二重総当たりリーグ戦における移動距離の最小化
Traveling distance minimization in double round-robin tournament

○平野　敬祐[1]
中央大学

阿部　敬太[2]
中央大学

今堀　慎治[3]
中央大学

[1]Keisuke Hirano
Chuo University
a14.ehsr@g.chuo-u.ac.jp

[2]Keita Abe
Chuo University

[3]Shinji Imahori
Chuo University
imahori@ise.chuo-u.ac.jp

Abstract

This paper studies the traveling tournament problem (TTP), a well-known benchmark problem in the field of sports scheduling. The objective of TTP is to make a double round-robin tournament schedule that minimizes the total traveling distance. We propose a new local search algorithm and evaluate the proposed algorithm on benchmark instances with up to forty teams.

1 序論

スポーツの試合のスケジュールを調整し，選手や主催者，ファンの負担の軽減，来場者数の増加及び興行収入の最大化，公平性のある試合開催などを目指すことをスポーツスケジューリングという．本研究で扱う巡回トーナメント問題（Traveling Tournament Problem: TTP）は二重総当たりリーグ戦の形式をとるスポーツに対して，各チームの移動距離の総和を最小化することを目的とした組合せ最適化問題である．TTP は，Michael Trick らによって提唱された問題であり[1], Trick の管理するウェブサイトにはベンチマーク問題とそれらの問題の暫定解がまとめられている[2]．TTP は厳密解を求めることが難しく，ベンチマーク問題においてもチーム数 10 以下の問題に対してのみしか厳密解が得られていない．そこで TTP に対する高性能な近似解法を提案し，提案手法の性能を数値実験により評価する．

2 問題定義

本研究では二重総当たりリーグ戦のスケジュールを扱う．二重総当たりリーグ戦とは，各チームが自身の本拠地と相手の本拠地で 1 度ずつ試合を行う形式で開催されるリーグ戦のことである．

今後用いる用語を，以下のように定義する．

- ホームゲーム：自身の本拠地で行う試合
- アウェイゲーム：相手の本拠地で行う試合
- ツアー：ホームゲームまたはアウェイゲームが連続して行われること

本研究では以下の入力に対し，制約を満たし，目的関数を最小化する二重総当たりリーグ戦のスケジュールを求める．

入力

- チーム数N（Nは偶数）
- 各チームの本拠地間の距離行列$D = \left(d_{i,j}\right)$
  Dは対称行列であり，本拠地間の距離は三角不等式を満たす

制約

- NoRepeat: 同じ対戦相手と連続する試合日で対戦してはいけない
- AtMost: ツアーの長さは３以下となる

目的関数

全てのチームの移動距離の総和.

各チームは初めに自身の本拠地に居るとし，リーグ戦終了後に自身の本拠地に戻るものとする．また，アウェイからアウェイの移動は自身の本拠地には戻らず直接移動する．よって，移動距離は以下の３つの移動距離の総和である．

- 自身の本拠地から 1 日目の試合開催地への移動距離
- i日目の試合開催地から$i+1$日目の試合開催地への移動距離$(i = 1,2,\ldots,2N-3)$
- 最終日の試合開催地から自身の本拠地への移動距離

3 初期階生成手法

本研究では TTP に対する局所探索法を提案する．3 節では，初期解となるスケジュールの構築方法について説明する．

3.1 スケジュール構築のアイデア

各チームの理想的な移動がどのようなものかを考えてみる. 図 1 において, チーム 1 に着目し，このチームがチーム 2,3,4 の本拠地で対戦する状況を考える．図 1 左はアウェイゲームとホームゲームを交互に行う場合の移動経路であり，自身のホームを毎回経由する非常に効率の悪い移動を行なっている．これを AtMost 制約を満たす，3 日間のツアーを行う経路（図 1 右）に変えると，効率の良い移動ができる．

次に，図 2 のようにチーム 1 が対戦相手となる 5 チームとのアウェイゲームを 2 回のツアーで実施することを考える．各ツアーで対戦するチームの本拠地間の距離が遠いもの（図 2 左）より図 2 右のような近いものの方が効率が良いとわかる．

