两个解决方案中，采用穷举法分别需要计算 432 次和 90 次，而采用动态规划算法只需要 49 次和 27 次.当问题的规模更大时，动态规划算法所需要的运算量可以比穷举法小若干个数量级，因此是一种十分高效的方法.

6. 附录:第二部分各阶段的代码

```
/*第 3 阶段代码*/
#include "stdafx.h"
#include "math.h"
```

```
#pragma warning(disable:4996)
#define TIME_MAX 10

float ita(int s, int x,int total_work_day) //效率函数
{
	float rest_day,k;
	rest_day = ((float)s) / 3 + (2 * ((float)x)) / 3;
	if (rest_day < 1)
		k = 1;
	else if (rest_day < 2)
		k = 2;
	else
		k = 3;
	return 0.6 + k /total_work_day;

}
float g1(int work_day) //提分函数
{
	switch (work_day)
	{
	case 1: return 0.88;
	case 2: return 1.1;
	case 3: return 1.98;
	case 4: return 2.2;
	case 5: return 2.64;
	}
}
int main(void)
{
	int s3, work_day, x3,total_work_day=5;//前一阶段休息天数、本阶段工作天数、本阶段休息
天数
	float p3,ita3,f3; //本阶段效益、本阶段效率、本阶段总效益
	FILE *result_3rd;
	char *location = "f:\\c\\optimal\\result_3rd.txt";
	result_3rd = fopen(location,"w");
	for (s3 = 0; s3 < 5; s3++)
	{
		for (x3 = 0; x3 < 3; x3++)
		{
			work_day = 5 - x3;
			ita3 = ita(s3, x3, total_work_day);
			p3 = ita3*g1(work_day);
			f3 = p3;
			printf("s3:%d\tx3:%d\tita3:%f\tp3:%f\tf3:%f\n",s3,x3,ita3,p3,f3);
			fprintf(result_3rd, "%d\t%d\t%f\t%f\t%f\n", s3, x3, ita3, p3,f3);
```

```
		}
	}
	return 0;
}
/*第 2 阶段代码*/

#include "stdafx.h"
#pragma warning(disable:4996)

float ita(int s, int x, int total_work_day) //效率函数
{
	float rest_day, k;
	rest_day = ((float)s) / 3 + (2 * ((float)x)) / 3;
	if (rest_day < 1)
		k = 1;
	else if (rest_day < 2)
		k = 2;
	else
		k = 3;
	return 0.6 + k / total_work_day;
}
float g2(int work_day) //提分函数
{
	switch (work_day)
	{
	case 1: return 0.66;
	case 2: return 1.32;
	case 3: return 2.2;
	case 4: return 2.86;
	case 5: return 3.3;
	case 6: return 3.74;
	case 7: return 3.96;
	case 8: return 4.18;
	case 9: return 4.4;
	}
}
float f3_cal(int s3)
{
	switch (s3)
	{
	case 0:return 2.112;
	case 1:return 2.2;
	case 2:return 2.376;
	case 3:return 2.64;
	case 4:return 2.64;
```

```
	}
}
int main(void)
{
	int s2, work_day, x2,total_work_day=8;//前一阶段休息天数、本阶段工作天数、本阶段休息天数、总工作天数
	float p2, ita2,f2,f3; //本阶段效益、本阶段效率、本阶段与之后所有阶段的总收益、下阶段与下阶段之后所有阶段收益
	FILE *result_2nd;
	char *location = "f:\\c\\optimal\\result_2nd.txt";
	result_2nd = fopen(location, "w");
	for (s2 = 0; s2 < 2; s2++)
	{
		for (x2 = 0; x2 < 5; x2++)
		{
			work_day = 8 - x2;
			ita2 = ita(s2, x2, total_work_day);
			p2 = ita2*g2(work_day);
			f3 = f3_cal(x2);
			f2 = f3 + p2;
			printf("s2:%d\tx2:%d\tita2:%f\tp2:%f\tf3:%f\tf2:%f\n", s2, x2, ita2, p2,f3,f2);
			fprintf(result_2nd, "%d\t%d\t%f\t%f\t%f\t%f\n", s2, x2, ita2, p2, f3,f2);
		}
	}
	return 0;
}
/*第 1 阶段代码*/

#include "stdafx.h"
#pragma warning(disable:4996)

float ita(int s, int x, int total_work_day) //效率函数
{
	float rest_day, k;
	rest_day = ((float)s) / 3 + (2 * ((float)x)) / 3;
	if (rest_day < 1)
		k = 1;
	else if (rest_day < 2)
		k = 2;
	else
		k = 3;
	return 0.6 + k / total_work_day;
}
float g3(int work_day) //提分函数
{
```

```
	switch (work_day)
	{
	case 1: return 0.72;
	case 2: return 1.2;
	case 3: return 1.56;
	}
}
float f2_cal(int s2)
{
	switch (s2)
	{
	case 0:return 5.8575;
	case 1:return 5.8575;
	}
}
int main(void)
{
	int s1, work_day, x1,total_work_day=3;//前一阶段休息天数、本阶段工作天数、本阶段休息天数
	float p1, ita1, f1, f2; //本阶段效益、本阶段效率、本阶段与之后所有阶段的总收益、下阶段与下阶段之后所有阶段收益
	s1 = 0;
	FILE *result_1st;
	char *location = "f:\\c\\optimal\\result_1st.txt";
	result_1st = fopen(location, "w");
	for (x1 = 0; x1 < 2; x1++)
		{
			work_day = 3 - x1;
			ita1 = ita(s1, x1, total_work_day);
			p1 = ita1*g3(work_day);
			f2 = f2_cal(x1);
			f1= p1 + f2;
			printf("s1:%d\tx1:%d\tita1:%f\tp1:%f\tf2:\%f\tf1:%f\n", s1, x1, ita1, p1, f2, f1);
			fprintf(result_1st, "%d\t%d\t%f\t%f\t%f\t%f\n", s1, x1, ita1, p1, f2, f1);
		}
	return 0;
}
```

参考文献

[1] Frederick S Hillier,Gerald J Lieberman. 运筹学导论（第 8 版）[M]. 清华大学出版社, 2007

[2]郭贵祥，胡连梅，认知理论基础上学习效率的计算、论证及应用[J] 教学研究 Jan.2007,Vol30,1