

ICS-0XM-XXX

前方に人がいることを許した自転車共有問題

指導教員 山田 敏規教授

令和3年2月10日提出

工学部情報工学科

18TI018

ハララノフ ヴァレリ

埼玉大学 理工学研究科・工学部
山田研究室
埼玉県さいたま市桜区下大久保255

第1章 はじめに

自律移動ロボットは、多くの製造、倉庫保管、およびロジスティクスアプリケーションでますます使用されている。最近では特に、人間と協調して近接して動作することを目的とした、いわゆるコボット (協働ロボット) の展開への関心が高まっている [4154820, wiki, 6386300]。このようなコボットは人間によって制御されるが、一緒に作業する人間の能力を増強・強化することを目的としている。

本論文では、輸送問題へのコボットの応用について考える。この問題では、協調する自律移動エージェント (人間またはロボット) は輸送コボット (本論文では自転車と呼ばれる) の使用時にエージェントの速度を上げることで輸送コボットから支援を受ける。エージェントは自律的で、能力が等しく、最高速度1で歩くことができる。自転車は自律的ではなく、自分で移動することはできないが、エージェントは自転車に乗って移動することができる。エージェントはいつでも高々1台の自転車に乗ることができ、自転車は高々1人のエージェントを乗せることができる。自転車 i に乗っているエージェントは速度 $v_i > 1$ で移動できる；バイクが異なれば速度も異なるかもしれないことに注意されたい。自転車は2つの役割を果たす：エージェントが自分の速度を上げるために利用できるリソースであるが、輸送する必要のある商品でもある。 m 人のエージェントと b 台の自転車があると、自転車 $1, 2, \dots, b$ の速度はそれぞれ $v_1 \geq v_2 \geq \dots \geq v_b > 1$ であるとする。また、エージェントと自転車の初期位置は全て単位区間 $[0, 1]$ にあるとし、エージェント i ($1 \leq i \leq m$) の初期位置は A_i であり、自転車の初期位置は0であるとする。ただし、 $A_i = 0$ であるエージェント i が少なくとも $b+1$ 人いると仮定する。エージェントの目標は、エージェントと自転車の最後の到着時間が最小となるように、自転車を輸送しながら目的地1へ到達することである。

Czyzowicz ら [8] は、全てのエージェント i に対して $A_i = 0$ である問題 (この問題を自転車共有問題、略して BS と呼ぶ) の最適解を求める多項式時間アルゴリズムを開発した。本論文では、 $A_i > 0$ であるエージェント i が存在することを許した問題 (これを前方に人がいることを許した自転車共有問題、略して FSABS と呼ぶ) の最適解を求める多項式時間アルゴリズムを提案する。

本論文の構成は以下の通りである．第 2 章 では文献 [8] による BS の定義を与え，最適解の下界と，それを達成する実行可能解すなわち最適解を求める多項式時間アルゴリズムについて紹介する．第 3 章 では FSABS を定義し，その解に関わるいくつかの性質を述べてから，BS と同様な方式で下界を定める．その次に，下界を達成する実行可能解すなわち最適解を求めるアルゴリズム SOLVE-FSABS を提案し，SOLVE-FSABS が実際に最適解を出力すること，および，SOLVE-FABS の時間計算量が人数 m の多項式で与えられることを示す．最後に第 4 章 では本論文の結果を踏まえながら，BS に対するさらなるいくつかの拡張を考え，それらに対する予想を述べる．

第2章 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 [8]. 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 であるとき, $u_i = \frac{1}{v_i}$ をその逆数として配列にまとめたもの.

ここで b は自転車の台数である. さらに, $b < m$ となるような入力のみを考えることとし, U は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ず, 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ならない. Czyzowicz ら [8] により提案されたアルゴリズムはそのような特別な共有パターンを求める.

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, $M = (M_{i,j})$ 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 j 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 $\widehat{M}_{i,j} = u_{M_{i,j}}$ 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

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

- 人 i が到着点に到着するのに必要な時間

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

- 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 $\bar{\tau}(m, U)$ 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.

