

昇降機 待期시스템의 시뮬레이션 모델***

(A Simulation Model for the Elevator Queueing System)

吳 炯 在*

閔 殷 基**

Abstract

Among the various types of waiting line systems, the elevator servicing system is quite different from the usual queueing system in view of the service characteristics. For example, the FIFO discipline is not always valid depending upon the situation when the direction of first-come customer's is opposite of the operating elevator direction and at that time a later-arrived one has a luck to be served first.

In this paper, a simulation model is constructed and tested by the sample data and the results have turned out to be fairly adequate. This model, therefore, will provide a good guide to anyone who is interested in the decision of optimal location selection of no-passenger elevator in high buildings whatsoever.

This model is also available, with slight modification, to the problem of city bus dispatching or any other waiting line problems where the servicing equipments are moving.

I. 序 論

昇降機는 서비스 시스템面에서 一般的인待期行列과 다음 몇가지 點에서 相異하다.

첫째는 FIFO(先入先出)에 依한 서비스가 아니라는 點이다. 卽 서비스 해주는 昇降機가 待期하는 顧客이 願하는 方向과 같을 때만이 顧客은 서비스를 받을 수 있고 그렇지 못한때는 뒤에 到着한 顧客이 먼저 서비스를 받는 境遇가 생길 수 있기 때문이다.

둘째는 서비스 能力이 隨時로 바뀐다는 點이다. 一般的인 待期行列에서는 서비스 能力이 처음부터 주어지고 顧客 혹은 서비스를 待期하는 entity는 可用한 채널을 찾아 進入해 들어가 一定한 時間을 經過하게 되지만 昇降機의 境遇는 層別로 物體가 移動할 때 마다 能力(capacity)이 可變的이다.

셋째는 昇降機는 한꺼번에 많은 顧客을 處理하는 bulk servicing system이라는 點이다.

* 서울市立大學

** 成均館大學校

*** 본 논문은 1984 년도 서울시 학술조성비에 의해 연구 되었음.

그러나 顧客은 實際로 bulk arrival을 하지 않는다.

본 研究에서는 위에서 列擧한 몇가지 特徵을 갖는 昇降機 시스템을 電算에 의해 模擬하여 各層別 利用者의 待期時間을 定量的으로 算出하고 아울러 서비스時間, 昇降機의 速度, 目的層의 選好分布, 利用者의 到着率, 서비스能力等의 變數를 如何히 變化시키더라도 顯하는 結果를 도출할 수 있도록 모델을 開發하였다.

2. 昇降機의 서비스原則

昇降機는 一旦 한 方向으로 進行하면 顧客은 到着順序에 關係없이 進行方向을 바꾸지 않고 繼續運行할 必要가 없어질때까지 運行을 繼續하다가 그 必要가 없어지면 進行方向을 바꾼다.

예를 들면 첫번째 顧客이 1層에서 7層까지 가는 途中 6層에서 한 顧客이 下向버튼을 누르고 그다음 9層에서 또 다른 顧客이 下向버튼을 눌렀다. 하자, 이때 7層에서 顧客을 내려 놓은 昇降機가 먼저 到着한 6層의 顧客을 위해 7層에서 方向을 轉換하는 것이 아니라 進行方向을 繼續하여 9層까지 올라간 후 方向을 轉換하여 9層 顧客을 태운다는 意味이다.

그리고 bulk service를 위해 한 昇降機가 어느 層에서 문을 열었을때 그 層에서 待期하는 고객이 있으면 昇降機의 能力의 限度內에서 모두 태운다.

3. 모델 構成의 基本概念

가. 시스템 構成

- (1) 昇降機 1台만을 고려한다.
- (2) 待期費用은 待期時間에 比例하는 것으로 看做한다.
- (3) 서비스費用은 電氣使用料가 主宗을 이룬다고 본다.
- (4) 入力資料로서는 昇降機의 定員, 各層別

顧客 到着率, 各層別 到着顧客의 다른 層으로의 移動確率 等이다.

나. 到 着

(1) 層別 到着은 포아송分布를 따른다. 이를 위하여 昇降機는 午前9時30分부터 11時30分이내에서 運行되는 境遇만 考慮하였다.

(2) 層別 bulk arrival은 發生하지 않는다고 假定한다.

다. 入力資料

本모델에서 使用되는 入力變數는 다음과 같다.

(1) 待機화일 構成時 入力變數

- 建物層數
- 昇降機 運行制限時間
- 各層別 顧客의 到着間 時間
- 各層別 到着顧客의 移動確率

(2) 昇降機 運行時 入力變數

- 建物層數
- 無顧客時 昇降機 位置
- 昇降機 定員
- 1層間 運行時間
- 昇降機의 顧客乘·下車를 위한 停車時間

라. 出力資料

出力資料中 重要한 資料만 列擧하면 다음과 같다.

- 無顧客 運行時間
- 昇降機 輸送時間
- 停止時間
- 顧客의 待期時間

4. 모델의 記述

모델은 2個의 副프로그램과 메인프로그램으로 構成되었다.

REVOLVE'RTN 副프로그램에서는 顧客이 願하는 目標層을 向해 段階的 移動을 함에 따라 發生하는 昇降機의 開閉時間과 이 時間이 經過하는 동안 다른 層에서 서비스를 기다리는 顧客의 기다림시간에 이 時間을 더하는 問題, 昇降機의 門이 열리고 顧客을 서비스 할 때에도 顧客이 願하는 方向과 昇降機의 進行方向이 맞지 않을 때는 이 方向과 一致하는 顧客이 待期行例에 있으면 이를 選別하는 問題 그리고 昇降機의 輸送時間등을 계산하는 問題를 다루고 있다.

