RECURRENT DYNAMIC PROGRAMMING

Aleksandr Alekseievitch Tsoi
Faculdade de Engenharia e Arquitetura
UPF –Passo Fundo , RS

RESUMO

Para o problema de programação não-linear separável discreta relativo a grafo, são desenvolvidas as técnicas para solução imediata e para otimização por partes. As duas fundamentam-se no método de programação dinâmica que resulta em um algoritmo de programação dinâmica embutido dentro do outro. Ambas utilizam essencialmente a estrutura do grafo do problema. A aplicação múltipla do particionamento gera uma técnica hierarquicamente recursiva do método de programação dinâmica.

SUMMARY

We developed the techniques for immediate solution and optimization by parts for discrete nonlinear separable programming problem on the graph. These two techniques are based on the use of the dynamic programming method that results in obtaining one algorithm of dynamic programming built into another one. The both techniques make use of the

graph structure. Multiple use of the decomposition is generalized in the frames of the hierarchically recurrent algorithm of dynamic programming.

1. INTRODUCTION

Let us consider the following separate discrete separable problem of nonlinear programming on the graph :

$$Z=\min_{\mathbf{x}} \sum_{j=1}^{n} f_j(x_j) \qquad (1)$$

subject to : $$\mathbf{A}\ \mathbf{G(X)} \leq \mathbf{B}. \qquad (2)$$

Where : $$G(X)=\begin{Vmatrix} g_1(X_1) \\ g_2(X_2) \\ . \\ . \\ g_n(X_n) \end{Vmatrix} ; \qquad B=\begin{Vmatrix} b_1 \\ b_2 \\ . \\ b_m \end{Vmatrix}$$

$A=\| a_{ij} \|$ is the (m . n) matrix of incidences in paths connecting sources and sinks of the graph ..

The graph .(V,E) is the directed and valued tree graph; x_j, $f_j(x_j)$, and $g_j(x_j)$ are the variables and functions, connected with its edges $e \in E$.

The components of the vector $X(x_1, x_2, \dots x_n)$ are non-negative and are restricted by requirements imposed on discretness , that is

$$x_j \geq 0\ ,\ \ x_j \in \mathbf{D}\ ,\ \ \text{for } \forall x_j \in \mathbf{X} \qquad (3)$$

Here, D is the set of discrete magnitudes of variables .

Within the structure of the model (1)-(3), one can formalize a number of network problems related to the resources distribution optimization, the problem of transportation, the network developments, the optimal decision problems, and others [1-6].

The dynamic programming method (DPM) finds its effective utility when solving the given class problems including both the problems of formal optimization [1-3] and the problems of practical engineering applications [4-7] .

The explanation of high effectiveness of the method as applied to the graph problems is in the very essence of the method itself : because the solution procedure, in accordance with this method, is the well-adapted to the tree graph structure. This circumstance makes it possible to eliminate the necessity of multiple-state process , even if the system (2) has many a number of constraints imposed.

Making use of this advantage we developed a DPM modification as applied to the energy distribution optimization [7,8]. The graph . represents in this case the network configuration ; the functions $g_j(x_j)$ take meanings of voltage loss, and the optimization variables are the cable sizes.

In the herein-presented paper we attempt to generalize the above-mentioned approach to the solution of the problem (1)-(3) by parts. Multiple hierarchical use of such generalization has led, as a result, to the creation of the built-in recurrent DP algorithm .

2. ALGORITHM OF IMMEDIATE OPTIMIZATION

In the process of immediate optimization we consider the original structure of the graph . of the problem (1)-(3).

As applied to the problem given the DP method uses the natural tree-graph hierarchy for cases when the source is connected with the higher hierarchy level ($y=Y$) and when the sinks belong to the low level ($y=1$), (Fig.1). There form , thus , the set of the level $Y = \{ y / y = 1, 2,\dots, Y \}$.

