Complexity of Optimal Accumulation of Partial Derivatives on DAGs

Andrew Lyons

University of Chicago / Argonne National Laboratory

MIGHTY XLVII
November 8, 2008

Outline

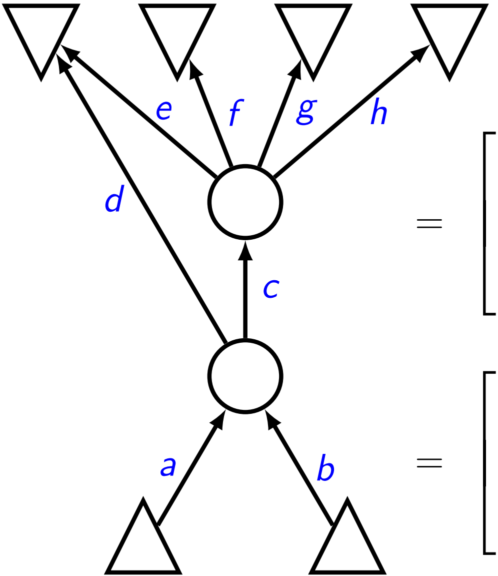
Accumulation of Partial Derivatives
A Generalization: OJA

SOJA with Unit Edges
Minimizing Total Operations
Minimizing Multiplications
Minimizing Additions with Multiplications Fixed

Further Notes on Complexity
Polynomially Solvable Cases of SOJA
Can We Ignore Commutativity?
The Utility of Subtraction

Conclusions

Derivative Accumulation



$$\left[\sum_{\mathcal{P}_x^y \in G} \prod_{e \in \mathcal{P}_x^y} e\right]_{x \in X}^{y \in Y} = \begin{bmatrix} ad + ace & bd + bce \\ acf & bcf \\ acg & bcg \\ ach & bch \end{bmatrix} = \begin{bmatrix} a(d + ce) & b(d + ce) \\ (ac)f & (bc)f \\ (ac)g & (bc)g \\ (ac)h & (bc)h \end{bmatrix}$$

Structural Optimal Jacobian Accumulation (SOJA)

Instance: Dag $G = (V, E)$, where each $e \in E$ is labeled with some c_e such that all c_e are unique real variables that are algebraically independent, positive integer K.

Question: Is there a straight-line program using operations in $\{+, *\}$ of length K or less that computes every entry in J such that every operand is either some c_e or the result of a previous operation?

Some Observations

- The complexity of SOJA is open. (We don't even know whether it's in **NP**, but there's evidence that it's in **BPP**?).

Some Observations

- The complexity of SOJA is open. (We don't even know whether it's in **NP**, but there's evidence that it's in **BPP**?).
- The number of paths can be **exponential** in the size of the graph.

Some Observations

- The complexity of SOJA is open. (We don't even know whether it's in **NP**, but there's evidence that it's in **BPP**?).
- The number of paths can be **exponential** in the size of the graph.
- Can always be done with a polynomial number of operations. (However, we want to generate code that may run millions of times.)

Some Observations

- The complexity of SOJA is open. (We don't even know whether it's in **NP**, but there's evidence that it's in **BPP**?).
- The number of paths can be **exponential** in the size of the graph.
- Can always be done with a polynomial number of operations. (However, we want to generate code that may run millions of times.)
- Can we justify the restriction to operations in $\{+, *\}$? (More on this later.)

a Generalization: OJA

Allow *algebraic dependences* (in particular, two edge labels may be *equivalent*). This deemphasizes the **structure** of G.

Theorem (Naumann, 2008)

Minimizing total ops and minimizing multiplications are **NP***-hard for* OJA *even when* J *is scalar (single input and single output) and all paths in* G *have length* ≤ 3*.*

Proof.

Reduction from ENSEMBLE COMPUTATION... $\square$

Ensemble Computation

Instance: Finite set S, collection $C = \{C_1, \ldots, C_r\}$ of subsets of S, positive integer K.

Question: Can the elements of C be built up from the elements of S using K or fewer union operations?

Ensemble Computation

Instance: Finite set S, collection $C = \{C_1, \ldots, C_r\}$ of subsets of S, positive integer K.

Question: Can the elements of C be built up from the elements of S using K or fewer union operations?

Example: $S = \{a, b, c, d\}, C = \{\{a, b\}, \{b, c, d\}, \{a, c, d\}\}, K = 4.$

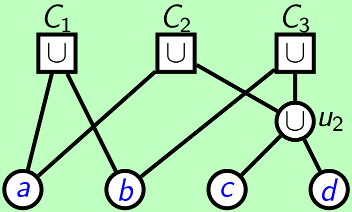
Ensemble Computation

Instance: Finite set S, collection $C = \{C_1, \ldots, C_r\}$ of subsets of S, positive integer K.

Question: Can the elements of C be built up from the elements of S using K or fewer union operations?

Example: $S = \{a, b, c, d\}$, $C = \{\{a, b\}, \{b, c, d\}, \{a, c, d\}\}$, $K = 4$.

$$
\begin{array}{lcll}
u_1 & = & \{a\} \cup \{b\} & = C_1 \\
u_2 & = & \{c\} \cup \{d\} & \\
u_3 & = & \{b\} \cup u_2 & = C_2 \\
u_4 & = & \{a\} \cup u_2 & = C_3
\end{array}
$$



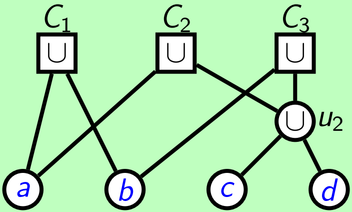
Ensemble Computation

Instance: Finite set S, collection $C = \{C_1, \ldots, C_r\}$ of subsets of S, positive integer K.

