

Hitachi Hokudai Lab. & Hokkaido University Contest 2020

サマリー記載例

(例として用いたアルゴリズム: toolkit 内 sample_B.cpp の transport_only_0)

- 基本情報

Screenname: XX

提出日: 2021/XX/XX

- メインの記載項目

(1) (a)

概要

TransportOnly Algorithm は、運搬のみを考慮して B 問題を解くシンプルなアルゴリズムである。各 EV はナノグリッド上に待機し、注文を先着順に運搬した後、最寄りナノグリッドに移動して待機する、という行動を繰り返す。

戦略について

* 各 EV は注文が入るまで、ナノグリッド上で待機する。

* 待機中、各 EV は一定量の充電を確保する。(438 行目 : safety_energy で指定する量。実装では 50 回の移動が行える量とした)

* ナノグリッド上で充電を終えた EV は、先着順で注文を処理する。

* この時、最大で 4 つの注文を一括して処理する。一般にはより多くの注文を一括して処理することが望ましいと考えられるが、EV の最大積載量による制約を簡単に処理するため、4 つに制限している。

(1)(b)

実装について

* 各 EV の行動は行動キューで管理されている。

* 疑似コードでは、n 番目の EV を `EV[n]`、対応する行動キューを `queue[n]` とする。

* `move_to` 関数を用いて、各 EV の移動処理を簡潔に行動キューに積み込めるようになっている。

* ソースコード(提出番号: XXXXXXXXXX)内の 426 行目~489 行目の transport_only_0 と、その内部で呼び出している find_transit_path_greedy(264 行目~297 行目)などに実装

疑似コード

```
``` text
```

```

入出力形式 1 を読み込む
for t = 1 to T_max do
 入出力形式 2 を読み込む
 for n = 1 to N_EV do
 // queue[n] に命令を積み込むための
 if queue[n] が空
 // EV[n]を稼働させ続けるため、ナノグリッドへの移動と充電は優先して判定する
 if EV[n]の場所がナノグリッドでない
 if EV[n]はナノグリッドに到着するのに十分な電力が残っていない
 queue[n]を stay で埋める。// つまり、EV[n]は今後移動しない
 continue
 else
 queue[n]に最も近いナノグリッドへの move 命令を積み込む
 continue
 end if
 else
 if EV[n]のエネルギーが safety_energy より少ない
 queue[n]に safety_energy 分の充電を行うための charge_from_grid 命令を積み
 込む
 continue
 end if
 end if
 end if

 // ここから運搬のための処理
 if 運搬する EV が割り当てられていない注文が存在する
 先着順に最大 4 つの注文に EV[n]を割り当て
 割り当てた注文の積み込み、積み下ろしスケジュールを貪欲法で決定
 if 積み込み、積み下ろしスケジュールに必要なだけ充電されていない
 queue[n]に必要なだけの充電を行うための charge_from_grid 命令を積み込む
 end if
 queue[n]に積み込み、積み下ろしスケジュールに従って move 命令を積み込む
 end if
 end if
end if
// 実際に命令を出力する部分
if queue[n]が空
 stay を出力

```

```
 else
 queue[n]から命令を一つ取り出して出力
 end if
end for
end for
入出力形式 2 を読み込む
入出力形式 3 を読み込む
...
```

- (2) 時間がなく運搬部分しか取り組めなかったが、比較的シンプルなアルゴリズムでも、そこそこの数の運搬をこなすことができた。電力部分にも取り組めるとスコアが更に伸ばせたと思う。問題文が長くて読むのが大変だったが、実用を見据えた問題という印象があり、取り組むモチベーションになった。