前方に人がいることを許した自転車共有問題

ハラランフ ヴァレリ *　　山田 敏規 †

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ばならない．Czyzowicz ら [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
$$t_{i,j}(X,M) = \sum_{k=1}^{k \le j} X_j \widetilde{M}_{i,j}.$$
- 人 i が到着点に到着するのに必要な時間
$$t_i(X,M) = t_{i,n}(X,M) = \sum_{k=1}^{k \le n} X_j \widetilde{M}_{i,j}.$$
- 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 $\overline{\tau}(m,U)$ として表す．

あらゆる (M,X) の中で，次の条件を満たすものを実行可能なスケジュールという．

定義 1. (M,X) *BS において実行可能解となるときかつそのときに限り，以下が満たされる．*

*埼玉大学工学部情報工学科
†埼玉大学大学院理工学研究科数理電子情報部門

1. $M_{i,j} \neq 0 \implies \exists i' \, s.t. \, M_{i',j-1} = M_{i,j}$.

2. $M_{i,j} \neq 0 \implies \forall i' \neq i, \, M_{i',j} \neq M_{i,j}$.

3. $M_{i,j} = M_{i',j-1} \neq 0 \implies t_{i',j-1} \leq t_{i,j-1}$.

4. $\forall 1 \geq j \geq b, \exists i \, s.t. \, M_{i,n} = j$.

これらの定義を用いることで 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 ら [1] は $\bar{\tau}$ の下界を 2 つ示し，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.

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

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あ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.

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

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 ら [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．

一方 $T(m, U) < u_b$ の場合，次の性質を活かすことができる．

補題 3. *(Czyzowicz ら [1])* (m, U) を *BS* の入力とする． $u_b \geq T(m, U)$ ならば次を満たす k が存在する．ただし， $U_k = [u_1, u_2, \ldots, u_k]$ 及び $m_k = m - b + k$ とする．

$$u_k \leq T(m_k, U_k) \leq u_b \tag{3}$$

このとき，上記の性質により与えられるより小さいグループを動かし，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，$\bar{\tau}(m, U) = u_b$ となる．

最後に， BS の解は任意の区間長に適用できることを明記しておく．以下の補題は Czyzowicz ら [1] らの主張を簡略化したものである．

主張 1. *BS* の入力 (m, U) に対し (M, X) を実行可能解とする．任意の区間 $[0, a]$ に対し *BS* を定義し

たとき，同じ入力に対し (M, aX) が実行可能解となり，$\tau(M, aX) = a\tau(M, X)$.

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する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，それの各要素が $(\alpha_{i,j}, \beta_{i,j})$ 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．また，S_i のことを S の行と呼ぶ．

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．

$$\rho_{i,j} = A_i + \sum_{k=1}^{k<j} \alpha_{i,k} \tag{4}$$

$$\sigma_{i,j} = \rho_{i,j} + \alpha_{i,k} \tag{5}$$

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

$$\tau'(S) = \max_i t'_i(1) \tag{6}$$

と定義できる．また，S において人 i が移動した距離を

$$L(S_i) = \sum_{j=1}^{n_i} \alpha_{i,j} \tag{7}$$

と表す．

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

定義 2. 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. $(\forall i\, s.t.\, 1 \leq i \leq m) \quad L(S_i) = 1 - A_i$.
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 $t'_i(\sigma_{i,j}) \leq t'_{i'}(\rho_{i',j'})$.
5. $\forall j\, s.t.\, 1 \leq j \leq b,\ \exists i\, s.t.\, \beta_{i,n_i} = j$.

