

電力系統計劃 및 運用의 最適化 技法으로서의 多重動的 計劃法

—차 례—

- 1. 緒 論
- 2. 動的計劃法의 基本原理
- 3. 第一過程多重動的計劃
- 4. 第 l 過程多重動的計劃
- 5. 狀態變數와 制御變數가 複數인 경우의 計算方法
- 6. 多重動的計劃法의 長點
- 7. 多重動的計劃法의 適用事例 및 效果

1. 緒 論

一般的으로 어떤 주어진 制約條件式

$$g_i(x_1, x_2, \dots, x_n) \{ \leq, =, \geq \} b_i, \dots \dots \dots (1)$$

$$i=1, 2, \dots, m$$

但, x_1, x_2, \dots, x_n ; 變數

를 만족시키면서 目的函數

$$Z=f(x_1, x_2, \dots, x_n) \dots \dots \dots (2)$$

를 最大 또는 最小로 하는 變數(x_1, x_2, \dots, x_n 과 이 경우에 대한 Z (最大値 또는 最小値를 決定하는 問題를 프로그래밍 문제(Programming Problem)라 하며, 이를 求하는 計算節次(Computational Procedures)를 프로그래밍(Programming)이라 한다. 그리고 函數 g_i 와 f 가 x_1, x_2, \dots, x_n 에 關하여 一次式으로 表示되는 경우를 線型計劃(Linear Programming)이라 하며, g_i 및 f 혹은 g_i 또는 f 가 非一次式으로 表示되는 경우를 非線型計劃이라고 한다.

그런데, 線型計劃 問題는 심플렉스法(Simplex Method) 등의 方法으로 그 解를 求하는 線型計劃서브루틴(LP Subroutine)이 널리 利用되고 있어 解를 얻기가 容易하나, 一般的으로 非線型計劃 問題는 그 解를 求하는 標準인 方法이 없을 뿐만 아니라 그 解가 얻어진다는 保障도 없다.

電力系統의 計劃 및 運用에 關係되는 最適化(Optimization) 問題는 거의 大部分이 非線型計劃問題에 속하므로 非線型計劃에 關한 現代技法의 導入 및 適用은 合

理的 電力系統計劃 및 運用에 必須的인 要素라 하겠다. 非線型 問題를 푸는 技法은 그 標準技法이 없는 대신, 問題의 性質에 따라서 가장 알맞는 各種 技法이 開發되고 있어 그 種類 또한 헤아릴 수 없을 程度라 하겠다. 그런데 그 中 電力系統計劃 및 運用에 關聯된 主要技法은 大略 다음과 같다.

- A) 傾斜法(Gradient Method) 및 이 部類에 속하는 技法
- B) 變分法(Variation Method)
- C) 最大原理法(Maximum Principle Method)
- D) 섀트(Sequential Unconstrained Minimization Technique)
- E) 몬테카를로法(Monte Carlo Method)
- F) 動的計劃法(Dynamic Programming Method)

특히 動的計劃法은 Richard Bellman이 開發한 技法으로서, 다른 技法과 比較할 때 마코프 過程(Markov Process)이 成立하는 最適化問題에 對하여는 그 解가 保障되고 計算알고리즘(Computational Algorithm)이 簡單하므로, 制約條件이 많은 複雜한 非線型 問題를 풀기에는 아주 適合하나, 變數가 많은 경우에는 反復的으로 計算하여야 하고, 解가 格子上的 交叉點에서 얻어지므로 固有誤差를 수반하며, 이 固有誤差를 줄이기 위하여는(즉, 計算精度를 높이기 위하여는) 計算時間이 길어지고, 假定 最適經路를 임시로 記憶해 두기 위하여는 計算機의 記憶 容量이 增加하는 短點이 있다. 筆者는 이와 같은 動的計劃法의 長點을 取하고 短點을 補完하기 위하여 새로이 多重動的計劃法(Multiple

* 正會員 · 서울工大副教授(工博) · 當學會事業理事

Dynamic Programming Method)을 開發하여, 그 理論的 長點을 證明하고, 이 技法을 漢江水系水資源의 最適多目的利用 問題에 適用한 結果 그 効用性이 立證되었으므로 아래에 이 多重動的計劃法의 原理에 對하여 紹介코자 한다. 그런데 多重動的計劃法을 從前의 動的計劃法의 原理에 바탕을 두고 있으며 또 多重動的計劃의 第1過程은 動的計劃法과 完全히 一致하므로, 于先動的計劃法의 基本原理에 對하여 略述한 다음 多重動的計劃法에 對하여 記述하기로 한다.

