An Automated Method for Determining Alimony to Children

Vladimír Vrecion

CONTENTS: I Legal and social formulation of the problem — II Brief description of the method — III Mathematical formulation of the method

I LEGAL AND SOCIAL FORMULATION OF THE PROBLEM

The legal regulations are based on the principle that two parents (maintainers) are sharing in alimony proportionally to their net income and proportionally to the needs of the children (or other maintained)

It is necessary proportionally to consider also other alimony duties of the parents (e.g. after divorce) and other special circumstances, which influence the living-costs of individual parents. Parents and children should have as far as possible the same standard of living (also after divorce).

The alimony duties of parents should be newly determined, if some relevant factor influencing the level of alimonies is meaningfully changed ¹

The children (maintained)

a) All persons, to whom somebody's alimony duty exists, should obtain at least as much alimony as they need to cover all minimal (or conventional) living needs. Subsistence level depends on the standard of living of nations, i.e., especially on prices of basic living needs, on age of children and/or some extraordinary needs of children (because of illnesses etc.)

b) It is necessary to determine how much alimony can exceed the level of minimal or conventional living needs in the case of extraordinary solvent maintainers.

c) We reduce the level of minimal or conventional needs according to the personal income of the child maintained.

The parents (maintainers)

a) The amount of alimony, which the maintainer pays to all, to whom he has alimony duties, is proportional to his net income.

b) To the maintainer there must remain, after paying all alimony duties,

¹ We published the first theoretical approach to this problem in the article V Vrecion, J Moravek, Algorithm of determination of alimony to entitled persons. «Stat a pravo» No 2/1967 NCSAV Prague

239
at least as much as covers his minimal, or conventional needs. His minimal needs can be adjusted with regard to his extraordinary circumstances (to illness etc.)

c) In the case that according to the foregoing principles the maintainers are not able to cover the minimal or conventional needs of the children, a part of the claims of the children (maintained) remains uncovered. It is necessary to minimize this uncovered part.

II Brief description of the method

1) The exact method for determining alimony, which respects the above mentioned principles, uses procedures of mathematical theory of graphs and algebra. The method is flexible and can be operatively adapted. The method considers all relevant factors, influencing the level of alimony: level of standard living-costs, age of child, personal income of child (or of other maintained), illness of child or other extraordinary circumstances, which change his needs, net income of father and of mother, illnesses or other extraordinary circumstances influencing their needs, other alimony duties of father and of mother and personal care of children.

2) Each case of determination of alimony includes a group of children (or other maintained) and parents responsible for the alimony, bound together by the relation of alimony duties. An example of alimony situation:

\[ A_1 \rightarrow B_1 \]
\[ A_2 \rightarrow B_1 \]
\[ A_3 \rightarrow B_2 \]

\( A_1, A_2, A_3 \) are parents (maintainers), \( B_1, B_2 \) are children (maintained). Points \( A_1, A_2, A_3, B_1, B_2 \) are called « nodes ». Lines joining points \( A_1, A_2, A_3, B_1, B_2 \) are « edges » showing an alimony relation of maintainers \( A_1, A_2, A_3 \) to the maintained \( B_1, B_2 \). Nodes \( A_1, A_2, A_3, B_1, B_2 \) joined by edges will be called a « coherently orientated graph ». The graph is coherent because from every node of the graph we can move along the edges to any other node of the graph.

3) The prerequisite of the method is that the court of law ascertains a complete (or sufficiently long) coherent group of maintainers and those who are to be maintained by these maintainers (parents and children).
4) To each child we add the sum of his conventional or indispensable needs according to a special table (differentiated according to the different ages of the children). These conventional needs (derived from data about actual living standards) are fixed within the state for certain periods.

If a child has his own income, the indispensable need is in such case decreased by that amount. We also adjust the amount of conventional needs according to other counter-typical circumstances (sickness etc.; it is possible also to use special tables).

5) To every maintainer (father, mother) we add a value \( a_i \) and we call it maintenance capacity of the maintainer.

This value is determined by the equation:

\[
a_i = f(P', b_1, b_2, \ldots, b_n, E) = \min \left\{ \sum_{j=1}^{n} b_j - P' - X - E \right\}
\]

\( P' \) = net income of the maintainer decreased by any counter-typical circumstances (sickness, stay in north area etc.)

