
Semantics for Reducing Complexity and Improving Accuracy in Model Creation Using Bayesian Network Decision Tools

Oscar Kipersztok
Boeing Research & Technology
P.O.Box 3707, MC: 4C-77
Seattle, WA 98124
oscar.kipersztok@boeing.com

Abstract

The work presented simplifies and makes accessible the process of using advanced probabilistic models to reason about complex scenarios without the need for advanced training. More specifically, it greatly simplifies the effort involved in building Bayesian Networks for making probabilistic predictions in complex domains. These methods typically require trained users with a sophisticated understanding of how to build and use these networks to predict future events. It entails the creation of simplified semantics that keeps the complexity of the methodology transparent to users. We provide more precise semantics to the definition of concept variables in the domain model, as well as using those semantics to assign more precise and robust meaning to predicted outcomes. This work is presented in the context of a tool and methodology, called *DecAid*, where complex cognitive models are created by defining domain-specific concepts using free language and defining relations and causal weights between them. In response to a user query the *DecAid*, unconstrained, directed graph is converted into a Bayesian network to enable predictions of events and trends.

1 INTRODUCTION

DecAid is a hypothesis-driven decision support tool that facilitates complex strategic decisions with features that allow for easy, fast, knowledge capture and modeling in complex domains. It identifies the key variables relevant to a specific query. While the cognitive, unconstrained, model is built, the defined concepts are used to create a probabilistic model to forecast events and trends. Similarly, the free-language used to define and label the concepts is used to generate a document search classifier to retrieve evidence for validation of hypotheses raised by the predictive model. *DecAid's* goal is to predict

likelihood, impact and timing of events and trends (Kipersztok, 2004).

DecAid is aimed at strategic decision making where the risk of making the wrong decision can be very costly and where there is need for argumentative rigor and careful documentation of ideas, associations and assumptions leading to the final decision. The modeling methodology was created to enable domain experts to create Bayesian networks (BN) without having to familiarize with the theory of graphical probabilistic networks or the practice of how to build them. Such users may not also require the involvement of a knowledge engineer. At the levels where high impact decisions are made, requiring high-level of abstraction and dealing with large number of variables and interdependencies, it is less likely that decision makers will use advanced decision analytic tools requiring learning specialized methodology to define and represent complex domain knowledge. The overall goals and requirements identified for the development of the *DecAid* tool were described in (Kipersztok, 2007).

In a world of rapid change it is increasingly challenging to stay abreast of occurring events and trends, making it more difficult to process information without the use of advanced technology tools designed to manage complexity and large volumes of information. Furthermore, strategic decision makers recognize the need for argumentative explanations to strategic decisions that capture the hypothetical reasoning and the evidential context behind each decision. For these reasons the need arises to rely on advanced methods to gather, organize, process and analyze data and knowledge.

Bayesian networks practitioners recognize the need to make the technology more accessible to end users due to the challenges presented during the model creation process. Some of the most significant challenges that *DecAid* aims to address are: 1) the complexity in eliciting expert knowledge, 2) defining a, potentially, large number

of parameters and relations in a particular domain, 3) adhering to conditional independence constraint in the definition of causal variables, and 4) requiring to avoid feedback reasoning during model creation that may result in graphs with cycles.

The first challenge has been addressed by various software packages (e.g., Netica, GeNIe, Hugin, etc.) that enable users to build BN with user-friendly interfaces equipped with knowledge elicitation tools. Learning algorithms have also provided the means for automated construction of BN structures and their parameters from data. To address the second challenge, canonical structures have been defined that reduce the number of parameters needed to construct conditional probability tables (CPT). (Farry et al, 2008) review several canonical models, including Influence Networks (Rose and Smith, 1996), Noisy-OR, Noisy-MAX, Qualitative Probabilistic Networks (QPN) and Causal Influence Models (CIM). They, in particular, emphasize usability of CIM models where the causal influence of each parent is captured by a single number and the combined influence of all parents is the mean of the individual parent values. (Pfautz et al, 2007) address the first three challenges and describe additional ones in findings from in-depth analyses of their experience in facilitation of model construction from numerous projects.

