stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 26/73 JULY

H.W. LENSTRA, JR

THE ACYCLIC SUBGRAPH PROBLEM

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS(MOS) subject classification scheme (1970): 05C20, 90B35, 90C10, 90C20

Abstract

The acyclic-subgraph problem can be stated as follows: given a finite directed graph in which a non-negative weight is assigned to each edge, determine an acyclic subgraph of maximum weight. We give several mathematical formulations of this problem and indicate some applications, such as the ordering of input-output matrices. The graph-theoretical implications of the problem and the suboptimal and optimal algorithms, proposed in the literature, are discussed. Finally, we describe a branch-and-bound method for the problem.

BW 26/73

Errata

page	line	
6	10	* $i,j\in I,i\neq j$ → $i,j\in I,i<j$ *
8	9	* Strater → Slater *
11	5	* $I_j = \emptyset$ → $I_j \neq \emptyset$ *
13	12	* donominator → denominator *
13	17	* $(j,k) \in I$ → $(j,k) \in K$ *
28	2	* $g(\tau_{k-1},\rho)$ → $g(\tau_{k-1}\cdot\rho)$ *
32	4	* be → by *
35	5	* a lower → an upper *

Index

I INTRODUCTION

§1 Relations and graphs 1
§2 The acyclic subgraph problem 3
§3 Examples 8

II TOURNAMENTS

§4 Scores and decomposition of tournaments 10
§5 Reduction to strong tournaments 12
§6 Two lower bounds 16

III SUBOPTIMAL ALGORITHMS

§7 Ranking according to scores 23
§8 Two travelling-salesman methods 25
§9 Relative optimality 27

IV OPTIMAL ALGORITHMS

§10 Explicit enumeration 32
§11 Linear programming 34
§12 Dynamic programming 36
§13 The quadratic assignment problem 38
§14 Branch-and-bound methods 39

Literature 44

I. INTRODUCTION

§1. Relations and graphs

In this section we list some basic notions and notations.

(1.1) Relations. Let I be a set. A *relation* on I is a subset of the cartesian product I × I. For a relation R, we put

$$R^{-1} = \{(x,y) \mid (y,x) \in R\}.$$

Instead of $(x,y) \in R$ we also write xRy. For $x \in I$ we define

$$R[x] = \{y \in I \mid (x,y) \in R\};$$

so we have

$$R^{-1}[x] = \{y \in I \mid (y,x) \in R\}.$$

A relation R is called *antisymmetric* if $R^{-1} \cap R = \emptyset$. We say R is *complete* if

$$\forall x \in I : \forall y \in I : x \neq y \implies (x,y) \in R \cup R^{-1}.$$

R is called *transitive* if for all $x,y,z \in I$ the implication

$$(x,y) \in R \wedge (y,z) \in R \implies (x,z) \in R$$

is valid.

(1.2) Orderings. A transitive antisymmetric relation is called a *(partial) ordering*. A complete transitive antisymmetric relation is called a *total ordering*. We denote the set of total orderings on I by V_I. If I is finite, $|I| = n$, one has $|V_I| = n!$.

(1.3) Graphs. By a *graph* we understand a pair (I,R) where I is a set and R is a relation on I. The elements of I are the *vertices* of the graph, the elements of R are the *edges*. A graph (I,R) is called *finite* if I is finite. One gets a diagram of the finite graph (I,R) by considering I as a subset of the plane and drawing an arrow $\overset{i}{\cdot}\!\longrightarrow\!\overset{j}{\cdot}$ for every edge $(i,j) \in R$. We call (I,S) a *subgraph* of (I,R) if $S \subset R$. For an integer $n \geq 0$, a *path of length* n in the graph (I,R) is a sequence $(x_i)_{i=0}^{n}$ of elements of I, such that $(x_i,x_{i+1}) \in R$ for $0 \leq i < n$. If (I,R) is finite a *hamiltonian path* in (I,R) is a path $(x_i)_{i=0}^{|I|-1}$ of length $|I|-1$ for which $I = \{x_i \mid 0 \leq i < |I|\}$. A graph (I,R) is called *acyclic* if there does *not* exist a path $(x_i)_{i=0}^{n}$ of length $n > 0$ in (I,R) with the property $x_0 = x_n$. An acyclic graph (I,R) is called *maximum acyclic* if there is no acyclic graph (I,S) with $R \subsetneq S$. One simply proves:

(1.4) (I,R) is maximum acyclic if and only if R is a total ordering on I.

(1.5) Tournaments. A graph (I,K) is called a tournament if I is *finite* and K is a *complete antisymmetric* relation on I. A *subtournament* of (I,K) is a pair $(J,K\cap(J\times J))$ with $J \subset I$.

(1.6) From now on I denotes a *finite* set with $|I| = n$, $n > 0$. For $0 \leq k \leq n$, we write $S_{k,I}$ for the set of *injective* maps

$$\alpha : \{1,2,\ldots,k\} \to I.$$

S_n is the full permutation group on $\{1,2,\ldots,n\}$; so $S_n = S_{n,I}$ if $I = \{1,2,\ldots,n\}$.

§2. The acyclic subgraph problem

(2.1) Let (I,R) be a finite graph and $a : R \to \mathbb{R}_{\geq 0}$ a map from the set of edges to the set of non-negative real numbers. Let the weight of a subgraph (I,S) be defined by

$$\sum\nolimits_{(i,j)\in S} a(i,j).$$

Question: determine an *acyclic subgraph* of (I,R) of maximum weight.

Defining

$$a(i,j) = 0 \qquad\qquad \text{for } i,j \in I,\ (i,j) \notin R,$$

and replacing R by I × I obviously does not change the problem. Since all a(i,j) are non-negative, it is sufficient to look only at *maximum* acyclic graphs (I,S), cf. (1.3); that is, we need only consider total orderings S on I, cf. (1.4). So we found the following formulation for (2.1):

(2.2) Let I be a finite set and $a : I \times I \to \mathbb{R}$ a map. For $* \in V_I$ (see (1.2)) we put

$$f(*) = \sum\nolimits_{i,j\in I, i*j} a(i,j).$$

Problem: maximize $f(*)$ subject to $* \in V_I$, i.e. determine a total ordering # on I for which

$$f(\#) = \max\{f(*) \mid * \in V_I\}.$$

Observe that the a(i,j) are allowed to be negative in this formulation: adding a constant c to every a(i,j) increases each f(*) by $\binom{n}{2}$.c, $n = |I|$.

For any bijective map σ: {1,2,...,n} → I a total ordering * on I can be defined by

$$(2.3) \qquad i * j \Longleftrightarrow \sigma^{-1}(i) < \sigma^{-1}(j), \qquad\qquad i,j \in I.$$

Conversely, for any $*\in V_I$ a $\sigma \in S_{n,I}$ (cf. (1.6)) can be found for which (2.3) holds. Taking $I = \{1,2,\dots,n\}$ and putting $g(\sigma) := f(*)$ if (2.3) holds, we find that the following problem is equivalent to (2.2):

(2.4) Let $n \geq 1$ be an integer and let $(a_{ij})_{1\leq i,j\leq n}$ be an $n\times n$-matrix with real entries a_{ij}. For $\sigma \in S_n$ (cf. (1.6)) put

$$g(\sigma) = \sum_{1\leq i<j\leq n} a_{\sigma(i)\sigma(j)}.$$

Problem: determine a $\tau \in S_n$ for which

$$g(\tau) = \max\{g(\sigma) \mid \sigma \in S_n\}.$$

The next formulation follows immediately.

(2.5) Let $n \geq 1$ be an integer, and let the matrix $(c_{ij})_{1\leq i,j\leq n}$ be defined by

$$\begin{array}{ll} c_{ij} = 1 & \text{if } i < j \\ c_{ij} = 0 & \text{else.} \end{array}$$

Let $(a_{ij})_{1\leq i,j\leq n}$ be an $n\times n$-matrix with real entries.
Exercise: maximize

$$\sum_{1\leq i,j\leq n} a_{\sigma(i)\sigma(j)}\, c_{ij}$$

subject to $\sigma \in S_n$.

This is a *quadratic assignment problem*.

Consider formulation (2.2), and let $* \in V_I$. Define x_{ij} for $i,j \in I$ by

$$(2.6)\quad \begin{cases} x_{ij} = 1 & \text{if } i * j \\ x_{ij} = 0 & \text{else.} \end{cases}$$

Then

(2.7) $x_{ij} \in \{0,1\}$ for all $i,j \in I$.

The *completeness* and the *antisymmetry* of $*$ are expressed by

(2.8) $x_{ij} + x_{ji} = 1$ for all $i,j \in I,\ i \neq j$,

(2.9) $x_{ii} = 0$ for all $i \in I$,

and, given (2.7), (2.8) and (2.9), the *transitivity* of $*$ is equivalent to

(2.10) $x_{ij} + x_{jk} + x_{ki} \leq 2$ for all $i,j,k \in I,\ i \neq j \neq k \neq i$.

