计算机数学建模

第三讲 简单优化模型(2)

周毓明
zhouyuming@nju.edu.cn

南京大学计算机科学与技术系

PageRank

Y.Y. Chen, Q. Gan, T. Suel. I/O-efficient techniques for Computing pagerank. CIKM 2002.

A. T.H. Haveliwala. Efficient computation of PageRank. 1999.

C. Kohlschütter, P. Chirita, W. Nejdl. Efficient parallel computation of PageRank. ECIR 2006.

…..

课程内容

1. 产品存贮模型
2. 软件发布时机

静 态 优 化 模 型

- 现实世界中普遍存在着优化问题
- 静态优化问题指最优解是数(不是函数)
- 建立静态优化模型的关键之一是根据建模目的确定恰当的目标函数
- 求解静态优化模型一般用微分法

静 态 优 化 模 型

(1) 分析题意，列出目标函数$f(x)$

(2) 求x使得，$f(x) \rightarrow$ **min** 或者**max**

(3) 解微分方程 $f'(x) = 0$，得$x = g(c)$

(4) 解的敏感性分析**(sensitivity)**：

$$S(x,c) = \frac{\Delta x / x}{\Delta c / c} \approx \frac{dx}{dc}\frac{c}{x}$$

(5) 目标函数的稳定性分析**(stability)**：

$$S(f,c) = \frac{\Delta f / f}{\Delta c / c} \approx \frac{df}{dc}\frac{c}{f}$$

1. 产品存贮模型

产品存贮模型

问 题

配件厂为装配线生产若干种产品，轮换产品时因更换设备要付生产准备费，产量大于需求时要付贮存费。该厂生产能力非常大，即所需数量可在很短时间内产出。

已知某产品日需求量100件，生产准备费5000元，贮存费每日每件1元。试安排该产品的生产计划，即多少天生产一次（生产周期），每次产量多少，使总费用最小。

要求

不只是回答问题，而且要建立生产周期、产量与需求量、准备费、贮存费之间的关系。

产品存贮模型

问题分析与思考

日需求100件，准备费5000元，贮存费每日每件1元。

- 每天生产一次，每次100件，无贮存费，准备费5000元。

每天费用5000元

- 10天生产一次，每次1000件，贮存费900+800+…+100 =4500元，准备费5000元，总计9500元。

平均每天费用950元

- 50天生产一次，每次5000件，贮存费4900+4800+…+100 =122500元，准备费5000元，总计127500元。

平均每天费用2550元

10天生产一次平均每天费用最小吗?

产品存贮模型

问题分析与思考

- 周期短，产量小 ⇨ 贮存费少，准备费多
- 周期长，产量大 ⇨ 准备费少，贮存费多

⇨ 存在最佳的周期和产量，使总费用（二者之和）最小

- 这是一个优化问题，关键在建立目标函数。

显然不能用一个周期的总费用作为目标函数

目标函数——平均每天的费用(总费用/周期)

产品存贮模型

模 型 假 设

1. 产品每天的需求量为常数 r;

2. 每次生产准备费为 c_1, 每天每件产品贮存费为 c_2;

3. T天生产一次（周期）, 每次生产Q件， 当贮存量为零时，Q件产品立即到来（生产时间不计）；

4. 为方便起见，时间和产量都作为连续量处理。

建 模 目 的

设 r, c_1, c_2 已知，求T, Q 使每天总费用的平均值最小。

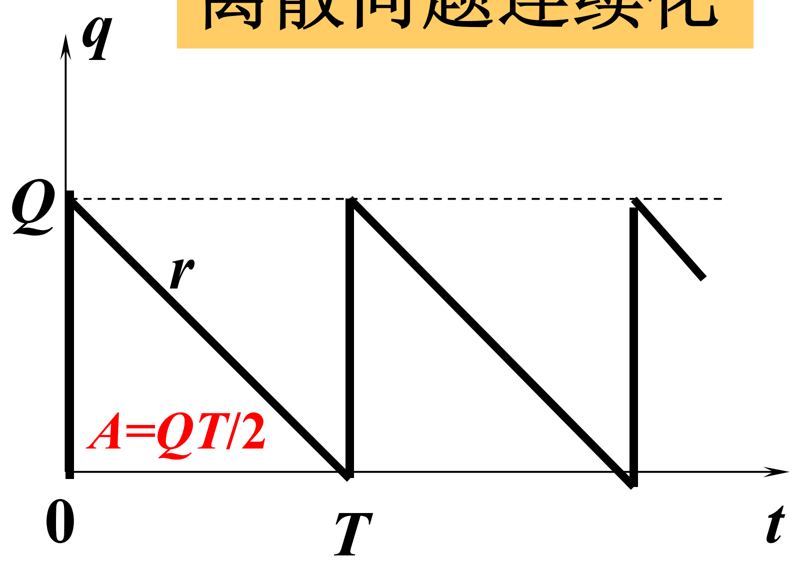
产品存贮模型

模 型 建 立

离散问题连续化

贮存量表示为时间的函数 $q(t)$

$t=0$生产Q件，$q(0)=Q$, $q(t)$以需求速率r递减，$q(T)=0$.