The purpose of this work is to describe formal semantics that enable *DecAid* to be directly accessible to domain experts to create BN models without having to concern themselves with these challenges. These semantics are aimed at easing the constraints imposed by the aforementioned challenges by enabling users to define concepts and their relations in free-association mode. Concepts are defined and labeled using free language and a single numerical weight is assigned to each parent-child relation. This effort results in the creation of the *DecAid* (unconstrained) network (DN), a directed graph, which allows cycles. The step of creating a BN from the DN starts with a query definition, and it involves the identification of the query-specific sub graph and removal of its cycles by, optimally, minimizing the information loss. The result is BN directed acyclic graph specific to the query.

2 FROM DECAID NETWORKS TO BAYESIAN NETWORKS

DecAid is a system for simple but powerful probabilistic modeling of arbitrary scenarios. It enables domain expert to create *DecAid* networks by defining *concepts* with free language and causal *relations* between them. For each pair of relations, the user assigns a *weight of causal belief*. There are two types of concepts: a) *Event* concepts that represent quantities that can occur or not-occur; and b) *Trend* concepts that represent quantities that *increase*,

remain *unchanged*, or *decrease*. Various levels of granularity can be selected to define the *trend* concept states.

In this section we describe the formal definitions that enable the creation of a DN and its subsequent conversion into a BN.

2.1 Definition of a *DecAid* Network (DN)

Similar to a Bayesian network, each *DecAid* variable (DV) represents a concept, which is some aspect of the domain modeled. More specifically, a DV defines a probability distribution over its possible values and it is discrete—i.e., finite-valued and typically taking 2, 3, 5, or 7 values. For example, we might have a DV named ‘Barometric Pressure’ that has 3 values: ‘decreasing’, ‘unchanged’, and ‘increasing’. The set of values is taken to have some natural ordering so that we can speak of high values versus low values. If the variable is binary, we would say that values such as false / off / does-not-occur would be “low” compared to true / on / occurs.

More formally, a *DecAid* model \mathbf{M} includes a set \mathbf{V} of DVs and, taken together, the variables in \mathbf{V} jointly describe a distribution over the entire scenario modeled by \mathbf{M} . Along with the set \mathbf{V} , the model \mathbf{M} includes a directed graph structure \mathbf{G} connecting the variables of \mathbf{V} . Each variable in \mathbf{V} is a node of \mathbf{G} and each arc denotes a direct probabilistic influence of the parent’s value on the distribution over the child’s values. The directed graph \mathbf{G} is *unconstrained*—all connections are allowed and cycles are permitted. Each arc is labeled with a single real number between -1 and 1 called the *weight*. Intuitively, the closer $|w|$ is to 1 , the stronger the influence of the parent over the child and the closer $|w|$ is to 0 , the weaker the influence. If the weight is positive, a high parent value makes high child values more likely and a low parent value make low child values more likely. A negative weight flips the influence so that a high parent value makes low child values more likely and a low parent value makes high child values more likely (other things being equal). Note that a moderate parent value will make moderate child values more likely.

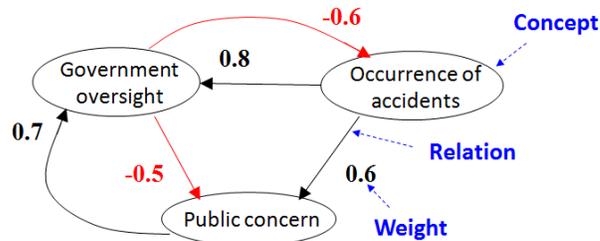


Figure 2.1: A *DecAid* Unconstrained Model

Figure 2.1 shows an example of an unconstrained DN representing three concept variables in the Aviation Safety domain and five parent-child relations with their corresponding weights.

Once, the unconstrained model is built, *DecAid* is capable of transforming the DN into a BN in order to make predictions in response to queries.

2.2 Transforming a *DecAid* Network (DN) into a Bayesian Network structure

A user can make a query to the DN by defining a set of *observation variables* and a *target variable*. In response to the query, *DecAid* is capable of transforming the unconstrained (directed graph) model to a Bayesian network by carrying out the following sequence of steps:

- 1) Identifying all cycles in the unconstrained model. We use an algorithm by (Johnson, 1975) that finds the elementary cycles in the directed graph by improving over the original algorithm by (Tarjan, 1973);
- 2) Eliminating the cycles in the unconstrained model by removing the weak edges. This is done, optimally, in order to minimize the information loss in the unconstrained model. This step constitutes a tradeoff between increased expressive power for domain-expert users and modest information loss resulting from removal of edges that least contribute to the information flow.
- 3) Identifying the sub graph relevant to the query by pruning the non relevant variables from the resulting Bayesian network (Geiger et al, 1990). This step constitutes an important feature of *DecAid* in that it can list all the relevant parameters to the user that are relevant to a specific user query.