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．

また，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．

解法アルゴリズムでは人をグループに分け，それぞれの移動を独立に考えてからそれぞれの自由スケジュールを組み合わせるという操作を頻用する．この操作を「縦に組み合わせる」と名付け，以下の命題でその正当性を保証する．

命題 1. *Show that optimal time is max of each schedule* $(m_1, U_1 = [u_1, u_2, \ldots, u_{b_1}], A_1)$ 及び $(m_2, U_2 = [u_{b_1+1}, u_{b_1+2}, \ldots, u_{b_1+b_2}], A_2)$ をそれぞれ *FSABS* の入力とし， $S^{(1)}$ 及び $S^{(2)}$ をそれぞれに対する実行可能解とする． A_1 及び A_2 の要素を A_3 にまとめ，昇順に並べることで新しくできた人の番号を用いて $S^{(1)}$ 及び $S^{(2)}$ の行を適切な順番に並べた $S^{(3)}$ を構成する．このとき， $S^{(3)}$ は入力 $(m_1+m_2, U_1+U_2, A^{(3)}$ に対する実行可能解となる．

Proof. $S^{(1)}$ 及び $S^{(2)}$ は独立に実行可能解であり，且つお互いの自転車を共有することがないので 定義 2 の条件は自明に満たされる． □

最後に， FSABS が線形スケール化可能であることを主張しておく．

主張 2. *FSABS* の入力 (m,U,A) に対し S が実行可能解であるとする． 定義 *2* や *FSABS* の入力に対する条件を適切に修正し，区間 $[0,1]$ ではなく $[0,a]$ に対し問題を定義したとする．以下で与えられる A' 及び S' に対し， S' は入力 $(m,U,A'$ に対する実行可能解となり， $\tau'(S) = a\tau'(S')$ となる．

$$A'_i = aA_i \tag{8}$$

$$(\alpha'_{i,j}, \beta'_{i,j}) = (a\alpha_{i,j}, \beta_{i,j}) \tag{9}$$

ただし， $(\alpha_{i,j}, \beta_{i,j})$ を S の元とし， $(\alpha'_{i,j}, \beta'_{i,j})$ を S' の元とする．

3.2 preliminary observations 修正が必要

本節では FSABS と BS の解の形の互換性について説明した上で， FSABS の下界を定める．

問題の定義上， BS のインスタンス集合は意味的に FSABS のインスタンス集合の部分集合である．すなわち，任意の BS のインスタンス (m,U) とそれに対する実行可能なスケジュール (M,X) に対し $\forall i,\ A_i = 0$ と置くと， $\tau(M,X) = \tau'(m,U,S)$ となるような実行可能な自由スケジュール S が存在する．この変換を行う写像を次の補題で定義する．

補題 4. *BS* の入力 (m,U) に対し (M,X) を任意のスケジュールとし， A を m 個の *0* からなる列とする．以下の条件を満たす自由スケジュール S が存在する．

1. (M,X) が入力 (m,U) に対し実行可能であるならば S も入力 (m,U,A) に対し実行可能である．

2. (M,X) が実行可能でないならば S も実行可能ではない．

3. (M,X) が実行可能なとき，$\tau(M,X) = \tau'(S)$.

Proof. f を次のように定義する．X の要素が n 個のとき， S をの要素を

$$(\alpha_{i,j}, \beta_{i,j}) = (x_j, M_{i,j}) \tag{10}$$

とする． S において明らかに $n_i = n$ となり，

$$\sum_{j}^{n_i} \alpha_{i,j} = \sum_{j}^{n} x_j \tag{11}$$

$$= 1 - A_i \tag{12}$$

$$= 1 \tag{13}$$

となる．したがって 定義 2 の条件 1 は常に満たされる．さらに， S に対し明らかに

$$\rho_{i,j} = \left\{ \sigma_{i-1,j-1} \right. \tag{14}$$

及び

$$\beta_{i,j} = M_{i,j} \tag{15}$$

が成り立つ．したがって，定義 1 の条件 1 と 定義 2 の条件 2 は同値である．同様に， $\chi_{i,j} \cap \chi_{i',j'} \neq \emptyset$ は $j = j'$ と同値であるため，定義 1 の条件 2 と 定義 2 の条件 3 も同値である．最後に， $\sigma_{i,j} \leq \rho_{i',j'}$ ならば $j < j'$ となり， $t'(\rho_{i',j'}) = t'(\sigma_{i',j'-1})$ となるため， $j' = j - 1$ と仮定すると 定義 2 の条件 4 は 定義 1 の条件 3 を含意する．他方で，帰納法を用いることでその逆も示すことができる．また， 定義 2 の条件 5 と 定義 1 の条件 4 も自明に同値となる．

