北大合宿 2020 Day1 H 問題
Traditional Company

原案: tsutaj
問題文: tsutaj
解答: tsutaj
解説: tsutaj

2020 年 9 月 14 日

問題

Traditional Company

- N 人の社員がいる
- $s_i :=$ 社員 i の出勤時間, $e_i :=$ 社員 i の退勤時間
  - 各社員は必ず正の整数時間働く ($s_i < e_i$)
  - 社員の仕事効率は a_i または b_i (勤務時間によって変動)
- (会社全体の仕事量 X 以上) かつ (オフィス開放時間が最小) にしたい
  - そのようにできない場合は、できないことを報告
- いくつか条件あり
  - $i < j$ ならば $s_i \leq s_j$ (若い社員は早めに出勤)
  - 社員の組 (i, j) は仲が良い $\rightarrow$ 最低 1 単位時間以上は区間が被る
  - 社員の組 (i, j) は仲が悪い $\rightarrow$ 1 単位時間でも区間が被ってはならない

想定解法

- 想定解法: 最小費用流の双対
  - 未履修ならこれを機にやってみましょう！！
  - tokoharu さんの解説 PDF: Link
  - ei1333 さんの解説記事: Link
  - beet さんの解説記事: Link
- 双対を 1 から説明するのは厳しいので、申し訳ないですが双対に関する知識や最小費用流の双対の定式化はある程度既知として解説します

想定解法: 定式化

答えが Y 以下になってほしいときについて定式化してみる

Traditional Company の定式化

$$\text{maximize} \sum_{i=1}^{N} \min(t_i, p_i)a_i + \max(0, t_i - p_i)b_i \ (\star)$$

$$\text{s.t.} \begin{cases} s_i \le s_{i+1} & (1 \le i < N) \\ 0 \le s_i < e_i \le Y & (1 \le i \le N) \\ i \text{ の区間と } j \text{ の区間が被る} & (i \text{ と } j \text{ の仲が良い}) \\ i \text{ の区間と } j \text{ の区間が被らない} & (i \text{ と } j \text{ の仲が悪い}) \end{cases}$$

ただし、$t_i = e_i - s_i$ とする

$(\star)$ 式、つまり目的関数が X 以上になるなら解として採用できる！
Y を増やせば増やすほど目的関数値は増える $\rightarrow$ 二分探索 できそう

想定解法: 細かく定式化

最小費用流の双対で扱える式 (see: ei1333 さんの記事 Link)

$$\text{maximize} \sum_{e} (l_e x_e - h_e y_e) - \sum_{v=1}^{N} D_v k_v$$

$$\text{s.t.} \begin{cases} (x_e - y_e) + (k_v - k_u) \leq w_{u,v} \\ x_e \geq 0, y_e \geq 0, k_v \geq 0 \end{cases}$$

今回の問題はこれにあてはめられるのか？ $\rightarrow$ 実はできる！

想定解法: 必要な式の列挙

- 目的関数
  - $\min, \max$ がついているところが面倒だが、p_i を境に仕事効率が変化することに注目する
  - 最初から仕事量 $\sum_i a_i p_i$ を獲得しているものとして考える
    - $\sum_i a_i p_i + \max \sum_i (\max(0, t_i - p_i) b_i - \max(0, p_i - t_i) a_i)$
    - $\sum_i a_i p_i$ は定数なので適当に足し込めば良い
    - 勤務時間が p_i 未満なら傾き $-a_i$, p_i を超えるなら傾き b_i

想定解法: 必要な式の列挙

- 目的関数
  - $\min, \max$ がついているところが面倒だが、p_i を境に仕事効率が変化することに注目する
  - 最初から仕事量 $\sum_i a_i p_i$ を獲得しているものとして考える
    - $\sum_i a_i p_i + \max \sum_i (\max(0, t_i - p_i) b_i - \max(0, p_i - t_i) a_i)$
    - $\sum_i a_i p_i$ は定数なので適当に足し込めば良い
    - 勤務時間が p_i 未満なら傾き $-a_i$, p_i を超えるなら傾き b_i
- 制約式 (二変数の差にして不等式にしてみよう)
  - $s_i \leq s_{i+1} \to s_i - s_{i+1} \leq 0$
  - (i, j) の仲が良い場合: $s_j - e_i \leq -1$
  - (i, j) の仲が悪い場合: $e_i - s_j \leq 0$
    - このように単純な式にできるのは $s_i \leq s_{i+1}$ が効いている
  - $0 \leq s_i < e_i \leq Y \to$ 変数 z を導入して分離！
    - $z - s_i \leq 0$
    - $s_i - e_i \leq -1$
    - $e_i - z \leq Y$

