EUR 600.e,f

EUROPEAN ATOMIC ENERGY COMMUNITY — EURATOM

EFFICIENT REALIZATION OF BOOLEAN FUNCTIONS BY MEANS OF DIODE CONJUNCTION MATRICES AND MINIMAL COST ASSEMBLING PROBLEMS

by

P. CAMION and L. VERBEEK

1964

Joint Nuclear Research Center
Ispra Establishment — Italy
Scientific Data Processing Center — CETIS

LEGAL NOTICE

This report can be obtained, at the price of Belgian Francs 40, from : PRESSES ACADEMIQUES EUROPEENNES — 98, Chaussée de Charleroi, Brussels 6.

Please remit payments to :

— BANQUE DE LA SOCIETE GENERALE (Agence Ma Campagne) — Brussels — account No 964.558,

— BELGIAN AMERICAN BANK and TRUST COMPANY — New York — account No 22.186,

— LLOYDS BANK (Europe) Ltd. — 10 Moorgate — London E.C.2,

giving the reference : « EUR 600 . e, f — EFFICIENT REALIZATION OF BOOLEAN FUNCTIONS BY MEANS OF DIODE CONJUNCTION MATRICES AND MINIMAL COST ASSEMBLING PROBLEMS ».

Printed by Vaillant-Carmanne, Liège.

Brussels, July 1964.

EUR 600.e,f

EFFICIENT REALIZATION OF BOOLEAN FUNCTIONS BY MEANS OF DIODE CONJUNCTION MATRICES AND MINIMAL COST ASSEMBLING PROBLEMS by P. CAMION and L. VERBEEK

European Atomic Energy Community — EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Scientific Data Processing Center (CETIS)
Brussels, July 1964 — 10 pages

In part I the realization of the 2^n Boolean functions of n binary variables given together with their negation by means of diode conjunction matrices is investigated. The different possible schemes of realization are compared in terms of the number of diodes involved. It is proved that partitioning of the n variables in two equal (if n is even) or almost equal (if n is odd) sets and continued partitioning of these sets according to the same rule until the sets obtained contain less than four variables results in the scheme involving the least number of diodes. This minimal number of diodes is given for n from 1 up to 20.

EUR 600.e,f

EFFICIENT REALIZATION OF BOOLEAN FUNCTIONS BY MEANS OF DIODE CONJUNCTION MATRICES AND MINIMAL COST ASSEMBLING PROBLEMS by P. CAMION and L. VERBEEK

European Atomic Energy Community — EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Scientific Data Processing Center (CETIS)
Brussels, July 1964 — 10 pages

In part I the realization of the 2^n Boolean functions of n binary variables given together with their negation by means of diode conjunction matrices is investigated. The different possible schemes of realization are compared in terms of the number of diodes involved. It is proved that partitioning of the n variables in two equal (if n is even) or almost equal (if n is odd) sets and continued partitioning of these sets according to the same rule until the sets obtained contain less than four variables results in the scheme involving the least number of diodes. This minimal number of diodes is given for n from 1 up to 20.

In part II a general assembling problem, containing the problem posed in section I.1 and I.2, is formalized in terms of schemes and following this particular example. This formalization leads to a solution recurrent on the number of inputs of a scheme. It is shown how a minimal cost disassembly problem can be treated in the same way. In this problem the cost function is such that the solution is not given by a foreseeable law.

In part II a general assembling problem, containing the problem posed in section I.1 and I.2, is formalized in terms of schemes and following this particular example. This formalization leads to a solution recurrent on the number of inputs of a scheme. It is shown how a minimal cost disassembly problem can be treated in the same way. In this problem the cost function is such that the solution is not given by a foreseeable law.