$$Q = rT$$

一周期贮存费为

$$c_2\int_0^T q(t)dt = c_2 A$$

一周期总费用

$$\tilde{C} = c_1 + c_2\frac{Q}{2}T = c_1 + c_2\frac{rT^2}{2}$$

每天总费用平均值（目标函数）

$$C(T) = \frac{\tilde{C}}{T} = \frac{c_1}{T} + \frac{c_2 rT}{2}$$

产品存贮模型

模型求解 求 T 使 $C(T)=\frac{c_1}{T}+\frac{c_2 rT}{2}\to \text{Min}$

$$\frac{dC}{dT}=0 \quad\Rightarrow\quad T=\sqrt{\frac{2c_1}{rc_2}} \qquad Q=rT=\sqrt{\frac{2c_1 r}{c_2}}$$

模型分析

$c_1\uparrow\Rightarrow T,Q\uparrow$ $\qquad c_2\uparrow\Rightarrow T,Q\downarrow$ $\qquad r\uparrow\Rightarrow T\downarrow,Q\uparrow$

模型应用

c_1=5000, c_2=1， r=100

- 回答问题 ⇨ T=10(天), Q=1000(件), C=1000(元)

产品存贮模型

目标函数

$$C(T)=\frac{\tilde{C}}{T}=\frac{c_1}{T}+\frac{c_2 rT}{2}$$

$$\Rightarrow \quad T=\sqrt{\frac{2c_1}{rc_2}}$$

敏感性分析:“解”如何随参数的变化而变化?

当c_1改变时,T如何改变?

当c_2改变时,T如何改变?

当r改变时,T如何改变?

产品存贮模型

$$T = \sqrt{\frac{2c_1}{rc_2}}$$

敏感性分析:

$$S(T, c_1) = \frac{\Delta T / T}{\Delta c_1 / c_1} \approx \frac{dT}{dc_1}\frac{c_1}{T} = 0.5$$

生产准备费增加1%，生产周期增加0.5%

$$S(T, c_2) = -0.5 \qquad S(T, r) = -0.5$$

c_1、c_2和r的微小变化对T的影响很小！

产品存贮模型

目标函数

$$C(T)=\frac{\tilde{C}}{T}=\frac{c_1}{T}+\frac{c_2 rT}{2}$$

$$\Rightarrow T=\sqrt{\frac{2c_1}{rc_2}}$$

稳健性分析: 目标函数值如何随参数的变化而变化?

当c_1改变时, C(T)如何改变?

当c_2改变时, C(T)如何改变?

当r改变时, C(T)如何改变?

产品存贮模型

$$C(T)=\frac{\tilde{C}}{T}=\frac{c_1}{T}+\frac{c_2 rT}{2}$$

稳健性分析:

$$T=\sqrt{\frac{2c_1}{rc_2}}$$

$$S(C,c_1)=\frac{\Delta C/C}{\Delta c_1/c_1}\approx\frac{dC}{dc_1}\frac{c_1}{C}$$

方法1：将T代入C(T)，然后对c_1求导数 $\frac{dC}{dc_1}=\sqrt{\frac{rc_2}{2c_1}}$

方法2：利用链式法则

$$\frac{dC}{dc_1}=\frac{\partial C}{\partial T}\frac{dT}{dc_1}+\frac{\partial C}{\partial c_1}=\frac{\partial C}{\partial c_1}=\sqrt{\frac{rc_2}{2c_1}}$$

$$S(C,c_1)=\frac{dC}{dc_1}\frac{c_1}{C}=\frac{1}{10}\times\frac{5000}{1000}=0.5$$

