Efficient algorithms for decode efficient prefix codes

Shashwat Banchhor, Rishikesh R. Gajjala
Yogish Sabharwal, Sandeep Sen

Why are decode efficient prefix codes important?

- Data compression techniques focus on achieving maximum compression but there is an inherent cost to encode and decode(decompress).

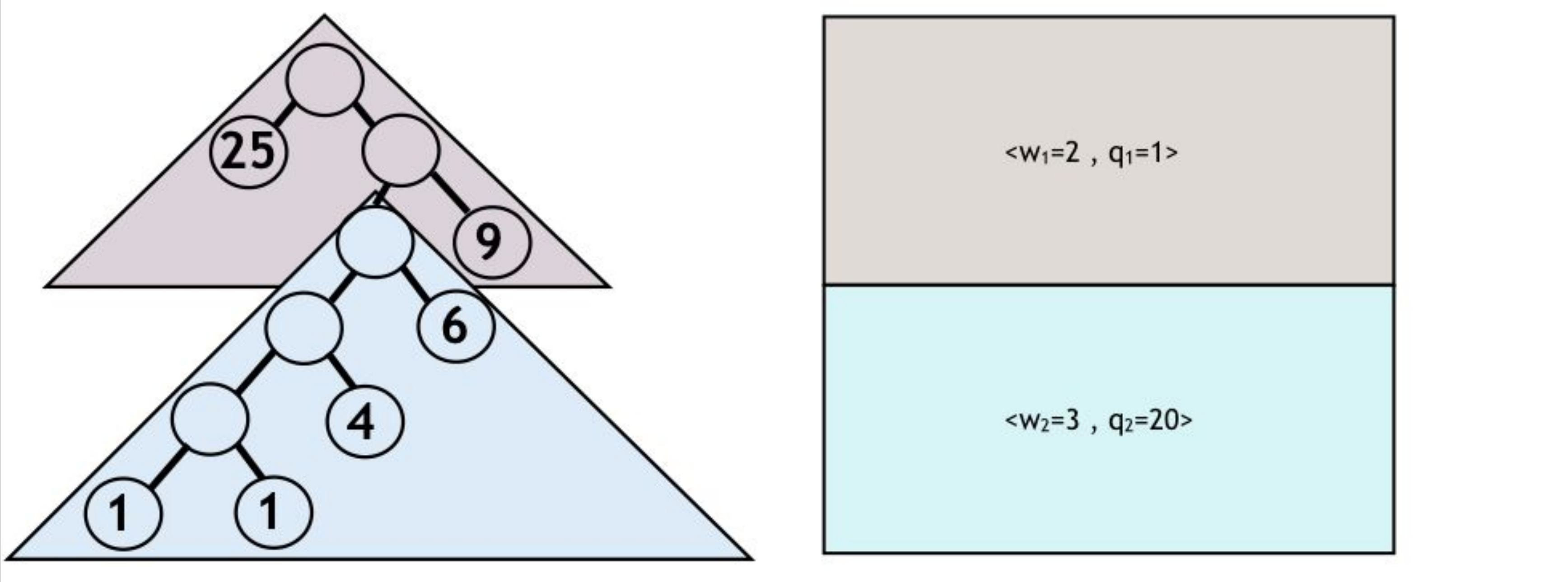
- Encoding is done once but decoding is done multiple times.

- The cost to decode can be very high for certain real time applications.
  - Eg: Inference from deep learning models

- This can reduced by using fast memories but proper consideration of this hasn't been done.

Memory model - Blocking Scheme

A blocking scheme of m block levels is a sequence of m block parameters < $(w_1,q_1),(w_2,q_2), .. ,(w_m,q_m)$ > wherein w_i represents the number of bits required to access the block and q_i represents the access cost of the block.