The last step in the creation of a query specific Bayesian network is the creation of the conditional probability tables (CPT). The semantics to achieve that are described in section 3.

For practitioners involved in high-level, strategic, decision making the use of Bayesian network building tools can be counterintuitive and may require significant training time, unavailable to such intended users. Making, however, the BN technology accessible through tools like *DecAid* not only will improve the accuracy of decision making but will also provide the means to document and track the chain of causal reasoning behind each decision.

3 SEMANTICS TO CREATE CONDITIONAL PROBABILITY TABLES

What follows is a description of the method used to express the random variable (RV) encoded by a DV. That is, we show how to calculate a conditional probability table (CPT) for each variable in the *DecAid* model given its parent set and the size of each variable.

3.1 Concepts Defined as Random Variables

Let X be an n -valued DV from a *DecAid* model \mathbf{D} . We say that the sample space \mathbf{S} for X is the real interval $[0,1)$. That is, we can suppose that X describes an experiment whose outcome is a real number r such that $0 \leq r < 1$. The values of the random variable X break the sample space into n disjoint events—namely, half-open intervals of equal length. The set of events is thus:

$$\{ r \in [k/n, (k+1)/n) : \text{for } 0 \leq k < n \}$$

Example (3.1.1)

If X has 2 states, the events corresponding to the states of X are:

$$\{ r \in [0.0, 0.5), r \in [0.5, 1.0) \}.$$

Example (3.1.2)

If X has 5 states, the events corresponding to the states of X are:

$$\{ r \in [0.0, 0.2), r \in [0.2, 0.4), r \in [0.4, 0.6), r \in [0.6, 0.8), \text{ and } r \in [0.8, 1.0) \}.$$

3.2 Conditional Probability Tables

The heart of the probabilistic semantics is the definition of local conditional probability distributions for *DecAid* variables. We consider the various cases below: a) where the variable has no parents, b) where it has one parent of weight 1, c) where it has one parent of arbitrary weight, and finally, d) where it has any number of parents.

Case 3.2.1 -Variables without parents

If X has no parents in \mathbf{D} , then it is simply given a uniform distribution:

$$P(X = x_k) = 1/n \quad \text{for } 0 \leq k < n.$$

That is, the event $X = x_k$ corresponds to $r \in [k/n, (k+1)/n)$. The probability equals the proportion of the total length of \mathbf{S} contributed by $X=x_k$. Since the total length of \mathbf{S} is 1.0, it is simply equal to the length of the interval, which is $(k+1 - k)/n = 1/n$.

Example (3.2.1.1)

If X has 2 states, $P(X = x_k) = 0.5$ for $0 \leq k \leq 1$.

Example (3.2.1.2)

If X has 5 states, $P(X = x_k) = 0.2$ for $0 \leq k \leq 4$.

Case 3.2.2 – Variables with one parent and $|w| = 1$

We first describe the case where we have a single parent Y and where the link from Y to its child X has weight 1. We need to show how to calculate the conditional probability $P(X = x_k | Y = y_j)$. This is given by the formula:

$$P(X = x_k | Y = y_j, w = 1) = P(X = x_k \ \& \ Y = y_j) / P(Y = y_j) .$$

That is, the conditional probability of the event $X = x_k$ given that $Y = y_j$ is equal to the intersection of the intervals corresponding to these events divided by the length of the interval corresponding to $Y = y_j$.

