4.1 PROGRAMAREA DINAMICĂ

Programarea dinamică reprezintă o tehnică de abordare a unei clase de probleme al căror model matematic prezintă caracteristicile unui proces secvențial de decizie. Acest tip de procese se caracterizează prin faptul că în cadrul fiecărei etape trebuie aleasă, dintr-o mulțime de decizii admisibile pentru etapa respectivă, o anumită decizie şi, pornind de la aceasta, se poate determina utilitatea alegerii făcute sub forma unei valori numerice. Problema cere determinarea deciziei globale (strategiei), care reprezintă o mulțime care are atâtea elemente câte etape are procesul şi cuprinde deciziile luate la fiecare etapă, care optimizează (maximizează sau minimizează) o funcție obiectiv globală care depinde de valorile utilității realizate în cadrul fiecărei etape.

Dacă mulțimea etapelor procesului este un domeniu continuu se vorbeşte de proces secvențial continuu, iar când mulțimea etapelor este numărabilă se spune că este un proces secvențial discret. Acestea pot fi, la rândul lor, cu număr finit sau infinit de stadii (după cum mulțimea stărilor este finită sau infinită). În continuare vor fi analizate procesele discontinue discrete cu număr finit de stadii.

Trebuie subliniat faptul că mulțimea deciziilor admisibile, deci şi decizia aleasă la fiecare pas, şi, prin aceasta, valoarea funcției obiectiv depind de starea procesului care trebuie optimizat. Starea procesului va fi precizată prin vectorul parametrilor de stare.

Deşi caracterul secvențial al anumitor procese este de natură spațială sau logică, se preferă formularea acestora astfel încât să fie transpusă sub forma unei secvențialități temporale. Astfel, se va nota cu: $T = \{1, 2, \dots, n\}$ mulțimea celor n stadii succesive ale problemei (sau, corespunzător, celor n momente de timp). Pentru fiecare stadiu $i \in T$ se notează cu ξ_i starea procesului şi cu X_i mulțimea deciziilor admisibile pentru această stare. Considerând o decizie x_i din X_i utilitatea acestei decizii va depinde atât de decizia respectivă cât şi de starea procesului: $u_i(x_i, \xi_i)$. Utilitatea totală asociată unei strategii va fi:

$$F(u_1(x_1, \xi_1), \dots, u_n(x_n, \xi_n)) \tag{4.1.}$$

În majoritatea problemelor tehnice funcția utilității totale se obține prin însumarea utilităților fiecărui stadiu.

Dacă se notează cu x strategia (un vector ale cărui componente sunt deciziile alese în fiecare etapă) :

$$x = [x_1, x_2, x_3, \dots, x_n]^T \tag{4.2}$$

se numeşte traiectorie a procesului corespunzătoare strategiei x funcția ξ care defineşte starea procesului, controlat de strategia x de-a lungul etapelor procesului:

$$\xi(i) = \xi_i \quad \forall i \in T \tag{4.3}$$

O traiectorie ξ^* se numeşte *optimă* dacă extremizează, în sensul dorit, funcția de utilitate globală.

Tehnica programării dinamice constă în determinarea strategiei optime prin analiza secvențială a unor probleme de optimizare succesive asociate procesului descris de model. Această metodă are la bază principiul de optimalitate al lui R. Bellman care poate fi enunțat astfel: o traiectorie ξ^* este optimă pe T dacă oricare ar fi $i < j$ din T atunci vectorul $\left[\xi_i^*, \xi_{i+1}^*, \dots, \xi_j^*\right]^T$ reprezintă o traiectorie optimă pentru restricția procesului la mulțimea de stadii $T_1 = \{i, i+1, \dots, j\}$ şi starea inițială ξ_i^*.

În domeniul energetic au aplicații în special procesele secvențiale staționare în modelul cărora timpul nu apare în mod direct. Pentru ca un proces să poată fi considerat staționar trebuie să îndeplinească următoarele condiții:

-mulțimea deciziilor admisibile depinde doar de starea procesului, nu şi de timp;

-funcția de utilitate are aceeaşi valoare dacă starea şi decizia sunt identice, chiar dacă acestea corespund la două momente de timp diferite;

- decizii identice luate la momente de timp diferite, dar cărora le corespund aceeaşi stare, determină trecerea sistemului în aceeaşi stare.

Se poate considera că metoda programării dinamice constă, în esență, în compararea valorii funcției obiectiv într-un număr mare, dar finit, de puncte.