想定解法: 必要な式の列挙

- 目的関数
  - $\min, \max$ がついているところが面倒だが、p_i を境に仕事効率が変化することに注目する
  - 最初から仕事量 $\sum_i a_i p_i$ を獲得しているものとして考える
    - $\sum_i a_i p_i + \max \sum_i (\max(0, t_i - p_i) b_i - \max(0, p_i - t_i) a_i)$
    - $\sum_i a_i p_i$ は定数なので適当に足し込めば良い
    - 勤務時間が p_i 未満なら傾き $-a_i$, p_i を超えるなら傾き b_i
- 制約式 (二変数の差にして不等式にしてみよう)
  - $s_i \le s_{i+1} \to s_i - s_{i+1} \le 0$
  - (i, j) の仲が良い場合: $s_j - e_i \le -1$
  - (i, j) の仲が悪い場合: $e_i - s_j \le 0$
    - このように単純な式にできるのは $s_i \le s_{i+1}$ が効いている
  - $0 \le s_i < e_i \le Y \to$ 変数 z を導入して分離！
    - $z - s_i \le 0$
    - $s_i - e_i \le -1$
    - $e_i - z \le Y$
- 制約部分はすべて二変数の差の形にできたので、あとは目的関数に関わる制約式を、二変数の差の式となるように作れば OK

想定解法: 目的関数について

- 勤務時間に関する不等式: $s_i - e_i \leq -p_i$
  - 勤務時間が p_i を超えるときに成立, 下回ると不成立
  - 勤務時間が p_i を超えるごとに制約に余裕が生まれ利得 b_i を獲得
  - 勤務時間が p_i を下回るごとに制約が厳しくなり利得 $-a_i$ を獲得
    - いいかえると、損失 a_i ということ
    - すでに $a_i p_i$ 獲得した状態からスタートしているので損失とみなせる
  - 5 ページで見た式に当てはめると、これは流量 $[b_i, a_i]$・コスト $-p_i$ の辺を、e_i を表す頂点から s_i を表す頂点に向かって張ることに等しい
    - 制約で $b_i \leq a_i$ が保証されているので流量がおかしくなることはない

解法まとめ

- $2N + 1$ 頂点のグラフを考える
  - 各社員について始業・終業時間があるので頂点倍化 + 追加変数 z
- 目的関数に関係する制約条件に対応する辺を張る
  - 前ページに書いた通り
- 制約式に対応する辺を張る
  - 辺が張られる対象・コストは制約式から決まる
  - 流量はすべて $[0, \mathrm{INF}]$
    - 制約に余裕があっても特に報酬ないので 0 とする
    - 制約を違反すると解として採用できないので罰則 INF
- 最小費用流を流して答えを求める
- 実際には解を二分探索しつつ、その中で上記を実行することになる

注意

- 二分探索の上限を適当に設定すると WA になる可能性があります
  - 二分探索の上限が INF と近いとダメ
  - 目指す解が大きすぎると、一部の制約式を無視して罰則を得つつ、他で大量に利得を得ることで目標を達成できてしまう場合がある
- 具体的にはどこまで探索すればいい？ $\rightarrow$ 解の上限は $O(N^2A+X)$
  - ただし A は、取りうる a_i, b_i の絶対値最大
  - これに気をつけながら、二分探索の上限と INF にある程度差を付ける

解の上限: 略証

- 目指すべき仕事量制約を無視 $\to$ 答えは $O(N)$
  - 社員の出勤時間を表す区間の端点がひとつもない座標は適当に圧縮可能
  - 端点の数は $O(N)$ なので答えは $O(N)$
- 答え $O(N)$ の状態で達成される仕事量合計 $\to O(N^2A)$
  - それぞれの区間長は N まで行きうる
  - なので社員 1 人の仕事量としてありえるのは $O(NA) \to$ 全体 $O(N^2A)$
- 仕事効率はすべて整数 $\to$ 伸ばして得できる区間について、1 伸ばせば 1 以上仕事量を増やせる
  - $a_i \geq b_i$ であるので、ある区間内にいる社員の集合が同じならば、それを伸ばし続けることで途中で利得増分が上がることはない
    - 最初は負の利得だったけど伸ばし続けていたら正になるとかはない
    - なので伸ばしても得にならないとわかった瞬間、伸ばすのをやめていい
  - 仕事量合計が $-N^2A$ まで落ち込んでいたとしても、また N^2A 伸ばせばチャラに
  - さらに X 伸ばすと目標を達成可能
  - 以上より $O(N^2A + X)$ あれば十分
    - Writer が思いついたもののうち答えが最大のものは 1254949 です

Writer 解・統計

- Writer 解
  - tsutaj (C++・322 行・10623 bytes)
    - ほぼライブラリ部分
- 統計
  - AC / tried: 2/5 (40.00 %)
  - First AC
    - F_flow_G_greedy_H_DP (152 min 59 sec)