

前方に人がいることを許した自転車共有問題

Haralanov Valeri *

Yamada Toshinori †

1 はじめに

TODO

2 自転車共有問題 (BS)

自転車共有問題を次のように定義する [1]. まず, 入力として与えられる情報は

- $m \in \mathbb{N}$: 人の数,
- $U \in (0, 1)^b$: それぞれの自転車の速度が v_i のとき, $u_i = \frac{1}{v_i}$ をその逆数として列にまとめたもの.

なお, b は自転車の数を表すことに注意する. さらに, $b < m$ となるような入力のみを考えることとし, U が昇順にソートされているとする.

はじめに全ての人と自転車が点 0 (以降出発点) に配置されているとする. 人は速度 1 で歩くか, とある自転車 i に乗って速度 v_i で移動することができる. 自転車には同時に一人しか乗ることができなく, その人が自転車に乗るためには人も自転車も同時に同じ場所にいないなければならない. さらに, 人は任意の時点で自転車を降りることができる. また, 人が乗っていない自転車は移動することができない.

BS の目標は全ての人及び自転車が点 1 (以降到着点) まで最も早く移動できるようなスケジュールを組み立てることである. なお自転車も到着地点まで移動しなければならないことに注意する. 自転車が人より少ないので, 最適なスケジュールを求めるのは自明ではなく, 人がどうにか自転車をうまく共有す

るように設計しなければならない. Czyzowicz et al. [1] により提案されたアルゴリズムはまさにそのような特別な共有パターンを活用している.

自転車の共有という概念は行列として表すことができる. 区間 $[0, 1]$ を n 個の小区間 x_j に分け, 人 i が小区間 j で乗った自転車の番号を $M_{i,j}$ と置き, M をスケジュール行列と呼ぶ. ただし, 人 i が徒歩で移動した小区間では $M_{i,j} = 0$ とする. しかし M は実際の計算では少し使いづらいので, 自転車の番号ではなく速度の逆数を格納した行列 $\widetilde{M}_{i,j} = u_{M_{i,j}}$ も定義しておく. それぞれの小区間の長さをベクトル $X \in [0, 1]^n$ にまとめ, 順序対 (X, M) をスケジュールと呼ぶ. さらに, スケジュールに対し以下の量を定義する.

- 人 i が小区間 j の終点に到着するのに必要な時間

$$t_{i,j}(X, M) = \sum_{k=1}^{k \leq j} X_k \widetilde{M}_{i,k}.$$

- 人 i が到着点に到着するのに必要な時間

$$t_i(X, M) = t_{i,n}(X, M) = \sum_{k=1}^{k \leq n} X_k \widetilde{M}_{i,k}.$$

- 全員が到着点に到着するのに必要な時間

$$\tau(X, M) = \max_i t_i(X, M).$$

また, 特定のスケジュールに関係なく, とある BS の入力に対し最適な時間を $\tau(m, U)$ として表す.

これらの定義を用いていくつかの条件を加えることで M に対する線形計画法で X と τ を求めることができる. したがって BS の鍵となるのは M の計算である. Czyzowicz et al. [1] は τ の下界を 2 つ

* 埼玉大学工学部情報工学科

† 埼玉大学工学部情報工学科

示し、いずれかが必ず満たされるようなスケジュール行列の計算方法を示した。それぞれの下界は以下の補題として定義する。

補題 1.

$$\bar{\tau}(m, U) \geq u_b. \quad (1)$$

Proof. 自転車も到着点まで移動しなければならないが、それぞれの移動速度が決まっているので一番遅い自転車が全区間を移動する時間は必ずかかる。□

補題 2.

$$\bar{\tau}(m, U) \geq T(m, U) = 1 - \frac{1}{m} \sum_{j=1}^b (1 - u_j) \quad (2)$$

Proof. 各人が止まることなく常に歩いているもしくは自転車に乗って動いていると仮定すれば、 $T(m, U)$ は全員の移動時間の平均値を表す。他方最適なスケジュール (M, X) に対し $\bar{\tau}(m, U) = \tau(M, X) = \max_i t_i$ となるのが、最大値が平均値以上でなければならないことから主張が成り立つ。□

補題 2 では人が常に動いているというのと、後退をしないという仮定が必要であるが、Czyzowicz et al. [1] はそのどちらを許したとしてもより早い到着時間が得られないことを示している。以下の主張は簡単であるが、解法アルゴリズムに対し重要なので敢えて述べておく。

系 1. 全員が同時に到着するときかつそのときに限り、 $\bar{\tau}(m, U) = T(m, U)$.