Meritul metodei constă în faptul că permite stabilirea şi parcurgerea sistematică a acestei mulțimi de puncte astfel încât să nu se piardă punctul de optim.

Această metodă nu impune algoritmi proprii pentru rezolvarea unei probleme tip, ci dă posibilitatea rezolvării unei probleme de optimizare prin înlocuirea sa cu o familie de probleme de optimizare, mai simple, care depind de un set mai restrâns de variabile de decizie.

Condiția ca o problemă de optimizare să poată fi rezolvată folosind metoda programării dinamice este ca funcția obiectiv să poată fi scrisă ca o combinație a unei mulțimi de funcții, fiecare dintre acestea depinzând de variabile distincte.

Avantajul principal al metodei programării dinamice îl reprezintă faptul că nu impune ca funcțiile care intervin în problema de optimizare să aibă forme particulare.

Se poate considera că o traiectorie a procesului descrie modul de succesiune a stărilor acestuia în cele *n* etape. Dacă ordinea de succesiune a stărilor respectă ordinea naturală a stadiilor se vorbeşte despre analiză prospectivă iar când este inversă în raport cu ordinea stadiilor se vorbeşte de analiză retrospectivă. În continuare se va considera o analiză prospectivă.

În cadrul acestei analize starea într-o anumită etapă va depinde de starea şi de decizia în etapa anterioară:

$$\xi_{i+1} = g_i\left(x_i, \xi_i\right) \tag{4.4}$$

prin funcția de trecere g.

Trebuie remarcat faptul că, prin funcțiile g, toate elementele care descriu problema sunt determinate de starea inițială ξ_1 şi de succesiunea de decizii alese la fiecare pas.

Pentru problema considerată se pot defini iterativ funcțiile $\widetilde{g}_i, \widetilde{u}_i, \widetilde{F}$ după cum urmează:

$$\begin{cases} \xi_2 = g_1(x_1,\xi_1) \overset{\text{not}}{=} \widetilde{g}_1(x_1,\xi_1) \\ u_1(x_1,\xi_1) \overset{\text{not}}{=} \widetilde{u}_1(x_1,\xi_1) \end{cases}$$

$$\begin{cases} \xi_3 = g_2(x_2,\xi_2) = g_2(x_2,\widetilde{g}_1(x_1,\xi_1)) \overset{\text{not}}{=} \widetilde{g}_2(x_1,x_2,\xi_1) \\ u_2(x_2,\xi_2) = u_2(x_2,\widetilde{g}_1(x_1,\xi_1)) \overset{\text{not}}{=} \widetilde{u}_2(x_1,x_2,\xi_1) \end{cases}$$

$$\begin{cases} \xi_4 = g_3(x_3,\xi_3) = g_3(x_3,\widetilde{g}_2(x_1,x_2,\xi_1)) \overset{\text{not}}{=} \widetilde{g}_3(x_1,x_2,x_{3,}\xi_1) \\ u_3(x_3,\xi_3) = u_3(x_3,\widetilde{g}_2(x_1,x_2,\xi_1)) \overset{\text{not}}{=} \widetilde{u}_3(x_1,x_2,x_3,\xi_1) \end{cases} \qquad (4.5)$$

$$\vdots$$

$$\begin{cases} \xi_{n+1} = g_n(x_n,\xi_n) = g_n(x_n,\widetilde{g}_{n-1}(x_1,\ldots,x_{n-1},\xi_1)) \overset{\text{not}}{=} \widetilde{g}_n(x_1,\ldots,x_{n,}\xi_1) \\ u_n(x_n,\xi_n) = u_n(x_n,\widetilde{g}_{n-1}(x_1,\ldots,x_{n-1},\xi_1)) \overset{\text{not}}{=} \widetilde{u}_n(x_1,\ldots,x_{n,}\xi_1) \end{cases}$$

$$F(u_1(x_1,\xi_1),\ldots,u_n(x_n,\xi_n)) =$$
$$= F(\widetilde{u}_1(x_1,\xi_1),\ldots,\widetilde{u}_n(x_1,\ldots,x_n,\xi_1)) \overset{\text{not}}{=} \widetilde{F}(x_1,\ldots,x_n,\xi_1)$$