EUR 600.e,f

EUROPEAN ATOMIC ENERGY COMMUNITY — EURATOM

EFFICIENT REALIZATION OF BOOLEAN FUNCTIONS BY MEANS OF DIODE CONJUNCTION MATRICES AND MINIMAL COST ASSEMBLING PROBLEMS

by

P. CAMION and L. VERBEEK

1964

Joint Nuclear Research Center
Ispra Establishment — Italy
Scientific Data Processing Center — CETIS

EFFICIENT REALIZATION OF BOOLEAN FUNCTIONS BY MEANS OF DIODE CONJUNCTION MATRICES AND MINIMAL COST ASSEMBLING PROBLEMS

1 — THE SITUATION

Given are n Boolean variables and their negations, $x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n$, in the form of n pairs of electric leads. The 2^n Boolean functions can be realized by a scheme as given in figure 1, where each pair of horizontal lines represents a variable and its negation; each vertical line represents one of the functions, and each circle on the crossing of two leads indicates a diode connecting these leads. We will refer to such a scheme as a diode conjunction matrix. A complete scheme for $n = 2$ is given in figure 2, here the functions realized on the vertical lines are as follows :

Line	Function
1	$x_1 x_2$
2	$\bar{x}_1 x_2$
3	$x_1 \bar{x}_2$
4	$\bar{x}_1 \bar{x}_2$

The realization of the 2^n Boolean functions of n variables by means of such a conjunction matrix involves for each function (i.e. each vertical line) n diodes, one for each variable or its negation (i.e. one for each pair of horizontal lines). The total number of diodes used is thus :

$$\text{(1)} \qquad N(n\,;n) = n.2^n$$

Another method to realize the 2^n functions consists in dividing the n variables in two parts of m_1 and m_2 variables, respectively, where $m_1 + m_2 = n$. Then constructing two conjunction matrices analogous to the scheme of figure 1, one realizing the 2^{m_1} functions of the m_1 variables, the other realizing the 2^{m_2} functions of the m_2 variables, and combining these two sets of functions in pairs of one of each set so as to form the $2^{m_1}.2^{m_2} = 2^n$ functions of the n variables. Figure 3 gives an example of this method for $m_1 = 2$ and $m_2 = 2$.

This last method can be generalized by dividing the n variables in two or more than two parts. The realization of the 2^n functions consists then in the following three steps :

1) Partition the n variables in k parts ($k \geqslant 2$) of $m_1, m_2, \ldots, m_k$ variables, respectively, with $m_1 + m_2 + \ldots m_k = n$.

2) For each i ($i = 1, 2, \ldots, k$) form the 2^{m_i} Boolean functions by means of a diode conjunction matrix. Clearly this involves $m_1.2^{m_1} + m_2.2^{m_2} + \ldots + m_k.2^{m_k}$ diodes.

3) Combine in a final conjunction matrix the partial results thus obtained.

This final matrix has on each of its 2^n vertical lines k diodes, each of these on the intersection with one horizontal line of each of the k batches coming from the k matrices formed under 2). Hence this final matrix involves $k.2^n$ diodes. The total number of diodes involved in this scheme is :

(2) $$N(n\,;\,m_1, m_2, \ldots, m_k) = \sum_{i=1}^{k} [m_i.2^{m_i}] + k.2^n., \text{ with } \sum_{i=1}^{k} m_i = n.$$

2 — THE PROBLEMS

The different schemes for the realization of the Boolean functions suggest practical questions about the number of diodes involved. More precisely, if n is given one can ask :

a) Does partitioning of the n variables give rise to schemes using less diodes than in case of no partitioning ?

b) If so, which partitioning involves the least number of diodes ? If partitioning reduces the number of diodes necessary for the realization, hence continued partitioning of the sets obtained upon the first partitioning is favorable, then one is led to ask :

c) Which type of continued partitioning leads to the least number of diodes necessary to realize the total scheme.

d) What is the number of diodes involved in this scheme.

3 — THE SOLUTIONS

The answers to the questions posed in section 2 can be obtained by setting up a specific case of partitioning n in two sets and comparing the number of diodes necessary to realize the 2^n Boolean functions in this particular case with the number necessary in other cases. The specific partitioning is suggested by inspection of the number of diodes necessary in the different possible cases for $n = 1, 2, 3, 4, 5$, and 6 calculated and listed in Appendix 1.

