

Knolmayer, Gerhard

Working Paper — Digitized Version

Computational experiments in the formulation of large-scale linear programs

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 89

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Knolmayer, Gerhard (1980) : Computational experiments in the formulation of large-scale linear programs, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 89, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/190922>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Nr. 89

COMPUTATIONAL EXPERIMENTS IN THE FORMULATION
OF LARGE-SCALE LINEAR PROGRAMS

BY

GERHARD KNOLMAYER

Paper presented at the Workshop on Large-Scale Linear Programming, organized by the International Institute for Applied System Analysis (IIASA) at Laxenburg, Austria

Kiel, December 1980

Professor Dr. Gerhard Knolmayer
Institut für Betriebswirtschaftslehre,
Christian-Albrechts-Universität zu Kiel
Olshausenstraße 40-60, 2300 Kiel

Computational Experiments in the Formulation
of Large-Scale Linear Programs

Abstract:

One of the decisions in the construction of a linear program is upon the formulation which should be used. This paper explains why there is usually a very large number of equivalent formulations and reports on the computational behavior of these formulations. The usual textbook-hypothesis which claims that CPU-time increases with the cube of the number of constraints is falsified; it is shown that the advantage of reducing the number of rows may be overcompensated by an increase in the number of nonzeros.

COMPUTATIONAL EXPERIMENTS IN THE FORMULATION OF
LARGE-SCALE LINEAR PROGRAMS

1. DECISIONS IN BUILDING A DECISION MODEL

Several (meta-) decisions have to be made in the construction of a decision model:

- Which section of reality should be modelled ?
- How accurate should one model this section of reality ?
- Which algorithm should be used ?
- Which people, computer, software should be employed ?
- Which formulation should be used ^{to represent} ~~for~~ a given degree of accuracy ⁱⁿ ~~of~~ the model ?

The implementation of different answers ~~on~~ these questions will result in different benefits and costs (of the decision model). The ultimate benefit of modelling is to gain insight into reality. In more detail one could distinguish between

- Benefit from ~~model~~ accuracy
- Benefit of the ease of understanding
 - the formulation
 - the solution-
 - the model.

On the other hand one can partition the costs of decision models into

- Costs of model construction
- Costs of collecting data
- Costs of manipulating data
- Costs of computation.

Many computational experiments have been performed in mathematical programming (MP). Most research has concentrated upon the comparison of algorithms and codes. Recently

the need for research on a methodology of formulating MP-models has been expressed /18/.

Most computational experiments compare "costs", usually by giving CPU-time. Sometimes costs and benefits are compared, e.g. if the "quality" of solutions obtained is compared to the CPU-time needed for exact and heuristic algorithms. From this point of view one can distinguish ~~the~~ ^{with care} four areas of computational experiments shown in Fig. 1. These areas have been investigated to a very different extent. This paper concentrates on a cost-comparison of equivalent formulations by using a production code for linear programming (LP).

Object of comparison \ Type of comparison	Cost-Comparison	Cost-Benefit-Comparison
	Algorithms (Codes)	
Formulations	X	

Fig. 1: Types of experiments and topic of the paper (X)

2. THE NEED TO STUDY EQUIVALENT FORMULATIONS

We define equivalent formulations as models from which identical optimal activity levels can be derived (by using a report writer); the optimal values of the objective functions coincide. Several researchers have compared two equivalent formulations for linear or mixed-integer problems; I make references to the well-known studies of H.P.Williams /17;19/ on (mixed-) integer models and to the confrontation of linear product-mix-models with a "normal" resp. "aggregated" technological matrix /12;14,p.148-157;16,p.27-82/. Such comparisons suffer from the fact that often not only

two but ~~plenty~~ of equivalent formulations exist. Especially if MP-models are generated from data bases containing information on ~~every-day~~ operation the model builder has to decide which of the equivalent formulations should be generated. This decision determines the computational effort for matrix generation and for optimization.

In principle one can define basic relations ~~from~~ the data base as activities of the LP model and connect these activities by balance equations. But often the so emerging model will be unsolvable by production codes due to an enormous number of balance equations. A product-mix-model for a manufacturing firm with 400 final and 10000 intermediate products, with 30000 materials, 300 capacities and an average number of 5 operations for the manufactured products would need 82301 rows and 82400 structurals! Therefore it is desirable to generate ~~a~~ ^{several possible move} compact models by eliminating balance equations. Fig. 2 shows a ~~small out~~ ^{for} of the very large number of equivalent LP-models that can be generated from a data base. In Fig. 2 the size of the model is measured by the number of rows. Data manipulation looks highly attractive from the usual textbook hypothesis that CPU-time grows with the cube of the number of rows /cf. e.g. 1, p.83; 3, p.16; 5, p.146; 6, p.181; 15, p.118; 20, p.10/. Few authors claim that CPU-time is ^{also} influenced by the density of the model, ~~120~~ /11, p.57; 14, p.190/. ^{In the usual case} ~~by eliminating balance~~ equations usually the number of rows is reduced and the density rises. ^{give} Therefore rules of thumb are wanted which ^{give} information about ^{the} presumable effects of matrix condensation. To support the decisions in model construction two types of experiments are necessary:

- Experiments ^{on} of generating MP-models out of (non-specialized) data bases
- Experiments on the optimization behavior of equivalent formulations.

This paper reports on the second type of experiments.