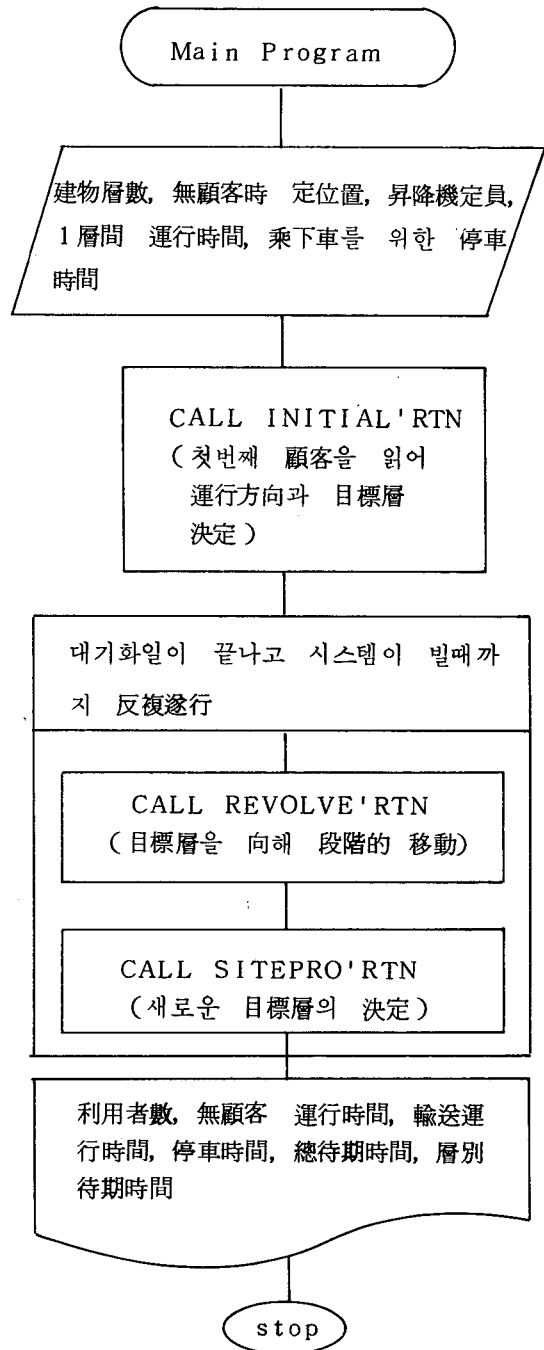
SITEPRO'RTN 副프로그램은 한 層에서 서비스 作業이 끝나고 그 다음 進行方向 및 目的層을 어디까지로 잡을 것인가를 決定하는 프로그램이다.

即 어느 한 層에서 昇降機는 서비스를 마쳤으나 昇降機의 進行方向으로 더 運行을 繼續하기 爲해서는 그 進行方向에서 顧客이 待期하고 있는가를 살펴보아야 한다. 그리고 그 以上の 進行方向에 顧客이 없으면 비로소 昇降機는 方向을 轉換할 수가 있다. 方向을 轉換한 다음에는 어느 層으로 移動할 것인가를 곧 決定해야 하며 서비스 도중 受容能力이 超過되면 昇降機 內의 顧客이 願하는 層까지 논·스톱으로 進行한다.

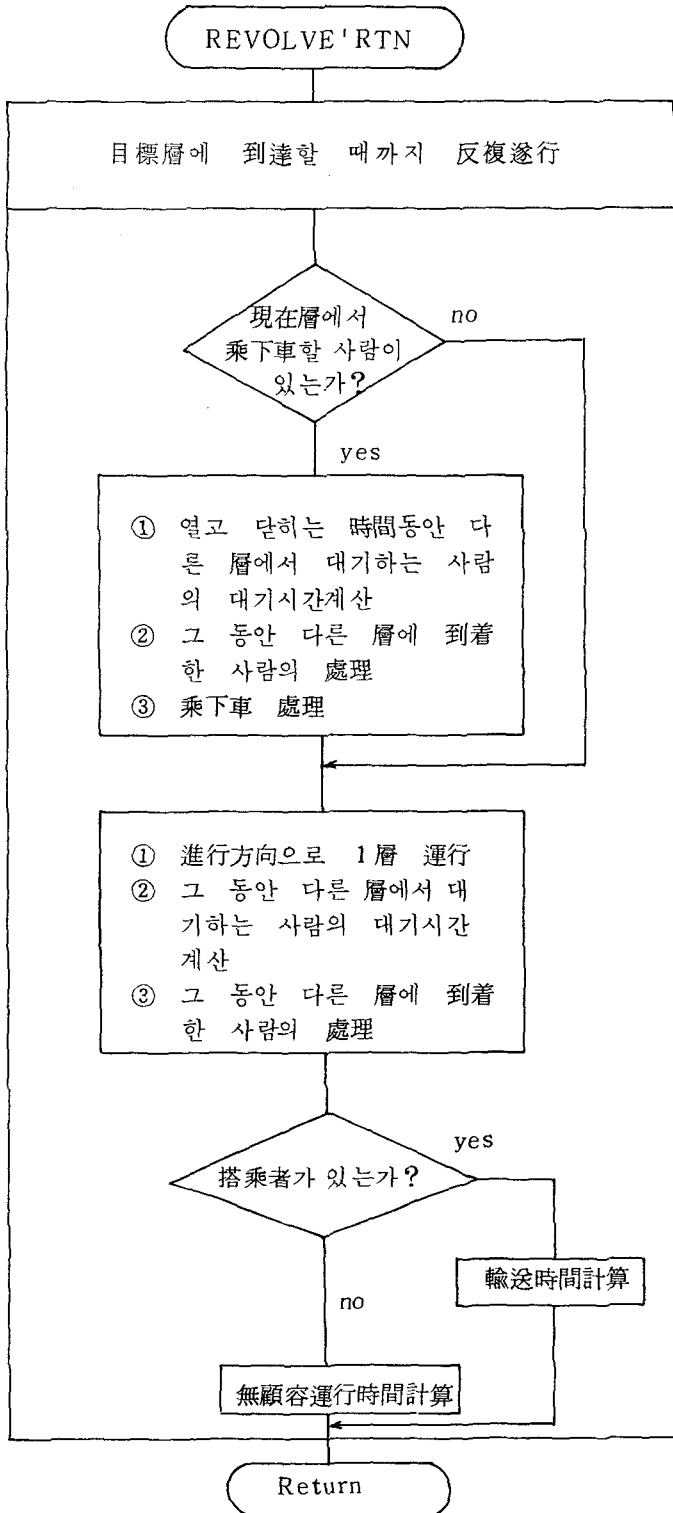
以上 두 個의 副프로그램을 메인으로 묶어 構成한 모델을 불럭 圖表로 表示하면 다음과 같다.(本모델을 위한 source program : 附錄參照).