そこで，まず初めに距離が近い 3 チームずつのグループに分けることを考える．

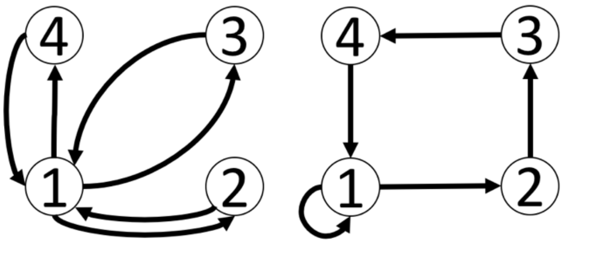

図 1: 非効率的な移動経路と効率的な移動経路

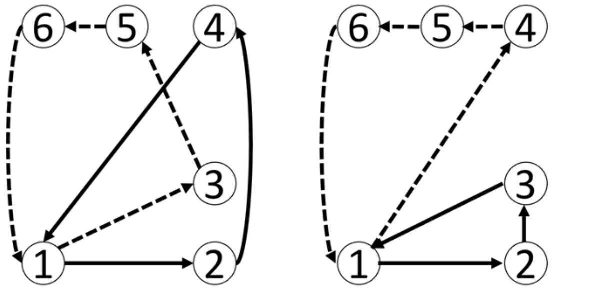

図 2: 非効率的なツアーと効率的なツアー

3.2 P_3-Packing

P_3-Packing とは与えられた点の集合を 3 点ずつのグループに分け各グループ内で作るパスの重みの総和を最小化する組合せ最適化問題である．

この問題は NP 困難であるが，小規模であれば定式化して数理計画ソルバーCPLEX12.8 を用いることで，最適解を求めることができる．

3.3 先行研究

3.3 節では先行研究におけるチーム数が$6k$と$6k+4$の 2 通りの問題に対する初期解生成手法について説明する．

3.3.1 チーム数$6k$の問題に対する TSP を用いた初期解生成手法

先行研究[3]ではチーム数が$N=6k$の問題に対して，サークルメソッドでスケジュールを構築する．巡回セールスマン問題（TSP）の解の経路順にチーム$1,\dots,N$と呼び，それらを図 3 のように配置する. 図 3 では細い矢印の根のチームの本拠地で，矢印の先のチームと戦うことを表す．これを太い矢印の方向に 1 周するまで動かす．ただし，1 チームのみ固定し，固定されたチームの矢印は 3 回ごとに矢印の向きを変える. 図 3 ではチーム 18 が固定されている．

この方法を用いると，チーム数が$6k$である条件のもとでは図 3 での左側を除いて同じ向きの矢印が 3 つずつ連続するため，各チームはホームあるいはアウェイがちょうど 3 回連続となる. さらに，図 3 のようにチームを 1 つ飛ばしに配置することによって，連続して対戦するチームが TSP の経路の順になるため，効率的な移動になる．

これらの操作によって全体の半分の$N-1$日分のスケジュールが構築できるので，さらにそのスケジュールのホームとアウェイを入れ替えて残り半分を作る．なお，類似の手法でチーム数が$6k$および$6k+2$の問題に対しても初期解を生成することができる．

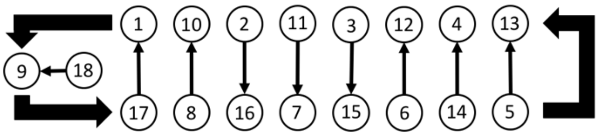

図 3: チーム数 18 のサークルメソッド

3.3.2 チーム数 $6k+4$ の問題に対する P_3-Packing を用いた初期解生成手法

3.1 節で説明したアイデアを元にスケジュールを構築する[3, 4]．具体的には，P_3-Packing を解くことによって求めた 3 チームのグループ単位で対戦を行うことを考える．あるグループの 3 チームが別のグループの 3 チームとホームとアウェイで 1 回ずつ戦う 6 日間の小さいスケジュールを作る．実際のスケジュールは図 4 のようになり，ホームとアウェイが逆転した 2 つ 1 組のスケジュールが作られる．あるグループから見たスケジュールを A のスケジュールとしたとき，対戦相手のグループから見たスケジュールは B のスケジュールになる．なお，スケジュール表のマイナスが付いているものはアウェイゲーム，それ以外はホームゲームを表す．

A		Day 1	2	3	4	5	6
Team	p	-c	-b	-a	c	b	a
	q	a	-c	-b	-a	c	b
	r	b	a	-c	-b	-a	c

B		Day 1	2	3	4	5	6
Team	a	-q	-r	p	q	r	-p
	b	-r	p	q	r	-p	-q
	c	p	q	r	-p	-q	-r

図 4: 6 日間のスケジュール

このような他のグループの 3 チームとの対戦を基本とするが，自身のグループの他の 2 チームとの対戦を行えないことが問題となる．これは，$6k+4$チームを 3 チームごとに分けたときの余りの 1 チームを用いて解決できる．図 5 では余りのチームを x として実際のスケジュールを示す．