O strategie $[x_1^*, x_2^*, \ldots, x_n^*]$ se numeşte *optimală* referitor la starea iniţială ξ_1 dacă:

$$\widetilde{F}(x_1^*, x_2^*, \ldots, x_n^*, \xi_1) = \max\{\widetilde{\mathrm{F}}(x_1,\ldots,x_n,\xi_1) | x_i \in \mathcal{X}_i, i=\overline{1,n}\} \qquad (4.6)$$

Dacă se notează prin F_{n-i+1} funcţia obiectiv asociată procesului de decizie limitat la etapele i, i+1,...,n (deci $F_n \equiv F$), problema de programare dinamică se numeşte *decompozabilă prospectiv* dacă există funcţiile $f_i : R^2 \to R$, i=1,...,n, $f_i(\alpha, y)$ nedescrescătoare în raport cu y pentru orice α fixat, astfel încât:

$$F_0 \equiv 0$$
$$F_{n-i+1}(u_i(x_i,\xi_i),\ldots,u_n(x_n,\xi_n)) = \qquad (4.7)$$
$$= f_{n-i+1}(u_i(x_i,\xi_i), F_{n-i}(u_{i+1}(x_{i+1},\xi_{i+1}),\ldots,u_n(x_n,\xi_n)))$$

sau, echivalent:

$$\begin{aligned}&\widetilde{F}_{n-i+1}(x_i, x_{i+1}, \ldots, x_n, \xi_i) = \\ &= f_{n-i+1}\left(u_i(x_i, \xi_i), \widetilde{F}_{n-i}(x_{i+1}, \ldots, x_n, g_i(x_i, \xi_i))\right)\end{aligned} \tag{4.8}$$

Proprietatea de decompozabilitate impune problemei de programare dinamică, în afara structurii de proces secvențial de decizie, ca, pentru fiecare stadiu i, procesul trunchiat corespunzător etapelor i, $i+1,...,n$ să fie caracterizat de o funcție obiectiv F_{n-i+1}, legătura dintre funcțiile obiectiv fiind de tip recursiv, de forma precizată în definiția anterioară a decompozabilității.

În continuare se va nota

$$\widetilde{h}_{n-i+1}(\xi_i) = \max_{\substack{x_j \in \widetilde{\mathcal{X}}_j(\xi_i) \\ i \le j \le n}} \widetilde{F}_{n-i+1}(x_i, \ldots, x_n, \xi_i) \tag{4.9}$$

unde:

$$\begin{aligned}&\widetilde{\mathcal{X}}_i(\xi_i) = \mathcal{X}_i(\xi_i) \\ &\widetilde{\mathcal{X}}_i(\xi_i) = \mathcal{X}_j\left(g_{j-1}\left(x_{j-1}, g_{j-2}\left(x_{j-2}, g_{j-3}(\ldots, g_i(x_i, \xi_i))\right)\right)\right); \quad j > i\end{aligned} \tag{4.10}$$

Cu această notație, se poate enunța următoarea teoremă:

Tr. (principiul optimalității) Dacă problema de programare dinamică este decompozabilă prospectiv, atunci:

$$\widetilde{h}_{n-i+1}(\xi_i) = \max_{x_i \in \widetilde{\mathcal{X}}_i(\xi_i)} f_{n-i+1}\left(u_i(x_i, \xi_i), \widetilde{h}_{n-i}(g_i(x_i, \xi_i))\right) \tag{4.11}$$

Această teoremă arată că orice strategie optimă pentru procesul trunchiat cuprinzând etapele $i+1,...,n$ trebuie să conțină, ca o etapă a sa, strategia optimă pentru procesul limitat la etapele $i+2,...,n$.

Deci, orice strategie optimală nu poate fi alcătuită decât din substrategii care sunt, la rândul lor, optimale.

Pe baza acestei teoreme se poate construi o tehnică secvențială pentru rezolvarea problemelor de programare dinamică, constituit din următoarele etape.

Etapa 1.

Se determină $x_n^*(\xi_n) \in \mathcal{X}_n(\xi_n)$ astfel încât:

$$\widetilde{F}_1\left(x_n^*(\xi_n), \xi_n\right) = \max_{x_n \in \mathcal{X}_n(\xi_n)} \widetilde{F}_1(x_n, \xi_n) \tag{4.12}$$

şi se obţine:

$$\widetilde{h}_1(\xi_n) = \widetilde{F}_1\left(x_n^*(\xi_n), \xi_n\right) \tag{4.13}$$

$\vdots$

Etapa i.

