

杉木原木最適造材之經濟分析

陳麗琴

摘 要

現今伐木業者所使用造材之方法，多屬經驗法則，能否達到最適經濟價值則無法得知，本文特別著重於研究有效的方法，使用作業研究之動態規劃 (Dynamic Programing) 中之正向解法 (positive solution method) 尋求最有利之造材方法。本研究係採用中部地區杉木市場規格與價格，並考慮造材應予之延寸及多種原木截取的可能長度，於動態規劃時，可作較實際與精密之多層決策，提供了杉木從 8m 到 24 m 長之幹直而無缺陷之伐倒木及不同直徑下之最佳造材方法，並經證明此法比經驗法更能提高林木利用價值，及伐木業者之利潤。

關鍵詞：杉木，造材，經濟分析，動態規劃。

陳麗琴 1987，杉木原木最適造材之經濟分析，林業試驗所研究報告季刊，2(3)：227—240。

An Economic Analysis of Optimal Bucking Pattern of China-fir in Taiwan

Lih-Chin Chen

[SUMMARY]

Generally, logger uses his experience to do bucking decision. Whether optimal economic return can be achieved by this method is not known.

In this study, based on log size and its price in China-fir log market in central Taiwan, Dynamic programing is conducted for optimal bucking decision in consideration of trim allowance and cutting size at the same time. In this analysis, a much practical and precise multiple-stage decision can be made to provide better bucking patterns for straight, non defective timbers of 8 to 24m in length and different diameters. This approach is proved to be better than decision by experience, and of course,

high efficient utilization of timber and better profit of logger are increased.

Key words: China fir, Bucking, Economic analysis, Dynamic Programming

Chen, L. C. 1987. An economic analysis of optimal bucking pattern of China-fir in Taiwan. Bull. Taiwan For. Res. Inst. New Series 2(3): 227-240.

一、緒 言：

所謂造材作業，是將一株伐倒的立木，經裁截成適合市場需要之產品，原木的價值依其在市場上的供需關係而定，當市場對原木需求高，價格亦隨之增加，反之則降低，因此所截取原木的長度、大小及品質，很明顯地決定了一株伐倒木的價值。

早期的造材方式，其目標多著重於如何在最短時間內造材取得最大材積而已，對市場需求考慮較少，如今的伐木業者多以市場為導向，但截取方式則使用經驗法，能否達到最佳經濟效益則無法得知，因此，本文特別著重於研究有效的分析方法，使用作業研究中動態規劃法 (Dynamic Programming) 尋求最適造材方法，以提高伐倒木利用價值。

動態規劃是一種計量的運算技巧，將一個整體的複雜問題分成多個彼此相關聯之階段，分析各種複雜的多階段決策程序，以尋求最有利的策略來解決連續性決策問題 (劉一忠，民國66年)。動態規劃問題有下列幾點特徵：(1)問題可分數個階段 (Stage)，每一階段均需作決策。(2)每一階段有不同情況 (State)，(3)每一階段所做決策會影響到下一階段的決策。(4)在現有情況下，對以後所做最佳決策並不受以前階段政策的影響。(5)有遞迴關係式 (recursive relation) 連繫相鄰之兩階段 (高強，1980)。原木造材方法即具有動態規劃問題之型態，因此極適於用此方法來分析。杉木是本省相當重要的造材樹種，全省造林面積有5萬餘公頃，多集中於中部地區，交易頻繁，因此本研究以杉木為對象，配合目前市場規格與價格，探討杉木伐倒木在不同長度及伐倒直徑下之造材方法，以求得

伐倒木之最適經濟價值。

二、前人研究：

早期造材作業，多著重於如何於短時間內造材，取得最大材積，現在的伐木業者，則依市場的需求規格與市價而造材，利潤提高。在鍾崇志，魏立志 (民國66年) 文中曾提到有關如何判斷造材之技術，以提高利潤，雖可提供參考，但造材選擇種類少並不完整，仍屬經驗法則。此種需要分層考慮應否截取原木的多層決策問題，極適合應用動態規劃法解決。動態規劃的創始者為1950年代Richard E. Bellman，此法與其他數學規劃法不同之點，在於它比較缺少規律性的求解屬，沒有符合一組特定形式的方程式，解答方法限制較少且較簡單 (劉一忠，民國66年)。