产品存贮模型

• 经济批量订货公式（EOQ公式）

用于订货、供应、存贮情形

每天需求量 r，每次订货费 c_1, 每天每件贮存费 c_2 ,

T天订货一次(周期), 每次订货Q件，当贮存量降到零时，Q件立即到货。

$$T = \sqrt{\frac{2c_1}{rc_2}} \qquad Q = rT = \sqrt{\frac{2c_1 r}{c_2}}$$

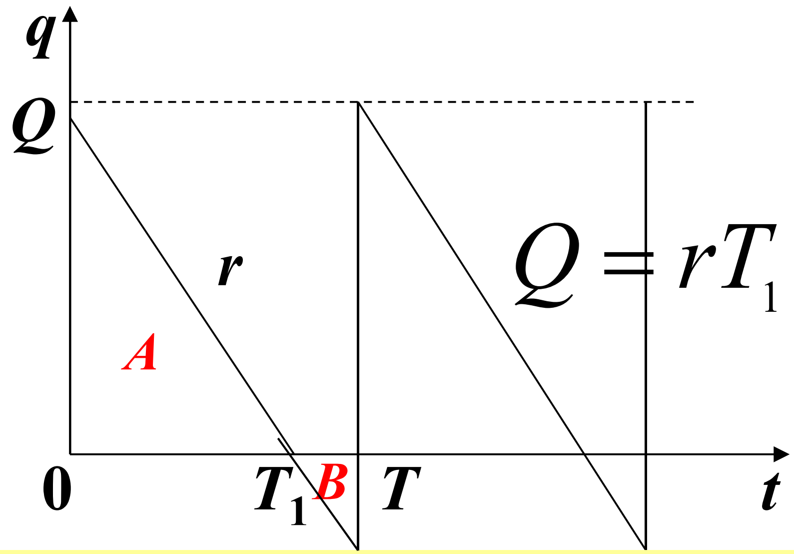
不允许缺货的存贮模型

产品存贮模型

允许缺货的存贮模型

当贮存量降到零时仍有需求r，出现缺货，造成损失

原模型假设：贮存量降到零时Q件立即生产出来(或立即到货)



现假设：允许缺货，每天每件缺货损失费 c_3，缺货需补足

周期T，$t=T_1$贮存量降到零

一周期贮存费 $c_2\int_0^{T_1} q(t)dt = c_2 A$

一周期缺货费 $c_3\int_{T_1}^{T} |q(t)|dt = c_3 B$

一周期总费用

$$\overline{C} = c_1 + c_2\frac{QT_1}{2} + c_3\frac{r(T-T_1)^2}{2}$$

产品存贮模型

一周期总费用 $$\bar{C} = c_1 + \frac{1}{2} c_2 Q T_1 + \frac{1}{2} c_3 r (T - T_1)^2$$

每天总费用平均值（目标函数）

$$C(T,Q) = \frac{\bar{C}}{T} = \frac{c_1}{T} + \frac{c_2 Q^2}{2rT} + \frac{c_3 (rT - Q)^2}{2rT}$$

求 ***T*,*Q*** 使 $C(T,Q) \rightarrow \mathrm{Min}$

$$\frac{\partial C}{\partial T} = 0, \quad \frac{\partial C}{\partial Q} = 0$$

为与不允许缺货的存贮模型相比，***T***记作***T'***，***Q***记作***Q'***

$$T' = \sqrt{\frac{2c_1}{rc_2} \frac{c_2 + c_3}{c_3}}$$

$$Q' = \sqrt{\frac{2c_1 r}{c_2} \frac{c_3}{c_2 + c_3}}$$

产品存贮模型

允许缺货模型

$$T' = \sqrt{\frac{2c_1}{rc_2}\frac{c_2+c_3}{c_3}}$$

$$Q' = \sqrt{\frac{2c_1 r}{c_2}\frac{c_3}{c_2+c_3}}$$

不允许缺货模型

$$T = \sqrt{\frac{2c_1}{rc_2}}$$

$$Q = rT = \sqrt{\frac{2c_1 r}{c_2}}$$

记 $\mu = \sqrt{\frac{c_2+c_3}{c_3}}$

$$T' = \mu T, \quad Q' = \frac{Q}{\mu}$$

$\mu > 1 \Rightarrow T' > T, \quad Q' < Q \qquad c_3 \uparrow \Rightarrow \mu \downarrow$