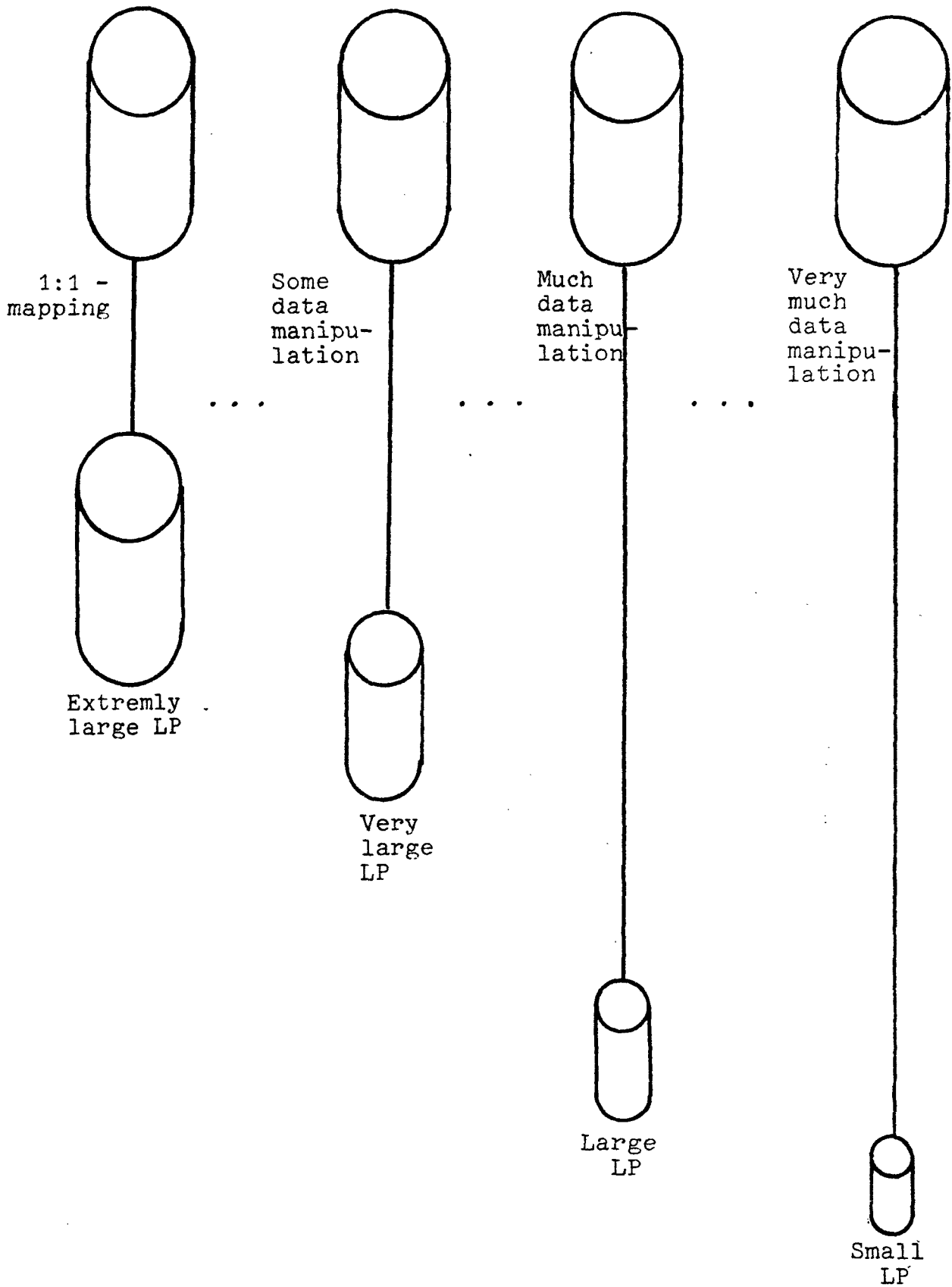


Fig.2: Different ways of matrix generation result in equivalent LP-models of different size

3. THE ELIMINATION OF BALANCE EQUATIONS

3.1. THEORY

Let

$$(1) \quad \begin{aligned} c' x &\rightarrow \max ! \\ A^{M1} x &= b^{M1} > 0 \\ A^{M2} x &= 0 \\ x &\geq 0 \end{aligned}$$

be a feasible LP with slack and surplus, but without artificial variables. The indices of the constraints i form the set $M=M1 \cup M2$. Rows $i \in M2$ are called balance equations. Let $M2=M21 \cup M22$. We search for a transformed LP with new variables \tilde{x}

$$(2) \quad \begin{aligned} c' T \tilde{x} &\rightarrow \max ! \\ A^{M1} T \tilde{x} &= b^{M1} \\ A^{M2} T \tilde{x} &= 0 \\ T \tilde{x} &\geq 0 \end{aligned}$$

which is equivalent to (1) but computationally more appropriate. The latter requirement might be achieved if

$$A^{M22} T = 0 .$$

In this case $|M22|$ rows can be dropped as redundant.

If the original formulation (1) contains $p=|M2|$ balance equations there are at least 2^p equivalent formulations! To overcome the problems due to this enormous number of equivalent formulations we restrict the discussion to those formulations which arise by a sequential elimination of balance equations. The sequence can be determined heuristically, ~~by some~~ ^{A ~~one~~} plausible criterium, In the sequential procedure we have

$$T = \begin{matrix} |M22| \\ \Pi \\ i=1 \end{matrix} T_i$$

where T_i is the transformation matrix for the i -th elimination of a balance equation.

It remains to determine matrices T_i in such a way that (1) and (2) are equivalent. After elimination of $f-1 < p$ balance equations there exists a row $k \in M_2$ so that

$$\bar{A}^k = A^k \prod_{i=1}^{f-1} T_i .$$

Let $\bar{POS}(k) = \{j \mid \bar{a}_{kj} > 0\}$ and $\bar{NEG}(k) = \{j \mid \bar{a}_{kj} < 0\}$. These sets are nonempty if there are no null variables. \bar{x}_r ($r \in \bar{POS}(k)$) can be positive if and only if at least one \bar{x}_j ($j \in \bar{NEG}(k)$) is positive. This "If-then"-relation allows $\bar{x}_r > 0$ and $\bar{x}_s > 0$ ($s \in \bar{NEG}(k)$) implicitly by a "coupled activity" $\bar{x}_u > 0$. The coefficients of the coupled activity are computed as

$$\bar{a}_{iu} = g (\bar{a}_{ir} \cdot |\bar{a}_{ks}| + \bar{a}_{is} \cdot |\bar{a}_{kr}|)$$

so that variable u has a zero in row k . g is an arbitrary positive factor; in the subsequent text we assume $g=1$.

