Marco Valtorta wrote these notes (mainly around 1982). Blaine Nelson typed them on October 2-4, 2002. The main reference is: Bertelè, Umberto and Francesco Brioschi. *Non-Serial Dynamic Programming*. Academic Press, 1972. Another reference used in notes written later is: Shenoy, Prakhash P. "Valuation-Based Systems for Discrete Optimization." In: P.P. Bonissone, M. Henrion, L.N. Kanal, and J.F. Lemmer (eds.). *Uncertainty in Artificial Intelligence 6*. Elsevier, 1991, pp.385-400.

Marco Valtorta

An Application of Dynamic Programming:

Globally Optimum Selection of Storage Patterns

Overview

This talk has two goals:

a) A review of the fundamentals of dynamic programming, and an introduction to nonserial dynamic programming;

b) An application of the techniques to some of the issues involved in the problem of determining globally optimum storage patterns.

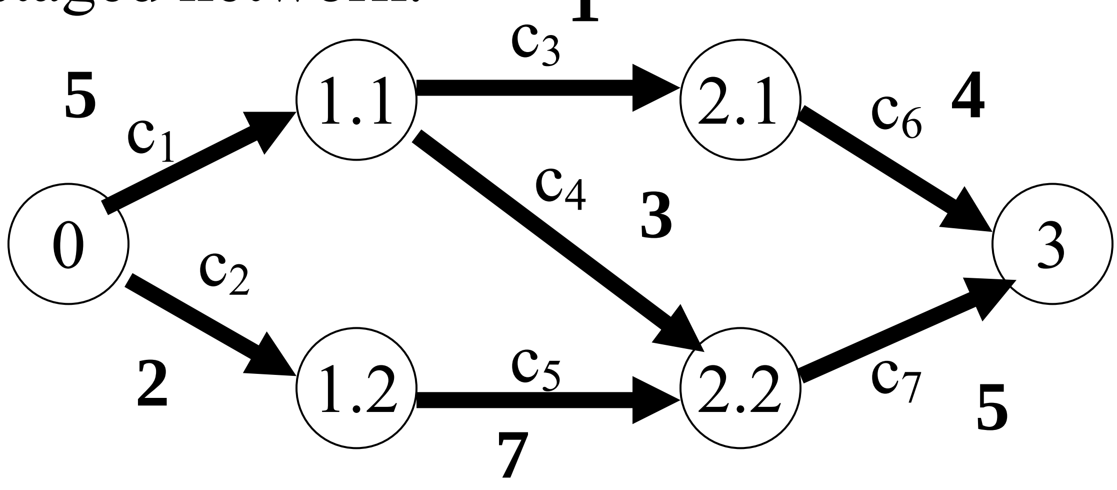
Dynamic Programming

Dynamic programming is a problem solving method which is especially useful to solve the problems to which Bellman's Principle of Optimality applies:

> "An optimal policy has the property that whatever the initial state and the initial decision are, the remaining decisions constitute an optimal policy with respect to the state resulting from the initial decision."

Example:

The shortest path problem in a directed staged network:



The principle of optimality can be stated as follows:

If the shortest path from 0 to 3 goes through X, then:

1. that part from 0 to X is the shortest path from 0 to X, and
2. that part from X to 3 is the shortest path from X to 3.

The previous statement leads to a forward and a backward algorithm for finding the shortest path in a directed staged network.

I shall now give a more formal definition of the “dynamic programming problem.” [Brioschi and Bretelè, 1972]

The statement of the nonserial (NSPD) unconstrained dynamic programming problem is

$$\min_{X} f(X) = \min_{X} \sum_{i \in T} f_i(X^i)$$

where

$X = \{x_1, x_2, \ldots, x_n\}$

is a set of discrete variables, S_{x_j} being the definition set of the variable x_j ($|S_{x_j}| = \sigma_{x_j}$);

$T = \{1, 2, \ldots, t\}$; and $X^i \subset X$.

The function f(x) is called the objective function, and the functions $f_i(X^i)$ are the components of the objective function.

Some useful definitions are now given.

Two variables $x \in X$ and $y \in X$ are said to interact if there exists a component $f_k(X^k)$ such that both x and y belong to X^k.

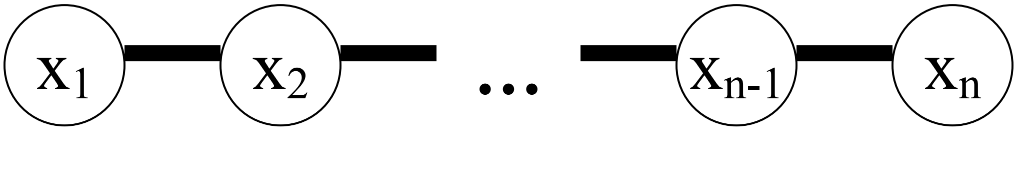
The interaction graph G = (X, L) of a nonserial unconstrained problem is an undirected graph, without self-loops and parallel edges, defined by the following:

a) The vertex set X of the graph is the set of variables of the problem.

b) Two vertices are adjacent if and only if the corresponding variables interact.

Example:

The interaction graph of a serial problem $\min_X \sum_{i=1}^{n-1} f_i(x_i, x_{i+1})$ is given below:



Rather than formally stating the way to solve a nonserial problem, I will present an example.

Example:

$$\min_{X} \{ f_1(x_1, x_2, x_3) + f_2(x_3, x_4, x_5)$$
$$+ f_3(x_4, x_5, x_6) \}$$

$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$
$X^1 = \{x_1, x_2, x_3\};$ $X^2 = \{x_3, x_4, x_5\};$
$X^3 = \{x_4, x_5, x_6\};$

f_1	x_1	x_2	x_3
1	0	0	0
3	0	0	1
5	0	1	0
8	0	1	1
2	1	0	0
6	1	0	1
2	1	1	0
4	1	1	1

f_2	x_3	x_4	x_5
4	0	0	0
8	0	0	1
0	0	1	0
5	0	1	1
3	1	0	0
5	1	0	1
8	1	1	0
2	1	1	1

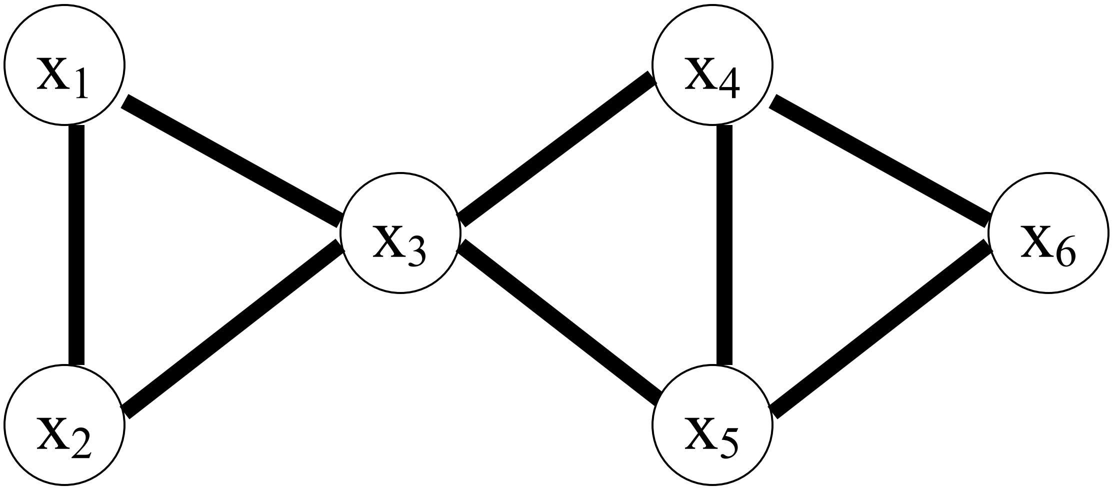
f_3	x_4	x_5	x_6
0	0	0	0
5	0	0	1
6	0	1	0
3	0	1	1
5	1	0	0
1	1	0	1
4	1	1	0
3	1	1	1