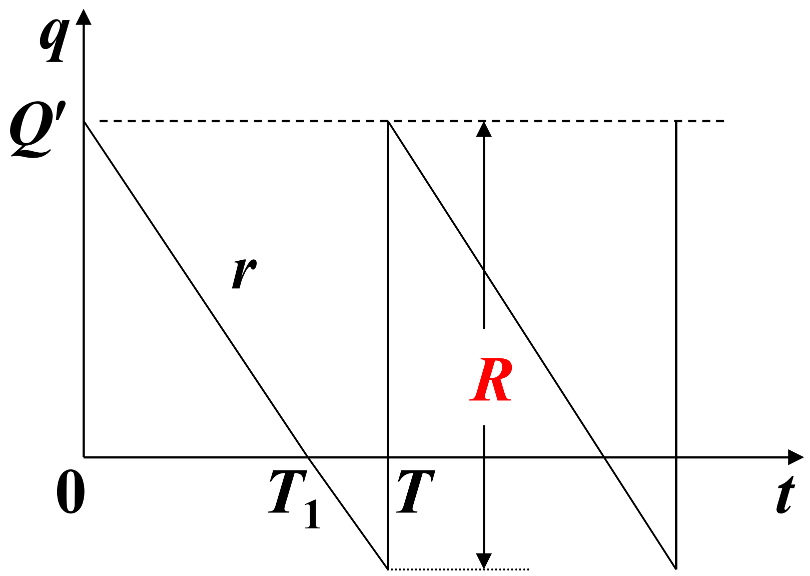
不允许缺货 $\Leftrightarrow$ $c_3 \to \infty \Rightarrow \mu \to 1 \Rightarrow T' \to T, \; Q' \to Q$

产品存贮模型

允许缺货模型

$$T' = \sqrt{\frac{2c_1}{rc_2}\frac{c_2+c_3}{c_3}}$$

$$Q' = \sqrt{\frac{2c_1 r}{c_2}\frac{c_3}{c_2+c_3}}$$



注意：缺货需补足

Q'~每周期初的存贮量

每周期的订货量R $R = rT' = \sqrt{\frac{2c_1 r}{c_2}\frac{c_2+c_3}{c_3}}$

$R = \mu Q > Q$ Q~不允许缺货时的产量(或订货量)

2. 软件发布时机

软件发布时机

问 题

软件系统在什么时间发布最佳？

(1) 如果花很长时间进行测试，虽然可以提高质量，但消耗测试成本，且推迟抢占市场的时间

(2) 如果发布太早，软件中残留的**bug**会很多。当软件失效时再修复缺陷，不仅成本高，且让用户不爽，有可能影响市场份额

要求

假设你是测试组经理，请确定一个最佳的发布时间，使得公司的总成本最低

软件发布时机

问题分析与思考

目标是最小化公司的总成本，而不是测试组的成本

为简便计，假定本公司自己使用该软件

- 在测试前，公司给定一个**deadline D**，延期发布会使公司受损
- 测试中发现**bug**和修正**bug**需要花费成本
- 软件发布后运行中发现**bug**会造成损失，修正**bug**也要成本
- 软件发布后，测试组人员解放出来将使得公司获益