Question: Can the elements of C be built up from the elements of S using K or fewer union operations?

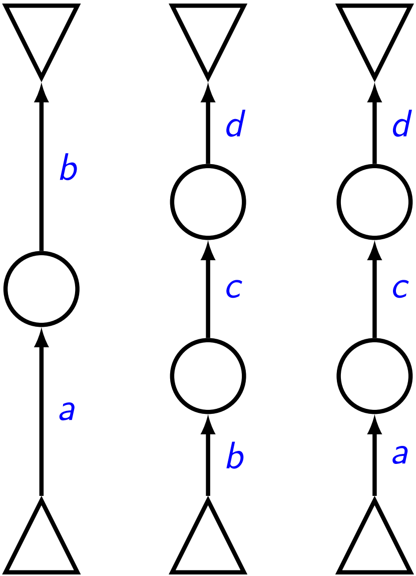
Example: $S = \{a, b, c, d\}, C = \{\{a, b\}, \{b, c, d\}, \{a, c, d\}\}, K = 4.$

$$\begin{aligned}
u_1 &= \{a\} \cup \{b\} &&= C_1 \\
u_2 &= \{c\} \cup \{d\} && \\
u_3 &= \{b\} \cup u_2 &&= C_2 \\
u_4 &= \{a\} \cup u_2 &&= C_3
\end{aligned}$$

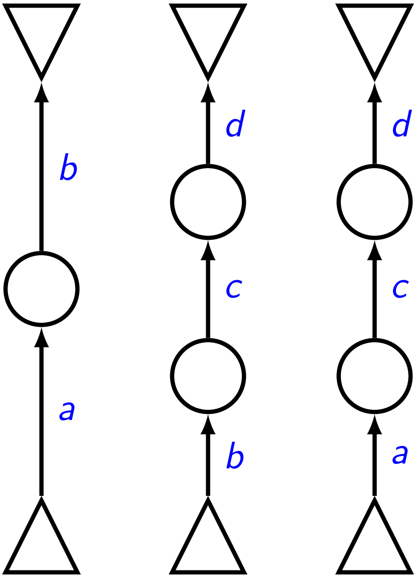


Answer: YES

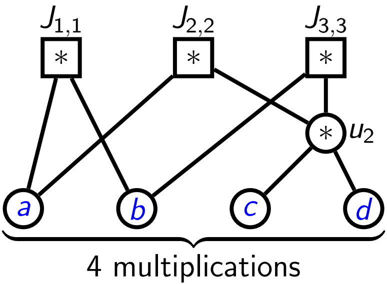
Reducing Ensemble Computation to OJA



Reducing Ensemble Computation to OJA



$$J = \begin{bmatrix} J_{1,1} & 0 & 0 \\ 0 & J_{2,2} & 0 \\ 0 & 0 & J_{3,3} \end{bmatrix}$$



Outline

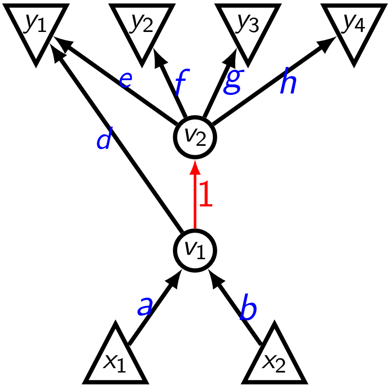
Accumulation of Partial Derivatives
A Generalization: OJA

SOJA with Unit Edges
Minimizing Total Operations
Minimizing Multiplications
Minimizing Additions with Multiplications Fixed

Further Notes on Complexity
Polynomially Solvable Cases of SOJA
Can We Ignore Commutativity?
The Utility of Subtraction

Conclusions

SOJA with Unit Edges



$$J_{j,i} \equiv \frac{\partial y_j}{\partial x_i} = \sum_{\mathcal{P}_{x_i}^{y_j} \in G} \prod_{e \in \mathcal{P}_{x_i}^{y_j}} e$$

$$J = \begin{bmatrix} ad + ae & bd + be \\ af & bf \\ ag & bg \\ ah & bh \end{bmatrix}$$

$$= \begin{bmatrix} a(d+e) & b(d+e) \\ af & bf \\ ag & bg \\ ah & bh \end{bmatrix}$$

- Recognize that some edges are unit labeled (multiplicative identity).

Minimizing *Total Operations* for SOJA with Unit Edges

Theorem (Lyons, in preparation)

Minimizing total operations for SOJA with Unit Edges *is* **NP***-hard under each of the following restrictions.*

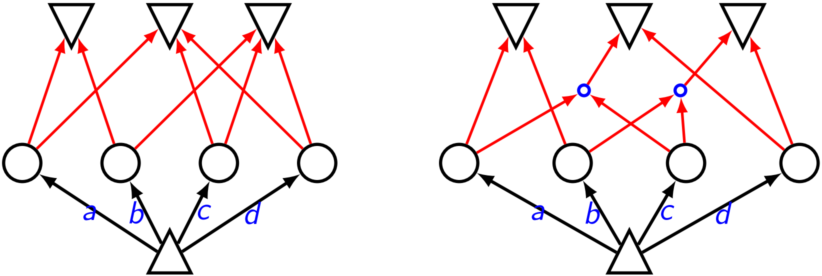
(i) *G has one input or one output and all paths in G have length ≤ 2.*

(ii) *G has one input or one output, all vertices in G have indegree ≤ 2, and all paths in G have length ≤ 3.*

Proof.