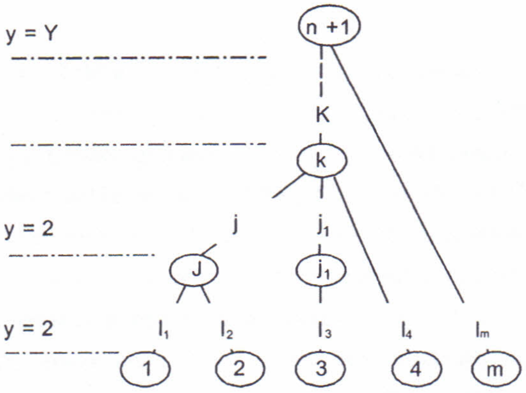

Fig. 1

In order to the state space discretization , we shall introduce the following sequence of vectors :

$$\mathbf{B} = \{\mathbf{B}^{(\xi)} / \xi = 1,2,.....,D\},$$

where

$$\mathbf{B}^{(\xi)} = \{ b_i^{(\xi)} / b_i^{(\xi)} = \ldots _i \ ; \ i = 1,2,...,(n+1)\}, \ . = 1,2,...,D\};$$

$$\Delta_i = b_i / D,$$

Here Δ_i is the step of discretization.

2.1: Procedure of solution

The DP process is implemented in the structure of the graph Γ. Following the general DPM logic, the multistage decision process consists of two procedures , that is the direct steps and the inverse steps.

2.1.1: Direct steps

The problem is solved in these steps by moving from sinks to the sources of the graph .. Making use of the Bellmann's principle of optimality

[2,3] we are able to obtain in the problem (1)-(3) the recursive relation in the form of the following sub-problem:

For $\forall \xi = 1,2,...,D,$ find :

$$F_k(\xi) = \min_{x_j,\psi} \{ f_j(x_j) + \sum_{l \in \mathbf{Y}_{(y-1)i}} F_l(\psi) \}, \tag{4}$$

subject to :

$$g_j(x_j) \le b_k^{(\xi)} - b_i(\psi); \tag{5}$$

$$x_j \ge 0 \quad , \quad x_j \in \mathbf{D}; \tag{6}$$

$$\psi = 1,2,...,D$$

$$k, j \in \mathbf{Y}_y \quad ; \quad i \in \mathbf{Y}_{y-1},.$$

where Y_y and Y_{y-1} are the subsets of vertices and edges of levels y and (y -1), respectively ; k, i are the initial and final vertices of the edge j ; $b_k^{(\xi)}$ and $b_i(\psi)$ are the components of B , where $b_k(\psi) = b_i(\psi)$ was renewed in the previous steps.

The renewal of B must be done in every step ; in so doing, the following relations are used :

$$b_k(\xi) = \max_{j \in \mathbf{Y}_{yk}} [b_i(\psi^*) + g_j(x_j^*(\xi))], (\text{for} : \forall k \in \mathbf{Y}_y \text{ and } \forall \xi = 1,2,...,D). \tag{7}$$

Here, Y_{yk} is the subset of edges adjacent to the vertex k of the levels equal to y= p or lower ; $b_i(\psi*)$ are the components of B associated with the final vertex of these edges . These components were already corrected in the previous steps .

The use of renewal makes it possible to lower the redundancy of resources in the final solution, and, hence, to exclude the mistakes of discretization.

The sum in (4) consists of the optimal solutions $F_l(\psi)$ connected with the edges ($l \in Y_{(y-1)i}$) of levels equal to (y-1) or lower, which are adjacent to the final vertex of j ; $Y_{(y-1)i}$ is the subset of such edges. These solutions can be obtained in the previous steps.

The optimal values being found, that is, $x_j(\xi)$, $\psi_i * (\xi)$ and $F_k(\xi)$, and also, $b_k^{(\xi)} = b_k(\xi)$ are stored in tables for their use in subsequent steps. Fig. 2 presents the model of these tables.

ξ	x^*_j	F_k	ψ^*_i	b_k
1				
2				
.				
ξ	$x^*_j(\xi)$	$F_k(\xi)$	$\psi^*_i(\xi)$	$b_k(\xi)$
.				
D				

Fig. 2.