We conclude that, conversely, for any system of numbers $(x_{ij})_{i,j\in I}$ satisfying (2.7), (2.8), (2.9) and (2.10), there exists a total ordering $*$ on I such that (2.6) holds for all $i,j \in I$, cf. (1.2).
So problem (2.2) can be rewritten as follows:

(2.11) Let $n \geq 1$ be an integer and $(a_{ij})_{1\leq i,j\leq n}$ an $n\times n$-matrix, with $a_{ij} \in \mathbb{R}$. Maximize

$$\textstyle\sum_{1\leq i,j\leq n} a_{ij}x_{ij}$$

subject to the constraints

$$\begin{array}{rll} x_{ij} \in \{0,1\} & & \text{for } 1 \leq i,\ j \leq n \\ x_{ii} = 0 & & \text{for } 1 \leq i \leq n \\ x_{ij} + x_{ji} = 1 & & \text{for } 1 \leq i < j \leq n \\ \left.\begin{array}{r} x_{ij} + x_{jk} + x_{ki} \leq 2 \\ x_{ik} + x_{kj} + x_{ji} \leq 2 \end{array}\right\} & & \text{for } 1 \leq i < j < k \leq n. \end{array}$$

In this way we exhibit the problem as a *discrete linear programming problem*.

Consider again formulation (2.2). Define a relation K' on I by

$$(i,j) \in K' \iff a(i,j) > a(j,i).$$

Obviously, this relation is antisymmetric. Extend K' to a *complete* antisymmetric relation K on I, by making a choice between

$$(i,j) \in K \quad \text{and} \quad (j,i) \in K$$

for every two $i,j \in I$ satisfying $a(i,j) = a(j,i)$, $i \neq j$. Define

$$b(i,j) = a(i,j) - a(j,i) \qquad \text{for } (i,j) \in K.$$

Then $b(i,j) \geq 0$ for $(i,j) \in K$, and for any $* \in V_I$ we have

$$f(*) = \sum_{i,j\in I,i\neq j} \max(a(i,j),a(j,i)) - \sum_{(i,j)\in K\setminus *} b(i,j).$$

This leads to the next version of our problem:

(2.12) Let (I,K) be a tournament (1.5) and $b : K \to \mathbb{R}_{\geq 0}$ a map. For $* \in V_I$ put

$$g(*) = \sum_{(i,j)\in K\setminus *} b(i,j).$$

Problem: determine a $\# \in V_I$ with

$$g(\#) = \min\{g(*) \mid * \in V_I\}.$$

In words: determine a total ordering which "resembles K as closely as possible". The case $b(i,j) = 1$ (for all $(i,j) \in K$) can be formulated thus:

(2.13) Let (I,K) be a tournament. Determine a $\# \in V_I$ with $|K \setminus \#|$ smallest possible.

That is: make K transitive by reversing as few edges as possible. For $* \in V_I$ and $\sigma \in S_{n,I}$ (cf. (1.6)) we can define $g(\sigma) := g(*)$ if

$$i > j \Longleftrightarrow \sigma(i) * \sigma(j) \qquad \text{for all } i,j \in \{1,2,\dots,n\}$$

(different from (2.3)). Then (2.12) transforms to:

(2.14) Let (I,K) be a tournament, and $b : K \to \mathbb{R}_{\geq 0}$ a map. For $\sigma \in S_{n,I}$ put

$$g(\sigma) = \sum_{1 \leq i < j \leq n, (\sigma(i),\sigma(j)) \in K} b(\sigma(i),\sigma(j)).$$

Problem: determine a $\tau \in S_{n,I}$ for which

$$g(\tau) = \min\{g(\sigma) \mid \sigma \in S_{n,I}\}.$$

Using formulation (2.2), we have for any $* \in V_I$:

(2.15) $$f(*) + f(*^{-1}) = \sum_{i,j \in I, i \neq j} a(i,j) = \text{constant}, \; *^{-1} \in V_I.$$

Hence:

(2.16) If one replaces "max" by "min" in (2.2), (2.4), (2.5) or (2.11) or "min" by "max" in (2.12), (2.13) or (2.14), one gets an equivalent problem.

Further one easily sees (cf. (2.12), (2.14)):

(2.17) In (2.2) and (2.4), (2.5), (2.11) one may assume

$$a(i,j) \geq 0, \quad a(i,j).a(j,i) = 0$$

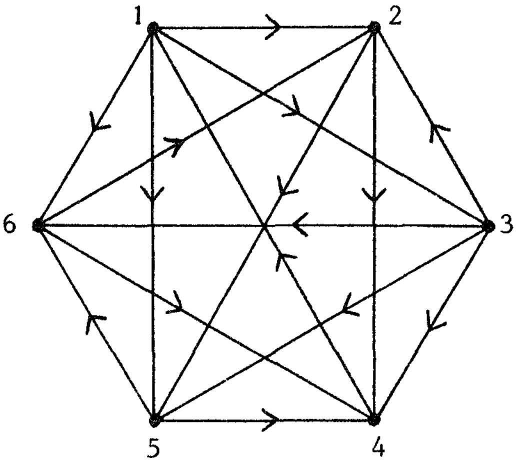
and

$$a_{ij} \geq 0, \quad a_{ij}.a_{ji} = 0$$

respectively, for all i,j.

§3. Examples

(3.1) *"The method of paired comparisons"*. Six dog foods have to be ranked according to taste. To this end one offers each of the fifteen possible pairs to a dog and makes note of his preference. A possible outcome is [18, example 11.1; 28]:



Here i $\longrightarrow$ j means that the dog prefers food i to food j. The question is to indicate a ranking of the foods "which fits best the outcome of the experiment". There are various ways to define exactly what it meant by the clause between quotes [23,§15]. P. Strater [31] suggests that the foods be ranked so as to minimize the number of "errors" of the dog. Precisely formulated: (2.13).
The same experiment can be done with different dogs. Putting a(i,j) = the number of dogs preferring i to j, we get problem (2.2).

(3.2) *Weighing of priorities*. A certain number of persons has to rank n alternatives according to desirability. To this end each of the persons determines the ranking he prefers. Let a(i,j) = the number of persons putting alternative i before alternative j, for $1 \leq i, j \leq n$. Then problem (2.2) asks for a ranking which minimizes the number of neglected preferences. Also in this case there are other ways to define what is meant by a "best ranking".

(3.3) *Triangulation of input-output-matrices*. Let the economy of a country be divided in n industry sectors, and let a matrix $(a_{ij})_{1\leq i,j\leq n}$ of non-negative real numbers indicate the mutual supplies between these sectors. (2.4) asks for a ranking of the sectors which maximizes the total supply from sectors to sectors which are placed lower: a ranking "from raw material to consumer". For an extensive discussion of this problem and its economic aspects, see [1;2;4;8;9;10;11;12;13;16;17;19;20;22;26;27;30].

(3.4) *A sportman's problem*. In a certain football competition each two clubs meet exactly once. Each match is decided, if necessary by lot. Further the rules require that in the final ranking at the end of the competition any club should have beaten each club which is placed lower. A league official is responsible for the observance of this rule. When treating the protests at the end of the season he is able to "rectify" the results of a number of matches. If club i has beaten club j, the amount of money required to change the result of this match equals b(i,j). The competition manager is interested in the cheapest way to meet the rules: problem (2.12).

II. TOURNAMENTS

§4. Scores and decomposition of tournaments

In this section we recall some facts about tournaments. Reference: J.W. Moon [23]. An alternative terminology has been designed by G. Jaeschke [14,15], who apparently is unaware of the known theory.

(4.1) Scores. Let (I,K) be a tournament. For $i \in I$ let $s_i = |K[i]|$ be the *score* of i. Suitably numbering I we may assume $I = \{1,2,\dots,n\}$ and

(4.2) $$s_i \leq s_{i+1} \qquad \text{for } 1 \leq i < n.$$

The sequence $(s_i)_{i=1}^n$ is called the *score vector* of (I,K).
Fix $1 \leq k \leq n$. Then there are $\sum_{i=1}^k s_i$ edges $(i,j) \in K$ for which $i \leq k$, and $\binom{k}{2}$ edges $(i,j) \in K$ for which $i,j \leq k$. Therefore:

(4.3) $$\sum_{i=1}^k s_i \geq \binom{k}{2} \qquad \text{for } 1 \leq k < n,$$

(4.4) $$\sum_{i=1}^n s_i = \binom{n}{2}.$$

Of course we have also

(4.5) $$0 \leq s_i \leq n-1, \quad s_i \in \mathbb{Z} \qquad \text{for } 1 \leq i \leq n.$$

Conversely, for every sequence $(s_i)_{i=1}^n$ satisfying (4.2), (4.3), (4.4) and (4.5) there is a tournament which has $(s_i)_{i=1}^n$ as its score vector (theorem of H.G. Landau, [23,§21;5,Ch.II,§1;14]). For $n \geq 5$ this tournament is not necessarily uniquely determined (cf. (6.5)).