\( b_1, b_2, b_3, \ldots, b_n \) = values added to those maintained according to paragraph 4 above, to whom the maintainer has an alimony duty.

\( E \) = evaluation of personal care of the maintainer for those maintained.

The evaluation of the personal care for children we can derive from a special table (differentiated according to the age and number of children).

\( X \) = minimal living expenses.

6) In cases of quite exceptionally insolvent payers it is possible to make a simple modification of values \( a_i \) according to elementary regressive functions.

7) In the graph so evaluated we perform such algorithm operations in order that:

a) children may have their parents as much as possible covered the cost of their conventional living needs, i.e. in order that the nodes of those children may be saturated "by flow" of money along the edges;

b) the parents in a coherent component may be equally charged according to their maximal maintenance capacity: in other words the total sum of the maintainer's fees for all those maintained in a coherent group will always equal the maximal maintenance capacity multiplied for all maintainers in a coherent group by the same coefficient \( \lambda \), \( 0 \leq \lambda \leq 1 \). \( \lambda \) is, therefore, a specific criterion of justice of algorithm.

c) with respect to the demands sub 7b) maximum amount of money may "flow" from maintainers to maintained.
d) thus the conventional needs of children, which were not covered may be minimum

8) By the use of 2-7 we get the equitable distribution of alimony duties among parents (according to the coefficient $\lambda$); on the other hand the children never have more than their conventional need determined. Therefore, in the case of extraordinary solvent parents, we provide according to elementary regressive function modifications of the resulting alimony.

Activity of the Court when this method is used

On deciding about alimony in typical simple cases, the Court will have at its disposal a systematic series of examples, counted in advance according to the programme of the method on the computer. In individual cases the judge would, therefore, merely look for an analogical example and would decide accordingly.

In more complicated case the Court ascertains the necessary data about the case and solves the problem by means of a minicomputer

III Mathematical formulation of the method

We present the basic mathematical description of the method here. We describe the procedure contained in paragraph 7 (above) of the description of the method. (In paragraphs 3-6 above is a description of the simple preparation of entrance data, in § 8 is a description of the possible simple modification of the results of the basic method)

Systematics:

§ 1) a) The elements of the problem are defined in terms of the mathematical theory of graphs; b) we formulate the problem as a system of linear equations

§ 2) We provide the necessary equivalent modifications and proofs in order to solve the problem by means of Ford-Fulkerson's algorithm.

§ 3 We describe the mathematical terms and procedures used — especially Ford-Fulkerson's algorithm from mathematical theory of graphs.

§ 1) *The Problem:* Let us take a set of maintainers (parents)

\[ X = \{x_1, x_2, \ldots, x_n\} \]

and those who are maintained (children)

\[ Y = \{y_1, y_2, \ldots, y_n\} \]

is valid

\[ X \cap Y = \emptyset \]

We define the (not one-one) mapping of the set \( X \) onto the set \( Y \)

The mapping defines the relation of alimony duties of a person \( x \in X \) to person \( y \in Y \), thus \( y_j \in \Gamma x_i \) means that person \( i \) from \( X \) has a alimony duty to person \( j \) from \( X \). Let us presume that we have fixed number \( a_i \) to every maintainer, which indicates his global potential and relative ability to fulfil his total alimony duties (\( a_i \) is fixed as a function of economic, social and other qualities of an individual, his wages, alimony duties, health conditions etc.)

As a plan of alimony duties we call a system of non-negative numbers \( \varphi_{ij}, \varphi_i, \lambda, 0 \leq \lambda \leq 1, 1 \leq i \leq m, 1 \leq j \leq n \) which fulfil these relations:

1. \[ \sum_j \varphi_{ij} = \lambda a_i \]  
2. \[ \sum_i \varphi_{ij} + \varphi_j = b_j \]
3. \[ \varphi_{ij} = 0 \text{ if } y_j \notin \Gamma x_i \]

Number \( \varphi_{ij} \geq 0 \) gives alimony duty of \( i \)-nth maintainer to \( j \)-nth maintained

Equations (1) mean that in the plan of alimony is the sum of all alimony duties of \( i \)-nth maintainer equal to certain a part of \( a_i \cdot \lambda a_i \) The coefficient \( \lambda \) is equal for all maintainers (in a coherent group of parents and children)