1. $M_{i,j} \neq 0 \implies \exists i' \text{ s.t. } M_{i',j-1} = M_{i,j}.$
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. $1 \leq \forall j \leq b, \exists i \text{ s.t. } M_{i,n} = j.$

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 [8] 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.

$$\bar{\tau}(m, U) \geq u_b. \quad (2.1)$$

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. \square

補題 2.

$$\bar{\tau}(m, U) \geq T(m, U) \stackrel{\text{def}}{=} 1 - \frac{1}{m} \sum_{j=1}^b (1 - u_j) \quad (2.2)$$

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. \square

補題 2 では人が常に動いている仮定が必要であるが, Czyzowicz ら [8] は待機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 [8]) (m, U) を BS の入力とする. $u_b \geq T(m, U)$ であるならば式 (2.3) を満たす k が存在する. ただし, $U_k = [u_1, u_2, \dots, u_k]$ 及び $m_k = m - b + k$ であるとする.

$$u_k \leq T(m_k, U_k) \leq u_b \quad (2.3)$$

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 [8] 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， $\tau(M, aX) = a\tau(M, X)$. \square

第3章 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 個の要素からなる配列であり, 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.

$$\rho_{i,j} = A_i + \sum_{k=1}^{k < j} \alpha_{i,k} \quad (3.1)$$

$$\sigma_{i,j} = \rho_{i,j} + \alpha_{i,k} \quad (3.2)$$

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

$$\tau'(S) = \max_i t'_i(1) \quad (3.3)$$

と定義できる．また、 S において人 i が移動した距離を

$$L(S_i) = \sum_{j=1}^{n_i} \alpha_{i,j} \quad (3.4)$$

と表す．

次に S が実行可能解であるために Czyzowicz ら [8] 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かつそのときに限り S を実行可能な自由スケジュールと呼ぶ．

1. $1 \leq \forall i \leq m, L(S_i) = 1 - A_i$.
2. $\beta_{i,j} \neq 0$ かつ $\rho_{i,j} \neq 0$ 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. $1 \leq \forall j \leq b, \exists i \text{ s.t. } \beta_{i,n_i} = j$.

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、入力 (m, U, A) 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、以下の主張でその正当性を保証する．

主張 2. $(m_1, U_1 = [u_1, u_2, \dots, u_{b_1}], A_1)$ 及び $(m_2, U_2 = [u_{b_1+1}, u_{b_1+2}, \dots, u_{b_1+b_2}], A_2)$ 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り, その到着時間が式 (3.5) となる.¹

$$\tau'(S^{(3)}) = \max\{\tau'(S^{(1)}), \tau'(S^{(2)})\} \quad (3.5)$$

□

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 3. 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.

$$A'_i = aA_i \quad (3.6)$$

$$(\alpha'_{i,j}, \beta'_{i,j}) = (a\alpha_{i,j}, \beta_{i,j}) \quad (3.7)$$

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. なお, 次の 2 つの主張を確認することは難しくないが, 証明がややテクニカルで長いため, ここでは省略する.

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り, BS の入力 (m, U) に対し $A_i = 0$ と置くと FSABS における同様なインスタンスが得られる. また, BS において実行可能なスケジュール (M, X) に対し以下の要素からなる自由スケジュール S を考える.

$$(\alpha_{i,j}, \beta_{i,j}) = (X_j, M_{i,j}) \quad (3.8)$$

X の要素数を n とすると, S は m 行からなり, 各行がちょうど $n_i = n$ 個の要素を持つ. このとき, 次の主張が自然に得られる.

¹ただし $U_1 + U_2$ は U_1 と U_2 の連結を意味し, 以降の議論でも配列同士の加算記号を連結の意味で用いる.

主張 4. BS の入力 (m, U) に対し任意の実行可能解 (M, X) 及び上記の方法で定義した S を考える. S に対し次が成り立つ.

- m 個の 0 からなる A に対し, S は FSABS の入力 (m, U, A) に対する実行可能解である.
- $\tau'(S) = \tau(M, X)$.

□

BS の入力では前方にいる人を表現することができないが, 任意の実行可能な自由スケジュール S を (M, X) の形に変換することができる. S に対し次の集合を考える.

$$I = \{\rho_{i,j} : i \leq m, j \leq n_i\} \setminus \{0\} \quad (3.9)$$

I の要素数を $n-1$ とする. 区間 $[0, 1]$ を I の要素で切って分割すると n 個の小区間ができる. それらの長さを X の要素とし, γ_k を X_k に対応する小区間とする. 各 i に対し γ_k はちょうど一つの χ_{i,j_k} に含まれ, すなわち $\gamma_k \subset \chi_{i,j_k}$ である. このことを利用し, 以下のように M を定義する.

$$M_{i,k} = \beta_{i,j_k} \quad (3.10)$$

最後に前方にいる人を考慮する必要がある. S の定義より $\rho_{i,1} = A_i$ であるため, 各 i に対しちょうど A_i で始まる小区間 γ_{k_i} が存在するので, $t_{i,k_i-1}(X, M) = 0$ となるように M の要素を設定する必要がある (ただし $t_{i,0}(M, X) = 0$ とする). 入力 U に関わらず $u_{-1} = 0$ と定義しておく, 前方にいる歩行者に対し以下のように M の要素を設定できる.

$$M_{i,k} = \begin{cases} -1 & k < k_i \\ \beta_{i,j_k} & k \geq k_i \end{cases} \quad (3.11)$$

上述同様に次の主張が自然に得られる.

