Column-Width Optimization

Quinten Kent

Definitions

An instance of the Column-Width Optimization (COL-OPT) problem consists of the following:

- A row by column ($r \times c$) matrix T, where each cell in T is empty or holds a sequence of characters
- A $c \times c$ constraint matrix C, where each cell is C holds a non-negative integer.

A solution to COL-OPT is given by the c-length vector $\vec{w}$, where $\vec{w}$:

- Gives the widths for each column in T such that the length of table T is minimal. The length of T is given by the sum of the length of each row. The length of a row is given by the maximum value of a cell divided by its assigned column width for all cells in the row.
  $len(T) = \sum_{i=1}^{r} \max_{j=1}^{c} \lceil len(T[i][j]) \,/\, \vec{w}[j] \rceil$
- Assigns each column a width of at least 1.
  for $1 \leq i \leq c$, $\vec{w}[i] > 0$.
- Satisfies constraint matrix C so that if $C[i][j]$ is non-zero, the sum of all column weights from i to j is equal to $C[i][j]$.
  for $1 \leq i \leq c$, $i \leq j \leq c$, if $C[i][j]$ is non-zero, $\sum_{k=i}^{j} \vec{w}[k] = C[i][j]$.

From this definition of $\vec{w}$, constraint matrix C must give a valid total width for the table. $C[1][c] \geq c$.

Define the "3-conjunctive normal form satisfiability" problem (3-CNF-SAT) as defined in CLRS [Introduction to Algorithms, 3rd ed., p. 1082].

3-CNF-SAT defines:

- A **literal** in a boolean formula is an occurrence of a variable or its negation.
- A boolean formula is in **conjunctive normal form**, or **CNF**, if it is expressed as an AND of **clauses**, each of which is the OR of one or more literals.
- A boolean formula is in **3-conjunctive normal form**, or **3-CNF**, if each clause has exactly three distinct literals.

An instance of 3-CNF-SAT consists of a boolean formula ϕ.

Theorem

Optimization of column widths for a table given by COL-OPT is NP-hard.

Proof

Reduction from 3-CNF-SAT to COL-OPT

Prove that COL-OPT is NP-hard by reducing 3-CNF-SAT to it.

Let $\tau(\phi)$ be a reduction from 3-CNF-SAT to COL-OPT.

Given m clauses of n unique variables in ϕ, construct a $7m \times 2n$ table T. Interpret the $2n$ columns as representing each possible literal. Construct the rows of the table in multiples of seven for each clause m_l in ϕ, where each clause m_l consists of three boolean variables.

With three boolean variables, there are $2^3 = 8$ possible truth assignments. Of the eight possible assignments, all but one will satisfy m_l. Construct a row for each of the seven truth assignments which satisfy m_l, where a cell in the table with length 2 represents a truth assignment and length 1 a false assignment. All variables not represented in clause m_l have length 1.

For example, given the clause $(x_1 \vee x_2 \vee \neg x_3)$, construct the following 7 rows from the corresponding truth assignments:

	x_1	$\neg x_1$	x_2	$\neg x_2$	x_3	$\neg x_3$	$\dots$	x_n	$\neg x_n$	$< x_1, x_2, x_3 >$
1	2	1	2	1	2	1		1	1	$< T, T, T >$
2	2	1	2	1	1	2		1	1	$< T, T, F >$
3	2	1	1	2	2	1		1	1	$< T, F, T >$
4	2	1	1	2	1	2		1	1	$< T, F, F >$
5	1	2	2	1	2	1		1	1	$< F, T, T >$
6	1	2	2	1	1	2		1	1	$< F, T, F >$
	1	2	1	2	2	1		1	1	$< F, F, T >$
7	1	2	1	2	1	2		1	1	$< F, F, F >$

Note that the row which is excluded is the row which corresponds to the assignment $< x_1, x_2, x_3 > = < F, F, T >$, since this is the only assignment which fails to satisfy the clause.

Construct the constraint matrix C starting with a $2n \times 2n$ zero matrix. Starting with $i = 1$, for $i < 2n$ and increasing i in increments of 2, let $C[i][i+1] = 3$. Let $C[1][2n] = 3n$.