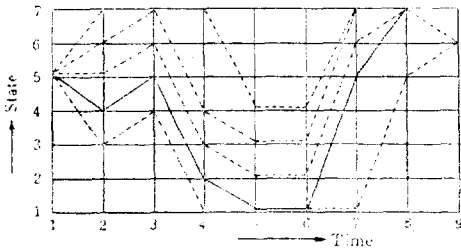


Fig. 1. Conventional Scheme of D.P

2. 動的計劃法의 基本原理

系統이 外部로부터의 制御量(Control)들에 의하여 狀態들이 變化할 경우, 例를 들면 水力發電所群의 各貯水池의 放水量에 依하여 各貯水池의 有效貯水量이 變化할 경우, 各 狀態와 制御量은 時間의 函數이다. 따라서 $X(t)$ 및 $U(t)$ 를 各各

$$X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_k(t) \end{pmatrix} \dots\dots\dots(3)$$

$$U(t) = \begin{pmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_k(t) \end{pmatrix} \dots\dots\dots(4)$$

를 임의 時刻 t 에 있어서의 狀態를 表示하는 k 次元 狀態벡터 및 狀態를 變化시키는 k 次元 制御벡터라 할 때 系統의 動的 特性은

$$\frac{d}{dt}X(t) = F(X(t), U(t), t) \dots\dots\dots(5)$$

로서 表示되고, F 는 系統의 特性에 따라 決定된다. 그리고 이 系統을 $t=0$ 에서 $t=t_f$ 의 期間동안 運用함으로써 얻어지는 利益(return) 또는 費用(cost) 即 系統의 汎函數(functional) C 는 그 初期 및 終期 狀態 및 이 期間中の 制御條件에 따라 決定되므로 C 는

$$C(X(0), X(t_f)) = \int_0^{t_f} r(X(t), U(t), t) dt \dots\dots\dots(6)$$

로서 주어진다. 例를 들면, 水力發電所群의 年間 運用費用은 初期 有效貯水量 및 年中 放水量 條件에 따라 決定된다.

그런데 우리의 關心事는 式 (6)의 C 를 最大(利益의 경우) 또는 最小(費用의 경우)로 하는 $U(t)$ 를 決定하는데 있고 $U(t)$ 를 알면 $X(t)$ 및 C 를 쉽게 얻을 수 있다. 式(5) 및 (6)은 連續系統(Continuous System) 이나, 動的計劃法에 適合하도록 離散系統(Discrete System)의 問題로 變換하면,

$$X(i+1) = \phi(X(i), U(i), i) \dots\dots\dots(7)$$

$$C(X(1), X(n+1)) = \sum_{i=1}^n r(X(i), U(i), i) \dots\dots\dots(8)$$

단, $i=1, 2, \dots, n$ 時列番號

와 같은 近似式으로 表示되고, 이 C 를 最大 또는 最小로 하는 n 個段의 最適 制御벡터 $\hat{U}(i)$ 에 對하여 全期間의 最高狀態벡터 $\hat{X}(i)$ 도 따라서 存在한다. 即,

$$\hat{C}(X(1), X(n+1)) = \sum_{i=1}^n r(\hat{X}(i), U(i), i) \dots\dots\dots(9)$$

그런데, 이 系統이 马尔可夫過程(Markov Process)이라고 하면, 最適性 原理(Principle of Optimality)가 成立하며, 最適性 原理는 現在 以後(即 第 i 時列 以後)의 最適制御方策(Optimal Policy)은 過去의 狀態나 制御方策과는 無關하고 다만 現在의 狀態에 의해서만 唯一하게 決定된다는 뜻이다. 動的計劃法은 이와 같은 最適性原理가 成立되는 马尔可夫過程에서만 適用할 수 있음에 注目하기 바란다.

이와 같은 最適性原理가 成立하는 系統에서 最小汎函數를 目的으로 하는 경우 式(9)는,

$$\hat{C}(X(1), X(2)) = \min_{u(1)} \{r(X(1), U(1), 1)\hat{C}(X(1), X(1))\} \dots\dots\dots(10)$$

와 같이 表示되고, $U(1)$ 의 값에 따라 $X(1)$ 이 달라야 하나 $X(1)$ 은 初期狀態로 固定되어 있으므로 $X(2)$ 를 指定할 경우 $U(1)$ 은 唯一하게 決定된다. 그리고

$$\hat{C}(X(1), X(1)) = 0$$

임은 분명하다. 다음 단계에서는

$$\hat{C}(X(1), X(3)) = \min_{u(2)} \{r(X(2), U(2), 2) + \hat{C}(X(1), X(2))\} \dots\dots\dots(11)$$

또는 一般의 으로는

$$\hat{C}(X(1), X(i+1)) = \min_{u(i)} \{r(X(i), U(i), i) + \hat{C}(X(1), X(i))\} \dots\dots\dots(12)$$

로서 표시되고 여러 값으로 $X(i+1)$ 指定하여 $C(X(1), X(i+1))$ 를 最小로 하는 最適制御벡터 即 $\hat{U}(i)|X(i+1)$ 探索한 後 이에 對應하는 $\hat{X}(i)|X(i+1)$ 를 記憶한다.