近年來動態規劃應用的相當廣泛，在林業上無論是林木經營、林產品或水資源經營，皆有這方面的研究，Pnevmaticos 及 Mann 二氏 (1972) 曾設計出簡單之動態規劃模式 (Dykstra, 1984; Pnevmaticos, 1972)，選用三種原木長度及5個階段來解決造材方法，本省至目前尚無人研究造材技術，本研究則考慮較複雜之實際原木市場規格、價格與截取時應予之延寸，以原木截取長度間隔為0.2 m，作成較完整之動態規劃問題，分析尋求最適造材方法。

三、材料與方法：

本研究係採用中部地區杉木市場規格與價格及蒐集杉木之伐倒木長與伐倒直徑，應用動態規劃法決定杉木最適造材方法。

研究方法如下：

(一)調查中部地區集材場一般杉木需求之原木規格與價格，對於特殊用材如電桿材等則不列入其中，並設定不在規格內之原木沒有價值，每一原木均有延寸6 cm。

(二)調查與蒐集本省杉木伐倒木之伐倒處至末徑3 cm處的長度及伐倒處直徑。

(三)杉木造材原木材積計算公式係根據林務局檢尺及分等辦法(林務局，民國71年)而定：

材積 = (末端直徑 + 定數)² × 0.79 × 長度
定數未滿40 cm者，就其材長每公尺加算0.7 cm，在40 cm以上者，每公尺加算1.6 cm；材長以實際長度計算。

(四)應用動態規劃之正向解法(Forward solution method)編寫造材動態規劃問題(其程式於附錄所示)，其遞迴關係式(recursive relation)如下所示：