以上の議論を以て，主張 1 と 2 が正しいと言える．主張 3 に関しては $t'_i(1) = t_i(X, M)$ が成り立つことで容易に満たされることが分かる． □

補題 4 は FSABS の解法アルゴリズムで用いることとなる．

続いて，いくつかの調整を加えることで任意の実行可能な自由スケジュールを実行可能なスケジュールへ変換できることを示す．その準備として， FSABS と BS の入力となる U に対し，元 $u_{-1} = 0$ が常に存在すると仮定する．

補題 5. *FSABS の入力 (m, U, A) に対し S を任意の実行可能な自由スケジュールとする．以下の条件を満たすスケジュール (M, X) が存在する．*

1. (M, X) *は実行可能である．*
2. *各 A_i に対し， $\sum_{j=1}^{k_i} x_j = A_i$ となる k_i が存在する．ただし $A_i = 0$ の場合は $k_i = 0$ とする．*
3. $t_{i,k_i}(M, X) = 0$.
4. $\tau(M, X) = \tau'(S)$.

Proof. (M, X) を次のように構成する．まず， S に対する全ての $\rho_{i,j}$ 及び $\sigma_{i,j}$ を実数直線の区間 $[0, 1]$ 上に並べ，重複を除いた上でそれらの点による分割を X の値とする．この構成により，始点が A_i となる小区間 x_{k_i} は $\rho_{i,0}$ によって与えられ，必ず存在するので主張の条件 2 が容易に満たされることが分かる． X の元の数を n とすると M の次元も $m \times n$ と決まる．次に， x_k に対する小区間を

$$\chi'_k = \left(\sum_{j=1}^{k-1} x_j, \sum_{j=1}^{k} x_j \right) \tag{16}$$

とすると， $\chi'_k \subset \chi_{i,k'}$ のときに $M(i, k) = \beta_{i,k'}$ とする．ただし， X の構成方法より χ'_k は $\chi_{i,k'}$ に完全に含まれるか，共通部分を持たないことが保証されることに注意する．一方で $j \leq k_i$ に対し $M(i, j) = -1$ とすると

$$t_{i,k_i}(M, X) = \sum_{j=1}^{k_i} x_j \times 0 = 0 \tag{17}$$

となるので条件 3 も容易に満たされる．

定義 2 より全ての $\beta_{i,j} \neq 0$ に対し $\sigma_{i',j'} = \rho_{i,j}$ かつ $\beta_{i',j'} = \beta_{i,j}$ を満たす i' と j' が存在するが， X の構成より $\sigma_{i',j'} = \rho_{i,j}$ を満たすものは隣同士の小区間となるので 定義 1 の条件 1 が満たされる．同様な考え方を用いることで 定義 1 の他の条件も 定義 2 の条件より容易に導出される．したがって (M, X) も実行可能である．また， $t_i(M, X) = t'_{i,n_i}(1)$ となることを利用すれば，主張の条件 4 が満たされる． □

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 6.

$$\bar{\tau}'(m, U, A) \geq T(m, U, A) \tag{18}$$

$$= 1 - \frac{1}{m} \sum_{j=1}^{b} (1 - u_j) - \frac{1}{m} \sum_{i=1}^{m} A_i \tag{19}$$

$$= T(m, U) - \frac{1}{m} \sum_{i=1}^{m} A_i \tag{20}$$

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． □

また，同じく 補題 1 と同様に次の主張も容易に成り立つ．

補題 7.