主張 5. FSABS の入力 (m, U, A) に対し任意の実行可能な自由スケジュール S 及び上記の方法で定義した (M, X) を考える. (M, X) に対し次が成り立つ.

- (M, X) は 定義 1 を満たし実行可能である.
- $\tau(M, X) = \tau'(S)$.

□

本節で定める FSABS の下界を示すのに主張 5 を用い、次節で与える解法アルゴリズムで主張 4 を用いる。

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 4.

$$\bar{\tau}'(m, U, A) \geq T'(m, U, A) \quad (3.12)$$

$$\stackrel{\text{def}}{=} 1 - \frac{1}{m} \sum_{j=1}^b (1 - u_j) - \frac{1}{m} \sum_{i=1}^m A_i \quad (3.13)$$

$$= T(m, U) - \frac{1}{m} \sum_{i=1}^m A_i \quad (3.14)$$

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。□

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_1 から A_k までの要素を含んだ配列とする。

補題 5. $m > b$ とする。このとき

$$\bar{\tau}'(m, U, A) \geq \bar{\tau}'(m-1, U, A_{:m-1}) \quad (3.15)$$

Proof. 背理法のために $\bar{\tau}'(m, U, A) < \bar{\tau}'(m-1, U, A_{:m-1})$ であると仮定し、 S を (m, U, A) に対する最適な自由スケジュールとし、 (M, X) を主張 5 により得られた S に対応するスケジュールとする。また、新たな記号を導入せずに本証明に限り $\tau'(S)$ の代わりに $\tau'(M, X, A)$ と書くことにする。

人 m がはじめて自転車に乗る小区間を i とし、その始点において人 m が乗る自転車 u_j に注目する。その自転車には人 α_1 が小区間 $i-1$ で乗っており、人 m が乗る前に小区間 i の始点まで運んでくれたはずである。同様に人 α_1 が小区間 i で自転車に乗っていたと仮定する。そのとき、小区間 $i-1$ でこの自転車に乗っていた人 α_2 存在する。これを繰り返すと、人 α_{k-1} が小区間 i で乗っていたのと同じ自転車を小区間 $i-1$ で乗っていた人 α_k が存在し、人 α_k は小区間 i で歩いていたような k が存在する。このとき、人 $m, \alpha_1, \dots, \alpha_k$ の小区間 i 以降のスケジュールを次のように変更する。

- 人 α_1 は人 m のスケジュールを実行する.
- 人 α_i は人 α_{i-1} のスケジュールを実行する ($2 \leq i \leq k$).
- 人 m は A_m を小区間 i の始点とした上で人 α_k のスケジュールを実行する.

このとき, 人 $\alpha_1, \dots, \alpha_k$ の新しいスケジュールはそれぞれ人 $m, \alpha_1, \dots, \alpha_{k-1}$ の元のスケジュールより遅れることはない. また, 人 m は小区間 1 から $i-1$ では自転車に乗っていないので, A_m を小区間 i の始点に動かしても自転車への影響を与えることはなく, 時刻 0 で小区間 i の始点を出発できるので, 人 m の新しいスケジュールは人 α_k の元スケジュールより遅れることはない. ここで, もし小区間 i 以降で自転車よりも人が先に到着した場合はそこで待つこととする.²

このような再スケジューリングを繰り返すことにより, 元々のスケジュールに要する時間を増やすことなく人 m が初めて自転車に乗る小区間を到着点側へ動かすことができ, 最終的には自転車に乗らないスケジュールを求めることができる. このときに得られる新たな初期位置 A' とスケジュール (M', X') に対し $\tau'(M', X', A') \leq \tau'(M, X, A)$ であり, 人 m とその他のスケジュールを分けることにより 主張 2 から $\bar{\tau}'(M', X', A') \geq \bar{\tau}'(m-1, U, A_{m-1})$ が得られるので,

$$\bar{\tau}'(m, U, A) = \tau'(M, X, A) \quad (3.16)$$

$$\geq \tau'(M', X', A) \quad (3.17)$$

$$\geq \bar{\tau}'(m-1, U, A_{m-1}) \quad (3.18)$$

より矛盾が生じる.