If n is even, take $m_1 = m_2 = \frac{n}{2}$. Substitution in (2) yields the number of diodes used to realize the functions with this method :

$$N\left(n\,;\frac{n}{2}, \frac{n}{2}\right) = n.2^{\frac{n}{2}} + 2.2^n$$

In case no partitioning is used the number of diodes involved is given by :

(1) $$N(n\,;\,n) = n.2^n$$

Now for $n \geqslant 4$ it is clear that

$$\frac{n}{n-2} < 2^{\frac{n}{2}}$$

and this yields

$$n.2^{\frac{n}{2}} + 2.2^n < n.2^n.$$

If n is odd, take $m_1 = \frac{n+1}{2}$, and $m_2 = \frac{n-1}{2}$. Substitution in (2) gives :

$$N\left(n\,;\frac{n+1}{2}, \frac{n-1}{2}\right) = (3n+1).2^{\frac{n-3}{2}} + 2.2^n.$$

For $n \geqslant 5$ it is clear that

$$\frac{3n+1}{n-2} < 2^{\frac{n+3}{2}},$$

and this results in

$$(3n+1).2^{\frac{n-3}{2}} + 2.2^n < n.2^n$$

These calculations justify an affirmative answer to question a of section 2, i.e. the partitioning of n in two equal, or almost equal, sets of variables yields a scheme using less diodes than in the case of no partitioning.

In order to answer question b of section 2, two situations have to be considered, one in which the partitioning is in two sets such that m_1 and m_2 are not equal to $\frac{n}{2}$ (if n even) or to $\frac{n+1}{2}$ and $\frac{n-1}{2}$ (if n odd), the other in which the partitioning is in more than two sets.

If n is even take $m_1 = \frac{n}{2} + r$, $m_2 = \frac{n}{2} - r$ with $1 \leqslant r < \frac{n}{2}$. Substitution in (2) yields :

$$\mathrm{N}\left(n\,;\frac{n}{2}+r, \frac{n}{2}-r\right) = \left(\frac{n}{2}+r\right).2^{\frac{n}{2}+r} + \left(\frac{n}{2}-r\right).2^{\frac{n}{2}-r} + 2.2^n.$$

$$= \frac{n}{2}.2^{\frac{n}{2}}[2^r + 2^{-r}] + r.2^{\frac{n}{2}}[2^r - 2^{-r}] + 2.2^n$$

$$> 2.\frac{n}{2}.2^{\frac{n}{2}} + 2.2^n = \mathrm{N}\left(n\,;\frac{n}{2}, \frac{n}{2}\right).$$

This last inequality follows from the fact that

$$2^r + 2^{-r} > 2 \text{ and } 2^r - 2^{-r} > 0 \text{ if } r \geqslant 1.$$

If n is odd take $m_1 = \frac{n+1}{2} + r$, $m_2 = \frac{n-1}{2} - r$ with $1 \leqslant r < \frac{n-1}{2}$.

Substitution in (2) yields :

$$\mathrm{N}\left(n\,;\frac{n+1}{2}+r, \frac{n-1}{2}-r\right) = \left(\frac{n+1}{2}+r\right).2^{\frac{n+1}{2}+r} + \left(\frac{n-1}{2}-r\right).2^{\frac{n-1}{2}-r} + 2.2^n$$

$$= \frac{n-1}{2}.2^{\frac{n-1}{2}}[2^{1+r} + 2^{-r}] + 2^{\frac{n-1}{2}}.2^{1+r} + r.2^{\frac{n-1}{2}}[2^{1+r} - 2^{-r}] + 2.2^n$$

$$> \frac{n-1}{2}.2^{\frac{n-1}{2}}.3 + 2.2^{\frac{n-1}{2}} + 2.2^n$$

$$= \left(\frac{n-1}{2}+1\right)2^{\frac{n-1}{2}}.2 + \frac{n-1}{2}.2^{\frac{n-1}{2}} + 2.2^n$$

$$= \frac{n+1}{2}.2^{\frac{n+1}{2}} + \frac{n-1}{2}.2^{\frac{n-1}{2}} + 2.2^n$$

$$= \mathrm{N}\left(n\,;\frac{n+1}{2}, \frac{n-1}{2}\right).$$

The inequality here follows from the fact that

$$2^{1+r} + 2^{-r} > 3 \text{ and } 2^{1+r} > 2 \text{ if } r \geqslant 1.$$

We can conclude that partitioning of n in two parts, such that $m_1 = m_2 = \frac{n}{2}$ if n is even, and $m_1 = \frac{n+1}{2}$, $m_2 = \frac{n-1}{2}$ if n is odd, leads to realizations with less diodes than any other partitioning in two parts.