Owing to the recursive character of (4)-(7), the values of $F_i(\psi)$ and $b_j(\psi)$ needed for calculations in the step y are always available and stored in the array of data obtained in previous steps.

In development, the recursive use of (4)-(7) gives the following recursive algorithm :

Step 1 (p=1): Here we consider the edges j of the first , that is , the lowest level (y=1) adjacent to the sinks. For the case shown in Fig.1 , j = $l_1, l_2, \ldots \ldots, l_m$.

For every of these j , a number of optimization sub-problems (4)-(6) are being solved ; these are trivial in nature, and the recursive relation takes here its reduced particular form, i. e.

For $\forall \xi = 1,2,\ldots,D,$ *find* $x_j^*(\xi)$ *so that*

$$F_j(\xi) = f_j[x_j^*(\xi)] = \min_{x_j} f_j(x_j), \tag{8}$$

subject to :

$$g_j(x_j) \le b_b^{(\xi)} \tag{9}$$

$$x_j \ge 0 \ , \ x_j \in \mathbf{D} \ ,$$

$$j,\ k \in \mathbf{Y}_1$$

Here, the vertex k is the initial vertex of the edge j (in Fig.1 $k = i$, $i_1 \in j, j_i$) ; $x_j^*(\xi)$ is the optimal magnitude of x_j for given ξ ; Y_1 is the sub-set of vertices and edges of the level 1.

Making use of the so-obtained $x_j^*(\xi)$, we compute $b_i^{(\xi)}$ for B for all vertices i of the first level ($i \in Y_1$) with the use of expressions (7)

$$b_k(\xi) = \max_{j \in \mathbf{Y}_{1i}} g_j[x_j^*(\xi)], \qquad \xi = 1,2,\ldots,D.$$

Here Y_{1k} is the sub-set of all edges of the level 1, that are adjacent to k. The optimal values x_j^* and $F_k(\xi) = f_j^*(\xi) = f_j[x_j^*(\xi)]$, and also $b_k^{(\xi)} = b_k(\xi)$ renewed are stored can in one of the tables. Fig.3, line y = 1, presents these tables used in the step 1 .

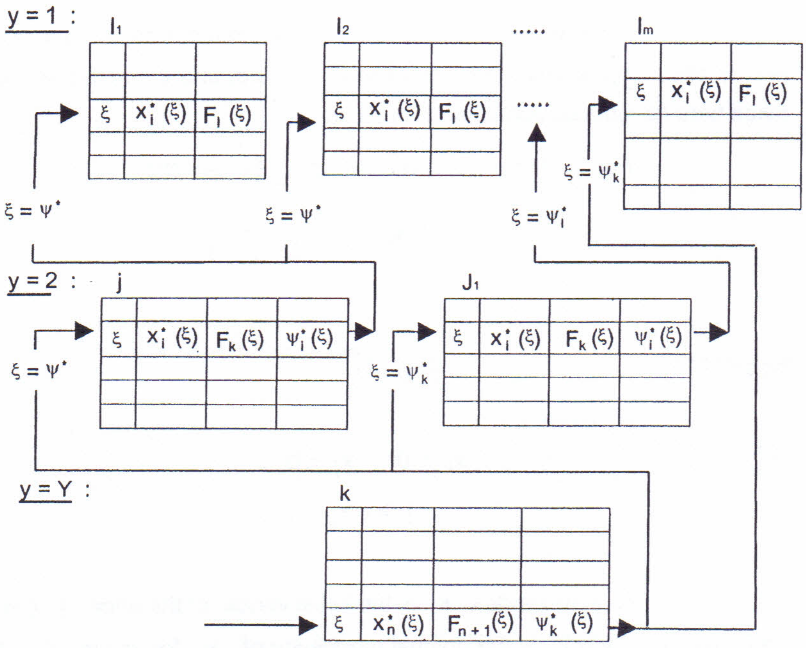

Fig. 3.