5. 모델 테스트

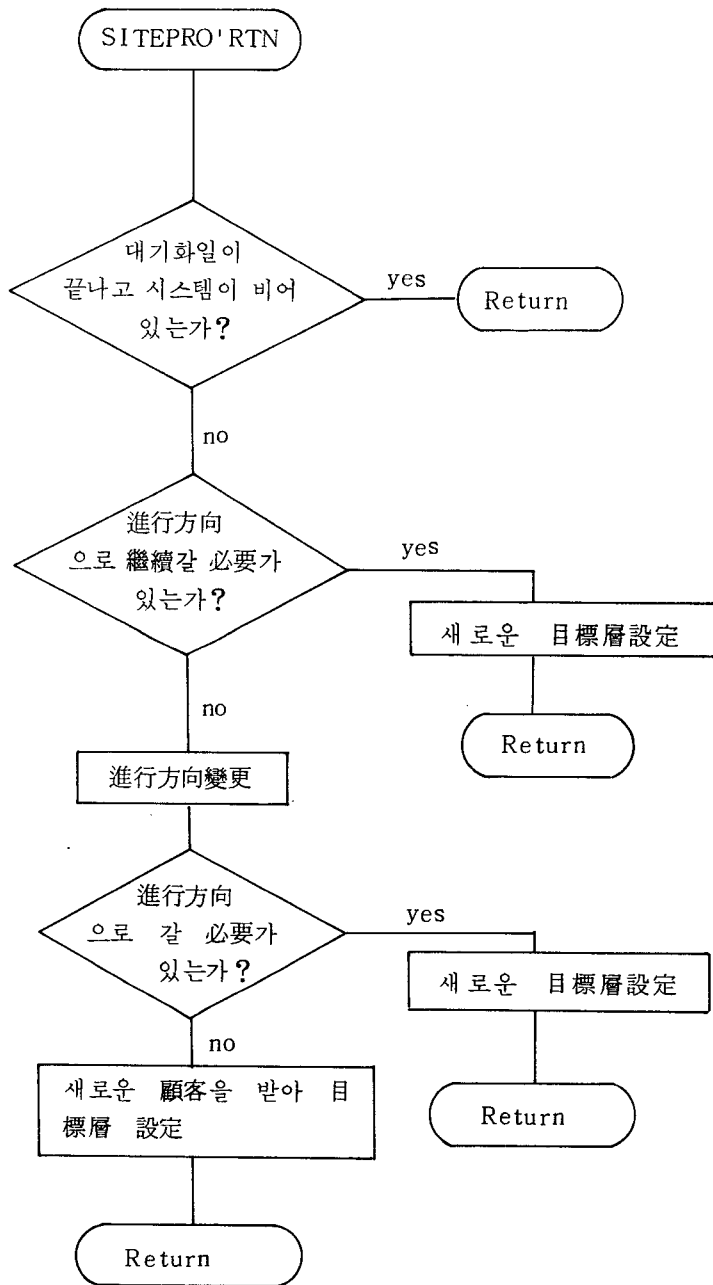
本모델의 結果에 대한 信賴性檢證을 위해 샘플 資料를 入力시키되 모델을 通해 算出된 結果는 一般的 常識과 一致하도록 入力資料를 作成하였다.



<도표1> 메인 프로그램



〈도표 2〉 REVOLVE'RTN 부프로그램



〈 도 표 3 〉 SITEPRO'RTN 부프로그램

가. 待期확일構成에 使用된 入力

- 建物層數 : 9
- 昇降機運行制限時間 : 10,000, 12,000, 14,000, 16,000 초
- 到着間時間 : 100, 120, 140, 160 초
- 各層別到着顧客의 移動確率 : 一樣分布

- 最終位置 : 1 層
- 昇降機定員 : 5 名, 10 名
- 1 層間運行時間 : 2 초
- 顧客乘下車時間 : 5 초

나. 昇降機 運行을 위한 入力資料

다. 테스트結果

以上과 같은 資料를 入力시킨 後 모델을 通한 結果를 보이면 다음과 같다.

表 1. 샘플資料에 대한 算出結果

到着間時間 (秒)	100		120		140		160	
	定員		定員		定員		定員	
各種時間(秒)	5	10	5	10	5	10	5	10
빈 運 行 時 間	2,686	2,686	3,572		3,930		4,404	
輸 送 時 間	5,460	5,460	5,836	左 同	6,140	左 同	6,396	左 同
停 止 時 間	3,403	3,399	7,864		12,243		12,992	
待 期 時 間	35,728	35,618	31,203		28,480		26,784	

表 1에 나타난 結果를 그림 1에 圖式化하였다.

그림 1을 보면 到着間時間이 길어질수록 待期時間이 減少하는 현상과 빈 運行時間이 增加하는 데는 다음과 같은 理由가 있다고 보여진다.

輸送時間의 完만한 增加는 到着率이 낮으면 昇降機가 거의 層마다 한 사람씩 태우게 되어 昇降機의 停止時間이 增加하고 따라서 輸送時間도 多少 더 所要되는 것으로 解析된다. 反對로 到着率이 높으면 그 만큼 每層마다 거의 한 각 以上을 태우게 되어 相對的으로 輸送에 所要되는 時間을 다소 減少시킨다.

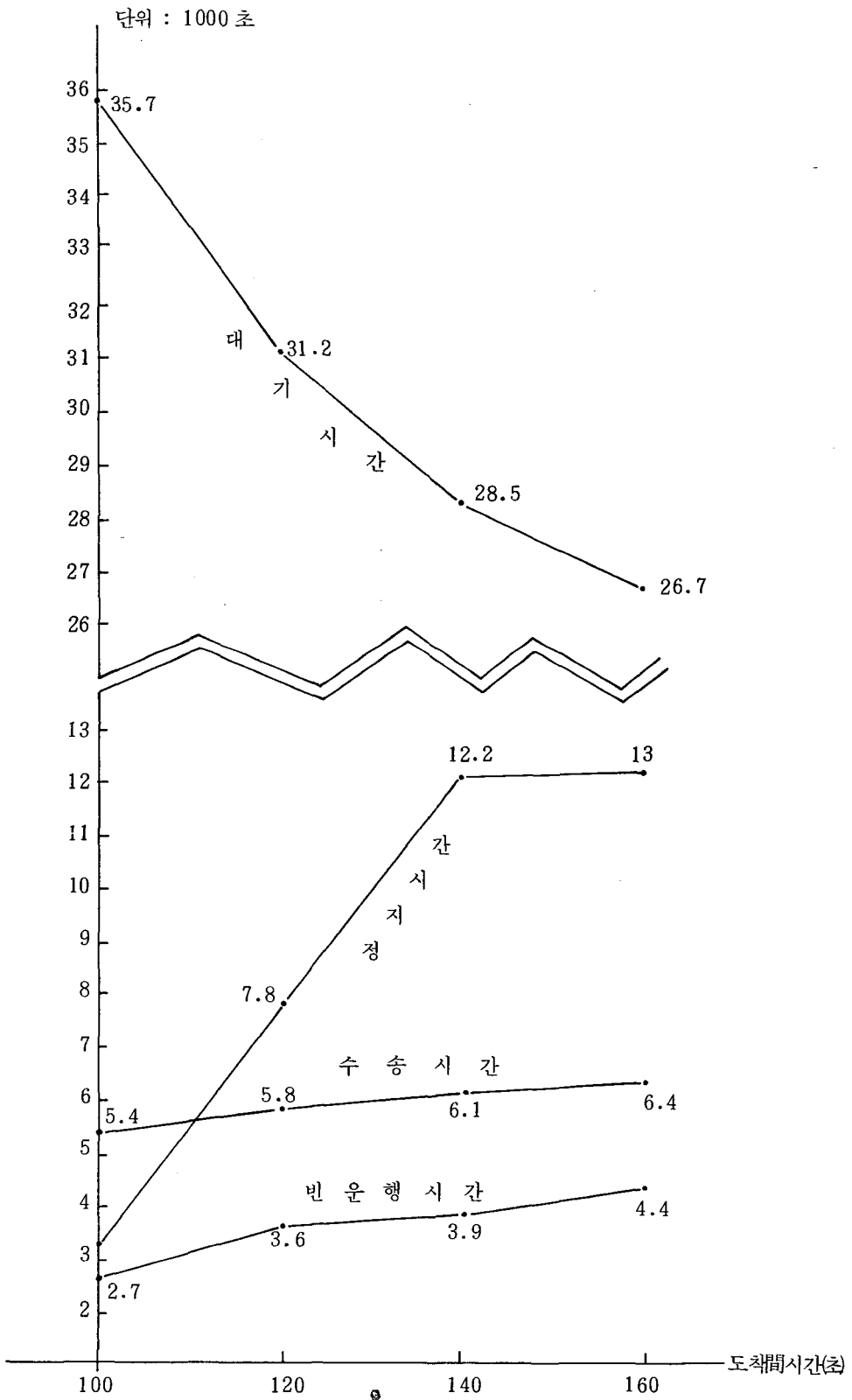
그림 1에서 한가지 흥미로운 것은 到着間時間이 100 초에서 140 초로 증가하는 동안 昇降機의 停止時間이 거의 4 배에 가까운 增加를 보인다는 事實이다. 이는 人員數 5 名을 태우

