

마이크로 컴퓨터용 경영·관리 프로그램 개발**

박 순 달*

1. 序 論

마이크로컴퓨터는 최근들어 급속도로 발전하고 있다. 이제 마이크로컴퓨터는 단순한 게임용을 넘어서 교육용, 기업의 행정 경영의사결정용으로 사용되기에 이르고 있다. 아직 연산속도가 느리고 기억용량이 제한되어 있다는 제약이 따르기는 하지만 대형컴퓨터에만 의존하던 생각을 벗어나 값 싸고 사용이 편리한 마이크로컴퓨터의 활용이 더욱 가속화될 것임이 틀림없다.

이 연구는 이러한 경향에 따라 경영·관리에 기본적으로 필요한 기초모형의 프로그램을 개발해 두고져 하는 것이다.

경영·관리의 기본모형이란 자연적으로 OR(經營科學)의 기본모형이 될 것이며 이 經營科學이 다루는 각종 모형에서 실제적으로 널리 사용되고 있는 모형이 이 연구의 대상이 될 기본모형이다.

이 연구에서 사용될 電算機는

MIGHTY(金星), SPC-1000(三星), APPLE系

로 하였다. APPLE을 선택한 이유는 세계에서 가장 널리 퍼져있는 전산기가 APPLE과 IBM-PC일 것이고 이 연구가 이들 전산기를 대상으로 한다는 것은당연할 것이다. 그러나 IBM-PC는 아직 한국에서

많이 활용되고 있지 않기 때문에 이 연구에서 제외하였다.

그리고 이러한 프로그램의 개발에 사용할 국산 전산기는 기본적으로 디스크를 사용할 수 있어야 하고 64 K RAM을 가진 전산기를 선택하였다. 이들 전산기에 사용할 言語는 BASIC을 사용하였다. BASIC을 사용한 이유는 다른 言語보다 프로그램작성이 쉽고 對話式入出力이 편리하기 때문이다. 프로그램수행속도가 느린 단점이 있지만 꼭 이런 단점을 보완하고 싶다면 컴파일시켜 사용할 수도 있기 때문이다.

2. 模型과 計算方法

經營科學은 새로운 문제가 模型化되고 계략적으로 처리됨에 따라 계속 그 활용범위가 넓어지고 學文으로서의 영역이 넓어지는 것인만큼 어디까지가 經營科學의 영역이라고 명사하기는 힘들다.

그래서 사람에 따라 범위가 조금씩 달라진다. 그러나 여기에서는 가장 보편적으로 받아들이고 있는 범위를 취하기로 한다.

經營科學을 분류할때 크게 確定的模型과 確率的模型으로 분류하는 경우가 많다. 確定的模型은 모든 변수나 係數에 確率(확률)이 들어있지 않는 경우이고 確率的模型에는 確率(확률)이 들어 있는 경우이다. 그러나 이런 경우 確率

* 서울工大

** 본 연구는 한국과학재단의 지원으로 이루어진 것임.

的模型에는 確定的 모형 이외의 모든 模型을 포괄시키는 경우가 많다. 그래서 우리가 가기마련이다. 그래서 여기에서는 좀더 세분하기로 한다.

먼저 確定的模型과 確率的模型을 축소시키고 그 협의의 뜻에 맞지 않는 분야는 별개의 분야로 독립시켜나가기로 한다.

確定的模型은 數理計劃法을 주축으로 하여 그래프·네트워크理論, 確定的在庫模型, 確定的形態의 게임모형등이 포함된다. 在庫模型에는 確定的模型과 確率的模型이 있어, 在庫模型을 분리시키기로 하며 게임모형 역시 같은 이유로 게임理論으로 분리시키기로 한다.

確率的模型에는 確率過程(Stochastic Process), 待期模型, 意思決定分析(Decision Analysis), 탐색이론(Search Theory), 確率的形態의 在庫模型, 確率的形態의 게임모형등이 있다. 여기서 在庫模型과 게임모형은 앞에서 분리되었던 在庫模型과 게임이론에 합할 수 있다.

이외에도 실제로 널리 활용되고 있는 모의실험(simulation), 대체이론, 信賴性模型등이 있다.

이들 모형들 중에서 범용성이 가장 큰 모형들을 골라보면

- 數理計劃法 : 線型計劃法, 整數計劃法, 0-1計劃法, 배낭모형, 配定模型, 輸送模型 動的計劃法, 目標計劃法, 多目的計劃法
 - 네트워크理論 : 最短經路模型, 最大流通模型, 最小費用模型, 活動네트워크模型
 - 게임理論 : 2人零合게임
 - 待期模型 : M/M/1 등 각종형태
 - 在庫模型 : 確定的模型, 確率的模型 등이
- 다. 물론 이들 模型 이외에도 예를 들어 모의실험은 가장 널리 活用되는 기법 중의 하나이지만 상황과 問題에 따라 방법과 절차가 달라지기 때문에 범용성을 지니기가 힘들다.

