Algebraic Path Problems

Jacek Jonczy

Reasoning under **UN**certainty Group
Institute of Computer Science and Applied Mathematics
University of Berne, Switzerland
http://www.iam.unibe.ch/∼run/index.html
email: jonczy@iam.unibe.ch

RUN Seminar

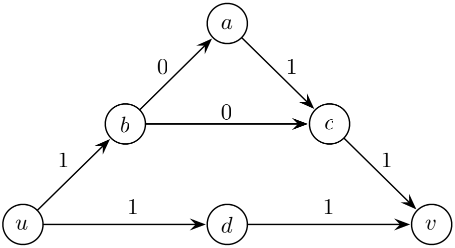
Bern

June 5, 2008

Outline

Outline

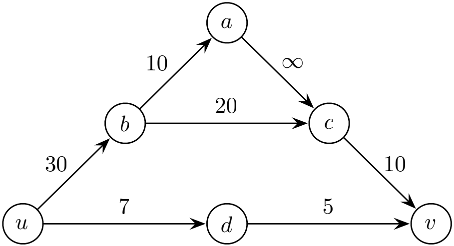
Is there a path from u to v?



$$\begin{array}{ll} 1 \wedge 1 & = 1 \\ 1 \wedge 0 \wedge 1 & = 0 \\ 1 \wedge 0 \wedge 1 \wedge 1 & = 0 \\ \hline \bigvee\{1,0,0\} & = 1 \end{array}$$

$\Longrightarrow$ Transitive Closure (Connectivity)

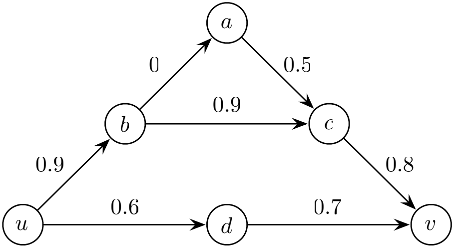
How long is the shortest path from u to v?



$$
\begin{array}{ll}
7+5 & = 12 \\
30+20+10 & = 60 \\
30+10+\infty+10 & = \infty \\
\hline
min\{\infty, 60, 12\} & = 12
\end{array}
$$

$\Longrightarrow$ Shortest Path Problem

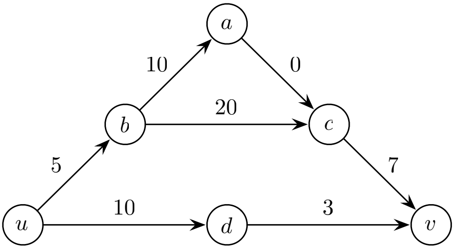
The highest attainable reliability from u to v?



$$
\begin{array}{lcr}
0.6 \cdot 0.7 & = & 0.42 \\
0.9 \cdot 0.9 \cdot 0.8 & = & 0.648 \\
0.9 \cdot 0 \cdot 0.5 \cdot 0.8 & = & 0 \\
\hline
max\{0.42, 0.648, 0\} & = & 0.42
\end{array}
$$

$\Longrightarrow$ Most Reliable Path Problem

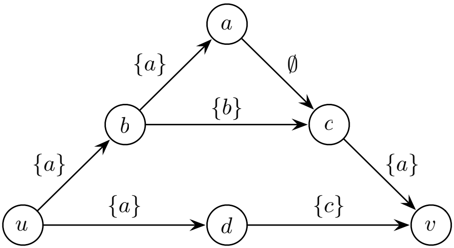
What is the maximum capacity from u to v?



$$\begin{aligned} min\{10, 3\} &= 3 \\ min\{5, 20, 7\} &= 5 \\ min\{5, 10, 0, 7\} &= 0 \\ \hline max\{3, 5, 0\} &= 5 \end{aligned}$$

$\Longrightarrow$ Maximum Capacity Path Problem

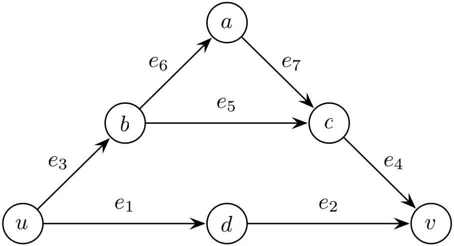
Which language takes the automaton from state u to state v?