는 昇降機가 顧客만 있으면 每層마다 바이패스 하지 않고 停車하게 되기 때문일 것이며 이러한 現象은 “一顧客當一停車”의 루율이 이루어지기 始作하는 到着率까지 奏効하고 그 以上일 때는 매우 完만한 停車時間의 增加를 보인다. 이 경우도 승강기의 定員을 줄이면 待ち시간이 다시 減少하는 것은 當然한 일이다.

다음은 以上 4 가지 出力의 하나 하나를 層別로 그리고 到着率別로 보이기로 한다. (圖表 4 - 圖表 7 參照).

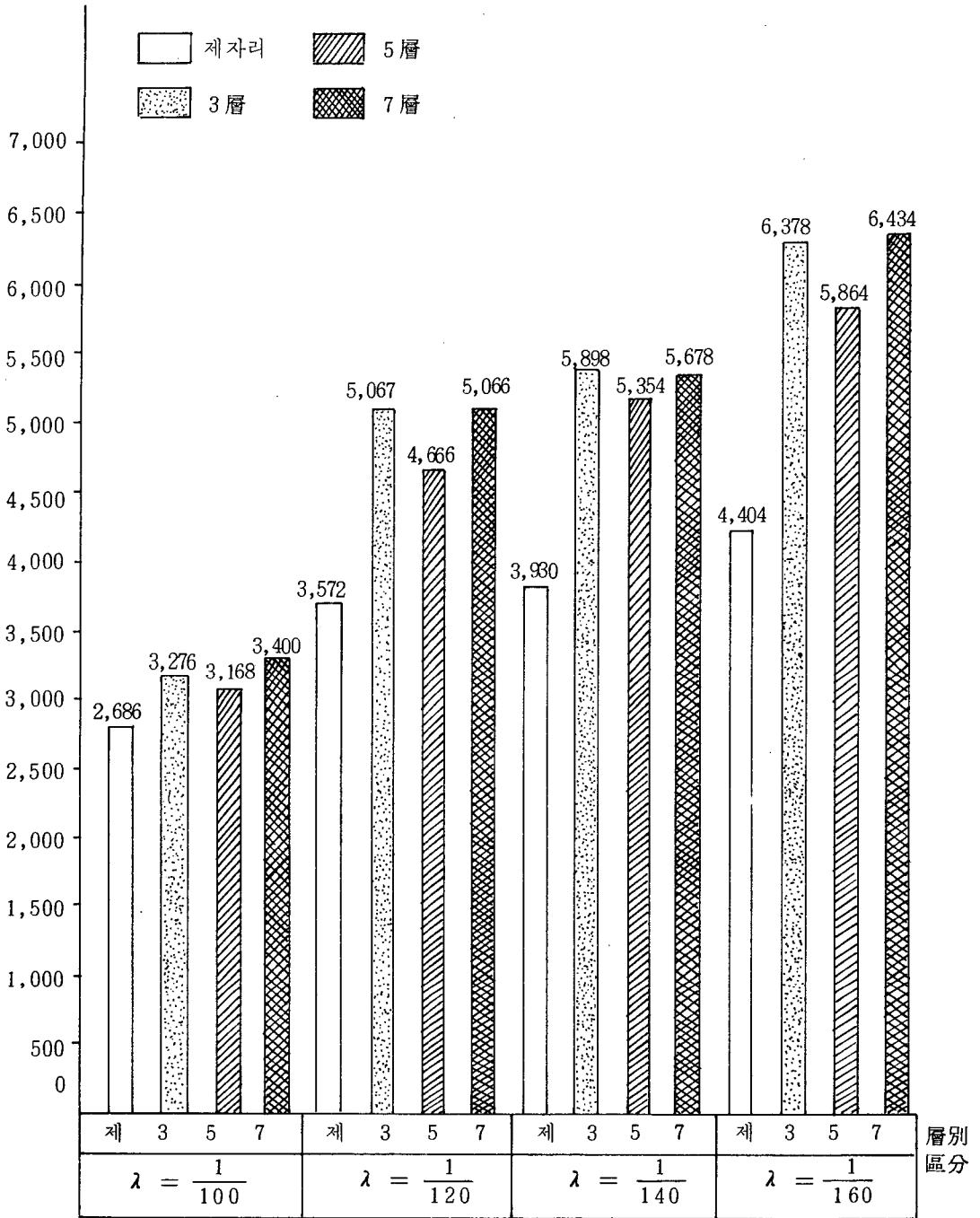
圖表-4는 到着率別, 顧客의 層別 無顧客 昇降機 運行時間을 보인 것이다. 圖表에서 제 자리라는 말은 昇降機가 서비스를 마쳤을때 그 瞬間以後 서비스를 받고자 하는 顧客이 없으면 昇降機를 現位置에 그대로 머무르게 한다는 말이다.

그 境遇 $\lambda = \frac{1}{100}$ 인 때는 現位置에 昇降機



<그림 1> 산출결과의 도식화

無顧客
運行時間(抄)