It guarantees equitable charge to maintainers

Equations (2) show that the total need of \( j \)-nth maintained person is equal to the sum of alimony rates from all maintainers with alimony duty to \( y_j \) together with possible uncovered alimentary need \( \varphi_j \)

Equations (3) mean: In the case that \( x_i \) has not alimony duty to \( y_j \) is \( \varphi_{ij} = 0 \)

All alimony, which is not covered is

\[ \sum_j \varphi_j \]
We formulate problem I: To find a solution, which minimizes sum \( \sum \varphi_j \)

\[ \sum \varphi_j = \min \]

We can formulate this problem in an equivalent way:

\[ 0 \leq \lambda \leq 1 \]
\[ \sum_i \varphi_{ij} = \lambda \quad a_i \]
\[ \sum_i \varphi_{ij} \leq b_j \quad (2') \]
\[ \varphi_{ij} = 0 \quad \text{if} \quad y_i \notin \Gamma a_i \quad (3) \]
\[ \lambda = \max \quad (4') \]

(1), (2'), (3) (4') is problem II.

We introduce an auxiliary notion. We shall analyze network \( J_n \) (\( n \) is an even number) from nodes \( X, Y, x_0, z \), where we add new nodes \( x_0 \) and \( z \); \( x_0 \) is called entrance, \( z \) exit. \( x_0 \) is connected with each node \( x_i \in X \) with the edge of permeability \( n \quad a_i \). Each node \( y_j \in Y \) is connected with node \( z \) with the edge of permeability \( b_j \) and each node \( x_i \) is connected with node \( y_j \) (\( y_j \in \Gamma x_i \)) with the edge of permeability \( +\infty \). Other edges are not in the network.

![Fig 1](image)

The nodes \( x_1, x_m, y_1, y_n \) are called inner nodes of the network. A flow (more accurately an admissible flow) is such evaluation of the edges of the network with numbers \( \psi_i, \psi_{ij}, \psi_j \geq 0 \), for which is valid

\[ \psi_i = \sum_j \psi_{ij} \]
\[ \psi_j = \sum_i \psi_{ij} \]
and on each edge of the network the flow is limited by the prescribed limitations. In our case evidently the flow is described with a system of numbers \( \varphi_{ij} \) which fulfil the equations:

\[
\sum_j \varphi_{ij} \leq n \quad a_i \\
\sum_j \varphi_{ij} \leq b_j \\
\varphi_{ij} = 0 \quad \text{if} \quad y_j \notin \Gamma x_i
\]

As maximal flow on a given network we call admissible flow for which is valid:

\[
\sum_{ij} \varphi_{ij} = \max
\] (5)

§ 2) We have to solve the linear programme:

\[
\sum_j \varphi_{ij} = \lambda \quad a_i, b_j > 0 \\
\sum_j \varphi_{ij} \leq b_j, \quad \varphi_{ij} \leq 0
\]

\[
\lambda = \max \quad (\varphi_{ij} = 0 \Rightarrow y_j \notin \Gamma x_i)
\]

Denote maximal \( \lambda \) as \( \lambda_{\text{max}} \)

\[
X = \{x_1, \ldots, x_m\} \text{ are maintainers} \\
Y = \{y_1, \ldots, y_n\} \text{ are maintained}
\]

\[
a(A) = \sum_{i, x_i \in A} a_i, \text{ where } A \subset X, \quad b(B) = \sum_{j, j_j \in B} b_j, \text{ where } B \subset Y
\]

**Theorem 1**: Let \( 0 \neq A \subset X \) If \( n \quad a(A) > b(\Gamma A) \) then

\[
a > \lambda_{\text{max}}
\]

**Proof**: \[\lambda_{\text{max}} \quad a(A) = \sum_{i, x_i \in A} \varphi_{ij} = \sum_{j \in \Gamma A, i \in A} \varphi_{ij} \leq \sum_{j \in \Gamma A, i} \varphi_{ij} \leq b(\Gamma A) < n \quad a(A) \]

\[
\lambda_{\text{max}} < n
\]

**Theorem 2**: \[\lambda_{\text{max}} = \min \frac{b(\Gamma A)}{a(A)} \]

\[A, \; \emptyset \neq A \subset X\]