$$\bar{\tau}'(m,U,A) \geq u_b \tag{21}$$

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 8. *(補題 or 定理？)* $m > b$ とする．このとき

$$\bar{\tau}'(m,U,A) \geq \bar{\tau}'(m-1,U,A_{:m-1}) \tag{22}$$

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) を補題 5 により与えられる (m,U,A) に対する最適な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 での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 $(M\prime, X\prime)$ 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．

$$\bar{\tau}'(m,U,A) \geq \bar{\tau}(m-1,U). \tag{23}$$

3.3 FSABS を解くアルゴリズム Solve-FSABS

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ければならない．*

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

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

if $\forall i,\ A_i = 1$ **then**
　return [];
$r \leftarrow$ 後方にいるライダーの数;
$f \leftarrow$ 前方にいる歩行者の数;
if $u_b > T(m-f-r, U_{:b-r})$ **then**
　$m_k \leftarrow$ Subgroup(m, $U_{:b-r}$);
　$r \leftarrow r + m - m_k$;
$(t_1, d_1) \leftarrow (\infty, \infty)$;
$(t_2, d_2) \leftarrow (\infty, \infty)$;
if $f > 0$ **then**
　$(t_1, d_1) \leftarrow$ ToNextW(m, U, A, r);
else
　$(t_1, d_1) \leftarrow$ ToEnd(m, U, A, r);
groupT $\leftarrow T(m-f-r, U_{:b-r})$;
if $r > 0$ **and** $u_{b-r} <$ groupT **then**
　$(t_2, d_2) \leftarrow$ ToNextR(m, U, A, r);
$(t, d) \leftarrow$ (NIL, NIL);
if $t_2 < t_1$ **then**
　$(t, d) \leftarrow (t_2, d_2)$;
else
　$(t, d) \leftarrow (t_1, d_1)$;
$S \leftarrow$ Move(t, m, U, A);
$A^{(1)} \leftarrow$ ApplyMovement(S, A);
$S^{(1)} \leftarrow$ [];
if $\forall i > r,\ A_i^{(1)} = 1$ **then**
　$S^{(1)} \leftarrow$ MoveRiders(m, U, $A^{(1)}$);
else if $d < 1$ **then**
　$A^{(2)} \leftarrow \frac{A^{(1)}-d}{1-d}$;
　$S^{(2)} \leftarrow$ Solve-FSABS(m, U, $A^{(2)}$);
　$S^{(1)} \leftarrow$ ScaleSchedule($S^{(2)}$, $1-d$);
return $S + S^{(1)}$;

上で再帰的に Solve-FSABS を呼び出し FS-ABS を解く．この祭，合流によって後方ライダ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より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いないが，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．なお，グループに入る自転車はライダーの自転車を含まないので， 条件 1 を活用しグループの自転車を列 $U_{:b-r}$ として表すことができることに注意する[1]． 補題 6 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 3 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．ただしその祭に入力変数 r を変更する必要があることに注意する．アルゴリズムの 第 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．まず，もし先に歩行者がまだいるならば，その歩行者と合流する点を求める．上述の通り合流点にグループも歩行者も同時に到着しなければならないので，合流点を x_1 とするとグループも歩行者もそれぞれ同じ時間でそこ到達しなければならない． 主張 1 より $\overline{\tau}(m-f-r,U) = T(m-f-r,U)$ をグループの速度として考えることができるので，出発点に一番近い歩行者を人 p とするとそれぞれが合流点までの移動にかかる時間を以下の式で表すことができる．

$$t_1 = (x_1 - A_p) \times 1 = x_1 T(m-f-r, U_{:b-r}) \quad (24)$$

x_1 について解くと合流点が求まる．

$$x_1 = \frac{A_p}{1 - T(m-f-r, U_{:b-r})} \quad (25)$$

もし $x_1 \geq 1$，つまり合流点が到着点以降になるのであれば， $x_1 = 1$ と置いて到着点までの移動だけを考える．この実際の移動距離を $d_1 = \min\{x_1, 1\}$ と置く．以上の計算を行うのが 第 11 行 の関数 toNextW の役割である．他方でもし先に歩行者がいなければ， $x_1 = 1$ の場合と同様に計算すれば良い．その場合を賄うのが 第 13 行 の関数 toEnd の役割である．