Reduction from Ensemble Computation... $\square$

Minimizing *Total Operations* for SOJA with Unit Edges



$$J = \begin{bmatrix} a+b \\ a+c+d \\ b+c+d \end{bmatrix}$$

Minimizing *Multiplications* for SOJA WITH UNIT EDGES

Theorem (Lyons, in preparation)

Minimizing multiplications is **NP***-hard for* SOJA WITH UNIT EDGES *under the following restrictions:*

- *J is scalar (one input and one output) and all paths in G have length ≤ 3.*

Minimizing *Multiplications* for SOJA WITH UNIT EDGES

Theorem (Lyons, in preparation)

Minimizing multiplications is **NP***-hard for* SOJA WITH UNIT EDGES *under the following restrictions:*

- *J is scalar (one input and one output) and all paths in G have length ≤ 3.*
- *J is scalar and all vertices in G have indegree ≤ 2.*

Minimizing *Multiplications* for SOJA WITH UNIT EDGES

Theorem (Lyons, in preparation)

Minimizing multiplications is **NP***-hard for* SOJA WITH UNIT EDGES *under the following restrictions:*

- *J is scalar (one input and one output) and all paths in G have length ≤ 3.*
- *J is scalar and all vertices in G have indegree ≤ 2.*

Proof.

Reduction from BICLIQUE PARTITION... □

Biclique Partition

Instance: Bipartite graph $G = (A, B, E)$, positive integer K.

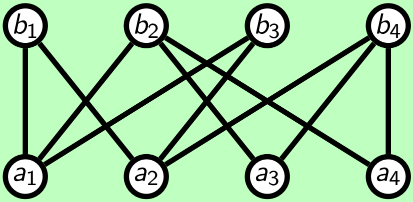
Question: Can the edge set $E(G)$ be partitioned into K or fewer complete bipartite graphs?

Biclique Partition

Instance: Bipartite graph $G = (A, B, E)$, positive integer K.

Question: Can the edge set $E(G)$ be partitioned into K or fewer complete bipartite graphs?

Example

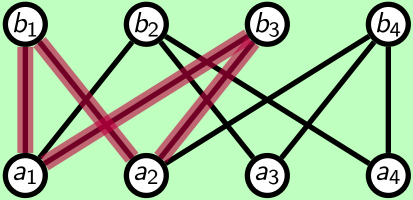


Biclique Partition

Instance: Bipartite graph $G = (A, B, E)$, positive integer K.

Question: Can the edge set $E(G)$ be partitioned into K or fewer complete bipartite graphs?

Example



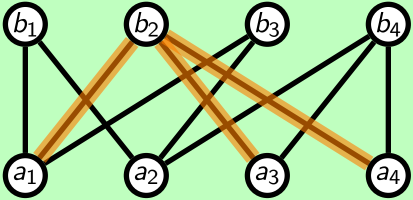
$\{\{a_1, a_2\}, \{b_1, b_3\}\}$

Biclique Partition

Instance: Bipartite graph $G = (A, B, E)$, positive integer K.

Question: Can the edge set $E(G)$ be partitioned into K or fewer complete bipartite graphs?

Example



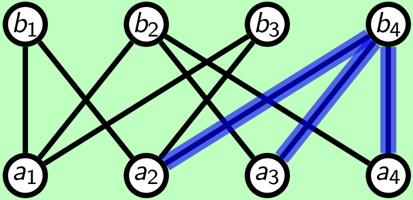
$\{\{a_1, a_3, a_4\}, \{b_2\}\}$

Biclique Partition

Instance: Bipartite graph $G = (A, B, E)$, positive integer K.

Question: Can the edge set $E(G)$ be partitioned into K or fewer complete bipartite graphs?

Example



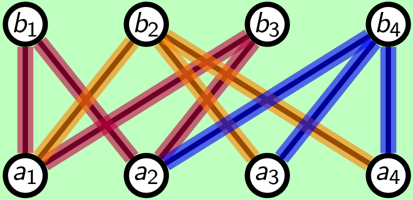
$\{\{a_2, a_3, a_4\}, \{b_4\}\}$

Biclique Partition

Instance: Bipartite graph $G = (A, B, E)$, positive integer K.

Question: Can the edge set $E(G)$ be partitioned into K or fewer complete bipartite graphs?

