# Quickest Path Distances on Context-Free Labeled Graphs*

PHILLIP G. BRADFORD
The University of Alabama
Department of Computer Science
Box 870290
Tuscaloosa, AL 35487-0290
USA

*Abstract:* Given $\sigma$ units of data and a digraph $G = (V, E)$ whose edges have delays, bandwidth constraints, and are labeled by terminals from a *CFG (context-free grammar)* $\mathcal{G}$. A path $p$ adheres to $\mathcal{G}$'s path constraints iff the concatenation of all terminals along $p$ forms a word of the language generated by $\mathcal{G}$. The *all-pairs quickest CFG labeled-path distance problem* is: for all pairs of vertices, find the minimum path-cost to send $\sigma$ data units accounting for edge delays while adhering to labeled path and bandwidth constraints. This paper iteratively applies dynamic programming-based labeled path algorithms to CFG-labeled bandwidth-stratified induced subgraphs of an input graph. More precisely, we use Rosen, Sun and Xue's quickest-path algorithm [14] as a framework giving bandwidth-stratified induced subgraphs. This approach is far more efficient than naively applying dynamic programming-based labeled path algorithms to bandwidth-augmented CFG-labeled graphs from algorithms such as Chen and Chin's [4]. Although, bandwidth-augmented graph algorithms, like Chen and Chin's, have merit for other applications of dynamic programming.

*Key–Words:* quickest path, context-free grammar, labeled graph, dynamic programming, algorithm design.

## 1 Introduction

Consider a digraph $G = (V, E)$ whose edges are labeled with the terminals of a CFG $\mathcal{G}$ and each edge has one of $r$ bandwidths $b_r > \cdots > b_1 > 0$. Let $\mathcal{G}$'s nonterminals be in the set $N$ and rules in the set $R$. Assume $\mathcal{G}$ is in Chomsky normal form. By combining a CFG labeled all-pairs shortest-path algorithm of Barrett, Jacob, and Marathe [1] with a quickest path algorithm of Rosen, Sun, and Xue [14] this paper gives an $O(r(|E| + |V|^3|N||R|))$ or $O(|E|^2 + |E||V|^3|N||R|)$ all-pairs quickest CFG-labeled path distances algorithm, since $r \leq |E|$. This assumes the edge weights or delays are non-negative.

For *DAGs (directed acyclic graphs)* labeled by Dyck and semi-Dyck CFGs [8] with a constant number of terminals, combining Rosen, *et al.*'s algorithm with Bradford and Choppella's [2] gives an $O(|E|^2 + |E||V|^\omega \log |V|)$ quickest path algorithm, where $\omega$ is the matrix multiplication exponent for multiplying a $|V| \times |V|$ by a $|V| \times |V|$ matrix. Bradford and Thomas [3] give a shortest path algorithm for graphs with positive and negative edge weights whose unlabeled versions have no negative cycles. They use a Johnson-style edge re-weighting and then apply Barrett, *et al.*'s $O(|V|^3|N||R|)$ algorithm to solve the all-pairs shortest path problem on this class of labeled graphs. Bradford and Thomas' algorithm has cost dominated by the $O(|V|^3|N||R|)$ time for invoking Barrett, *et al.*'s algorithm.

Our motivation is driven by a cryptographic constrained routing problem: In optimizing cryptographic routing, different cryptographic protocols generally do not commute. Let $m$ be the plaintext and $c$ be a ciphertext, then two cryptographic functions $\mathcal{E}$ and $\mathcal{E}'$ commute iff

$$\begin{aligned} c &= \mathcal{E}_{k_1}[\mathcal{E}'_{k_2}[m]] = \mathcal{E}'_{k_2}[\mathcal{E}_{k_1}[m]], \\ m &= \mathcal{D}_{k_1}[\mathcal{D}'_{k_2}[c]] = \mathcal{D}'_{k_2}[\mathcal{D}_{k_1}[c]], \end{aligned}$$

for all valid keys $k_1$ for $\mathcal{E}$ and all valid keys $k_2$ for $\mathcal{E}'$. Several public key protocols are commutative, for example two RSA public key systems using the same modulus commute [6, p. 8].