$$S_{x_1} = S_{x_2} = \ldots = S_{x_6} = \mathrm{S_X} = \{0, 1\}$$

$$|S_{x_1}| = \sigma_{x_1} = |S_{x_2}| = \sigma_{x_2} = \ldots = |S_{x_6}| = \sigma_{x6} = 2$$

$$\mathrm{T} = \{1, 2, 3\}$$

The interaction graph of the problem is:



I choose to eliminate variable x_6 first. To do so, I consider with which variable x_6 interacts: x_4 and x_5. For every assignment to x_4 and x_5, I compute the value of x_6 for which f_3 is minimal (note that x_6 is a member of X^3 only, i.e., it is only involved in the component f_3). This leads to the following table:

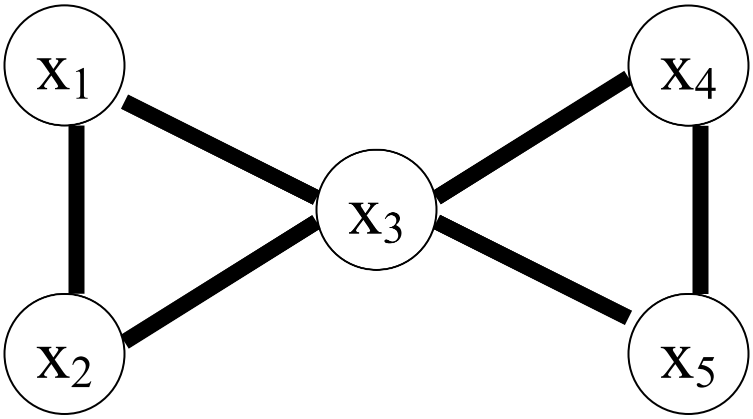
x_6*	h_1	x_4	x_5
0	0	0	0
1	3	0	1
1	1	1	0
1	3	1	1

Bellman's Principle of Optimality holds because, once he optimal values for x4 and x5 have been determined, the optimal value for x6 is x6*. Therefore, we can consider a new problem, in which x6 does not appear (x6 has been eliminated):

$$\min_{X-\{x_6\}} \{ f_1(x_1, x_2, x_3) + f_2(x_3, x_4, x_5)$$

$$+ h_1(x_4, x_5) \}$$

The interaction graph for the new problem is:



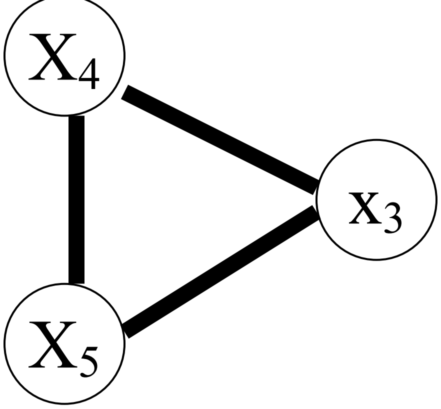
At this point, I note that x_1 and x_2 interact only with x_3 in f_1, so I decide to eliminate them in block, by building the following table.

x_1*	x_2*	h_2	x_3
0	0	1	0
0	0	3	1

The new problem to be solved is:

$$\min_{X-\{x_6, x_1, x_2\}} \{ h_2(x_3) + f_2(x_3, x_4, x_5)$$
$$+ h_1(x_4, x_5) \}$$

The corresponding interaction graph is:



I eliminate x_4 and x_5 in block, by considering both f_2 and h_1 to build h_3.

Note that

$$h_3 = \min_{\{x_4, x_5\}} \{ h_1(x_4, x_5) + f_2(x_3, x_4, x_5) \}$$

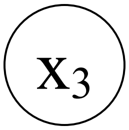
x_4*	x_5*	h_3	x_3
1	0	1	0
0	0	3	1

Now the problem to be solved is

$$\min_{X\text{-}\{x_6, x_1, x_2, x_4, x_5\}} \{ h_2(x_3) + h_3(x_3) \} \equiv$$

$$\equiv \min_{\{x_3\}} \{ h_2(x_3) + h_3(x_3) \}$$

The corresponding interaction graph is



, and the solution is

x_3* = 0, which corresponds to

$$h_2 + h_3 = 2 = \min_X \{f_1 + f_2 + f_3\}$$

To find out the optimal values of all the variables (an optimizing arrangement) we use Bellman's principle of optimality and the tables we have built.

The computational "cost" of solving a nonserial dynamic programming problem is the sum of two terms of functional evaluation and table lookups. However, "the maximum number of interacting variables of interacting variables is also a reasonable index of the computing time." [Bertelè and Brioschi, 1972].

I shall now introduce the reordering optimization problem.
Our elimination reordering is the ordered sequence of the eliminated variables

(the first variable to be eliminated is the first in the ordering).

The dimension of the ordering w, D(w) is the maximum of the degrees of the vertices in w, at the time of their elimination. This definition might be modified for the block elimination case.

Example:

The elimination ordering for the solution given in the previous example is

$w_1 = \{x_6, x_1, x_2, x_4, x_5, x_3\}$.

Its vector dimension is

$d(w_1)= \{2, 2, 1, 2, 1, 0\}$

Its dimension is:

$D(w_1) = 2$.

One could solve the same problem by eliminating variables in this order:

$w_2 = \{x_3, x_6, x_2, x_1, x_4, x_5\}$,
whose dimension is
$D(w_2) = 4$,
as can be easily verified.

In a simplified formulation, the secondary optimization problem is the problem of finding the elimination ordering with minimal dimension.
A general solution to the secondary optimization problem is computationally very heavy [Brioschi and Bertelè, chapter 3] so that heuristic criteria are used instead. The simplest criterion "which often determines an optimal or near optimal ordering" is a greedy criterion, as expressed in the minimum degree algorithm: at any step, we eliminate a minimum degree vertex in the current interaction graph.

Globally optimum storage patterns

In this part of the talk, I shall map a simplified version of the storage pattern problem into the formalism described in the previous section, describe some examples, and draw some considerations.