□

この結果から以下の系が容易に得られる.

系 2. $i \leq m - f$ に対し $A_i = 0$ とする.

$$\bar{\tau}'(m, U, A) \geq \bar{\tau}(m - f, U). \quad (3.19)$$

以上の議論を次の定理にまとめる.

²自由スケジュールの定義では「待つ」という行動は実現できないが, 主張 5 と同様な方法で容易に追加することができる.

定理 1. (m, U, A) を FSABS の入力とする．また f を前方にいる人の数とし，FSABS における最適な到着時間 $\bar{\tau}'(m, U, A)$ は次の下界により制限される．

$$\bar{\tau}'(m, U, A) \geq \max \begin{cases} T'(m, U, A) \\ T'(m-1, U, A_{:m-1}) \\ T'(m-2, U, A_{:m-2}) \\ \vdots \\ T'(m-f+1, U, A_{:m-f+1}) \\ \bar{\tau}(m-f, U) \end{cases} \quad (3.20)$$

Proof. 補題 5 に 補題 4 と 系 2 を適用すれば良い． □

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．

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  $[]$ ;
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    $(t_1, d_1) \leftarrow (\infty, \infty)$ ;
9    $(t_2, d_2) \leftarrow (\infty, \infty)$ ;
10 if  $f > 0$  then
11    $(t_1, d_1) \leftarrow \text{TONEXTW}(m, U, A, r)$ ;
12 else
13    $(t_1, d_1) \leftarrow \text{TOEND}(m, U, A, r)$ ;
14  $\text{groupT} \leftarrow T(m - f - r, U_{:b-r})$ ;
15 if  $r > 0$  and  $u_{b-r} < \text{groupT}$  then
16    $(t_2, d_2) \leftarrow \text{TONEXTR}(m, U, A, r)$ ;
17  $(t, d) \leftarrow (\text{NIL}, \text{NIL})$ ;
18 if  $t_2 < t_1$  then
19    $(t, d) \leftarrow (t_2, d_2)$ ;
20 else
21    $(t, d) \leftarrow (t_1, d_1)$ ;
22  $S \leftarrow \text{MOVE}(t, m, U, A)$ ;
23  $A^{(1)} \leftarrow \text{APPLYMOVEMENT}(S, A)$ ;
24  $S^{(1)} \leftarrow []$ ;
25 if  $\forall i > r, A_i^{(1)} = 1$  then
26    $S^{(1)} \leftarrow \text{MOVERIDERS}(m, U, A^{(1)})$ ;
27 else if  $d < 1$  then
28    $A^{(2)} \leftarrow \frac{A^{(1)} - d}{1 - d}$ ;
29    $S^{(2)} \leftarrow \text{SOLVE-FSABS}(m, U, A^{(2)})$ ;
30    $S^{(1)} \leftarrow \text{SCALESCCHEDULE}(S^{(2)}, 1 - d)$ ;
31 return  $S + S^{(1)}$ ;

```

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に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くても、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．補題 4 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、一定区間動かしただけで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．ただしその際に入力変数 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 $\bar{\tau}(m-f-r, U) = T(m-f-r, U)$ をグループの速度として考えることができるので、出発点が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．

$$t_1 = (x_1 - A_p) \times 1 = x_1 T(m-f-r, U_{:b-r}) \quad (3.21)$$

x_1 について解くと合流点が求まる.

$$x_1 = \frac{A_p}{1 - T(m - f - r, U_{:b-r})} \quad (3.22)$$

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

$$t_2 = (x_2 - A_r)u_{b-r} = x_2 T(m - f - r, U_{:b-r}) \quad (3.23)$$

から合流点 x_2 について解くことで値が求まる.

$$x_2 = \frac{A_r u_{b-r}}{u_{b-r} - T(m - f - r, U_{:b-r})} \quad (3.24)$$

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.
- 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, 主張 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。その際、 $i > r$ 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。その際、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。

$$A_i^{(2)} = \frac{A_i^{(1)} - d}{1 - d} \quad (3.25)$$

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

$$(\alpha_{i,j}^{(1)}, \beta_{i,j}^{(1)}) = ((1 - d)\alpha_{i,j}^{(2)}, \beta_{i,j}^{(1)}) \quad (3.26)$$

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、および m に対する多項式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 6. 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. さらに, それぞれの自由スケジュールに対し以下の仮定が成り立つとする.

1. とある定数 c 及び全ての i に対し $t_i(L(S_i^{(1)})) = c$ である.
2. とある定数 d 及び全ての i に対し $A_i^{(1)} + L(S_i^{(1)}) = \frac{1}{1-d}(A_i^{(2)} - d)$ である.