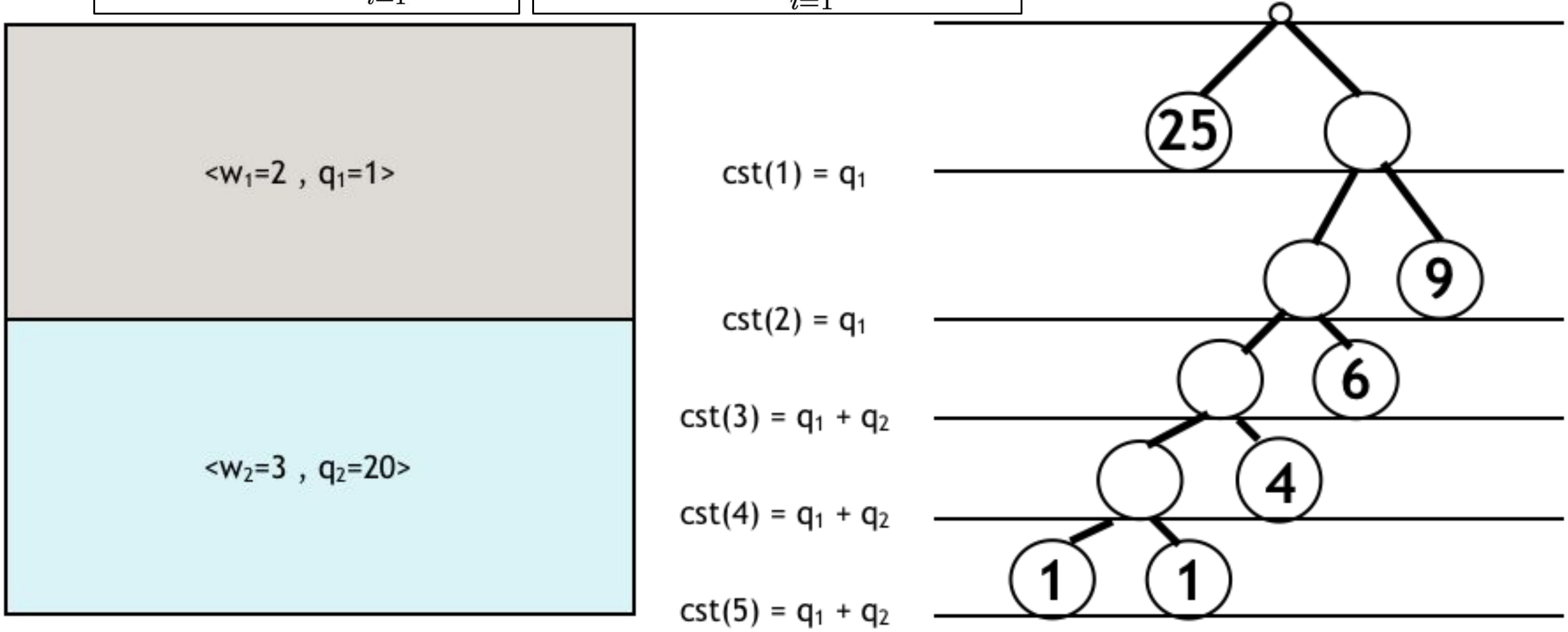
a) Prefix Tree stored in *BS* b)

b) *BS :* <(2,1), (3,20)>

Blocking Scheme viewed as Cost function(cst)

- Given *BS*: For any character c_i at depth l_i in the tree stored in the *BS,* the access cost of c_i is **cst(l_i)**

Code Length: $\sum_{i=1}^{n} l_i \cdot f_i$	Decode Time: $\sum_{i=1}^{n} cst(l_i) \cdot f_i$



Problem Definition

Given input parameter ***L***, alphabet *C* with **n characters** (s.t. any character c_i has frequency f_i) and a non-decreasing **cost function *cst*** *s.t.* the cost to access a character at depth l_i is $cst(l_i)$. Find depth $l_1, l_2, ..., l_{n-1}, l_n$ corresponding to each character:

$$\textbf{Minimize} \sum_{i=1}^{n} cst(l_i) \cdot f_i$$

(Decode Time)

$$\textbf{s.t.} \sum_{i=1}^{n} l_i \cdot f_i \leq L$$

(Codelength Constraint)

$$\textbf{and} \sum_{i=1}^{n} 2^{-l_i} \leq 1$$

(Kraft’s Inequality - ensures valid prefix tree)

We call this the DOPT(*L*) problem (Decode Optimum).

Our Contribution

- A dynamic program to solve DOPT(L) problem in $O(n^3.L)$ time.

- An approximation algorithm to find a prefix tree having code length at most $(1+\epsilon).L$ and decode time at most the decode time of the optimum solution of DOPT(L) in $O(n^4/\epsilon)$ time.