Of course, for every encryption there must be a symmetric decryption. This paper does not deal with

*An extended version of this paper is also being submitted to a Journal.

key distribution. Thus, we must choose a path in a network that suitably pairs encryption and decryption functions.

To model the cryptographic constrained routing problem consider graphs whose edges are labeled by open and closed parentheses as a way to model encryption and decryption in networks. See Figure 1, where dummy nodes are inserted between neighboring nodes to model encryption and decryption between neighbors. For this figure, note: triple DES, Twofish (TF) and AES have no known commutative properties. Thus, Figure 1(b) represents the case of non-commutative cryptographic functions that do not provide a suitable cryptographic connection.

Dyck and semi-Dyck languages are parenthesis languages [8] allowing several types of parentheses. Dyck languages allow reductions using both right-inverses and left-inverses where semi-Dyck languages are more restricted.

The Dyck language $\mathcal{D}$ and the semi-Dyck language $\acute{s}\mathcal{D}$ are both context-free languages derivable from the following grammars:

$$\begin{aligned} \acute{s}\mathcal{D} &\implies \epsilon \mid \acute{s}\mathcal{D}\ \acute{s}\mathcal{D} \mid a_i\ \acute{s}\mathcal{D}\ a_i^{-1} \\ \mathcal{D} &\implies \epsilon \mid \mathcal{D}\mathcal{D} \mid a_i\mathcal{D}a_i^{-1} \mid a_i^{-1}\mathcal{D}a_i, \end{aligned}$$

for all $i : \ell \geq i \geq 1$ where the terminals $a_i$ are open parentheses and $a_i^{-1}$ are close parentheses.

**Labeled Shortest-Path Algorithms.** In addition to other important results, Barrett, Jacob, and Marathe [1] gave two dynamic programming based CFG labeled digraph shortest-path algorithms. Their fastest algorithm costs $O(|V|^3|N||R|)$. Bradford and Choppella [2] find shortest paths on Dyck and semi-Dyck labeled DAGs in $O(|V|^\omega \log |V|)$ where $\omega$ is the exponent of the best $|V| \times |V|$ times $|V| \times |V|$ matrix multiplication. These two CFGs are assumed to have a constant number of terminals.

**Dynamic Programming Solutions of the Quickest Path Problem.** Different cryptographic network paths have different delays. For example, security-based delays may be caused by: negotiating crypto protocols, generating and exchanging keys, and even encryption and decryption may have different costs. Other sources of network delays include physical network distance, network media, and routing.

Chen and Chin [4] showed the principle of optimality fails for the quickest path problem since a subpath of a quickest path is not necessarily a quickest path. Thus direct applications of dynamic programming to quickest path problems does not work. Currently, quickest path algorithms either (1) augment instances of the quickest path problem into larger instances of shortest path problems–which are amenable to dynamic programming [4], or (2) solve the quickest path problem iteratively using bandwidth stratified graphs [14]. Here, each bandwidth stratified graph is amenable to dynamic programming and the solutions are easily combined.

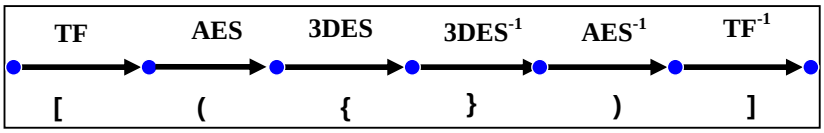
**The Quickest Path Problem and Labeled Shortest Paths.** There are numerous variants of the quickest path problem, see for example Pascoal, *et al.* [13]. Park, *et al.* [12] provide an overview of relevant algorithmic work on the quickest path problem.

Given a digraph $G = (V, E)$, Chen and Chin [4] give an $O(r(|E| + |V| \log |V|))$ solution to the quickest path problem with $r$ bandwidths. Their method converts an instance of the single-source quickest path problem into an augmented instance of the single-source shortest path problem. Solving this larger instance of the shortest path problem solves the given instance of the quickest-path problem. In particular, given a digraph $G = (V, E)$ with edge weight and bandwidth functions, their method constructs a new graph $G' = (V', E')$ where $|E'| = O(|E|^2)$ and $|V'| \leq |V||E| = O(|V|^3)$, see [4]. This means applying Barrett, *et al.*'s $O(|V|^3|N||R|)$ all-pairs CFG-labeled shortest path algorithm to $G'$ costs $O(|V|^9|N||R|)$. Also, Park, *et al.*'s [12] algorithm for solving the quickest-path problem gives similar complexity blow-up.