3. $A_i^{(2)} < 0$ であるならば $\beta_{i,1}^{(2)} = \beta_{i,n_i^{(1)}}^{(1)}$ であり, 且つ $A^{(2)}$ に対し 条件 1 が成り立つ.
4. $A_i^{(2)} > 0$ であるならば全ての j に対し $\beta_{i,j}^{(1)} = 0$.

このとき, $S^{(3)} = S^{(1)} + S^{(2)}$ は $(m, U, A^{(1)})$ に対する実行可能解である.

$$(\alpha_{i,j}^{(3)}, \beta_{i,j}^{(3)}) = \begin{cases} (\alpha_{i,j}^{(1)}, \beta_{i,j}^{(1)}) & j \leq n_i^{(1)} \\ (\frac{1}{1-d}\alpha_{i,j'}^{(2)}, \beta_{i,j'}^{(2)}) & otherwise \end{cases} \quad (3.27)$$

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

$$L(S_i^{(3)}) = L(S_i^{(1)}) + \frac{1}{1-d}L(S_i^{(2)}) \quad (3.28)$$

$$= \frac{1}{1-d}(A_i^{(2)} - d + L(S^{(2)})) - A_i^{(1)} \quad (3.29)$$

$$= \frac{1}{1-d}(1-d) - A_i^{(1)} \quad (3.30)$$

$$= 1 - A^{(1)} \quad (3.31)$$

となるので 定義 2 の条件 1 が満たされる. 仮定 3 及び 仮定 4 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ているので確認すべきところは $S^{(1)}$ と $S^{(2)}$ が重複する区間である. $S^{(2)}$ は同時出発を前提に実行可能であるため, $S^{(3)}$ において $S^{(2)}$ の部分が始まるときに全員が同時出発をすれば $S^{(2)}$ の実行可能性より条件 3 が成り立ち, 仮定 1 により同時出発が保証される. 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. \square

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、主張 4 により S は実行可能である。また、全員が同時に到着するため $A_i^{(1)} = 1$ 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、 $\tau'(S) = \tau(M, X) = \bar{\tau}(m, U)$ であることから系 2 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。

$$S_i^{(out)} = \begin{cases} [(1, u_{b-i})] & i \leq r \\ S_i^{(sub)} & i > r \end{cases} \quad (3.32)$$

ライダーの分と部分グループの分に対し独立に 定義 2 の条件が成り立つので、命題?? 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、系 2 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 6 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 6 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?? 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が減るか, 前方の歩行者が減るかのいずれの状態でも再帰的な呼び出しが発生するが, 上記同様に補題 6 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 2. 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?? より補題 5 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 4 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 3. 入力 (m, U, A) に対し SOLVE-FSABS は最適解を出力する.

Proof. SOLVE-FSABS が出力する自由スケジュールを S とする． 定理 2 より S は実行可能であり， 上述の議論より以下を満たすため， 定理 1 より最適である．

$$\tau'(S) = \max \begin{cases} \bar{\tau}(m - f, U) \\ T'(m - f', U, A') \end{cases} \quad (3.33)$$

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 の時間計算量について簡単に述べる． アルゴリズム 1 においては再帰呼び出しを除き， 全ての処理が定数時間か m に対する多項式時間で終了する． また， 再帰自体は最大でも $O(m)$ 回しか起こらないので， 全体の時間計算量は m に対する多項式時間である．

第4章 結論と今後の課題

小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 条件 1 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. 我々は前後関係なく任意の位置に任意の人及び自転車を置くことを許した問題でも SOLVE-FSABS と同様な方法で解くことができると予想している.

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, 最適な到着時間を直接計算することでさえ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.

Bibliography

- [1] Niels Agatz ゾ. “Optimization for dynamic ride-sharing: A review”. In: *European Journal of Operational Research* 223.2 (2012), pp. 295–303. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2012.05.028>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221712003864>.
- [2] Jérémie Chalopin ゾ. “Data Delivery by Energy-Constrained Mobile Agents on a Line”. In: *Automata, Languages, and Programming*. Ed. by Javier Esparza ゾ. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 423–434. ISBN: 978-3-662-43951-7.
- [3] Longbiao Chen ゾ. “Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data”. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’15. Osaka, Japan: Association for Computing Machinery, 2015, pp. 571–575. ISBN: 9781450335744. DOI: 10.1145/2750858.2804291. URL: <https://doi.org/10.1145/2750858.2804291>.