All possible activity levels of the prior formulation can be expressed by $|\bar{POS}(k)| \cdot |\bar{NEG}(k)|$ coupled activities. After the transformation all variables \bar{x}_j ($j \in \bar{POS}(k) \cup \bar{NEG}(k)$) can be deleted. In the matrix

$$T_f = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & |\bar{a}_{ks}| \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & |\bar{a}_{kr}| & \dots \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

there are unity column vectors for the untouched activities and two non-zeros in those columns which represent coupled

activities. We have $T_f \geq 0$ and therefore $T \geq 0$. Furthermore we get modified sets:

$$\bar{M}_{21} = \bar{M}_{21} - \{k\}$$

$$\bar{M}_{22} = \bar{M}_{22} + \{k\}.$$

The transformation reduces the number of rows by one. The effect on the number of legitimate variables depends on the number of positive and negative coefficients in row k :

$$\bar{n} = \bar{n} + |\bar{POS}(k)| - |\bar{NEG}(k)| - |\bar{POS}(k)| - |\bar{NEG}(k)|$$

Table 1 shows how the number of legitimate (=non-artificial) variables changes with the sign of the non-zeros in the eliminated balance equation. The effects of condensation on model structure are illustrated in Table 2 and Fig. 3 for a refinery model given by Meyer-Steinmann /10, p.390-393/. The points ~~on the right hand of~~ ^{x and y in} Fig. 3 characterize the original formulation; the effect of sequential data manipulation on problem structure is shown by going to the left. Activity coupling reduces the number of rows far more than e.g. the REDUCE-module of APEX-III.

From an economic point of view one can describe the condensation by the isoquant given in Fig. 4. It might happen that both ~~formulations compared in literature~~ ^{of these} are unsolvable on the system used while some ~~equivalent formulations might~~ ^{other} be computationally well suited. The isoquant must be read from right to left.

$\bar{POS}(k)$	$\bar{NEG}(k)$	Number of		Net Effect
		new legitimate variables	legitimate variables deleted	
3	1	3	4	-1
2	2	4	4	0
7	4	28	11	17

Table 1: Effects of eliminating a balance equation k

rows	Number of structural structurals variables nonzeros			nonzeros	density	⁹ structural density
70	76	146	323	393	3.84540 %	6.07143 %
69	75	144	321	390	3.92512 %	6.20290 %
68	74	142	315	383	3.96645 %	6.25994 %
67	73	140	313	380	4.05117 %	6.39951 %
66	72	138	307	373	4.09530 %	6.46044 %
65	71	136	301	366	4.14027 %	6.52221 %
64	70	134	295	359	4.18610 %	6.58487 %
63	69	132	289	352	4.23280 %	6.64826 %
62	68	130	283	345	4.28040 %	6.71752 %
61	67	128	281	342	4.38012 %	6.87546 %
60	66	126	279	339	4.48413 %	7.04545 %
59	65	124	277	336	4.59267 %	7.22295 %
58	64	122	275	333	4.70605 %	7.40841 %
57	63	120	273	330	4.82456 %	7.60734 %
56	62	118	271	327	4.94855 %	7.80530 %
55	61	116	269	324	5.07837 %	8.01788 %
54	60	114	267	321	5.21442 %	8.24074 %
53	59	112	265	318	5.35714 %	8.47458 %
52	58	110	263	315	5.50699 %	8.72016 %
51	57	108	261	312	5.66449 %	8.97833 %
50	56	106	259	309	5.83019 %	9.25000 %
49	55	104	257	306	6.00471 %	9.53618 %
48	54	102	255	303	6.18873 %	9.83796 %
47	53	100	253	300	6.38298 %	10.15655 %
46	52	98	251	297	6.58829 %	10.49331 %
45	51	96	249	294	6.80556 %	10.84967 %
44	50	94	247	291	7.03578 %	11.22727 %
43	49	92	245	288	7.28008 %	11.62791 %
42	48	90	243	285	7.53968 %	12.05357 %
41	47	88	241	282	7.81596 %	12.50649 %
40	46	86	239	279	8.11047 %	12.98913 %
39	45	84	237	276	8.42491 %	13.50427 %
38	44	82	235	273	8.76123 %	14.05502 %
37	43	80	233	270	9.12162 %	14.64489 %
36	42	78	231	267	9.50855 %	15.27778 %
35	41	76	229	264	9.92481 %	15.95819 %
34	40	74	227	261	10.37361 %	16.69118 %
33	39	72	224	257	10.81650 %	17.40482 %
32	38	70	222	254	11.33929 %	18.25658 %
31	37	68	218	249	11.81214 %	19.00610 %
30	36	66	214	244	12.32323 %	19.81491 %
29	35	64	210	239	12.87716 %	20.68966 %
28	34	62	206	234	13.47926 %	21.63866 %
27	33	60	213	240	14.81481 %	23.90572 %
26	32	58	220	246	16.31300 %	26.44231 %
25	32	57	240	265	18.59649 %	30.00000 %
24	32	56	261	285	21.20536 %	33.98438 %
23	32	55	282	305	24.11067 %	38.31522 %
22	32	54	303	325	27.35690 %	43.03977 %
21	32	53	316	337	30.27853 %	47.02781 %
20	32	52	329	349	33.55769 %	51.40625 %
19	4E	65	497	516	41.78138 %	56.86493 %
18	70	88	853	871	54.98737 %	67.69841 %
17	116	133	1375	1392	61.56568 %	69.72617 %
16	240	256	3075	3091	75.46387 %	80.07813 %