Example (3.2.2.1)

Suppose $Y \rightarrow X$ and Y has 5 states and X has 2 states,

$$P(X = x_0 | Y = y_2) = | \text{Intersection of } [0, 0.5] \ \& \ [0.4, 0.6] \ | / | 0.6 - 0.4 | = 0.5$$

The full CPT would be:

$P(x y)$	x_0	x_1
y_0	1	0
y_1	1	0
y_2	0.5	0.5
y_3	0	1
y_4	0	1

Example (3.2.2.2)

Suppose $Z \rightarrow X$ and Z has 3 states and X has 5 states,

$$P(X = x_0 | Z = z_0) = | \text{Intersection of } [0, 0.2] \ \& \ [0, 0.33] \ | / | 0.33 - 0 | = 0.6$$

The full CPT is:

$P(x z)$	x_0	x_1	x_2	x_3	x_4
z_0	0.6	0.4	0	0	0
z_1	0	0.2	0.6	0.2	0
z_2	0	0	0	0.4	0.6

Example (3.2.2.3)

Suppose $Y \rightarrow X$ and Y has 2 states and X has 5 states,

$$P(X = x_0 | Y = y_0) = | \text{Intersection of } [0, 0.2] \ \& \ [0, 0.5] \ | / | 0.5 - 0 | = 0.2 / 0.5 = 0.4$$

The full CPT is:

$P(x y)$	x_0	x_1	x_2	x_3	x_4
y_0	0.4	0.4	0.2	0	0
y_1	0	0	0.2	0.4	0.4

Case 3.2.3 – Variables with one parent and $|w| < 1$

We next look at the case where the weight is different than 1. It is useful to refer to the distribution defined in Case 2a as the *full-weight* distribution—i.e., where $w=1$.

Let $P_{\text{full}}(X | y_j)$ be the distribution over the values of X given $Y = y_j$ under the assumption that the arc from Y to X has weight $w = 1$. Let $U(X)$ be the uniform distribution over the values of X . Then, if the weight is $0 \leq w < 1$, we have

$$P(X | y_j, 0 \leq w < 1) = w \cdot P_{\text{full}}(X | y_j) + (1 - w) \cdot U(X)$$

That is, the final distribution is a weighted combination of the distribution calculated in Case 3.2.1 and the uniform distribution—which is the default distribution if there were no parent. Note that the weight acts as the probability that we get the full-weight distribution instead of a uniform distribution.

Example (3.2.3.1)

Following the previous example (II.2.3), suppose $Y \rightarrow X$ and Y has 2 states and X has 5 states. But now suppose that the weight of the arc is $w = 0.6$, then we have

$$\begin{aligned} P(X = x_0 | Y = y_0) &= w \cdot P_{\text{full}}(X | y_0) + (1 - w) \cdot U(X) \\ &= 0.6 \cdot 0.4 + (1.0 - 0.6) \cdot (1/5) \\ &= 0.24 + 0.4 \cdot 0.2 = 0.3 + .08 = 0.32 \end{aligned}$$

The full CPT is:

$P(x y)$	x_0	x_1	x_2	x_3	x_4
y_0	0.32	0.32	0.2	0.08	0.08
y_1	0.08	0.08	0.2	0.32	0.32

If the weight is negative, the direction of the parent’s influence is reversed. If Y is an m -valued variable, we can calculate the resulting distribution using a similar calculation above but for the “opposed” value of the parent. By “opposed” we mean the value at the other side of the range—i.e., highest is opposed to lowest, second-highest is opposed to second-lowest, etc. More specifically, if the weight $w < 0$, we have

$$P(X | y_j, -1 \leq w < 0) = w \cdot P_{\text{full}}(X | y_{m-j-1}) + (1 - w) \cdot U(X)$$

$$= c \cdot [0.03, 0.03, 0.02, 0.03, 0.04]$$

$$= [0.2, 0.2, 0.13, 0.2, 0.27]$$

Example (3.2.3.2)

Following the previous example (II.2.3), suppose $Y \rightarrow X$ and Y has 2 states and X has 5 states. But now suppose that the weight of the arc is $w = -0.6$, then we have

$$P(X = x_2 | Y = y_1) = 0.6 \cdot 0.2 + (1.0 - 0.6) \cdot (1/5) = 0.12 + 0.4 \cdot 0.2 = 0.12 + .08 = 0.2$$

The full CPT is:

$P(x y)$	x_0	x_1	x_2	x_3	x_4
y_0	0.08	0.08	0.2	0.32	0.32
y_1	0.32	0.32	0.2	0.08	0.08

The full CPT is:

$P(x y, z)$	x_0	x_1	x_2	x_3	x_4
y_0, z_0	0.2	0.2	0.13	0.2	0.27
y_0, z_1	0.15	0.3	0.4	0.1	0.05
y_0, z_2	0.48	0.36	0.08	0.04	0.04
y_1, z_0	0.04	0.04	0.08	0.36	0.48
y_1, z_1	0.05	0.1	0.4	0.3	0.15
y_1, z_2	0.27	0.2	0.13	0.2	0.2