BS を解くアルゴリズムは 補題 2 及び 系 1 を活用したものである。その概ねの挙動を以下に示す。

- $u_b \leq T(m, U)$ の場合、一部の人に順番に先に自転車に乗ってもらって途中で降りて歩いてもらう。なお空間的に自転車の位置が自転車の速さと同順であり、速い自転車が先にある状態を維持し、自転車を降りた人達が歩行するときに同時に同じところを歩く状態を作る。全員が一回自転車に乗ったあと、最後に乗っている人が先の歩行者に追いついた時点でそのグループに加

わり、次の人が追いつくまでの区間では歩行者と追いついた自転車一台でまたグループとして動いてもらうことを考える。これはつまりより小さい入力に対して同じ問題を解くことを意味する。なおグループの動きの性質として、全員が同じ場所から同時出発をすると、次に人が追いついたときにまた全員が同じときに同じ場所にいることが保証される。この性質を用いて後ろの人と自転車をどんどん吸収していき、一番遅い自転車に乗っている一番後ろの人がちょうど到着点に他の人と合流するようなスケジュールを組むことで $\bar{\tau}(m, U) = T(m, U)$ を満たすスケジュールを得ることができる。

- $T(m, U) < u_b$ の場合、遅い自転車から取り除いていくと $u_k \leq T(m - b + k, U_k) \leq u_b$ を満たすグループが作れることが保証される。ただし U_k は k 番目までの自転車のみを含んだ列である。このとき、小さいグループを上記の方法で動かし、余った自転車は余った人に全区間を走ってもらうことで u_b に乗っている人が最後に到着するようなスケジュールを作ることができる。このとき、 $\bar{\tau}(m, U) = u_b$ となる。

3 前方に人がいることを許した自転車共有問題 (FSABS)

3.1 問題設定と定義

FSABS を定義する前に 2 つの補助的概念を導入する。

まず、各人の初期位置を表すための配列 $A \in (0, 1)^m$ を考え、 A の要素が全員分あることに注意する。理論的には前方にいる人の初期位置だけで十分であるが、以降で紹介するアルゴリズムの動作の都合上、全員分の初期位置を用意の方が扱いやすいため、ここでは出発点にいる人の初期位置を 0 とし、その人達の初期位置を含めた A を考える。

BS の出力は各人が各小区間で使用した自転車の番号を表す行列 M と各小区間の長さを表すベクト

ル X と定義した．FSABS の出力にもこのような形を採用することはできるが，上記同様に解法アルゴリズムの動作の都合上，次のような形の出力を考える． m 個の要素からなる配列 S を考え，それぞれの要素がそれぞれの人に対応するとする．人 i に対して S_i の値は n_i 次元配列であり，その各要素が $(\alpha_{i,j}, \beta_{i,j})$ のような順序対であり，人 i が自転車 $\beta_{i,j}$ で距離 $\alpha_{i,j}$ を連続移動したことを意味する．ただし，歩行したならば $\beta_{i,j} = 0$ と置き， $0 < j < n_i$ とする．このような S を自由スケジュールと呼ぶ．

S_i の各要素に対応する $[0, 1]$ 上の始点と終点をそれぞれ $\rho_{i,j}$ 及び $\sigma_{i,j}$ で表すことができる．

$$\rho_{i,j} = A_i + \sum_{k=1}^{k < j} \alpha_{i,k} \quad (3)$$

$$\sigma_{i,j} = \rho_{i,j} + \alpha_{i,k} \quad (4)$$

これらをまとめて， $\alpha_{i,j}$ に対応する小区間を $\chi_{i,j} = (\rho_{i,j}, \sigma_{i,j})$ と表す．

S において人 i が点 $x \geq A_i$ に到達する時間を $t'_i(x)$ とする． $t'_i(x)$ の値は S_i の要素の線形和として求めることができるが，複雑な表記を必要とするためここでは省略する．全体の到着時間は BS と同様にそれぞれの人の到着時間の最大値となるので， S における到着時間を

$$\tau'(S) = \max_i t'_i(1) \quad (5)$$

と定義できる．

次に S が実行可能解であるために Czyzowicz et al. [1] と同様な条件を述べる．

定義 1. 入力 (m, U, A) と上述の構造を持つ自由スケジュール S に対し，以下が成り立つとき且つそのときに限り S を実行可能な自由スケジュールと呼ぶ．

1. $(\forall i \text{ s.t. } 1 \leq i \leq m) \quad \sum_{j=1}^{n_i} \alpha_{i,j} = 1 - A_i$.
2. $\beta_{i,j} \neq 0$ であるならば $\sigma_{i',j'} = \rho_{i,j}$ かつ $\beta_{i,j} = \beta_{i',j'}$ を満たす i', j' が存在する．
3. $\beta_{i,j} \neq 0$ であるならば $\chi_{i,j} \cap \chi_{i',j'} \neq \emptyset$ となるような i' 及び j' に対し $\beta_{i,j} \neq \beta_{i',j'}$.