Table 2: Effects of condensation for the refinery model /10, p. 390 - 393/

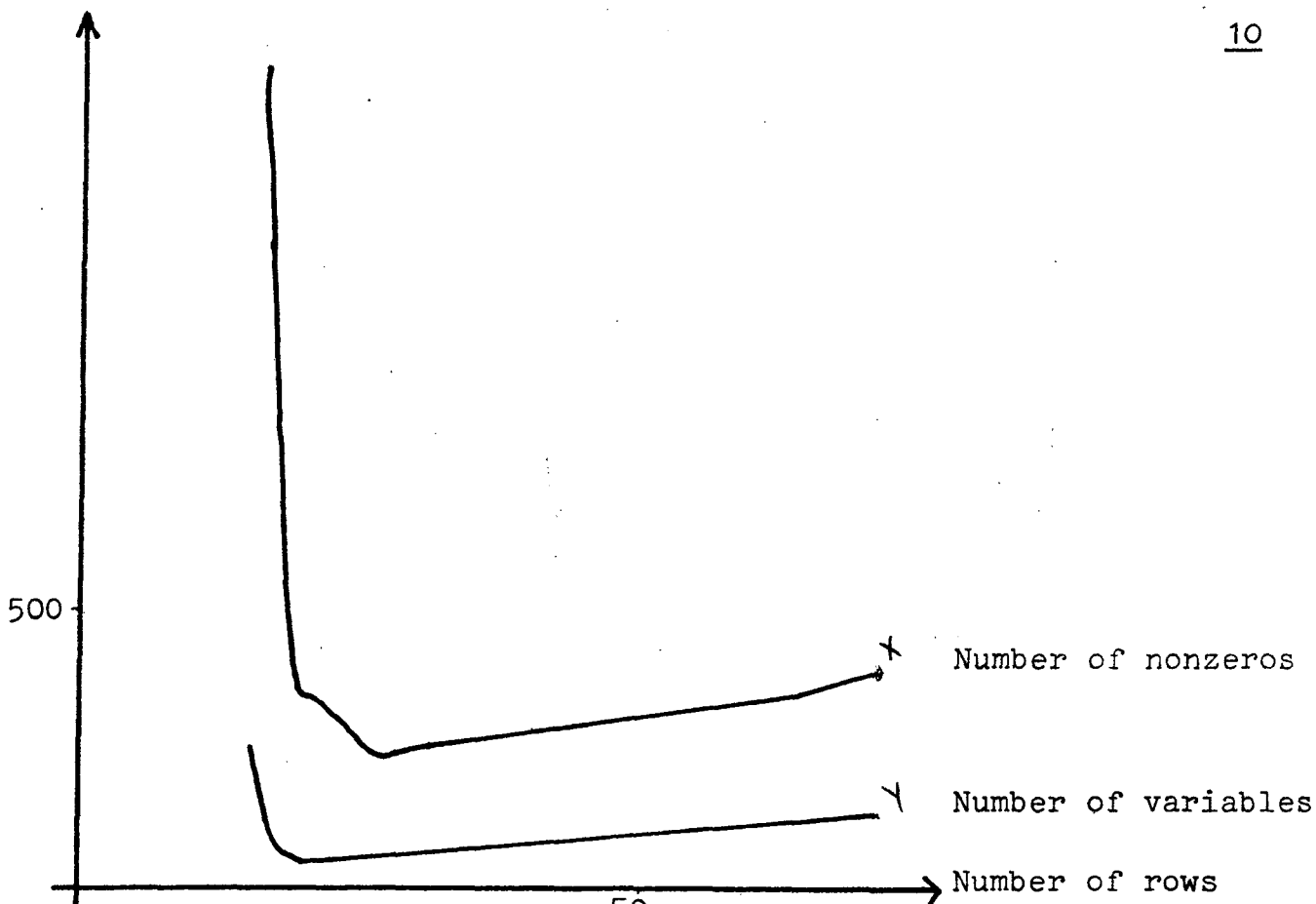


Fig.3: Effects of condensation⁵⁰ for the refinery model /10, p.390-393/

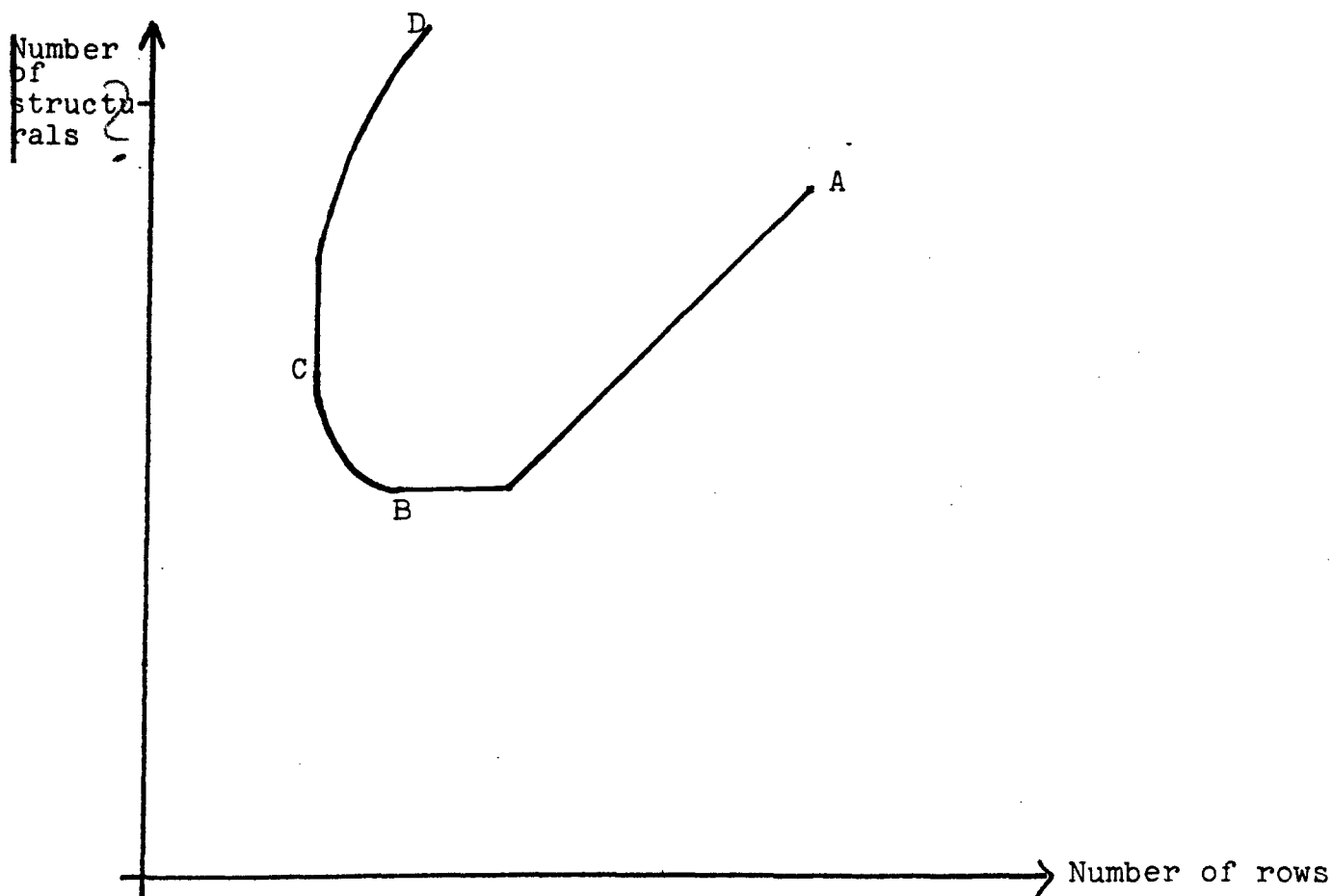


Fig.4: Isoquant for equivalent formulations

In economic theory only the part BC of the isoquant would be regarded as efficient. ~~In this connection the~~ ^{Section} part AB is efficient too because ~~it takes resources to go from A to B.~~ ^{it costs} ~~The part CD is explained by the reason that~~ ^{are used} by eliminating a balance equation one or more bounds can become regular rows; this part of the isoquant is inefficient.

As soon as a formulation (2) is reached which is regarded computationally well suited the optimal levels of the activities \tilde{x}° are determined. The optimal values of the original variables can be computed by

$$(3) \quad x^\circ = T \tilde{x}^\circ .$$

3.2. AN EXAMPLE

Consider a problem in which two final products x_1 and x_2 are produced by using a part, which can be either purchased (x_3) or produced (x_4):

$$\begin{array}{rll} \text{Max.} & 500 x_1 + 1000 x_2 - 200 x_3 - 150 x_4 & \\ \text{s.t.} & 2 x_1 + 1 x_2 + 1 x_4 \leq 1000 & \\ & 4 x_2 + 2 x_4 \leq 2000 & \\ & 1 x_1 + 4 x_2 - 1 x_3 - 1 x_4 = 0 & \\ & x_j \geq 0 \quad \text{all } j & \end{array}$$

The first two constraints represent capacities, the third is the balance equation for the part. The definitions