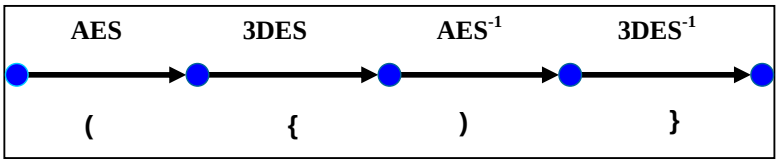
Applying Chen and Chin's single-source algorithm from each of the $|V|$ nodes gives an $O(|V||E|^2 + |V|^2|E| \log |V|)$ time all-pairs quickest-path algorithm. However, Lee and Papadopoulou [10] as well as Chen and Hung [5] both give $O(|E||V|^2)$ all-pairs quickest path algorithms. These algorithms use greedy methods which are naturally stratified by bandwidths.

**Assumptions.** All graphs in this paper have positive bandwidths and no self-loops. The graphs may have multiple edges between two nodes provided each edge has a different terminal labeling it.

This paper presents Rosen, *et al.*'s algorithm directly with Barrett, *et al.*'s algorithm. This is because Bradford and Thomas' algorithm directly uses Barrett, *et al.*'s algorithm and Barrett, *et al.*'s algorithm is simpler than Bradford and Choppella's.




**Part (a): Matching Cryptographic Protocols**




**Part (b): Protocols Not Properly Matching**

Figure 1: Two examples of parenthesis labeled paths, assuming the encryption and matching decryption functions have matching keys. We assume the keys are already distributed to the cryptographic functions. 3DES is triple DES, TF is Twofish and AES is the Advanced Encryption Standard, see [16].

## 2 Quickest CFG-Labeled Paths

A *labeled directed graph* (LDG) $G = (\Sigma, V, E)$ is a multigraph consisting of a set $V$ of vertices and an edge set

$$E \subseteq V \times V \times (\Sigma \cup \{\epsilon\})$$

of labeled, directed, and weighted edges. Say there is an edge from node $u$ to node $v$ with label $t$, then the weight of this edge is $w(u, v, t)$ and this weight may be called its delay or lead time. Given $e = (u, v, t) \in E$, then the label function is $l(e) = t$.

An LDG $G$ is a multigraph in that it may have several edges (all labeled differently) between any two nodes. In particular, take two edges, both from $u$ to $v$: $e_1 = (u, v, t_1)$ and $e_2 = (u, v, t_2)$. If $t_1 = t_2$, then $w(u, v, t_1) = w(u, v, t_2)$ and $b(u, v, t_1) = b(u, v, t_2)$. That is, if $t_1 = t_2$, then $e_1$ and $e_2$ are the same edge between $u$ and $v$.

A context-free grammar $\mathcal{G} = (N, \Sigma, R, A_s)$ contains the start symbol $A_s$ and a set of: nonterminals $N$, terminals $\Sigma$, and rules $R$, see for example [8]. Let $\epsilon \notin \Sigma$, be the empty symbol and assume any context-free grammar is in Chomsky normal form. Given a CFG $\mathcal{G}$, a string of terminals $s$ is derivable from $\mathcal{G}$ iff $s \in L(\mathcal{G})$.

**Definition 1 (Labeled Paths)** Given a LDG $G = (\Sigma, V, E)$ with CFG $\mathcal{G}$. Let $p : v_i \overset{A}{\rightsquigarrow} v_j$ indicate there is a path $p$ from node $v_i$ to node $v_j$ in $G$ whose labeled string $l(p) = l(v_i, v_{i+1}) \cdots l(v_{j-1}, v_j)$ can be derived by starting from the non-terminal $A$ of $\mathcal{G}$.

**Definition 2 (Labeled Directed Graph Problem)** Given a LDG $G = (\Sigma, V, E)$ and its associated CFG $\mathcal{G}$ having start symbol $A_s$, then for every pair of vertices $(u, v) \subseteq V \times V$ find the shortest labeled path $p : u \overset{A_s}{\rightsquigarrow} v$ so that $l(p) \in L(\mathcal{G})$.