$$f_{n+1}^*(S) = \max_x \{ V(X_{n+1}, S) + f_n^*(X_n) \}$$

動態規劃問題之階段(Stage)與情況(State)的代表意義分別說明如下：

階段：伐倒木經截取之原木的順序位置，例如第1階段代表第1段原木。

情況：為至第i階段之原木總長度(從第1階段至第i階段原木長度合計)。

茲將求解之過程簡略說明如下：

1. 第1階段

$$f_1(S, X_1) = V_1(S, X_1) \dots\dots\dots(1)$$

X₁：第1階段原木截取之長度，設原木截取最小長度為0.2 m，最大截取長度為7.2 m，其間隔為0.2 m。

S：第1階段原木長度。

V₁(S, X₁)：第1階段在(S, X₁)狀況下之原木價值。

f₁(S, X₁)：第1階段在(S, X₁)狀況下之

原木價值。

2. 第i階段

$$f_i(S, X_i) = V_i(S, X_i) + f_{i-1}^*(S - X_i) \dots\dots\dots(2)$$

i > 1

S：從第1階段至第i階段之原木總長度。

S - X_i：為至第i階段原木總長度減去第i階段之原木長作為前一階段(i - 1)之原木總長。

V_i(S, X_i)：第i階段在(S, X_i)狀況下之該原木價值。

f_{i-1}^{*}(S - X_i)：至第i - 1階段，情況為(S - X_i)下，i - 1段原木總長之最適價值。

f_i(S, X_i)：第i階段在(S, X_i)狀況下，i段原木總長之價值。

3. 方程式(1)與(2)合併

$$f_i^*(S) = \max_{X_i} \{ V_i(S, X_i) + f_{i-1}^*(S - X_i) \}$$

X_i：為各種原木截取之長度。

f_i^{*}(S)：第i階段在S下之i階段原木總長最適價值。

由最後階段的最適值，可得知最後階段應截取之原木長，經遞迴關係，可倒推至各階段應截取之原木長，而得最佳之造材方法。

(五)動態規劃程式之輸入方式：

- 1. 輸入杉木市場需求的原木長度，末徑與相對應之價格。
- 2. 輸入所欲造材之伐倒木長度，伐倒直徑及末徑。

(六)動態規劃程式之輸出方式：

將程式與輸入資料上機後，所得結果報表為：

- 1. 該伐倒木長度，伐倒直徑及末徑。
- 2. 伐倒木最適經濟價值。
- 3. 伐倒木最適造材方式。

四、結果與討論：

(一)長度從8 m到24 m之伐倒木，經動態分析後，所得造材結果如下表：

表 1 不同伐倒木長度及直徑之最適造材方式

伐倒木長度 (m)	伐倒直徑 (cm)	最 適 值 (NT\$)	原 木 截 取 長 度 (m)							
			1	2	3	4	5	6	7	
8	12	175	4.2	3.6						
8	14	203	4.2	3.6						
8	16	233	4.2	3.6						
8	18	243	0.2	4.2	3.0					
8	20	255	0.4	4.2	3.0					
12	16	365	4.2	3.6	3.6					
12	18	435	4.2	4.2	3.0					
12	20	497	4.2	4.2	3.0					
12	22	564	4.2	4.2	3.0					
12	24	636	4.2	4.2	3.0					
12	26	738	4.2	4.2	3.0					
14	16	464	4.2	4.2	3.6					
14	18	540	4.2	4.2	3.6					
14	20	621	4.2	4.2	3.6					
14	22	676	4.2	0.4	4.2	3.6				
14	24	785	4.2	0.8	4.2	3.6				
14	26	864	4.2	4.2	3.6					
14	30	1,082	4.2	1.8	4.2	3.0				
16	18	641	4.2	0.2	4.2	3.6	3.0			
16	20	705	4.2	4.2	3.6	3.6				
16	22	805	4.2	4.2	3.6	3.6				
16	24	950	4.2	4.2	4.2	3.0				
16	26	1,068	4.2	4.2	4.2	3.0				
16	30	1,325	4.2	4.2	4.2	3.0				
16	32	1,465	4.2	4.2	4.2	3.0				
16	34	1,636	4.2	4.2	4.2	3.0				
18	18	726	4.2	4.2	4.2	3.6				
18	20	838	4.2	4.2	4.2	3.6				
18	22	983	4.2	4.2	4.2	3.6				
18	24	1,115	4.2	4.2	4.2	3.6				
18	26	1,256	4.2	4.2	4.2	3.6				
18	28	1,387	4.2	4.2	0.2	4.2	3.6			
18	30	1,546	4.2	4.2	0.8	4.2	3.6			
18	32	1,696	4.2	4.2	4.2	3.6				
18	34	1,869	4.2	4.2	4.2	3.6				
18	36	2,051	4.2	4.2	4.2	3.6				
18	38	2,242	4.2	4.2	4.2	3.0	1.8			

表 1 不同伐倒木長度及直徑之最適造材方式 (續)

伐倒木長度 (m)	伐倒直徑 (cm)	最 適 值 (NT\$)	原 木 截 取 長 度 (m)						
			1	2	3	4	5	6	7
20	18	841	4.2	4.2	4.2	3.6	3.0		
20	20	971	4.2	4.2	4.2	3.6	3.0		
20	22	1,137	4.2	4.2	4.2	3.6	3.0		
20	24	1,248	4.2	4.2	0.8	4.2	3.6		
20	26	1,395	4.2	4.2	4.2	3.6	3.0		
20	28	1,587	4.2	4.2	4.2	3.6	3.0		
20	30	1,768	4.2	4.2	4.2	3.6	3.0		
20	32	1,967	4.2	4.2	4.2	4.2	2.4		
20	34	2,168	4.2	4.2	4.2	4.2	2.4		
20	36	2,378	4.2	4.2	4.2	4.2	2.4		
20	38	2,599	4.2	4.2	4.2	4.2	2.4		
20	40	2,829	4.2	3.6	4.2	4.2	3.0		
22	18	949	4.2	4.2	4.2	3.6	3.6	1.8	
22	20	1,079	4.2	4.2	0.2	4.2	3.6	3.6	
22	22	1,244	4.2	4.2	4.2	4.2	4.2	3.6	
22	24	1,413	4.2	4.2	4.2	4.2	3.6		
22	26	1,618	4.2	4.2	4.2	4.2	3.6		
22	28	1,812	4.2	4.2	4.2	4.2	3.6		
22	30	2,018	4.2	4.2	4.2	4.2	3.6		
22	32	2,236	4.2	4.2	4.2	4.2	3.6		
22	34	2,465	4.2	4.2	4.2	4.2	3.6		
22	36	2,670	4.2	4.2	4.2	4.2	3.6		
22	40	3,200	4.2	4.2	4.2	0.8	4.2	3.6	
22	42	3,459	4.2	4.2	4.2	4.2	3.6		
22	44	3,743	4.2	4.2	4.2	3.0	1.8		
24	18	1,060	4.2	4.2	2.0	4.2	4.2	3.6	2.4
24	20	1,194	4.2	4.2	4.2	4.2	3.6	3.0	
24	22	1,395	4.2	4.2	4.2	4.2	3.6	3.0	
24	24	1,585	4.2	4.2	4.2	4.2	3.6	3.0	
24	26	1,814	4.2	4.2	4.2	4.2	3.6	3.0	
24	28	2,009	4.2	4.2	4.2	0.2	4.2	3.6	2.4
24	32	2,448	4.2	4.2	4.2	4.2	3.6	3.0	
24	36	2,994	4.2	4.2	4.2	4.2	3.6	3.0	
24	40	3,575	4.2	4.2	4.2	4.2	4.2	2.4	
24	44	4,199	4.2	4.2	4.2	4.2	4.2	2.4	