A		Day 1	2	3	4	5	6
Team	a	-x	-c	b	x	c	-b
	b	c	-x	-a	-c	x	a
	c	-b	a	x	b	-a	-x
	x	a	b	-c	-a	-b	c

B		Day 1	2	3	4	5	6
Team	a	x	c	-b	-x	-c	b
	b	-c	x	a	c	-x	-a
	c	b	-a	-x	-b	a	x
	x	-a	-b	c	a	b	-c

図 5: 余りの 1 チームを含めたスケジュール

これら 4 つの 6 日間のスケジュールを組み合わせて全体のスケジュールを作る．図 6 はグループ単位のスケジュールを表しており，グループ x は余りのチーム x そのものを表す．対戦表の x の部分は自身のグループと余りのチームを混ぜた図 5 のスケジュールを割り当てる．ただし，*が付いていない箇所には A，付いている箇所には B のスケジュールを割り当てる．このルールに従うことで，6 日間のスケジュール同士の接続部分において，AtMost 制約に違反しないように二重総当たりリーグ戦のスケジュールを作ることができる．

		Term 1	2	3	4	5	6	7
Group	1	x	2	3	4	5	6	7
	2	7*	1*	x	3	4	5	6
	3	6*	7*	1*	2*	x	4	5
	4	5*	6*	7*	1*	2*	3*	x
	5	4	x*	6*	7*	1*	2*	3*
	6	3	4	5	x*	7*	1*	2*
	7	2	3	4	5	6	x*	1*
	x	1	5*	2	6*	3	7*	4

図 6: チーム数 22 のグループ単位の対戦表

3.4 提案手法

チーム数$6k$の問題に対して，P_3-Packing の解を用いた初期解の生成法を提案する．チーム数$6k$の場合，3 チームごとのグループに分けることができ，図 4 で示したグループ間の対戦を利用できる．しかし，同じグループの他の 2 チームとの対戦ができないという問題が生じる．先行研究では解決できていなかったこの問題を，2 つのグループを混ぜて，各チームが自身と同じグループの 2 チームと，一緒に戦うグループの 3 チームの計 5 チームとの対戦を行う図 7 のような 10 日間の小さいスケジュールを用いることで解決する．

A		Day 1	2	3	4	5	6	7	8	9	10
Team	p	-c	-b	-a	q	r	a	-r	-q	b	c
	q	a	-c	-b	-p	c	-r	b	p	r	-a
	r	b	a	-c	-a	-p	q	p	c	-q	-b

B		Day 1	2	3	4	5	6	7	8	9	10
Team	a	-q	-r	p	r	b	-p	-c	-b	c	q
	b	-r	p	q	c	-a	-c	-q	a	-p	r
	c	p	q	r	-b	-q	b	a	-r	-a	-p

図 7: 10 日間のスケジュール

この方法において，10 日間のスケジュールは，全て同じ期間に挿入する．チーム数$6k$の問題では，10 日間のスケジュール以外において，次の節とスケジュールの種類 A,B が変わる部分に AtMost 制約に違反する点が存在する．このような部分には，6 日間のスケジュールの最後の二日間を入れ替えた図 8 のスケジュールを使用することで制約を満たすことができる．

A		Day					
		1	2	3	4	5	6
Team	p	**-c**	**-b**	**-a**	c	a	b
	q	a	**-c**	**-b**	**-a**	b	c
	r	b	a	**-c**	**-b**	c	**-a**

B		Day					
		1	2	3	4	5	6
Team	a	**-q**	**-r**	p	q	**-p**	r
	b	**-r**	p	q	r	**-q**	**-p**
	c	p	q	r	**-p**	**-r**	**-q**

図 8: 特殊ケースの 6 日間のスケジュール

図 8 のスケジュールを使うと効率が悪い移動が増えるため，できるだけ少なくしたい．そこで，グループ単位のスケジュール生成問題とみなし，図 8 のスケジュールの使用数を目的関数として，数理計画ソルバーを用いて解いた. チーム数 30 の問題例までは解けたが，チーム数 36 の問題例ではソルバーの目的関数値と下界が一致しなかった．ただし，問題の性質として解は偶数になることが明らかであり， 目的関数値が 8 となり，下界が 6 を超えたため，厳密解は求まった．図 9 はソルバーで解いたチーム数 30 の場合のグループ単位のスケジュールである．+が付いている第 5 節には 10 日間のスケジュールを挿入し，黒く塗られたところは図 8 のスケジュールを割り当てる.