请确定发布日期**t***（**t*** ≤ **D?**）

软件发布时机

模 型 假 设

1. t: 时间变量　　　　t*: 最优发布日期

2. D: 截止日期

3. C1: 发布后每个bug的平均修复成本

C2: 发布后每个bug平均造成的损失　　C3 = C1 + C2

4. C4: 单位测试时间所消耗的CPU成本

5. C5: 测试组完成任务后的人力收益

6. C6: 系统每单位时间的运行成本

7. C7: 单位测试时间所消耗的人力成本

软件发布时机

模 型 假 设

8. B： 系统成功运行每单位时间的收益

9. θ: 测试期间发现一个bug的平均时间

10. N: 系统的bug总数

11. p(t): 系统延期发布的损失函数

$$p(t)=\begin{cases}0, & t<D \\ p_0+p_1(t-D), & t\geqslant D\end{cases}$$

12. C(t): 系统的成本收益函数

求t，使得C(t)最小

软件发布时机

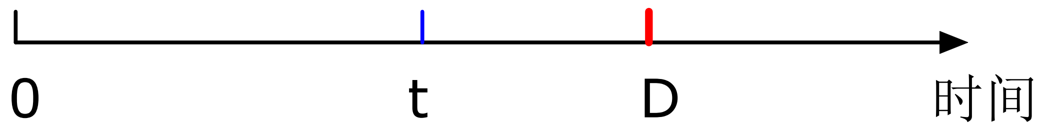
模 型 建 立

C4: 测试CPU成本
C7: 测试人力成本

C6: 单位时间运行成本
C5: 测试完成人力收益
B: 成功运行收益

C3: = C1+C2 缺陷发现和修复成本

Case 1： $t \le D$



[0, t]: 测试成本 $(C4+C7)t$

[t, D]: 系统运行的成本-收益 $(C6-C5-B)(D-t)$

[t, ∞): 维护成本 [t, ∞): C3×t时刻时系统中剩余的bug数目

软件发布时机

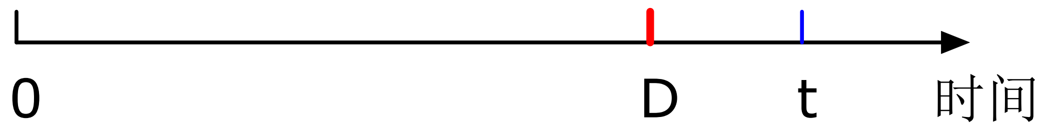
模 型 建 立

C4: 测试CPU成本
C7: 测试人力成本

P(t): 延期损失

C3: = C1+C2 缺陷发现和修复成本

Case 2： t > D



[0, t]: 测试成本 $(C4+C7)t$

[D, t]: 延期发布的损失 $p(t)$

[t, ∞): 维护成本 [t, ∞): C3×t时刻时系统中剩余的bug数目

软件发布时机

模 型 建 立

$$C(t)=\begin{cases}(C4+C7)t+(C6-C5-B)(D-t)+C3\times R(t) & t\le D\\ (C4+C7)t+p(t)+C3\times R(t) & t>D\end{cases}$$

t时刻系统中剩余的bug数目R(t)怎样求？

软件发布时机

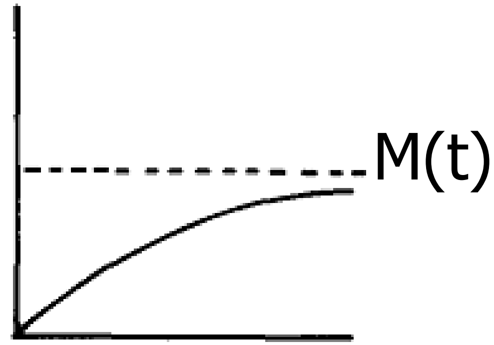
模 型 建 立

t时刻系统中剩余的bug数目R(t)怎样求？

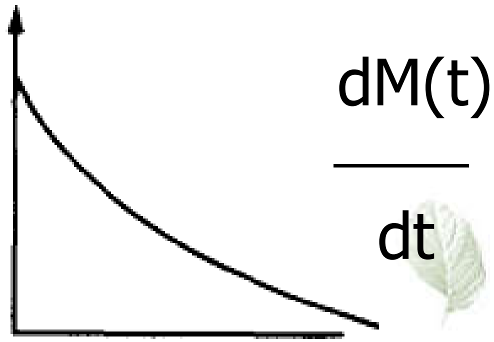
若M(t)为到t时检测到的累积的bug数目，则N=M(t) + R(t)

假设t时刻bug的检测速度与R(t)成正比

$$\frac{dM(t)}{dt}=\lambda R(t)=\lambda(N-M(t))$$



$$M(t)=N(1-e^{-\lambda t})$$



$$R(t)=Ne^{-\lambda t}=Ne^{-t/\theta}$$

软件发布时机

模型建立

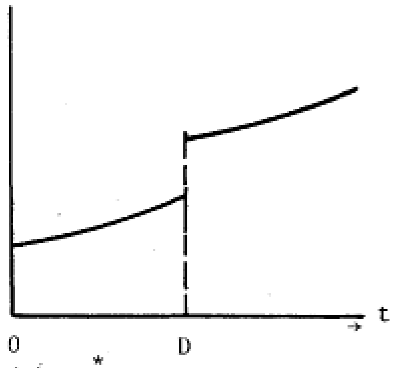
$$C(t)=\begin{cases} C3Ne^{-t/\theta}+(C4+C7-C6+C5+B)t+(C6-C5-B)D & t\le D \\ C3Ne^{-t/\theta}+(C4+C7)t+p(t) & t>D \end{cases}$$