- \tilde{x}_1 ... quantity of final product 1 produced by using parts purchased
- \tilde{x}_2 ... quantity of final product 1 produced by using parts produced by the firm

\tilde{x}_3 ... quantity of final product 2 produced by using parts purchased

\tilde{x}_4 ... quantity of final product 2 produced by using parts produced by the firm

allow the formulation

$$\begin{aligned} \text{Max.} \quad & 300 \tilde{x}_1 + 350 \tilde{x}_2 + 200 \tilde{x}_3 + 400 \tilde{x}_4 \\ \text{s.t.} \quad & 2 \tilde{x}_1 + 3 \tilde{x}_2 + 1 \tilde{x}_3 + 5 \tilde{x}_4 \leq 1000 \\ & 2 \tilde{x}_2 + 4 \tilde{x}_3 + 12 \tilde{x}_4 \leq 2000 \\ & \tilde{x}_j \geq 0 \quad \text{all } j. \end{aligned}$$

Formally such a reformulation can be obtained by multiplying the original coefficient matrix with the transformation matrix

$$T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 1 & 0 & 4 \end{bmatrix}$$

We have $M2 = M22 = \{3\}$, $A^{M22} T = (1 \ 4 \ -1 \ -1)$. $T=0$ and $\bar{n} = 4 + 2 \cdot 2 - 2 - 2 = 4$. The optimal solution for the condensed LP is $\tilde{x}^* = (250 \ 0 \ 500 \ 0)'$. Optimal levels of the original variables can be determined by $x^* = T\tilde{x}^* = (250 \ 500 \ 2250 \ 0)'$.

The different paths through the networks in Fig. 5 show that a general series transformation is employed for eliminating a balance equation.