Step 2 (p = 2) : In this step, we optimize all branches that are incident to the vertices of the level y = 2 . For every vertex k and every edge j of the level 2 connected with this vertex , we solve a number of the following sub-problems :

For $\forall\ \xi = 1,2,\dots,D,$ *find:*

$$F_k(\xi) = \min_{x_j,\psi} \{f_j(x_j) + \sum_{l\in \mathbf{Y}_{1i}} F_l[x_l^*(\xi)]\} \quad ; \qquad (10)$$

Subject to :

$$g_j(x_j) \le b_k^{(\xi)} - b_i(\psi),$$
$$x_j \ge 0\ ,\quad x_j \in \mathbf{D}\ , \qquad (11)$$

$$\psi = 1,2,\dots,D$$

$$k, j \in \mathbf{Y}_2, l \in \mathbf{Y}_1$$

The sum in (10) takes account for all edges-sinks $l \in Y_{1i}$ adjacent to the edge j in its end j (belonging to $y = 1$).

The set B is renewed for $\forall i \in Y_2$ in accordance with the following formula :

$$b_k(\xi) = \max_{j \in \mathbf{Y}_{2i}} [b_i(\xi) + g_j(x_j^*(\xi))], \qquad (\forall \xi = 1,2,\dots,D)$$

Here, Y_{2i} is the set of backward edges of the vertex i of the level $y = 1 ; 2$.

The step 2 results thus in intermediate optimal solution for every edge of the level 2 and all of its backward edges of the level 1. These optimal solutions are represented in the form of a function of resources associated with vertices of the level 2.

Step 3,4,... ($p = 3,4...$) : In this and all subsequent steps, we repeat our calculations that are similar to those made earlier ; in doing so, we make use of the relation (4)-(7) in its general complete form. In every step of these calculations, we consider the vertices and edges belonging to the level $y = p$. For every of these edge $j \in Y_y$ its forking branches are optimized .

The steps of iteration must be continued until the higher level $y = Y$ be reached, more specifically, until the source of the tree graph be reached. In this stage, we determine the values $b_k{}^{(\xi)}$ (they are equal to $b_{(N+1)}$).

Last step ($p = Y$): In the last direct step, we make use of the recursive relations (4)-(7) in the form of the following special problem :

Find

$$F_{(n+1)}(\xi) = Z = \min_{x_j, \psi} \{f_j(x_j) + \sum_{l \in \mathbf{Y}_{(y-1)n}} F_l(\psi)\} ; \qquad (12)$$

subject to :

$$\begin{gathered} g_j\,(x_j) \leq b_{(n+1)} - b_n(\psi) \\ x_j \geq 0\ ,\ \ x_j \in \mathbf{D} \\ \psi = 1,2,\ldots,D \\ (n+1)\,j \in \mathbf{Y}_Y, n \in \mathbf{Y}_{Y-1} \end{gathered} \tag{13}$$

Here, $F_k(\xi) = F_{(n+1)}(\xi) = Z$, that is, it represents the economic criterion for the whole graph .. The solution of (12)-(13) makes it possible to determine its minimum as a function of the available resources $b_{(n+1)}$.

If $b_{(n+1)}$ is known, the relations yield the definitive optimal solution for the initial edge and give the optimal value for the parameters x_j^* and ψ^* of its final vertex .

In doing it, use can be made of the inverse steps process whose function is to obtain the definitive optimal solution for all other edges and vertices.

2.1.2: Inverse steps.

In the last direct step the optimal solution for the edge adjacent to the source (y = Y ;1) was determined . We determine here also the optimal values for ψ^*, and for every vertex n of the level (Y-1) that represent the final ends of such edges.

Technically, it means that the optimal solution for the source specifies the optimal values of parameters, distributed throughout the rest of the graph.

The vertex n represents, in its turn, the source for the section trees of such point derivation. For this reason, if $b_n(\psi^*)$ is known, the results of optimization in the next to last step p = (y-1) contain the optimal solutions for backward edges adjacent to vertices n (y = (Y-1) ;1). The afore-mentioned results are stored in the tables of date on this next to last step or the step p = 1.