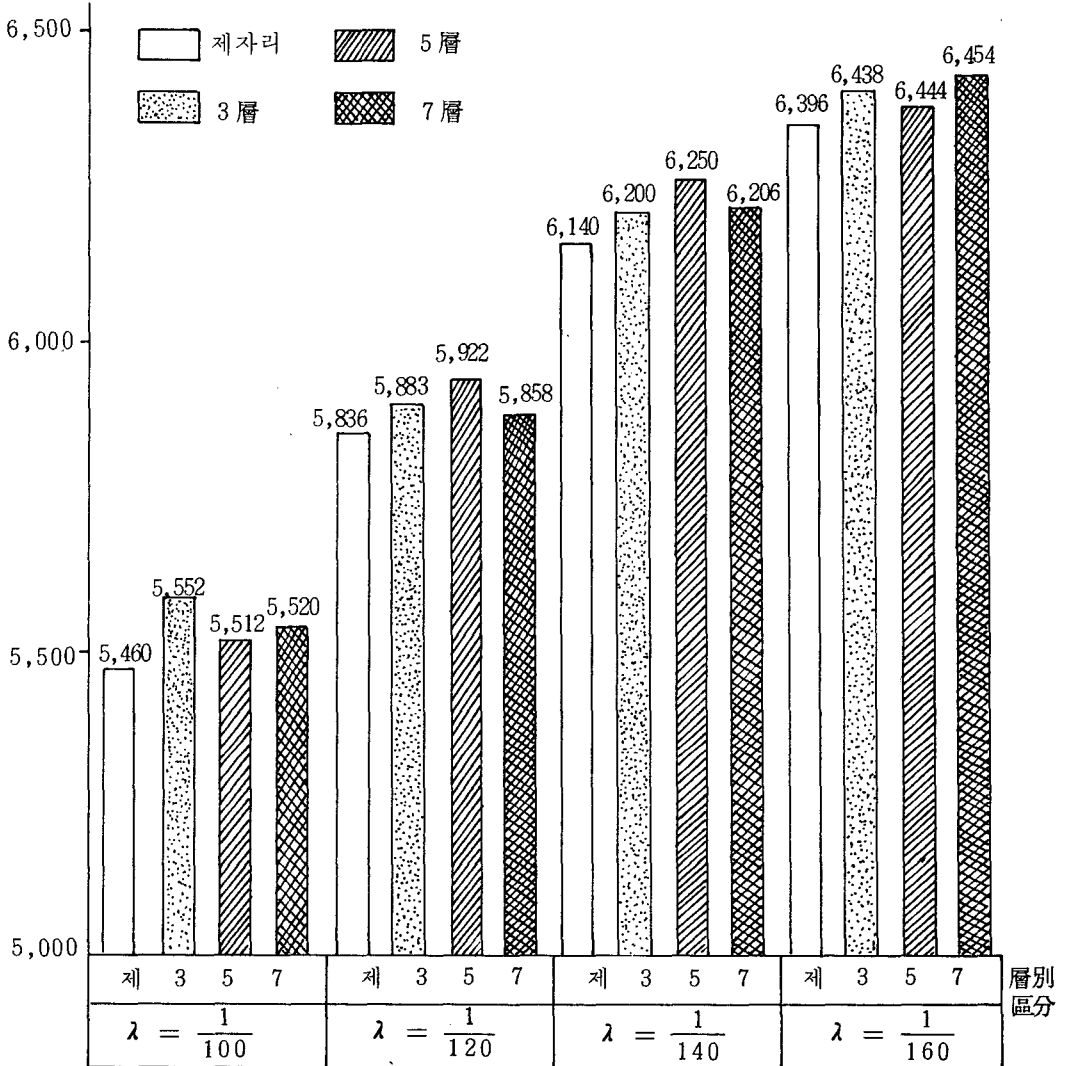
<圖表-4> 無顧客 運行時間

가 그대로 머물러 있건, 3層, 5層 혹은 7層으로 옮겨놓건 昇降機의 無顧客 運行時間에는 $\lambda = \frac{1}{120}$ 以下인 境遇에 比하여 큰 差異가 없다는 事實이다. 그 理由는 顧客이 빈번하게 到着하기 때문에 無顧客時 特定層에 갖다 놓을 틈이 없기 때문이다. 이러한 效果는 $\lambda = \frac{1}{100}$ 일

때 더 현저하게 나타나고 있다.

그러나 $\lambda = \frac{1}{120}$ 以下인 세가지 경우는 3層, 5層, 7層 및 제자리中 제자리에 그대로 둘 때 無顧客 運行時間이 제일적이고 그 다음은 5層 即 “中間層”임을 알수 있다.

顧客輸送時間 (抄)