※最適值為伐倒木經動態規劃造材後之最佳經濟價值。

(二)不同伐倒木長度與直徑有不同截取方法，從表中可看出，長 4.2m 之原木出現率最大，顯示 4.2m 長的原木其價格較高，截取較為有利，其次是 3.6m，3.0m，2.4m。其中偶而出現無價值之 0.2m 或 0.4m 之原木段，就其原因是因為原木價格的高低視市場的需求而定，並非既長且粗的原木單價一定比瘦短的原木來得有價值，因此必須注意截取的位置，有時寧可棄置一小段，以換取較高價值之原木段。例如長度為 8m 之伐倒木伐倒直徑為 18cm，經動態規劃得第一段 0.2m，第二段 4.2m，第三段 3.0m，其總價值為 243 元，若依一般經驗法，則可能取第一段 4.2m，第二段 3.6m，其總價值為 197 元，損失 46 元。前者雖然增加了截取的費用和需要額外的處理費用，但與因截取正確所增加之利潤比較應是不成比例的。

(三)經由動態規劃之造材方法，可提高立木之利用價值，從而引導林木經營者如何調整林木生長空間及伐期的長短，以增加林地之經濟生產力。

(四)本研究適用的對象為幹直且無缺點之杉木，對於樹幹彎曲及有缺陷之林木尚須作進一步之探討。

(五)本研究係配合現階段中部地區杉木市場規格與價格，求得最適造材方法，將來若市場規格及價格改變，只須將程式之輸入資料改變，經上機，又可得新的造材方法，極為方便。

(六)本研究截取原木長度的間隔設定為 0.2m，此係數可隨樹種的價值高低而定，如紅檜樹種可改為 0.1m，如雜木等不值錢之樹種則間隔可設定大些，以節省動態規劃上機時間。

五、結 論：

(一)動態規劃係為解決連續性政策問題而設計的，造材方法的決定亦具有此種特性，因此極適用此法來解決。

(二)經由動態規劃造材，可提高立木之利用價值，並了解何種性質之立木最具價值，從而引導林木經營者如何調整林木生長空間及伐期的長短，以增加林地之經濟生產力。

(三)本研究適用的對象為幹直且無缺點之杉木，對於樹幹彎曲及有缺陷之林木尚須再作進一步之探討。

(四)當木材市場規格與價格變化時，應用動態規劃分析，可很迅速地調整出最適造材方法，比經驗法要有利的多。

引用文獻：

林務局 1982. 林務局木材檢尺及分等。

高 強 1980. 育林措施施行之最適時期與集約度，中華林學季刊 Vol. 13 No. 4.