4. $\beta_{i,j} = \beta_{i',j'}$ 及び $\sigma_{i,j} \leq \rho_{i',j'}$ ならば $t'_i(\sigma_{i,j}) \leq t'_{i'}(\rho_{i',j'})$.

なお，BS の出力となる実行可能スケジュール (M, X) が常に実行可能な自由スケジュールへの変換が可能で，上記の条件を満たすことに注意する．

以上の内容をまとめて，FSABS を次のように定義する．

FSABS

入力: $m \in \mathbb{N}$: 人の数．

$U \in (0, 1)^b$: 自転車の速度の逆数を格納した昇順配列．

$A \in (0, 1)^m$: 各人の初期値を格納した昇順配列．

出力: S : 実行可能な自由スケジュール．

目的: $\tau'(S)$ を最小化すること．

以降の議論では具体的な出力に関係なく，とある入力に対する最適な到着時間を $\tau'(m, U, A)$ と表す．

3.2 preliminary observations

以降は FSABS の下界について論じる．補題 2 と同様に以下の補題で FSABS に対する下界を定義できる．

補題 3.

$$\tau'(m, U, A) \geq T(m, U, A) \quad (6)$$

$$= 1 - \frac{1}{m} \sum_{j=1}^b (1 - u_j) - \frac{1}{m} \sum_{i=1}^m A_i \quad (7)$$

$$= T(m, U) - \frac{1}{m} \sum_{i=1}^m A_i \quad (8)$$

Proof. 補題 2 と同様に $T(m, U, A)$ は全員の移動時間の平均値を表すので，最大値が平均値以上であることから主張が成立する． \square

次に新しい形での下界を導入する．人を増やすことによって到着時間が早くなることは直感的に考えにくく，BS においては $T(m, U)$ の m に対する単

調増加性を示すことによってそれを形式的に証明できる。以下の補題では同じ考え方を FSABS について証明する。ただし $A_{:k}$ を A_k までを含んだ列とする。

補題 4. (補題 or 定理?) $m > b$ とする。このとき

$$\bar{\tau}'(m, U, A) \geq \bar{\tau}'(m-1, U, A_{:m-1}) \quad (9)$$

Proof. 背理法を用いて $\bar{\tau}'(m, U, A) < \bar{\tau}(m-1, U, A_{:m-1})$ であると仮定し、 (M, X) を (m, U, A) に対する最適スケジュールとする。人 m がはじめて自転車に乗る小区間を x_i とし、その始点に乗る自転車 u_j に注目する。その自転車には、人 α が小区間 x_{i-1} で乗っており、事前に x_i の始点まで運んでくれたはずなので人 α に引き続き x_i で乗ってもらう。そうすることで自転車 u_j は少なくともスケジュール通りに x_i の終点に到着する (少なくともというのは、乗り換えの都合で u_j が一定時間使用されない可能性があるのに対し、人 α がずっと乗ることでその時間が省けるということを意味する)。しかし、元々人 α は x_i で u_j を使わないことになっている。もし人 α が x_i で u_k の自転車に乗る予定だったならば、今度は同じ議論を u_k を運んでくれた人に対して適用する。それを繰り返すと、いずれ x_i を徒歩で移動する予定だった人 β にたどり着く (なぜなら自転車の数が $b \leq m-1$ だからである)。その人は x_{i-1} で何かの自転車に乗っている前提だが、その自転車を引き続き使えば良い。

ここまでの処理を施すと、元々 x_i の終点にとある時刻に到着すべきだった人達が人 m 以外全員揃い、誰かが早く到着したとしてもそこで待てば良い。ただし、元々と違うのは x_{i+1} 以降の役割が入れ替わっており、上記で言う人 α が人 m の役割を果たすようになっている。「役割を果たす」というのは x_{i+1} 以降のスケジュールを入れ替え、人 α が人 m のスケジュールを取れば良い。しかし人 m を無視することによって一人役割が余っている人がいる。