$$p(t)=\begin{cases} 0, & t<D \\ p_0+p_1(t-D), & t\geqslant D \end{cases}$$

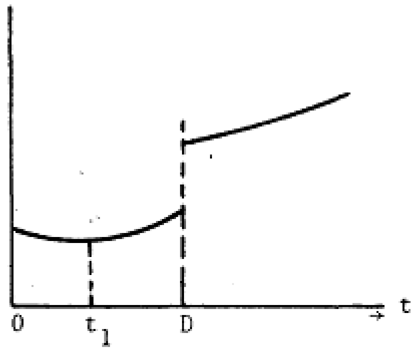
求t，使得C(t)最小

软件发布时机

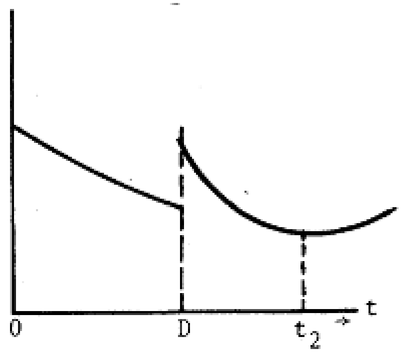
模 型 求 解



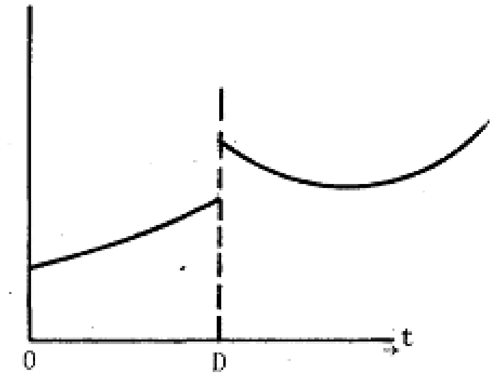
(a) $t^* = 0$



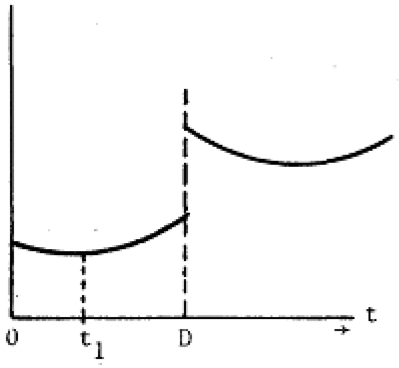
(b) $t^* = t_1$



(c) $t^* = t_2$



(d) $t^* = 0$



(e) $t^* = t_1$

软件发布时机

模型求解

$$C(t)=\psi_1(t)I_{[0,D)}(t)+\psi_2(t)I_{[D,\infty)}(t)$$

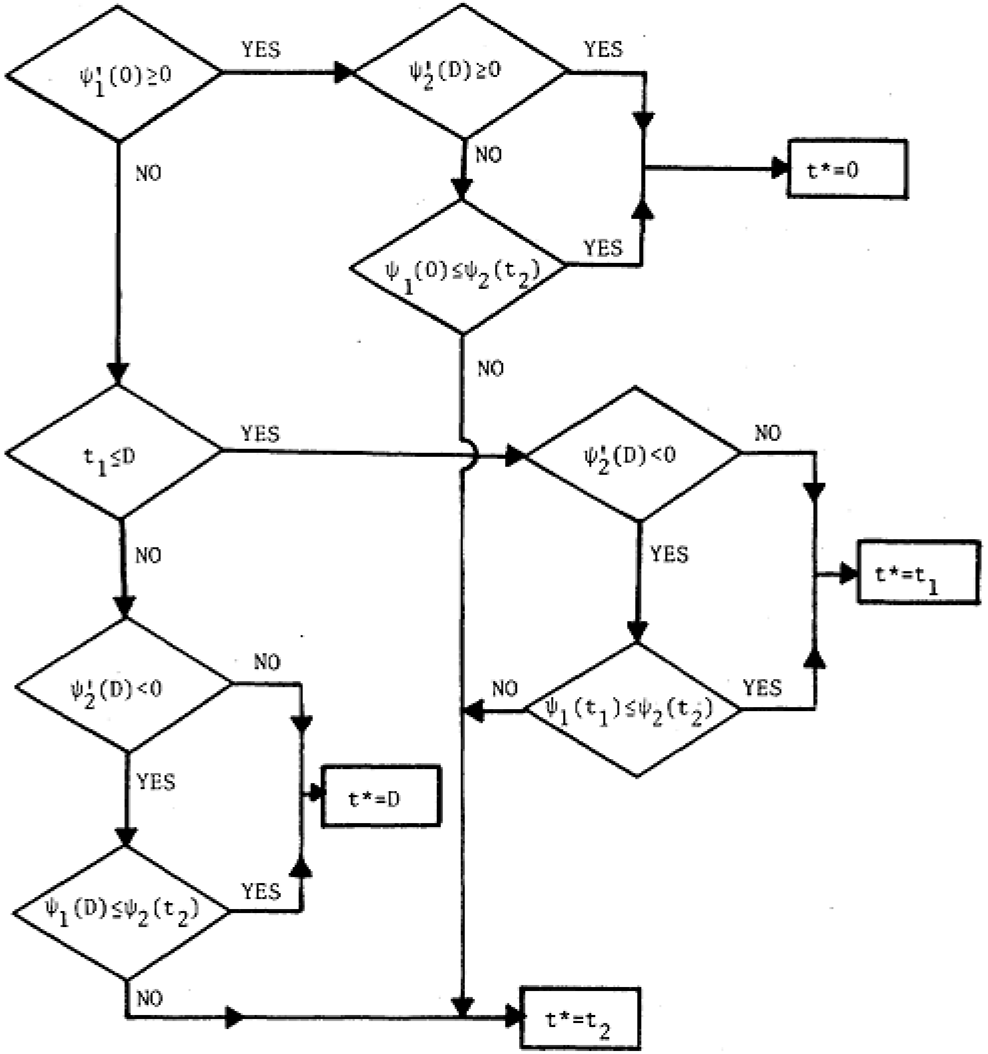
$$\psi_1(t)=C_3Ne^{-t/\theta}+(C_4+C_7-C_6+C_5+B)t$$
$$+(C_6-C_5-B)D, \qquad -\infty<t<\infty$$