Se determină $x_i^*(\xi_i) \in \mathcal{X}_i(\xi_i)$ astfel încât:

$$\begin{aligned} &\widetilde{F}_{n-i+1}\left(u_i\left(x_i^*(\xi_i), \xi_i\right), \widetilde{h}_{n-i}\left(g_i\left(x_i^*(\xi_i), \xi_i\right)\right)\right) = \\ &= \max_{x_i \in \mathcal{X}_i(\xi_i)} \widetilde{F}_{n-i+1}\left(u_i\left(x_i(\xi_i), \xi_i\right), \widetilde{h}_{n-i}\left(g_i\left(x_i(\xi_i), \xi_i\right)\right)\right) \end{aligned} \tag{4.14}$$

şi se obţine:

$$\widetilde{h}_{n-i+1}(\xi_i) = \widetilde{F}_{n-i+1}\left(u_i\left(x_i^*(\xi_i), \xi_i\right), \widetilde{h}_{n-i}\left(g_i\left(x_i^*(\xi_i), \xi_i\right)\right)\right) \tag{4.15}$$

Odată cu pasul n este determinată ultima componentă, şi anume $x_1^*(\xi_1)$. În continuare, pornind de la starea iniţială ξ_1, se poate găsi strategia optimă: x^* ale cărei componente pot fi determinate ca: $x_1^* = x_1^*(\xi_1)$, $x_2^* = x_2^*\left(g_1\left(x_1^*, \xi_1\right)\right)$ etc.

Deci, rezolvarea problemei de optimizare iniţiale, care cere să se găsească maximul funcţiei $\widetilde{F}(x_1, \ldots, x_n, \xi_1)$, se face la capătul acestui proces secvenţial în cadrul căruia, în fiecare etapă, se cere rezolvarea unei probleme de optimizare cu o singură variabilă din cele n ale problemei date.

4.1 Repartiția unei cantități de resurse între diferite obiective

Se consideră ca exemplu problema împărțirii unei cantități totale X de resurse între n obiective diferite. Dacă se notează cu x_1, x_2,..., x_n cantitatea de resurse alocată fiecărui obiectiv, se reprezintă utilitatea repartizării resurselor fiecărui obiectiv prin funcțiile $u_1(x_1)$, $u_2(x_2),...,u_n(x_n)$ care îndeplinesc condiția:

$$u_i(0)=0; \quad i=\overline{1,n} \tag{4.16}$$

Problema cere maximizarea utilității totale reprezentată de suma utilităților parțiale:

$$F(x_1,x_2,\ldots,x_n)=u_1(x_1)+u_2(x_2)+\ldots+u_n(x_n) \tag{4.17}$$

atunci când sunt respectate condițiile:

$$\begin{aligned}&\sum_{i=1}^{n} x_i = X\\ &x_i \geq 0; \quad i=\overline{1,n}\end{aligned} \tag{4.18}$$

Pentru aplicarea metodei programării dinamice se transformă acest proces de repartiție într-un proces secvențial în timp, considerând că se repartizează o cantitate de resurse obiectivului cu numărul n, apoi obiectivului n-1 şi aşa mai departe până când se încheie cu alocarea resurselor pentru primul obiectiv.

În continuare se defineşte şirul de funcții $\{F_1, F_2, \ldots, F_n\}$ prin:

$$F_j(Y)=\max\left(u_1(x_1)+u_2(x_2)+\ldots u_j(x_j)\right) \tag{4.19}$$

atunci când variabilele satisfac restricțiile:

$$\begin{aligned}&\sum_{i=1}^{j} x_i = Y\\ &x_i \geq 0; \quad i=\overline{1,j}\end{aligned} \tag{4.20}$$

În acest şir sunt valabile relaţiile:

$$F_1(Y) = g_1(Y)$$
$$F_j(0) = 0; \quad j = \overline{1,n} \tag{4.21}$$

Fiecare dintre funcţiile $F_j(Y)$ astfel definite reprezintă utilitatea maximă care poate fi obţinută prin alocarea cantităţii Y de resurse obiectivelor de la 1 la j.

Pentru a obţine relaţia de recurenţă între funcţiile din acest şir se consideră că se începe prin alocarea cantităţii x_n de resurse din cantitatea totală X ultimului obiectiv. Corespunzător principiului de optimalitate, pentru a obţine o repartiţie optimă, cantitatea rămasă $Y_{n-1} = X - x_n$ trebuie repartizată astfel încât să se obţină utilitatea maximă de la celelalte n-1 obiective rămase.

