

## ソフトテニスのダブルス対戦表生成手法

田邊 紘幹<sup>†</sup> 山口 一章<sup>††</sup> 中村 匡秀<sup>†</sup>

<sup>†</sup> 神戸大学 工学部 〒657-8501 兵庫県神戸市灘区六甲台町 1-1

<sup>††</sup> 追手門学院大学 理工学部 〒567-8502 大阪府茨木市西安威 2-1-15

E-mail: <sup>†</sup>{tanahiro,staff-es4}@es4.eeddept.kobe-u.ac.jp, <sup>††</sup>ka-yamaguchi@otemon.ac.jp

**あらまし** ソフトテニスのイベントや練習会では交流促進のためにダブルス形式の試合が広く行われる。しかし、参加者数やコート数、試合可能回数などの制約下で、試合数や役割の偏りを抑えつつ多様な組合せを実現する対戦表を手作業で作成することは困難である。本研究では、ポジション希望、実力差、同一ペアおよび同一対戦の回避など複数条件を考慮し、ハード制約を満たした上でソフト制約をペナルティとして最小化する対戦表生成手法を提案する。

**キーワード** ソフトテニス, ダブルス, 対戦表生成, スケジューリング, 組合せ最適化

## A Method for Generating Soft-Tennis Doubles Match Schedules

Hiroki TANABE<sup>†</sup>, Kazuaki YAMAGUCHI<sup>††</sup>, and Masahide NAKAMURA<sup>†</sup>

<sup>†</sup> Faculty of Engineering, Kobe University 1-1 Nada-ku, Kobe, Hyogo, 657-8501 Japan

<sup>††</sup> Faculty of Science and Engineering, Otomon Gakuin University 2-1-15 Nishi-Ai, Ibaraki, Osaka, 567-8502 Japan

E-mail: <sup>†</sup>{tanahiro,staff-es4}@es4.eeddept.kobe-u.ac.jp, <sup>††</sup>ka-yamaguchi@otemon.ac.jp

**Abstract** Doubles matches are widely played in soft tennis events to promote social interaction. However, manually creating match schedules is challenging under constraints such as the number of participants, courts, and matches. We propose a scheduling method that satisfies hard constraints while minimizing penalties for soft constraints, including position preferences, skill gaps, and the avoidance of repetitive pairings or opponents.

**Key words** soft tennis, doubles, Tournament scheduling, combinatorial optimization, constraint satisfaction

### 1. はじめに

ソフトテニスのイベントや練習会において、参加者同士の交流を促進するために、ダブルス形式の試合が広く行われている。特に大学のサークル活動や地域の交流イベントでは、できるだけ多くの参加者と試合を行うことや、試合数や前衛・後衛といった役割の偏りを抑えることが重要な目的となる。しかし、参加者数や使用可能なコート数、各参加者の試合可能回数といった制約が存在する中で、これらの条件を同時に満たす対戦表を手作業で作成することは困難である。

スポーツにおける対戦表生成は、ラウンドロビン方式やリーグ戦設計などを含むスポーツスケジューリング問題として広く研究されている [1]。一方で、この種の問題は一般に組合せ最適化問題として定式化され、問題規模の増大に伴い計算が困難となることが知られている [2]。したがって、実運用を想定した計算時間内で実用的な対戦表を安定して得るには、制約条件と評価基準を明確化し、探索的に解を選択する枠組みが有効である。

実際の運営現場では、ランダムにペアを組んで試合を進行したり、運営者がその場で判断して次の試合を決定したりする方

法が一般的である。しかし、このような方法では、待機同じ相手や同じペアとの試合が繰り返される、前衛・後衛といったポジションの希望が考慮されない、ある参加者に試合が集中するなどの問題が生じやすい。例えば、ある参加者は短時間に連続して複数試合に出場する一方で、別の参加者は長時間待機する状況が生じるなど、参加者間で待ち時間や疲労の偏りが発生する場合がある。特に、試合が終了したコートから順に次の試合を開始する形式では、使用するコート数を考慮した試合順の設計が不可欠であり、適切なスケジューリングが行われない場合、コートの空き時間や参加者の過度な発生するなど、イベント全体の満足度低下につながるおそれがある。