Barrett, *et al.*'s $O(|V|^3|N||R|)$ algorithm is given in Figure 3. Its initialization function is in Figure 2.

Barrett, *et al.*'s algorithm [1] assumes a heap $H$ containing the triples $D = V \times V \times N$. $D(u, v, A)$ starts as an approximation of the shortest path distances of a path $u \overset{A}{\rightsquigarrow} v$ where $A$ is a non-terminal. Note in CNF each terminal is at the end of a rule coming from a non-terminal. When $D(u, v, A)$ is put in $S$ then $D(u, v, A)$ is the shortest labeled-path distance from $u$ to $v$. $P(u, v, A)$ is a pointer to heap element $D(u, v, A)$.

### 2.1 Extending the Labeled Path Problem

An edge from node $u$ to node $v$ with label $t$ has bandwidth $b(u, v, t) > 0$. Let $\sigma$ be the units of data to be transmitted, then the quickest path problem assigns the total weight $w(p)$ to the path $p = v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_k$ as follows:

// Given an LDG $G = (\Sigma, V, E)$ where the grammar is in CNF.
1. Initialize-Matrix_D($G$)
2. **for all** pairs $(u, v) \in V \times V$ **do**
3. **for all** nonterminals $A \in N$ **do**
4. $D(u, v, A) \leftarrow \infty$
5. **for all** vertices $v \in V$ **do**
6. $D(v, v, A_s) \leftarrow 0$, where $A_s$ is the start nonterminal
7. **for all** edges $(u, v) \in E$ **do**
8. **for all** productions $A \rightarrow t$ where $t \in \Sigma$ **do**
9. **if** $l(u, v) = t$ **then**
10. $D(u, v, A) \leftarrow D(u, v, t) \leftarrow w(u, v, t)$
11. Return $D$

Figure 2: Initializing Barrett, Jacob, and Marathe's All-Pairs Shortest CFG Labeled Path Algorithm.

$$w(p) \quad = \quad d(p) + \frac{\sigma}{b(p)}$$

where for the path $p$ the total delay is:

$$d(p) \quad = \quad \sum_{i=0}^{k-1} d(v_i, v_{i+1})$$

the path $p$ has the label

$$l(p) \quad = \quad l(v_0, v_1) \ \cdots \ l(v_{k-1}, v_k),$$

where juxtaposition of edge terminals represents concatenation and the bandwidth of the path $p$ is:

$$b(p) \quad = \quad \min_{k-1 \geq i \geq 0} \{ \, b(v_i, v_{i+1}) \, \}.$$

Given a LDG $G = (\Sigma, V, E)$ associated CFG $\mathcal{G}$ and delay function $d$ and bandwidth constraint function $b$. Then for any pair of vertices $(v_0, v_k) \in V \times V$ and for the start non-terminal $A_s$ from $\mathcal{G}$,compute the minimal CFG-constrained path distance:

$$T(v_0, v_k, \sigma) \quad = \quad \min_{p: v_0 \overset{A_s}{\rightsquigarrow} v_k} \left\{ \, d(p) + \frac{\sigma}{b(p)} \, \right\},$$

so $l(p) \in L(\mathcal{G})$. The quickest path problem for labeled directed graph problem is:

**Definition 3 (All-Pairs Quickest Labeled Path Distances Problem)**
Given $\sigma$ units of data to be transmitted in an LDG problem $G = (\Sigma, V, E)$ and its associated CFG $\mathcal{G}$ with start symbol $A_s$ and each edge has a bandwidth from $b_r > \cdots > b_1 > 0$. Then the *all-pairs quickest labeled path distances problem* is: For all pairs of nodes $(u, v) \in V \times V$ and for all paths $p : u \overset{A_s}{\rightsquigarrow} v$ such that $l(p) \in L(\mathcal{G})$, then find the minimal path distance $T(u, v, \sigma)$.

## 2.2 Rosen, *et al.*'s Quickest-Path Algorithm

Rosen, Sun, and Xue solve the quickest path problem by stratifying the input graph by bandwidths.

