Dijkstra, Kleene, Knuth
(revised version)

Bernhard Möller

Institut für Informatik, Universität Augsburg
1 The Shortest Path Problem

informal problem statement:

- given:
  - directed graph \((n, e)\)
  - with node set \(n\) and non-negatively weighted edge set \(e\)
  - a starting node \(s \in n\)

- task: for each \(v \in n\) return
  - length of a shortest path from \(s\) to \(v\)
  - or \(\infty\) if there is no path from \(s\) to \(v\).
algebraic formulation:

- calculate \( d = s \; e^* \)

- where \( ; \) is path concatenation (under adjustment of costs)

aim of derivation: eliminate the expensive star operation

earlier version: [Backhouse et al. 92/94]
2 Some Properties of Paths

- general idea: work with an algebra of path sets (and their costs)
- edge sets: sets of paths with 2 nodes
- node sets: sets of singleton paths
- concatenation: glue at common intermediate node (associative)
- for node set \( m \) and path set \( a \)
  - \( m ; a \) set of paths in \( a \) that start in \( m \)-nodes
  - \( a ; m \) set of paths in \( a \) that end in \( m \)-nodes
- hence set \( n \) of all nodes is the identity of composition
- \( a^* \) arbitrary finite iteration of \( a \), i.e., all paths that can be constructed out of an arbitrary finite number of \( a \)-paths
choice:

- **$a \sqsubset b$:** for all pairs of nodes take shortest connecting paths
  provided by $a$ or $b$

- refinement order: $a \sqsubseteq b =_{df} a \sqsubset b = b$
  
  ($b$ refines $a$ iff it offers the less costly paths)

- since singleton paths are always cheapest (cost 0), set $n$ of all
  nodes refines all sets: $a \sqsubseteq n$

- a full graph may offer better paths than a restricted one:
  $m \sqsubseteq a$

- composition distributes over choice, hence is $\sqsubseteq$-isotone

- convention: composition binds tighter than choice

further details in Appendix II
three essential properties used in the derivation:

- for graph node \( p \) and path set \( a \):
  \[
  p ; a ; p \subseteq p
  \]
  \((\text{no detours})\)
  since all path costs are non-negative, any path from \( p \) to itself cannot be cheaper than the 0-cost trivial singleton path consisting just of \( p \)

- \( a^* = n \cdot a ; a^* = n \cdot a^* ; a \) \((\text{star recursion})\)
  iteration of \( a \) either uses zero \( a \)-paths or one \( a \)-path followed/preceded by zero or more others

- \((b \cdot c)^*\): arbitrary alternations of \( b \) paths and \( c \) paths
  \[
  (b \cdot c)^* = c^* ; (n \cdot b ; (c \cdot b)^*)
  \]
  \((\text{path grouping})\)
  \[
  = (n \cdot (b \cdot c)^* ; b) ; c^*
  \]
  exhibit maximal \( c \)-sequences at the beginning or end
3 Dijkstra’s Algorithm

central ideas:

- generalise the problem by using a set \( ok \) of nodes for which the problem is solved exactly
- initially, \( ok \) is empty
- extend this set node by node till all are in \( ok \)
- for each node outside \( ok \) the algorithm computes an approximation to \( d \), viz.
- the length of a shortest path whose interior nodes are from \( ok \)
formalisation:

- use the path algebra with node set \( n \) and edge set \( e \)

- for \( ok \leq n \) define generalised function \( dd \) by
  \[
  dd(ok) =_{df} s ; (ok ; e)^* 
  \]

- expresses that \( dd(ok) \) only considers paths with interior nodes in \( ok \)

- then, by neutrality of \( n \) w.r.t. composition, \( d = dd(n) \)

- “strategy”: extract maximal subexpressions of form \( p ; a ; p \) to allow application of no-detours rule
plan of derivation: find an inductive version of $dd$ that does not use star operations anymore