Example

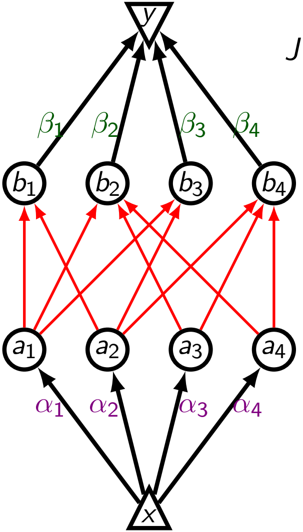


$\{\{a_1, a_2\}, \{b_1, b_3\}\}$

$\{\{a_1, a_3, a_4\}, \{b_2\}\}$

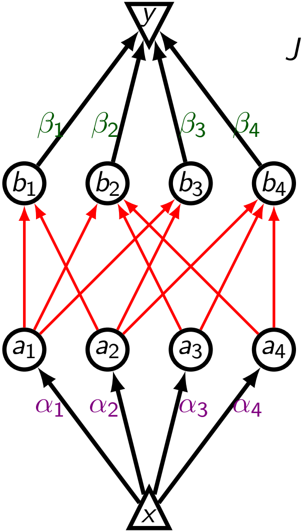
$\{\{a_2, a_3, a_4\}, \{b_4\}\}$

Example: scalar, all paths length ≤ 3



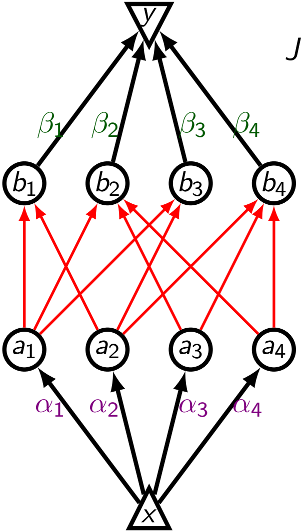
$$J = \sum_{\mathcal{P}_x^y \in G} \prod_{e \in \mathcal{P}_x^y} e$$

Example: scalar, all paths length ≤ 3



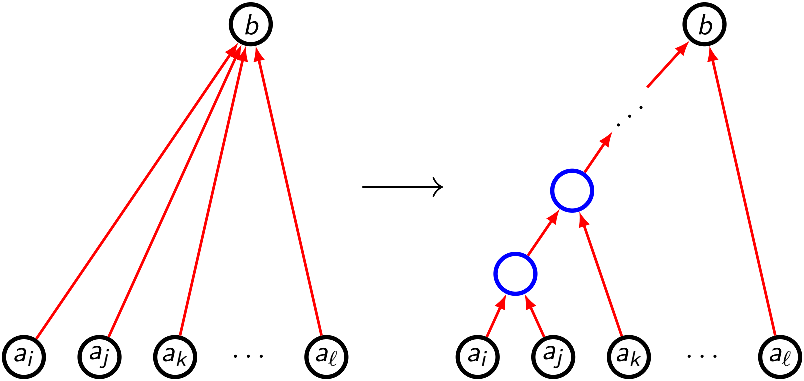
$$
\begin{aligned}
J &= \sum_{\mathcal{P}_x^y \in G} \prod_{e \in \mathcal{P}_x^y} e \\
&= \underbrace{\begin{array}{l} \alpha_1\beta_1 + \alpha_1\beta_2 + \alpha_1\beta_3 + \alpha_2\beta_1 + \alpha_2\beta_3 \\ \quad + \alpha_2\beta_4 + \alpha_3\beta_2 + \alpha_3\beta_4 + \alpha_4\beta_2 + \alpha_4\beta_4 \end{array}}_{\text{10 multiplications}}
\end{aligned}
$$

Example: scalar, all paths length ≤ 3



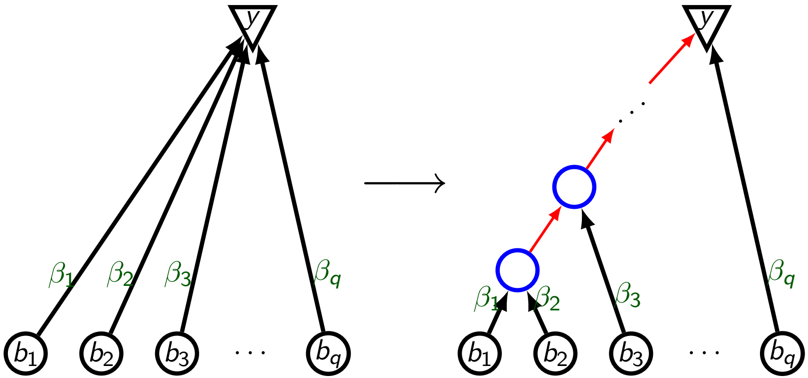
$$
\begin{aligned}
J &= \sum_{\mathcal{P}_x^y \in G} \prod_{e \in \mathcal{P}_x^y} e \\
&= \underbrace{\begin{matrix}\alpha_1\beta_1 + \alpha_1\beta_2 + \alpha_1\beta_3 + \alpha_2\beta_1 + \alpha_2\beta_3 \\ + \alpha_2\beta_4 + \alpha_3\beta_2 + \alpha_3\beta_4 + \alpha_4\beta_2 + \alpha_4\beta_4\end{matrix}}_{\text{10 multiplications}} \\
&= \underbrace{\begin{matrix}(\alpha_1 + \alpha_2)(\beta_1 + \beta_3) \\ + (\alpha_1 + \alpha_3 + \alpha_4)\beta_2 \\ + (\alpha_2 + \alpha_3 + \alpha_4)\beta_4\end{matrix}}_{\text{3 multiplications}}
\end{aligned}
$$