これらの問題に対して、対戦表生成を組合せ最適化問題として定式化し、制約条件と評価基準を明確に定義した上で、自動的に対戦表を生成する手法が有効であると考えられる。本研究では、ソフトテニスのダブルス対戦表生成問題を対象とし、参加者のポジション希望、実力差、同一ペアおよび同一対戦の回避といった複数の条件を考慮したスケジューリング手法を提案する。特に、使用可能なコート数を考慮した試合順の制約を明示的にモデル化し、実運用において発生しやすい制約違反を防

ぎながら、対戦の多様性と試合バランスを両立する点に特徴がある。

本研究では、必ず満たすべき条件をハード制約として定義し、それ以外の条件についてはソフト制約としてペナルティ関数により評価する。すべての条件を厳密に満たす最適解を求めることを目的とするのではなく、現実的な計算時間内で、実際のイベント運営において十分に利用可能な品質の対戦表を安定して生成することを目的とする。

次章では、本研究で想定するイベント形式および制約条件を整理し、ソフトテニスのダブルス対戦表生成問題の定義を明確にする。

## 2. 準備

### 2.1 問題設定

本研究では、ソフトテニスにおけるダブルス形式の試合イベントを対象とする。本イベントは勝敗や順位付けを主目的とするものではなく、参加者同士の交流を促進することを目的とした交流型イベントを想定する。

イベントには  $n$  人のプレイヤーが参加し、各試合は 2 組のペア、計 4 人で構成される。使用可能なコートの面数を  $C$  とし、イベント開始時には最大  $C$  試合が同時に開始されるものとする。その後は、いずれかの試合が終了した時点で空いたコートに新たな試合を順次割り当てる形式を仮定する。

このような運営形式では、各試合の正確な開始・終了時刻を事前に決定することは困難である。そこで本研究では、試合の開始時刻ではなく、「第 1 試合、第 2 試合、…」などの試合番号に基づいて試合順を管理するモデルを採用する。すなわち、各プレイヤーがどの試合番号に出場するかを決定する問題として、対戦表生成を扱う。

### 2.2 目的関数の考え方

本研究で扱うソフトテニスのダブルス対戦表生成問題では、すべての条件を同時に厳密に満たす解が常に存在するとは限らない。特に、参加者数、使用コート数、イベント時間といった現実的な制約の下では、試合数の均等化、ペアや対戦相手の重複回避、ポジションや実力のバランスといった条件の間にトレードオフが生じる。このように複数の評価観点で競合する状況は、多目的最適化として一般に知られている。

例えば、試合数の均等化を優先すると同一対戦の回避が困難になる一方で、対戦の多様性を重視すると試合間隔や疲労に偏りが生じる場合がある。このような状況において、すべての条件をハード制約として扱うことは現実的ではない。

そこで本研究では、必ず満たすべき条件をハード制約として定義し、それ以外の条件についてはペナルティとして数値化する。複数の評価観点を単一の目的関数に統合し、その値を最小化することにより、全体として望ましい対戦表を評価・生成する。

### 2.3 プレイヤーの属性

プレイヤー集合を  $P$  とし、各プレイヤー  $p \in P$  は以下の属性を持つものとする。

- ポジション希望 (前衛または後衛)

- 実力を表す数値

- 性別

ソフトテニスのダブルスでは、前衛と後衛が役割分担を行うことが一般的であるため、前衛希望者と後衛希望者のペアは望ましい。一方で、参加者数やポジションの偏りによっては、同一ポジション同士のペアが必要となる場合もある。そのため、ポジション希望は厳密な制約ではなく評価対象として扱う。