그리고 마지막 단계에서는

$$\hat{C}(X(1), X(n+1)) = \min_{u(n)} \{r(X(n), U(n), n) + \hat{C}(X(1), X(n))\} \dots\dots\dots(13)$$

와 같이 되고, 終刻狀態벡터 $X(n+1)$ 은 唯一하게 指定된다. 따라서 $\hat{X}(n)|X(n+1)$, $\hat{X}(n-1)|X(n)$, ..., $\hat{X}(i-1)|\hat{X}(i)$, ..., $\hat{X}(2)|X(3)$ 의 最適狀態벡터의 決定이 可能하다.

3. 第1過程多重動的計劃

多重動的計劃法은 多重過程으로 計算되고, 時間隔을 等間隔으로 分割하는 대신 任意로 便利하게 不等間隔으로 分割하는 것이 그 特徵이라 할 수 있다. 그런데 第1過程에서는 不等間隔으로 分割하는 것 以外에는 從前의 動的計劃法과 同一하다.

于先 第一過程에 對하여 記術하자면 다음과 같다.

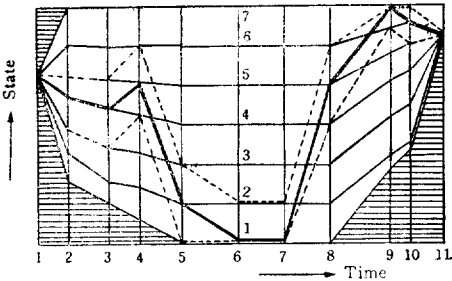


Fig. 2. The first stage of multiple D.P

即 그림 1은 從前의 動的計劃法의 格子圖를 表示하며, 여기서는 時間隔과 狀態間隔을 모두 等間隔으로 分割하고 있으나, 그림 2의 多動的計劃法의 第1過程은 前述한 바와 같이 時間隔을 不等間隔으로 分割하되, 分割方法이 다음과 같이 多少 差異가 있다. 即 實際로 狀態벡터 $X(i)$ 와 制御벡터 $U(i)$ 는 系統의 性質에 의하여 存在可能集合 $X(i)S$ 및 $U(i)S$ 에 屬하여야 한다는 制限條件 即

$$X(i) \in X(i)S \dots\dots\dots(14)$$

$$U(i) \in U(i)S \dots\dots\dots(15)$$

의 條件이 부과되고 通常的으로는

$$X(i)_{min} \leq X(i) \leq X(i)_{max} \dots\dots\dots(16)$$

$$U(i)_{min} \leq U(i) \leq U(i)_{max} \dots\dots\dots(17)$$

의 上下限 內에 存在하여야 하므로 多重動的計劃法에서는 다음과 같이 m_1 個의 等間隔 狀態로 分割함으로써 (m_1+1) 個의 狀態를 假定하는 格子를 構成한다.

$$X(i)_j = X(i)_{min} + \{X(i)_{max} - X(i)_{min}\} \frac{j-1}{m_1} \dots\dots\dots(18)$$

$$\text{단, } j=1, 2, \dots, m_1, m_1+1$$

따라서 第1過程에서는 그림 2에서 보는 바와 같은 格子에서, 式(12)로부터

$$C(X(1), X(i+1)_j) = r(X(i)_p, U(i)_p, i) + \hat{C}(X(1)_1, X(i)_p) \dots\dots\dots(19)$$

의 關係에서 指定된 各 j 에 대하여 $C(X(1)_1, X(i+1)_j)$ 를 最小로 하는 $P(P=1, 2, m+1)$ 와 이에 對應하는 最適 $\hat{C}(X(1)_1, X(i+1)_j)$ 를 決定하고, $i \rightarrow P$ 의 指數 連結(index connection)關係와 $\hat{C}(X(1), X(i+1)_j)$ 를 記憶 保存한다. 이 경우 $U(i)_p$ 가 式(17)의 制限條件이 充足 시키지 못하면 이 P 는 除外한다. 그리고 마지막 단계 j 는 $j=1$ 의 唯一한 값을 갖음은 分明하다. 따라서 第1過程에서의 最適狀態經路는 $\hat{C}(X(1)_1, X(n+1)_1)$ 의 決定이 끝난 후 完成된 下記 指數連結關係로 부터 얻어진다.

