

Introduction to Heuristics Contest 解いてみた感想

1. はじめに

こんにちは、マラソン初心者 (tenorist) と申します。最近 AHC が開催されたので、マラソンに少し首を突っ込んでみようということで Introduction to Heuristics Contest の問題を解いてみました。

マラソンコンテストとは、普段やっているように「最適解」を求めるのではなく、「できるだけ良い解」を目指すコンテストのことです。と言われても何のことだかさっぱり分からなかったなので、とりあえず問題を読んでみることにしました。

2. 問題と解き方 (本題)

2.1. 問題概要 [A 問題](#) (クリックすると開きます)

要は、あるコンテストが開かれることで得る満足度、各コンテストが開かれないことで減る満足度を考慮しながら、D 日目の (終わりの) 満足度を最大化しようということです。

また、以下では何日目か・コンテストのタイプは 0-indexed で書いてます。

2.2. [B 問題](#) を解く → 貪欲解を作る

B 問題では、与えられるコンテストの日程をもとに満足度を計算します、やるだけ。 [提出コード](#)

次に、これをもとに貪欲的に日程を決めることを考えます。d 日目のタイプ i が開催されなかったときの満足度減少量を $down[i]$ とおきます。一方、その日タイプ i のコンテストが開催されれば満足度は $+s[d][i]$ なので、d 日目終了時の満足度を最大化させるには、 $s[d][i] + down[i]$ が最も大きいような i を選べばよいです。 62634806 点 を獲得です。 [提出コード](#)

2.3. [C 問題](#) を解く → 局所探索法へ

C 問題では、コンテストの日程の変更が与えられ、高速に最終満足度の変化を計算することが求められます。まず、そのコンテストが行われるまでの日数を n 、d 日目以前にタイプ i が行われた日から d 日目まで m 日経ったとします。d 日目のタイプ i コンテストを削除すると、満足度は $c[i] \times mn$ 減少します。次に、d 日目にタイプ q のコンテストを生やすと、同じような考えで満足度の増加を計算可能です。2 重 vector で各タイプの行われる日を管理すると、二分探索を使って前後の日が分かるので、 $O(\text{クエリ数} \times \log D)$ であり、十分高速です。 [提出コード](#)

局所探索法 (山登り法) とは、一部をランダムに変化させることで、結果が悪化したなら戻し、良くなったらそのまま、というのを時間いっぱい繰り返す解法です。ここでは、2.2. 節で求めた貪欲解をベースにしました。random 関数と clock 関数は初めて使いました。時間目いっぱい使って 107623406 点 獲得です (急上昇、感激)。 [提出コード](#)

2.4. 1 点変更と 2 点 swap

直前のコードでは、ある 1 つのコンテストのタイプを変更しましたが、解説 (wata さん) によると、2 つの日をランダムに選び、近かったらそれらのコンテストタイプを入れ替える、と良いそうです。これを実装して何回かやってみると、微増して 107873376 点 獲得です。 [提出コード](#)

2.5. その他の解法

解説曰く、局所探索法はこの問題では相性が悪いそうです。「少し下がった先の高い山を見逃してしまう」というのはとても納得できます。そこで、スコアの悪化を確率的に受け入れる「焼きなまし法」が用いるとより良いです（でもムズイ）。また、ビームサーチ (chokudai search はこれの亜種)、巨大近傍法などの解法もあるそうです。へー...

3. おわりに

以下、小学生並みの感想です。

- ・同じコードでも、得られる得点が変わったりしておもしろかった (random なので)
- ・得点が出るのを待つのはワクワクしました
- ・マラソンの世界は深いなーと思いました
- ・一応まだ若いので、速さと解けた問題数を競うコンテストの方がやっぱり楽しいかな

ありがとうございました。