Now we have to consider the case of partitioning in more than two parts, say in k parts with $k \geqslant 3$. The number of diodes involved in such a case is given by (2) :

$$N(n\,;m_1, m_2, \ldots, m_k) = \sum_{i=1}^{k} [m_i . 2^{m_i}] + k . 2^n > 3 . 2^n = 2^n + 2 . 2^n$$

If n is even :

$$N\left(n\,;\frac{n}{2}, \frac{n}{2}\right) = n . 2^{\frac{n}{2}} + 2 . 2^n \leqslant 2^n + 2 . 2^n \text{ for } n \geqslant 4.$$

If n is odd :

$$N\left(n\,;\frac{n+1}{2}, \frac{n-1}{2}\right) = \frac{n+1}{2} . 2^{\frac{n+1}{2}} + \frac{n-1}{2} . 2^{\frac{n-1}{2}} + 2 . 2^n$$

$$= 2^{\frac{n-1}{2}} \left[\frac{3n+1}{2}\right] + 2 . 2^n < 2^{\frac{n-1}{2}} . 2^{\frac{n+1}{2}} + 2 . 2^n = 2^n + 2 . 2^n$$

for $n \geqslant 5$.

Hence we can conclude that the partitioning in two equal (if n is even) or almost equal (if n is odd) parts yields a scheme using less diodes than a scheme with partitioning in more than two parts.

In the proof of this last conclusion no use is made of the number of diodes involved in realizing the functions of the sets of m_i variables. The fact that $k \geqslant 3$ and at least one diode is involved in realizing the functions of the sets of m_i variables is sufficient to write

$$N(n\,;m_1, m_2, \ldots, m_k) > 3 . 2^n \text{ for } k \geqslant 3.$$

This consideration enables us to state that whatever further subdivision is used in realizing the sets of m_i variables always more diodes are involved than in case the partitioning is in two equal or almost equal sets. Hence also in case of further subdivision this last method leads to the use of the least number of diodes. This conclusion answers question c of section 2 and can be stated as follows :

The realization of the 2^n Boolean functions of n variables which are given together with their negation, by means of diode conjunction matrices involves the least number of diodes if the n variables are partitioned in two sets containing $\frac{n}{2}$ and $\frac{n}{2}$ variables if n is even, and $\frac{n+1}{2}$ and $\frac{n-1}{2}$ variables if n is odd, and each of these sets likewise is partitioned in two sets, and so on, until the number of variables in each set is smaller than four.

In answer to question d of section 2, we remark that a closed formula expressing the number of diodes necessary to realize the Boolean functions of n variables cannot be given in an easy calculable form. The reason being that the numbers depend on how many factors 2 are contained in n. Moreover, for given n the procedure is very easy as is indicated in the following example. Take $n = 18$, this number can be partitioned in steps :

$$18 = 9 + 9 = (5 + 4) + (5 + 4) = (3 + 2) + (2 + 2) + (3 + 2) + (2 + 2)$$

The number of diodes necessary to realize this scheme is :

$$N_{min}(18) = (24 + 8 + 2.2^5) + (8 + 8 + 2.2^4) + (24 + 8 + 2.2^5) + (8 + 8 + 2.2^4) + 2.2^9 + 2.2^9 + 2.2^{18} = 526.624$$

In Appendix 2 these numbers of diodes involved in the realization with the most efficient scheme is listed for $n = 1$ through 20.

4 — REMARK

The problem as given in section 1 and 2 of this report was posed to one of the authors by Dr. W. Becker, Service Electronique, Ispra.