4. 第l過程多重動的計劃

第1過程의 多重動的計劃이 끝나면, 第2過程, 第3過程, ..., 第l過程의 計劃順序로 計算節次가 移行된다.

그림 3은 第2過程에 對한 格子圖를 표시한다. 第l過程이 第1過程과 다른 點은 式(14)~(17)의 制限條件이 보다 局限되는 데 있다. 即 第(l-1)過程에서 얻은 最適狀態軌跡(Optimum State trajectory)를 $\hat{X}(i)$

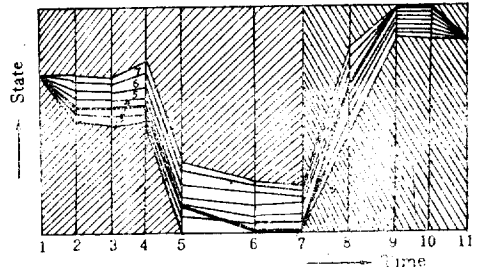


Fig. 3. The second stage of multiple D.P

i^{-1} , 또 第(l-1)過程의 狀態分割間隔을 $\Delta X(i)^{l-1}$ 이라 할 때,

$$\bar{S}(i)^{l-1} \triangleq \hat{X}(i)^{l-1} + \Delta X(i)^{l-1} \dots\dots\dots(20)$$

$$S(i)^{l-1} \triangleq \hat{X}(i)^{l-1} - \Delta X(i)^{l-1} \dots\dots\dots(21)$$

로서 定義되는 $\bar{S}(i)^{l-1}$ 및 $S(i)^{l-1}$ 에 對하여

$$X(i)_{max}^l = \text{Min}\{\bar{S}(i)^{l-1}, X(i)_{max}^{l-1}\} \dots\dots\dots(22)$$

$$X(i)_{min}^l = \text{Max}\{S(i)^{l-1}, X(i)_{min}^{l-1}\} \dots\dots\dots(23)$$

$$\text{단 } \begin{cases} \text{Min}(A, B) = \begin{cases} A & \text{if } A < B \\ B & \text{if } B < A \end{cases} \\ \text{Max}(A, B) = \begin{cases} A & \text{if } A > B \\ B & \text{if } B > A \end{cases} \\ X(i)_{max}^l = X(i)_{max} \\ X(i)_{min}^l = X(i)_{min} \end{cases}$$

를 計算한 後, 이 下限値와 下限値에 對하여 狀態벡터
를 다음과 같이 m_l 개의 等間隔으로 分割한다.

$$X(i)_j = X(i)_{m_l n} + \{X(i)_{max} - X(i)_{min}\} \frac{j-1}{m_l}$$

단, $j=1, 2, \dots, m_l, m_l+1$

이밖의 計算方法과 節次는 第1過程과 同一하다.

5. 狀態變數와 制御變數가 複數인 경우의 計算方法

以上에서는 狀態變數와 制御變數가 多變數인 경우, 即 式(3) 및 (4)로서 주어지는 벡터인 경우에 對하여 記述하였으나, 實際로 電子計算機로 處理할 때, 이와 같은 k 개의 變數에 對하여 式(19)의 最適 P 는 狀態變數의 數만큼 存在한다. 即 P 는 k 개의 要素로서 이루어진 集合이다. 따라서 任意的 各 時例 i 에서 最適 P 를 決定하기 위하여는 $(m_l+1)^k$ 개의 선택이 필요하기 때문에 아무리 能率이 좋은 計算機로서도 거의 處理가 不可能하다. 따라서, 實際 計算에 있어서는 于先 可能 狀態空間(feasible state space)範圍內에서 任意的 k 개의 狀態軌跡를 가정한다. k 와 $X_2(t), X_3(t), \dots, X_k(t)$ 의 狀態軌跡를 固定시킨 후 $X_1(t)$ 를 可變으로 한 單一 狀態變數로 近似化하여 $X_1(t)$ 의 最適軌跡를 決定한다. 이 경우 모든 制御變數 $U_1(t), U_2(t), \dots, U_k(t)$ 는 $X_1(t)$ 에 따라서 變하게 됨을 유의할 필요가 있다. 다음에는 $X_1(t), X_3(t), \dots, X_k(t)$ 를 固定시킨 후 $X_2(t)$ 의 最適軌跡를 決定한다. 이와같은 順에 의하여 마지막에는 $X_1(t), X_2(t), \dots, X_{k-1}(t)$ 를 固定시키고 $X_k(t)$ 의 最適軌跡를 決定한다. 이렇게 하여 第1回 反復計算이 끝난 後에는 第2回 反復計算過程에 들어가며, 다시 $X_1(t)$ 의 最適軌跡, $X_2(t)$ 의 最適軌跡의 順으로 逐次 計算을 行한다. 第 q_1 回 反復計算이 끝나면 第2過程의 多重動的計劃에 移行되며, 여기서도 마찬가지로 第 q_2 回 까지의 反復計算이 行하여지고, 第 l 過程의 多重動的計劃에서는 第 q_l 회까지의 反復計算이 行해진다.