- [4] Brian Coltin and Manuela Veloso. “Towards ridesharing with passenger transfers”. In: *Proceedings of 12th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS ’13)*. May 2013, pp. 1299–1300.
- [5] Jurek Czyzowicz, Kostantinos Georgiou, and Evangelos Kranakis. “Group Search and Evacuation”. In: *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*. Ed. by Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Cham: Springer International Publishing, 2019, pp. 335–370. ISBN: 978-3-030-11072-7. DOI: 10.1007/978-3-030-11072-7_14. URL: https://doi.org/10.1007/978-3-030-11072-7_14.
- [6] Jurek Czyzowicz ゾ. “Optimal online and offline algorithms for robot-assisted restoration of barrier coverage”. In: *Discrete Applied Mathematics* 285 (Oct. 2020), pp. 650–662. DOI: 10.1016/j.dam.2020.04.027.

- [7] Jurek Czyzowicz ら. “The Beachcombers’ Problem: Walking and Searching with Mobile Robots”. In: *Structural Information and Communication Complexity*. Ed. by Magnús M. Halldórsson. Cham: Springer International Publishing, 2014, pp. 23–36. ISBN: 978-3-319-09620-9.
- [8] Jurek Czyzowicz ら. “The Bike Sharing Problem”. In: *Proceedings of 15th International Conference and Workshops on Algorithms and Computation (WALCOM 2021)*. Vol. 12635. Lecture Notes in Computer Science. 2021, pp. 65–77.
- [9] Jurek Czyzowicz ら. “The Power of Tokens: Rendezvous and Symmetry Detection for Two Mobile Agents in a Ring”. In: *Proceedings of the 34th Conference on Current Trends in Theory and Practice of Computer Science. SOFSEM’08*. Nový Smokovec, Slovakia: Springer-Verlag, 2008, pp. 234–246. ISBN: 354077565X.
- [10] J. Djugash ら. “Range-only SLAM for robots operating cooperatively with sensor networks”. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. 2006, pp. 2078–2084. DOI: 10.1109/ROBOT.2006.1642011.
- [11] F. Drews and D. Luxen. “Multi-hop ride sharing”. In: *Proceedings of the 6th Annual Symposium on Combinatorial Search, SoCS 2013* (Jan. 2013), pp. 71–79.
- [12] M. R. Garey, D. S. Johnson, and Ravi Sethi. “The Complexity of Flowshop and Jobshop Scheduling”. In: *Math. Oper. Res.* 1.2 (May 1976), pp. 117–129. ISSN: 0364-765X. DOI: 10.1287/moor.1.2.117. URL: <https://doi.org/10.1287/moor.1.2.117>.
- [13] Boyoon Jung and G.S. Sukhatme. “Cooperative tracking using mobile robots and environment-embedded, networked sensors”. In: *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515)*. 2001, pp. 206–211. DOI: 10.1109/CIRA.2001.1013198.
- [14] E. Kranakis ら. “Mobile agent rendezvous in a ring”. In: *23rd International Conference on Distributed Computing Systems, 2003. Proceedings*. 2003, pp. 592–599. DOI: 10.1109/ICDCS.2003.1203510.

- [15] Matthias Kropff ら. “MM-ulator: Towards a Common Evaluation Platform for Mixed Mode Environments”. In: *Lecture Notes in Artificial Intelligence*. Ed. by Modeling Simulation and Programming for Autonomous Robots (SIMPAR 2008). Springer, Nov. 2008, pp. 41–52.
- [16] J. Schuijbroek, R.C. Hampshire, and W.-J. van Hoes. “Inventory rebalancing and vehicle routing in bike sharing systems”. In: *European Journal of Operational Research* 257.3 (2017), pp. 992–1004. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2016.08.029>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221716306658>.
- [17] C.K. Seow, Winston K.G. Seah, and Z. Liu. “Hybrid Mobile Wireless Sensor Network Cooperative Localization”. In: *2007 IEEE 22nd International Symposium on Intelligent Control*. 2007, pp. 29–34. DOI: 10.1109/ISIC.2007.4450856.