(4.6) Decomposition. The tournament (I,K) is called *reducible* if there is a subset $J \subset I$ such that

$$\emptyset \neq J \times (I \setminus J) \subset K.$$

One easily sees that (I,K) is reducible if and only if

$$\exists k : 1 \leq k < n, \ \textstyle\sum_{i=1}^{k} s_i = \binom{k}{2}.$$

A tournament which is not reducible is *irreducible* or *strong*. For every tournament (I,K) there is a unique integer $\ell > 0$ and a unique decomposition

(4.7) $$I = \bigcup_{j=1}^{\ell} I_j, \quad I_j = \emptyset \ (1 \leq j \leq \ell), \quad I_j \cap I_{j'} = \emptyset \ (1 \leq j < j' \leq \ell),$$

such that

(4.8) $$I_j \times I_{j'} \subset K \qquad \text{for } 1 \leq j' < j \leq \ell,$$

(4.9) $$(I_j, K \cap (I_j \times I_j)) \quad \text{is an irreducible tournament for } 1 \leq j \leq \ell.$$

This "decomposition in irreducible subtournaments" can be read from the scorevector $(s_i)_{i=1}^{n}$ in the following manner:
let $\{k_j \mid 0 \leq j \leq \ell\}$ be the set of k's for which

$$\textstyle\sum_{i=1}^{k} s_i = \binom{k}{2}, \qquad 0 \leq k \leq n,$$

indexed in such a way that $k_{j-1} < k_j$ $(0<j\leq\ell)$; in particular

$$k_0 = 0, \quad k_\ell = n;$$

then

$$I_j = \{i \mid k_{j-1} < i \leq k_j\}$$

for $1 \leq j \leq \ell$.

§5. Reduction to strong tournaments

Let (I,K) be a tournament. Suppose $\# \in V_I$ is such that $|K\backslash\#|$ is minimum (2.13). Let σ be the unique bijection $\sigma : \{1,2,\ldots,n\} \to I$ for which

$$i < j \Longleftrightarrow \sigma(i) \# \sigma(j)$$

for all $i,j \in \{1,2,\ldots,n\}$. Then

(5.1) $(\sigma(1),\sigma(2),\ldots\sigma(n))$ is a hamiltonian path in (I,K).

Proof of (5.1): choose i, $1 \leq i < n$, fixed; we have to prove $(\sigma(i),\sigma(i+1)) \in K$; if $(\sigma(i),\sigma(i+1)) \notin K$, then for

$$\#' = (\#\backslash\{(\sigma(i),\sigma(i+1))\}) \cup \{(\sigma(i+1),\sigma(i))\}$$

we would have

$$\#' \in V_I$$

$$K \backslash \# = (K\backslash\#') \cup \{(\sigma(i+1),\sigma(i))\} \qquad \text{(disjoint union)},$$

contradicting the minimality of $|K\backslash\#|$; this proves (5.1).

Note that (5.1) yields an easy proof of the existence of a hamiltonian path in an arbitrary tournament.

Now let $(I_j)_{j=1}^{\ell}$ be as in (4.7), (4.8) and (4.9). If $(x_{ji})_{i=0}^{n_j}$ is a hamiltonian path in $(I_j,K\cap(I_j\times I_j))$, for $1 \leq j \leq \ell$, then (4.8) implies that combination of these paths

$$((x_{\ell i})_{i=0}^{n_\ell},(x_{\ell-1\ i})_{i=0}^{n_{\ell-1}},\ldots,(x_{1i})_{i=0}^{n_1})$$

yields a hamiltonian path in (I,K). Conversely, every hamiltonian path in

(I,K) has this form, as is easily deduced from (4.8). Applied to the hamiltonian path (5.1) this yields

$$(5.2) \qquad i \in I_j,\ i' \in I_{j'},\ j > j' \Longrightarrow i \# i'.$$

Putting $\#_j = \# \cap (I_j \times I_j)$ and $K_j = K \cap (I_j \times I_j)$ we get from (5.2):

$$(5.3) \qquad |K \backslash \#| = \textstyle\sum_{j=1}^{\ell} |K_j \backslash \#_j|$$

$$\#_j \in V_{I_j} \qquad\qquad (1 \leq j \leq \ell).$$

Conversely, if $\#_j \in V_{I_j}$ are given $(1 \leq j \leq \ell)$, then one can find $\# \in V_I$ for which (5.3) holds. We conclude that when solving (2.13) one can restrict to the case of an *irreducible* tournament. This is a substantial reduction if $\ell > 1$, but unfortunately most tournaments have $\ell = 1$: putting

$$Q(n) = \frac{\text{number of reducible tournaments } (I,K) \text{ with } I = \{1,2,\ldots,n\}}{\text{number of tournaments } (I,K) \text{ with } I = \{1,2,\ldots,n\}}$$

(the donominator equals $2^{\binom{n}{2}}$), we have

$$\left| Q(n) - \frac{n}{2^{n-2}} \right| < \frac{1}{2} \cdot \left(\frac{n}{2^{n-2}} \right)^2 \qquad \text{for } n \geq 2,$$

cf. Moon [23,§2].

The following result generalizes (5.2) but seems less useful:

(5.4) Define $* \subset I \times I$ by

$$(i,j) \in * \Longleftrightarrow (i,j) \in K \wedge \nexists k \in I : [(j,k) \in I \wedge (k,i) \in K].$$

Then $*$ is a partial ordering on I, and for every $\# \in V_I$ with minimum $|K \backslash \#|$ we have $* \subset \#$.

The proof may be left to the reader.

Next we consider problem (2.12). Let $b : K \to R_{\geq 0}$ be a map, and

$$g(*) = \sum_{(i,j)\in K\setminus *} b(i,j)$$

for $* \in V_I$.

(5.5) Among the orderings $\# \in V_I$ which satisfy

$$g(\#) = \min\{g(*) \mid * \in V_I\}$$

there is at least one with the following property: if $\sigma \in S_{n,I}$ is the unique bijection for which

$$i < j \iff \sigma(i) \# \sigma(j) \qquad (i,j\in\{1,2,\ldots,n\}),$$

then $(\sigma(1),\sigma(2),\ldots,\sigma(n))$ is a hamiltonian path in (I,K).

Proof of (5.5). Choose

$$(5.6) \qquad \# \in \{\#' \in V_I \mid g(\#') = \min\{g(*) \mid * \in V_I\}\}$$

such that $|K\setminus\#|$ is minimum subject to condition (5.6). It is easily checked that $\#$ has the indicated property.

Remark. Because of the possibility that an edge has zero weight:

$$\exists (i,j) \in K : b(i,j) = 0,$$

we cannot assert that *every* $\#$ satisfying (5.6) has the property indicated in (5.5). For example, if all weights $b(i,j)$ equal zero, then *all* $\# \in V_I$ satisfy (5.6), but not all $\# \in V_I$ induce a hamiltonian path, if $n \geq 2$.

From (5.5) we conclude in the same manner as above, that when solving (2.12) we may restrict to the case of an irreducible tournament. This reduction has been recommended by G. Jaeschke [15].

If $i,j \in I$ satisfy

$$(i,j) \in K, \quad b(i,j) = 0, \quad s_i \leq s_j,$$

then it may be advantageous to replace K by

$$(K \setminus \{(i,j)\}) \cup \{(j,i)\},$$

putting $b(j,i) = 0$. This does not change the problem, but the probability of decomposition increases.

§6. Two lower bounds

Let (I,K) be a tournament, where $I = \{1,2,\dots,n\}$ is numbered in such a way that the score vector $(s_i)_{i=1}^n$ is non-decreasing (4.2). Let $(x,y) \in K$ be an edge, and let the tournament (I,K') arise from (I,K) by

first replacing (x,y) by (y,x) in K, and

secondly reindexing I in such a way that the new score vector $(s_i')_{i=1}^n$ is again non-decreasing.

Then it is easily checked that

$$\sum_{i=1}^n |s_i - s_i'| \leq 2.$$

By induction on k it follows that:

(6.1) let the tournament $(I,K^{(k)})$ arise from (I,K) by

first reversing the orientation of k edges from K, and

secondly reindexing I in such a way that the new score vector $(s_i^{(k)})_{i=1}^n$ is again non-decreasing,

then

$$\sum_{i=1}^n |s_i - s_i^{(k)}| \leq 2k.$$

If we take $k = |K\backslash *|$ for some $* \in V_I$, we can choose $K^{(k)}$ to be a total ordering. Then $s_i^{(k)} = i-1$, so (6.1) implies:

$$|K\backslash *| \geq \frac{1}{2} \sum_{i=1}^n |s_i - (i-1)| \qquad \text{for all } * \in V_I,$$

that is:

$$\text{(6.2)} \qquad \min\{|K\backslash *| \mid * \in V_I\} \geq \frac{1}{2} \sum_{i=1}^n |s_i - (i-1)|.$$