次に，もし後ろにライダーがいて，且つライダーがグループより速く動くのであれば，ライダーと先に合流できないかを調べる必要がある．その際には上記と同様な方法で合流時間

$$t_2 = (x_2 - A_r) u_{b-r} = x_2 T(m-f-r, U_{:b-r}) \quad (26)$$

から合流点 x_2 について解くことで値が求まる．

$$x_2 = \frac{A_r u_{b-r}}{u_{b-r} - T(m-f-r, U_{:b-r})} \quad (27)$$

この部分が 第 16 行 の関数 toNextR の役割である．最後に，ライダーと先に合流するか，歩行者と先に合流するか，両方よりも先に到着点に到達するかを時間的に比較した上で，一番早くできる行動を採用して，それにかかる移動を時間 t 及び距離 d で表す．

移動距離を決めてから次にその移動を実行するためのスケジュール S を 第 22 行 で求める．手続き Move は各人 i に対し次のようなスケジュールを構成し，それらをまとめたものを返す．

- 人 i が既に到着点にいる ($A_i = 1$) ならば，その人の自由スケジュールを空列とする．

[1] このような表記は Python 言語に基いており， U の中で $b-r$ 未満の番号を持つ元を含めた部分列という意味である．

- 人 i がライダー $(0 < i \leq r)$ ならば，その人の自由スケジュールを一つの順序対 $(tu_{b-i}, b-i)$ からなる配列とする．

- 人 i がグループの人 $(r < i < m-f)$ ならば，グループ全体のスケジュール (M, X) を BS の解として求め，主張 1 を用いて距離 $\frac{t}{T(m-f-r, U_{:b-r})}$ に合わせた上で，補題 4 により与えられる (M, X) に相当する自由スケジュール S' を組み立て，人 i の自由スケジュールを S'_i とする．

- 人 i が歩行者 $(m-f < i < m)$ ならば，その人の自由スケジュールを一つの順序対 $(\min\{1-A_i, t\}, 0)$ からなる配列とする．

時間 t 分の移動スケジュールを求めた上で，その移動を A に対する操作として施し，A の値にそれぞれの人の移動距離を加算したものを $A^{(1)}$ に格納する (第 23 行).

次に移動後の状態を改めて分析した上で次の挙動を決める．もしグループの移動距離が全区間ならば，前方の歩行者とグループが必ず到着点に到達したと解釈できる．その祭，$i > r$ に対し $A_i^{(1)}$ の値は 1 となる．しかし，もしライダーがいる場合にはライダーがまだ到着点に到達していない可能性があり，ライダーのスケジュールにそれぞれが到着するまでの分を追加しなければならない．この場合に対応するのが 第 25 行 の条件分岐である．

他方で，もしグループが全区間を移動していないのであれば，少なくともグループのメンバーはまだ到着点に到達していない．その祭，全員の相対位置をスケールした上で，残りの区間に対し新たに Solve-FSABS を用いて問題を解く．ただし，相対位置を計算するときには以下の式を用いる．

$$A_i^{(2)} = \begin{cases} \frac{A_i^{(1)}-d}{1-d} & A_i^{(1)} < 1 \\ A_i^{(1)} & A_i^{(1)} = 1 \end{cases} \tag{28}$$

この処理を施し，Solve-FSABS を再び呼び出した結果が自由スケジュール $S^{(2)}$ となり，アルゴリズムが正当であると仮定すると $S^{(2)}$ の移動を施せば全員が到着点に到達する．しかし，$S^{(2)}$ では移動区間を $[0,1]$ としているので，元の問題に適用するには全員の移動を区間 $[d, 1-d]$ に合うようにスケールしなければならない．すなわち，$S^{(2)}$ の全ての順序対に対し以下の順序対

$$(\alpha_{i,j}^{(1)}, \beta_{i,j}^{(1)}) = ((1-d)\alpha_{i,j}^{(2)}, \beta_{i,j}^{(1)}) \tag{29}$$