Continuând cu acest raţionament se poate cere ca, dacă la pasul i a rămas nealocată cantitatea Y_i de resurse şi, din aceasta, se alocă cantitatea x_{n-i+1} obiectivului n-i+1, cantitatea care rămâne ($Y_i - x_{n-i+1}$) trebuie repartizată astfel încât să se obţină utilitatea maximă de la obiectivele de la 1 la n-i.

Această utilitate a fost definită ca fiind tocmai $F_{n-i}(Y_i - x_{n-i+1})$, ceea ce înseamnă că utilitatea totală care poate fi obţinută după ce se alocă cantitatea x_{n-i+1} obiectivului n-i+1 va fi:

$$u_{n-i+1}(x_{n-i+1}) + F_{n-i}(Y_i - x_{n-i+1}) \tag{4.22}$$

Valoarea maximă a acestei utilităţi, care este tocmai $F_{n-i+1}(Y_i)$, va putea fi determinată ca:

$$F_{n-i+1}(Y_i) = \max_{0 \le x_{n-i+1} \le Y_i} \left\{ u_{n-i+1}(x_{n-i+1}) + F_{n-i}(Y_i - x_{n-i+1}) \right\} \tag{4.23}$$

Pentru aplicarea metodei se consideră un pas de alocare Δ, care transformă intervalul $[0, X]$ în care variabilele iau valori într-o mulţime discretă $\{0, \Delta, 2\Delta, \ldots, M\Delta = X\}$.

În continuare se construieşte în mod iterativ un tabel care conține valorile funcțiilor F_{n-i+1} şi argumentelor lui u_{n-i+1} pentru care se atinge maximul reprezentat de F_{n-i+1}, tabel care are forma următoare:

Y	$F_1(Y)$	$X_1(Y)$	$F_2(Y)$	$X_2(Y)$		$X_{n-1}(Y)$	$F_n(Y)$	$X_n(Y)$
0	0	0	0	0		0	0	0
Δ	$u_1(\Delta)$	Δ						
2Δ	$u_1(2\Delta)$	2Δ						
$\vdots$								
$M\Delta$	$u_1(M\Delta)$	$M\Delta$						

Pentru completarea acestui tabel se porneşte de la primele coloane ținând cont că:

$$\begin{aligned} &F_1(Y)=u_1(Y) \\ &X_1(Y)=Y \end{aligned} \tag{4.24}$$

Determinarea lui $F_i(k\Delta)$ se face calculând valorile:

$$\begin{aligned} &u_i(0)+F_{i-1}(k\Delta)=F_{i-1}(k\Delta) \\ &u_i(\Delta)+F_{i-1}((k-1)\Delta) \\ &u_i(2\Delta)+F_{i-1}((k-2)\Delta) \\ &\vdots \\ &u_i((k-1)\Delta)+F_{i-1}(\Delta) \\ &u_i(k\Delta)+F_{i-1}(0)=u_i(k\Delta) \end{aligned} \tag{4.25}$$

valoarea lui $F_i(k\Delta)$ fiind cea mai mare din cele determinate la acest pas, iar $X_i(k\Delta)$ fiind argumentul lui u_i din expresia căreia îi corespunde valoarea maximă obținută.

După ce se încheie completarea tabelului se poate trece la găsirea strategiei optime prin utilizare unui tabel derivat din cel completat, păstrând doar coloanele $X_i(Y)$.

În acest tabel se porneşte pe linia corespunzătoare lui $M\Delta = X$ şi se merge în ultima coloană în care se găseşte cantitatea de resurse care trebuie alocate ultimului obiectiv $X_n = Q_n \Delta$.

În continuare se scade cantitatea de resurse cu cea alocată ultimului obiectiv (obținând $(M - Q_n)\Delta$), se revine în tabel pe linia corespunzătoare până în penultima coloană unde se găseşte cantitatea care trebuie repartizată penultimului obiectiv $X_{n-1} = Q_{n-1}\Delta$, se reduce cantitatea de resurse disponibile obținând $(M - Q_n - Q_{n-1})\Delta$ şi se revine în tabel pe linia corespunzătoare în coloana $X_{n-2}(Y)$ unde se găseşte cantitatea de resurse care trebuie alocată obiectivului n-2 şi aşa mai departe până la repartizarea întregii cantități de resurse.