For finding of a solution for one edge $y = Y-1$, it is sufficient to enter the corresponding table it its line with the number $\xi = \psi^*$. The value of x_j^* written in this table will be the sought optimal solution. This very line gives the optimal value for the parameter $\xi = \psi^*$ of the vertex of the level (Y-2), i.e., the final vertex of the reference edge.

The value $b_n(\psi^*)$ generates, thus, the complete set of optimal solutions for the edges $y = Y-1$. We determine simultaneously the optimum $b_k(\psi^*)$ in all vertices of the level $y = Y-2$, which make it possible to find optimal solutions $b_i(\psi^*)$ for the edges $y = (Y-2)$ at vertices of the level $y = (Y-3)$, and so on.

The procedure is continued until $y = 1$ be reached (when the edges adjacent to sinks are achieved). At this moment the inverse procedure completes, and, as a result, we obtain a number of optimal values for all variables $x_j \in X$ of the initial problem (1)-(3).

Figure 3 presents the schematic tracing back trajectory (shown by arrows).

The fact, that the optimal and the admissible intermediate solutions are available in tables for every step of direct process, can guarantee the optimality of solutions obtained and the satisfaction of constraints imposed.

The DP method ensures, in its turn, the achievement of the global optimum for a solution given .

3: DECOMPOSITION

The afore-described DP algorithm can be extended to the case of solving the problem (1)-(3) by parts. Instead of the formal decomposition of the problem (1)-(3), it would be more appropriate, to make use of the advantage of the own graph structure in its partition.

With this aim, we carry out the partition of the initial graph Γ (V,E) into the section graphs $\Gamma_i(V_i,E_i)$, i=1,2,..., N, in a way shown in Fig.4a.

The decomposition is carried out here by cutting of certain vertices. It is obvious that all Γ_i are trees. It is of importance that the partition is carried out so that every sub-tree has maximum two points of connection with other sub-trees. Connections with sinks can be multiple.

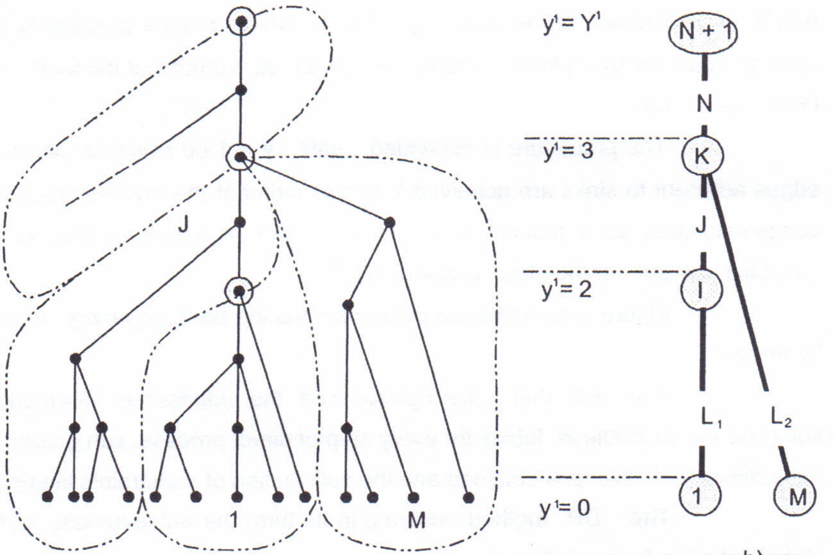

Fig. 4

3.1: Graph of Decomposition

Taking account of the fact that all the section graphs Γ_i , i=1,2,...,N have two points of interconnection, the method of condensation offer the possibility of their transformation into the generalized edges, and introduce, in such a manner , the generalized graph of the problem , let us call it the graph of decomposition Π (C,S) (Fig.4b). The elements of this graph are the set C of the separation points and the set S of the generalized edges

The graph of decomposition represents the directed graph of the tree-like structure. It is also the valued graph and its components $c_i \in C$, $s_j \in S$ contain therefore the characteristics of the corresponding section graphs Γ_i.

The vertices and the edges of the graph Π can be arranged hierarchically in a way shown in Fig.4b, similar to that applied earlier for the graph Γ.