- maintain the invariant that $dd$ solves the problem exactly, i.e., using all possible paths, for end nodes in $ok$:

$$s; (ok ; e)^* ; ok = s; e^* ; ok$$

- more compactly,

$$dd(ok) ; ok = d; ok$$  \hspace{1cm} (1)$$

- induction base: $ok = \emptyset$

$$dd(\emptyset) = s; \emptyset^* = s; n = s$$

- invariant holds trivially for $dd(\emptyset)$
induction step: calculate behaviour of $dd$ when $ok$ is extended by a node $w \leq \neg ok$

from this infer how to choose $w$ appropriately to maintain the invariant
\[ dd(w \lbrack ok) \]
\[ = \{ \text{definition } dd \text{ and distributivity } \} \]
\[ s ; (w ; e \lbrack ok ; e)^* \]
\[ = \{ \text{path grouping and distributivity } \} \]
\[ s ; (ok ; e)^* ; (n \lbrack w ; e ; ((w \lbrack ok) ; e)^*) \]
\[ = \{ \text{definition } dd \text{ and abbreviation } h =_{df} (w \lbrack ok) ; e \} \]
\[ dd(ok) ; (n \lbrack w ; e ; h^* \) \]
simplification of second alternative \((h =_{df} (w \mathbf{\Dagger} \ ok); e)\):

\[
w; e; h^*
\]

\[= \ \{ \text{star recursion and definition of } h \ \}\]
\[
w; e \mathbf{\Dagger} w; e; h^*; (w \mathbf{\Dagger} \ ok); e
\]

\[= \ \{ \text{distributivity } \\}\]
\[
w; e \mathbf{\Dagger} w; e; h^*; w; e \mathbf{\Dagger} w; e; h^*; ok; e
\]

\[= \ \{ \text{middle summand } \subseteq \text{ first one by no-detours rule } \ \}\]
\[
w; e \mathbf{\Dagger} w; e; h^*; ok; e
\]

substituted back:

\[
\text{dd}(w \mathbf{\Dagger} \ ok) = \text{dd}(ok); (n \mathbf{\Dagger} w; e \mathbf{\Dagger} w; e; h^*; ok; e)
\]

now continue simplification with third alternative (after distribution)
\[ \text{informal interpretation: shortest paths to nodes outside } ok \text{ cannot loop back through } ok \]
in sum:

\[
dd(w \parallel ok) = dd(ok); (n \parallel w ; e) \quad (*)
\]

algebraic equivalent of the usual set of assignments

\[
dd[v] = \min (dd[v], dd[w] \parallel weight(w,v))
\]

for \( v \leq n \)

(where by the invariant \( dd(ok); ok = d; ok \) only the subset \( \neg ok - \{w\} \) needs to be considered)

now choose \( w \) such that the invariant holds for \( w \parallel ok \) again

sufficient: \( d; w = dd(w \parallel ok); w \)

by (*) and no-detours rule the rhs is equal to \( dd(ok); w \)
abbreviation: \( f =_d f \; dd(\textit{ok}) = s ; (\textit{ok} ; e^*) \)
\[
d ; w
\]
\[
= \quad \{ \text{definition of } d \}
\]
\[
s ; e^* ; w
\]
\[
= \quad \{ \text{path grouping, using } e = \textit{ok} ; e \; [\neg \textit{ok} ; e] \}
\]
\[
s ; (\textit{ok} ; e^*) ; (n \; [\neg \textit{ok} ; e ; e^*]) ; w
\]
\[
= \quad \{ \text{definitions of } f \text{ and setting } e^+ =_d e ; e^* \}
\]
\[
f ; (n \; [\neg \textit{ok} ; e^+]) ; w
\]
\[
= \quad \{ \text{splitting } \neg \textit{ok} \text{ into its nodes and distributivity } \}
\]
\[
f ; w \; [\; v \leq \neg \textit{ok} \; f ; v ; e^+ ; w)
\]

so goal achieved if \( [\; v \leq \neg \textit{ok} \; f ; v ; e^+ ; w \subseteq f ; w \)
reduction:
\[ \forall \leq \neg ok: f; \nu; e^+; w \subseteq f; w \]
\[ \iff \{ \text{universal characterisation of choice} \} \]
\[ \forall \nu \leq \neg ok: f; \nu; e^+; w \subseteq f; w \]
\[ \iff \{ \text{instance } f; w; e^+; w \subseteq f; w \text{ of no-detours rule} \} \]
\[ \forall \nu \leq \neg ok: f; \nu \subseteq f; w \]

this holds iff \( w \) is a node with minimal cost along \( ok \) paths
complete algorithm:

\[
\begin{align*}
\text{dd} (\emptyset) &= s \\
\text{dd} (ok \parallel w) &= \text{dd} (ok) \; (n \parallel w ; e) \\
\text{if } ok \neq \emptyset \text{ and } w \leq \neg ok \text{ satisfies } \\
\forall v \leq \neg ok : \text{dd} (ok) ; v \sqsubseteq \text{dd} (ok) ; w
\end{align*}
\]
4 Knuth’s Generalisation

observations:

- edge $XY$ with weight $m$ corresponds to an automaton transition $X \xrightarrow{m} Y$
- matrix algebra approach works, because the problem is essentially about automata/regular languages
- Knuth generalises this to a context-free setting
approach:

- use restricted cflgs of with productions of the shape \((n \geq 0)\)
  \[ X_i ::= f(X_{i1}, \ldots, X_{in}) \]

- and associated \textbf{\black{N}}-valued interpreting functions \(f^I\) that are

- isotone in each argument

- \textbf{\textit{superior}}, i.e., satisfy

\[
\forall j : f^I(x_1, \ldots, x_n) \geq x_j
\]

- task: compute for all \(i\)

\[
m(X_i) =_{df} \min \{w^I : w \in L(X_i)\}
\]
the shortest path example:

- edge $X \xrightarrow{m} Y$ gives production
  \[ X ::= f(Y) \]

- with $f^I(x) =_{df} m + x$

- $f$ is isotone and superior

- for start node $S$ add a production $S ::= 0$
algorithm:

- use again a set \( ok \) and an auxiliary function \( mm \)
- \( ok \) is the set of nonterminals \( X \) for which \( m(X) \) has been determined
- for all other \( Y \) the value \( mm(Y) \) approximates \( m(Y) \)
- invariant: \( \forall X \in ok : mm(X) = m(X) \)
- initialisation: \( ok := \emptyset ; \forall X : mm(X) := \infty \)
loop:

- if all nonterminals are in \( \text{ok} \), stop
- otherwise, for all \( Y \not\in \text{ok} \), compute

\[
\text{mm}(Y) =_{df} \min \{ f^I(m(X_1), \ldots, m(X_n)) \mid Y ::= f(X_1, \ldots, X_n) \land \{X_1, \ldots, X_n\} \subseteq \text{ok} \}
\]

(if the set involved is empty then \( \text{mm}(Y) = \infty \))
- choose a \( Y \) with minimum \( \text{mm}(Y) \)
- \( \text{ok} := \text{ok} \cup \{Y\} \)
- \( m(Y) := \text{mm}(Y) \)
challenge:

find a nice calculational correctness proof/derivation for Knuth’s algorithm
Appendix I: Just for Fun - The Floyd/Warshall Algorithm

dthis is the all-pairs shortest non-empty path problem

specification even simpler than for Dijkstra: compute $e^+$

central idea: use again a set $ok$ that restricts the inner nodes of
paths and increment it stepwise

specification of auxiliary function:

$$rt(ok) =_{df} e ; (ok ; e)^*$$

("restricted transitive closure")
here another star property is useful:

\[(a \cdot b)^* = a^* ; (b ; a^*)^* = (a^* ; b) ; a^* \quad \text{(star of sum)}\]

induction base:

\[rt(\emptyset) = e ; \emptyset^* = e ; n = e\]
induction step: for arbitrary node $w$:

\[
rt(\text{ok } [] w)
\]
\[
= \{ \text{ definition rt and distributivity } \}\]
\[
e; (\text{ok } ; e [] w ; e)^*
\]
\[
= \{ \text{ star of sum } \}
\]
\[
e; (\text{ok } ; e)^* ; (w ; e ; (\text{ok } ; e))^*
\]
\[
= \{ \text{ fold e;(ok ; e)* twice to } f =_{df} rt(\text{ok}) \}\]
\[
f ; (w ; f)^*
\]
\[
= \{ \text{ star recursion and distributivity } \}
\]
\[
f [] f ; w ; f ; (w ; f)^*
\]
\[
= \{ \text{ star recursion and distributivity } \}
\]
\[
f [] f ; w ; f [] f ; w ; f ; (w ; f)^* ; w ; f
\]
\[
= \{ \text{ since third alternative } \preceq \text{ second one by no-detours rule } \}
\]
\[
f [] f ; w ; f
\]
to guarantee termination, choose $w \not\in \textit{ok}$

complete algorithm:

\[
rt(\emptyset) = e \\
rt(\textit{ok} \sqcup w) = f \sqcup f; w; f
\]

where $f = rt(\textit{ok})$ and $w \not\in \textit{ok}$

depending on the underlying cost semiring (see Appendix II) this is the Floyd or Warshall algorithm
Appendix II: Algebraic Background