また、対戦するペア間の実力差が極端に大きい試合は、交流イベントとしての満足度を低下させる要因となる。そこで本研究では、各プレイヤーの実力を数値として与え、試合ごとの実力差を評価する。

性別は、男女混合あるいは同性同士といった試合構成に関する希望を考慮するために用いる。

### 2.4 試合数と試合順

イベント全体で行われる試合数を  $K$  とする。各プレイヤーが出場する試合数は、できるだけ均等になることが望ましい。

ダブルスの試合は 4 人で構成されるため、参加者数と 1 人あたりの試合数の積が 4 の倍数となることが理想である。しかし、実際のイベント運営では、時間的制約や参加者数の都合により、この条件を厳密に満たすことは困難な場合が多い。

そのため本研究では、各プレイヤーの試合数に多少のばらつきが生じることを許容し、その偏りをできるだけ小さく抑えることを評価関数によって誘導する。

### 2.5 ハード制約

以下の制約は、試合として物理的に成立するために必ず満たされなければならない条件である。

(1) 同一ペア制約: 同じ 2 人のプレイヤーが、イベント中に複数回ペアを組むことはない。

(2) 試合構成制約: 各試合には必ず 2 組のペアが出場し、合計 4 人のプレイヤーで構成される。

(3) 同一試合内重複出場禁止制約: 同一の試合において、同一プレイヤーが複数のペアに同時に含まれてはならない。

(4) 同時出場禁止制約: 使用可能なコート数を  $C$  とすると、同時に行われる  $C$  試合の間で、同一のプレイヤーが複数の試合に出場してはならない。

### 2.6 ソフト制約

次に、できるだけ満たすことが望ましいが、現実的な制約から完全に満たすことが難しい条件をソフト制約として定義する。

- ポジション制約: 同一ポジション同士のペアはできるだけ避けたい。

- 試合間隔制約: 同一プレイヤーの出場試合の間隔は、使用コート数  $C$  以上空くことが望ましい。

- 同一対戦回避制約: 過去に同じ試合に出場したプレイヤー同士が、再び同一試合に出場する状況はできるだけ避けたい。

- 実力差制約: 試合ごとの実力差が極端に大きくならないようにしたい。

- 性別構成制約: 男女混合または同性同士といった希望に応じた試合構成が望ましい。

これらのソフト制約は、第 3 章においてペナルティとして数

式化する.

### 2.7 問題の定義

以上より, 本研究で扱うソフトテニスのダブルス対戦表生成問題は, ハード制約を必ず満たした上で, 複数のソフト制約に対するペナルティの合計が最小となるように, ペアの組合せおよび試合順を決定する組合せ最適化問題として定義される. 一般にこの種の組合せ最適化問題は, 規模の増大に伴い厳密探索が困難となることが知られている [2].

本研究では, 各試合番号に対して 2 組のペア (計 4 人) を割り当てるとともに, それらの試合順を同時に決定する対戦表を生成対象とする.

## 3. 提案手法

### 3.1 変数とモデル

プレイヤー集合を  $P = \{1, 2, \dots, n\}$ , 試合番号の集合を

$$M = \{1, 2, \dots, K\}$$

とする. 試合  $m \in M$  は 2 組のペア (計 4 人) から構成される.

プレイヤー  $p, q \in P$  ( $p < q$ ) が試合  $m$  においてペアとして出場するかどうかを表す二値変数を

$$y_{pq}^m \in \{0, 1\} \quad (p < q, m \in M)$$

と定義する.

さらに, プレイヤー  $p$  が試合  $m$  に出場しているかどうかを表す二値変数を

$$z_p^m = \sum_{q < p} y_{qp}^m + \sum_{q > p} y_{pq}^m, \quad z_p^m \in \{0, 1\}$$

と定義する.

### 3.2 ハード制約

以下の制約は, 試合として物理的に成立するために必ず満たされなければならない条件である.