	Y	$X_1(Y)$		$X_{n-2}(Y)$	$X_{n-1}(Y)$	$X_n(Y)$
	0	0		0	0	0
	$\vdots$					
3	$(M - Q_n - Q_{n-1})\Delta$			$Q_{n-2}\Delta$		
	$\vdots$					
2	$(M - Q_n)\Delta$				$Q_{n-1}\Delta$	
	$\vdots$					
1	$M\Delta$					$Q_n\Delta$

Acest algoritm de rezolvare se poate fi utilizat foarte uşor pentru implementare într-un limbaj de programare în scopul realizării unui program de calcul.

Aplicații

1. Să se determine planul de aprovizionare cu materiale pe durata unui an a unui şantier hidroenergetic. Se notează cu b_1, b_2, b_3 şi b_4 necesarul trimestrial şi se consideră că stocul maxim admisibil este reprezentat de valoare d. Planul optim este acela care minimizează cheltuielile de aprovizionare, exprimate prin valorile specifice c_1, c_2, c_3 şi c_4, care pot fi diferite pentru cele patru trimestre. Cantitatea existentă la începutul primului trimestru are valoarea a care este mai mică decât necesarul primului trimestru. Se admite că aprovizionarea trimestrială se face o singură dată, la începutul trimestrului.

Se vor considera următoarele valori numerice:

$$
\begin{aligned}
&b_1 = 1000; \quad b_2 = 900; \quad b_3 = 1300; \quad b_4 = 800;\\
&c_1 = 70; \quad c_2 = 60; \quad c_3 = 50; \quad c_4 = 80;\\
&a = 500; \quad d = 1500.
\end{aligned}
$$

Dacă se notează cu x_1, x_2, x_3 şi x_4 variabilele ce desemnează cantitățile aprovizionate la începutul fiecăruia din cele patru trimestre, restricțiile ce definesc domeniile $\mathscr{X}_i$ sunt reprezentate prin:

$$
\begin{aligned}
&x_i \in \left[\underline{x}_i, \overline{x}_i\right]\\
&\underline{x}_1 = b_1 - a;\\
&\overline{x}_1 = d - a;\\
&\underline{x}_2 = \max\{0, b_1 + b_2 - a - x_1\};\\
&\overline{x}_2 = d + b_1 - a - x_1;\\
&\underline{x}_3 = \max\{0, b_1 + b_2 + b_3 - a - x_1 - x_2\};\\
&\overline{x}_3 = d + b_1 + b_2 - a - x_1 - x_2;\\
&\underline{x}_3 = \max\{0, b_1 + b_2 + b_3 + b_4 - a - x_1 - x_2\};\\
&\overline{x}_3 = d + b_1 + b_2 - a - x_1 - x_2;
\end{aligned}
$$

Starea va fi descrisă de două variabile definite ca:

$$
\begin{aligned}
&\xi_1^1 = b_1 - a; \quad \xi_1^2 = d - a;\\
&\xi_{i+1}^1 = \xi_i^1 + b_{i+1} - x_i; \quad \xi_{i+1}^2 = \xi_i^2 + b_i - x_i; \quad i \ge 1
\end{aligned}
$$

Elementele care definesc procesul secvențial de decizie cu analiză prospectivă sunt:

$$T = \{1,2,3,4\}$$
$$\mathcal{X}_i(\xi_i) = X_i(\xi_i^1, \xi_i^2) = [\max\{0, \xi_i^1\}, \xi_i^2];$$
$$\xi_{i+1} = g_i(x_i, \xi_i);$$
$$u_i(x_i, \xi_i) = c_i x_i;$$
$$F(x_1, x_2, x_3, x_4, \xi_1, \xi_2, \xi_3, \xi_4) = \sum_{i=1}^{4} u_i(x_i, \xi_i);$$

unde funcțiile de trecere au următoarele expresii:

$$g_3(x_3, \xi_3) = [\xi_3^1 + 800 - x_3; \quad \xi_3^2 + 1300 - x_3]$$
$$g_2(x_2, \xi_2) = [\xi_2^1 + 1300 - x_2; \quad \xi_2^2 + 900 - x_2]$$
$$g_1(x_1, \xi_1) = [\xi_1^1 + 900 - x_1; \quad \xi_1^2 + 1000 - x_1]$$