Definition 4.1 *semiring*: structure \((S, +, \cdot, 0, n)\) such that

- \((S, +, 0)\) is a commutative monoid
- \((S, \cdot, 1)\) is a monoid
- the distributive laws hold
- \(0\) is an annihilator: \(0 \cdot a = 0 = a \cdot 0\)

if \(S\) is idempotent, i.e., \(x + x = x\), the relation \(a \leq b \iff df a + b = b\)
is a partial order, the *natural* order
interpretation:

\(+ \leftrightarrow \text{choice,}\)
\(\cdot \leftrightarrow \text{sequential composition}\)

\(0 \leftrightarrow \text{empty set of choices}\)
\(1 \leftrightarrow \text{identity}\)
\(\leq \leftrightarrow \text{increase in information or in choices}\)

**Example 4.2** tropical semiring:

- \((\text{min}, +) = (\mathbb{N}_\infty, \text{min}, +, \infty, 0)\)
- natural ordering: converse of the standard ordering on \(\mathbb{N}_\infty\)
- \(1 = 0\) is the largest element.
generalisation: \textit{cost algebra}

- idempotent semiring with total natural order
- in which \( 1 \) is the greatest element

further examples:

- \( \mathbb{R}_{\geq 0} \cup \{\infty\} \) with the operations as above
- Booleans \( \mathbb{B} \) with implication order
\[ \text{MAT}(M, S) = (S^{M \times M}, +, \cdot, 0, 1) \]

- set of matrices with indices in \( M \) and elements of semiring \( S \) as entries
- again a semiring
- idempotent iff \( S \) is
- natural order: componentwise
- \( \text{MAT}(M, \mathbb{B}) \) isomorphic to semiring \( \text{REL}(M) \) of binary relations over \( M \) under union and composition

modelling graphs with edge weights:
- \( \text{MAT}(N, S) \) where \( S \) is a cost algebra
representing sets of graph nodes

- **test semiring** [Kozen 97]: pair \((S, \text{test}(S))\) with Boolean subalgebra \(\text{test}(S) \subseteq [0, 1]\) such that
  - \(0, 1 \in \text{test}(S)\)
  - + is join and \(\cdot\) is meet in \(\text{test}(S)\)
  - \(S\) is **discrete** if \(\text{test}(S) = \{0, 1\}\)

- \(S = (\text{min}, +)\) is discrete, but \(\text{MAT}(M, S)\) can be made non-discrete:

  - choose as tests all matrices with tests on the main diagonal and \(0\) outside
over discrete $S$, matrix $p$ is a *point* if it is an atom in $\text{test}(\text{MAT}(M, S))$,

i.e., if it has exactly one entry $1$ in its main diagonal (and hence $0$ everywhere else)

general tests represent subsets of $M$ in the analogous way

for points $p$ and $q$ and matrix $a$

$$(p \cdot a \cdot q)_{uv} = \begin{cases} 
a_{uv} & \text{if } u = p \land v = q \\
0 & \text{otherwise}
\end{cases}$$
Lemma 4.3  Consider a discrete cost algebra $S$, a point $p$ and an arbitrary matrix $a$ of $\text{MAT}(M, S)$. Then $p \cdot a \cdot p \leq p$.

since $\mathbb{B}$ is a cost algebra, this property holds for the relation semiring $\text{REL}(M)$, too
iteration: add Kleene star and plus with standard axioms [Kozen94]

**Example 4.4** Since in \((\min,+)\) the multiplicative unit \(1 = 0\) is the largest element, and \(x^* = 1\) for all \(x \leq 1\), we can extend \((\min,+)\) uniquely to a Kleene algebra by setting \(n^* = 1\) for all \(n \in \mathbb{N}_\infty\).

useful law

\[(b+c)^* = (1+(b+c)^* \cdot b) \cdot c^* = b^* \cdot (1+b \cdot (b+c)^*)\] (path grouping)
fact [Conway'71]: MAT(M, S) over Kleene algebra S can be extended to a Kleene algebra

Corollary 4.5 Consider a discrete cost algebra S, a point p and an arbitrary matrix a of MAT(M, S). Then p · a* · p = p.

reason: 1 ≤ a* holds for all Kleene algebras
connection to path problems:

- for graph matrix $a \in \text{MAT}(M, S)$ over cost algebra $S$ and $x, y \in M$:
- element $a^i_{xy}$ gives the minimum cost of paths with exactly $i$ edges from $x$ to $y$
- hence $a^*_{xy}$ is the minimum cost along arbitrary paths from $x$ to $y$