劉一忠 1977. 作業研究，政大企管叢書。

鍾崇志，魏立志 1977. 林木採運作業技術——如何在區分造材中獲得較大之利潤，臺灣林業，3(9)、(10)、(11)、(12)、4(1)。

Dykstra, Dennis P. 1984 Mathematical Programming for Natural Resource Management PP. 287-309 Mc Graw-Hill Book Company.

Pnevmaticos, S. M., and Mann, S. H., 1972 Dynamic Programing in Tree Bucking For. Prod. Jor. 22(2): 26-30

附 錄：

```
//FO4BBUCK JOB FO4,CHIN,CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),
//          NOTIFY=FO4B
//          EXEC FORTXCLG,REGION=6,000 K,TIME.GO=120
//FORT,SYSIN DD*
      INTEGER DN,DECS,TIM,SDIM,STATE,DIM,TLOG,DIF
      REAL*8 TAPER,VOL
      REAL MAXVAL,MAXL
      DIMENSION RDIM(150,150),DECS(150),STATE(150)
      COMMON MAXVAL(150,150),LOG(50),DIM(50)
      COMMON /A/VALUE(50,150,150),IND(150,150)
      COMMON /B/LM,DN,PRICE1(50,50)
      COMMON /C/TLOG(150,150)
      5  FORMAT (I4,13I4/(4X,13I4))
      10 FORMAT (3I4)
      15 FORMAT (7F5.0)
      READ (5,5) LM,(LOG(I),I=1,LM)
      READ (5,5) DN,(DIM(I),I=1,DN)
      WRITE (6,11)
      11 FORMAT(1H1,40X,"DIAMETER"//11X,"LENGTH",15X,"3",
      *5X,"7",5X,"8",5X,"10",5X,"15",5X,"17",5X,"18"/)
      DO 17 I=1,LM
      READ (5,15) (PRICE1(I,J),J=1,DN)
      17 WRITE (6,12)LOG(I),(PRICE1(I,J),J=1,DN)
      12 FORMAT (1X,10X,I4,12X,7F7.0)
      READ (5,5) NUMBER
      DO 1000 NUM=1,NUMBER
      READ (5,10) TIM,LDIM,SDIM
      DUM1=LDIM-SDIM
      DUM2=TIM
      TAPER=DUM1/DUM2
      LOGS=(TIM-6)/LOG(1)
      WRITE (6,18) TAPER,LOGS
      18 FORMAT(1H1,"TAPER = ",F6.4,5X,"POSSIBLE MAX. NO. OF LOGS :",14)
      WRITE (6,13)
```

```

13  FORMAT ( ///1X, "LENGTH", 3X, "DIAMETER", 4X, "VOLUME" )
      DO 20 J=1, LM
      TLOG ( J,1)=LOG ( J )
      RDIM ( J,1)=LDIM-TAPER* ( LOG ( J )+6 )
C*** CALL CORECT ( RDIM1, RDIM ( J,1), LOG ( J ) )
      CALL CORECT ( RDIM1, RDIM ( J,1), LOG ( J ) )
      VOL=RDIM1 * RDIM1 * 0.79 * LOG ( J ) / 1000000
      WRITE ( 6,19 ) LOG ( J ), RDIM ( J,1 ), VOL
19  FORMAT ( 1X, I6, 5X, F6.2, F10.7 )
      IND ( J,1)=J
20  MAXVAL ( J,1)=PRICE ( LOG ( J ), RDIM ( J,1 ) ) * VOL
C    WRITE ( 6,22 ) ( MAXVAL ( J,1 ), J=1, LM )
C 22  FORMAT ( 1X, "MAXVAL", 5F10.0 )
      STATE ( 1 )=LM
      K = 2
200 K1=K-1
      LSTATE = STATE ( K1 )
      JJ=0
      DO 40 J=1, LSTATE
      DO 40 I=1, LM
      JJ=JJ+1
      TLOG ( JJ, K )=LOG ( I )+TLOG ( J, K1 )
      IF ( ( TLOG ( JJ, K )+K * 6 ) .GT. TIM ) JJ=JJ-1
40  CONTINUE
      IF ( JJ .NE. 0 ) GO TO 42
      GO TO 201
42  STATE ( K )=JJ
C*** CALL ORDER ( STATE ( K ), K )
      CALL ORDER ( STATE ( K ), K )
      LSTATE=STATE ( K )
C    WRITE ( 6,43 ) K, LSTATE
C 43  FORMAT ( ///1X, "STAGE= ", I3, "      STATE=" , I4 )
C    WRITE ( 6,34 ) ( LOG ( I ), I=1, LM )
C 34  FORMAT ( /10X, 5I10, "      MAX.VALUE" )
      DO 100 J=1, LSTATE
      RDIM ( J, K )=LDIM-TAPER * [ TLOG ( J, K )+K * 6 ]