În continuare se scriu funcțiile de utilitate asociate proceselor secvențiale parțiale:

$$F_4(x_1, x_2, x_3, x_4, \xi_1, \xi_2, \xi_3, \xi_4) = \sum_{i=1}^{4} u_i(x_i, \xi_i)$$
$$= \sum_{i=1}^{4} c_i x_i = F(x_1, x_2, x_3, x_4, \xi_1, \xi_2, \xi_3, \xi_4);$$
$$F_3(x_2, x_3, x_4, \xi_2, \xi_3, \xi_4) = \sum_{i=2}^{4} u_i(x_i, \xi_i) = \sum_{i=2}^{4} c_i x_i;$$
$$F_2(x_3, x_4, \xi_3, \xi_4) = \sum_{i=3}^{4} u_i(x_i, \xi_i) = \sum_{i=3}^{4} c_i x_i;$$
$$F_1(x_4, \xi_4) = u_4(x_4, \xi_4) = c_4 x_4;$$

După cum se constată problema este decompozabilă prospectiv dacă se consideră funcțiile f_i de forma:

$$f_i(u, v) = u + v$$

care îndeplinesc condițiile din definiție.

Deoarece se cere determinarea strategiei care conduce la cheltuieli minime se va putea aplica algoritmul prezentat dacă se înlocuieşte peste tot condiția de maxim prin condiție de minim.

Pasul 1.

Se caută decizia pentru ultima etapă care conduce la minimizarea costurilor:

$$\min_{x_4 \in \mathscr{X}_4(\xi_4)} \left\{ \widetilde{F}_1(x_n, \xi_n) \right\} = \min_{x_4 \in \mathscr{X}_4(\xi_4)} \left\{ 40 x_4 \right\}$$

unde: $\mathscr{X}_4(\xi_4) = \left[\xi_4^1, \xi_4^2\right]$

Deoarece funcția care trebuie minimizată este crescătoare, minimul său va fi atins în punctul $x_4^* = \xi_4^1$ şi se obține:

$$\widetilde{h}_1(\xi_4) = 80\xi_4^1$$

Pasul 2

Se urmăreşte determinarea deciziei pentru penultima etapă care conduce la minimizarea costurilor pentru ultimele două etape:

$$\min_{x_3 \in \mathscr{X}_3(\xi_3)} \left\{ 50 x_3 + 80\xi_4^1 \right\}$$

în care ξ_4^1 se obține din expresia funcției g_3:

$$\left[\xi_4^1, \xi_4^4\right] = g_3(x_3, \xi_3) = \left[\xi_3^1 + 800 - x_3; \quad \xi_3^2 + 1300 - x_3\right]$$

ceea ce conduce la:

$$\min_{x_3 \in \mathscr{X}_3(\xi_3)} \left\{ 50 x_3 + 80\left(\xi_3^1 + 800 - x_3\right) \right\} = \min_{x_3 \in \mathscr{X}_3(\xi_3)} \left\{ 80\xi_3^1 - 30 x_3 + 64000 \right\}$$

Minimul va fi atins pentru $x_3^* = \xi_3^2$ (deoarece funcția este descrescătoare în raport cu x_3) iar valoarea acestui minim va fi:

$$\widetilde{h}_2(\xi_3) = 80\xi_3^1 - 30\xi_3^2 + 64000$$

Pasul 3

Procedând ca şi în cazul anterior rezultă, cu ξ_3^1 şi ξ_3^2 obținute din expresia lui g_2:

$$\min_{x_2 \in \mathscr{X}_2(\xi_2)} \left\{60x_2 + \widetilde{h}_2(\xi_3)\right\} = \min_{x_2 \in \mathscr{X}_2(\xi_2)} \left\{60x_2 + 80\xi_3^1 - 30\xi_3^2 + 64000\right\} =$$
$$= \min_{x_2 \in \mathscr{X}_2(\xi_2)} \left\{60x_2 + 80\left(\xi_2^1 + 1300 - x_2\right) - 30\left(\xi_2^2 + 900 - x_2\right) + 64000\right\} =$$
$$= \min_{x_2 \in \mathscr{X}_2(\xi_2)} \left\{10x_2 + 80\xi_2^1 - 30\xi_2^2 + 141000\right\}$$