$$\begin{array}{lcr}
\{a\} \cdot \{c\} & = & \{ac\} \\
\{a\} \cdot \{b\} \cdot \{a\} & = & \{aba\} \\
\{a\} \cdot \{a\} \cdot \emptyset \cdot \{a\} & = & \emptyset \\
\hline
\bigcup \{ac, aba, \emptyset\} & = & \{aba, ac\}
\end{array}$$

$\Longrightarrow$ Language accepted by a Finite State Automaton

What are the pathsets for the u, v-connectivity of the network?



$$
\begin{array}{lcr}
e_1 \wedge e_2 & = & p_1 \\
e_3 \wedge e_5 \wedge e_4 & = & p_2 \\
e_3 \wedge e_6 \wedge e_7 \wedge e_4 & = & p_3 \\
\hline
\bigvee\{p_1, p_2, p_3\} & = & p_1 \vee p_2 \vee p_3
\end{array}
$$

$\Longrightarrow$ Structure Function for u, v-Connectivity

Outline

Single-Source Shortest-Paths Problem

- Given: weighted graph $G = (V, E, \lambda)$, $\lambda : E \to \mathbb{R}$, and $s \in V$ the source of all considered paths
- Problem: compute for every $v \in V$ the minimal weight of all paths from s to v, $d_{s,v}$
- Relaxation: relaxing an edge (u, v)
  - test whether the shortest path to v found so far can be improved by going through u. If so, update the distance estimate of v.
  - formally: maintain a function $\delta : V \to \mathbb{R} \cup \{\infty\}$, initialized as follows:

$$\delta(v) = \begin{cases} 0 & \text{if } s = v, \\ \infty & \text{otherwise.} \end{cases}, \ \forall\, v \in V$$

  - define the operation $relax(u, v)$:

$$\textbf{if } \delta(v) > \delta(u) + \lambda(u, v) \textbf{ then } \delta(v) := \delta(u) + \lambda(u, v)$$

- NB: relaxation used by algorithms such as Dijkstra and Bellman-Ford

Algorithms for the Single-Source Problem

- Dijkstra Algorithm
  - input: $\mathbb{R}_{\geq 0}$-weighted graph.
  - output: $\forall\ v \in V :\ \delta(v) = d_{s,v}$.
  - complexity:
    1. with priority queue (as linear list): $\mathcal{O}(|V|^2 + |E|) = \mathcal{O}(|V|^2)$
    2. with binary heap: $\mathcal{O}((|V| + |E|) \cdot log|V|) = \mathcal{O}(|E| \cdot log|V|)$
    3. with fibonacci heap: $\mathcal{O}(|V| \cdot log|V| + |E|)$
- Bellman-Ford Algorithm
  - input: $\mathbb{R}$-weighted graph.
  - output:
    1. graph does not contain negative-weighted cycles: $\forall\ v \in V :\ \delta(v) = d_{s,v}$.
    2. graph contains such cycles: stop with corresp. message
  - complexity: $\mathcal{O}(|V| \cdot |E|)$, hence in the worst case: $\mathcal{O}(|V|^3)$

All-Pairs Shortest-Paths Problem

- Given: weighted graph $G = (V, E, \lambda)$, $\lambda : E \to \mathbb{R}$, such that there are no negative weighted cycles.
- Initial data structure: weighted matrix $W_G = (w_{u,v})$ where

$$w_{u,v} = \begin{cases} 0 & \text{if } u = v, \\ \infty & \text{if } u \neq v,\ (u,v) \notin E, \\ \lambda(u,v) & \text{if } u \neq v,\ (u,v) \in E \end{cases}.$$

- Problem: compute the shortest distance matrix $D = (d_{u,v})_{1 \leq u,v \leq n}$.