を求め，それらを $S^{(1)}$ にまとめる．この処理が 第 27 行 の条件分岐に相当する．

最後に，合流までのスケジュール S と合流後のスケジュール $S^{(1)}$ を組み合わせたものを返せば良い．ただし，組み合わせるとはそれぞれの i に対し配列 S_i 及び $S_i^{(1)}$ を連結させる処理とする．

3.4 Solve-FSABS の正当性及び計算時間

本節では Solve-FSABS が必ず実行可能解を出力し，その実行可能解が必ず以前定義した下界を満たすため最適解であること及び，$O(P(m,n,b))$ 時間で終了することを示す．

まず最初に，2 つのスケジュールを「横に組み合わせる」という操作を次の補題で定義する．

補題 9. 任意の入力 $(m, U, A^{(1)})$ 及び $(m, U, A^{(2)})$ それぞれに対し自由スケジュール $S^{(1)}$ 及び $S^{(2)}$ を考える．$S^{(1)}$ に対し 定義 *2* の条件 *1* 以外が成り立つとし，$S^{(2)}$ が実行可能である *(*条件が全て成り立つ*)* とする．もしとある定数 d 及び $S^{(1)}$ の全ての行 $S_i^{(1)}$ に対し

1. $A_i^{(1)} + L(S_i^{(1)}) = \frac{1}{1-d}(A_i^{(2)} - d)$,

2. $A_i^{(2)} < 0$ ならば $\beta_{i,1}^{(2)} = \beta_{i,n_i^{(1)}}^{(1)}$

が成り立つなら，以下で生成される $S^{(3)} = S^{(1)} + S^{(2)}$ は $(m, U, A^{(1)})$ に対する実行可能解である．

$$(\alpha_{i,j}^{(3)}, \beta_{i,j}^{(3)}) = \begin{cases} (\alpha_{i,j}^{(1)}, \beta_{i,j}^{(1)}) & j \leq n_i^{(1)} \\ (\frac{1}{1-d}\alpha_{i,j'}^{(2)}, \beta_{i,j'}^{(2)}) & otherwise \end{cases} \tag{30}$$

ただし，$j' = j - n_i^{(1)}$ とし，$S^{(3)}$ の各行の要素数を $n_i^{(3)} = n_i^{(1)} + n_i^{(2)}$ とする．

Proof. 定義 2 の条件を順番に確認する． $S^{(3)}$ の各行に対し

$$L(S_i^{(3)}) = L(S_i^{(1)}) + \frac{1}{1-d}L(S_i^{(2)}) \tag{31}$$
$$= \frac{1}{1-d}(A_i^{(2)} - d + L(S^{(2)})) - A_i^{(1)} \tag{32}$$
$$= \frac{1}{1-d}(1-d) - A_i^{(1)} \tag{33}$$
$$= 1 - A^{(1)} \tag{34}$$

となるので条件 1 が満たされる． 2 より $S^{(2)}$ における各自転車の初期値が後方ライダーもしくはグループの出発点に決まるので， $S^{(2)}$ の実行可能性より $S^{(3)}$ に対しても条件 2 が成り立つ．条件 3 はそれぞれのスケジュールで満たされるため，確認すべき部分は $S^{(1)}$ と $S^{(2)}$ が重複する区間である． $S^{(1)}$ において指定される移動で人 i_1 が自転車 u' を区間 $[x, y]$ で利用し， $S^{(2)}$ において人 i_2 が同じ自転車を同じ (もしくはそれを含む) 区間で利用したとすると 2 と矛盾する．なぜなら $S^{(1)}$ と $S^{(2)}$ が重複している区間が $S^{(2)}$ において出発点以前にあり，少なくとも出発点までライダーは自転車を変えられないからである．同様な議論をすることにより条件 4 も同じく満たされることが分かる．また，条件 5 は $S^{(2)}$ がそれを満たすことにより $S^{(3)}$ も必然的にそれを満たす． □