(1) 同一ペア重複禁止制約

同じ 2 人のプレイヤーがイベント中に複数回ペアを組まない:

$$\sum_{m \in M} y_{pq}^m \leq 1 \quad (\forall p < q)$$

(2) 試合構成制約

各試合は必ず 2 組のペアから構成される:

$$\sum_{p < q} y_{pq}^m = 2 \quad (\forall m \in M)$$

(3) 同一試合内重複出場禁止制約

同一プレイヤーが同一試合内で 2 つのペアに同時に含まれない:

$$z_p^m \leq 1 \quad (\forall p \in P, \forall m \in M)$$

(4) 同時出場禁止制約 (同時進行  $C$  面の近似)

使用可能なコート数を  $C$  とし, 同時に進行する  $C$  試合を「連続する  $C$  試合番号」として近似する. このとき, 連続する  $C$  試

合の間で同一プレイヤーが複数回出場しない:

$$\sum_{t=m}^{m+C-1} z_p^t \leq 1 \quad (\forall p \in P, m = 1, \dots, K - C + 1)$$

### 3.3 目的関数設計

本研究では, ハード制約を必ず満たした上で, 複数の評価観点をペナルティとして数値化し, その重み付き和を最小化することにより, 実運用に適した対戦表の生成を目指す.

#### 3.3.1 ポジション不一致ペナルティ

プレイヤー  $p$  のポジション希望を

$$pos(p) \in \{\text{front}, \text{back}\}$$

とする.

同一ポジション同士のペアに対してペナルティを課すため,

$$\delta_{\text{pos}}(p, q) = \begin{cases} 1 & pos(p) = pos(q) \\ 0 & \text{otherwise} \end{cases}$$

を定義する.

このとき, ポジション不一致ペナルティは次式で表される:

$$P_{\text{pos}} = \sum_{m \in M} \sum_{p < q} \delta_{\text{pos}}(p, q) y_{pq}^m.$$

#### 3.3.2 同一対戦回避ペナルティ

交流促進の観点から, 同じ顔合わせが繰り返されることを避けるため, 異なる 2 試合  $m, m'$  に共通して出場したプレイヤー数に基づきペナルティを定義する.

試合  $m$  と  $m'$  に共通して出場した人数を

$$\text{overlap}(m, m') = \sum_{p \in P} z_p^m z_p^{m'}$$

と定義する.

このとき, 同一対戦回避ペナルティを

$$P_{\text{overlap}} = \sum_{m < m'} \max(0, \text{overlap}(m, m') - 1)$$

と定める.

#### 3.3.3 実力差ペナルティ

プレイヤー  $p$  の実力を  $s(p)$  とし, ペア  $(p, q)$  の実力を

$$S_{pq} = s(p) + s(q)$$

と定義する.

試合  $m$  において選択された 2 組のペアを  $(p_1^m, q_1^m), (p_2^m, q_2^m)$  とすると, 試合  $m$  における実力差ペナルティを

$$P_{\text{skill}}^m = \left| S_{p_1^m q_1^m} - S_{p_2^m q_2^m} \right|$$

と定義する.

全試合に対する実力差ペナルティは次式で与えられる:

$$P_{\text{skill}} = \sum_{m \in M} P_{\text{skill}}^m.$$

### 3.3.4 総合目的関数

以上の各ペナルティ項を重み付きで加算し、総合目的関数を次式で定義する：

$$\min w_1 P_{\text{pos}} + w_2 P_{\text{overlap}} + w_3 P_{\text{skill}}$$

ここで  $w_1, \dots, w_3$  は各評価項目の重要度を表す重み係数であり、イベントの目的や参加者構成に応じて調整される。

### 3.4 アルゴリズム

本研究では、ハード制約を必ず満たす対戦表を候補として生成し、目的関数値が最小となるものを選択する反復探索アルゴリズムを採用する。本アルゴリズムは、(1) ハード制約を満たす対戦表の生成、(2) 目的関数による評価、(3) 最良解の更新から構成される。各試行は互いに独立であり、既存解に対する近傍操作や局所的な改変は行わない。