For every score vector $(s_i)_{i=1}^n$ one can construct a tournament for which the equality sign holds in (6.2) (H.J. Ryser [29], D.R. Fulkerson [6], Moon [23,§21,ex.7]). So $\frac{1}{2}\sum_{i=1}^n |s_i-(i-1)|$ is the best lower bound for $|K\backslash *|$ which depends only on the score vector. For most tournaments, however, (6.2) is a bad estimate, since on the one hand we have

$$(6.3) \qquad \frac{1}{2}\sum_{i=1}^n |s_i-(i-1)| \le \frac{1}{2} \cdot [\frac{n+1}{2}].[\frac{n-1}{2}] \sim \frac{1}{4}\binom{n}{2}$$

for every score vector $(s_i)_{i=1}^n$, and on the other hand

$$(6.4) \qquad \lim_{n\to\infty} \frac{\text{number of } K \text{ for which } \min\{|K\backslash *| \mid * \in V_I\} > (\frac{1}{2}-\varepsilon).\binom{n}{2}}{2^{\binom{n}{2}}} = 1$$

for every $\varepsilon > 0$ ("for almost all K one has to reverse nearly half of the edges to get a total ordering"). A proof of (6.4) can be found in Moon [23,§8], more precise results are given by J. Spencer [32] and Moon [24]. We do not give the proof of (6.3) here. That the equality sign can hold in (6.3) may be seen by taking

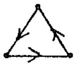
$$s_i = \frac{n-1}{2} \qquad \text{(n odd)},$$

$$s_i = \frac{n-2}{2} + [\frac{2i-1}{n}] \qquad \text{(n even)}.$$

One may hope that the tournaments encountered in the applications bear already a close resemblance to a total ordering. This resemblance may add to the probability of decomposition (§5) and the sharpness of (6.2).

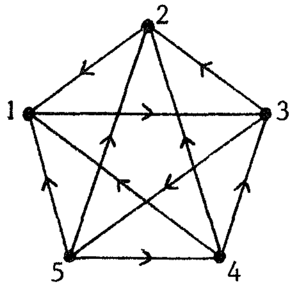
In connection with (6.2) we remark that the number of 3-cycles in (I,K) equals

$$\frac{1}{2}\sum_{i=1}^n \{(i-1)^2 - s_i^2\},$$

cf. Moon [23,§5]. (A 3-cycle is a subtournament of the form  .)

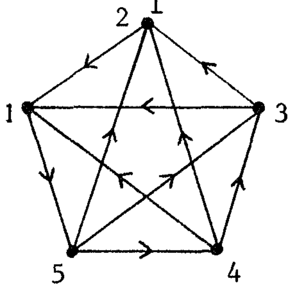
Examples

(6.5) The tournament

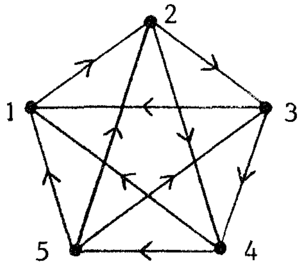


is the smallest example for which the equality sign in (6.2) does not hold. The score vector is (1,1,2,3,3), so $\frac{1}{2}\sum_{i=1}^{5} |s_i-(i-1)| = 1$. But the presence of the 3-cycles (3,5,4,3) and (3,2,1,3) implies $|K\backslash *| \geq 2$ for all $* \in V_I$. Reversing (3,5) and (1,3) one sees that in fact $\min\{|K\backslash *| \mid * \in V_I\} = 2$ holds.

A tournament with the same score vector for which equality holds in (6.2) is drawn at the right.



(6.6) The tournament



has score vector (1,2,2,2,3), so $\frac{1}{2}\sum_{i=1}^{5} |s_i-(i-1)| = 2$. There is exactly one $* \in V_I$ for which $|K\backslash *| = 2$. Also for example (3.1) the equality sign holds in (6.2).

Next we deduce two lower bounds for (2.14) which generalize (6.2).

(6.7) Notation. Let V be a finite set, and $r_v \in \mathbb{R}$ for $v \in V$. If k is an integer, $0 \leq k \leq |V|$, then we put

$$\sum_{v\in V}^{[k]} r_v = \min\{\sum_{v\in A} r_v \mid A \subset V, |A| = k\}$$

(the sum of the k smallest elements of V).

Let $b : K \to \mathbb{R}_{\geq 0}$ be a map, and

$$g(\sigma) = \sum_{1 \leq i < j \leq n, (\sigma(i),\sigma(j)) \in K} b(\sigma(i),\sigma(j))$$

for $\sigma \in S_{n,I}$, as in (2.14). Then (6.2) immediately implies our first lower bound for $g(\sigma)$:

(6.8) Let $(s_i)_{i=1}^n$ be the non-decreasing score vector of (I,K), and

$$t = \frac{1}{2} \sum_{i=1}^n |s_i - (i-1)|$$

(notice that t is an integer, cf. (4.4)). Then for all $\sigma \in S_{n,I}$ we have

$$g(\sigma) \geq \sum_{(i,j) \in K}^{[t]} b(i,j)$$

(using notation (6.7)).

We improve upon this lower bound. Write

(6.9) $$g(\sigma) = \sum_{i=1}^n \sum_{j, i<j \leq n, \sigma(j) \in K[\sigma(i)]} b(\sigma(i),\sigma(j)).$$

For fixed i the number of terms in the sum

(6.10) $$\sum_{j, i<j \leq n, \sigma(j) \in K[\sigma(i)]} b(\sigma(i),\sigma(j))$$

is at least $\max(0, |K[\sigma(i)]| - (i-1))$. If we put

(6.11) $$t_0(i,\ell) = \max(0, |K[\ell]| - (i-1)) \qquad (1 \leq i \leq n,\ \ell \in I),$$

then the sum in (6.10) is at least

$$\sum_{k \in K[\sigma(i)]}^{[t_0(i,\sigma(i))]} b(\sigma(i),k)$$

(using notation (6.7)). Therefore (6.9) yields

(6.12) $$g(\sigma) \geq \sum_{i=1}^{n} \sum_{k\in K[\sigma(i)]}^{[t_0(i,\sigma(i))]} b(\sigma(i),k).$$

Starting from

$$g(\sigma) = \sum_{j=1}^{n} \sum_{i,1\leq i<j,\sigma(i)\in K^{-1}[\sigma(j)]} b(\sigma(i),\sigma(j))$$

we find in the same way

(6.13) $$g(\sigma) \geq \sum_{j=1}^{n} \sum_{k\in K^{-1}[\sigma(j)]}^{[t_1(j,\sigma(j))]} b(k,\sigma(j))$$

where

(6.14) $$t_1(j,\ell) = \max(0,j-1-|K[\ell]|) \qquad (1\leq j\leq n,\ \ell\in I).$$

Adding (1-λ) × (6.12) and λ × (6.13), for $\lambda \in [0,1]$, we find our second lower bound:

(6.15) For $i \in \{1,2,\ldots,n\}$, $\ell \in I$ and $\lambda \in \mathbb{R}$, $0 < \lambda < 1$, we define

(6.11) $$t_0(i,\ell) = \max(0,|K[\ell]|-(i-1))$$

(6.14) $$t_1(i,\ell) = \max(0,(i-1)-|K[\ell]|)$$

$$\left.\begin{aligned} e_0(i,\ell) &= \sum_{k\in K[\ell]}^{[t_0(i,\ell)]} b(\ell,k) \\ e_1(i,\ell) &= \sum_{k\in K^{-1}[\ell]}^{[t_1(i,\ell)]} b(k,\ell) \end{aligned}\right\} \qquad \text{cf. (6.7)}$$

$$e_\lambda(i,\ell) = (1-\lambda).e_0(i,\ell) + \lambda.e_1(i,\ell).$$

Then

$$g(\sigma) \geq \sum_{i=1}^{n} e_\lambda(i,\sigma(i))$$

for every $\sigma \in S_{n,I}$ and $\lambda \in [0,1]$, and therefore

(6.16) $$\min_{\sigma\in S_{n,I}} g(\sigma) \geq \min_{\sigma\in S_{n,I}} \sum_{i=1}^{n} e_\lambda(i,\sigma(i))$$

for every $\lambda \in [0,1]$.

The lower bound (6.16) may be considered as a special case of the lower bound given by P.C. Gilmore for the general quadratic assignment problem [7], cf. (2.5).

We add some remarks.

(6.17) For fixed λ, the determination of the right hand side of (6.16):

(6.18) $$\min_{\sigma\in S_{n,I}} \sum_{i=1}^{n} e_\lambda(i,\sigma(i))$$

is a linear assignment problem. The dual of this problem is (cf. [5]):

$$\text{maximize } \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

subject to

$$u_i+v_j \leq e_\lambda(i,j) \qquad (1\leq i,j\leq n).$$

Here we have taken $I = \{1,2,\ldots n\}$.

Therefore, choosing λ such that (6.18) is maximum comes down to solving the linear programming problem:

$$\text{maximize } \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

subject to

$$u_i + v_j + \lambda.(e_0(i,j)-e_1(i,j)) \leq e_0(i,j) \qquad (1\leq i,j\leq n)$$

$$0 \leq \lambda \leq 1, \quad -\infty < u_i < \infty, \quad -\infty < v_j < \infty \qquad (1\leq i,j\leq n).$$

It seems possible to find an efficient algorithm for this problem.