Solve-FSABS の出力を S とすると， S が 定義 2 の条件を満たすことを示せば良い．このことを前方歩行者及び後方ライダーの数に対する帰納法を用いて示す．まず，前方も後方も人がいない場合の挙動を考える． $u_b \leq T(m, U)$，すなわち部分グループを考える必要がないのであれば， 第 13 行 の処理より全ての人及び自転車が全区間を移動することとなる．その際の自由スケジュール S は BS の解として与えられるので， 補題 4 に S は実行可能である．また，全員が一気に動くから $A_i^{(1)} = 0$ となり，ライダーもいない状態であるため $S^{(1)}$ は空列のままとなる．したがってこのような入力に対し Solve-FSABS の出力は BS の解に対する自由スケジュールであり，実行可能である．ただし， $\tau'(S) = \tau(M, X) = \bar{\tau}(m, U)$ であることから 補題?? の下界を満たすことに注意する．

他方で $u_b < T(m, U)$ の場合を考えると，第 13 行及び 第 22 行の処理において S の値は部分グループが到着点まで移動する時間の分のみとなり，余ったライダー達の残りの移動は指定されない．しかし， 第 25 行 の条件が成立するので， $S^{(1)}$ の値にライダー達の到着点までの移動が格納される． S と $S^{(1)}$ を合併させたときにそれぞれの人の移動は次のように与えられる．部分グループの移動を BS の解 (M, X) から $S^{(sub)}$ に変換したとすると，Solve-FSABS の出力 $S^{(out)}$ は以下の要素から構成される．

$$S_i^{(out)} = \begin{cases} [(1, u_{b-i})] & i \leq r \\ S_i^{(sub)} & i > r \end{cases} \tag{35}$$

ライダーの分と部分グループの分に対し独立に 定義?? の条件が成り立つので， 命題 1 より $S^{(out)}$ は実行可能であり．到着時間が $\tau'(S^{(out)}) = \bar{\tau}(m, U) = u_b$ となり， 補題 7 が満たされる．

続いて，あらゆる m と U に対し後方に r 人以下のライダーと前方に f 人以下の歩行者がいたときに Solve-FSABS の出力が実行可能だったと考え，ライダー一人及び自転車一台を追加したときの挙動を考える．ただし，追加する自転車は新たに最も遅い自転車であると仮定し， 条件 1 より追加したライダーが一番後ろにいると仮定する．

追加したライダーを人 1 とし，その位置を $A_1 < 0$ とする．もし現在の呼び出しでグループが人 1 と合流しなければ，グループ及び前方の歩行者のスケジュールは人 1 がいないときと同様になり，他方で人 1 のスケジュールには単なる一定区間の移動が追加される．移動が終わったあとにもしライダー以外全員到着している場合，人 1 の残りの移動は 第 26 行 により与えられるが，他の人の移動は人 1 がいないときと変わらない．すなわち，他の人達に関して 定義 2 は帰納法の仮定より満たされている．しかし人 1 と他の人がお互いの自転車を共有する機会が一切ないので人 1 に関しても 定義 2 の条件 2, 3, 4 が満たされ，また移動の与え方から 1 と 5 も自明に満たされる．したがって，この場合の出力は実行可能

であり，到着時間が $\tau'(S) = u_{b+1}(1 - A_1)$ となる．一方でもしライダー以外にもまだ到着していない人がいるならば再帰的な呼び出しによりスケジュール $S^{(2)}$ が得られるので， $S + S^{(2)}$ が実行可能であることを確認しなければならない．しかし， $A^{(2)}$ の計算方法より 補題 9 が成立し， $S^{(out)}$ は実行可能解となる．同様に，人 1 がグループと合流した場合も， S と $S^{(2)}$ に対し 補題 9 の条件が成り立つので $S^{(out)} = S + S^{(2)}$ が実行可能解となる．なお，ライダーが一人でも現れたら，それ以降ライダーが増えることがないことに注意する．