#### 3.4.1 入力と出力

アルゴリズムの入力は、プレイヤー集合  $P$ 、試合数  $K$ 、同時進行試合数（コート数） $C$ 、各プレイヤーの属性（ポジション希望、実力値など）、および目的関数の重み係数  $w_1, w_2, w_3$ 、試行回数  $T$ 、乱数シードである。

出力は、各試合  $m \in M$  に対して 2 組のペア（計 4 人）を割り当てた対戦表  $S$  である。

#### 3.4.2 ハード制約を満たす対戦表の生成

各試行では、乱数に基づいて対戦表を新規に構成する。生成にあたっては、同一ペア重複禁止制約、試合構成制約、および同時出場禁止制約といったハード制約を満たすことを前提とする。

生成過程において、これらのハード制約を満たす対戦表を構成できない状態（例えば、有効な組合せが存在しない場合）が生じた場合、当該試行は失敗として棄却し、次の乱数シードによる試行へ移行する。試行間に依存関係はなく、失敗した試行の情報は次の生成には引き継がれない。

#### 3.4.3 評価および最良解の更新

生成に成功した対戦表  $S$  に対して、第 3.3 節で定義した目的関数

$$f(S) = w_1 P_{\text{pos}}(S) + w_2 P_{\text{overlap}}(S) + w_3 P_{\text{skill}}(S)$$

を計算する。

この操作を試行回数  $T$  回繰り返す、これまでに生成された対戦表のうち、目的関数値  $f(S)$  が最小となるものを最終的な対戦表  $S^*$  として採用する。

#### 3.4.4 反復探索アルゴリズムの概要

以上の処理を擬似コードとしてまとめると、以下のようになる。

#### Algorithm 1: Random-Generate-and-Select

- (1)  $best \leftarrow +\infty, S^* \leftarrow \emptyset$  と初期化する。
- (2) 試行回数  $T$  回について、以下を繰り返す。
  - (a) 乱数シードを設定し、ハード制約を満たす対戦表  $S$  を生成する。
  - (b) 生成に失敗した場合は、次の試行へ移行する。
  - (c) 目的関数値  $f(S)$  を計算する。

(d)  $f(S) < best$  のとき、 $best \leftarrow f(S), S^* \leftarrow S$  と更新する。

(3)  $S^*$  を最終解として出力する。

## 4. 実験方法

### 4.1 実験の目的と方針

本研究では、提案するダブルス対戦表生成手法の有効性を検証するため、条件の異なる複数のケースにおいて、人手で作成された対戦表と、提案手法により自動生成された対戦表を比較する実験を行った。特に、以下の観点に着目して評価を行う。

- 同一ペア重複や同時出場といったハード制約を満たす対戦表を安定して生成できるか
- 評価関数に基づく対戦表の質が、人手作成案と比べてどのように変化するか
- 出場回数の公平性指標と、評価関数値との間にどのような関係が現れるか

本研究では、評価関数による最適化と、出場回数に基づく公平性評価を分離して扱い、そのトレードオフを分析することを基本方針とする。

### 4.2 共通の実験条件

すべての実験ケースにおいて、以下の条件を共通とした。

- 同時進行試合数（使用コート数）： $C = 5$
- 試行回数：200
- 乱数シード：1, 2, ..., 200
- 評価関数の重み： $w_1 = 1$ （ポジション）、 $w_2 = 1$ （同一対戦）、 $w_3 = 1$ （実力差）