(6.19) If the tournament (I,K) is reducible, then the lower bound (6.16) can be improved by decomposing (I,K) in subtournaments $(I_j,K\cap(I_j\times I_j))$ (cf. (4.6)) and adding the lower bounds obtained for these subtournaments by (6.16). An analogous statement is true for (6.8).

(6.20) From (6.11) and (6.14) it follows that $t_0(i,\ell)$ and $t_1(i,\ell)$ are determined by

$$\left.\begin{aligned} t_0(i,\ell) - t_1(i,\ell) &= |K[\ell]| - (i-1) \\ t_0(i,\ell) + t_1(i,\ell) &= \big||K[\ell]| - (i-1)\big| \end{aligned}\right. \qquad (1\leq i\leq n,\ \ell\in I).$$

(6.21) Finally we show that for problem (2.13) the bound (6.2) is implied by (6.16). In the case of (2.13) we have $b(k,\ell) = 1$ for all $(k,\ell) \in K$, so

$$e_0(i,\ell) = t_0(i,\ell),\quad e_1(i,\ell) = t_1(i,\ell) \qquad (1\leq i\leq n,\ \ell\in I).$$

Hence (6.20) implies

$$e_\lambda(i,\ell) = \frac{1}{2}\,\big||K[\ell]| - (i-1)\big| + (\tfrac{1}{2}-\lambda).(|K[\ell]|-(i-1))$$

for $1 \leq i \leq n$, $\ell \in I$, $\lambda \in [0,1]$. Since

$$\textstyle\sum_{\ell\in I} |K[\ell]| = \binom{n}{2} = \sum_{i=1}^{n} (i-1)$$

it follows that for each bijection $\sigma : \{1,2,\ldots,n\} \to I$ and each $\lambda \in [0,1]$ we have

$$\textstyle\sum_{i=1}^{n} e_\lambda(i,\sigma(i)) = \frac{1}{2}\sum_{i=1}^{n} \big||K[\sigma(i)]| - (i-1)\big|.$$

It is easily seen that the right hand side is minimal if σ is chosen such that the sequence $(K[\sigma(i)])_{i=1}^{n}$ is non-decreasing. Therefore in this case (6.16) yields exactly the bound (6.2), for which we thus found a new proof.

III. SUBOPTIMAL ALGORITHMS

§7. Ranking according to scores

Let $(a_{ij})_{1\leq i,j\leq n}$ be an n×n-matrix with real non-negative entries. We want to maximize

$$g(\sigma) = \sum_{1\leq i<j\leq n} a_{\sigma(i)\sigma(j)}$$

subject to $\sigma \in S_n$ (problem (2.4)).

Some methods to determine a crude ranking are based on the scores

$$(7.1) \qquad t_i = \sum_{j=1}^{n} a_{ij}$$

$$u_i = \sum_{j=1}^{n} a_{ji}.$$

For example, one can choose σ such that $(t_{\sigma(i)})_{i=1}^{n}$ is non-increasing, or such that $(u_{\sigma(i)})_{i=1}^{n}$ is non-decreasing, or such that $(t_{\sigma(i)}-u_{\sigma(i)})_{i=1}^{n}$ is non-increasing: ranking of football clubs according to the number of goals scored by themselves, the number of goals scored by their opponents, and the difference between these numbers, respectively.

O. Becker proposes to define $\sigma(1)$ by

$$\frac{t_{\sigma(1)}}{u_{\sigma(1)}} = \max\{\frac{t_i}{u_i} \mid 1 \leq i \leq n\}$$

and to apply induction on n to the (n-1)×(n-1)-matrix $(a_{ij})_{i,j\neq\sigma(1)}$ to determine the remainder of the sequence [2]. However, there seems to be no reason to share his optimism about the optimality of this priority rule [20]. Analogous methods have been suggested by G. Chaty [3] for the case of problem (2.13).

H. Aujac [1] and D. Masson [22] say that j is *dominated* by i if

$$\frac{a_{ij}}{t_i} < \frac{a_{ji}}{t_j} ;$$

this relation needs not be transitive, and it remains unclear in which way this "principle of dominance" leads to a solution.

The algorithm of H.B. Chenery and T. Watanabe [4,p.496], referred to in [20], does not apply to the problem under consideration.

(7.2) Using formulation (2.14), a σ for which the right hand side of (6.16) is minimal can be used as a first solution. For problem (2.13) this comes down to a ranking according to scores, cf. (6.21).

§8. Two travelling-salesman methods

In [20] B. Korte and W. Oberhofer describe two suboptimal algorithms for problem (2.4), which are modifications of travelling-salesman methods given by H. Müller-Merbach [25;26].

(8.1) *Successive insertion of points.* This method can be formulated with induction on n as follows:

for $n = 1$, the problem is trivial;

for $n > 1$, one starts by solving the problem for the $(n-1)\times(n-1)$-matrix $(a_{ij})_{1\leq i,j\leq n-1}$; this yields a $\sigma^* \in S_{n-1}$; from the n elements $\sigma \in S_n$ with the property

$$\forall 1 \leq i,j \leq n-1 : \sigma^{-1}(i) < \sigma^{-1}(j) \Longleftrightarrow \sigma^{*-1}(i) < \sigma^{*-1}(j)$$

the one which maximizes $g(\sigma)$ is chosen. This solution is expected to be a good one.

Of course, instead of $(a_{ij})_{1\leq i,j\leq n-1}$ one can use the $(n-1)\times(n-1)$-matrix $(a_{ij})_{i,j\neq k}$, where $k \in \{1,2,\dots,n\}$ is arbitrary. This makes it possible to construct several "good" solutions from which the best one can be chosen.

(8.2) *Choosing the nearest neighbour.* This algorithm determines $\sigma(1),\dots,\sigma(n)$ in the following way: let, for an m with $1 \leq m \leq n$, all $\sigma(i)$ with $i < m$ be determined, then $\sigma(m)$ is chosen from $\{1,2,\dots,n\}\setminus\{\sigma(i)\,|\,i<m\}$ in such a way that

$$\text{(8.3)} \qquad \sum\nolimits_{1\leq i<j\leq m} a_{\sigma(i)\sigma(j)}$$

is maximized. After n steps σ is found.

This rule does not prescribe how to choose $\sigma(1)$. Also it is not clear why the larger sum

$$\sum\nolimits_{1\leq i<j\leq m} a_{\sigma(i)\sigma(j)} + \sum\nolimits_{i=1}^{m} \sum\nolimits_{1\leq j\leq n, j\notin\{\sigma(1),\dots,\sigma(m)\}} a_{\sigma(i)j}$$

is not used instead of (8.3) (cf. §14). In that case, the method would be equivalent to the following rule: select $\sigma(1)$ by $t_{\sigma(1)} = \max\{t_i \mid 1 \le i \le n\}$ with t_i as in (7.1), and apply induction on $(a_{ij})_{i,j\neq\sigma(1)}$ to determine the remainder of the sequence. (Here we assume $a_{ii} = 0$ for all i, cf. (2.17)).

§9. Relative optimality

We consider again formulation (2.4). For $\tau \in S_n$ the requirement

$$\forall \sigma \in S_n : g(\tau) \geq g(\sigma)$$

is equivalent to

$$(9.1) \qquad g(\tau) \geq g(\tau\rho)$$

for all $\rho \in S_n$. Some heuristic methods construct a $\tau \in S_n$ satisfying (9.1) for all $\rho \in R$, where $R \subset S_n$ is a certain subset.

O. Becker [2] takes R to be the subgroup of S_n generated by the cyclic permutation (n 1 2 ... n-1); so we have $|R| = n$. In this case

$$(9.2) \qquad \forall \rho \in R : g(\tau) \geq g(\tau\rho)$$

means, that for no $m \in \{1,2,\dots,n-1\}$ one can improve the ranking $(\tau(1),\tau(2),\dots,\tau(n))$ by replacing it by the ranking $(\tau(m+1),\tau(m+2),\dots,\tau(n), \tau(1),\tau(2),\dots,\tau(m))$. One easily checks:

(9.3) Let

$$z_i = t_i - u_i = \sum_{j=1}^{n} (a_{ij} - a_{ji})$$

for $1 \leq i \leq n$, and let $R \subset S_n$ be the subgroup generated by (n 1 2 ... n-1). Then (9.2) is equivalent to

$$\forall k \in \{1,2,\dots,n-1\} : \sum_{i=1}^{k} z_{\tau(i)} \geq 0.$$

According to [20], Becker recommends the following iteration:

$$\tau_0 = \text{arbitrary element of } S_n;$$

τ_k for $k \geq 1$ is defined by

$$\tau_k \in \tau_{k-1}.R, \qquad g(\tau_k) = \max\{g(\tau_{k-1},\rho) \mid \rho \in R\}.$$

However, after the first step of this iteration no further improvement is found: since R is a subgroup, $\tau_1 \in \tau_0 R$ implies $\tau_1 R = \tau_0 R$. So, if Korte and Oberhofer write [20, p.405]:

"Nach k Iterationsschritten sind erst n.k Permutationen untersucht"

one should read: "After k iteration steps each one of n permutations has been examined k times". No value can be attached to this method.