Scalar, indegree ≤ 2



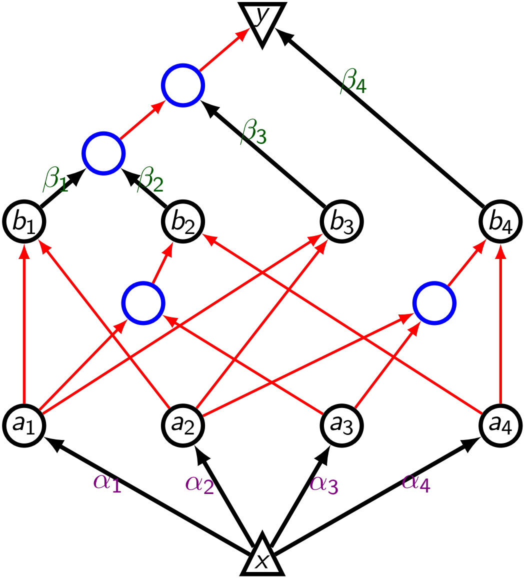
Creating $O(|B||A|)$ new vertices.

Scalar, indegree ≤ 2



Creating $|B| - 1$ new vertices.

Scalar, indegree $\leq$ 2



Minimizing Additions with Multiplications Fixed

Optimal accumulation for example:

$$(\alpha_1 + \alpha_2)(\beta_1 + \beta_3) + (\alpha_1 + \alpha_3 + \alpha_4)\beta_2 + (\alpha_2 + \alpha_3 + \alpha_4)\beta_4$$

Results in two instances of Ensemble Computation:

- $S = \{\alpha_1, \ldots, \alpha_{|A|}\}$,
  $C = \{\{\alpha_1, \alpha_2\}, \{\alpha_1, \alpha_3, \alpha_4\}, \{\alpha_2, \alpha_3, \alpha_4\}\}$
- $S = \{\beta_1, \ldots, \beta_{|B|}\}$,
  $C = \{\{\beta_1, \beta_3\}, \{\beta_2\}, \{\beta_4\}\}$

Outline

Accumulation of Partial Derivatives

A Generalization: OJA

SOJA with Unit Edges

Minimizing Total Operations

Minimizing Multiplications

Minimizing Additions with Multiplications Fixed

Further Notes on Complexity

Polynomially Solvable Cases of SOJA

Can We Ignore Commutativity?

The Utility of Subtraction

Conclusions

Polynomially Solvable Cases of SOJA

- Trees (or tree-dags, or whatever)

Polynomially Solvable Cases of SOJA

- Trees (or tree-dags, or whatever)
- Two-Terminal Series-Parallel DAGs (Lyons, in preparation)

Can We Ignore Commutativity?

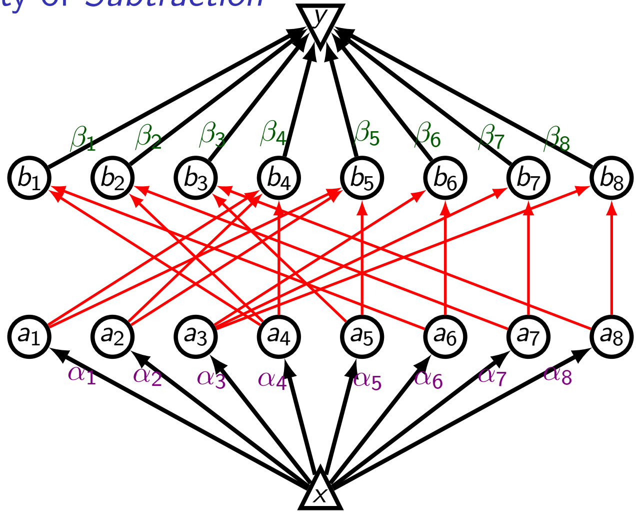
Bilinear Forms (Gonzalez and JáJá, 1980)

$$
\begin{aligned}
J &= \alpha_1\beta_4 + \alpha_1\beta_5 + \alpha_2\beta_4 + \alpha_2\beta_5 + \alpha_3\beta_6 + \alpha_3\beta_7 + \alpha_3\beta_8 \\
&\quad + \alpha_4\beta_1 + \alpha_4\beta_2 + \alpha_4\beta_4 + \alpha_5\beta_3 + \alpha_5\beta_5 + \alpha_6\beta_1 + \alpha_6\beta_6 \\
&\quad + \alpha_7\beta_2 + \alpha_7\beta_7 + \alpha_8\beta_3 + \alpha_8\beta_8 \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 \end{pmatrix}
\begin{pmatrix}
\square & \square & \square & \blacksquare & \blacksquare & \square & \square & \square \\
\square & \square & \square & \blacksquare & \blacksquare & \square & \square & \square \\
\square & \square & \square & \square & \square & \blacksquare & \blacksquare & \blacksquare \\
\blacksquare & \blacksquare & \square & \blacksquare & \square & \square & \square & \square \\
\square & \square & \blacksquare & \square & \blacksquare & \square & \square & \square \\
\blacksquare & \square & \square & \square & \square & \blacksquare & \square & \square \\
\square & \blacksquare & \square & \square & \square & \square & \blacksquare & \square \\
\square & \square & \blacksquare & \square & \square & \square & \square & \blacksquare
\end{pmatrix}
\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \end{pmatrix}
\end{aligned}
$$

Bilinear Forms (Gonzalez and JáJá, 1980)