Case 3.2.4 – Variables with multiple parents

The remaining situation is when we have a variable with multiple parents. In this situation, we assume that the influence of each parent is independent of the influence of other parents. So, if X has parents Y^1, Y^2, \dots, Y^N , we set

$$P(X | Y^1, Y^2, \dots, Y^N) = c \cdot P(X | Y^1) \cdot P(X | Y^2) \cdot \dots \cdot P(X | Y^N)$$

where c is normalization constant to make the distribution sum to 1.

Example (3.2.4.1)

Suppose X has 5 states and two parents: Y with 2 states and weight 0.5 and Z with 3 states and weight -0.5 . As we saw above from examples (3.2.2.1) and (3.2.2.2), if we ignore the weights of the arcs and the fact that there are multiple parents, we have for parent Z :

$$P_{\text{full}}(X | z_0) = [0.6, 0.4, 0.0, 0.0, 0.0]$$

And for parent Y :

$$P_{\text{full}}(X | y_0) = [0.4, 0.4, 0.2, 0.0, 0.0]$$

Next, adding in the effect of the weights on the distributions, we get:

$$P(X | z_0, w = -0.5) = [0.1, 0.1, 0.1, 0.3, 0.4]$$

and

$$P(X | y_0, w = .5) = [0.3, 0.3, 0.2, 0.1, 0.1]$$

Now, combining both parents we get

$$\begin{aligned} P(X | y_0, z_0) &= c \cdot P(X | y_0) \cdot P(X | z_0) \\ &= c \cdot [0.3, 0.3, 0.2, 0.1, 0.1] \cdot [0.1, 0.1, 0.1, 0.3, 0.4] \end{aligned}$$

4 DISCUSSION

DecAid is used for strategic decision making. Here are a few examples of such decisions: a) when to launch a new product into a specific market, b) how close is a rouge country to achieving nuclear weapon capability, or c) whether to invest in a particular emerging technology. These are decisions that involve several variables and their inter relations. The system enables decision makers to define concepts of the problem in a simple, intuitive, manner using free language. As the user defines the concepts and relations, the system is creating an unconstrained model. Once, the model is built, *DecAid* is capable of making predictions in response to queries by converting the unconstrained model into a Bayesian network.

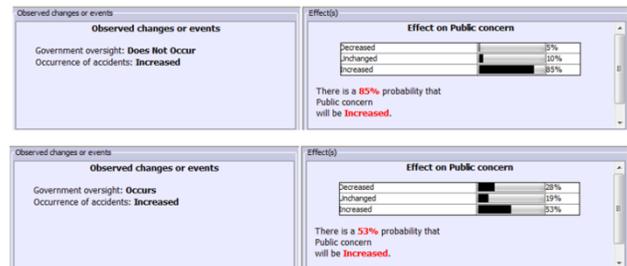


Figure 4.2: Predictions made by *DecAid* Model

Figure 4.2 shows two such predictions derived from the model in Figure 2.1. The first prediction forecasts a 0.85 probability that “Public concern” will increase given that “Occurrence of accidents” has increased and “Government oversight” does not occur. The second prediction lowers the forecast that “Public concern” will increase to a 0.53 probability, if “Government oversight” occurs.

At the final stage of making decisions, summarization and argumentation becomes critical steps. It is the aim of *DecAid* to facilitate the capture of knowledge and

information for that last stage, as well, by combining the predictive analytic capability obtained from the cognitive models with the ability to retrieve evidential data and information to validated predictive hypotheses, which is outside the scope of this paper.

The complete probabilistic semantics of a *DecAid* model include how the local probability models are combined (not discussed here). The cornerstone of the semantics, however, is the definition given here for the complete CPT of a local variable from the simple numeric weights associated with its parents as provided by the end-user creating the model.