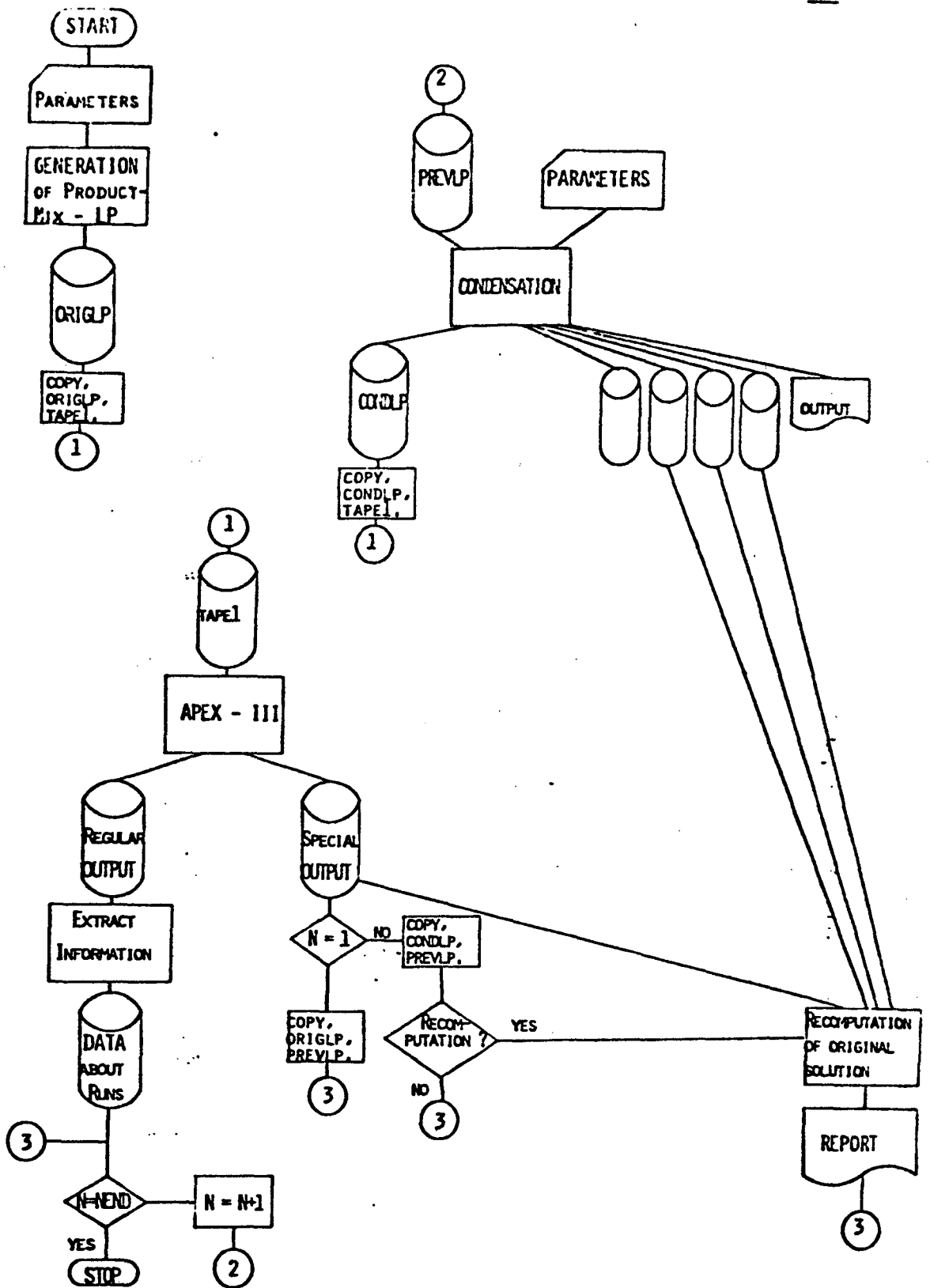


Fig. 6: Flow of information in computational experiments

generator PPPGEN was written in FORTRAN to create LP-models of product-mix-type. The user specifies the type of model to be created (~~in much detail~~) by setting 16 scalar and 3 vector parameters. One set of parameters generates different LP-problems with very similar but not identical structures by use of random numbers.

2. Preprocessing

A FORTRAN-program performs the transformations discussed above. The user controls the order in which the balance equations are selected for elimination by 7 parameters. This selection is based on an estimation of the number of additional non-zeros an elimination might create.

3. Optimization

Optimization was done by the in-core-system BASE-APEX-III using the standard parameters (except LOG=1) on a CYBER 74 under NOS/BE. The reported CPU-time was needed for optimization only. (The maximal deviation of CPU-time due to multiprogramming is only about 1 % on the system used.)

4. Postprocessing

APEX-III produces an FORTRAN-accessible file which was used to determine the optimal levels of the activities in the original formulation. This postprocessing is based on (3) although the matrix T was not computed explicitly.

5. Recording Information about Optimization

The regular OUTPUT-file of APEX-III contains information which is necessary to analyse the optimization behavior. This output-file was read by a program which recorded the structure of the model and the specifics of the solution process.

6. Regression Analysis

The data collected in step 5 were examined by regression analysis. First the exponents of the variables in various

regression models were determined by SPSS' module for non-linear regression. The results were used to define transformed variables for a "linear" regression through the origin. Several hypotheses on the dependence of optimization time on model structure were compared by the coefficient of determination, R^2 .

7. ~~Control~~ Experiments ~~to~~

A test developed by Hoel /7/ ~~was used~~ to compare the best regression equation against the textbook-hypothesis

A more detailed description of the experiments and the program lists are given in /9/.

5. RESULTS OF THE COMPUTATIONAL EXPERIMENTS

Four problem classes and four problem sizes for each problem class have been examined. For each of the 16 cases 5 models were generated. Three problem classes were used to develop an appropriate explanation for the CPU-time observed; problem class 4 was used to control the results. Table 3 shows the approximate structure of the models in largest size. For problems of smaller size the figures in Table 3 have to be reduced by 25%, 50% and 75%.

All 80 formulations were condensed in five steps. In these steps a balance equation was eliminated if not more than a certain number of additional non-zeros were expected to arise. For problem class 3 and largest size Table 4 shows the effects of these condensations. The optimization was done by the procedures CRASH and PRIMAL of BASE-APEX-III. All formulations were optimized using constant field length $RFL, 100000_8 (=32768$ decimal words of 60 bits each). Optimization time was reduced remarkably in the first phases of the condensation but in latter phases the condensation did not pay off.

	Number of rows/columns in problem class			
	1	2	3	4
	R O W S			
Objective function	1	1	1	1
Capacity constraints	25	25	25	25
Balance equations for final products	45	45	45	45
Balance equations for intermediate products	200	430	200	430
Balance equations for materials	250	20	250	20
	521	521	521	521
C O L U M N S				
Sales variables for final products	45	45	45	45
Sales variables for intermediate products	~20	~43	~20	~43
Purchase variables for intermediate products	~20	~43	~20	~43
Purchase variables for materials	250	20	250	20
Production variables for 45 final products	~135	~135	~59	~59
Production variables for intermediate products	~600	~1290	~260	~559
	~1070	~1576	~654	~769

Table 3: Structures of product-mix-models generated

Form. #	Max. add. nz	Average number of			Aver. CPU- time	CPU - time estimated by	
		rows	columns	nonzeros		textbook	"best" regression
1	-	521	651	3009	40.8	59.4	35.0
2	-5	351	481	2040	26.0	18.2	18.8
3	0	185	315	1619	8.4	2.7	7.8
4	30	144	286	1953	7.5	1.3	6.1
5	100	137	294	2288	10.8	1.1	6.0
6	1000	134	338	3206	10.4	1.0	6.5

Table 4: Effects of condensation in problem class 3

The data obtained from 317 LPs belonging to problem classes 1, 2 and 3 were analysed by regression models. Table 5 compares the quality of fit for several hypotheses and some other plausible equations.