次に，あらゆる m と U に対し後方に r 人以下のライダーと前方に f 人以下の歩行者がいたときに Solve-FSABS の出力が実行可能だったと考え，前方に歩行者を一人追加したときの挙動を考える．ただし，追加する歩行者を人 $m+1$ とし，新たに最も到着点に近い位置にあるとする．もし現在の呼び出しでその人がグループと合流せずにそのまま到着点に達したら，その人のスケジュールが他の人と独立し実行可能となるため 命題 1 より全体のスケジュールも実行可能となる．また到着点に到達しなかった場合及びグループと合流した場合に，再帰的な処理によってライダーげ減るか，前方の歩行者が減るかのいずれの状態で再帰的な呼び出しが発生するが，上記同様に 補題 9 より最終的なスケジュールが実行可能となる．

上述の結果を以下の定理にまとめる．

定理 1. Solve-FSABS は常に実行可能解を出力する．

今度は出力が実行可能であることを踏まえ，Solve-FSABS が最適解を出力することを示す．まず，自由スケジュールの定義上人が常に前進するので，最後に到着する人が誰かにより下界のいずれかが満たされることを示せば良い．もし前方の歩行者の一部が先に到着すれば，その人達のスケジュールが独立となるので， 命題 1 より 補題 8 が満たされることが分かる．したがって，先に到着した人達を除いた入力に対し下界を考えれば良い．Solve-FSABS の挙動では上記の場合を除いて，ライダーが最後に到着するか，グループと合流しグループが同時に到着するかのいずれかが起こる．もしライダーが最後だった場合， 条件 1 より u_b に乗っている人が最後になるため，全体の到着時間は $\tau'(S^{(out)} = u_b = \bar{\tau}(m-f, U)$ となる．ただし f は元々前方にいた人数を指す．他方で全員が同時に到着すれば 補題 6 より $\tau'(S^{(out)}) = T'(m - f', U, A')$ となる．ただし f' は先に到着できた歩行者の人数を指し， A' は元々の入力 A から先に到着する前方歩行者のデータを取り除いたものである．この議論を以下の定理にまとめる．

定理 2. 入力 (m, U, A) に対し Solve-FSABS が出力する実行可能な自由スケジュールを S とする．S に対する到着時間は以下を満たし最適である．

$$\tau'(S) = \max \begin{cases} \bar{\tau}(m-f, U) \\ T'(m-f', U, A') \end{cases} \tag{36}$$

ただし， f' は先に到着できる前方の人数を指し， A' はそれらの分を取り除いた入力とする．

最後に， Solve-FSABS の計算量について述べる． アルゴリズム 1 の処理は再帰呼び出しを除き概ね以下の計算オーダーに分かれる．

- $O(b)$: 部分グループの構成 () や T の計算， 第 26 行
- $O(m)$: $A^{(i)}$ の計算
- $O(mn)$: 第 22 行 及び 第 30 行 (ただし n は小区間の数であり， BS の解に依存する)
- $O(1)$: その他

再帰自体は最大でも $O(m)$ 回しか起こらないので，全体の計算量は $O(m^2 n)$ で与えられる．

4 結論と今後の課題

本論文では自転車共有問題 (BS) について紹介し，その拡張として考えられる前方に人がいることを許

した自転車共有問題 (FSABS) を定義した．また，FSABS の解法としてアルゴリズム Solve-FSABS を与え，部分問題として BS を解くことによって FSABS が多項式時間で解けることを示した．

今回紹介したアルゴリズムは計算途中の状態として condrefriders-order の下で出発点より後ろにライダーがいること，つまり FSABS よりさらに拡張された入力を許している．このような入力を許した問題は本論文の範囲外であるが，我々は Solve-FSABS と同様な方法で解くことができると予想している．また後退を許さない限り，前後関係なく任意の位置に任意の人及び自転車を置いた場合も同様な方法で最適なスケジュールが得られると考えられる．

他方で後退を許し，前方の他に後方に人や自転車を置いた問題も考えることができる．その場合，移動距離の上界がなくなるため，最適な到着時間の値を直接計算することがより難しくなり，解法アルゴリズムを求めることもより難しいと考えられる．その一方で後退を許した方が現実に応用できるので，より価値がある問題だと考えられる．