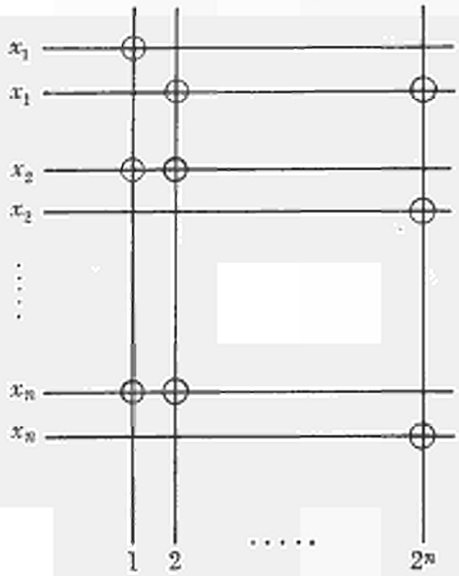

Fig. 1. — Diode conjunction matrix realizing 2^n Boolean functions of n variables

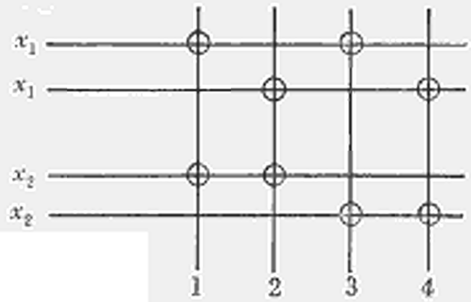

Fig. 2. — Diode conjunction matrix for $n = 2$

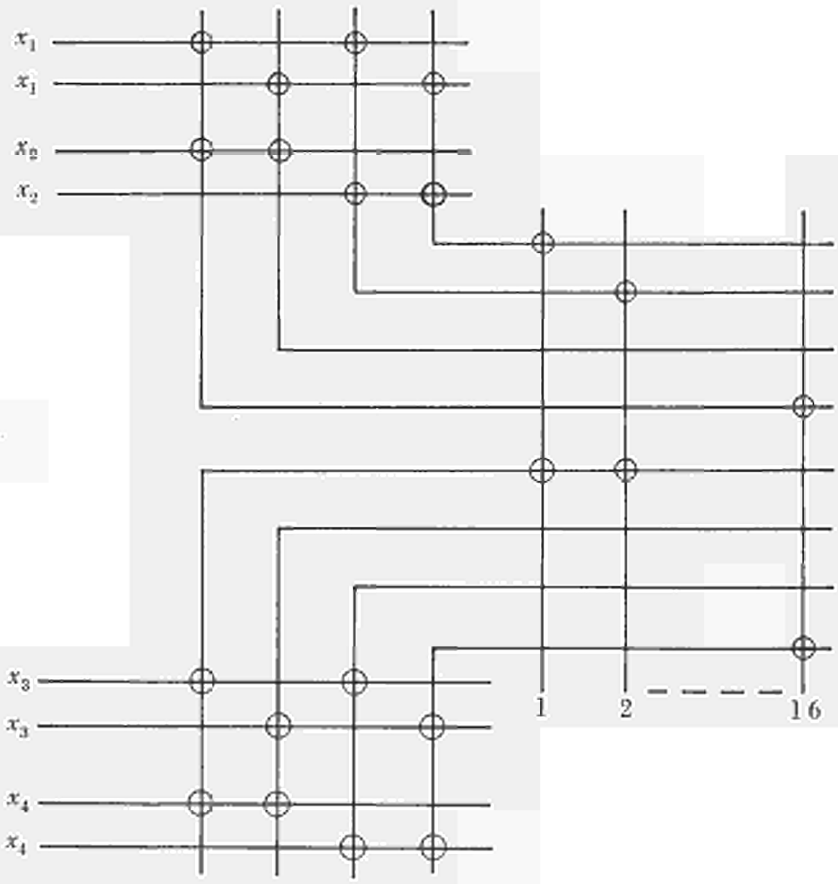

Fig. 3. — Diode conjunction matrices for $n = 4$ with partitioning in two sets of two variables

APPENDIX 1

N(1 ; 1)	= **2**
N(2 ; 2)	= **8**
N(2 ; 1, 1)	= 12
N(3 ; 3)	= **24**
N(3 ; 2, 1,)	= 36
N(3 ; 1, 1, 1)	= 30
N(4 ; 4)	= 64
N(4 ; 3, 1)	= 58
N(4 ; 2, 2)	= **48**
N(4 ; 2, 1, 1)	= 60
N(4 ; 1, 1, 1, 1)	= 72
N(5 ; 5)	= 160
N(5 ; 4, 1)	= 130
N(5 ; 3, 2)	= **96**
N(5 ; 3, 1, 1)	= 124
N(5 ; 2, 2, 1)	= 114
N(5 ; 2, 1, 1, 1)	= 142
N(5 ; 1, 1, 1, 1, 1)	= 170
N(6 ; 6)	= 384
N(6 ; 5, 1)	= 290
N(6 ; 4, 2)	= 200
N(6 ; 4, 1, 1)	= 260
N(6 ; 3, 3)	= **176**
N(6 ; 3, 2, 1)	= 226
N(6 ; 3, 1, 1, 1)	= 286
N(6 ; 2, 2, 2)	= 216
N(6 ; 2, 2, 1, 1)	= 276
N(6 ; 2, 1, 1, 1, 1)	= 336
N(6 ; 1, 1, 1, 1, 1, 1)	= 396