$$\psi_2(t)=C_3Ne^{-t/\theta}+(C_4+C_7)t+p(t), \qquad -\infty<t<\infty$$

and $I_A(t)$ denotes the indicator function, i.e., $I_A(t)=1$ if $t\in A$, $I_A(t)=0$ if $t\notin A$.

软件发布时机

模 型 求 解



ψ'1(0)≥0
YES
ψ'2(D)≥0
YES
NO
t*=0
NO
YES
ψ1(0)≤ψ2(t2)
NO
t1≤D
YES
ψ'2(D)<0
NO
NO
YES
t*=t1
NO
ψ1(t1)≤ψ2(t2)
YES
ψ'2(D)<0
NO
YES
t*=D
YES
ψ1(D)≤ψ2(t2)
NO
t*=t2

软件发布时机

模型求解

NECESSARY AND SUFFICIENT CONDITIONS FOR THE OPTIMAL DECISION

Optimal Decision t^*	$t_1 \geq 0$		$t_1 < 0$
	$t_1 \leq D$	$t_1 > D$	
0			(i) $t_2 \leq D$, or (ii) $t_2 > D, \psi_1(D) \leq \psi_2(t_2)$
t_1	(i) $t_2 \leq D$, or (ii) $t_2 > D, \psi_1(t_1) \leq \psi_2(t_2)$		
D		(i) $t_2 \leq D$, or (ii) $t_2 > D, \psi_1(D) \leq \psi_2(t_2)$	
t_2	$t_2 > D, \psi_1(t_1) > \psi_2(t_2)$	$t_2 > D, \psi_1(D) > \psi_2(t_2)$	$t_2 > D$, $\psi_1(D) > \psi_2(t_2)$

软件发布时机

模 型 验 证

常量	例1	例2	例3
C3	3500	6000	6000
C4	1000	1000	1000
C5	5000	5000	5000
C6	2500	2500	2500
C7	8000	8000	8000
B	10000	10000	10000
P0	0	10000	10000
D	3	2	3
N	30	50	50
θ	1.2	1.6	1.6
t*	7.4	2.4	2.4

软件发布时机

存在的问题

1. 若系统开发给其他公司使用，怎样求**t*?**

H.S. Koch, P. Kubat. Optimal release time of computer software. IEEE TSE, 1983.

2. 先前假定**bug**数目固定，修复**bug**不引入新的**bug**。如果是“**imperfect debugging**”，怎样求**t*?**

M. Xie, B. Yang. A study of the effect of imperfect debugging on software development cost. IEEE TSE, 2003.

3. 先前假定系统在测试期间不变。如果是边开发边测试，怎样求**t*?**

S.R. Dalal, A.A. McIntosh. When to stop testing for large software systems with changing code. IEEE TSE, 1994.

软件发布时机

存在的问题

4. 先前求的C(t)是期望成本，不是实际成本，如何量化它的不确定性并分析其对t*的影响?

B. Yang. A study of uncertainty in software cost and its impact on optimal software release time. IEEE TSE, 2008.

5. 如果用户对系统的可靠性给出了特定的要求，此时如何求t*?

H. Pham, X. Zhang. A software cost model with warranty and risk costs. IEEE TC, 1999.

6. 如果公司的成本预算是固定的，怎样求t*?

Y.W. Leung. Optimum software release time with a given cost Budget. Journal of Systems and Software, 1992.