I shall not consider the case in which loops are present. The problem has been described in a previous talk. It can be stated formally as follows:
Get a program in the form of a sequence of binary operations and assignments on matrices, M_i be.

$$\min_X f(x) = \min_X \sum_{i \in T} f_i(X^i)$$

where

$X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ and

x_i = shape(M_i)

S_{x_i} = {all possible shapes for M_i} = CANNOT READ

$f(X^i)$ = cost of performing an operation i, with the elements of X^i as shapes. $X^i \subset X$.

T = {1, 2, …, t}, where |T| = t is the number of operations performed by the program.

|X| = n is the number of matrices the program deals with.

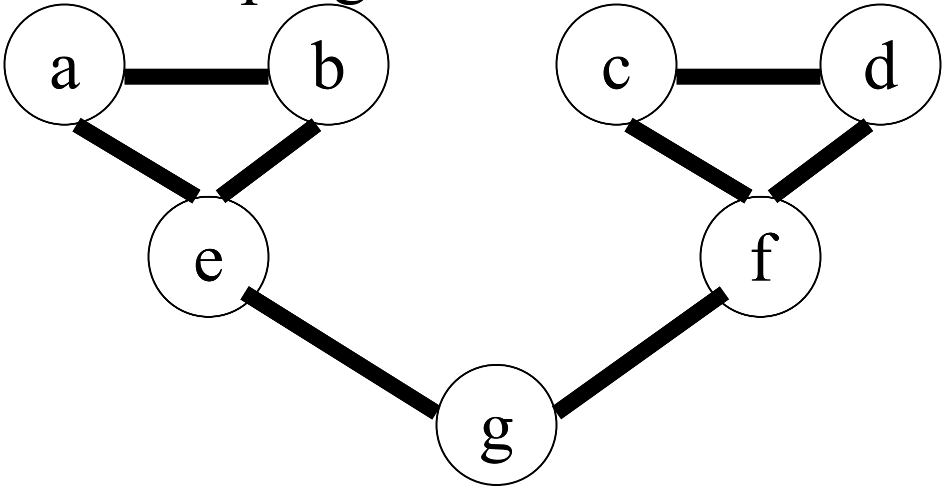
Examples:

```
E := A+B;
F := C*D;
G := E-F.
```

Let a, b, c, d, e, f, be the shapes of A, B, C, D, E, F.

$$\min_{\{a, b, c, d, e, f\}} \{ f_1(a, b, e) + f_2(c, d, f) + f_3(e, f, g) \}$$

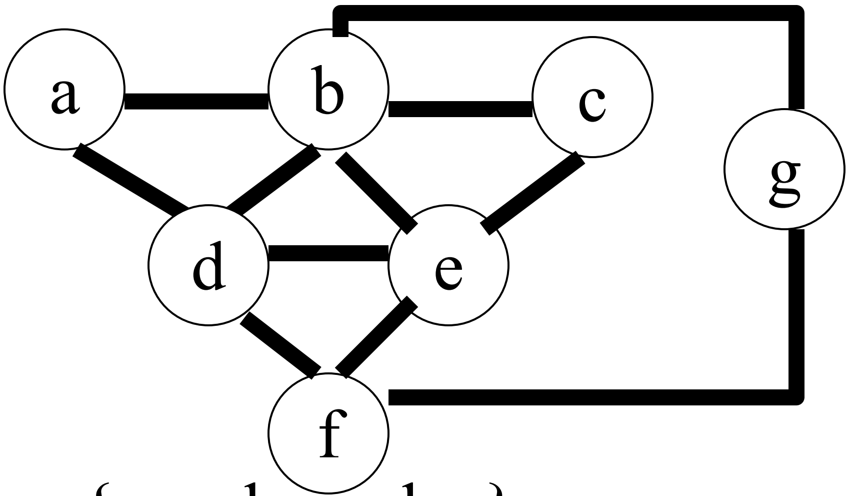
The interaction graph corresponding to the above program is:



The dimension of the elimination ordering {a, b, c, d, e, f, g} is two.

```
D := A+B;
E := B*C;
F := D-E;
G := F+B;
```

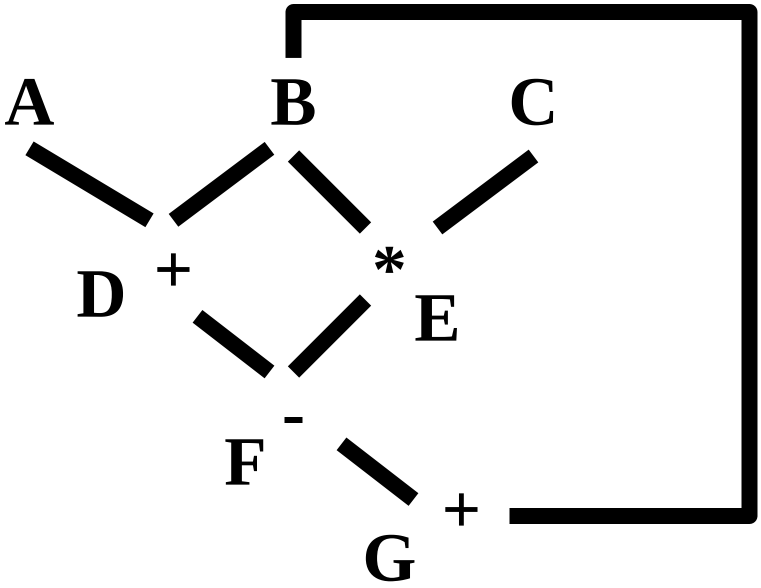
$$\min\{ f_1(a, b, d) + f_2(b, c, e) + f_3(d, e, f) + f_4(f, b, g) \}$$



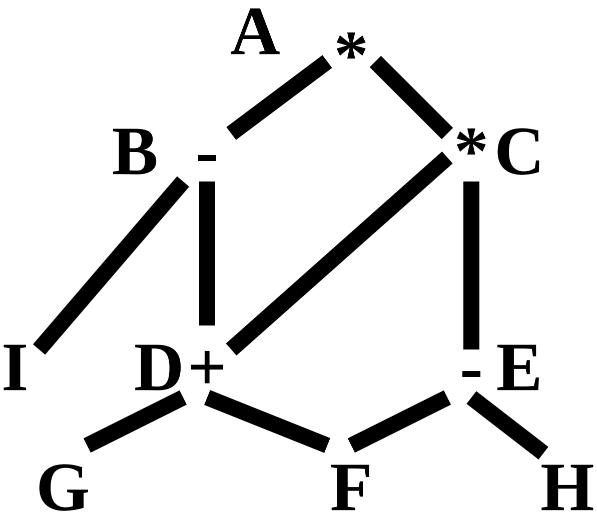
w = {a, c, b, g, d, e}

D(w) = 3.

Note: the interaction graph is not a series-parallel graph, but the program graph is.