Algorithms for the All-Pairs Problem

- Iteration algorithm
  - complexity: $\mathcal{O}(|V|^4)$
- Iteration algorithm with Doubling Up
  - complexity: $\mathcal{O}(|V|^3 \cdot log|V|)$
- Floyd-Warshall Algorithm
  - complexity: $\mathcal{O}(|V|^3)$
- Johnson's Algorithm
  - uses Dijkstra and Bellman-Ford
  - well-suited for sparse graphs
  - complexity: $\mathcal{O}(|V| \cdot |E| \cdot log|V|)$ (in the simplest case)

Further Variants of the Shortest Path Problem

- *Single-destination shortest-paths problem:* Find a shortest path to a given destination vertex (terminal) t from every other vertex v. NB: by reversing the direction of edges in the graph, the problem can be reduced to a single-source problem.
- *Single-pair shortest-path problem:* Find the shortest path from u to v for given vertices u and v. Special case of single-source and all-pairs problems. NB: no algorithm is known that is (asymptotically) faster than the best single-source algorithms in the worst case.
- *k-shortest (k-best) paths problem:* Find the k shortest (different) paths (single-source or all-pairs problem)

Outline

Semiring

Definition (Semiring)

A semiring is an algebraic structure $(S, \oplus, \otimes)$ such that

- $(S, \oplus)$ is a commutative monoid:
  - i. $\oplus$ is commutative
  - ii. $\oplus$ is associative
  - iii. $\bar{0}$ is the neutral element for $\oplus$
- $(S, \otimes)$ is a monoid:
  - iv. $\oplus$ is asssociative
  - v. $\bar{1}$ is the neutral element for $\otimes$
- $\otimes$ distributes over $\oplus$ (from right and left), i.e. $\forall\ a, b, c \in A$:
  - vi. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
- $\bar{0}$ annihilates $\otimes$, i.e. $\forall\ a \in A$:
  - vii. $a \otimes \bar{0} = \bar{0}$

Idempotent, Commutative, and Bounded Semirings

Definition (Idempotent Semiring)

A semiring $(S, \oplus, \otimes)$ is called *idempotent* if

$$\forall\ a \in S \text{ holds : } a \oplus a = a$$

Definition (Commutative Semiring)

A semiring $(S, \oplus, \otimes)$ is called *commutative* if

$$\forall\ a, b \in S \text{ holds : } a \otimes b = b \otimes a$$

Definition (Bounded Semiring)

A semiring $(S, \oplus, \otimes)$ is called *bounded* if

$$\forall\ a \in S :\ \bar{1} \oplus a = \bar{1} \quad (\bar{1} \text{ annihilates } \oplus)$$

Ordered Semirings

Definition (Ordered Semiring)

A semiring $(S, \oplus, \otimes)$ is called *ordered* when its partial order relation $\preccurlyeq$ is monotone w.r.t to both operations. Then we have:

$$a \preccurlyeq b \text{ and } a' \preccurlyeq b' \Longrightarrow a \oplus a' \preccurlyeq b \oplus b' \quad \text{and} \quad a \otimes a' \preccurlyeq b \otimes b'$$

Obtaining an ordered semiring:

- Take idempotent semiring and define partial order by
$$a \preccurlyeq b \iff a \oplus b = b \quad \text{(natural order)}$$
- Take (partially or linearly) ordered semigroup $(S, \otimes)$ with neutral element $\bar{1}$, and $\oplus$ is the sup or inf operation (max or min in case of a total order). The order relation of an ordered semigroup is monotone w.r.t. the multiplication.

Complete Semirings

Definition (Complete Semiring)

A semiring $(S, \oplus, \otimes)$ is called *complete* if the existence of (countable) infinite sums is guaranteed. In particular, for every countable subset of S, we require:

- infinite commutativity of $\oplus$
- infinite associativity of $\oplus$
- infinite distributivity ($\otimes$ distributes over infinite sums)

Closed Semirings

Definition [Lehmann, 1977]

A semiring $(S, \oplus, \otimes)$ is called *closed* if there is an additional unary operation *, called closure, such that

$$\forall\, a \in S: \; a^* = \bar{1} \oplus a \otimes a^* = \bar{1} \oplus a^* \otimes a$$

Definition [Rote, 1989]

Consider a semiring $(S, \oplus, \otimes)$ and the iteration equation

$$x = \bar{1} \oplus a \otimes x, \tag{1}$$

where $a, x \in S$. If there is always a solution a^* of (1), i.e. a fixed point exists, then the semiring is called *closed*.