DecAid variables represent a tradeoff between simplicity of model definition and expressive power. Aside from adding temporal modeling (Nodelman, et. al. 2002, 2003) to *DecAid*, there are additional areas where the balance between simplicity and expressivity could be further enhanced. In one such area there is, currently, complete symmetry between the positive effect of a parent taking on a high value and the negative effect of a parent taking on a low value. Sometimes this symmetry is warranted but sometimes it is not. For example, consider a child variable “Strength of a Fire” (“fire”) with a parent “Oxygen Level Present” (“oxygen”). Increasing oxygen will tend to increase the fire and decreasing oxygen will tend to lessen the fire. But now consider an alternative parent “Use of fire-extinguisher”. If the fire-extinguisher is used, that will tend to lessen the fire. But lack of fire-extinguisher use does not, in itself, increase the fire. So we may, in general, want to allow an asymmetry between the impact of a high-value parent and a low-value parent where the high-value has the regular effect but the low-value has no special impact on the child.

Furthermore, the assumption that the effects of multiple parents are independent of each other is strong. Obviously, there are many cases where this assumption is unwarranted. The problem would be to find a simple, understandable way for end-users to convey extra information about covariance and to find an algorithm that could examine the link structure in other parts of the *DecAid* model and extract some useful information about the dependencies among the parents.

5. SUMMARY

The semantics definition is given in this paper for the complete CPT of a local variable from simple numeric weights associated with its parents as provided by the end-user creating a *DecAid* model. Explicitly, 1) The values of a DV are represented as equal-length subintervals of the unit interval and making explicit that they have a natural ordering so they can be seen as coming in opposed pairs (except for a possible middle-most value). 2) A single parent full-weight conditional

probability is defined as the size of the intersection of parent and child intervals divided by the size of parent interval. 3) The magnitude of the weight is used as the probability that you get the full-weight conditional distribution instead of a uniform distribution. 4) The sign of the weight is used to reverse the direction of influence. And 5) the probabilistic influence of multiple parents on a child are assumed to be independent of one another.

The semantics described in this paper enable the creation of a Bayesian networks from an unconstrained, directed graph model created by a user within a simpler, more intuitive, framework implemented in a tool called *DecAid*, without requiring specialized training in how to build Bayesian networks.

Acknowledgement

This work would have not been possible without the contributions and valuable, long-term, collaboration with Uri Nodelman for which the author and The Boeing Company are greatly thankful.

References

- Farry M, Pfautz J, Cox Z, Bisantz A, Stone R, and Roth E (2008), “An Experimental Procedure for Evaluating User-Centered methods for Rapid Bayesian Network Construction”, *Proceedings of the 6th Bayesian Modeling Applications Workshop at the 24th Annual Conference on Uncertainty in AI: UAI 2008, Helsinki, Finland*.
- Geiger, Verma, and Pearl, (1990), "Identifying Independence in Bayesian Networks", *Networks* 20:507-534.
- Johnson, B. (1975), “Finding all the elementary circuits in a directed graph”, *SIAM Journal of Computing*, 4(1).
- Kipersztok, O (2007). “Using Human Factors, Reasoning and Text Processing for Hypothesis Validation.” *Third International Workshop on Knowledge and Reasoning for Answering Questions. International Joint Conference in Artificial Intelligence, Hyderabad, India*.
- Kipersztok O (2004). “Combining Cognitive Causal Models with Reasoning and Text Processing Methods for Decision Support.” *Second Bayesian Modeling Applications Workshop at the Uncertainty in AI Conference, Banff Canada*.
- Nodelman, U., Shelton, C. R., and Koller, D. (2002). “Continuous Time Bayesian Networks.” *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 378-387, 2002.

Nodelman, U., Shelton, C.R., and Koller, D. (2003). "Learning Continuous Time Bayesian Networks." *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pp. 451-458, 2003.

Pfautz J, Cox Z, Catto G, Koelle D, Campolongo J, Roth E (2007), "[User-Centered Methods for Rapid Creation and Validation of Bayesian Networks](#)". In *Proceedings of 5th Bayesian Applications Workshop at Uncertainty in Artificial Intelligence (UAI 07)*.

Rosen J and Smith W (1996), "Influence Net Modeling with Causal Strengths: An Evolutionary Approach," *In the Proceedings of the 1996 Command and Control Research and Technology Symposium, Monterey CA*.

Tarjan, R. E. (1972), "Depth-first search and linear graph algorithms", *SIAM Journal on Computing* 1 (2): 146–160.

Tarjan, R.E. (1973), "Enumeration of the elementary circuits of a directed graph", *SIAM Journal of Computing* 2 :211-216.