**Definition 4 (Rosen, *et al.* [14]'s Graph Stratification, see also [10])**
Given $G = (V, E)$ along with edge delays $d$ and bandwidth constraints $b$. Assume the bandwidths of the edges of $G$ are $b_r > \cdots > b_1 > 0$. Let $G(b_i) = (V, E(b_i))$ where $(u, v) \in E(b_i)$ iff $(u, v) \in E$ and $b(u, v) \geq b_i$.

The next theorem gives cases where the principle of optimality holds in bandwidth-stratified induced subgraphs of the input graph.

**Theorem 5 (Rosen, Sun, and Xue [14])** Given $\sigma$ units of data to transmit through a digraph $G = (V, E)$ with edge delay function $d$ and bandwidth constraint function $b$.

Let $p$ be a quickest path from $s$ to $t$ in $G$. Then

1. $p$ is a shortest path in $G(b(p))$ from $s$ to $t$ considering only the delay function $d$,
2. any sub-path of $p$ is a shortest path in $G(b(p))$ considering only the delay function $d$.

```
1. Fast_BJM(G) // where G = (Σ, V, E)
2. Initialize-Matrix_D(G)
3. Initialize-Heap_H(G)
4. S ← ∅
5. while H ≠ ∅ do
6.     D(u, v, X) ← extractMin(H)
7.     S ← S ∪ { D(u, v, X) }
8.     for all productions of the form A → BC do
9.         if B = X then
10.            for all vertices v_0 ∈ V do
11.                L ← P(u, v_0, A)
12.                val ← D(u, v, B) + D(v, v_0, C)
13.                if val < D(L) then
14.                    decreaseKey(H, L, val)
15.        if C = X then
16.            for all vertices u_0 ∈ V do
17.                L ← P(u_0, v, A)
18.                val ← D(u_0, u, B) + D(u, v, C)
19.                if val < D(L) then
20.                    decreaseKey(H, L, val)
21. Return D
```

Figure 3: Barrett, Jacob, and Marathe's $O(|V|^3|N||R|)$ All-Pairs Shortest CFG Labeled Path Algorithm [1].

Recall this paper is focused on shortest-path distances and not actually producing the shortest paths themselves.

**Theorem 6 (Rosen, Sun, and Xue [14])** Given a digraph $G = (V, E)$ along with edge delays $d$ and bandwidth constraints $b$. Let $p_j$ be a shortest path from $s$ to $t$ in $G(b_j)$, for all $b_r > \cdots > b_1 > 0$, only considering the delay function $d$ given $\sigma$ units of data. If

$$d(p_l) + \frac{\sigma}{b(p_l)} \quad = \quad \min_{r \geq j \geq 1} \left\{ d(p_j) + \frac{\sigma}{b(p_j)} \right\}$$

then $p_l$ is a quickest path from $s$ to $t$ in $G$ for $\sigma$ units of data.

## 3 Rosen, *et al.* as a Framework

Given a LDG $G = (\Sigma, V, E)$ with delays $d$ and bandwidths $b_r > \cdots > b_1 > 0$. The next definition is a trivial generalization of Rosen, *et al.*'s Definition 4.

**Definition 7 (See Rosen, *et al.*'s Definition 4.)** Given an LDG $G = (\Sigma, V, E)$ along with edge delays $d$ and bandwidth constraints $b_r > \cdots > b_1 > 0$. Let $G(b_i) = (\Sigma, V, E(b_i))$ where $(u, v) \in E(b_i)$ iff $(u, v) \in E$ and $b(u, v) \geq b_i$.

The principle of optimality also holds for shortest CFG labeled paths as is shown in the next lemma. This lemma is a generalization of Rosen, *et al.*'s Theorem 5.

**Lemma 8** Given $\sigma$ units of data, consider the problem of finding CFG labeled shortest paths in LDGs $G = (\Sigma, V, E)$ with CFG $\mathcal{G}$ along with edge delay function $d$ and bandwidths $b_r > \cdots > b_1 > 0$.

Let $p$ be a quickest labeled path from $s$ to $t$ in $G$. Then

1. $p$ is a shortest labeled path in $G(b(p))$ from $s$ to $t$ considering only the delay function $d$,
2. any labeled sub-paths $p_1 = s \overset{B}{\rightsquigarrow} k$ and $p_2 = k \overset{C}{\rightsquigarrow} t$ of $p = s \overset{A}{\rightsquigarrow} t$ are both shortest labeled paths in $G(b(p))$ considering only the delay function $d$ provided $A \rightarrow BC$ is a rule of $\mathcal{G}$.