```



```

DO 60 I=1,LM
C*** CALL CORECT ( RDIM1, RDIM ( J,K ), LOG ( I ) )
CALL CORECT ( RDIM1, RDIM ( J,K ), LOG ( I ) )
VOL=RDIM1 * RDIM1 * 0.79 * LOG ( I ) / 1000000
DUM3=PRICE ( LOG ( I ), RDIM ( J,K ) )
DIF=TLOG ( J,K ) - LOG ( I )
IF ( ( DIF .GT. TLOG ( STATE ( K1 ), K1 ) ) .OR. ( DIF .LT. TLOG ( 1,K1 ) ) )
*
GO TO 50
C*** CALL RELATE ( DIF,L,K1,STATE ( K1 ) )
CALL RELATE ( DIF,L,K1,STATE ( K1 ) )
VALUE ( I,J,K ) = DUM3 * VOL + MAXVAL ( L,K1 )
GO TO 60
50 VALUE ( I,J,K ) = 0.
C 63 WRITE ( 6,64 ) LOG ( I ), RDIM ( J,K ), VOL,DUM3,MAXVAL ( L,K1 )
C 64 FORMAT ( 1X,16.5X,F6.2,F10.7,2F8.0 )
60 CONTINUE
C*** CALL MAX ( J,K,LM )
CALL MAX ( J,K,LM )
C WRITE ( 6,61 ) TLOG ( J,K ), ( VALUE ( I,J,K ), I=1,LM ),
C * MAXVAL ( J,K )
C 61 FORMAT ( 1X,14.6X,10F10.0 / ( 10X,10F10.0 ) )
100 CONTINUE
K=K+1
GO TO 200
201 ITER=K1
WRITE ( 6,202 ) ITER
202 FORMAT ( /5X, " REAL ITERATION=" ,I4 )
C*** CALL MAXLST ( MAXL,INDLST,INDST,ITER,STATE )
CALL MAXLST ( MAXL,INDLST,INDST,ITER,STATE )
I=IND ( INDLST,INDST )
DECS ( INDST ) = LOG ( I )
INDST1=INDST-1
L=INDLST
DO 300 K=1,INDST1
IK=INDST1-K+1
DIF=TLOG ( L,IK+1 ) - LOG ( I )