		Term								
		1	2	3	4	5+	6	7	8	9
Group	1	**7***	**9***	**3***	**4***	**5***	2	6	**10***	**8***
	2	**3***	**5***	**7***	**9***	**4***	**1***	**8***	**6***	**10***
	3	2	6	1	8	10	4	7	9	**5***
	4	**9***	**7***	**10***	1	2	**3***	**5***	**8***	**6***
	5	8	2	6	10	1	9	4	7	3
	6	**10***	**3***	**5***	**7***	**9***	**8***	**1***	2	4
	7	1	4	2	6	8	**10***	**3***	**5***	**9***
	8	**5***	**10***	**9***	**3***	**7***	6	2	4	1
	9	4	1	8	2	6	**5***	**10***	**3***	**7**
	10	6	8	4	**5***	**3***	7	9	1	2

図 9: 数理計画ソルバーで生成したチーム数 30 のグループ単位の対戦表

4 局所探索法

4.1 近傍

4.1.1 先行研究の近傍

先行研究[5]は，次に挙げる 5 種類の近傍を使用した局所探索法を提案した.

1. SwapHomes: 任意の 2 チームのホームとアウェイの開催順序を逆にする.
2. SwapDays: 任意の 2 日の試合日程を入れ替える.
3. PartialSwapDays: 任意の 1 チームに関して任意の 2 試合の日付を入れ替える．ただし，スケジュールの整合性を保つため，影響を受ける部分も入れ替える.
4. SwapTeams: 任意の 2 チームの試合日程を入れ替える．ただし，その影響を受ける部分も入れ替える．選択した 2 チーム同士の対戦は入れ替えない.
5. PartialSwapTeams: 任意の日付に関して任意の 2 試合の対戦相手を入れ替える．ただし，その影響を受ける部分も全て入れ替える.

4.1.2 提案手法の近傍

提案手法では 2 種類の近傍を用いる.

1. GroupSwapDay
この近傍では， SwapHomes， SwapDays， PartialSwapDays の 3 つを集約する.
まず初めに，PartialSwapDays 近傍の影響を受ける部分をグループ分けする．例えば，図 10 のスケジュールにおいて, 2 日目と 10 日目のスケジュールを入れ替える近傍を考える.
グループ分けの操作は図 11 のように，サイクルとなっている部分を 1 つのグループとみなして行う．グループごとに入れ替え後のスケジュールが制約を満たすか確認し，制約を満たすグループは入れ替えによる解の評価を行う．グループごとの評価は独立であるため，解が改善されるグループは全て入れ替える.

		Day									
		1	2	3	4	5	6	7	8	9	10
Team	1	-3	2	6	4	-5	-6	-4	5	3	-2
	2	-4	-1	5	3	4	-5	-6	-3	6	1
	3	1	6	4	-2	-6	-4	5	2	-1	-5
	4	2	5	-3	-1	-2	3	1	6	-5	-6
	5	-6	-4	-2	6	1	2	-3	-1	4	3
	6	5	-3	-1	-5	3	1	2	-4	-2	4

図 10: 6 チームのスケジュール例

DAY		…	2	…	10		グループ
Team	1		2	→	-2		①
	2		-1	←	1		①
	3		6	→	-5		②
	4		5	→	-6		②
	5		-4	←	3		②
	6		-3	←	4		②

図 11: GroupSwapDays のグループ分け

2. GroupSwapTeams

この近傍は，SwapHomes と PatiralSwapTeams を集約する．GroupSwapDays と同様に，初めに PartialSwapTeams において影響を受ける日程のグループ分けを行う．図 10 のスケジュールに対して，チーム 2 とチーム 5 のスケジュールの入れ替えを考える．グループ分けは，図 13 のようにサイクルになっている部分を 1 つのグループとみなす. 図 13 の一番下の数字が各日のグループ番号を表す．また，図 13 のグループ 0 のような選択した 2 チーム同士の対戦においては, SwapHomes と同様の入れ替えを行うと考え，それぞれのグループごとに入れ替え後のスケジュールが制約を満たすかを確認し, 入れ替えの評価を行う. また, 図 13 の網掛けの部分は，入れ替え対象のチームとの対戦であり，この部分も入れ替えることで，スケジュールの整合性が保たれる．GroupSwapTeams では，どのグループから入れ替えを行うかによって評価が変わり，グループの入れ替えが独立ではないため，最も評価の良いグループのみを入れ替える．そのため, 先行研究での SwapTeams をこの近傍に含めることはできないが，SwapTeams は大幅にスケジュールを変える近傍であり，そのような大きく解を変えるような近傍の評価がよくなることは少ないため，SwapTeams は含めないままで良いと考える.