1. RSX($G, \sigma$)
   // where $G = (V, E)$ with delay function $d$ and bandwidths $b_r > \cdots > b_1 > 0$
2. **for** $j \leftarrow 1$ **to** $r$ **do**
3. **for all** $s, t : |V| \geq s, t \geq 1$ **do**
4. Compute a shortest path $p$'s distance $d_j(s,t)$ from $s$ to $t$ in $G(b_j)$
   considering only the delays $d$
   Store the minimal bandwidth in $p$ between $s$ and $t$ in $b_j(s,t)$
5. **for all** $s, t : |V| \geq s, t \geq 1$ **do**
6. $k \leftarrow 1$; $m \leftarrow d_1(s,t) + \frac{\sigma}{b_1(s,t)}$
7. **for** $j \leftarrow 2$ **to** $r$ **do**
8. **if** $m > d_j(s,t) + \frac{\sigma}{b_j(s,t)}$ **then**
9. $m \leftarrow d_j(s,t) + \frac{\sigma}{b_j(s,t)}$
10. Quickest-Distance$(s,t) \leftarrow m$

Figure 4: Rosen, Sun, and Xue's Quickest Path Algorithm [14]: **RSX**.

**Proof:** In the first part of the proof follows that of Rosen, *et al.*'s Theorem 5 very closely. In particular, let $p$ be a quickest labeled path from $s$ to $t$ in $G$ and suppose $q$ is any shortest labeled path from $s$ to $t$ in $G(b(p))$. First, $b(q) \geq b(p)$ because $q$ is in $G(b(p))$, and see [14]. Moreover,

$$d(p) + \frac{\sigma}{b(p)} \quad \leq \quad d(q) + \frac{\sigma}{b(q)},$$

but this means $d(q) \geq d(p)$, thus $p$ is a shortest path in $G(b(p))$.

In the second part of the proof, given a LDG $G$ and CFG $\mathcal{G}$, the labeled path problem exhibits the following variation of the principle of optimality: Let $A$ be a non-terminal from $\mathcal{G}$.

Given any shortest labeled path $p = u \overset{A}{\rightsquigarrow} v$ in $G$, then any labeled subpath of $p$ whose terminals are derivable from $A$ is also a shortest labeled path.

There are two cases:

**Case 1:** $A \rightarrow a$**.** Assuming $a \in \Sigma$.

In this case, $u \overset{a}{\rightsquigarrow} v$ is a single edge $(u, v, a)$ and trivially, it satisfies the principle of optimality.

**Case 2:** $A \rightarrow BC$**.** Assuming $\{B, C\} \subseteq N$.

In this case, there must be some node $k \in V$ so that $u \overset{B}{\rightsquigarrow} k$ and $k \overset{C}{\rightsquigarrow} v$. Where both of these paths must also be labeled shortest paths.

This completes the proof. ∎

Rosen, *et al.*'s Theorem 6 also generalizes for labeled quickest paths to give:

**Theorem 9** Given a LDG $G = (\Sigma, V, E)$ with associated CFG $\mathcal{G}$ along with edge delays $d$ and bandwidth constraints $b$. Let $p_j$ be a shortest labeled path from $s$ to $t$ in $G(b_j)$, for all $b_r > \cdots > b_1 > 0$, only considering the delay function $d$ given $\sigma$ units of data. If

$$d(p_l) + \frac{\sigma}{b(p_l)} \quad = \quad \min_{r \geq j \geq 1} \left\{ d(p_j) + \frac{\sigma}{b(p_j)} \right\}$$

then $p_l$ is a quickest labeled path from $s$ to $t$ in $G$ for $\sigma$ units of data.

The proof is a straight forward generalization of Rosen, *et al.*'s proof.

If $d_r(s,t) > \cdots > d_1(s,t)$, then all quickest-path distance candidates from $s$ to $t$ are $d_i(s,t) + \frac{\sigma}{b_i}$, for all $i : r \geq i \geq 1$. This is because, in this case the delays are strictly decreasing from $d_r$ down to $d_1$ while the sequence $\frac{\sigma}{b_r} < \cdots < \frac{\sigma}{b_1}$ is strictly increasing.