上記の議論では「事前に」という条件を付けているが、これはつまり自転車の運搬を遡るにつれて、ある自転車を運んだ人がその自転車を使う人よりも

早く x_i の始点に到着して、 x_i で移動を始めているという前提である。そうでないと自転車の使用者が自転車の到着よりも先に「乗る」ことになり、おかしい。ここで元々 x_i で歩行する予定だった上記の人 β に注目する。人 β は x_i で現在自転車を使用し、違う人の役割を担っている。しかし元々のスケジュールでは歩行する予定で、少なくとも人 m と同じタイミングか、それより早く x_i の始点に到着し次の移動に移る。もし人 β が人 m と同時出発だったのであれば、人 m は初期位置を変えずそのまま歩けば良い。他方でもし人 m より早い場合は、人 m の初期位置を適切にずらすことで、人 m が元々の人 β の到着時刻に x_i の終点に到着するように変更できる。

この操作を施すことによって元々のスケジュールにかかる時間を保ちながら、人 m が最初に自転車に乗る区間を前にずらすことができる。最悪 n 回 (小区間の数) 繰り返せば、人 m が自転車を使わないスケジュール (M', X') が得られ、 $\tau(M, X, A) \leq \bar{\tau}'(m, U, A)$ を満たす。しかし人 m が自転車を使わなければ $\tau(M, X, A) \geq \bar{\tau}(m-1, U, A_{:m-1})$ が成り立つので、 $\bar{\tau}(m-1, U, A_{:m-1}) \leq \bar{\tau}(m-1, U, A_{:m-1})$ となり矛盾が生じる。□

上記の議論から以下の系が容易に得られる。

系 2. $i \leq m-1$ に対し $A_i = 0$ とする。

$$\bar{\tau}'(m, U, A) \geq \bar{\tau}(m-1, U). \quad (10)$$

3.3 FSABS を解くアルゴリズム SolveFSABS

本節では FSABS を解くアルゴリズム SOLVE-FSABS を定義し、その計算量と正当性について論じる。まず、アルゴリズムの挙動をアルゴリズム 1 に擬似コードで示した。

SOLVE-FSABS の挙動を説明する前にいくつかの補助的な概念を導入する。

1. グループ: 出発点にいる人と自転車のまとまり。グループの移動は BS のインスタンスとして計算

Algorithm 1: SOLVE-FSABS

```

input : 自然数  $m$ 
        昇順に並べた列  $U \in (0, 1)^b$ 
        昇順に並べた列  $A \in (-\infty, 1)^m$ 

output: 自由スケジュール  $S$ 

1 if  $\forall i, A_i = 1$  then
2   return  $\square$ ;
3  $r \leftarrow$  後方にいるライダーの数;
4  $f \leftarrow$  前方にいる歩行者の数;
5 if  $u_b > T(m - f - r, U_{:b-r})$  then
6    $m_k \leftarrow \text{SUBGROUP}(m, U_{:b-r});$ 
7    $r \leftarrow r + m - m_k;$ 
8  $(t, d) \leftarrow (\text{NIL}, \text{NIL});$ 
9 if  $f > 0$  then
10   $(t, d) \leftarrow \text{TONEXTW}(m, U, A, r);$ 
11 else
12   $(t, d) \leftarrow \text{TOEND}(m, U, A, r);$ 
13  $\text{groupT} \leftarrow T(m - f - r, U_{:b-r});$ 
14 if  $r > 0$  and  $u_{b-r} < \text{groupT}$  then
15   $(t_1, d_1) \leftarrow \text{TONEXTR}(m, U, A, r);$ 
16  if  $0 < d_1 < d$  then
17     $(t, d) \leftarrow (t_1, d_1);$ 
18  $S \leftarrow \text{MOVE}(t, m, U, A);$ 
19  $A^{(1)} \leftarrow A;$ 
20 for  $i \leftarrow 1$  to  $r$  do
21   $A_i^{(1)} \leftarrow \min\{1, \frac{A_i + t u_{b-i+1} - d}{1-d}\};$ 
22 for  $i \leftarrow m - f + 1$  to  $m$  do
23   $A_i^{(1)} \leftarrow \min\{1, \frac{A_i + t - d}{1-d}\};$ 
24  $S_1 \leftarrow \text{SOLVE-FSABS}(m, U, A^{(1)});$ 
25 return  $\text{MERGE}(S, S_1, 1 - d);$ 

```

することができる。グループの人に対し $A_i = 0$ が成り立つ。

2. 後方ライダー: 出発点より後ろにおり、ずっと同じ自転車に乗っている人。ライダーに対し $A_i < 0$ が成り立つ。

3. 前方歩行者: 出発点より前におり、ずっと歩いている人。歩行者に対し $A_i > 0$ が成り立つ。

これらを踏まえた上で SOLVE-FSABS の入力に対する次の制約を述べる。

条件 1. ライダーとなる ($A_i < 0$ となる) 人 i は u_{b-i} の自転車を使っていなければならない。つまり、ライダーとなる人達は常に一番遅い自転車に乗っており、さらに遅ければ遅いほど後ろにいないといけない。