The following choice for R is more promising. Define $\rho_{k\ell} \in S_n$ for $1 \leq k < \ell \leq n$ by

$$\begin{aligned} \rho_{k\ell}(i) &= i && \text{if } 1 \leq i < k \text{ or } \ell < i \leq n, \\ \rho_{k\ell}(i) &= i-1 && \text{if } k < i \leq \ell, \\ \rho_{k\ell}(k) &= \ell, \end{aligned}$$

and put

$$(9.4) \qquad R = \{\rho_{k\ell}, \rho_{k\ell}^{-1} \mid 1 \leq k < \ell \leq n\},$$

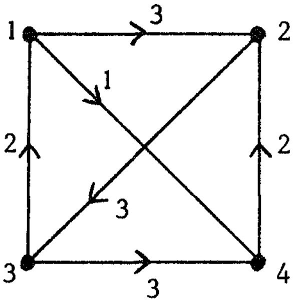
so $|R| = (n-1)^2$ (not $n(n-1)$, cf. [20,p.417]). For this R (9.2) means that the ranking $(\tau(1),\tau(2),\ldots,\tau(n))$ cannot be improved by moving one $\tau(m)$ to a different place. One easily checks:

(9.5) Let R be defined as in (9.4), and $\tau \in S_n$. Then (9.2) holds if and only if for all k,ℓ with $1 \leq k < \ell \leq n$:

$$(9.6)\begin{cases}\sum_{j=k+1}^{\ell} (a_{\tau(k)\tau(j)}-a_{\tau(j)\tau(k)}) \geq 0\\ \\ \sum_{j=k}^{\ell-1} (a_{\tau(j)\tau(\ell)}-a_{\tau(\ell)\tau(j)}) \geq 0.\end{cases}$$

A $\tau \in S_n$ having the property (9.6) for all k,ℓ with $1 \leq k < \ell \leq n$ is called *relatively optimal*. It follows from the above that each *optimal* τ is relatively optimal. The converse does not hold [31;17]: let $n = 4$, and define

(9.7)



$(a_{ij})_{1\leq i,j\leq 4}$ by $a_{12} = a_{23} = a_{34} = 3$, $a_{31} = a_{42} = 2$, $a_{14} = 1$, all other $a_{ij} = 0$, cf. figure (9.7); for τ = identity we have $g(\tau) = 10$, and τ is relatively optimal (in fact, strict inequalities hold in (9.6)); however, τ is not optimal, for $g(\sigma) = 11$ with $(\sigma(1),\sigma(2),\sigma(3),\sigma(4)) = (3,1,4,2)$. If in this example all $a_{ij} \neq 0$ are replaced by 1, cf. (2.13), then τ is relatively optimal again, while $g(\tau) = 4$ and $g(\sigma) = 5$.

The introduction of the notion "relative optimality" involved some difficulties, for which we refer to the discussion between E. Helmstädter on the one side and H.J. Jaksch and H. König on the other side [8;16;9;17;10].

A relatively optimal τ is constructed by the following iteration:

A. choose $\tau_0 \in S_n$ arbitrarily, $k := 0$, go to B.

B. if $\forall\rho \in R : g(\tau_k\rho) \leq g(\tau_k)$, then τ_k is relatively optimal and the iteration comes to an end; else: go to C.

C. choose $\rho \in R$ such that $g(\tau_k\rho) > g(\tau_k)$, put $\tau_{k+1} := \tau_k\rho$, $k := k+1$, go to B.

The first algorithm of this type has been given by G.G. Alway for problem (2.13) [31], i.e. for the case

$$a_{ij} + a_{ji} = 1$$

$$a_{ij} \cdot a_{ji} = 0$$

for $1 \leq i < j \leq n$. By putting $\tau_{k+1} := \tau_k\rho$ also in some cases where $g(\tau_k\rho)$ is *equal* to $g(\tau_k)$ his algorithm computes an *optimal* solution if $n < 8$ [31;28].

The second method using this iteration scheme is described by Korte and Oberhofer [19]. In [20] they classify it wrongly as an optimal algorithm. The fastness of their procedure allows us to use several τ's as starting points and to choose the best one of the constructed relatively optimal solutions.

Finally, we make some remarks about the set of those $\rho \in S_n$ for which (9.1) holds if τ is relatively optimal.

If $a_0, a_1, b_0, b_1 \in \mathbb{R}$ we say that $\{a_0, a_1\}$ and $\{b_0, b_1\}$ are *separating pairs* if

(9.8) there are $i,j \in \{0,1\}$ such that

$$a_i < b_j < a_{1-i} < b_{1-j}$$

or

$$b_j < a_i < b_{1-j} < a_{1-i}.$$

One can prove:

(9.9) If $\rho \in S_n$ satisfies

(9.10) $\forall 1 \leq i,j \leq n$: $\{i,\rho(i)\}$ and $\{j,\rho(j)\}$ are *no* separating pairs,

then $g(\tau\rho) \leq g(\tau)$ for every relatively optimal τ.

Notice that ρ has property (9.10) if

$$|\{i \mid \rho(i) \neq i\}| \leq 3.$$

Put $y_n = |\{\rho \in S_n \mid \rho \text{ satisfies } (9.10)\}|$, and $y_0 = 1$. Let Y be the formal power series

$$Y = \sum_{n=0}^{\infty} y_n x^n.$$

One can prove

$$x^3Y^3 + xY^2 - Y + 1 = 0.$$

This formula can be used to determine the y_n successively; one finds:

$$y_0 = 1,\ y_1 = 1,\ y_2 = 2,\ y_3 = 6,\ y_4 = 19,\ y_5 = 63,\ y_6 = 219,\ y_7 = 787.$$

Using the theory of algebraic functions one can show

$$\lim_{n\to\infty} \sqrt[n]{y_n} = \sqrt{9 + 6\sqrt{3}}.$$

IV. OPTIMAL ALGORITHMS

§10. Explicit enumeration

One way to solve (2.4) is checking all n! permutations. Even though one can stop half way be using (2.15), this method seems too laborious.

Because of (5.5) it suffices to check all hamiltonian paths in the corresponding tournament. For *irreducible tournaments*, G. Jaeschke [15] proposes to enumerate the permutations τ for which

(10.1) $(\tau(1),\dots,\tau(n))$ is a hamiltonian path, and

(10.2) $\forall k \in \{1,2,\dots,n-1\} : \sum_{i=1}^{k} \sum_{j=1}^{n} (a_{\tau(i)j} - a_{j\tau(i)}) \geq 0$ (cf. (9.3)).

He reduces *reducible* tournaments to irreducible ones (§5).
Korte and Oberhofer [19] enumerate all *relatively optimal* solutions. This is done lexicographically:

```
$\tau(1)$ takes the values $1,2,\dots,n$;
if $\tau(1)$ is fixed, $\tau(2)$ takes those values from $\{1,2,\dots,n\}\setminus\{\tau(1)\}$
   for which (9.6) holds for $k = 1$ and $\ell = 2$;
:
if $\tau(1),\dots,\tau(\ell-1)$ are fixed, $\tau(\ell)$ takes those values from
   $\{1,2,\dots,n\}\setminus\{\tau(1),\dots,\tau(\ell-1)\}$ for which (9.6) holds for all $k < \ell$
:
```

It is clear that in this way one obtains all relatively optimal solutions τ. From

$$\sum_{j=\ell+1}^{n} (a_{\tau(\ell),\tau(j)} - a_{\tau(j),\tau(\ell)}) =$$

$$= \sum_{i=1}^{n} (a_{\tau(\ell),i} - a_{i,\tau(\ell)}) - \sum_{j=1}^{\ell-1} (a_{\tau(\ell),\tau(j)} - a_{\tau(j),\tau(\ell)}),$$

one sees, that if $\tau(1),\ldots,\tau(\ell-1)$ are fixed, one can also impose the condition

$$(10.3) \qquad \sum_{j=1}^{\ell-1} (a_{\tau(j),\tau(\ell)}-a_{\tau(\ell),\tau(j)}) \geq \sum_{i=1}^{n} (a_{i,\tau(\ell)}-a_{\tau(\ell),i})$$

on $\tau(\ell)$ [19]. The possibility to require also

$$(10.4) \qquad \sum_{i=1}^{\ell} \sum_{j=1}^{n} (a_{\tau(i),j}-a_{j,\tau(i)}) \geq 0$$

(cf. (9.3)) is not used by Korte and Oberhofer (notice that some relatively optimal solutions which are not optimal may be eliminated by (10.4)).

Numerical results of Korte and Oberhofer suggest that the average number of relatively optimal solutions increases exponentially with n [20,pp.423-424].

A similar method to enumerate all relatively optimal solutions in the case all a_{ij} are 0 or 1 has been proposed by G. Chaty [3].

§11. Linear programming

In (2.11) we formulated the problem as a discrete linear programming problem. This, however, does not immediately imply an efficient solution method.