なお、本実験では試合間隔ペナルティは使用せず、同一対戦回避および実力差に関する評価を主として扱う。

### 4.3 比較対象

本研究では、以下の 3 種類の対戦表を比較対象とする。

- **human**：実運営者が手作業で作成した対戦表。出場回数の公平性を重視して作成される傾向があるが、ハード制約違反が含まれる可能性がある。
- **code.base**：提案手法により、乱数シードを固定 ( $seed=42$ ) して生成した対戦表。
- **code.fair**：複数の乱数シード ( $seed=1, \dots, 200$ ) を用いて対戦表を生成し、ハード制約を満たす解のみを候補とした上で、公平性指標を考慮して選択した対戦表。

### 4.4 評価指標

各対戦表について、以下の指標を算出する。

- a) ハード制約違反
  - 同一ペア重複の種類数 ( $pair\_dupKinds$ )
  - 同時出場違反の件数 ( $simultaneous\_play\_violation$ )
- b) 評価関数値

提案手法で定義した評価関数に基づき、以下を算出する。

- 同一対戦回避ペナルティ  $P_{\text{Overlap}}$
  - 実力差ペナルティ  $P_{\text{Skill}}$
  - 総評価値  $total = P_{\text{Overlap}} + P_{\text{Skill}}$
- c) 公平性指標

各プレイヤーの出場回数に基づき、以下を算出する。

表1 Case A における評価結果 (小規模条件)

手法	pairDupKinds	simViol	POverlap	PSkill	total	range	std
human	2	2	32	78	110	1	0.318
code_base	0	0	34	74	108	5	1.326
code_fair	0	0	28	59	87	5	0.979

表2 Case B における評価結果 (中規模条件)

手法	pairDupKinds	simViol	POverlap	PSkill	total	range	std
human	1	2	44	170	214	2	0.447
code_base	0	0	49	122	171	5	1.140
code_fair	0	0	43	107	150	6	1.095

- 出場回数の最大値と最小値の差 (play\_range)
- 出場回数の標準偏差 (play\_std)

#### 4.5 公平性を考慮した解の選択基準

code\_fair の選択にあたっては、評価関数値と出場回数の公平性を同時に考慮するため、次式で定義されるスコアを用いる。

$$\text{score} = \text{total} + \alpha \cdot \text{play\_std}$$

ここで  $\alpha$  は公平性指標の重みであり、本実験では  $\alpha = 30$  とした。

複数の乱数シードにより生成された対戦表のうち、ハード制約を満たす解のみを候補とし、上式のスコアが最小となる解を公平性を考慮した対戦表 (code\_fair) として採用する。

#### 4.6 実験ケース

本研究では、プレイヤー数および試合数の異なる以下の三つのケースを設定した。

- Case A (小規模条件)
  - プレイヤー数: 35 人
  - 試合数: 36 試合
- Case B (中規模条件)
  - プレイヤー数: 40 人
  - 試合数: 50 試合
- Case C (条件差比較)
  - プレイヤー数: 40 人
  - 試合数: 40 試合

#### 4.7 本章のまとめ

本章では、提案手法の有効性を検証するための実験方法を示した。評価関数に基づく最適化と、出場回数の公平性評価を分離して扱うことで、両者のトレードオフを分析可能な実験設計とした。次章では、本章で定義した評価指標に基づき、各ケースにおける実験結果を定量的に示す。

## 5. 評価

本章では、第4章で示した実験方法に基づいて得られた結果を示す。各ケースにおいて、人手で作成された対戦表 (human)、提案手法による自動生成対戦表 (code\_base)、および公平性指標を考慮して選択した対戦表 (code\_fair) を比較し、ハード制約の満足状況、評価関数値、および出場回数の公平性指標の観点から定量的に評価を行う。

なお、人手作成対戦表 (human) は、ハード制約違反を含む場合があるため、実運用上は不適切な解となる。ただし、比較のため、参考値として評価関数値および公平性指標を算出する。

#### 5.1 Case A の評価 (小規模条件)

Table 1 に Case A における評価結果を示す。

Case A では、人手作成案において出場回数の標準偏差が小さ

表3 Case C における評価結果 (条件差比較)