To implement it, there exist all conditions for organization of the DP process with subsequent finding of an optimal solution of the problem (1)-(3).

This process can be build on the graph Π by using the method that is similar to the immediate optimization method. The solution procedure here moves from step to step from the generalized edges of lower levels until the optimal solution for the source of the graph Π is found . In each step p, we solve the sub-problems related to the optimization of edges of the graph Π of the level $y^1 = p$ together with their branches of the preceding levels whose optimal solutions have already been found in the previous steps.

One reduction of the recursive relation is carried out in every step.

3.2: Recursive Relation

In its general form, the recursive relation is related with the arbitrary step p in which the optimality of derivations of Π for the level $y^1 = p$ is examined. For every of them, a number of the following sub-problems is solved:

For $\forall \xi = 1,2,...,D,$ *find:*

$$F_K(\xi) = \min_{\xi,\psi} \{f_j^*(\xi,\psi) + \sum_{k \in Y_{y-1}^1} F_L(\psi)\}; \qquad (14)$$

subject to :

$$b_l(\psi) \quad \le b_K(\xi \qquad (15)$$

$$\psi = 1,2,\dots,D$$

$$k, j \in \mathbf{Y}_y^1, \quad l \in \mathbf{Y}_{y-1}^1$$

where: $f^*_J(\xi,\psi)$ is the solution of the optimization problem that is associated with the section graph $\Gamma_J(V_{J,},E_J)$:

Minimize:

$$Z = \min \sum_{j \in E_J} f_j(x_j \qquad (16)$$

subject to

$$\sum_{j \in E_J} a_{ij} g_j(x_j) \le b_K(\xi) - b_l(\psi) \quad i = 1,2,\dots,m_J \qquad (17)$$

$$x_j \ge 0 \ , \ x_j \in \mathbf{D} \ ,$$

where: Y^l_y and $Y^1{}_{(y-1)}$ are the sub-sets of vertices and edges of the graph Π for the level y^1 and y^1-1, respectively ; l , K are the initial and the final vertices of the edge J of the graph Π; $b_l{}^{(\xi)}$ and $b_K(\psi)$ are the components of B, connected with the vertices of this edge, where $b_k(\psi)$ is renewed in the course of the previous steps realization. The process of renewal in these steps takes place automatically when solving the problems (16),(17).

Sum in (14) unites the optimal solutions $F_L(\psi)$ that are related to the edges of the level y^1-1 , which are adjacent to the final vertex of the edge J. These solutions are found in the previous steps.

The optimal values of $\psi^*(\xi)$ and $F_K(\xi)$, and also the renewed value of $b_K{}^{(\xi)} = b_K(\xi)$, are stored in tables for use in subsequent steps of calculations.

In development, the recurrent use of (16),(17) results in a possibility of constructing the direct steps algorithm.

The recurrent relation takes its trivial form in the first step that consists in solving the sub-problems (14),(15) alone. It is related with the section graphs Γ_J condensed in the generalized edges adjacent to the sinks of the graph Π.

In the last step too, this relation is distinguished by one peculiarity, since ξ is unique and $b_K(\xi) = b_{(N+1)}(\xi) = b_{n+1}$ is known and it makes it possible to obtain the definitive solution for edges of higher levels and vertices of preceding levels. Further, the procedure of inverse steps has conditions for being developed.

Use of inverse steps makes it possible to determine the optimal parameters of the points of seperation with the aim of finding optimal solutions for every section graph, and the problem can be solved in a similar way by parts depending on sizes of the section graphs Γ_J (J=1,2,...,N).

In solving the sub-problems (16),(17) use is made of the DP algorithm, and, as a result, we have one algorithm of DP process the built into another one.

3.3: Multiple Decomposition

The use of DPM at both levels of solution of the problem (1)-(3) offers a possibility of realizing two types of decomposition : the geometric partition and the recurrent partition of problem.