软件发布时机

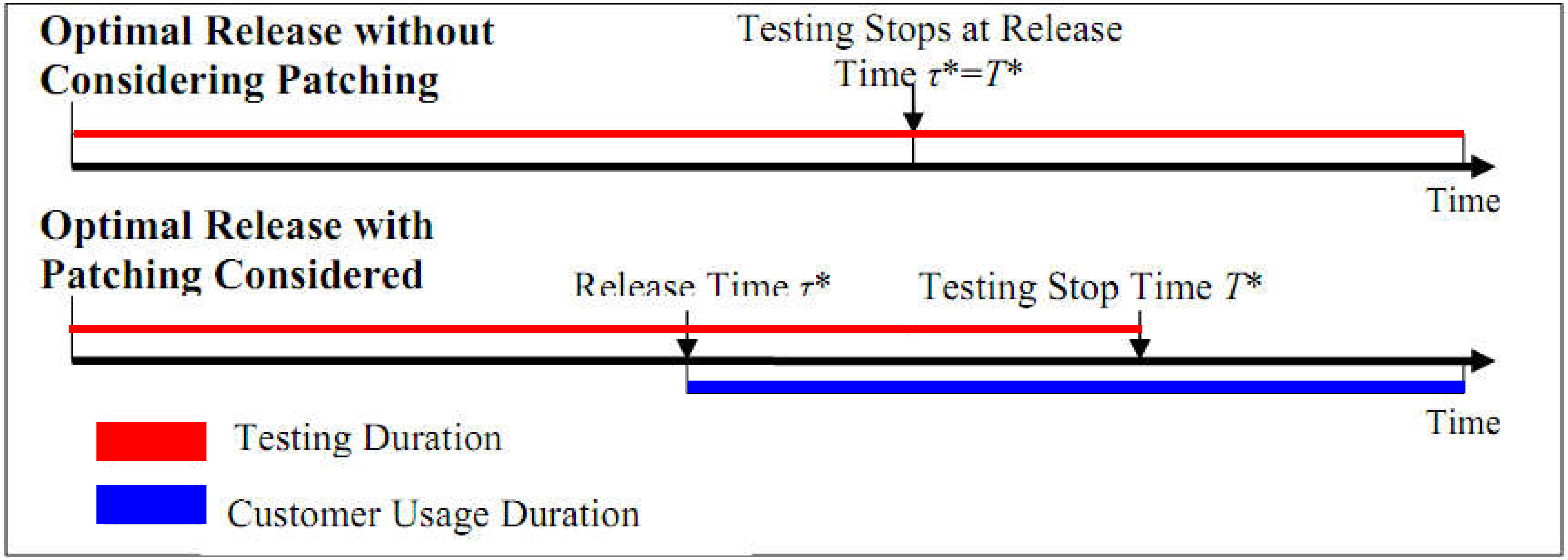
存在的问题

7. 如果公司经常给已发布给用户的系统打补丁，怎样求**t*?**

Z. Jiang, S. Sarkar. Optimal software release time with patching considered, WITS 2003

软件发布时机

7. 打补丁时



软件发布时机

N –Total number of bugs in the software when testing begins T – Testing stop time τ – Release time $u(\tau)$ – Expected number of undetected bugs at the time of release k – Cost of testing per unit time a – Cost of one software failure in the field

1. 不考虑打补丁时

测试成本： $C_1(\tau) = k \cdot \tau$

维护成本： $C_2(\tau) = a \cdot u(\tau) = a \cdot N \cdot e^{-\lambda\tau}$

市场机会成本： $C_3(\tau) = m \cdot \tau^2,\ (m > 0)$

总成本

$$EC(\tau) = k \cdot \tau + a \cdot N \cdot e^{-\lambda\tau} + m \cdot \tau^2$$

软件发布时机

2. 考虑打补丁时

$$M(t) = N(1 - e^{-\lambda t})$$

[0, τ]间检测一个bug的概率：

$$F_1(\tau, T) = 1 - e^{-\lambda\tau}$$

[τ, T]间检测一个bug的概率：

$$F_2(\tau, T) = e^{-\lambda\tau} \cdot (1 - e^{-(r+1)\lambda(T-\tau)}) = e^{-\lambda\tau} - e^{r\lambda\tau} \cdot e^{-(r+1)\lambda T}$$

[T,∞)间检测一个bug的概率：

$$F_3(\tau, T) = e^{-\lambda\tau} \cdot e^{-(r+1)\lambda(T-\tau)} = e^{r\lambda\tau} \cdot e^{-(r+1)\lambda T}$$

软件发布时机

2. 考虑打补丁时

总成本＝ 测试成本+发布后的维护成本(客户检测到的bug)+市场机会成本

$$EC_p(\tau,T) = k \cdot T + a \cdot N[F_2(\tau,T)\frac{r}{r+1} + F_3(\tau,T)] + m \cdot \tau^2$$

$$客户bug检测速度 = r\lambda$$

k=50, λ = 0.1, N = 1000, a = 20, m = 7, r = 0.4

Without patching: τ* = T* = 19, EC* = 6468

With patching: τ* = 12, T* = 30, EC* = 4575

小结

- 产品存储模型
- 软件发布时机

思考题

1. 怎样解决NRP(Next Release Problem)?
2. 怎样解决无线网中的许多问题（例如基站布局规划）？

Thanks for your time and attention!