Proof follows immediately from theorem 1

245
Theorem 3: Let \( \sum_j \phi_{ij} \leq n \cdot a_i \)
\[ \sum_i \phi_{ij} \leq b_j \]

is a problem about maximal flow on the network \( J_n \). If \( \phi_{ij} \) is a maximal flow, which does not saturate the entrance edges of the network (the edges with evaluation \( n \cdot a_i \)), \( \sum_j \phi_{ij} < n \cdot a(x) \) and if \( (R, S) \) is the corresponding minimal cut, let us put \( A = X \cap R \)
then
\[ n \cdot a(A) > b(\Gamma A) \]

**Proof:** \( \Gamma A \subseteq Y \cap R \)

\[
(i \in A, j \in Y - \Gamma A) \Rightarrow \phi_{ij} = 0
\]
\[
(i \in X - A, j \in \Gamma A) \Rightarrow \phi_{ij} = 0
\]

Hence \( b(\Gamma A) = \sum_{j \in \Gamma A} \phi_{ij} = \sum_{i \in A} \phi_{ij} = \sum_{i \in A} \phi_{ij} < n \cdot a(A) \)

**Theorem 4:** \( \lambda_{\text{max}} = \min \frac{b(\Gamma A)}{a(A)} \)

\( \emptyset \neq A \subset X \)

Proof follows immediately from theorems 2 and 3.

**Algorithm:**

1. We put the starting \( n_0 = \frac{b(Y)}{a(X)} \)

2. If we have found \( k \geq 0 \), let \( \phi_{ij} \) is maximal flow. If \( \phi_{ij} \) saturates the entrance edges of the network 3 otherwise follows otherwise 2 follows

2. We construct cut \( (R, S), A = X \cap R \) and put

\[ \mu_{k+1} = \frac{b(\Gamma A)}{2(A)} \]

As starting (admissible) flow we take \( \phi'_{ij} = \frac{\mu_{k+1}}{\mu_k} \phi_{ij} \)

We seek maximal flow \( \phi_{ij(k+1)} \) We pass to point 1 of the algorithm 3 Algorithm ends, \( \phi_{ij} \) is the solution

Note: Algorithm ends after a finite number of steps because it is only finitely many of the sets \( A, \emptyset \neq A \subset X \)

246
§ 3) Description of mathematical notions and procedures used

Network
Flow on network
Maximization of flow on network
Minimization of cur
Theorem of Ford-Fulkerson

By network we mean a set of nodes $X$, with two nodes $x_0$ and $z$, $x_0$ is entrance, $z$ is exit and with evaluation of the Cartesian product $X \times Y$. Then to each couple $x$, $y$ is added number $c(x, y)$. Each node $x_0 \neq x \neq z$ is called inner node of network.

As flow on network we mean the real function $\varphi(x, y)$, $x \in X$, $y \in Y$, which has these properties:

1° $\varphi(x, y) = -\varphi(y, x)$
2° $\sum_{y \in x} \varphi(x, y) = 0$, $x_0 \neq x \neq z$
3° $\varphi(x, y) \leq c(x, y)$

1° means: flow from $x$ to $y$ is equivalent to flow from $y$ to $x$ with negative sign-property of antisymmetry;

2° means: formulation of the law of conservation of matter on network

Algebraic sum of flow in node $x$ equals 0;

3° binds flow with limitations $c(x, y)$

Example: Nodes $x \in X$ are nodes of an electric circuit. Couples $x$, $y$ are conductors (possibly zero conductors), $\varphi(x, y)$ is the intensity of the electric current. Similarly, flow of water through complex branched pipe we can describe with a network.

We introduce several definitions:

Let $A \subset X$, $B \subset X$ be a set of nodes of a network.

We put

$$\varphi(A, B) = \sum_{\substack{a \in A \\ b \in B}} \varphi(a, b)$$

$\varphi(A, B)$ is total flow from $A$ to $B$.

$$c(A, B) = \sum_{\substack{a \in A \\ b \in B}} c(a, b)$$

c($A$, $B$) is total limitation from $A$ to $B$. 

