Modeling and Optimization with OPL 4 Optimization of Graph Problems

Andreas Popp



These slides are licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

4 Optimization of Graph Problems

CC-BY-SA A. Popp

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

I.4 OPL: conditional operators

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

4.4 OPL: conditional operators

- 4.1 Short introduction into graph theory
- 4.2 Representation of graphs in OPL
- 4.3 OPL: custom tuples as data
- 4.4 OPL:

4.1 Short introduction into graph theory

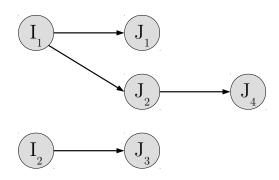
Example: Lewig Adelburg



4 Optimization of Graph Problems

- 4.1 Short introduction into graph theory
- of graphs in OPL
- I.3 OPL: custom uples as data tructure
- 4.4 OPL: conditional operators

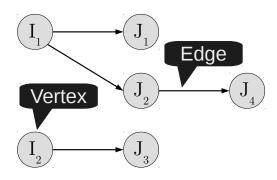
Concept of a graph: components



4 Optimization of Graph Problems

- 4.1 Short introduction into graph theory
- 4.2 Representation of graphs in OPL
- 4.3 OPL: custom tuples as data structure
- 4.4 OPL:

Concept of a graph: components



4 Optimization of Graph Problems

- 4.1 Short introduction into graph theory
- 4.2 Representation of graphs in OPL
- 4.3 OPL: custom tuples as data
- 4.4 OPL:

▶ **Directed** Graphs are defined as a tuple G = (V, E) with a set of vertices V and a set of edges $E \subseteq V \times V$. in example:

$$G = (\{I_1, I_2, J_1, J_2, J_3, J_4\}, \{(I_1, J_1), (I_1, J_2), (I_2, J_3), (J_2, J_4)\})$$

- Undirected graphs are graphs whose edges do not have a specific direction.
- ▶ Weighted graphs are defined as a tuple G = (V, E, g) with a set of vertices V, a set of edges $E \subseteq V \times V$ and a weight function $g : E \to \mathbb{R}$.

- 4.1 Short introduction into graph theory
- 4.2 Representation of graphs in OPL
- 4.3 OPL: custom tuples as data structure
- 4.4 OPL: conditiona operators

4.2 Representation of graphs in OPL

4 Optimization of Graph Problems

CC-BY-SA A. Popp

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

1.4 OPL: conditional

4.2 Representation of graphs in OPL

Index sets:

set of products

set of ressources

Parameters:

price of product $i \in I$ D;

capacity of ressource $r \in R$

capacity consumption of product $i \in I$ on ressource $r \in R$

set of edges in the sequence graph

Decision variables:

production quantity of product $i \in I$

Modellbeschreibung:

$$\max \sum_{i \in I} p_i \cdot x_i$$

$$s.t. \sum_{i \in I} v_{ri} \cdot x_i \le c_i \qquad \forall r \in R \quad (I)$$

$$x_i \ge \sum_{(i,j) \in E} x_j \qquad \forall i \in I \quad (II)$$

$$x_i \ge 0 \qquad \forall i \in I$$

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

I.4 OPL: conditional

Application example for graphs

$$x_i \ge \sum_{(i,j) \in E} x_j \quad \forall i \in I$$

Question: How can the graph be represented in the optimization model?

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

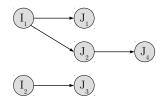
4.4 OPL: conditional operators

Definition: adjacency matrix

The adjacency matrix of a graph G=(V,E) with $V=\{V_1,\ldots,V_N\}$ is a quadratic $N\times N$ -Matrix (a_{ij}) , which holds:

$$a_{ij} = \left\{ egin{array}{ll} 1 & \mathsf{Edge} \ V_i
ightarrow V_j \ \mathsf{exists} \ 0 & \mathsf{otherwise} \end{array}
ight.$$

Adjacency matrix in the example



 \downarrow Translation into adjacency matrix \downarrow

4 Optimization of Graph Problems

CC-BY-SA A. Popp

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

1.3 OPL: custom cuples as data structure

4 OPL: onditional perators

4.3 OPL: custom tuples as data structure

.4 OPL: onditional perators

```
{string} I = ...;
int a [I,I] = [
   [0, 0, 1, 1, 0, 0],
   [0, 0, 0, 0, 1, 0],
   [0, 0, 0, 0, 0, 0],
   [0, 0, 0, 0, 0, 0],
   [0, 0, 0, 0, 0, 0],
   [0, 0, 0, 0, 0, 0],
   [0, 0, 0, 0, 0, 0],
   [0, 0, 0, 0, 0, 0],
];
```

$$x_i \ge \sum_{(i,j) \in E} x_j \qquad \forall i \in I$$

↓ OPL ↓

```
forall(i in I)
  x[i] >= sum (j in I)(a[i,j]*x[j]);
```

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

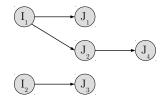
tuples as data structure

4.4 OPL: conditional

Definition: adjacency lists

The adjacency list of a vertex $v \in V$ of a graph G = (V, E) is a set $A_v \subseteq V$, which contains all successors of v.