We remark that an optimal solution to (2.11) need not be an optimal solution to the linear programming problem which arises from (2.11) if one replaces the constraints

$$x_{ij} \in \{0,1\} \qquad \text{for } 1 \leq i,j \leq n$$

by

$$0 \leq x_{ij} \leq 1 \qquad \text{for } 1 \leq i,j \leq n.$$

To prove this remark, consider a tournament (I,K) for which

$$I = \{1,2,\ldots,n\}$$

$$\forall * \in V_I : |K\backslash *| \geq \frac{1}{3} \cdot \binom{n}{2}; \tag{11.1}$$

if n is sufficiently large, such a tournament exists by (6.4). Define a_{ij} for $i,j \in I$ by

$$a_{ij} = 1 \quad \text{if } (i,j) \in K,$$

$$a_{ij} = 0 \quad \text{else.}$$

For every feasible solution $(x_{ij})_{1\leq i,j\leq n}$ to (2.11) we have by (11.1):

$$\sum_{1\leq i,j\leq n} a_{ij}x_{ij} < \frac{2}{3} \binom{n}{2}.$$

But if we define

$$x_{ij} = \frac{2a_{ij} + a_{ji}}{3} \qquad \text{for } i,j \in I,$$

then $(x_{ij})_{1\leq i,j\leq n}$ is a feasible solution of the non-discrete version of (2.11) described above, and

$$\sum\nolimits_{1\leq i,j\leq n} a_{ij}x_{ij} = \frac{2}{3}\binom{n}{2}.$$

The non-discrete version of (2.11) may be used to determine a lower bound.

§12. Dynamic programming

If (J,L) is a tournament and $c : L \to \mathbb{R}_{>0}$ a function, then we define

$$g(*) = \sum\nolimits_{(i,j)\in L\setminus *} c(i,j) \qquad\qquad \text{for } * \in V_J,$$

$$m(J,L,c) = \min\{g(*) \mid * \in V_J\}.$$

Let $* \in V_J$, and choose $p \in J$ such that

$$\forall i \in J : i = p \vee i * p$$

(so p is "minimal" if $*$ = "larger than"). Then

$$(12.1) \qquad g(*) = \sum\nolimits_{j\in L[p]} c(p,j) + \sum\nolimits_{(i,j)\in L_p\setminus *_p} c_p(i,j)$$

where

$$J_p = J\setminus\{p\}, \qquad L_p = L \cap (J_p\times J_p),$$

$$*_p = * \cap (J_p\times J_p), \qquad c_p = c|L_p.$$

From (12.1) it follows that

$$(12.2) \qquad m(J,L,c) = \min_{p\in J} \left(\sum\nolimits_{j\in L[p]} c(p,j) + m(J_p,L_p,c_p)\right),$$

while

$$|J_p| = |J| - 1$$

for all $p \in J$.

Now consider problem (2.12). Applying (12.2) to the subtournaments of (I,K), we can successively determine all values

$$m(J,K\cap(J\times J)),\ b|K\cap(J\times J)) \qquad (J\subset I),$$

starting from

$$m(J,L,c) = 0 \qquad \text{for } |J| = 2.$$

Then we know m(I,K,b) at the end of the computation, and by an easy device also an optimal solution $* \in V_I$ can be found.

This algorithm is given by R. Remage and W.A. Thompson [28] for the case of problem (2.13), and by Korte and Oberhofer [20] for the general case. The memory storage required for the execution of the algorithm increases exponentially with n.

§13. The quadratic assignment problem

In (2.5) we exhibited the problem as a quadratic assignment problem. Korte and Oberhofer [20,p.417] write:

> "Es liegt daher nahe, die bekannten Algorithmen zur Lösung des quadratischen Assignment Problems von L. Steinberg [27], P.C. Gilmore [28] und E.L. Lawler [29] gegebenenfalls mit einigen Modifikationen auch auf das Triangulationsproblem anzuwenden. Da diese Algorithmen aber im allgemeinen nach dem Branch-and-Bound-Prinzip aufgebaut sind, wird es effizienter sein, Input-Output-Matrizen mittels eines speziellen Branch-and-Bound-Algorithmus, wie in Abschnitt b) dargestellt, zu triangulieren".

However, the main difference between the algorithm described in "Abschnitt b)" and the method of Gilmore [7] consists of the considerably less sharp lower bounds used by Korte and Oberhofer. Cf. §14.

[27] [33]

[28] [7]

[29] [21]

§14. Branch-and-bound methods

In this section we make use of the notations from (1.6) and (2.14).

To describe a branch-and-bound method for the solution of (2.14) we have to indicate a certain class of subsets of $S_{n,I}$, and for each of these subsets a *branching rule* and a *bounding rule*.

For $0 \leq k \leq n$ and $\alpha \in S_{k,I}$ let

$$T_\alpha = \{\sigma \in S_{n,I} \mid \sigma|\{1,2,\ldots,k\} = \alpha, \text{ i.e. } \sigma(i) = \alpha(i) \text{ for } 1 \leq i \leq k\}.$$

The class of subsets of $S_{n,I}$ to be used is

$$\{T_\alpha \mid 0 \leq k \leq n,\ \alpha \in S_{k,I}\}.$$

Taking $k = 0$, we see that $S_{n,I}$ itself is one of these subsets. For $k = n$, $\alpha \in S_{k,I}$, the set T_α is equal to $\{\alpha\}$. For $k < n$, $\alpha \in S_{k,I}$, we have a partition

$$T_\alpha = \cup_{\beta\in S_{k+1,I},\beta|\{1,2,\ldots,k\}=\alpha}\ T_\beta.$$

This is our *branching rule.*

Next we look for a *bounding rule,* i.e. we want to indicate a lower bound for

$$\min\{g(\sigma) \mid \sigma \in T_\alpha\}$$

for every $0 \leq k \leq n$, $\alpha \in S_{k,I}$.

Korte and Oberhofer [20] remark that for $\sigma \in T_\alpha$ one has

$$(14.1) \qquad g(\sigma) \geq \sum_{\substack{1\leq i<j\leq k \\ (\alpha(i),\alpha(j))\in K}} b(\alpha(i),\alpha(j)).$$

This is seen by suppressing the terms for which

$$i \leq k < j$$

or

$$k < i < j$$

in the definition of $g(\sigma)$, cf. (2.14). They use the right hand side of (14.1) as a lower bound for $\min\{g(\sigma) \mid \sigma \in T_\alpha\}$.
This lower bound can be sharpened considerably. Putting $I_\alpha = I\backslash\{\alpha(1),\ldots,\alpha(k)\}$, we have for $\sigma \in T_\alpha$:

$$(14.2) \qquad g(\sigma) = \sum\nolimits_{1\leq i<j\leq k,(\alpha(i),\alpha(j))\in K} b(\alpha(i),\alpha(j))$$
$$+ \sum\nolimits_{i=1}^{k} \sum\nolimits_{j\in K[\alpha(i)]\cap I_\alpha} b(\alpha(i),j)$$
$$+ \sum\nolimits_{k+1\leq i<j\leq n,(\sigma(i),\sigma(j))\in K} b(\sigma(i),\sigma(j)).$$

The first two terms of the right hand side depend only on α, and their sum is a lower bound for $\min\{g(\sigma) \mid \sigma \in T_\alpha\}$ which is sharper than the lower bound of Korte and Oberhofer. Minimizing the last term of (14.2) is equivalent to solving (2.14) for the subtournament $(I_\alpha, K\cap(I_\alpha \times I_\alpha))$. In this way we find the following bounding rule:

(14.3) Write

$$m(I,K,b) = \min\{g(\sigma) \mid \sigma \in S_{n,I}\}$$

and

$$m(\alpha) = \min\{g(\sigma) \mid \sigma \in T_\alpha\},$$

$$I_\alpha = I\backslash\{\alpha(1),\ldots,\alpha(k)\}$$

for $\alpha \in S_{k,I}$, $0 \leq k \leq n$.
Then

(14.4) $m(\alpha) = c(\alpha) + m(I_\alpha, K\cap(I_\alpha \times I_\alpha), b|K\cap(I_\alpha \times I_\alpha))$

where

$$c(\alpha) = \sum_{\substack{1\leq i<j\leq k \\ (\alpha(i),\alpha(j))\in K}} b(\alpha(i),\alpha(j)) + \sum_{i=1}^{k} \sum_{j\in K[\alpha(i)]\cap I_\alpha} b(\alpha(i),j).$$

The second term of the right hand side of (14.4) can be estimated from below by §6.

The lower bounds for $\min\{g(\sigma) \mid \sigma \in T_\alpha\}$ found in this way are slightly better than the lower bounds given by Gilmore [7] for the general quadratic assignment problem.

Korte and Oberhofer, who use formulation (2.4), could also have improved their lower bound (14.1) by applying (2.17). The limited success of their branch-and-bound algorithm is not surprising.

In the above we described a branch-and-bound algorithm for the solution of (2.14) by specifying a class of subsets of $S_{n,I}$, a branching rule and a bounding rule. We add some remarks about this algorithm.