```

```
C*** CALL RELATE ( DIF , L , IK , STATE ( IK ) )
      CALL RELATE ( DIF , L , IK , STATE ( IK ) )
      I = IND ( L , IK )
      DECS ( IK ) = LOG ( I )
300 CONTINUE
      WRITE ( 6 , 305 ) MAXL
305 FORMAT ( / 1X , " OPTIMAL SOLUTION = " , F10.0 )
      WRITE ( 6 , 310 ) TIM , LDIM , ( DECS ( IK ) , IK = 1 , INDST )
310 FORMAT ( 1X , " LENGTH = " , 15.5X , " DIAMETER = " , 15.5X , " LENGTH OF
* LOGS : " , 10 I 6 )
1000 CONTINUE
      STOP
      END
C*** SUBROUTINE MAX ( J , K , LM )
      SUBROUTINE MAX ( J , K , LM )
      INTEGER DIM
      REAL MAXVAL
      COMMON MAXVAL ( 150 , 150 ) , LOG ( 50 ) , DIM ( 50 )
      COMMON / A / VALUE ( 50 , 150 , 150 ) , IND ( 150 , 150 )
      MAXVAL ( J , K ) = VALUE ( 1 , J , K )
      IND ( J , K ) = 1
      DO 100 I = 2 , LM
      IF ( VALUE ( I , J , K ) .LE. MAXVAL ( J , K ) ) GO TO 100
      MAXVAL ( J , K ) = VALUE ( I , J , K )
      IND ( J , K ) = I
100 CONTINUE
      RETURN
      END
C*** SUBROUTINE MAXLST ( MAX , INDLST , INDST , ITER , STATE )
      SUBROUTINE MAXLST ( MAX , INDLST , INDST , ITER , STATE )
      INTEGER DIM , STATE ( 150 )
      REAL MAXVAL , MAX
      COMMON MAXVAL ( 150 , 150 ) , LOG ( 50 ) , DIM ( 50 )
      MAX = MAXVAL ( 1 , 1 )
      DO 110 K = 1 , ITER
      LSTATE = STATE ( K )
```

```

DO 100 J=1,LSTATE
IF (MAX .GE. MAXVAL (J,K)) GO TO 100
MAX=MAXVAL (J,K)
INDLST=J
INDST=K
100 CONTINUE
110 CONTINUE
RETURN
END
C*** FUNCTION PRICE (LOG,RDIM)
FUNCTION PRICE (LOG,RDIM)
INTEGER DN,DIM,DN1
REAL MAXVAL
COMMON MAXVAL (150,150),LEN (50),DIM (50)
COMMON /B/LM,DN,PRICE1 (50,50)
DO 100 M=1,LM
IF (LOG .EQ. LEN (M)) GO TO 101
100 CONTINUE
101 LENIND=M
DN1=DN-1
DO 200 N=1,DN1
IF (RDIM .GE. DIM (N) .AND. RDIM .LT. DIM (N+1)) GO TO 201
200 CONTINUE
201 DIMIND=N
IF (RDIM .GE. DIM (DN)) DIMIND=DN
PRICE=PRICE1 (LENIND,DIMIND)
RETURN
END
C*** SUBROUTINE ORDER (STATE,K)
SUBROUTINE ORDER (STATE,K)
INTEGER STATE,TLOG
COMMON /C/TLOG (150,150)
DO 110 I=2,STATE
DO 100 JJ=2,STATE
IF (TLOG (JJ,K) .GE. TLOG (JJ-1,K)) GO TO 100
TEMP=TLOG (JJ,K)

```

```
TLOG (JJ , K ) = TLOG ( JJ - 1 , K )
TLOG ( JJ - 1 , K ) = TEMP
100 CONTINUE
110 CONTINUE
    JJJ = 1
    DO 200 JJ = 2 , STATE
        IF ( TLOG ( JJJ , K ) .EQ. TLOG ( JJ , K ) ) GO TO 200
        JJJ = JJJ + 1
        TLOG ( JJJ , K ) = TLOG ( JJ , K )
200 CONTINUE
    STATE = JJJ
    RETURN
END
C*** SUBROUTINE RELATE ( DIF , L , K1 , STATE )
SUBROUTINE RELATE ( DIF , L , K1 , STATE )
    INTEGER STATE , TLOG , DIF
    COMMON /C/ TLOG ( 150 , 150 )
    DO 100 J = 1 , STATE
        IF ( DIF .EQ. TLOG ( J , K1 ) ) GO TO 110
100 CONTINUE
110 L = J
    RETURN
END
C*** SUBROUTINE CORECT ( RDIM1 , RDIM , LOG )
SUBROUTINE CORECT ( RDIM1 , RDIM , LOG )
    IF ( RDIM .LT. 40 . ) GO TO 111
    RDIM1 = RDIM + LOG * 1.6 / 100 .
    GO TO 121
111 RDIM1 = RDIM + LOG * 0.7 / 100 .
121 RETURN
END
/*
//LKED.SYSPRINT DD DUMMY
//GO.FT05F001 DD *
36  20  40  60  80  100  120  140  160  180  200  220  240  260
    280 300 320 340 360 380 400 420 440 460 480 500 520
```


0	0	0	0	0	0	0
0	0	0	0	2880	2880	2880

10

2200	38	3
2200	42	3
2200	44	3
2400	20	3
2400	24	3
2400	28	3
2400	32	3
2400	36	3
2400	40	3
2400	44	3

/*

//GO.FT06F001 DD SYSOUT=*

//