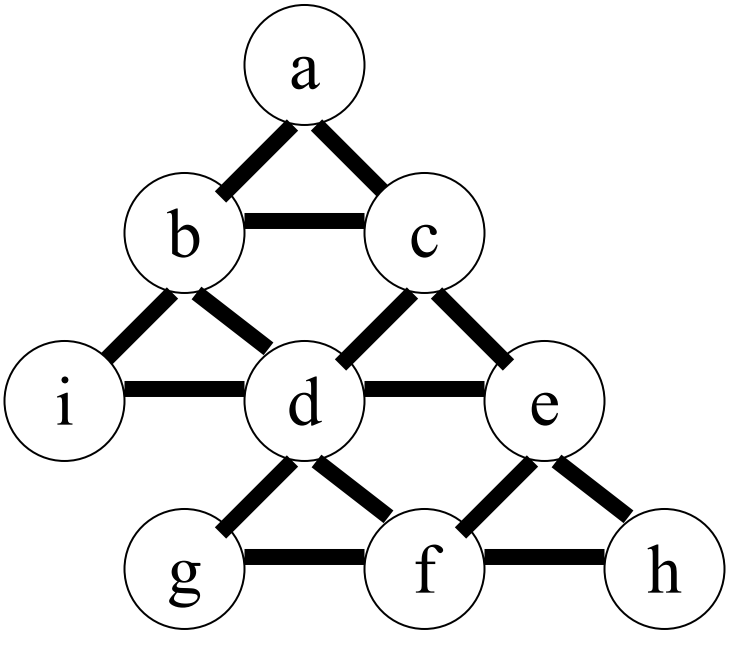
```
D := G+F;
E := F-H;
C := D*E;
B := I-D;
A := B*C;
```



The program graph is not a series parallel graph

$$\min\{ f_1(b, c, a) + f_2(i, d, b) + f_3(d, e, c) +$$

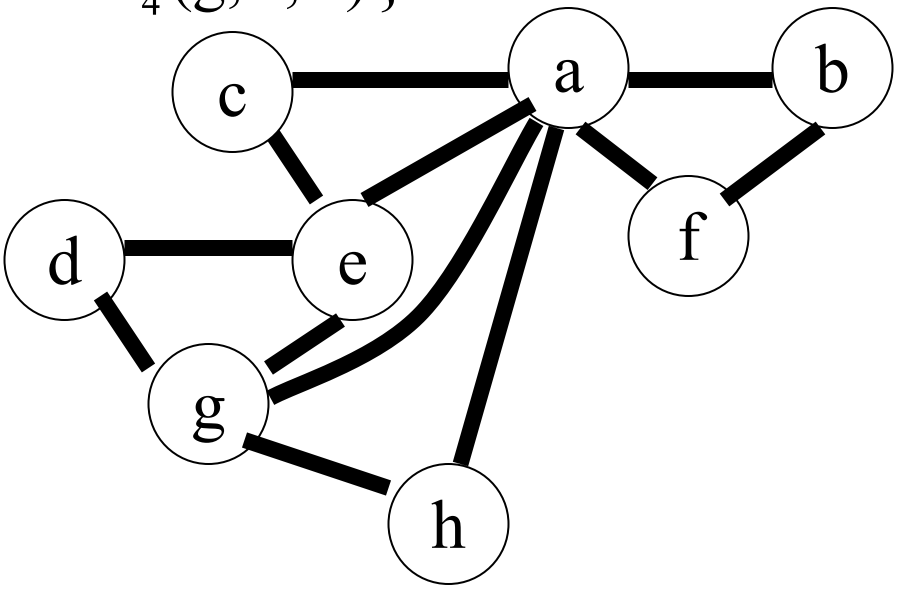
$$+ f_4(f, h, e) + f_5(g, f, d) \}$$



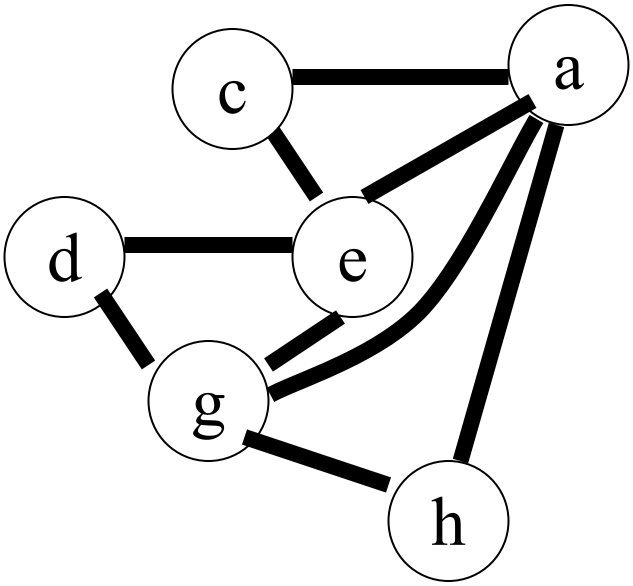
w = {i, g, b, a, f, b, e, d, c}
D(w) = 2

```
E := A*C;
F := A+B;
G := D+E;
H := G+A;
```

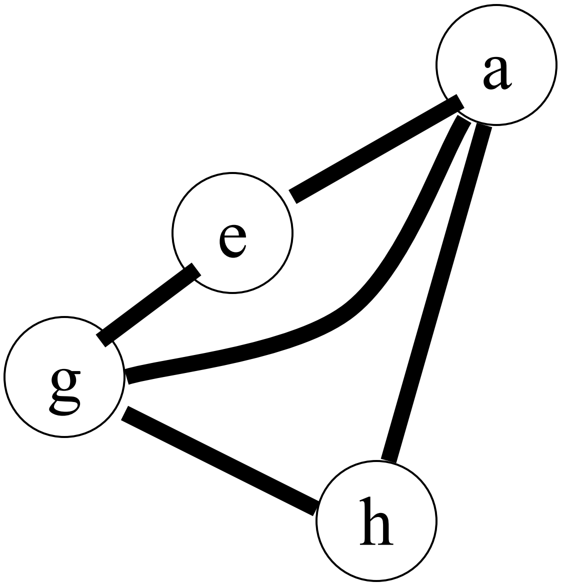
$$\min\{ f_1(a, c, e) + f_2(a, b, f) + f_3(d, e, g) +$$
$$+ f_4(g, a, b) \}$$



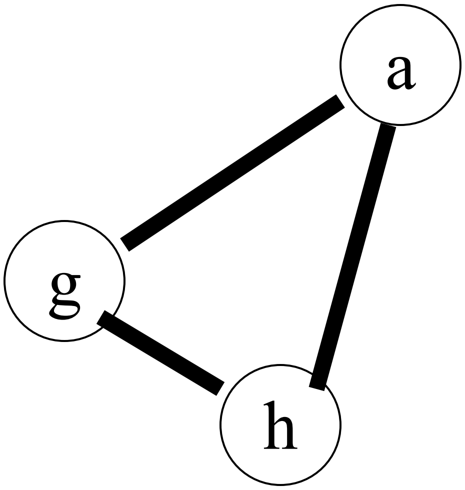
$w = \{b, f, c, d, e, g, h, a\}$



the interaction graph
after the elimination
of b and f.



the interaction graph after the elimination of c and d.