手法	pairDupKinds	simViol	POverlap	PSkill	total	range	std
human	1	1	29	135	164	2	0.387
code_base	0	0	25	100	125	5	1.095
code_fair	0	0	17	87	104	3	0.866

く、高い公平性が確認された。一方で、同一ペアの重複および同時出場違反が発生しており、ハード制約を満たしていない。

これに対し、提案手法による自動生成案では、すべてのハード制約を満たす対戦表が生成された。特に code\_fair では、重複対戦および実力差に関するペナルティが低減され、評価関数値 (total) が最小となった。小規模条件では、公平性と評価関数値の両立が難しい傾向が見られるものの、制約を満たした上での改善が確認された。

#### 5.2 Case B の評価 (中規模条件)

Table 2 に Case B における評価結果を示す。

Case B では、人手作成案が出場回数の公平性において極めて良好な結果を示した。しかし、同一ペアの重複が確認され、ハード制約を満たしていない。

一方、code\_base はハード制約を満たすものの、評価関数値および公平性指標の双方で改善の余地が見られた。これに対し、複数の乱数シードを用いた探索により選択した code\_fair では、評価関数値 (total) および出場回数の標準偏差の双方が改善された。この結果は、評価関数による最適化を行った後に、公平性指標を考慮して解を選択する手法が、中規模条件において有効であることを示している。

#### 5.3 Case C の評価 (条件差比較)

Table 3 に Case C における評価結果を示す。

Case C では、人手作成案において、出場回数の公平性は高いものの、同一ペア重複および同時出場違反が確認された。この結果は、試合数と参加人数が同規模となる条件では、人手による調整が困難になりやすいことを示している。

これに対し、提案手法による自動生成案では、すべてのハード制約を満たした対戦表が生成された。特に code\_fair では、評価関数値 (total) が最小となると同時に、code\_base と比較して出場回数のばらつきが改善された。この結果から、評価関数と公平性指標の間に存在するトレードオフを、後段の解選択によって調整可能であることが確認された。

#### 5.4 評価結果のまとめ

以上の実験結果より、提案手法はすべてのケースにおいて、同一ペアの重複や同時出場といったハード制約を確実に満たす対戦表を安定して生成できることが示された。

また、人手作成案は出場回数の公平性が高い傾向を示す一方で、制約違反を含む場合があることが確認された。これに対し、

表4 Case Cにおける human, code\_base, code\_fair の評価結果比較

手法	total	play_std	play_range	ハード制約
human	164	0.387	2	違反あり
code_base	125	1.095	5	満足
code_fair	104	0.866	3	満足

提案手法では、評価関数に基づいて対戦の多様性や実力差を制御しつつ、公平性指標を考慮した解選択を行うことで、実運用に適した対戦表を得られることが明らかとなった。

## 6. まとめ

本研究では、ソフトテニスの交流型イベントを対象として、ダブルス対戦表生成問題をハード制約とソフト制約を分離した組合せ最適化問題として定式化し、評価関数に基づく自動生成手法を提案した。また、人手で作成された対戦表と比較することで、提案手法の有効性および実運用上の特性を検証した。

本章では、実験結果を踏まえて本研究の結論を整理し、得られた知見を考察するとともに、今後の課題と展望について述べる。

### 6.1 結論

実験結果より、提案手法はすべての実験ケースにおいて、同一ペアの重複や同時出場といった試合として成立しない状況を確実に回避し、ハード制約を満たす対戦表を安定して生成できることが確認された。

一方、人手で作成された対戦表は、出場回数の公平性が高い傾向を示すものの、同一ペア重複や同時出場違反といったハード制約違反が混入する可能性があることが明らかとなった。この結果は、参加者数や試合数が増加した条件において、人手による調整のみですべての制約を満たすことが困難であることを示している。