Regression equation	Explaining variables proposed by	R ²
(4) .00000042 m ³	/e.g. 1;3;5;6;15;20/	.701
.0627 m	/8/	.867
.000000015 n ³	/4;13/	.532
.0081 m ^{1.36}		.889
.0147 n ^{1.05}		.766
.0381 nz ^{.70}		.697
(5) .0010 m ^{1.25} nz ^{.33}		.916
.0015 m ^{1.14} n ^{.45}		.913
.0293 n ^{2.53} nz ^{-1.24}		.816
.00085 m ^{2.35} [nz/(m.n)] ^{.86}		.912
(6) .00094 m ^{1.29} n ^{-.17} nz ^{.44^a}		.916

Table 5: Comparison of regression models for explaining CPU-time in problem classes 1 to 3

The improvement of (6) over (5) is so small that

$$(7) \quad \text{PREDCPU} = a m^b nz^c$$

is regarded as most suitable. For this model the approximate 95% confidence intervals for the exponents are computed in the non-linear regression by SPSS as

$$1.15 \leq b = 1.25 \leq 1.34$$

$$.26 \leq c = .33 \leq .39$$

Although these intervals ~~are~~ ^{may seem to be} tight they lead to rather wide confidence intervals for CPU-time.

The new assumption (5) was compared ~~with~~ ^{to} the established hypothesis (4) via a test developed by Hoel /7/. This test

leads to a linear regression of type

$$(8) \quad (\text{OBSCPU} - \text{ESTHYP}) = w \cdot (\text{NEWHYP} - \text{ESTHYP})$$

for additional data. The usefulness of NEWHYP is confirmed if w is significantly positive. Regression (8) gives a coefficient $w=1.37$ for 111 cases belonging to problem class 4; the 95% confidence interval is $w \geq 1.22$. The t -value for regression (8) is 15.11. This value can be compared with the one-sided value for 95% and $DF=110$ which is 1.66. The scattergram in Fig. 7 shows that in 93 of 111 cases the signs of the differences in (8) are identical. Therefore the new formula (5) predicts significantly better than the established hypothesis (4).

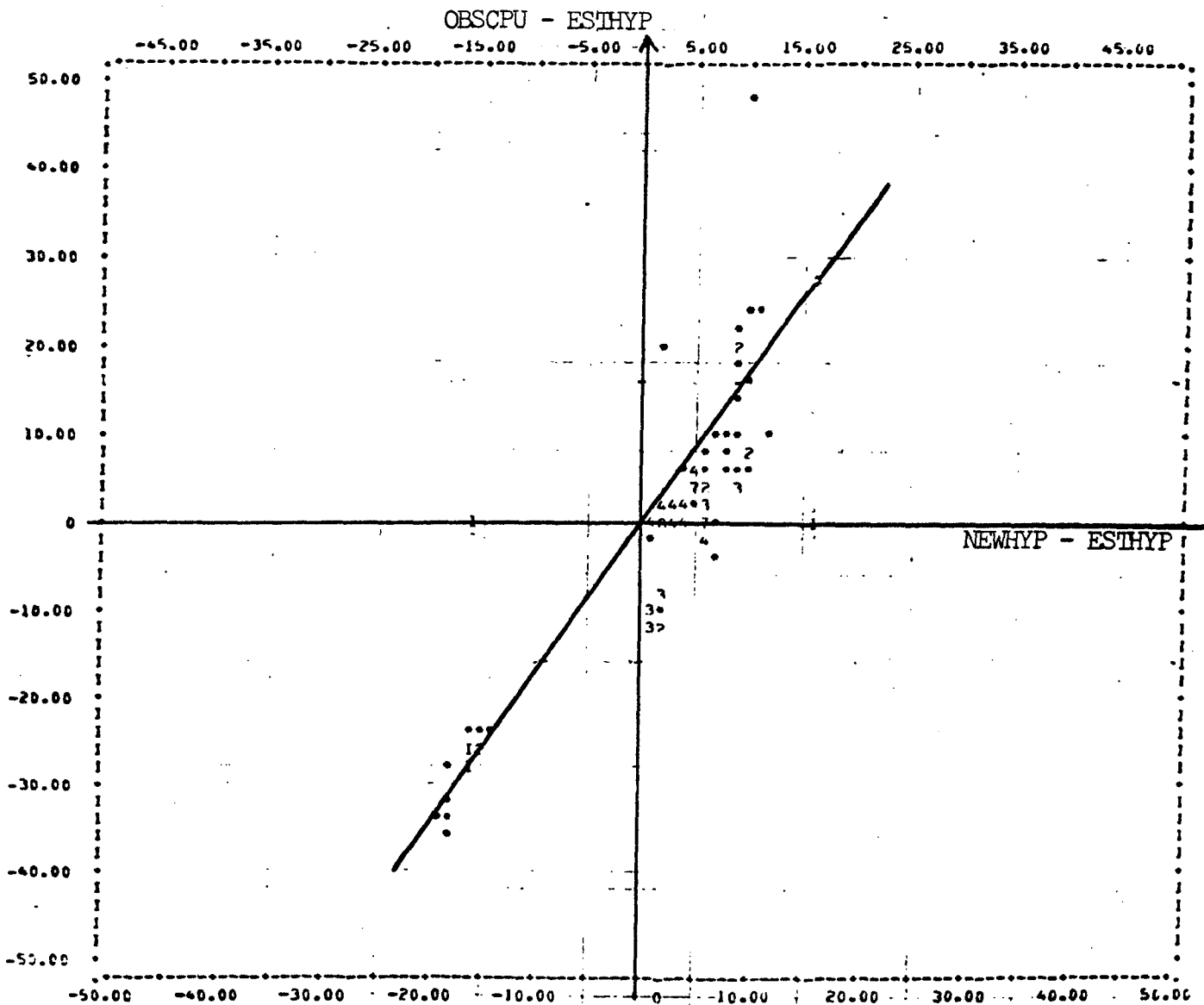


Fig. 7: Comparison of the predictions from (4) and (5) with observed CPU-time

6. SUMMARY

An identical LP-optimum usually can be obtained by many equivalent problem formulations. Data condensation is necessary if large models are generated from data bases containing information about ^{day to day} every-day-operations. Most textbooks recommend ~~to reduce~~ the number of rows as much as possible. Our experiments show that the usual m^3 -hypothesis is misleading and should be ^{taken out of} ~~cancelled from~~ textbooks. The experiments described above indicate that the number of nonzeros has remarkable influence on computational effort. The rule given by E.M.L. Beale /1, p.83/ that it is normally not worth saving a row by substituting a variable if this adds more than about half a dozen nonzeros, remains useful in the light of our experimental results. The number of nonzeros may rise if the number of structurals is reduced. Taking into account the effort for matrix generation, one might propose an even ^{easier} rule of thumb:

"Eliminate balance equations only if

- the model is so large that the number of rows is a burden ~~to~~ ⁱⁿ the computational environment used
- the number of structurals is reduced by the elimination and the number of nonzeros ^{AD} rises only slightly."