With the geometric partition, the process is carried out in one step on which the graph Γ is partitioned once and for all into the section graphs Γ_J (J=1,2,...,N) of necessary sizes. After that, the problem is solved by using the DP process as applied to the relations (14),(15),and also by solving the sub-problem (16),(17) with the use of immediate DP algorithm and the recursive relations (4)-(6).

The recurrent partition is in its essence of hierarchical character. It is supposed that the sub-problem (16),(17) can be partitioned, in turn, into more smaller sub-problems by introducing so the second level of partition; these problems can be solved again either immediately or by parts, and so on. The process of partition can be then repeated recurrently by building hierarchically, in such a manner, one DP process into another one until the sub-problem (16),(17) of required size be obtained. In the limiting case of recursive partition the section graphs Γ_J coincide with the initial graph edges. In these cases, the algorithm of partition is converted into the immediate DP algorithm connected with the relations (4)-(6).

These facts make it possible to conclude to suppose that there exist, for this type of partition , same optimal sizes of the section graphs.

Besides, the process of implementation for both types of partition is always of multi-variant character, even in the presence of section graph sizes predetermined.

There appears thus the problem of partition optimization where we determine a number, size, and structure of the section graphs Γ_J (J=1,2,...,N). Optimal partition may be submit to criterion of optimization of memory resources, or time resources, or be the combination of both.

The problem of partition optimization needs more thorough and deeper specific studies, its discussion is here omitted, and we leave it for consideration of future researchers.

CONCLUSIONS

Based on the dynamic programming method, we developed the algorithm for immediate solution and the technique of optimization by parts for the separable nonlinear programming problem as applied to its tree-like graph.

Within the limits of dynamic programming algorithm, significant use is made of the graph structure and its natural hierarchy. On this basis the recursive relation of immediate optimization process in its general form was obtained. Besides, its particular forms were obtained for use in the first and the last steps.

The technique of solution by parts uses partition of the problem graph by means of cuts in vertices . The condensation of section graphs obtained results in the generalized graph, the so-called graph of decomposition. In this paper the decomposition problem is solved with the use of the dynamic programming algorithm that is identical to that used in immediate optimization. One and the same hierarchical form is used here for the graph of decomposition, and the section graphs are treated as some generalized edges. For this case, we obtained the recurrent relation where was used, for solving sub-problems as applied to the section graphs, the algorithm for immediate dynamic programming. Thus, there takes place the solution by parts within the scope of one dynamic programming process built in the other .

The herein-presented paper discusses the problem of multiple decomposition, potentialities of geometric and recurrent partition, and also the problem of the partition optimization.

REFERENCES

[1] Hadely , G , Nonlinear and Dinamic Programming , Mir, Moscow , 1967,508p.

[2] Ravindran,A. , Philips, D.T., Solberg ,J.J. , Operation Reserch . Principles and Pratice . Second Edition, John Wiley&Sons 1986 , 637p.

[3] Vagner ,G , Osnovy Issledovanya Operatsiy., T.2, Mir, Moskva, 1973, 488p.

[4] Levin, M.S. , Muradyan,A.E. , Syrikh, N.N. , Katchestvo energii v setyah selskih rayonov, Energia , Moskva , 1975 , 224s .

[5] Dale, V.A. , Krishan Z.P. , Paegle O.G. , Dinamitcheskoe programmirovanie v raschetah razvitiya eletricheskih setey Zinatne , Riga, 1979 , 192 s.

[6] Hasilev,V.N. , Merenkov,A.P., Sumarocov,S.V. O vybore diametrov trub razvetvlennih teplovih setey s ispolzovaniem EVM , Tepoloenergetika , No 6, 1966, 60-65.

[7] Goryatchkin, V.P. , Tereshuk, V.S. , Tsoi,A.A., CAD Program Package for Automobile Electric Equipment Vestnik KGTU im. A.N.Tupoleva , 1996 . No2, 22-25

[8] Tsoi A.A. , Singularidade de realização algoritmica do método de otimização multiregime de redes elétricas de aviões . Colet. Interunivers. "Eletro-equipamentos[de Aparelhos Aeronauticos , Ed. KAI: Kazan, 1984.