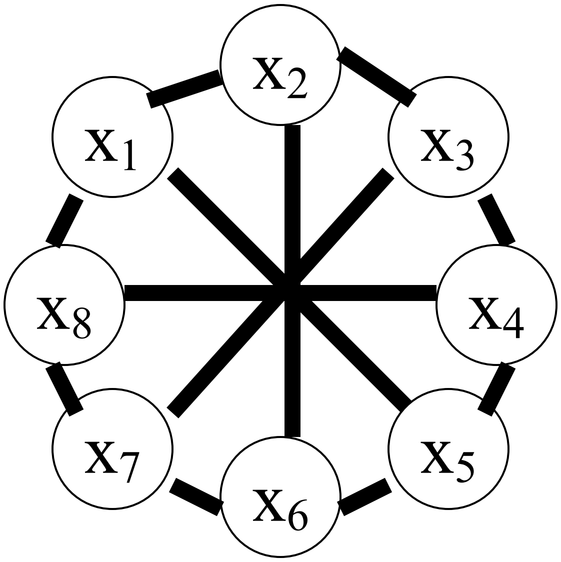
the interaction graph after the elimination of e.

D(w) = 2.

The examples indicate the in many cases of practical significance the dimension of the ordering obtained by using the minimal degree algorithm is bounded by a constant independent of the number of matrices in the program.

Moreover, the dimension of the ordering seems to be usually less than the maximum number of operations in which a matrix is involved.
But there are cases in which “the maximal degree [of the interaction graph] is not an upper bound to the dimension,” as it is shown in the following example.

Example:



$$\begin{aligned}\min_{X}\{\, & f_1(x_1, x_8, x_2, x_5) + \\ & + f_2(x_2, x_1, x_3, x_6) + \\ & + f_3(x_3, x_2, x_4, x_7) + \\ & + f_4(x_4, x_3, x_5, x_8) + \\ & + f_5(x_5, x_4, x_6, x_1) + \\ & + f_6(x_6, x_5, x_7, x_2) + \\ & + f_7(x_7, x_6, x_8, x_3) + \\ & + f_8(x_8, x_7, x_1, x_4)\,\}\end{aligned}$$

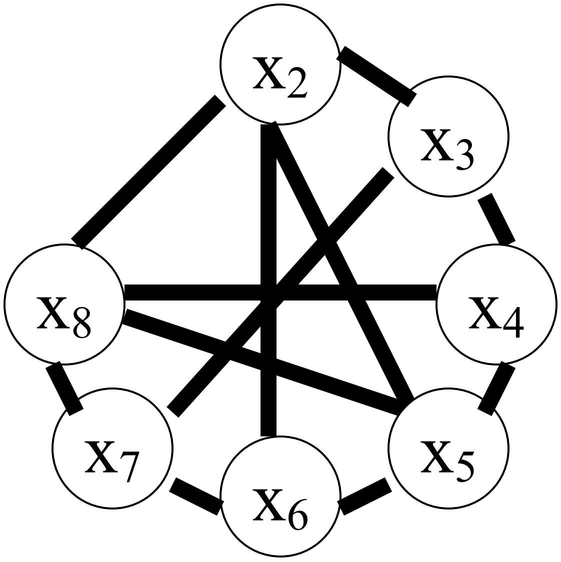
I eliminate x_1 first

x_1^*	h_1	x_8	x_2	x_5
.	.	.	.	.
.	.	.	.	.

The new problem is:

$$\min_{X-\{x1\}} \{ h_1(x_8, x_2, x_5) + f_2(x_2, x_1, x_3, x_6) + f_3 \ldots.\}$$

The corresponding interaction graph is:



Note that the degree of modes x_8, x_2, x_5 has increased from three to four. It can be shown that, whichever order of elimination one chooses, the dimension of the problem presented in this example is four.

For completeness, I would like to point out that I could not find an example in which the dimension of the problem was greater than the maximal degree, when each component involves only three variables.

Open Problems

For which programs in the maximum number of operations in which a matrix is involved an upper bound on the dimension of the corresponding storage problems?
How many such problems are there?
What is the average dimension of the optimal storage problem?

Non-Serial Dynamic Programming

Example:

Five variables A, B, C, D, E
Two values per variable: W_A = (the frame of A) = {a, ~a)
etc.

A problem: $F(v, w, x, y, z) = F_1(v, x, z) + F_2(v, w) + F_3(w, y, z)$

Find the minimum value of F and a configuration
(v, w, x, y, z) that minimizes F.

References: Don Rose

Bertelè, Umberto and Francesco Brioschi. Non-Serial Dynamic Programming. Academic Press, 1972.

Prakash P. Shenoy. "Valuation-Based Systems for Discrete Optimization." In: P.P. Bonissone, M. Henrion, L.N. Kanal and J.F. Lemmer (Eds.). Uncertainty in Artificial Intelligence. Vol. 6, 1991, pp. 385-400.

Figure 2. The valuation network for the optimization problems.

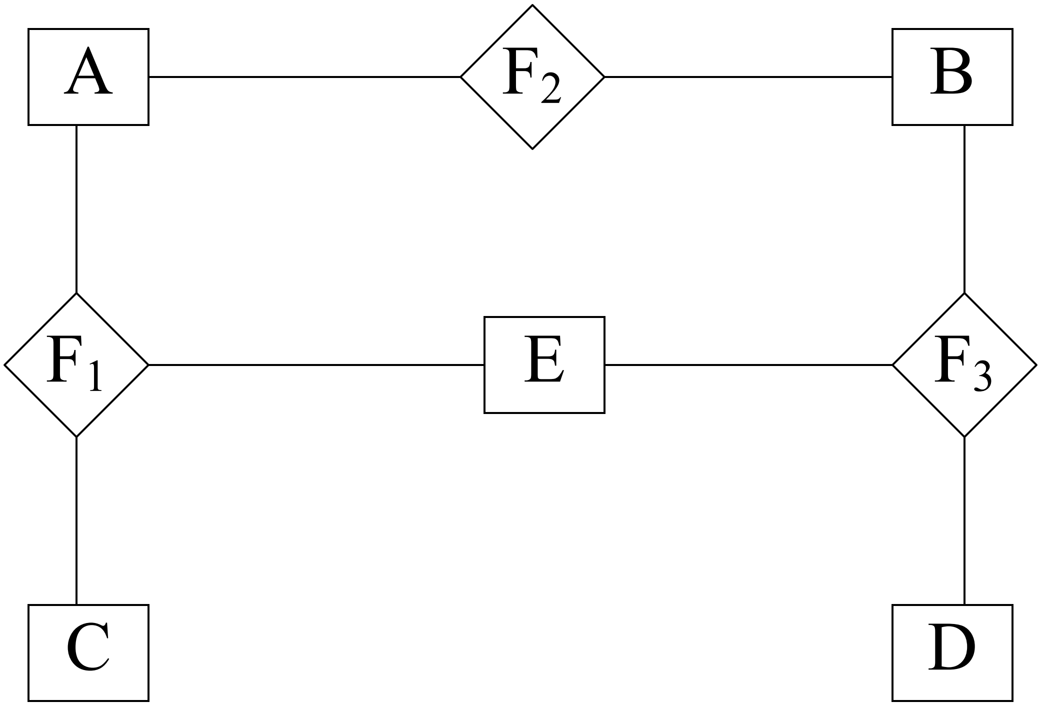


Figure 1. The factors of the objective function, F1, F2, and F3.