**Lemma 10** Consider Rosen, *et al.*'s algorithm (Figure 4) given $\sigma$ data units and a digraph $G = (V, E)$ along with edge delay function $d$ and bandwidths $b_r > \cdots > b_1 > 0$. Then without loss, we may assume $d_{i+1}(s,t) > d_i(s,t)$ for all $i : r > i \geq 1$.

**Proof:** Let $p_{i+1}$ be a shortest path with delay $d_{i+1}(s,t)$ and let $p_i$ be a shortest path with delay $d_i(s,t)$. All edges of $p_{i+1}$ have bandwidths of $b_{i+1}$ or larger and all edges of $p_i$ have bandwidths of $b_i$ or larger.

| Replace lines 3-5 of RSX | Cost of Combined Algs | Significant Details |
|---|---|---|
| Barrett, *et al.* [1]'s $O(\|V\|^3\|N\|\|R\|)$ | $O(\|E\|^2 + \|E\|\|V\|^3\|N\|\|R\|)$ | Non-neg. Edge Delays |
| Bradford and Choppella [2] | $O(\|E\|^2 + \|E\|\|V\|^\omega \log \|V\|)$ | Dyck and Semi-Dyck Langs. on DAGs with a constant number of symbols |
| Bradford and Thomas [3] | $O(\|E\|^2 + \|E\|\|V\|^3\|N\|\|R\|)$ | Neg or Positive edge Delays but no negative cycles in the unlabeled version of the graph |

Figure 5: Summary of All-Pairs Context-Free Grammar Labeled Shortest Path Distance Results

If $d_{i+1}(s,t) > d_i(s,t)$ then there must be at least one bandwidth $b_i$ edge in $p_i$ since the only difference between $G(b(p_i))$ and $G(b(p_{i+1}))$ is $G(b(p_i))$ can have bandwidth $b_i$ edges. This is true even if all of $p_{i+1}$'s edges have larger bandwidths than $b_{i+1}$.

If $d_r(s,t) = d_{r-1}(s,t)$ and since $b_r > b_{r-1}$, then it must be that

$$d_r(s,t) + \frac{\sigma}{b_r} \quad < \quad d_{r-1}(s,t) + \frac{\sigma}{b_{r-1}}.$$

Thus, we can discard $d_{r-1}(s,t)$ as contributing to a quickest path. Note, $d_{i+1}(s,t) < d_i(s,t)$ is impossible, since $E(b_{i+1}) \subseteq E(b_i)$. Thus, given $d_r(s,t), \cdots, d_1(s,t)$, consider the next cases:

**Case 1:** $d_r(s,t) = \cdots = d_{i+1}(s,t) > d_i(s,t)$**.** In this case, only $d_r(s,t) + \frac{\sigma}{b_r}$ and $d_i(s,t) + \frac{\sigma}{b_i}$ can give quickest path distances between $s$ and $t$. The values $d_{r-1}(s,t)$ through $d_{i+1}(s,t)$ cannot give quickest path distances.

**Case 2:** $d_r(s,t) > \cdots > d_{i+1}(s,t) = d_i(s,t)$**.** In this case, the quickest path distances from $s$ to $t$ may be any of $d_j(s,t) + \frac{\sigma}{b_j}$ for $j : r \geq j \geq i+1$ since $d_i(s,t)$ cannot contribute to a quickest path distance.

In either of the aforementioned cases, the subsequent cases complete the proof:

**Case A:** $d_i(s,t) = d_{i-1}(s,t)$**.** Here $d_{i-1}(s,t)$ cannot contribute to a quickest path distance. Since there is some $j : r - i \geq j \geq 0$ so $d_{i+j}(s,t) = d_{i-1}(s,t)$ where $d_{i+j}(s,t) + \frac{\sigma}{b_{i+j}}$ may be a quickest path distance and $b_{i-1} < b_i \leq b_{i+j}$.

**Case B:** $d_i(s,t) > d_{i-1}(s,t)$**.** Here we must consider $d_{i-1}(s,t) + \frac{\sigma}{b_{i-1}}$. Moreover, $d_i(s,t) > d_{i-1}(s,t)$ indicates there is a bandwidth $b_{i-1}$ edge in a path of delay $d_{i-1}(s,t)$ in $G(b_{i-1})$.