(14.5) It is important to know a low upper bound for $\min\{g(\sigma) \mid \sigma \in S_{n,I}\}$ in an early stage of the computations. Such an upper bound can be found by constructing a solution by one of the methods described in §§7-9. For example, a first solution arises from the determination of the lower bound (6.16), cf. (7.2). A similar remark applies if one wants to find a "good" $\sigma \in T_\alpha$, for $\alpha \in S_{k,I}$.

(14.6) Let $S' \subset S_{n,I}$ be the set of relatively optimal solutions, i.e.

$$S' = \{\tau \in S_{n,I} \mid \forall\rho \in R : g(\tau) \leq g(\tau\rho)\}$$

with R as in (9.4). Since every optimal solution is relatively optimal, minimizing $g(\sigma)$ subject to $\sigma \in S_{n,I}$ is equivalent to minimizing $g(\sigma)$ subject to $\sigma \in S'$. The latter problem can be solved by a completely analogous branch-and-bound method, using the subsets

$$T'_\alpha = T_\alpha \cap S' \qquad (\alpha \in S_{k,I},\ 0 \le k \le n)$$

with the branching rule

$$(14.7) \qquad T'_\alpha = \cup_{\beta \in S_{\ell,I}, \beta|\{1,\dots,\ell-1\}=\alpha}\, T'_\beta \qquad (\alpha \in S_{\ell-1,I},\ 1 \le \ell \le n)$$

and for T'_α the same lower bound as for T_α. This has the advantage that some T'_β, occurring in (14.7), can be eliminated immediately since they are empty; in fact, $T'_\beta \ne \emptyset$ implies (cf. (9.6) and (10.3)):

$$(14.8) \qquad \forall k,\ 1 \le k < \ell : \begin{cases} \sum_{j=k+1}^{\ell} (b(\beta(k),\beta(j))-b(\beta(j),\beta(k))) \le 0 \\ \\ \sum_{j=k}^{\ell-1} (b(\beta(j),\beta(\ell))-b(\beta(\ell),\beta(j))) \le 0, \end{cases}$$

$$(14.9) \qquad \sum_{j=1}^{\ell-1} (b(\beta(\ell),\beta(j))-b(\beta(j),\beta(\ell))) \ge \sum_{i \in I} (b(\beta(\ell),i)-b(i,\beta(\ell))),$$

where we define $b(i,j) = 0$ for $i,j \in I$, $(i,j) \notin K$.
Narrowing slightly the notion "relatively optimal", we can also require

$$(14.10) \qquad \sum_{j=1}^{\ell} \sum_{i \in I} (b(\beta(j),i)-b(i,\beta(j))) \le 0,$$

cf. (10.4).
From $\beta|\{1,2,\dots,\ell-1\} = \alpha$ it follows that, for $\alpha \in S_{\ell-1,I}$ fixed, the inequalities (14.8), (14.9) and (14.10) are conditions which should be satisfied by $\beta(\ell)$. In [19, 3.2] one can find a simple method to check (14.8) and (14.9).
Finally we remark that if the tournament

$$(I_\alpha, K \cap (I_\alpha \times I_\alpha)) \qquad \text{where } I_\alpha = I \backslash \{\alpha(1),\dots,\alpha(\ell-1)\},$$

is reducible (4.6), say

$$I_\alpha = I_{\alpha 1} \cup I_{\alpha 2}, \qquad \emptyset \neq I_{\alpha 1} \times I_{\alpha 2} \subset K,$$

we can impose the condition

$$\beta(\ell) \in I_{\alpha 1}.$$

Of course it is possible that, if T_β' has been eliminated for one of these reasons, T_β contains a solution which is better than the best one found so far; however, that solution cannot be relatively optimal.

A branch-and-bound algorithm as described in this section has not yet been programmed for a computer.

Literature

1. H. AUJAC, La hiérarchie des industries dans un tableau des échanges interindustriels, *Revue Economique* 2 (1960) 171 sqq.
2. O. BECKER, Das Helmstädtersche Reihenfolgeproblem - die Effizienz verschiedener Näherungsverfahren, in: *Computer Uses in the Social Sciences*, Bericht einer Working Conference des Inst. f. höh. Studien u. wiss. Forsch., Vienna, January 1967.
3. G. CHATY, Cheminements remarquables dans les graphes: existence, obtention, conservation, Thèse présentée à l'université de Paris VI, 1971.
4. H.B. CHENERY, T. WATANABE, International Comparisons of the Structure of Production, *Econometrica* 26 (1958) 487-521.
5. L.R. FORD, JR., D.R. FULKERSON, *Flows in Networks*, Princeton 1962.
6. D.R. FULKERSON, Upsets in round robin tournaments, *Canad.J.Math.* 17 (1965) 957-969.
7. P.C. GILMORE, Optimal and suboptimal algorithms for the quadratic assignment problem, *J.Soc.Indust.Appl.Math.* 10 (1962) 305-313.
8. E. HELMSTÄDTER, Produktionsstruktur und Wachstum, *Jb.f.Nat.ök.u.Stat.* 169 (1957/8) 173-212, 427-449.
9. E. HELMSTÄDTER, Replik auf H.J. Jaksch und H. König: Zur Ordnung der Produktionsstruktur von Vielsektorenmodellen, *Jb.f.Nat.ök.u.Stat.* 173 (1961) 146-153.
10. E. HELMSTÄDTER, Die geordnete Input-Output Struktur, *Jb.f.Nat.ök.u. Stat.* 174 (1962) 322-361.
11. E. HELMSTÄDTER, Die Dreiecksform der Input-Output-Matrix und ihre möglichen Wandlungen im Wachstumsprozess, in: F. NEUMARK (ed.), *Strukturwandlungen einer wachsenden Wirtschaft* (Schr. d. Vereins f. Socialpolitik, N.F., 30/II), Berlin, 1964, 1005-1054.
12. E. HELMSTÄDTER, Ein Vergleich der Hierarchie der Wirtschaftsgruppen in den EWG-Ländern, *Stat.Hefte* V, 1/2 (1964) 19 sqq.
13. E. HELMSTÄDTER, Ein Mass für das Ergebnis der Triangulation von Input-Output-Matrizen, *Jb.f.Nat.ök.u.Stat.* 177 (1965) 456 sqq.
14. G. JAESCHKE, Types of Complete, Antisymmetric Graphs without Loops, Report 70.04.002, IBM Heidelberg Scientific Center, Heidelberg, 1970.

15. G. JAESCHKE, Vicinal Sequencing Problems, *Opns.Res.* 20 (1972) 984-992.

16. H.J. JAKSCH, H. KÖNIG, Zur Ordnung der Produktionsstruktur von Vielsektorenmodellen, *Jb.f.Nat.ök.u.Stat.* 172 (1960) 400-415.

17. H.J. JAKSCH, H. KÖNIG, Zur Ordnung der Produktionsstruktur von Vielsektorenmodellen: Eine Erwiderung, *Jb.f.Nat.ök.u.Stat.* 174 (1962) 56-60.

18. M.G. KENDALL, *Rank correlation methods*, London, 1948.

19. B. KORTE, W. OBERHOFER, Zwei Algorithmen zur Lösung eines komplexen Reihenfolgeproblems, *Unternehmensforschung* 12 (1968) 217-231.

20. B. KORTE, W. OBERHOFER, Zur Triangulation von Input-Output-Matrizen, *Jb.f.Nat.ök.u.Stat.* 182 (1969) 398-433.

21. E.L. LAWLER, The Quadratic Assignment Problem, *Management Sci.* 9 (1963) 586-599.

22. D. MASSON, Méthode de triangulation du tableau européen des échanges interindustriels, *Revue Economique* 2 (1960) 239 sqq.

23. J.W. MOON, *Topics on Tournaments*, New York etc., 1968.

24. J.W. MOON, Four Combinatorial Problems, in: D.A. WELSH (ed.), *Combinatorial Mathematics and its Applications*, London-New York, 1971, 185-190.

25. H. MÜLLER-MERBACH, Drei neue Methoden zur Lösung des Traveling Salesman Problems, *Ablauf- und Planungsforschung* 7 (1966) 32-46, 78-91.

26. H. MÜLLER-MERBACH, *Optimale Reihenfolgen*, 169-170, Berlin etc., 1970.

27. A.E. OTT, Bemerkungen zu Ernst Helmstädter: Produktionsstruktur und Wachstum, *Jb.f.Nat.ök.u.Stat.* 171 (1959) 14-24.

28. R. REMAGE, JR., W.A. THOMPSON, JR., Maximum likelihood paired comparison rankings, *Biometrika* 53 (1966) 143-149.

29. H.J. RYSER, Matrices of zeros and ones in combinatorial mathematics, in: *Recent Advances in Matrix Theory*, Madison, 1964, 103-124.

30. J. SCHUMANN, *Input-Output Analyse*, 34-35, Berlin etc., 1968.

31. P. SLATER, Inconsistencies in a schedule of paired comparisons, *Biometrika* 48 (1961) 303-312.

32. J. SPENCER, Optimal Rankings of Tournaments, *Networks* 1 (1971) 135-138.

33. L. STEINBERG, The backboard wiring problem: a placement algorithm, *SIAM Rev.* 3 (1961) 37-50.