$w \in W_{\{A,C,E\}}$	$F_1(w)$
a c e	1
a c ~e	3
a ~c e	5
a ~c ~e	8
~a c e	2
~a c ~e	6
~a ~c e	2
~a ~c ~e	4

$w \in W_{\{A,B\}}$	$F_1(w)$
a b	4
a ~b	8
~a b	0
~a ~b	5

$w \in W_{\{B,D,E\}}$	$F_1(w)$
b d e	0
b d ~e	5
b ~d e	6
b ~d ~e	3
~b d e	5
~b d ~e	1
~b ~d e	4
~b ~d ~e	3

A valuation is a function from configurations to values (usually integers or reals).

Projection of configurations simply means dropping coordinates.
Ex. (~a, ~c, e) is a projection of $\underline{x}$ = (~a, b, ~c, d, e)

config. of   config of
h = (A, C, E)   g = (A, B, C, D, E)
(~a, ~c, e) = $\underline{x}^{\downarrow h}$

Combination Values © Values → Values
Values ⊕ Values → Values

for $\underline{x} \in W_{g \cup h}$ ($\underline{x}$ is a configuration of $g \cup h$)

$$\underbrace{(G \oplus H)(\underline{x})}_{\text{valuation for } g \cup h} = \underbrace{G(\underline{x}^{\downarrow g})}_{\text{valuation for } g} \copyright \underbrace{H(\underline{x}^{\downarrow h})}_{\text{valuation for } h} \text{ =(for NSDP)=}$$

$$=G(\underline{x}^{\downarrow g}) + H(\underline{x}^{\downarrow h})$$

pointwise sum

Ex: F(v, w, x, y, z) = F1 ⊕ F2 ⊕ F3

Marginalization is a mapping $\downarrow h : \{V_g \mid g \supseteq h\} \rightarrow V_h$ s.t. if G is a valuation for g and $g \supseteq h$, then $G^{\downarrow h}$ is a valuation for h. We call $G^{\downarrow h}$ the marginal of G for h.

For NSDP, we define
$G^{\downarrow h}(\underline{x}) = \text{MIN}\,\{G(\underline{x}, \underline{y}) \mid \underline{y} \in W_{g\text{-}h}\}$
for all $\underline{x} \in W_h$.

Note that $F^{\downarrow \varphi}(\blacktriangle) \mid \; W_\varphi = \{\blacktriangle\}$, represents the minimum value for F.

Solution for a valuation. Suppose H is a valuation for h. We call $\underline{x} \in W_h$ a solution for H if $H(\underline{x}) = H^{\downarrow \varphi}(\blacktriangle)$

The Axioms

A1 (Communtativity and Associativity of Combination)

Suppose u, v, w are values. Then

u©w = w©u andu©(v©w)=(u©v) ©w

A2 (Consonence of Marginalization)

Suppose G is a valuation for g, and $k \subseteq h \subseteq g$. Then

$$(G^{\downarrow h})^{\downarrow k} = G^{\downarrow k}.$$

A3 (Distributivity of marginalization over combinations)

Suppose G and H are valuations for g and h, repectively. Then

$$(G \oplus H)^{\downarrow g} = G \oplus (H^{\downarrow g \cap h)}$$

Note that Axiom 3 states that $(G \oplus H)^{\downarrow g}$ can be computed without computing $G \oplus H$!

Figure 3. A rooted Markov tree of the optimization problem.

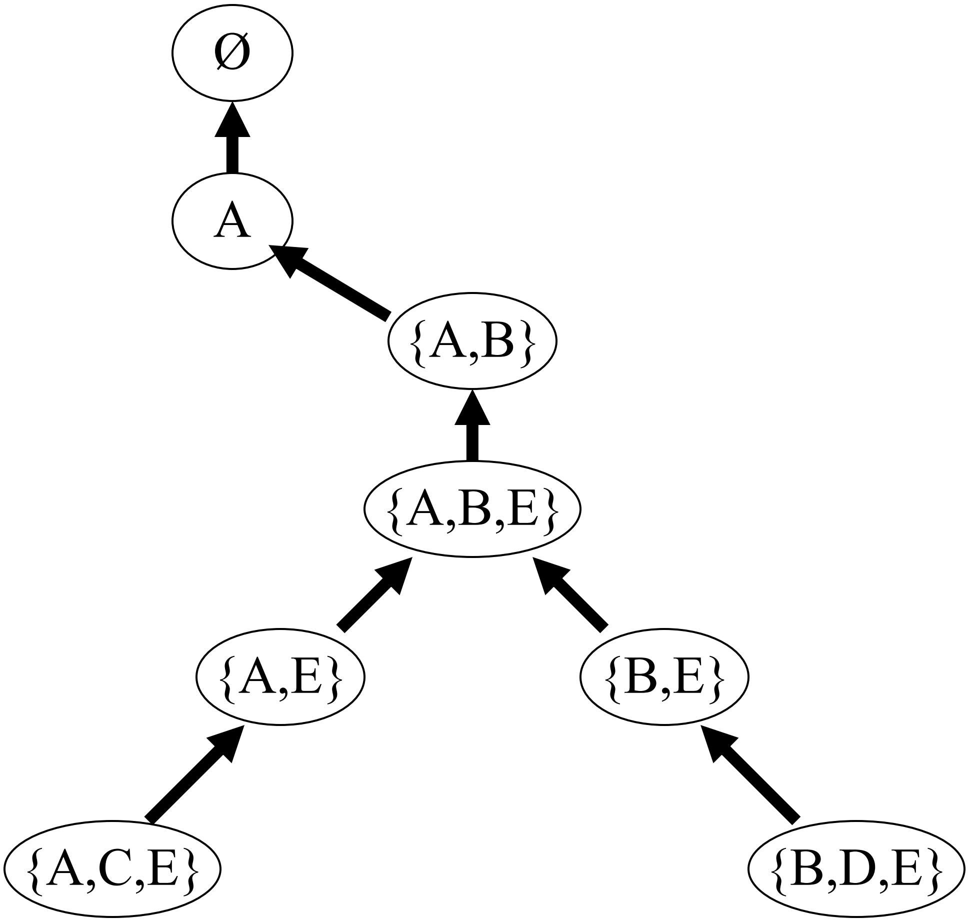
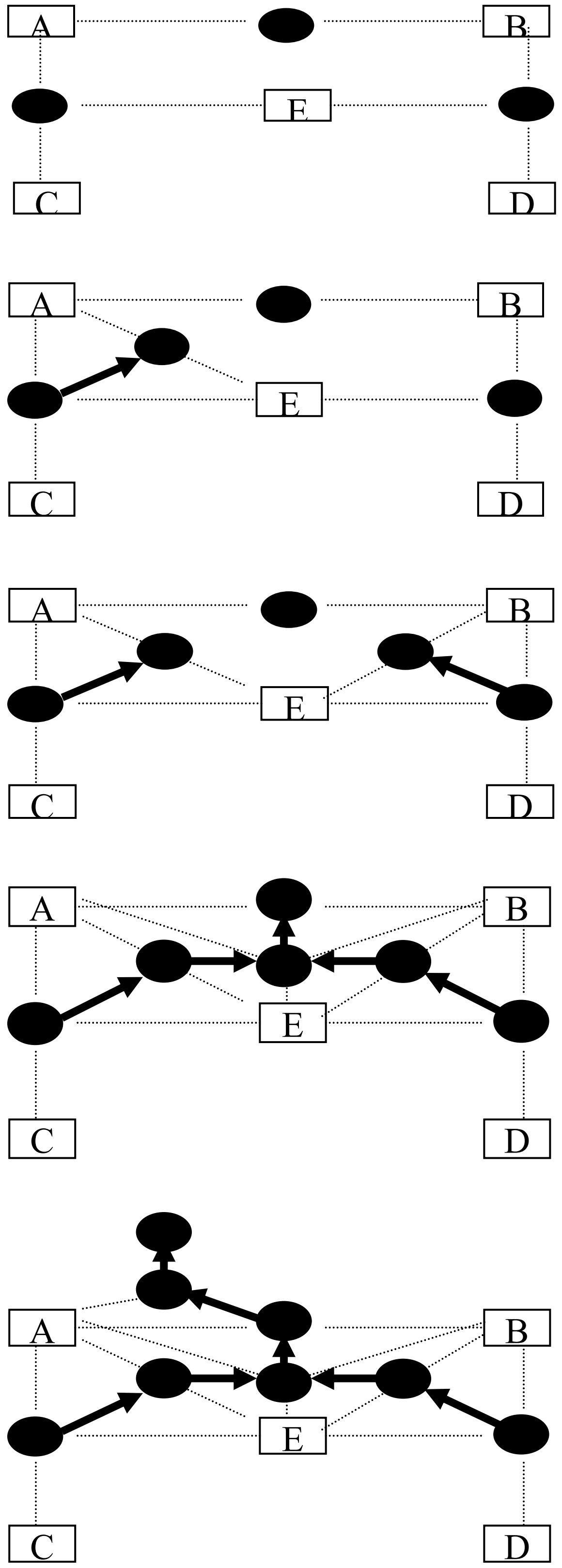