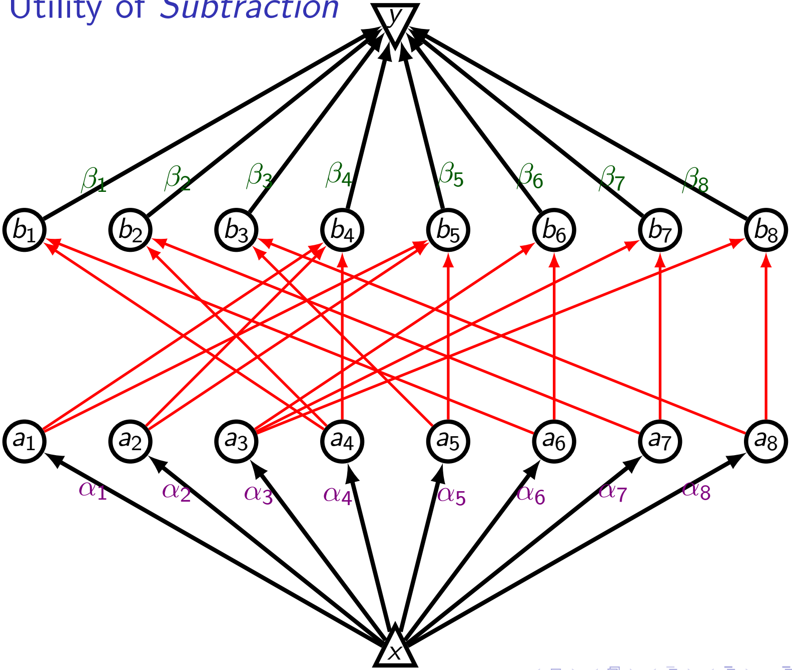
$$
\begin{aligned}
J &= \alpha_1\beta_4 + \alpha_1\beta_5 + \alpha_2\beta_4 + \alpha_2\beta_5 + \alpha_3\beta_6 + \alpha_3\beta_7 + \alpha_3\beta_8 \\
&\quad + \alpha_4\beta_1 + \alpha_4\beta_2 + \alpha_4\beta_4 + \alpha_5\beta_3 + \alpha_5\beta_5 + \alpha_6\beta_1 + \alpha_6\beta_6 \\
&\quad + \alpha_7\beta_2 + \alpha_7\beta_7 + \alpha_8\beta_3 + \alpha_8\beta_8 \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 \end{pmatrix}
\begin{pmatrix}
\square & \square & \square & \blacksquare & \blacksquare & \square & \square & \square \\
\square & \square & \square & \blacksquare & \blacksquare & \square & \square & \square \\
\square & \square & \square & \square & \square & \blacksquare & \blacksquare & \blacksquare \\
\blacksquare & \blacksquare & \square & \blacksquare & \square & \square & \square & \square \\
\square & \square & \blacksquare & \square & \blacksquare & \square & \square & \square \\
\blacksquare & \square & \square & \square & \square & \blacksquare & \square & \square \\
\square & \blacksquare & \square & \square & \square & \square & \blacksquare & \square \\
\square & \square & \blacksquare & \square & \square & \square & \square & \blacksquare
\end{pmatrix}
\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \end{pmatrix} \\
&= (\alpha_1 + \alpha_2 + \alpha_3)(\beta_1 + \beta_2 + \beta_4) \\
&\quad + (\alpha_3 + \alpha_6)(\beta_1 + \beta_6) + (\alpha_1 + \alpha_2 + \alpha_5)(\beta_3 + \beta_5) \\
&\quad + (\alpha_3 + \alpha_7)(\beta_2 + \beta_7) + (\alpha_3 + \alpha_8)(\beta_3 + \beta_8) \\
&\quad \boxed{- (\alpha_1 + \alpha_2 + \alpha_3)(\beta_1 + \beta_2 + \beta_3)}
\end{aligned}
$$