Closed Semirings, cont.

Definition [Mohri, 2002]

A semiring $(S, \oplus, \otimes)$ is called k-*closed* if

$$\forall\ a \in S: \quad \bigoplus_{n=0}^{k+1} a^n = \bigoplus_{n=0}^{k} a^n$$

where $k \geq 0$.

NB: a bounded semiring is a special k-closed semiring, namely for $k = 0$.
(0-closeness = boundedness)

Simple Semirings

Definition (Simple Semiring)

A *simple* semiring is a semiring which is bounded and closed.

Definition (Dijkstra Semiring)

A *Dijkstra* semiring is a simple semiring $(S, \oplus, \otimes)$ with the property

$$a \oplus b = \text{ either } a \text{ or } b \quad \forall\, a, b \in S.$$

In other words, its natural order defined by

$$a \succcurlyeq b \iff a \oplus b = a$$

is a total order.

Outline

General Definition

Definition (Algebraic Path Problem)

The *Algebraic Path Problem* consists in performing a special unary operation, called the *closure*, over a square matrix with entries in a semiring [Fink, 1992].

General distinction between:

- graph approach, and
- matrix approach.

Graph Approach

Consider a semiring $(S, \oplus, \otimes)$ and a graph $G = (V, E, \lambda)$, where $\lambda : E \rightarrow S$ is the weight function.

Definition (Algebraic Path Problem)

Algebraic Path Problem: compute the sum of the weights of all paths from v_i to v_j in terms of the semiring, for all pairs v_i, v_j:

$$d(v_i, v_j) = \bigoplus_{p \in \mathcal{P}_{i,j}} \lambda(p) \qquad \text{(sum-weight function)}$$

where $\lambda(p) = \bigotimes_{k=i}^{j} \lambda(v_k, v_{k+1})$. [Fink, 1992, Rote, 1989]

Prerequisites:

- Complete, idempotent Semiring

Graph Approach, cont.

Definition (Algebraic Path Problem)

Algebraic Path Problem: for a *complete semiring* and the graph G, compute explicitly the $n \times n$-matrix $D = (d_{ij})$, the "distance matrix", such that

$$d_{ij} = \bigoplus_{p \in \mathcal{P}_{i,j}} \lambda(p)$$

where $\lambda(p) = \bigotimes_{k=i}^{j} \lambda(v_k, v_{k+1})$. [Vogler, 2006]

Prerequisites:

- Complete Semiring

Matrix Approach

- Define addition and multiplication of $n \times n$-matrices $A = (a_{ij})$ and $B = (b_{ij})$ over the semiring $(S, \oplus, \otimes)$:

$$A \oplus B = (a_{ij} \oplus b_{ij}), \quad A \otimes B = (c_{ij}) \quad \text{where } c_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes b_{kj}$$

- Define the k-th power of matrix A:

$$A^k = (d_{ij}) \text{ where } d_{ij} = \bigoplus_{r=0}^{k-1} a_{ir} \otimes a_{rj}, \;\; A^0 = I$$

- Define the *closure* of the matrix A:

$$A^* = \bigoplus_{k \geq 0} A^k \tag{2}$$

Matrix Approach, cont.

Definition (Algebraic Path Problem)

Algebraic Path Problem: given a matrix A over a semiring $(S, \oplus, \otimes)$, compute its closure A^* defined by (2).

- The closure operation on matrices satisfies the closure property [Lehmann, 1977]:

$$A^* = I \oplus A \otimes A^* = I \oplus A^* \otimes A$$

- The closure of a $n \times n$-matrix over a *simple* semiring may be computed as follows [Lehmann, 1977]:

$$A^* = \bigoplus_{k=0}^{n-1} A^k = I \oplus A \oplus A^2 \oplus \ldots \oplus A^{n-1}$$

NB: semiring does neither require completeness, idempotency, nor annihilation of $\otimes$ by $\bar{0}$.