다음에는 이들 模型에 대한 프로그램을 개발함에 있어서 가장 적절한 計算方法에 대해

알아보도록 하자. 計算方法(algorithm)은 계속 발전하고 있고 가장 最近의 것이 일반적으로 더욱 효율적이다. 그러나 計算方法이 오래전에 개발되었다고 널리 알려져 있고 經營科學의 많은 교과서가 즐겨 사용하는 計算方法은 그것이 비록 효율면에서 떨어진다고 하더라도 이런 計算方法은 선택하는 것이 기초프로그램 개발에는 더욱 뜻이 있을 것이다. 이런 점을 감안하여 여기에서는 각 模型에 대해 다음과 같은 計算方法을 선택하였다.

線型計劃法 : 單體法(연립 1 次式維持)

明示型修正單體法

整數計劃法 : Gomory 切斷平面法

分枝限界法

0-1計劃法 : 分枝限界法(Balas Algorithm)

배낭모형 : Toyoda 法

Loulou & Michaelides 法

배정모형 : 항가리法

輸送模型 : 單體法(輸送型)

動的計劃法 : 투자문제, 역마차문제, 설비대

체문제, 재고문제

目標計劃法 : 修正 Balas 法등

多目的計劃法 : 線型計劃單體法

最短經路模型 : Dijkstra 法

最大流通模型 : Karzanov 法

最小費用模型 : Out-of-Kilter 法

PERT, CPM : 보편적인 방법사용

게임理論 : 축차근사법

線型計劃單體法

待期模型 : 보편적인 방법사용

在庫模型 : 보편적인 방법사용

여기서 PERT·CPM, 待期理論, 在庫模型은 계산방법이 비교적 간단하고 쉽기 때문에 특별한 계산방법을 지칭하지 않았다.

3. 프로그램작성

전술한 각 모형의 프로그램을 작성할때 특히 計算過程과 入出力形態의 印刷를 고려하여야 한다. 이 연구는 기본적인 프로그램을 개발하는 것이고 이 기본적인 프로그램들은 교육목적에 사용될 수도 있을 뿐만 아니라 이들 프로그램을 이용하여 더 複雜하고 발전된 프로그램을 개발할 것이기 때문에 특히 計算過程이 잘 나타나도록 해 주는 것이 좋다.

入出力設計에 대해서는 우선 入力は 두가지 방법이 있다. 하나는 BASIC에 있어서 READ란 명령어를 통해 資料를 入力시키는 것으로서 DATA文을 이용하여 프로그램안에 資料를 包含시키는 방법이다.

또 하나의 방법은 프로그램을 수행시킨 후에 對話式으로 資料를 入力시키는 것이다. 이 對話式은 편리한 방법이기 때문에 프로그램개발에 있어서 可能하면 이러한 入力方法을 包含시키는 것이 당연히 좋다. 마이콤에서 BASIC을 사용할 경우 이 대화식입력방법이 可能하기 때문에 이번 프로그램 개발에서는 이러한 대화식입력방법을 包含시켰다.

出力에 있어서는 教育用으로 사용될 경우와 실무용으로 사용될 경우를 고려하여 教育用으로 사용할 경우에는 세부적인 계산과정을 자세히 印刷토록 하였고 실무용으로 사용될 경우에는 最適解만 나타나면 충분하기 때문에 중간 계산과정이 나타나지 않도록 하였다. 이런 것을 印刷命令語에 代案을 두어 選擇하도록 하였다.

그래서 예로써 다음과 같은 선형계획법 문제가 있을 때

$$\begin{aligned} \text{Max } & 40x_1 + 50x_2 + 80x_3 + 110x_4 \\ \text{s. t. } & x_1 + x_2 + x_3 + x_4 \leq 15 \\ & 3x_1 + 5x_2 + 10x_3 + 15x_4 \leq 105 \\ & 7x_1 + 5x_2 + 3x_3 + 2x_4 \leq 120 \\ & x_i \geq 0 \end{aligned}$$

DATA文을 이용할 경우에는

```
10000 DATA 1, 2
10010 DATA 3, 4, 3, 0, 0
10020 DATA 1,1,1,1,3,5,10,15,
      7,5,3,2
10030 DATA 15,105,120
10040 DATA 40,50,80,110
```

