طراحی الگوریتم

۲۸ آبان ۹۸

ملکی مجد

Topic	Reference
Recursion and Backtracking	Ch.1 and Ch.2 JeffE
Dynamic Programming	Ch.3 JeffE and Ch.15 CLRS
Greedy Algorithms	Ch.4 JeffE and Ch.16 CLRS
Amortized Analysis	Ch.17 CLRS
Elementary Graph algorithms	Ch.6 JeffE and Ch.22 CLRS
Minimum Spanning Trees	Ch.7 JeffE and Ch.23 CLRS
Single-Source Shortest Paths	Ch.8 JeffE and Ch.24 CLRS
All-Pairs Shortest Paths	Ch.9 JeffE and Ch.25 CLRS
Maximum Flow	Ch.10 JeffE and Ch.26 CLRS
String Matching	Ch.32 CLRS
NP-Completeness	Ch.12 JeffE and Ch.34 CLRS

Linear programming

to optimize a linear function subject to a set of linear inequalities

We focus on

a **special case** that can be reduced to finding shortest paths from a single source

Linear programming - definition

In the general ***linear-programming problem***,

we are given an ***m*** $\times$ ***n* matrix** A, an ***m*-vector** b, and an ***n*-vector** c.

We wish to find a **vector *x*** of n elements that

maximizes the ***objective function*** $\sum_{i=1}^{n} c_i x_i$

subject to the ***m* constraints** given by

$$Ax \leq b$$

General solution

- There are linear programming algorithms that do run in polynomial time

- There are many special cases of linear programming for which faster algorithms exist.
  - the single source shortest-paths problem is a special case of linear programming

feasibility problem

- Not care about the objective function
- Find any ***feasible solution***,

Systems of difference constraints

- each row of the linear-programming matrix A contains one 1 and one -1, and all other entries of A are 0.
- the constraints given by $Ax \leq b$ are a set of m ***difference constraints*** involving n unknowns

example

$$
\begin{pmatrix}
1 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & -1 \\
-1 & 0 & 1 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & -1 & 1
\end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}
\le
\begin{pmatrix} 0 \\ -1 \\ 1 \\ 5 \\ 4 \\ -1 \\ -3 \\ -3 \end{pmatrix}
$$

$$
\begin{aligned}
x_1 - x_2 &\le 0\,, \\
x_1 - x_5 &\le -1\,, \\
x_2 - x_5 &\le 1\,, \\
x_3 - x_1 &\le 5\,, \\
x_4 - x_1 &\le 4\,, \\
x_4 - x_3 &\le -1\,, \\
x_5 - x_3 &\le -3\,, \\
x_5 - x_4 &\le -3\,.
\end{aligned}
$$

$$x = (-5, -3, 0, -1, -4)$$

Number of feasible solution

Let $x = (x_1, x_2, \ldots, x_n)$ be a solution to a system $Ax \leq b$ of difference constraints, and let d be any constant.

Then $x + d = (x_1 + d, x_2 + d, \ldots, x_n + d)$ is a solution to $Ax \leq b$ as well.

Constraint graphs

The $m \times n$ linear-programming matrix A can be viewed as the transpose of an incidence matrix for a graph with n vertices and m edges

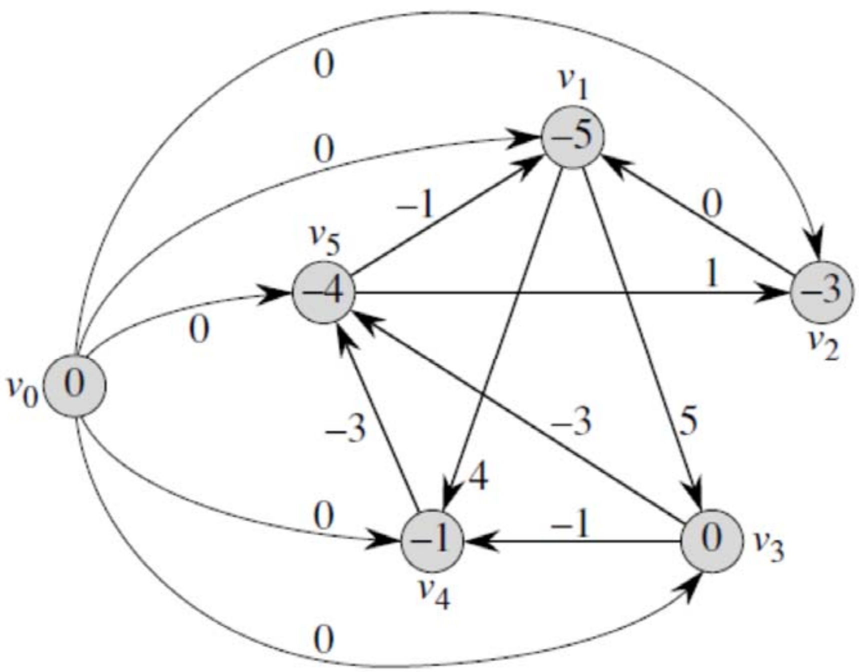
the corresponding ***constraint graph*** is a weighted, directed graph $G = (V, E)$, where

$$V = \{v_0, v_1, \ldots, v_n\}$$

and

$$E = \{(v_i, v_j) : x_j - x_i \leq b_k \text{ is a constraint}\} \cup \{(v_0, v_1), (v_0, v_2), (v_0, v_3), \ldots, (v_0, v_n)\}.$$

Constraint graphs - example



$$
\begin{aligned}
x_1 - x_2 &\leq 0\,, \\
x_1 - x_5 &\leq -1\,, \\
x_2 - x_5 &\leq 1\,, \\
x_3 - x_1 &\leq 5\,, \\
x_4 - x_1 &\leq 4\,, \\
x_4 - x_3 &\leq -1\,, \\
x_5 - x_3 &\leq -3\,, \\
x_5 - x_4 &\leq -3\,.
\end{aligned}
$$

$$x = (-5, -3, 0, -1, -4)$$

Theorem 24.9

Given a system $Ax \leq b$ of difference constraints, let $G = (V, E)$ be the corresponding constraint graph. If *G* contains no negative-weight cycles, then

$$x = (\delta(v_0, v_1), \delta(v_0, v_2), \delta(v_0, v_3), \ldots, \delta(v_0, v_n))$$

is a feasible solution for the system.

If *G* contains a negative-weight cycle, then there is **no feasible** solution for the system.

Solving systems of difference constraints

- use the Bellman-Ford algorithm to solve a system of difference constraints
- any negative-weight cycle in the constraint graph is reachable from v_0
- If the Bellman-Ford algorithm returns
  - TRUE, then the shortest-path weights give a feasible solution to the system
  - FALSE, there is no feasible solution to the system of difference constraints

Time complexity of

- A system of difference constraints with m constraints on n unknowns produces a
- graph with $n + 1$ vertices and $n + m$ edges

using the Bellman-Ford algorithm, we can solve the system in

$$O((n+1)(n+m)) \ = \ O(n^2 + nm) \text{ time}$$

Sample problem?

- How to modify the algorithm to run in $O(nm)$ time, even if m is much less than n.

Solve thee following systems of difference constraints

$x2 - x1 \leq 5$,

$x1 - x5 \leq -5$,

$x2 - x4 \leq -6$,

$x3 - x2 \leq 1$,

$x4 - x1 \leq 3$,

$x5 - x3 \leq 4$, .