特に Case C (プレイヤー数と試合数が同規模となる条件) では、人手作成案において公平性は高いものの、ハード制約違反が発生した。これに対し、提案手法では、ハード制約を満たした上で、評価関数値および出場回数の公平性をバランス良く両立した対戦表が生成された。この結果から、提案手法は実運用を想定した条件下において、人手作成に比べてより安定的かつ再現性の高い対戦表生成を可能にすることが示された。

### 6.2 考察

人手による調整が公平性を向上させる一方で制約違反を招きやすい点は、スポーツスケジューリングにおける制約遵守と評価指標のトレードオフとして既存研究でも指摘されている [4]。

評価結果から、評価関数値 (対戦の多様性および実力差) と、出場回数の公平性との間には、明確なトレードオフ関係が存在することが確認された。

表4に、Case Cにおける human, code\_base, code\_fair の評価関数値 (total) および出場回数の公平性指標を示す。図中の左下ほど、評価関数値と公平性の両方に優れた対戦表であることを意味する。human は出場回数の公平性が高い位置に分布する一方で、同一ペア重複や同時出場といったハード制約違反を含むため、実運用上は不適切な解である。

これに対し、提案手法により生成された code\_base はハード制約を満たすものの、評価関数値および公平性指標の点で改善の余地が残る結果となった。一方、複数の乱数シードを用いた探索により選択した code\_fair は、ハード制約を満たした上で、評価関数値と公平性のトレードオフをバランス良く改善していることが確認できる。この結果は、公平性を目的関数に直接組み込むことが難しい場合においても、後段で解を評価・選択するという設計方針が、実運用上有効に機能することを示している。

また、人手作成案が高い公平性を示す一方で制約違反を含みやすいという結果は、人が局所的な調整や直感的判断を優先することで、全体構造に関わる制約を見落としやすいことを示唆している。これに対し、提案手法はハード制約を厳密に満たす枠組みの中で評価関数に基づく探索を行うため、実運用における破綻を防ぎつつ、一定水準以上の品質を安定して確保できる点に特徴がある。

以上より、「人手による直感的な公平性」と「アルゴリズムによる制約遵守および定量評価」を組み合わせることが、交流型イベントにおける対戦表生成において有効なアプローチであると結論づけられる。

### 6.3 今後の課題と展望

今後の課題として、以下の点が挙げられる。

- 出場回数の公平性指標を評価関数に直接組み込む方法や、多目的最適化問題として定式化した場合との比較検討
- 局所探索法やメタヒューリスティクスを導入することによる、探索効率の向上および解品質の改善
- プレイヤーの技能差やポジション希望をより詳細に表現可能な評価モデルへの拡張
- イベントの目的に応じて、男女混合あるいは同性同士といった性別構成を柔軟に制御可能な制約・評価モデルへの拡張
- 実際のイベント運営を想定し、対戦表の可読性や修正容易性を考慮した出力形式およびユーザインタフェースの設計

これらの課題に取り組むことで、より柔軟かつ実運用に適した対戦表生成システムの構築が可能になると考えられる。本研究で提案した枠組みは、ソフトテニスに限らず、交流型スポーツイベントやレクリエーションにおけるスケジューリング問題にも応用可能であり、今後の発展が期待される。

謝辞 本研究の一部は JSPS 科研費 JP25H01167, JP25K02946, JP25K24389, JP24K02765, JP24K02774, JP23K17006, JP23K28091, JP23K28383 の研究助成を受けて行われている。

## 文献

- [1] G. Kendall, S. Knust, C. C. Ribeiro, and S. Urrutia, "Scheduling in sports: An annotated bibliography," *Computers & Operations Research*, vol. 37, no. 1, pp. 1–19, 2010.
- [2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
- [3] E. Aarts and J. K. Lenstra (Eds.), *Local Search in Combinatorial Optimization*, Princeton University Press, 2003.
- [4] 宮代隆平, 松井知己, 「スポーツスケジューリング未解決問題を中心に」, オペレーションズ・リサーチ, Vol. 50, No. 2, pp. 119–124, 2005.