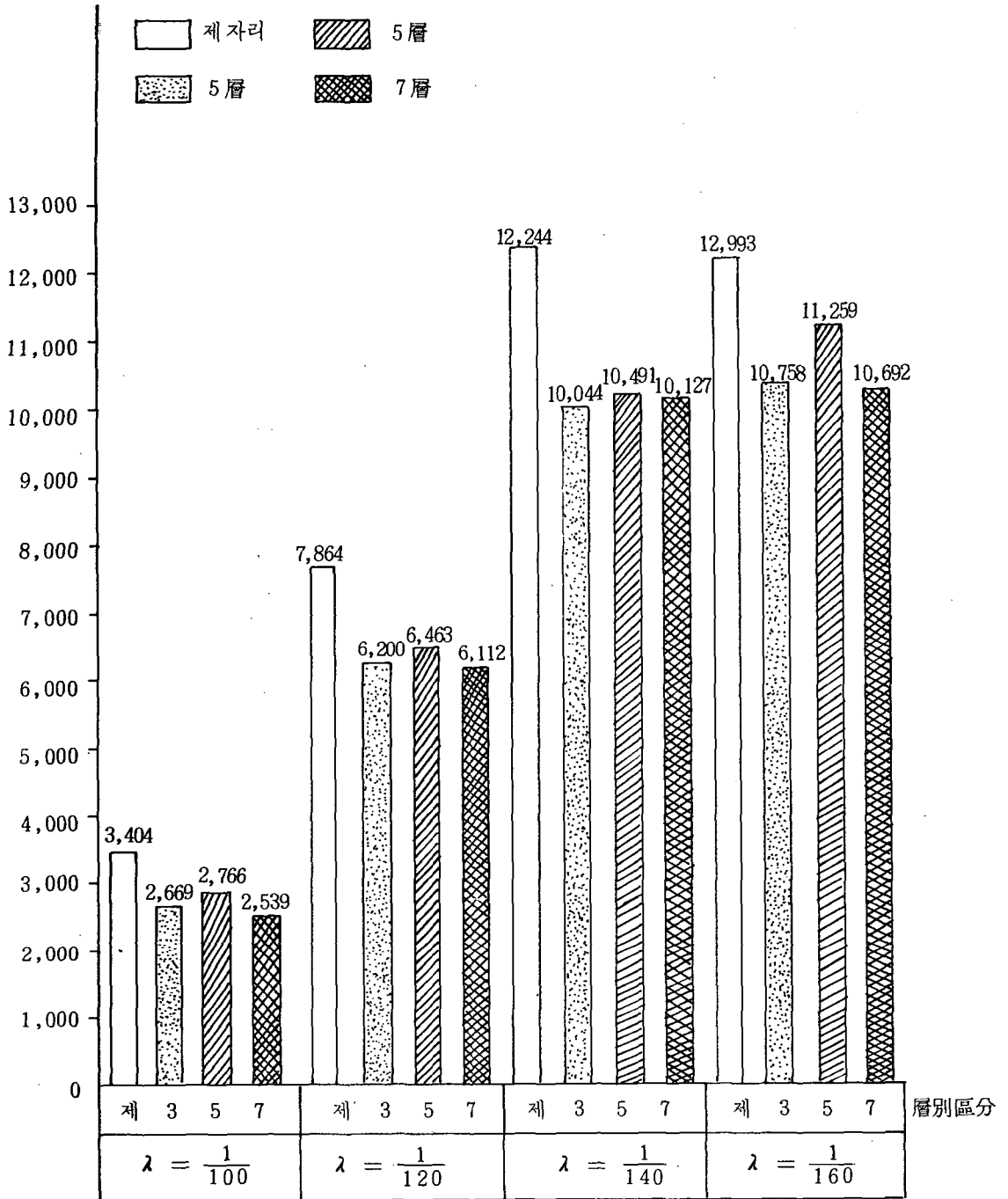


< 圖表 - 5 > 顧客 輸送 時間

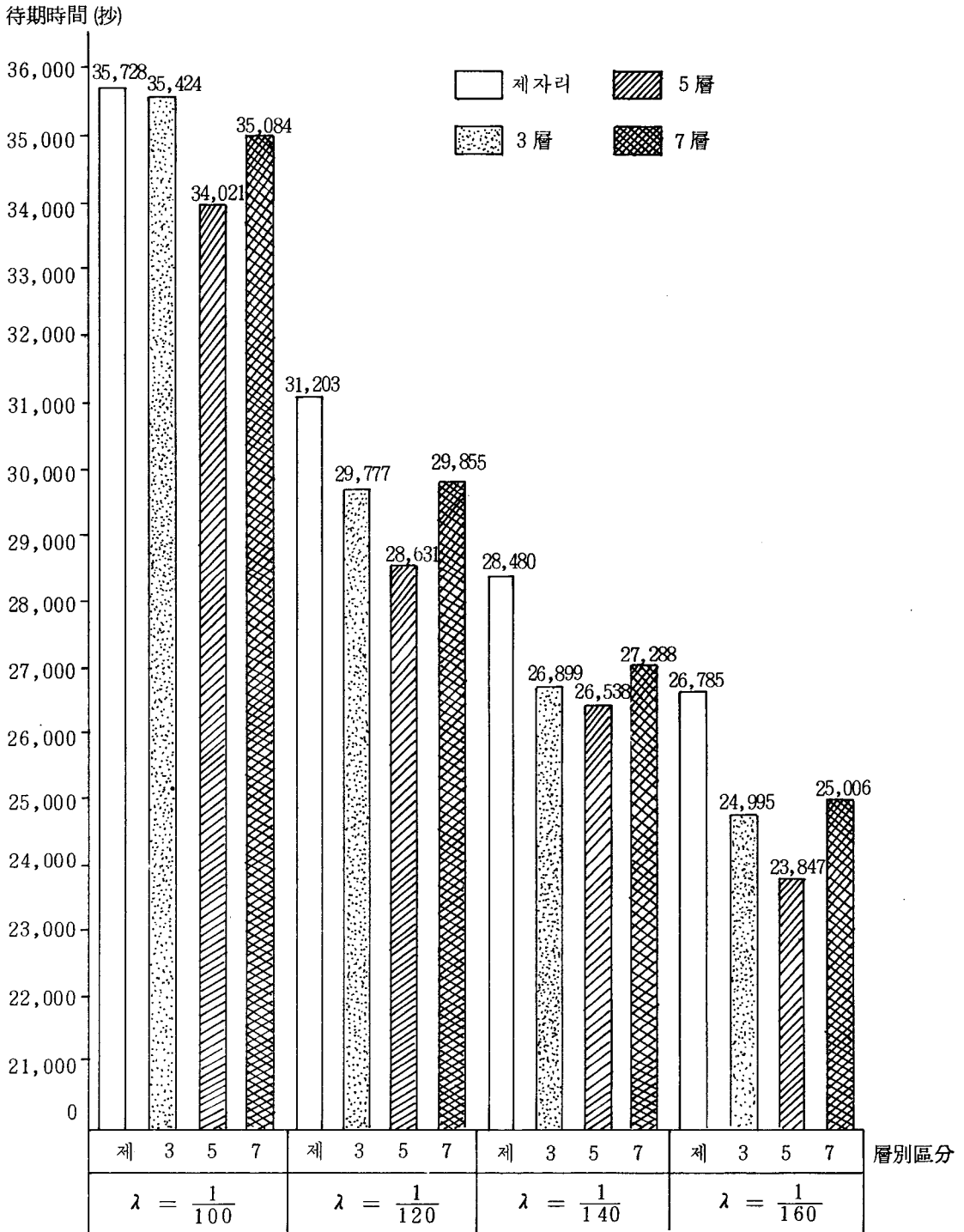
圖表-5의 顧客輸送時間에서는 한결같이 無顧客時 昇降機를 「停止가 發生한 現位置」에 그대로 두는 것이 제일 輸送時間을 最少化 하

는 方法임을 말해준다. 輸送時間을 最大化하면 그만큼 서비스費用(主로 電氣使用料)을 節減하는 原因이 된다.

停車時間(秒)



<圖表-6> 昇降機 停車時間



〈圖表-7〉 顧客의 待期時間

圖表-6의 昇降機 停車時間은 제자리인 境遇가 많으나 이 效果는 到着率이 큰 경우와 ($\lambda = \frac{1}{100}$) 적은 경우($\lambda = \frac{1}{160}$)에는 큰 差異가 없다는 結論이다.

問題는 圖表-7의 顧客의 待期時間이다. 昇降機를 無顧客時 現位置에 두는 것은 위의 세 가지의 경우에 매우 만족스러우나 顧客의 待期時間을 생각하면 不合格이다.

이 때는 5層이 제일 좋다.

그러나 意思決定者는 運行時間과 이에 比例하여 發生하는 서비스費用, 그리고 待期時間과의 trade-off 分析을 通하여 政策을 決定하여야 할 것이다.

6. 使用者를 위한 案內

모델 테스트時 사용한 入力資料에서 얻은 컴퓨터 結果值는 우리의 直觀의 理解와 잘 一致하고 있어 이 結果는 시뮬레이션모델의 信賴性을 잘 立證해 주고 있다. 그러므로 使用者는 이제 모델에 現實에 맞는 入力資料를 作成하여 入力만 시키면 된다.

入力は 크게 둘로 나누어 待期화일 構成과 昇降機 運用이며 이를 샘플로 보이면 다음과 같다.

가. 待期화일의 構成 資料

DATA(建物層數); (運行制限時間)

DATA(層別到着率), ..., (層別到着率)

DATA(0), (N₁₂), ..., (N_{1n})

}
(N_{n1}), (N_{n2}), ..., (0)

(但, N_{ij} : i層到着顧客總人員中 j層에 가기를 願하는 人員)

나. 昇降機 運用資料

DATA(建物層數), (머무르게 할 位置層),
(定員)

DATA(一個層運行時間), (乘下車時間)

(더 자세한 것은 附錄의 프로그램 參照).

7. 結 論

本 研究는 서비스 루율이 特殊한 高層建物에서 昇降機의 效率인 運營을 위해 顧客의 到着率, 顧客이 願하는 目的層分布, 昇降機의 運行時間에 比例하는 서비스費用, 顧客의 待期費用, 昇降機의 受容能力等を 自由自製로 變化시키면서 우리가 願하는 出力資料(層別顧客의 平均待期時間, 總서비스費用, 無顧客時 昇降機의 最適位置層等)를 얻을 수 있는 시뮬레이션 모델을 개발한 것이다.