This produces the matrix:

$$C = \begin{array}{c} \\ x_1 \\ \neg x_1 \\ x_2 \\ \neg x_2 \\ \vdots \\ x_n \\ \neg x_n \end{array} \begin{array}{c} \begin{array}{ccccccc} x_1 & \neg x_1 & x_2 & \neg x_2 & \dots & x_n & \neg x_n \end{array} \\ \left(\begin{array}{ccccccc} & 3 & & & & & 3n \\ & & 0 & & & & \\ & & & 3 & & & \\ & & & & \ddots & & \\ & & & & & 0 & \\ & & & & & & 3 \\ & & & & & & \end{array} \right) \end{array}$$

Together, the matrices T and C give an instance of the COL-OPT problem. This describes the reduction τ in full. Furthermore, it is evident that $\tau(\phi)$ is polynomial, since m and n bound the size of ϕ and producing the $7m \times 2n$ table T is polynomial in terms of m and n.

Proof of Equivalence

To prove that τ is a valid reduction, let T and C be an instance of COL-OPT produced by $\tau(\phi)$, where ϕ is an instance of 3-CNF-SAT. Let $\vec{w}$ be the vector of column weights subject to C that minimizes the length of T.

Following the construction of T, $\vec{w}$ consists of n pairs of weights. Subject to C, for each pair of weights there are two possible assignments: $< 2, 1 >$ or $< 1, 2 >$. This forces each column pair, representing a variable and its negation, to be a valid truth assignment.

Consider the cost function $len(T)$ that $\vec{w}$ minimizes. Following the construction of T, each sequence of 7 rows corresponds to a clause m_l in ϕ, and will have cells with length 2 restricted to exactly 3 column pairs.

Since every column must have a width of at least 1, cells with length 1 have no effect on the cost function for a row. Therefore, for every 7 rows we can compute the cost entirely in terms of the 3 column pairs represented by the 3 variables in m_l.

Suppose the truth assignment given by $\vec{w}$ satisfies the clause. Following the construction of T, this assignment must be represented as one of the rows in the sequence. The row which matches the truth assignment will have length 1, while all other rows have length 2, giving the cost: $1 + 2 \cdot 6 = 13$.

On the other hand, suppose this truth assignment does not satisfy the clause. In this case, the assignment is not represented in one of the seven rows in T, so all rows have length 2, giving the cost $2 \cdot 7 = 14$.

For $1 \leq l \leq m$, let $r_l = 7(l-1)$ give the row index for each sequence of 7 rows. For all r_l,

$$len(T) \text{ from rows } (r_l + 1) \text{ to } (r_l + 7) = \begin{cases} 13 & \text{if clause } m_l \text{ is satisfied} \\ 14 & \text{if clause } m_l \text{ is } \textit{not} \text{ satisfied} \end{cases}$$

Note that computing the cost for each clause's rows is independent of all other clauses.

Therefore, if $\vec{w}$ minimizes the length of table T globally, it also minimizes the number of clauses which are dissatisfied. Conversely, it maximizes the amount of satisfied clauses.

This gives the result that if ϕ is satisfiable, the length of T will be exactly $13m$, since every m sequence of 7 rows must have length 13. If the length of $T > 13m$, ϕ is not satisfiable, since at least one clause was not satisfied.

$\therefore$ if $\phi \in$ 3-CNF-SAT, $\tau(\phi) \in$ COL-OPT.

Conversely, suppose $\tau(\phi) \in$ COL-OPT.

Following the definition of τ and COL-OPT, the weights $\vec{w}$ which minimize T will give a length such that $13m \leq len(T) \leq 14m$.

If $len(T) = 13m$, then each sequence of 7 rows must have length 13. Given each sequence of 7 rows, the clause that would produce this sequence can be determined following the construction in τ. Furthermore this clause will be satisfied with the truth assignment given by $\vec{w}$.

$\therefore$ if $\tau(\phi) \in$ COL-OPT, $\phi \in$ 3-CNF-SAT.

Therefore, since a polynomial reduction from 3-CNF-SAT to COL-OPT exists, COL-OPT is NP-hard.

□