Adjacency lists in the example



$$A_{I_1} = \{J_1, J_2\}$$
 $A_{I_2} = \{J_3\}$
 $A_{J_1} = \{\}$ $A_{J_2} = \{J_4\}$
 $A_{J_3} = \{\}$ $A_{J_4} = \{\}$

Application of adjacency lists in optimization problems

```
{string} I = ...;
{string} A[I] = [
    {"J1", "J2"},
    {"J3"},
    {},
    {"J4"},
    {},
    {},
    {}
}
```

```
x_i \ge \sum_{(i,j) \in E} x_j \qquad \forall i \in I
```

↓ OPL ↓

```
forall (i in I)
  x[i] >= sum(j in A[i])(x[j]);
```

4 Optimization of Graph Problems

CC-BY-SA A. Popp

4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

4 OPL: onditional perators

4 Optimization of Graph Problems

CC-BY-SA A. Popp

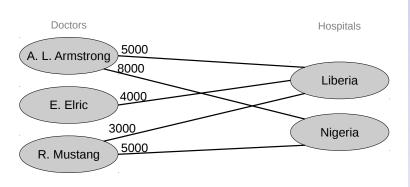
4.1 Short introduction into graph theory

4.2 Representation of graphs in OPL

4.3 OPL: custom tuples as data structure

4.4 OPL: conditional

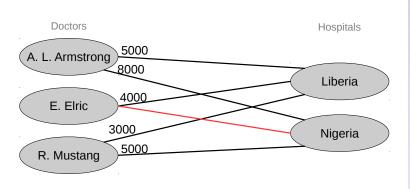
Example: Relieve Doctors



4 Optimization of Graph Problems

- 1.1 Short ntroduction into graph theory
- of graphs in OPL
- 4.3 OPL: custom tuples as data structure
- 4.4 OPL: conditional operators

Example: Relieve Doctors



4 Optimization of Graph Problems

- I.1 Short ntroduction into graph theory
- 4.2 Representation of graphs in OPL
- 4.3 OPL: custom tuples as data structure
- 4.4 OPL: conditional

4.3 OPL: custom tuples as data

structure

Index sets:

set of ressources

set of tasks

Parameters:

set of egdes in the assignment graph

cost of each edge $(r, t) \in E$

Decision variables:

binary variable representing the choice of edge $(r, t) \in E$

Model description:

$$\min \sum_{\substack{(r,t)\in E}} c_{rt} \cdot x_{rt}$$

$$s.t. \sum_{\substack{(r,t)\in E}} x_{rt} = 1 \qquad \forall t \in T \qquad \text{(I)}$$

$$\sum_{\substack{(r,t)\in E\\ x_{rt} \in \{0,1\}}} x_{rt} \leq 1 \qquad \forall r \in R \qquad \text{(II)}$$

- 4.3 OPI : custom tuples as data structure

- Assign prohibitively high costs to missing edges. Disadvantages:
 - unnecessary binary variables
 - susceptible to machine rounding errors
 - only applicable to weighted graphs (if at all)
- Adjacency matrix. Disadvantages:
 - unnecessary binary variables
- Adjacency lists. Disadvantages:
 - \triangleright x[r in R][t in A[r]] \rightarrow Error: "Variable indexer size" not allowed for a generic array."

Tuples are custom data structurs, consisting of elements of other data types.

Definition of a new tuple data type

```
tuple Name_of_the_tuple_data_type {
 Data_type_of_1st_Element Name_of_1st_Element;
 Data_type_of_2nd_Element Name_of_2nd_Element;
  . . .
```

Example: Edges as tuple data type

```
\{\text{string}\}\ V = \{\text{"A", "B", "C"}\};
tupel edge {
  string start;
  string end;
};
```

4 Optimization of **Graph Problems**

> CC-BY-SA A. Popp

4.3 OPI : custom tuples as data structure

tuples as data structure

4.3 OPI : custom

Example: getting the starting vertex of an edge

Example: Definition of an edge as literal

In a tuple data type's literals the elements are sorted into

Single elements of a tuple data type are addressed with a dot.

e.start **" Δ "**

edge e = < "A", "B">;

angle brackets.

Vertices and Edges shall be defined as above.

Application example

$$\sum_{(r,t)\in E} x_{rt} = 1 \qquad \forall t \in T$$

J OPL J

```
forall(t in T)
   sum(\langle r,t \rangle in E)(x[\langle r,t \rangle]) == 1:
```

- 4.1 Short introduction into graph theory
- 4.2 Representation of graphs in OPL
- 4.3 OPL: custom tuples as data structure
- 4.4 OPL: conditional operators

4.4 OPL: conditional operators

conditional operators

4.4 OPL:

Using a colon, we can apply conditions to iteration indexes, which have to be fulfilled for an index to be incorporated by the operator:

sum(iteration index in index set : condition)

resp.

forall (iteration index in index set : condition)

Conditions are logical expressions (not boolean decision variables!)

Construtction of conditions

Literals for logical values

true, false

Comparison operators for logical values

math. notation	=	\neq	\leq	<	\geq	>
OPL Syntax	==	!=	<=	<	>=	>

Logical operators for logical values

math. notation	7	\wedge	V	V
OPL syntax	!	&&	П	! =

4 Optimization of Graph Problems

- 4.1 Short introduction into graph theory
- 4.2 Representation of graphs in OPL
 - 1.3 OPL: custom cuples as data
- 4.4 OPL: conditional operators

4.2 Representation of graphs in OPL

1.3 OPL: custom cuples as data

4.4 OPL: conditional operators

Vertices and Edges shall be defined as above.

Application example

$$\sum_{(r,t)\in E} x_{rt} = 1 \qquad \forall t \in T$$

```
forall(t in T)
  sum(e in E : e.task == t)(x[e]) == 1;
```