Figure 4. The construction of the rooted Markov tree for the optimization problem.



1. The initial hypergraph. Variables are shown as squares and subsets are shown as black disks. The elements of each subset are indicated by dotted lines.

2. The Markov tree fragment after C is marked. Subset {A, E} is added to the hypergraph. Subset {A, C, E} is now arranged.

3. The Markov tree fragment after D is marked. Subset {B, E} is added to the hypergraph. Subset {B, D, E} is now arranged.

4. The Markov tree fragment after E is marked. Subset {A, B, E} is added to the hypergraph. Subsets {A, E}, {B,E}, and {A, B, E} are now arranged.

5. The Markov tree fragment after B and then A are marked. Subset {A} and Ø are added to the hypergraph. All subsets are now arranged.

Figure 5. The propagation of valuations in the optimization problem. The valuation messages are shown as rectangles overlapping the corresponding edges. The valuations associated with the vertices are shown as diamonds linked to the corresponding vertices by dotted lines.

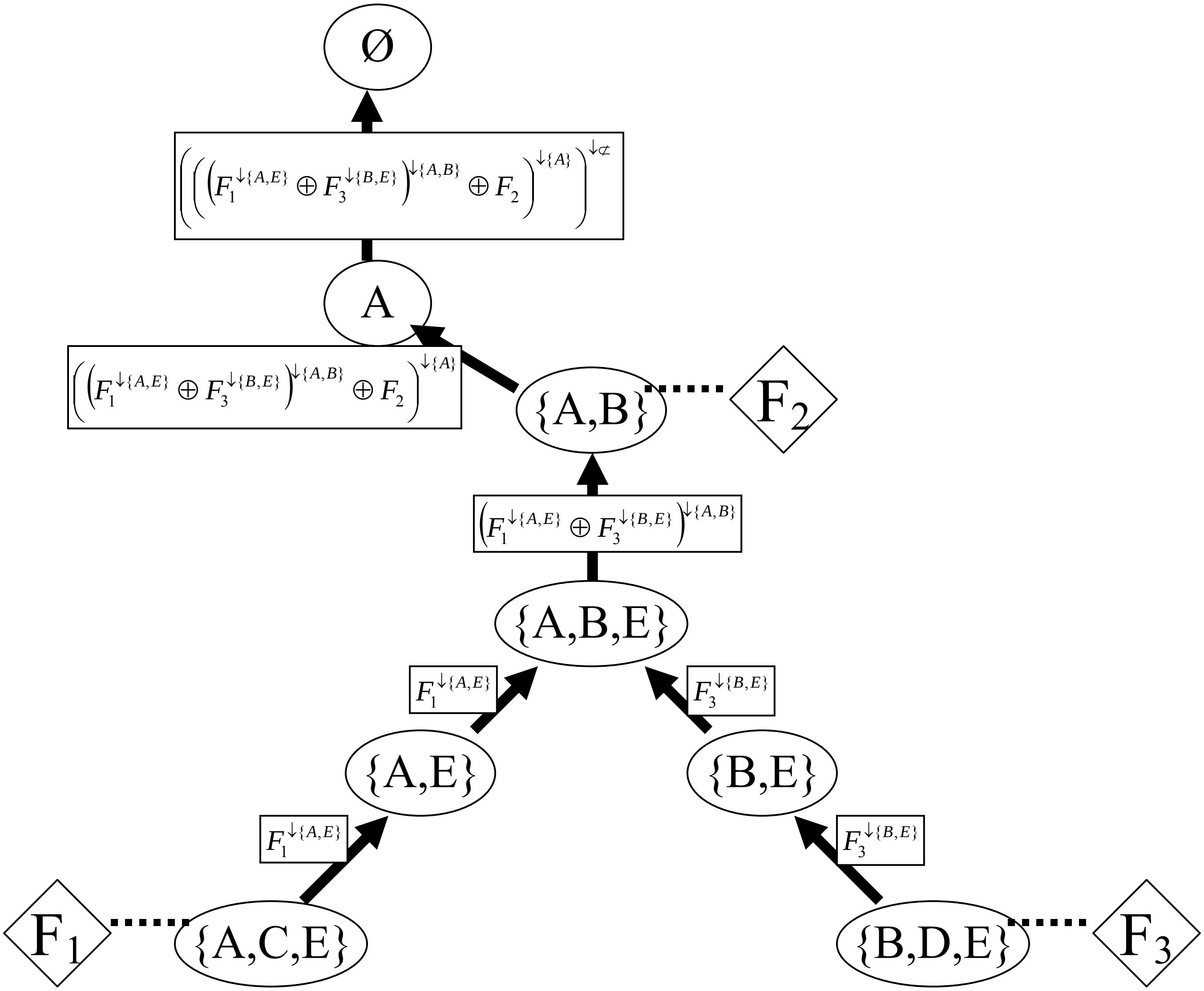

Ø
$\left(\left(\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}} \oplus F_2\right)^{\downarrow\{A\}}\right)^{\downarrow\not\subset}$
A
$\left(\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}} \oplus F_2\right)^{\downarrow\{A\}}$
{A,B}
F_2
$\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}}$
{A,B,E}
$F_1^{\downarrow\{A,E\}}$
$F_3^{\downarrow\{B,E\}}$
{A,E}
{B,E}
$F_1^{\downarrow\{A,E\}}$
$F_3^{\downarrow\{B,E\}}$
F_1
{A,C,E}
{B,D,E}
F_3

Figure 6. The details of the valuation messages for the optimization problem.