APPENDIX 2

n	$N_{min}(n)$	n	$N_{min}(n)$
1	2	11	4.368
2	8	12	8.544
3	24	13	16.888
4	48	14	33.424
5	96	15	66.472
6	176	16	132.288
7	328	17	263.920
8	608	18	526.624
9	1.168	19	1.051.984
10	2.240	20	2.101.632

ASSEMBLAGES DE MOINDRES COUTS

1 — A chaque entier positif n (nombre d'éléments d'input), correspond un ensemble S_n de schémas réalisés selon toutes les méthodes données en I.1 et I.2. Le coût $f(s)$ d'un schéma s de S_n est le nombre de diodes utilisées dans ce schéma. Nous pouvons ordonner les schémas de S_n selon leur coût, de sorte que à chaque entier positif α corresponde un schéma s_α tel que à chaque schéma corresponde au moins un entier et que

(1) $$\alpha^\times > \alpha^{\times\times} \Rightarrow f(s_{\alpha^\times}) \geqslant f(s_{\alpha^{\times\times}}).$$

Puisque $| S_n |$ (le nombre d'éléments de S_n) est fini, il existe un entier m tel que

(2) $$\alpha > m \Rightarrow f(s_\alpha) = f(s_m)$$

De cette façon, à l'ensemble des couples d'entiers (α, n) correspond biunivoquement l'ensemble des schémas définis en 1 et 2. Nous désignerons donc un schéma par le couple (α, n).

2 — On appelle *partition* (au sens de la théorie des nombres) une application univoque σ d'un ensemble $I = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$ d'entiers >0 dans l'ensemble des entiers >0 ; il est d'usage de noter la partition qui fait correspondre à $\lambda_1, \lambda_2, \ldots, \lambda_s$ les nombres $\gamma_1, \gamma_2, \ldots, \gamma_s$, par :

(3) $$\sigma_I = (\lambda_1^{\gamma_1}, \lambda_2^{\gamma_1}, \ldots, \lambda_s^{\gamma_s}).$$

On appelle *poids* de la partition σ l'entier

(4) $$p(\sigma_I) = \gamma_1\lambda_1 + \ldots + \gamma_s\lambda_s$$

La *dimension* de la partition σ est l'entier

(5) $$d(\sigma_I) = \gamma_1 + \ldots + \gamma_s.$$

Soit $\mathscr{I}_n$ la famille de tous les ensembles I_n d'entiers positifs tels que il existe une partition σ_{I_u} de I_n de poids n. Un schéma (α, n) est réalisé par assemblage de $d(\sigma_{I_n})$ sous-schémas. Soit :

(6) $$\{(\beta_i, j_{I_n}(i)) \mid 1 \leqslant i \leqslant d(\sigma_{I_n})\}$$

un tel ensemble de sous-schémas. Remarquons que α peut être exprimée comme une fonction

$$\theta(\beta_1, \ldots, \beta_{d(\sigma_{I_n})}, j_{I_n}(1), \ldots, j_{I_n}(d(\sigma_{I_n})))$$

où

$$\beta_1, \ldots, \beta_{d(\sigma_{I_n})}$$

sont des *variables positives entières indépendantes*. Soit $\varphi(\alpha, n)$ le coût du schéma (α, n), il est clair que si

(7) $$\varphi(\alpha, n) = \Psi(\beta_1, \ldots, \beta_{d(\sigma_{I_n})}, j_{I_n}(1), \ldots, i_{I_n}(d(\sigma_{I_n})))$$

est monotone non décroissante en $\beta_1, \ldots, \beta_{d(\sigma_{I_n})}$, donc si