SOLVE-FSABS は再帰的なアルゴリズムであるが、呼び出すときには各人が必ず上記のいずれかの分類に含まれる。なお、最初に呼び出すときにはライダーがいないことに注意する。

続いて SOLVE-FSABS の概ねの挙動を説明する。

1. 全員を前進させ、出発点にいるグループが後ろから追いつてくるライダーもしくは前方にいる歩行者のいずれか早い方と合流する点までの距離と移動スケジュールを計算する。なお、もし到着点までに合流できない場合は全区間を移動させる。
2. グループが動いた時間だけ他のライダーや歩行者を動かす。
3. 合流地点を新たな出発点として考え、後方ライダーや前方歩行者の相対的な位置を再計算した上で再帰的に SOLVE-FSABS を呼び出し FSABS を解く。この際、合流によって後方ライダーもしくは前方歩行者が少なくとも一人グループに吸収されるので入力がいよる簡単になることに注意する。
4. 新たに解いた問題のスケジュールを最初に計算したスケジュールと合併させる。

最初にアルゴリズムを呼び出すときにライダーがないが、ステップ 1 ではライダーを考慮する必要がある。SOLVE-FSABS は再帰的なアルゴリズムなので、ライダーがどのように登場するかを後ほどの議論に任せ、以降ではライダーがいる前提での挙動について論じる。

ステップ 1 の通り、グループを移動させ、ライダーもしくは歩行者と合流させたい。補題?? を満たすためには到着点に到達していない人が全員常に動いていなければならないので、合流する時にグループも歩行者 (ライダー) も同時に合流点に到着する必要がある。しかし、グループが持っている自転車の速度によっては、一定区間動かしたときに全員が同時に到着しない場合があり得る。幸いなことに、そのような場合には 補題?? より同時到着を実現できる部分グループの存在が保証されるので、その部分グループを新たに「グループ」として考え、余った人と自転車をライダーとして考える。アルゴリズムの 5 行 の条件分岐がこの部分に対応する。

部分グループの採用による調整を行った後、次に合流できる人を定める。

IRRELEVANT

元々のライダーの数を r 、歩行者の数を f をする。部分グループを採用したならば元々のグループから r' 人が余ったとし、元々のライダーと合わせて $r^{(1)} = r + r'$ とする。グループが次に前方の歩行者の中で一番出発点に近い人と合流することを考える。補題?? より $\tau(m - f - r, U) = T(m - f - r, U)$ をグループの速度として考えることができるので、一番近い歩行者が人 p だとすると、合流点 x を以下の式で求めることができる。

$$xT(m - f - r, U) = (x - A_p) \times 1 \quad (11)$$

$$x = \frac{A_p}{1 - T(m - f - r, U)} \quad (12)$$

もし $x \geq 1$ 、つまり合流点が到着点以降になるのであれば、 $x = 1$ と置いて到着点までの移動だけを考える。グループも歩行者も常に動いていなければならないので、合流までにかかる時間は

$$t = (x - A_i) \times 1 = xT(m - f - r, U) \quad (13)$$

となる。したがって、全員を移動させることを考えたときに、グループと合流する歩行者以外に他の歩行者もライダーも時間 t 分だけ動かす必要がある。この動きを A に対する更新として表すことができる。しかし、この処理の目的は FSABS のより簡単な入力を生成することなので、全員を動かした後に到着点までの残りの距離 $1 - x$ を単位区間とし、相対位置をその単位区間に合わせなければならない。この操作を以下の式で表す。

$$A_i^{(1)} = \begin{cases} \frac{A_i - x + tu_b - r + i}{1 - x} & i \leq r + r_1 \\ \frac{A_i - x + t \times 1}{1 - x} & i > m - f \end{cases} \quad (14)$$

ここ

ここでグループについて考慮しなければならない場合を詳しく述べる。アルゴリズム SOLVE-FSABS が前節で定義した下界を満たすためには全員が止まることなく常に移動するスケジュールを出力しなければならない。そのため、グループが合流地点まで移動するときに全員が同時に到着しなければならないが、使っている自転車の速度によってそのようなことが実現できるとは限らない。しかし、補題?? より同時到着を実現できる部分グループの存在が保証される。したがって、その部分グループを先の人と合流させ、余ったライダーが単純に後ろでずっと乗り続ける状況を考えれば良い。他方で 補題?? よりグループが人を吸収するたびに遅くなり、条件によっては一度離れたライダーと再び合流でき、再び同時到着を満たせる状態で一緒に移動できる場合が考えられる。

4 結論と今後の課題

TODO