Examples revisited

Examples of closed (or complete) semirings

1. Transitive Closure

- Graph $G = (V, E, \lambda)$, $\lambda : E \to \{0, 1\}$
- Boolean semiring: $\texttt{Bool} = (\{0, 1\}, \vee, \wedge, 0, 1)$
- Satisfies properties of a Dijkstra semiring
- Problem of computing the transitive closure of G: APP over G and `Bool`

Examples revisited

Examples of closed (or complete) semirings

2. Shortest Path Problem

- Graph $G = (V, E, \lambda)$, $\lambda : E \to \mathbb{R}_{\geq 0} \cup \{\infty\}$
- Tropical semiring: $\texttt{Trop} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, min, +, \infty, 0)$
- Satisfies properties of a Dijkstra semiring
- Problem of computing the length of the shortest path (for all pairs): APP over G and $\texttt{Trop}$

Examples revisited

Examples of closed (or complete) semirings

3. Maximum Reliability Path Problem

- Graph $G = (V, E, \lambda)$, $\lambda : E \to [0, 1]$
- Viterbi semiring: $\texttt{Viterbi} = ([0, 1], max, \cdot, 0, 1)$
- Satisfies properties of a Dijkstra semiring
- Problem of computing the highest attainable reliability between two nodes: APP over G and `Viterbi`

Examples revisited

Examples of closed (or complete) semirings

4. Maximum Capacity Path Problem

- Graph $G = (V, E, \lambda)$, $\lambda : E \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$
- Bottleneck semiring: `Bottle` $= (\mathbb{R}_{\geq 0} \cup \{\infty\}, max, min, 0, \infty)$
- Satisfies properties of a Dijkstra semiring
- Problem of computing the greatest transfer capacity between two nodes: APP over G and `Bottle`

Examples revisited

Examples of closed (or complete) semirings

5. Language accepted by a Finite State Automaton (FSA)

- FSA $M = (Q, \Sigma, \delta, F, q_0)$, where $\delta : Q \times \Sigma \to Q$
- Set of words which lead from state q_1 to state q_2:

$$\mathcal{L}(q_1, q_2) = \{\omega \in \Sigma^* \mid q_2 \in \bar{\delta}(q_1, \omega)\}, \quad \text{where } \bar{\delta} : Q \times \Sigma^* \to Q$$

$$\text{such that } \bar{\delta}(q, \epsilon) = \{q\} \text{ and } \bar{\delta}(q, \omega a) = \bigcup_{q' \in \bar{\delta}(q,\omega)} \delta(q', a)$$

- Graph $G = (V, E, \lambda)$, where $V = Q$, $E = Q \times Q$, $\lambda : E \to \mathscr{P}(\Sigma^*)$
- Kleene semiring: $\texttt{Kleene} = (\mathscr{P}(\Sigma^*), \cup, *, \emptyset, \{\epsilon\})$ (closed)
- Problem of determining $\mathcal{L}(q_1, q_2)$ for all pairs of states q_1 and q_2: APP over G and `Kleene`

Examples revisited

Examples of closed (or complete) semirings

6. Two-Terminal Connectivity

- Network $G = (V, E, \lambda)$, $\lambda : E \rightarrow \mathcal{B}_n$
- $\mathcal{B}_n$ set of n-ary Boolean functions
- Semiring of Boolean functions: $\texttt{BF} = (\mathcal{B}_n, max, min, 0, 1)$
- Simple semiring, but not a Dijkstra semiring
- Problem of computing the structure function for u, v-connectivity for all pairs u and v: APP over G and $\texttt{BF}$