6. 多重動的計劃法の 長點

多重動的計劃法の 長點은 計算時間의 短縮과 所要記憶容량의 節減面에서 찾을 수 있다.

計算時間面에서 從來의 動的計劃法과 比較하기 위해, 所要 計算時間이 大略 狀態벡터의 分割에 의한 선택수에 비례한다고 하면, 第 l 過程까지의 計算을 行한 多重動的計劃法の 所要計算時間과 從來의 動的計劃法の 所要時間의 比, 即 時間能率 η_s 는

$$\eta_s = \frac{M.D.P. \text{ time}}{D.P. \text{ time}} = \frac{m_1 + m_2 + \dots + m_l + l}{m+1} \dots \dots \dots (24)$$

로서 表示되고, 한편 計算精度가 같기 위하여는 兩者의 分割間隔數가 같아야 한다는 條件

$$m = \frac{m_1 m_2 \dots m_l}{2^{l-1}} \dots \dots \dots (25)$$

을 式(24)에 代入하면

$$\eta_s = \frac{2^{l-1}(m_1 + m_2 + \dots + m_l + l)}{m_1 m_2 \dots m_l + 2^{l-1}} \dots \dots \dots (26)$$

와 같이 된다. 그런데 η_s 를 最小로 하는 m_1, m_2, \dots, m_l 을 決定하기 위하여 $\frac{\partial \eta_s}{\partial m_i} = 0$ ($i=1, 2, \dots, l$)의 條件을 求하면,

$$m_s = m_1 = m_2 = \dots = m_l \dots \dots \dots (27)$$

이 얻어지고, 따라서 式(27)을 式(25) 및 (26)에 代入하면,

$$m = \frac{m_s^l}{2^{l-1}} \dots \dots \dots (28)$$

$$\eta_s = \frac{2^{l-1}(m_s + 1)}{m_s^l + 2^{l-1}} \dots \dots \dots (29)$$

이 된다. 따라서 多重動的計劃時 各 過程의 狀態벡터의 分割數는 式(27)에서와 같이 同一值 m_s 로 維持하는 것이 計算時間의 短縮面에서 가장 有利하다. 그리고 式(29)의 能率值를 實例를 들어서 計算한 結果는 表1과 같다. 表1에 依하면 多重動的計劃法은 動的計劃法에 比하여 狀態變數를 50分割하는 경우는 43%, 250分割하는 경우는 13%로 計算時間이 短縮되며, 分割數가 많을수록 그 威力이 더욱 發揮된다. 計算精度面에서는 式(25)의 條件에 의하여 同一한 것으로 간주하였으나, 事實은 式(22) 및 (23)의 關係로 부터 計算精度가 더욱 높을 可能性이 많은 反面 낮아지는 경우는 없다. 記憶容량의 節減面에서 考察한다면 所要記憶容량이 大略 狀態變數를 分割하는 格子目數에 比例한다고 보면, 그 兩者의 比, 即 記憶容량·節減能率 η_s 는 다음 式으로 表示된다.

$$\eta_s = \frac{M.D.P. \text{ memory capacity}}{D.P. \text{ memory capacity}} = \frac{m_s + 1}{m+1} \dots \dots \dots (30)$$

그런데 式(28)의 條件을 式(30)에 代入하면,

$$\eta_s = \frac{2^{l-1}(m_s + 1)}{m_s^l + 2^{l-1}} \dots \dots \dots (31)$$

와 같이 된다.

$m=50$ 및 250 의 경우에 對한 η_s 의 값을 계산한 값들은 表1에 表示하였는데, 所要記憶容량의 節減效果는 至大함을 알 수 있다.

따라서 筆者가 開發한 多重動的計劃法은 從來의 動的
<P. 77에 계속>