Cases A and B may be repeatedly applied to determine which bandwidths match with which delays for computing all path distances. ∎

Lemma 10 immediately gives an algorithm to find quickest-paths given all delays and bandwidths and Theorem 6.

The results given in this subsection immediately give:

**Theorem 11** Given $\sigma$ units of data to transmit through an LDG $G = (\Sigma, V, E)$ along with edge delays $d$ and bandwidth constraints $b$. Then the results given in Figure 5 hold.

# 4 Conclusion

This paper shows different methodological applications of dynamic programming have a substantial impact on the efficiency of the solutions. In particular, direct applications of dynamic programming to expanded input structures may make dynamic programming more expensive than applying dynamic programming iteratively to substructures, then combining the solutions of these substructures.

Future directions include: a distributed version of quickest labeled path algorithms for large distributed systems. First we would have to find a distributed solution to the CFG labeled path problem. However, there are several distributed quickest path algorithms [9, 15].

Also, building an efficient single-source labeled shortest path algorithm would allow us to build a single-source shortest path CFG-labeled quickest path algorithm.

**Acknowledgement:** Thanks to Yang Xiao for suggested modeling networks of crypto protocols using labeled path problems.

*References:*


[1] C. Barrett, R. Jacob, M. Marathe: "Formal Language Constraint Path Problems," *SIAM Journal on Computing*, **30(3)**, 809-837, 2000.

[2] P. G. Bradford and V. Choppella: "Fast Dyck and Semi-Dyck Constrained Shortest Paths on DAGs," *submitted*.
**Presented:** Colloquium on 6 July 2005: "Fast Dyck Constrained Shortest Paths," The University of Alabama at Birmingham, Birmingham AL.

[3] P. G. Bradford and D. A. Thomas: "Labeled Shortest Paths in Digraphs with Negative and Positive Edge Weights," *submitted*.
**Presented:** Research Meeting on 6 July 2007: "Labeled Shortest Paths in Digraphs with Negative and Positive Edge Weights," The University of Alabama, Tuscaloosa, AL.

[4] Y. L. Chen and Y. H. Chin. "The Quickest Path Problem," *Computers & Operations Research*, 17(2), 153 - 161, 1990.

[5] G.-H. Chen and Y.-C. Hung: "On the quickest path problem," *Information Processing Letters*, **46**, 125-128, 1993.

[6] V. Cortier, S. Delaune, and P. Lafourcade: "A survey of algebraic properties used in cryptographic protocols," *Journal of Computer Security*, 14(1), 1-43, 2006.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein: *Introduction to Algorithms*, 2nd Edition, MIT Press, 2001.

[8] J. E. Hopcroft and J. D. Ullman: *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[9] Y.-C. Hung and G.-H. Chen: "Distributed algorithms for the quickest path problem," *Parallel Computing*, **19**, 823-834, 1992.

[10] D. T. Lee and E. Papadopoulou: "The all-pairs quickest path problem," *Information Processing Letters*, **45**, 261-267, 1993.

[11] M. H. Moore: "On the Fastest Route for Convoy-Type Traffic in Flowrate-Constrained Networks," *Transportation Science*, **10(2)**, 113-124, May 1976.

[12] C.-K. Park, S. Lee and S. Park: "A label-setting algorithm for finding a quickest path," *Computers & Operations Research*, **31(14)**, 2405-2418, December 2004.

[13] M. M. B. Pascoal, M. E. V. Captivo, and J. C. N. Clímaco: "A comprehensive survey on the quickest path problem," *Annals of Operations Research*, 147(1), 5-21, 2006.

[14] J. B. Rosen, S. Sun, G. Xue: "Algorithms for the quickest path problem and the enumeration of quickest paths," *Computers & Operations Research*, **18(6)**, 579-584, 1991.

[15] J. Ben Rosen, Guoliang Xue: "Sequential and Distributed Algorithms for the All Pairs Quickest Path Problem," *ICCI 1991*, Lecture Notes in Computer Science, 471-473.

[16] Douglas R. Stinson: *Cryptography, Theory and Practice*, Third Edition, CRC Press, 2005.