Examples revisited

Examples of closed (or complete) semirings

7. Final example: matrix inversion

- Consider the (partial) closed semiring $(\mathbb{R}, +, \cdot, 0, 1)$
- The closure of $x \in \mathbb{R}$ is defined by $x^* = \frac{1}{1-x}$, $\forall\ x \neq 1$
- This semiring can be completed to a closed semiring. Requires $\bar{0}$ to be not annihilating for $\otimes$ [Lehmann, 1977].
- This semiring for matrix inversion cannot be described by the graph approach, since addition is not idempotent

Methods to solve the APP

- Iterative methods (search for a fixed point):
  - Jacobi iteration
  - Gauss-Seidel iteration
- Direct methods
  - generalized versions of known algorithms, such as:
    - Warshall's algorithm for transitive closure
    - Warshall-Floyd algorithm for shortest path
    - Kleene's algorithm for regular expressions
    - Gauss-Jordan algorithm for matrix inversion
  - Aho's algorithm
  - Mahr's algorithm
  - Adapted versions of Knuth's and Dijkstra's algorithms

The General Solution

Generalized version of the Warshall-Floyd-Kleene (WFK) algorithm

---

Algorithm 1: Generalized WFK

Input: $x \times n$ matrix $A = (a_{ij})$, $a_{ij} \in S$, S is a closed semiring.
Output: C, the closure of A.

begin
  $A^{(0)} \leftarrow A$
  for k **from** 1 **to** n **do**
    foreach $1 \leq i, j \leq n$ **do**
      $[A^{(k)}]_{ij} \leftarrow [A^{(k-1)}]_{ij} \oplus [A^{(k-1)}]_{ik} \otimes ([A^{(k-1)}]_{kk})^* \otimes [A^{(k-1)}]_{kj}$
    end
  end
  $C \leftarrow I_n + A^{(n)}$
end

- Intuitively, $([A^{(k-1)}]_{kk})^*$ represents the "sum" of all cycles with nodes in $\{1, \ldots, k-1\}$ that pass through node k (length $k-1$)
- Complexity: $\mathcal{O}(n^3 \cdot (T_\oplus + T_\otimes + T_*))$

Overview: Instances of the APP

The APP generalizes many important problems (choice):

- Graph and Network Problems
  - transitive closure and transitive reduction
  - shortest distance problems (distance functions)
  - capacity problems (max flow, network capacity, tunnel-problem)
  - connectivity measures for reliability networks
  - stochastic communication network problems
- Linear Algebra
  - computing the inverse of a matrix
- Regular Language Problems
  - correspondance: regular expressions and finite state automata

Outline

Our Approach I

1. Computing the structure function from the APP point of view:
   - Semiring of Boolean functions
   - All-pairs connectivity → generalized WKF-algorithm
   - Single-source connectivity → e.g. Dijkstra
   - Generalized single-source connectivity → JH-algorithm
2. Obtaining the struture function for two-terminal reliability (s, t-connectivity) as instance of the APP
3. Extending the APP
   - All-pairs APP
   - Single-source APP
   - Generalized single-source APP (source-to-any, source-to-all,. . .)
4. Network reliability computation (computing the structure function) as projection problem in a VA

Outline

References I

[Fink, 1992] E. Fink.
A survey of sequential and systolic algorithms for the algebraic path problem.
Technical Report CS-92-37, Department of Computer Science, University of Waterloo, 1992.

[Mohri, 2002] M. Mohri.
Semiring frameworks and algorithms for shortest-distance problems.
Journal of Automata, Languages and Combinatorics, 7(3):321–350, 2002.

[Lehmann, 1977] Daniel Lehmann.
Algebraic structures for transitive closure.
Theoretical Computer Science, 4:59–76, 1977.

References II

[Rote, 1989] Günter Rote.
Path problems in graphs.
In G. Tinhofer, E. Mayr, H. Noltemeier, M. M. Syslo, and R. Albrecht, editors, *Computational Graphs Theory*, Computing Supplementum 7. Springer-Verlag, 1990.

[Shier, 1991] Douglas R. Shier.
Network reliability and algebraic structures.
Oxford Clarendon Press, New York, USA, 1991.

[Warshall, 1962] Stephen Warshall.
A theorem on boolean matrices.
Journal of the ACM, 9(1):11 – 12, 1962.