$W_{\{A,C,E\}}$			F₁
A	c	e	1
A	c	~e	3
A	~c	e	5
A	~c	~e	8
~a	c	e	2
~a	c	~e	6
~a	~c	e	2
~a	~c	~e	4

$W_{\{A,E\}}$		$F_1^{\downarrow\{A,E\}}$	Ψ_D
a	e	1	c
a	~e	3	c
~a	e	2	c|~c
~a	~e	4	c

$W_{\{B,D,E\}}$			F₃
b	d	e	0
b	d	~e	5
b	~d	e	6
b	~d	~e	3
~b	d	e	5
~b	d	~e	1
~b	~d	e	4
~b	~d	~e	3

$W_{\{B,E\}}$		$F_1^{\downarrow\{A,E\}}$	Ψ_D
b	e	0	d
b	~e	3	~d
~b	e	4	~d
~b	~e	1	d

$W_{\{A,B,E\}}$			$F_1^{\downarrow\{A,E\}}$	$F_3^{\downarrow\{B,E\}}$	$F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}$
a	b	E	1	0	1
a	b	~e	3	3	6
a	~b	E	1	4	5
a	~b	~e	3	1	4
~a	b	E	2	0	2
~a	b	~e	4	3	7
~a	~b	E	2	4	6
~a	~b	~e	4	1	5

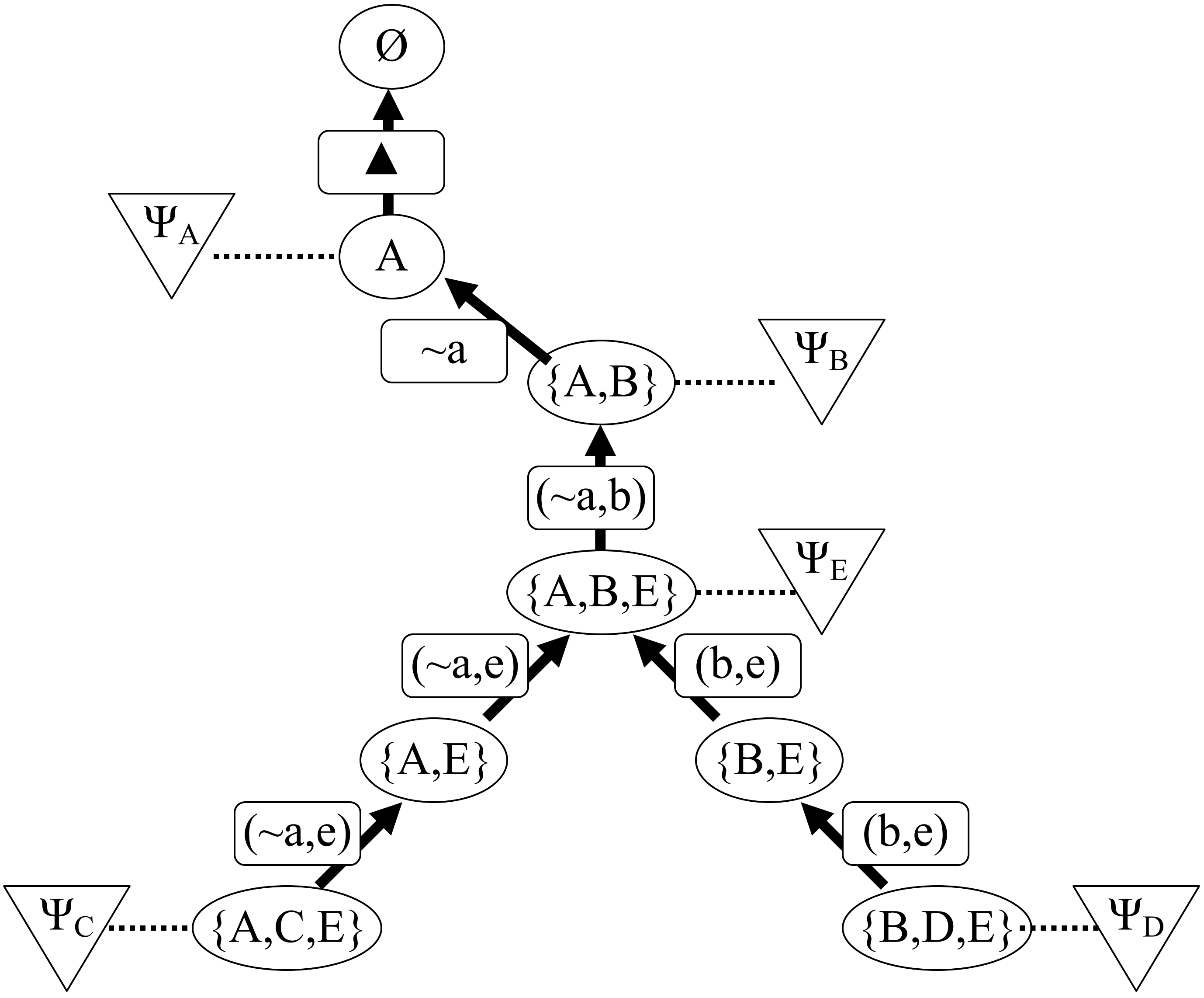
$W_{\{A,B\}}$		$\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}}$	Ψ_E
a	b	1	e
a	~b	4	~e
~a	b	2	e
~a	~b	5	~e

$W_{\{A,B\}}$		$\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}}$	F₂	$\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}} \oplus F_2$
a	b	1	4	5
a	~b	4	8	12
~a	b	2	0	2
~a	~b	5	5	10

$W_{\{A\}}$	$\left(\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}} \oplus F_2\right)^{\downarrow\{A\}}$	Ψ_B
a	5	b
~a	2	b

$W_{\not\subset}$	$\left(\left(\left(F_1^{\downarrow\{A,E\}} \oplus F_3^{\downarrow\{B,E\}}\right)^{\downarrow\{A,B\}} \oplus F_2\right)^{\downarrow\{A\}}\right)^{\downarrow\not\subset}$	Ψ_A
▲	2	~a

Figure 7. The propagation of configuration messages in the optimization problem. The configuration messages are shown as rectangles with rounded corners overlapping the corresponding edges. Note that the direction of messages is opposite to the direction of the edges. The solutions for the five variables are shown as inverted triangles attached to the vertices (where they are stored) by dotted lines.


Ø
▲
Ψ_A
A
~a
{A,B}
Ψ_B
(~a,b)
{A,B,E}
Ψ_E
(~a,e)
(b,e)
{A,E}
{B,E}
(~a,e)
(b,e)
Ψ_C
{A,C,E}
{B,D,E}
Ψ_D