으로 入力한다. 그리고 對話式으로 入力할 때는 다음과 같이 된다.

```
** ENTER DATA BY INTERACTIVE
MODE **
TYPE 1 FOR MAX, -1 FOR MIN? 1
<2> FOR DETAIL PRINTOUT
<1> FOR ONLY OPTIMAL SOLUTION
WHAT IS YOUR CHOICE? 2
NO OF CONSTRAINTS? 3
NO OF VARIABLES? 4
NO OF <= CONSTRAINTS? 3
NO OF = CONSTRAINTS? 0
NO OF >= CONSTRAINTS? 0
A(1,1)= ?1
A(1,2)= ?1
A(1,3)= ?1
A(1,4)= ?1
A(2,1)= ?3
A(2,2)= ?5
A(2,3)= ?10
A(2,4)= ?15
A(3,1)= ?7
A(3,2)= ?5
A(3,3)= ?3
A(3,4)= ?2
B(1)= ?15
B(2)= ?105
B(3)= ?120
C(1)= ?40
C(2)= ?50
C(3)= ?80
C(4)= ?110
```

다음으로 고려하여야 할 것은 프로그램작성 철학이다. 즉 프로그램을 계산효율에 치중할 것인가 또는 독자가 이해하기 쉽게

작성할 것인가 하는 것등이다.

이 연구에서는 경영·관리에 있어서의 기본적인 모형에 대한 프로그램개발이란 점을 감안하여

- 프로그램 이해의 편이성
- 부분프로그램 (subroutine) 化
- 확장성
- 入出力의 편이성
- 계산과정의 出力

에 많은 努力을 기울였다.

프로그램 이해의 편이성은 이 프로그램들이 실제에 活用될 때에 손질이 가해질 경우 우선 이 프로그램을 이해하여야 하는데 이 때 프로그램이 이해하기 힘들게 작성되어 있으면 應用하기 힘들다. 그래서 가능한한 프로그램을 이해하기 쉽게 작성하였다.

프로그램의 부분프로그램化는 프로그램이 가능한 계산과정에 있어서 가장 작은 단위별로 부분프로그램化되어 있으면 이런 프로그램은 活用, 應用 내지 확장시켜 사용하기가 편리하다. 그래서 이 연구가 기본프로그램의 개발이라는 입장을 감안하여 가능한 부분프로그램化시켰다.

확장성에 있어서 이 研究를 진행함에 있어서 이들 프로그램이 더 이상 확장시키지 않는 完成品이라는 생각보다는 이 프로그램을 이용하여 보다 複雜한 문제를 풀 수 있게 확장될 것이라고 생각하여 이들 프로그램을 부분프로그램으로 사용하기에 편리하도록 하였다.

이외 入出力이 편리하도록 入力方法을 두 가지로 선택할 수 있도록 하였다. 그리고 教育用으로 사용될 경우를 감안하여 계산과정이 나타나도록 하였다.

이런 점을 감안하여 完成한 프로그램은 다음의 23개 프로그램이다.

- | | |
|-----------------|-----------|
| 1. 선형계획법 | - LPBASIC |
| 2. 선형계획법 (고급) | - LPRESE |
| 3. 목표계획법 (교육용) | - GPRESE |
| 4. 목표계획법 (선형) | - GPILS |
| 5. 목표계획법 (비선형) | - GPNLS |
| 6. 목표계획법 (01) | - GP01 |
| 7. 다목적계획법 | - MLPRESE |
| 8. 정수계획법 | - IPCUTT |
| 9. 0-1 계획법 | - IP01 |
| 10. 배낭문제 | - IPKNAP |
| 11. 배정문제 | - ASSIGN |
| 12. 동적계획법 | - DPISIR |
| 13. 수송문제 | - TRANSI |
| 14. 최단거리문제 | - NETDIJ |
| 15. 극대나무문제 | - MINTRI |
| 16. 최소비용문제 | - NETOKF |
| 17. 최대유통문제 | - NETFLD |
| 18. PERT | - PERTGS |
| 19. CPM | - CPMEN |
| 20. 대기이론 | - QUEUE |
| 21. 재고모형 (확정적) | - INVEND |
| 22. 재고모형 (확률적) | - INVENP |
| 23. 게임이론 | - GAMESA |
| 24. 게임이론 (선형계획) | - GAMELP |
| 25. 마코프 인쇄 | - MARKOV |
| 26. 회귀분석 | - REGRES |
| 27. 확 율 | - PROBAB |