그러나 이 모델은 버스會社에서 버스를 配車(dispatching)할때도 큰 修正없이 使用可能하다고 思料된다. 왜냐하면 高層建物을 “땅에 놓으면” 昇降機는 버스가 되고 層別顧客은 버스停留場에 車를 타러 나오는 손님에 해당되기 때문이다.

本 研究에서의 最大 弱點은 昇降機를 한 臺로 局限시킨 點이다. 이를 더욱 發展시켜 n臺로 擴張할 必要를 느낀다.

또한 本 研究에서는 서비스와 待期費用을 算出하는 模型을 提示하지 않고 單純히 서비스費用은 昇降機의 運行時間에, 그리고 待期費用은 待期時間에 比例한다고 하면서 費用算出에 對해 暗示的(implicit) 方法論만 제시한 느낌이나 이 分野도 계속 研究課題로 思料된다.

參 考 文 獻

1. Geoffrey Gordon, 'System Simulation', Prentice-Hall Inc., 1969.
2. Leonard Kleinrock, "Queueing Systems", Vol. 1, Vol. 2, John Wiley & Sons, 1975.
3. 왕창중, "BAS I C 프로그래밍 연습", 기술연구소, 1984.

附錄: 프로그램 리스트 ("명필 - 2"에 의해 수행)

1. 대기 화일 생성 프로그램

>>list

```
10 Rem ===== QUEUE, BAS =====
20 On Esc Goto Last
30 Integer I, J, K, P, P1, P2, P3
40 Integer Build, Recd, Inwon
50 Integer Q, St, Co, Le, Ri, Po
60 Short Time, Clck, Arrv, Lam
70 Read Build, Inwon
80 Dim Lamda(Build)
90 For I=1 To Build
100 Read Lamda(I)
110 Next I
120 Dim Proba(Build, Build)
130 For I=1 To Build
140 For J=1 To Build
150 Read Proba(I, J)
160 Proba(I, 0)=Proba(I, 0)+Proba(I, J)
170 Next J
180 Next I
190 P3=0
200 For I=1 To Build
210 Mans=0
220 For J=1 To Build
230 Mans=Mans+Proba(I, J)
240 Proba(I, J)=Mans/Proba(I, 0)
250 Next J
260 Next I
270 Open#2, 14W "Q-150-00, fl "
280 Get#2, 0WP, Recd
290 Randomize
300 I=2
310 Rem For I=2 To Build
320 Time=0
330 For K=1 To Inwon
340 Clck=-Lamda(I)*Log(Rnd(1))
350 If Clck=0 Then 340
360 Time=Time+Clck
370 Recd=Recd+1
380 Prbl=Rnd(1)
390 Gosub Goal`rt
```

```

400 PutW2, RecdWTime, I, J, P3, P3, P3
410 Call .Srch'rt (P)
420 PutW2, P1, P2WRecd
430 @ Time, K, I, J
440 Next K
450 Rem Next I
460 *Last
470 PutW2, 0, 2WRecd
480 Close : End
490 *Goal'rt
500 For J=1 To Build
510 If Prbl<=Proba(I, J) Then 530
520 Next J
530 Return
540 Procedure .Srch'rt (Q)
550 GetW2, QWArrv, St, Go, Le, Ri
560 If Time<Arrv Then Do
570 If Le=0 Then Do
580 P1=Q
590 P2=8
600 Else
610 Call .Srch'rt (Le)
620 Enddo
630 Else
640 If Ri=0 Then Do
650 P1=Q
660 P2=10
670 Else
680 Call .Srch'rt (Ri)
690 Enddo
700 Enddo
710 Endproc
720 Data 9, 255
730 Data 150, 150, 150, 150, 150, 150, 150, 150, 150
740 Data 0, 20, 20, 20, 20, 20, 20, 20, 20
750 Data 20, 0, 20, 20, 20, 20, 20, 20, 20
760 Data 20, 20, 0, 20, 20, 20, 20, 20, 20
770 Data 20, 20, 20, 0, 20, 20, 20, 20, 20
780 Data 20, 20, 20, 20, 0, 20, 20, 20, 20
790 Data 20, 20, 20, 20, 20, 0, 20, 20, 20
800 Data 20, 20, 20, 20, 20, 20, 0, 20, 20
810 Data 20, 20, 20, 20, 20, 20, 20, 0, 20
820 Data 20, 20, 20, 20, 20, 20, 20, 20, 0

```

2. 승강기 운행 프로그램

>>list

```

10  Rem =====
20  Rem  무고객 승강기의 위치선정에 관한 연구
30  Rem =====
40  Rem Homef  :  무고객 승강기의 원위치
50  Rem Build  :  승강기 운행건물의 층수
60  Rem Speed  :  승강기의 입층 운행시간
70  Rem Stage  :  승하차를 위한 정차시간
80  Rem State  :  승강기에 대한 요구상태
90  Rem Leave  :  탑승고객의 하차층 기억
100 Rem Maxps  :  승강기의 최대 탑승인원
110 Rem -----
120 Rem Emptm  :  무고객 승강기 운행시간
130 Rem Trant  :  유고객 승강기 운행시간
140 Rem Waitm  :  승강기 고객의 대기시간
150 Rem Stopt  :  원위치 승강기 정차시간
160 Rem -----
170 Rem Flor   :  승강기의 현재 위치층수
180 Rem Trgt   :  승강기의 다음 목적층수
190 Rem Site   :  승강기의 상하 운행방향
200 Rem Intv   :  운행이나 정차한 시간차
210 Rem Side   :  고객이 원하는 상하방향
220 Rem Clck   :  승강기 고객의 도착시간
230 Rem Strt   :  승강기 고객의 탑승층수
240 Rem Goal   :  승강기 고객의 목적층수
250 Rem Pers   :  승강기 내부의 탑승인원
260 Rem Hold   :  승강기의 정차여부 확인
270 Rem Time   :  승강기 운행시의 타워어
280 Rem Numb   :  승강기 탑승자의 인원수
290 Rem =====
300 Rem  M A I N  P R O G R A M
310 Rem =====
320  Imode  :  X=2.5
330  If X=2.5 Then Run
340  On Esc Goto Last
350  Read Build  :  U=2
360  Read Homef, Maxps
370  Read Speed, Stage
380  Short Clck, Emptm
390  Short Time, Trant
400  Dim Leave(Build)