The Utility of *Subtraction*

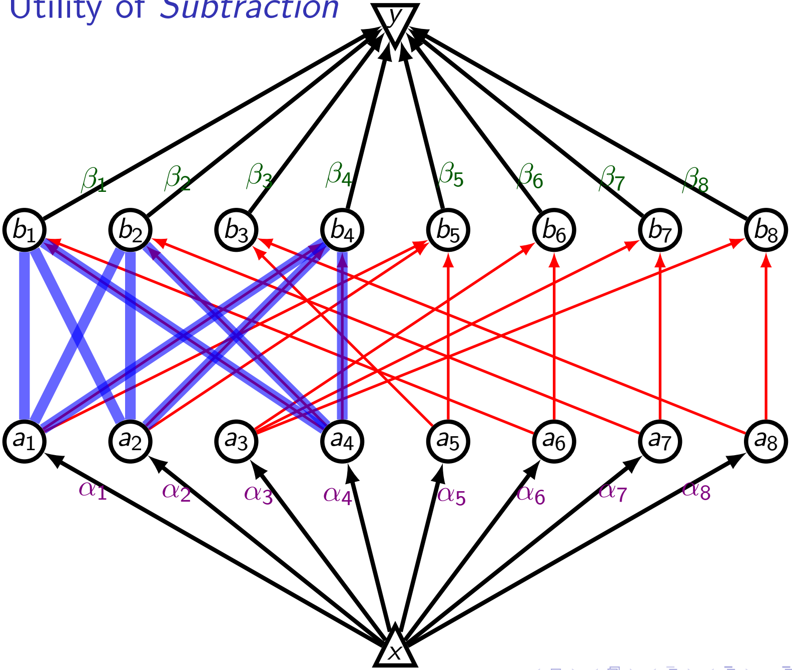


$$
\begin{aligned}
J \quad = \quad & \alpha_1\beta_4 + \alpha_1\beta_5 + \alpha_2\beta_4 + \alpha_2\beta_5 + \alpha_3\beta_6 + \alpha_3\beta_7 + \alpha_3\beta_8 \\
& + \alpha_4\beta_1 + \alpha_4\beta_2 + \alpha_4\beta_4 + \alpha_5\beta_3 + \alpha_5\beta_5 + \alpha_6\beta_1 + \alpha_6\beta_6 \\
& + \alpha_7\beta_2 + \alpha_7\beta_7 + \alpha_8\beta_3 + \alpha_8\beta_8
\end{aligned}
$$