(8) $$\beta_i^\times > \beta_i^{\times\times} \Rightarrow \Psi(\beta_1, \ldots, \beta_i^\times, \ldots, \beta_{d(\sigma_{I_n})}, j_{I_n}(1), \ldots, j_{I_n}(d(\sigma_{I_n}))) \geqslant$$
$$\Psi(\beta_1, \ldots, \beta_i^{\times\times}, \beta_{d(\sigma_{I_n})}, j_{I_n}(1), \ldots, j_{I_n}(d(\sigma_{I_n}))).$$

On aura :

(9) $$\min_{\beta_1, \dots, \beta_{d(\sigma_{I_n})}} \Psi(\beta_1, \dots, \beta_{d(\sigma_{I_n})}, j_{I_n}(1), \dots, j_{I_n}(d(\sigma_{I_n}))) = \Psi(1, \dots, 1, j_{I_n}(1), \dots, j_{I_n}(d(\sigma_{I_n}))).$$

Pour trouver (1, n), donc le schéma de S_n de coût minimum, il suffira pour toute partition σ_{I_n} de $I_n \varepsilon \mathscr{I}_n$ d'envisager le seul assemblage de l'ensemble de sous-schémas qui sont le moins coûteux :

(10) $$\{(1, j_{I_n}(i)) \mid 1 \leqslant i \leqslant d(\sigma_{I_n})\}$$

et de retenir celui du coût minimum. Dans le problème posé, on a

(11) $$\varphi(\alpha, n) = \sum_{1 \leqslant i \leqslant d(\sigma_{I_n})} [\varphi(\beta_i, j_{I_n}(i))[+ d(\sigma_{I_n}) . 2^n$$

$\varphi(\alpha, n)$ satisfait les conditions (7) et (8) à cause de la manière par laquelle les β_i sont définis dans la section II.1, et on peut calculer $\varphi(1, n)$ par récurrence sur n. On a pu calculer à la main en un temps raisonnable $\varphi(1, n)$ pour $n = 1, \dots, 6$ qui sont donnés dans l'Appendix 1 de la partie I.

3 — Un problème d'assemblage qui sera résolu par la même méthode est celui du démontage d'un ponton. Il existe des chantiers sur la rive qui désassemblent les bateaux et les tronçons de ponts sur bateau. Une opération de désassemblage consiste à désunir deux tronçons, le coût de désassemblage peut être calculé et on suppose qu'il dépend seulement des longueurs des deux tronçons à séparer, c'est $f(n_1, n_2)$.

On voit que, quelle que soit cette fonction, le problème est du type traité ci-dessus, n est le nombre de tronçons élémentaires, donc le nombre de bateaux. I sera ici un ensemble de deux entiers positifs de somme égale à n, ou d'un seul entier égal à $\frac{n}{2}$.

(12) $$\min_{\alpha} \varphi_\alpha(n) = \min_{n_1 + n_2 = n} [\min_{\beta_1} \varphi_{\beta_1}(n_1) + \min_{\beta_2} \varphi_{\beta_2}(n_2) + f(n_1, n_2)]$$

Le tableau suivant donne pour $1 \leqslant n \leqslant 10$ la longueur optimale des tronçons n_1 et n_2 avec $n_1 + n_2 = n$ lorsque le coût de désassemblage est $f(n_1, n_2) = \sqrt{n_1} + \sqrt{n_2}$.

n	n_1	n_2	Coût total d'assemblage minimum ou $\min_\alpha \varphi_\alpha(n)$
	1	0	0
2	1	1	2
3	2	1	$3 + \sqrt{2}$
4	2	2	$4 + 2\sqrt{2}$
5	3	2	$5 + 2\sqrt{2} + \sqrt{3}$
6	4	2	$8 + 3\sqrt{2}$
7	4	3	$9 + 3\sqrt{2} + \sqrt{3}$
8	4	4	$12 + 4\sqrt{2}$
9	5	4	$11 + 4\sqrt{2} + \sqrt{3} + \sqrt{5}$
10	6	4	$14 + 5\sqrt{2} + \sqrt{6}$

On voit que le schéma optimum de désassemblage pour $n = 10$ est donné par l'arbre :