şi se obține valoarea minimă pentru $x_2^* = \xi_2^1$:

$$\widetilde{h}_3(\xi_2) = 90\xi_2^1 - 30\xi_2^2 + 141000$$

Pasul 4

$$\min_{x_1 \in \mathscr{X}_1(\xi_1)} \left\{70x_1 + \widetilde{h}_3(\xi_2)\right\} = \min_{x_1 \in \mathscr{X}_1(\xi_1)} \left\{70x_2 + 90\xi_2^1 - 30\xi_2^2 + 141000\right\} =$$
$$= \min_{x_1 \in \mathscr{X}_1(\xi_1)} \left\{70x_2 + 90\left(\xi_1^1 + 900 - x_1\right) - 30\left(\xi_1^2 + 1000 - x_1\right) + 141000\right\} =$$
$$= \min_{x_1 \in \mathscr{X}_1(\xi 1)} \left\{10x_2 + 90\xi_1^1 - 30\xi_1^2 + 1192000\right\}$$

cu minimul atins pentru $x_1^* = \xi_1^1$, minim cu valoarea:

$$\widetilde{h}_4(\xi_1) = 1000\xi_1^1 - 30\xi_1^2 + 1192000$$

Deoarece: $\xi_1 = [500{,}1000]$

rezultă:

$$x_1^* = 500$$

şi, în continuare:

$$\xi_2 = g_1\left(x_1^*, \xi_1\right) = \left[\xi_1^1 + 900 - x_1^*;\quad \xi_1^2 + 1000 - x_1^*\right] = [900;1500]$$
$$x_2^* = \xi_2^1 = 900$$
$$\xi_3 = g_2\left(x_2^*, \xi_2\right) = \left[\xi_2^1 + 1300 - x_2^*;\quad \xi_2^2 + 900 - x_2^*\right] = [1300;1500]$$
$$x_3^* = \xi_3^2 = 1500$$
$$\xi_4 = g_3\left(x_3^*, \xi_3\right) = \left[\xi_3^1 + 800 - x_3^*;\quad \xi_3^2 + 1300 - x_3^*\right] = [600;1300]$$
$$x_3^* = \xi_3^2 = 1300$$

2. Să se stabilească locul optim de instalare a cinci grupuri termoenergetice în cadrul a patru centrale. Utilitatea instalării grupurilor în fiecare din cele patru centrale (care cuprind toate tipurile de pierderi, inclusiv transportul combustibilului şi pierderile în reţea) este cuprinsă în tabelul următor.

Nr. grupuri	Utilitate			
	$u_1(n)$	$u_2(n)$	$u_3(n)$	$u_4(n)$
0	0	0	0	0
1	38	40	37	41
2	81	84	79	82
3	118	117	119	115
4	152	148	149	150
5	198	191	196	195

În continuare se completează tabelul derivat. Coloana $F1$ reprezintă de fapt prima coloană din tabelul utilităţilor, coloana X_1 este identică cu coloana N. Ca exemplu se arată modul în care se calculează $F_2(3)$. În acest scop se calculează următoarele valori:

$$u_2(0)+F_1(3)=0+118=118$$
$$u_2(1)+F_1(2)=40+81=121$$
$$u_2(2)+F_1(1)=84+38=122$$
$$u_2(3)+F_1(2)=117+0=117$$
$$\max\{118,121,122,117\}=122$$

deci valoarea lui $F_2(3)$ va fi 122 iar argumentul lui u_2 (n=2) în expresia care are valoarea maximă va fi valoarea lui $X_2(3)$.

N	F_1	X_1	F_2	X_2	F_3	X_3	F_4	X_4
0	0	0	0	1	0	0	0	0
1	38	1	40	2	40	0	41	1
2	81	2	84	2	84	0	84	0
3	118	3	122	2	122	0	125	1
4	152	4	165	2	165	0	166	2
5	198	5	202	2	203	3	206	1

În continuare se păstrează tabelul care va fi folosit pentru determinarea soluției optime:

N	X_1	X_2	X_3	X_4
0	0	1	0	0
1	1	2	0	1
2	2	2	0	0
3	3	2	0	1
4	4	2	0	2
5	5	2	3	1

Pentru N=5 se merge în ultima coloană şi se găseşte $x_4^* = 1$ deci se repartizează un grup în ultima centrală.

Se calculează: $N - x_4^* = 5 - 1 = 4$ ceea ce înseamnă că, la următorul pas se caută pe penultima linie până în penultima coloană, de unde rezultă $x_3^* = 0$ şi $N - x_4^* - x_3^* = 4$

În continuare se caută tot pe linia corespunzătoare lui N=4 în coloana X_3, rezultând $x_2^* = 2$ şi, mai departe:

$$N - x_4^* - x_3^* - x_2^* = 2$$

$$x_1^* = 2$$

Deci, repartiția optimă presupune repartizarea celor cinci grupuri în felul următor: câte două grupuri în prima şi a doua centrală şi un grup în ultima.