```

```

410 Short Waitm(Build)
420 Fln$="Q-100.FIL"
430 OpenWU,14WFln$
440 OpenW6W "$LP"
450 Short Stopt
460 Gosub Initial'rtm
470 Repeat
480 Gosub Revolve'rtm
490 Gosub Sitepro'rtm
500 Until P=9999 And Leave(0)≠0 And State(0,0)=0
510 @W6W "===== "
520 @W6W "File Name = ";Fln$
530 @W6W "Home Flor = ";Homef
540 @W6W "The Maximum s Persons = ";Maxps
550 @W6W "The Number of Customer = ";Numb
560 @W6W "Empty Moving Time = ";Emptm
570 @W6W "Transportation Time = ";Trant
580 @W6W "Elevator Stopping Time = ";Stopt
590 @W6W "Customer Waiting Time = ";Waitm(0)
600 @W6W "Average Waiting Time = ";Waitm(0)/Numb
610 @W6W "----- "
620 For I=1 To Build
630 @W6W " ";I;" Step Waiting Time = ";Waitm(I)
640 Next I
650 @W6W "===== "
660 *Last : Close : End
670 *Initial'rtm : Rem(승강기 운행의 초기작업)
680 Dim State(Build, Build)
690 GetWU, PWA, B, C, D, E, F, P
700 @ A, B, C, D, E, F, P
710 GetWU, PWClck, Strt, Goal, E, F, P
720 Flor=Homef : Numb=0
730 Gosub Stat'rt
740 Gosub Site'rt
750 GetWU, PWClck, Strt, Goal, E, F, P
760 Return
770 *Revolve'rtm : Rem(목표층을 향해 단계이동)
780 Repeat
790 Gosub Opencls'rtm
800 If Hold=1 Then Do
810 Intv=Stage
820 Time=Time+Intv
830 Gosub Waiting'rtm
840 Gosub Arrival'rtm
850 Gosub Goinout'rtm

```



```

860      Enddo
870      Intv=Speed
880      Flor=Flor+Site
890      Time=Time+Intv
900      If Pers=0 Then Do
910          Emptm=Emptm+Intv
920          Else
930              Trant=Trant+Intv
940          Enddo
950      Gosub Waiting'rtm
960      Gosub Arrival'rtm
970      Until Flor=Trgt
980      Return
990 *Sitepro'rtm : Rem<운행방향 변경여부 결정>
1000  If P=9999 And Leave(0)=0 And State(0,0)=0 Then Return
1010  If Flor=1 Or Flor=Build Then 1040
1020  Gosub Trgt'rt
1030  If Trgt#0 Then Return
1040  Site=-Site
1050  Gosub Trgt'rt
1060  If Trgt#0 Then Return
1070  If Homef#1 And Flor#Homef Then Do
1080      Trgt=Homef
1090      If Flor<Trgt Then Do
1100          Site=1 : Else : Site=-1 : Enddo
1110      Else
1120          Gosub Site'rt
1130          If Trgt=0 Then Gosub Goinout'rtm
1140      Enddo
1150  Return
1160 *Waiting'rtm : Rem<고객 대기시간 계산>
1170  Waitm(0)=Waitm(0)+State(0,0)*Intv
1180  For I=1 To Build
1190      Waitm(I)=Waitm(I)+State(I,0)*Intv
1200      Waitm(I)=Waitm(I)+State(I,I)*Intv
1210  Next I
1220  Return
1230 *Arrival'rtm : Rem<시간내 도착고객의 처리>
1240  If Time<Clck Or P=9999 Then Return
1250  Repeat
1260      Gosub Stat'rt
1270      If Intv=Stage And Side=Site And Flor=Strt Then 1300
1280      Waitm(Strt)=Waitm(Strt)+Time-Clck
1290      Waitm(Zero)=Waitm(Zero)+Time-Clck
1300  Get#U, P#Clck, Strt, Goal, E, F, P

```

```

1310     Until Time<Click Or P=9999
1320     Return
1330 *Opencls 'rtn : Rem(승강기의 정차여부 결정)
1340     Hold=0
1350     If Leave(Flor)#0 Then 1390
1360     If Pers=Maxps Then Return
1370     If Site=1 And State(Flor,Flor)#0 Then 1390
1380     If Site#-1 Or State(Flor,Zero)=0 Then Return
1390     Hold=1 : Return
1400 *Goinout 'rtn : Rem(고객의 하차처리)
1410     If Leave(Flor)#0 Then Do
1420         Mans=Leave(Flor)
1430         Leave(0)=Leave(0)-Mans
1440         Pers=Pers-Mans
1450         Leave(Flor)=Zero
1460     Enddo
1470 *Input 'rtn : Rem(탑승고객의 유무 확인)
1480     If Site=1 And State(Flor,Flor)#0 Then 1500
1490     If Site#-1 Or State(Flor,Zero)=0 Then Return
1500     If Site=1 Then Do
1510         T=Flor : Else : T=0 : Enddo
1520     I=Flor
1530     Repeat
1540         I=I+Site
1550         If State(Flor,I)#0 Then Gosub Proc'ss
1560         Until I=1 Or I=Build
1570     Return
1580 *Site 'rt : Rem(승강기의 운행방향 결정)
1590     If P=9999 Then Return
1600     Stopt=Stopt+Click-Time
1610     Time=Click
1620     If Flor=Strt Then Do
1630         Trgt=Goal
1640         If Strt<Goal Then Do
1650             Site=1 : Else : Site=-1 : Enddo
1660         Else
1670             Trgt=Strt
1680             If Flor<Strt Then Do
1690                 Site=1 : Else : Site=-1 : Enddo
1700         Enddo
1710     Return
1720 *Stat 'rt : Rem(도착고객의 각종 정보기억)
1730     Numb=Numb+1
1740     @ Using "####", Numb:
1750     If Strt<Goal Then Do

```

```

1760     Totl=Strt : Side=1
1770     Else
1780     Totl=Zero : Side=-1
1790     Enddo
1800     State(Strt, Goal)=State(Strt, Goal)+1
1810     State(Strt, Totl)=State(Strt, Totl)+1
1820     State(Zero, Zero)=State(Zero, Zero)+1
1830     Return
1840 *Proc'ss : Rem(고객 탑승처리)
1850     If Pers=Maxps Then Return
1860     If Pers+State(Flor, I)>Maxps Then Do
1870         Mans=Maxps-Pers
1880     Else
1890         Mans=State(Flor, I)
1900     Enddo
1910     Pers=Pers+Mans
1920     Leave(I)=Leave(I)+Mans
1930     Leave(0)=Leave(0)+Mans
1940     State(0, 0)=State(0, 0)-Mans
1950     State(Flor, T)=State(Flor, T)-Mans
1960     State(Flor, I)=State(Flor, I)-Mans
1970     Return
1980 *Trgt'rt : Rem(목표층의 유무확인)
1990     Kk=Flor : Trgt=0
2000     Repeat
2010     Kk=Kk+Site
2020     If Leave(Kk)#0 Or State(Flor, Kk)#0 Then Trgt=Kk
2030     If State(Kk, 0)#0 Or State(Kk, Kk)#0 Then Trgt=Kk
2040     Until Kk=1 Or Kk=Build
2050     Return
2060     Data 9, 3, 10, 2, 5
2070     Rem == END PROGRAM ==

```