247
We show several consequences from the mentioned definitions:

\[ \varphi(x, x) = 0 \quad \text{from 1}^o \]
\[ \varphi(A, B) = -\varphi(B, A) \quad \text{from 1}^o \]

2° we can rewrite in this way:

\[ x_0 \neq x \neq z \quad \varphi(x, X) = \varphi(X, x) = 0 \]

It is valid

\[ \varphi(A, X) = 0 \quad x_0 \in A \]
\[ z \in A \]
\[ \varphi(A, B) = c(A, B) \quad \text{from 3}^o \]
\[ \varphi(A \cup B, C) = \varphi(A, C) + \varphi(B, C) \]

if

\[ A \cap B \neq \emptyset \]

The meaning the latter: \( \textcircled{A} \textcircled{B} \textcircled{C} \) Total flow from \( A \cup B \) we get with the sum \( \varphi(A, C) + \varphi(B, C) \)

**Definition** Decomposition \((R, S)\) of set \( X \) we call cut

if \( X = R \cup S, \quad R \cap S = \emptyset \)
and if \( x_0 \in R, \quad z \in S \)

R — lower group of cut
S — upper group of cut

**Theorem 1:** It is valid: if \( \varphi \) is the flow made in the network and \((R, S)\) and \((R', S')\) are two arbitrary cuts then

\[ \varphi(R, S) = \varphi(R', S') \]

The theorem has quite a plain meaning: "from the front to the back" on network flows amount \( \varphi(H, S) \)

Finally we adduce an auxiliary theorem:

**Lemma:** If \( A \) is a set merely of inner nodes, i.e

\[ x_0 \notin A, \quad z \notin A \] then

\[ \varphi(A, X) = 0 \]
If \( x_o \neq a \neq z, \ a \in \mathcal{A} \), then \( \varphi(a, X) = 0 \)

and \( \varphi(A, X) = \sum_{a \in \mathcal{A}} \varphi(a, X) = 0 \)

Proof of theorem 1: Firstly we prove the assertion with the special precondition that \( R \subset R' \) and \( S \supset S' \)

We obtain from the equation, which we are to prove \( \varphi(R, S) = \varphi(R', S') \)

and we shall equivalently modify it:

\[
\begin{align*}
\varphi(R, S) &= \varphi(R, S') + \varphi(R' - R, S') \\
\varphi(R, S') &= \varphi(R, S) - \varphi(R, S - S') \\
\varphi(R, S) &= \varphi(R, S) - \varphi(R, S - S') + \varphi(R' - R, S') \\
(S - S' &= R' - R) \\
\varphi(R, R' - R) &= \varphi(R' - R, S') \\
\varphi(R, R' - R) &= \varphi(R' - R, X - R') \\
\varphi(R' - R, X) &= 0 \\
\varphi(R, R' - R) &= - \varphi(R' - R, R') \\
\varphi(R, R' - R) &= - \varphi(R' - R, R) - \varphi(R', - R, R' - R)
\end{align*}
\]

is however fulfilled because

\( \varphi(A, B) = - \varphi(B, A) \)

By it the proof is accomplished. Generally we prove it so that we prove:

\( (R \cap R', S \cup S') \) is cut \( (R_0, S_0) \),

\[
R_0 = R \cap R' \\
S_0 = S \cup S'
\]

by it \( R_0 \subset R, R' \)

\( S_0 \supset S, S' \) and \( \varphi(R', S') = \varphi(R_0, S_0) = \varphi(R, S) \) is valid according to the foregoing

It remains then to prove that \( (R_0, S_0) \) is cut

\[
\begin{align*}
R_0 \cap S_0 &= (R \cap R') \cap (S \cup S') - (R \cap R' \cap S) \subset (R \cap R' \cap S') = \emptyset \\
R_0 \cup S_0 &= (R \cap R') \cup (S \cup S') = \underbrace{(R \cup S \cup S')}_{X} \cap \underbrace{(R' \cup S \cup S')}_{X} = X
\end{align*}
\]

Value \( \varphi(R, S) \), which is constant for all cuts of the given network, we call the value (work) of flow

Number \( c(R, S) \) is called permeability of cut

Theorem 2:

\( \varphi(R, S) \leq c(R, S) \)

Proof is trivial
Problem one: to find flow with maximal work $\varphi(R, S) = \max$
Problem two: to find cut with minimal permeability $c(R, S) = \min$.