For product-mix-models this rule suggests ~~to use~~ ^{using} balance equations for products which have more than one way of preparation (e.g. make or buy; manufacturing variants) and more than one way of utilization (e.g. sell or process). Thus if options are available a "combined" formulation is recommended which ~~differs from both formulations compared in literature.~~ ^{is intermediate between those usually presented in the literature.}

If the resulting model is still too large the following actions could be ~~taken into mind:~~ ^{considered:}

- ~~May the problem be solved easily~~ ^{Can they} by ~~codes~~ ^{more} with GUB-facilities ~~and can such a code be made amenable?~~ ^{using}

- Should one define ~~in the first (i-1) LP only options~~ ^{these} ~~which are expected to be optimal and generate for~~ ^{are defined as variables, then the next} ~~optimization run (i+1) new variables for options~~ ^{should be generated,} which improve the solution of run ~~i~~ ^{the previous}. These candidates can be determined by the ~~i-th~~ ^{using previous} dual solution.
- Is it possible to develop better algorithms for dense LP-problems ?

If all these questions have to be ~~denied~~ there is an effective "solution constraint" on the LP originally proposed. In this case one must take into account the potential benefits of ~~differently~~ ^{less} accurate models and judge whether a ~~less accu-~~ ^{would} ~~rate model will~~ ^{sufficient} allow enough insight into the real-world ~~problem, that it pays to develop this less accurate model.~~

MAIN SYMBOLS

DF	Degrees of freedom
ESTHYP	CPU-time predicted by the established hypothesis (4)
m	number of rows
M1	set of indices i with $b_i \geq 0$
M2	set of indices i with $b_i = 0$ (balance equations)
M21 \subseteq M2	set of indices i for balance equations not eliminated
M22 \subseteq M2	set of indices i for balance equations eliminated
n	number of structurals
nz	number of non-zeros
NEG(k)	set of indices j with $a_{kj} < 0$
NEWHYP	CPU-time predicted by the ^{kj} new assumption (5)
OBSCPU	Observed CPU-time
POS(k)	set of indices j with $a_{kj} > 0$
PREDCPU	Predicted CPU-time
R ²	Coefficient of determination
RFL	Requested Field Length
SET	number of elements in a SET
T, T _i	Transformation matrices

REFERENCES

- /1/ Beale, E.M.L., Mathematical Programming in Practice, Pitman Publishing: London et al. 1968
- /2/ Beale, E.M.L., The current algorithmic scope of mathematical programming systems, in: Mathematical Programming Study 4, ed. by M.L.Balinski and E. Hellerman, North-Holland: Amsterdam 1975, 1-11
- /3/ Bradley, S.P., Hay, A.C., Magnanti, T.L., Applied Mathematical Programming, Addison-Wesley: Reading et al. 1977
- /4/ Bramsemann, R., Controlling, Gabler-Verlag: Wiesbaden 1978
- /5/ Driebeek, N.J., Applied Linear Programming, Addison-Wesley: Reading et al. 1969
- /6/ Hillier, F.S., Lieberman, G.J., Operations Research, 2nd ed., Holden-Day: San Francisco et al. 1974
- /7/ Hoel, P.G., On the choice of forecasting formulas, in: Journal of the American Statistical Association 1947, Vol. 42, 605-611
- /8/ Holm, S., Klein, D., Size Reduction of Linear Programs with Special Structure, Skrifter fra Institut for Historie og Samfundsvidenskab, Odense Universitet: Odense 1975
- /9/ Knolmayer, G., Programmierungsmodelle für die Produktionsprogrammplanung, Ein Beitrag zur Methodologie der Modellkonstruktion, Birkhäuser-Verlag: Basel-Boston-Stuttgart, to appear.
- /10/ Meyer, M., Steinmann, H., Planungsmodelle für die Grundstoffindustrie, Physica-Verlag: Würzburg-Wien 1971
- /11/ Mitra, G., Theory and Application of Mathematical Programming, Academic-Press: London-New York-San Francisco 1976
- /12/ Müller-Merbach, H., Switching Between Bill of Material Processing and the Simplex Method in Certain Linear Large-Scale Industrial Optimization Problems, in: Decomposition of Large-Scale Problems, ed. by D.M.Himmelblau, North-Holland/American Elsevier: Amsterdam-London-New York 1973, 189-199
- /13/ Shah, A.A., Saber, J.C., A Transformation to Improve Computing in Linear Programs, Paper presented at the Joint National ORSA/TIMS Meeting at Las Vegas, November 1975
- /14/ Smith, D., Linear Programming in Business, Polytech Publishers: Stockport 1973
- /15/ Wagner, H.M., Principles of Operations Research, Prentice-Hall: Englewood Cliffs 1969
- /16/ Wiggert, H., Programoptimierung im Maschinenbau mit Hilfe der linearen Planungsrechnung, Beuth-Vertrieb: Berlin-Köln-Frankfurt am Main 1972
- /17/ Williams, H.P., Experiments in the formulation of integer programming problems, in: Mathematical Programming Study 2, ed. by M.L.Balinski, North-Holland: Amsterdam 1974, 180-197

- /18/ Williams, H.P., The Formulation of Mathematical Programming Models, in: Omega 1975, Vol. 3, 551-556
- /19/ Williams, H.P., The reformulation of two mixed integer programming problems, in: Mathematical Programming 1978, Vol. 14, 325-331
- /20/ Williams, H.P., Model Building in Mathematical Programming, Wiley-Interscience: Chichester et al. 1978