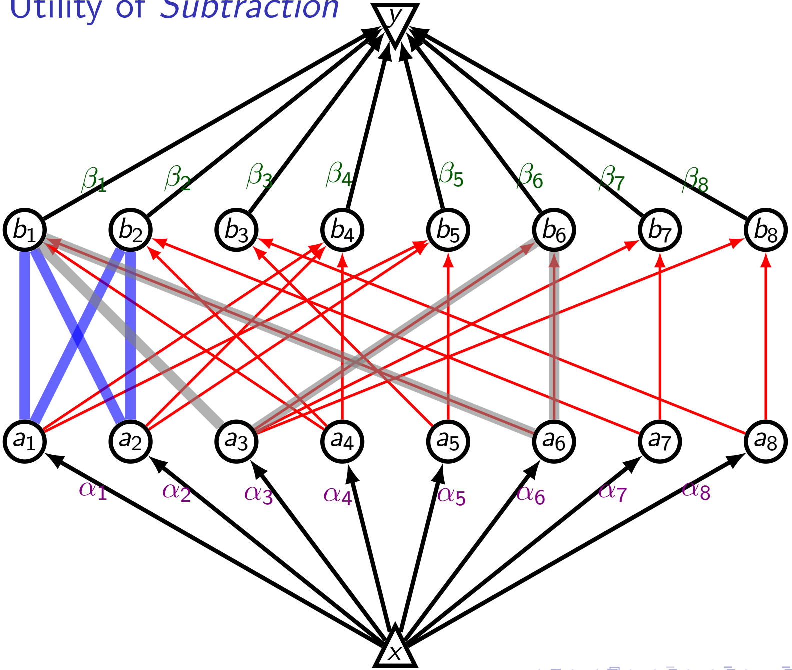
The Utility of *Subtraction*



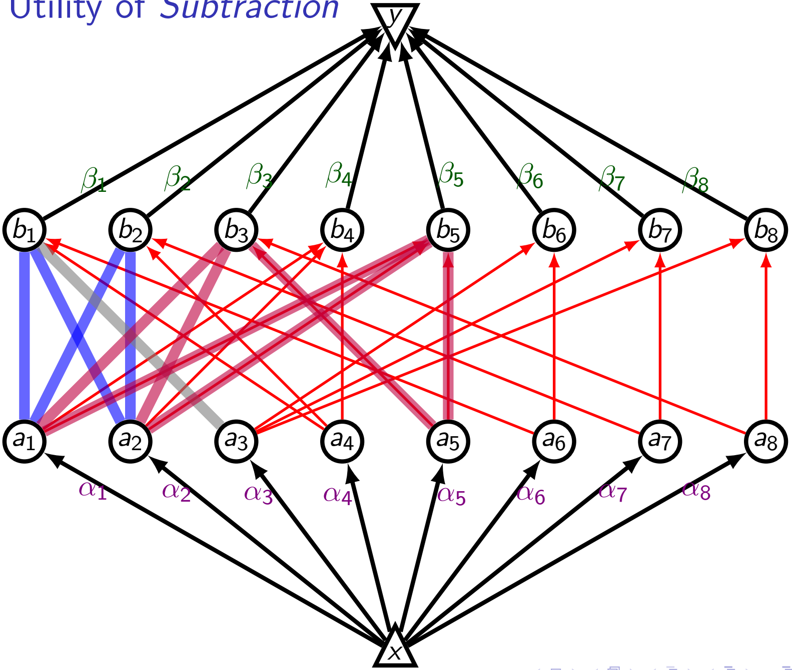
The Utility of *Subtraction*



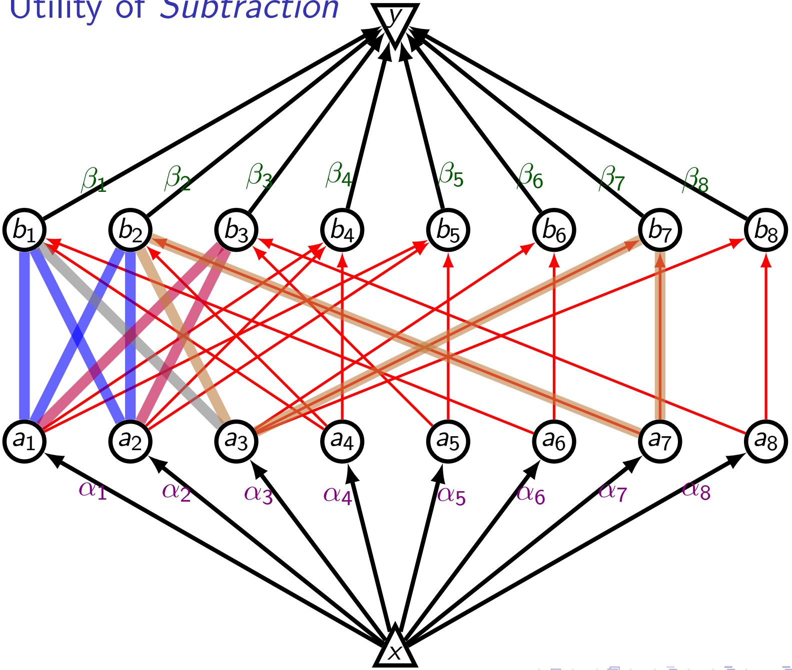
The Utility of *Subtraction*



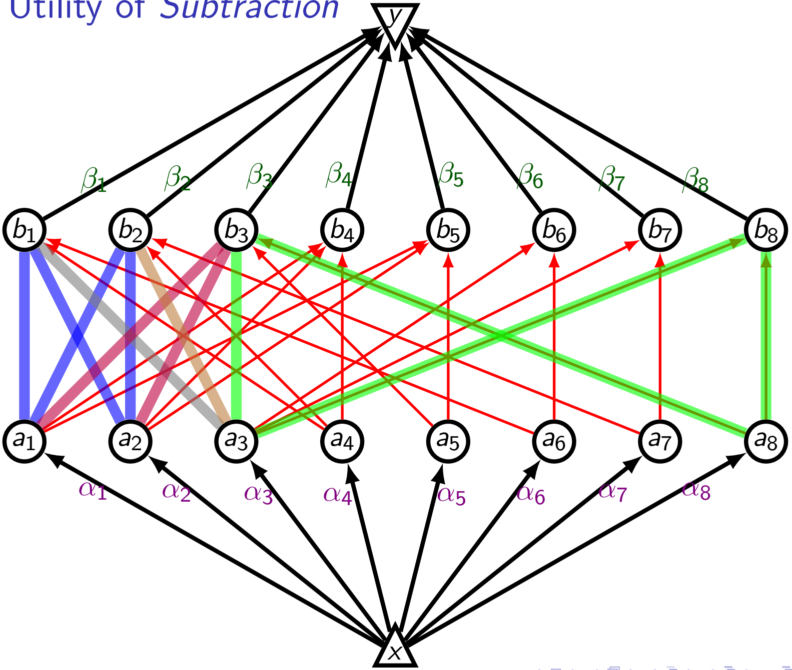
The Utility of *Subtraction*



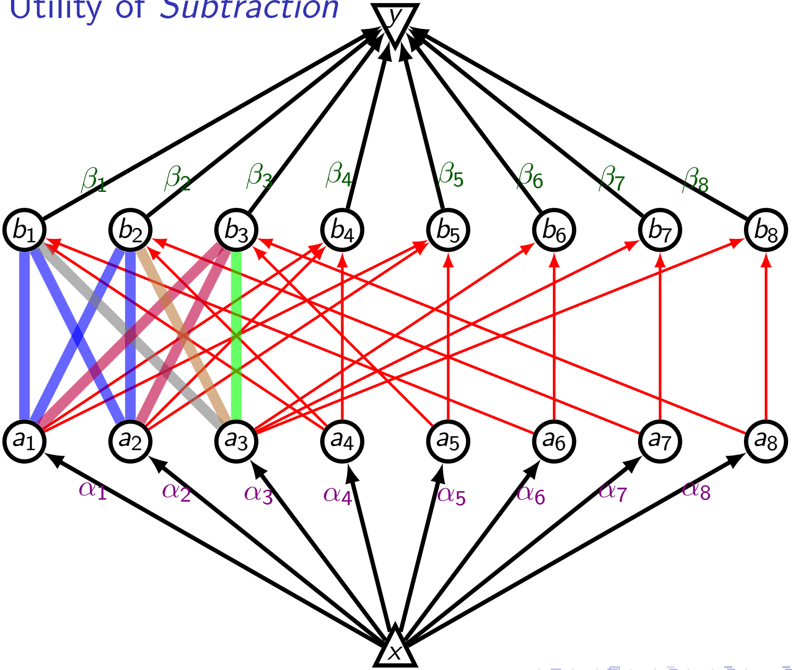
The Utility of *Subtraction*



The Utility of *Subtraction*



The Utility of *Subtraction*



Outline

Accumulation of Partial Derivatives

A Generalization: OJA

SOJA with Unit Edges

Minimizing Total Operations

Minimizing Multiplications

Minimizing Additions with Multiplications Fixed

Further Notes on Complexity

Polynomially Solvable Cases of SOJA

Can We Ignore Commutativity?

The Utility of Subtraction

Conclusions

Conclusions

- Complexity of SOJA (original problem definition) is *still open*(!)
- Various Generalizations (algebraic dependencies, unit edges) yield **NP**-hardness.
- We have polynomial time solutions for some **very restricted** cases.
- There is some evidence that we can ignore commutativity (which greatly reduces search space)
- *Subtraction* is useful.

Future Work:

- Are these problems in **NP**?
- Parameterized complexity? Approximation?
- Could subtraction help in SOJA?