**Theorem 3:** If $\varphi(R, S) = c(R, S)$, then flow is maximal and cut is minimal (i.e. with maximal permeability)

The problem about maximal flow is a dual problem of the problem about minimal cut

**Definition:** Let us take flow $\varphi$ on network. Edge $(x_i, y_j)$ is called unsaturated, if $\varphi(x_i, y_j) < c(x_i, y_j)$; path $x_0, x_1, x_2, \ldots, x_k$ from mutually different nodes is called unsaturated, if each edge $(x_0, x_1), (x_1, x_2), \ldots, (x_{k-1}, x_k)$ is unsaturated,

$$\begin{align*}
\begin{cases}
    c(x_0, x_1) - \varphi(x_0, x_1) \\
    c(x_1, x_2) - \varphi(x_1, x_2) \\
    \vdots \\
    c(x_{k-1}, x_k) - \varphi(x_{k-1}, x_k)
\end{cases}
\end{align*}$$

Number $h = \min$

we call measure of the unsaturation of the path

Algorithm for increasing of flow on path:
We are seeking an unsaturated path from $x_0$ to $z$

If such a path is found we determine its measure of unsaturation and we increase all the work of flow by $h$. Then we define the new flow more accurately:

$$\varphi'(x, y) = \begin{cases}
\varphi(x, y) & \text{if } (x, y) \text{ are not neighbouring nodes of the path} \\
\varphi(x, y) + h & x = x_{i-1}, y = x_i \\
\varphi(x, y) - h & x = x_i, y = x_{i-1}
\end{cases}$$

If the limitations on edges $c$ are rational numbers, we get maximal flow after a finite number of steps. Thus step by step we increase the flow until we get maximal flow. Symptom of maximal flow is that there does not exist an unsaturated path from $x_0$ to $z$.

We formulate it now more in detail
Theorem 4: If the algorithm has ended, flow is maximal

Proof: Let us denote as $R$ the set of all nodes which it is possible to reach from $x_0$ along an unsaturated path

According to precondition $z \in R$, $S = X - R$. Then $(R, S)$ is a cut of network. It is enough to prove that

$$\phi(R, S) = c(R, S)$$

Let $x \in R$, $y \in S$. We affirm that $\phi(x, y) = c(x, y)$; if $\phi(x, y) < c(x, y)$ we prove that $y \in R$ and it is a contention.

There really exists an unsaturated path $x_0, x_1, \cdots, x_k, y$, $y$ is also unsaturated and then according to definition $y \in R$.

Theorem 5: (Ford-Fulkerson)

If $(R, S)$ is the arbitrary minimal cut and $\phi$ arbitrary maximal flow then $\phi(R, S) = c(R, S)$

Proof from foregoing is trivial.

It remains to formulate a method of construction of an unsaturated path. We describe algorithm:

Let $\phi$ be some flow on the network. We construct an orientated graph $(X, \Gamma \phi)$, where $\Gamma \phi$ is relation of unsaturation.

$$(y \in \Gamma x \iff (\phi x, y) < c(x, y))$$

We construct sets of nodes $A_0$, $A_1$, $A_2$, $\cdots$, $A_k$, $A_m$ (in finite number) in the following way:

$$A_0 = \{x_0\}$$
$$A_1 = \{\Gamma y A_0 - A_0\}$$
$$A_2 = \{\Gamma y A_1 - (A_0 \cup A_1)\}$$

$$A_k = \{\Gamma y A_{k-1} - (A_0 \cup A_1 \cup \cdots \cup A_{k-1})\}$$
Let \( m \) be minimal number of this quality:

Either 1) \( (A_m \in z \) is valid or 2) \( A_{m+1} = \emptyset \)

In case 1) unsaturated paths exist. We construct one of them \((x_0, x_1, \ldots, x_{m-1}, x_m)\) in this way:

\[
\begin{align*}
x_m &= s \\
x_{m-1} &\in A_{m-1}, \Gamma_y x_{m-1} \ni z \\
x_{m-2} &\in A_{m-2}, \Gamma_y x_{m-2} \ni x_{m-1}
\end{align*}
\]

\[
x_{k-1} \in A_{k-1}, \Gamma_y x_{k-1} \ni x_k
\]

\[
x_0 \in A_0, \Gamma_y x_0 \ni x_1
\]

The path is constructed in a general case evidently not uniquely.
In case 2) the algorithm has ended, flow is maximal and cut \((R, S)\) (where \( R = A_0 \cup A_1 \cup \cdots \cup A_m \)) is minimal.