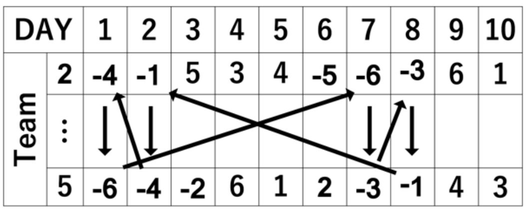

図 12: GroupSwapTeams のグループ分け

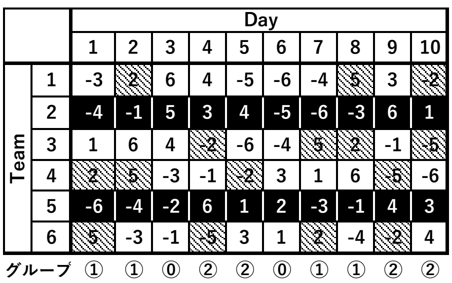

図 13: GroupSwapTeams の入れ替わる部分

4.1.3 近傍計算時間の比較

先行研究と提案手法の近傍の評価時間を比較する．表 1 では解の評価時間の比較を行う．既存手法に対して，提案手法ではグループ分けの処理が追加されているが，これは 1 つの近傍に対して 1 度だけ行い，その計算時間は$O(N)$である．解の評価時間はグループ分け，制約チェック，目的関数値計算それぞれの計算時間の和なので，解の評価時間は既存手法と提案手法どちらも$O(N)$である．

表 1: 解の評価時間の比較

	既存手法	提案手法
グループ分け	-	$O(N)$
制約チェック	$O(N)$	$O(N)$
目的関数値計算	$O(N)$	$O(N)$
解の評価時間	$O(N)$	$O(N)$

次に総計算時間を考える．既存手法では近傍サイズは$O(N^3)$であるが, 提案手法では, $O(N^2)$に減っている．これにより，表 2 のように，総計算時間は提案手法では改善されている．

表 2: 総計算時間の比較

	既存手法	提案手法
解の評価時間	$O(N)$	$O(N)$
近傍サイズ	$O(N^3)$	$O(N^2)$
総計算時間	$O(N^4)$	$O(N^3)$

4.2 多スタートタブー探索法

本研究では，タブー探索法を用いて近傍探索を行う．選択した近傍の種類と選択した 2 日，あるいは 2 チームをタブーとして，探索する．探索を深く行うための集中化と局所解から離れるための多様化の二つに分けて説明する．

集中化においては，解が一定の範囲内のものを探索するために，スケジュール同士の距離を定義する．距離はスケジュールを表としてみたときに，異なるセルの数とする．探索の基準点となるスケジュールに対して，近傍移動によってスケジュールの距離が一定以上になったとき，基準スケジュールに戻り，タブーリストの一部を消して再度探索を繰り返す．この操作を一定回数行っても解が改善されないときに多様化を考える．

P_3-Packing を利用した初期解は，効率の良いツ

アーを作っており，近傍探索によって得られるスケジュールは初期解に大きく依存している．そこで，厳密最適解よりわずかに悪い程度のスケジュールを初期解として複数用意することによって多様化する．

5　数値実験

まず初めに，第 3 節で説明したP_3-Packing と，グループ単位のスケジュール生成の計算時間を表 3 に示す．チーム数 36 のグループ単位のスケジュールに関しては 3.4 節で説明したようにソルバーの計算が終了した時間ではなく，厳密解が求まったと確定するまでの時間を示している．

表 3: 数理計画ソルバーの計算時間（秒）

チーム数	24	30	36
P_3-Packing	1.8	829.8	15742.0
グループ単位のスケジュール	8	634	186000*

次に局所探索法を適用した結果得られた解の目的関数値を示す．チーム数 10 以下の問題例に対しては厳密解が求められており，チーム数 12 の問題例に関しては，下界と暫定解の差がタイトになっている．そこで，チーム数 16 以上の問題例のうち，チーム数$6k$と$6k+4$の問題例に対して，過去の最良解と，本研究の提案手法を用いて求めた自己最良解の比較を表 4 に示す．Galaxy に続く数字はチーム数を表しており，表中の数値は全てのチームの移動距離の和である．また，増減は$\frac{(\text{自己最良解})-(\text{過去最良解})}{(\text{過去最良解})}$であり，増減が負の問題例では過去最良解を更新している．

6　まとめと今後の課題

本研究では，巡回トーナメント問題に対して新しい初期解の構築法と近傍の設計，探索の方法を提案した．提案法により，ベンチマーク問題におけるチーム数 16 から 40 の問題例のうち，チーム数が 6 の倍数の全ての問題例で暫定解を更新した．チーム数$6k+4$の問題例に対する改善度合いよりも大幅に改善することができており，P_3-Packing を利用する初期解生成による効果が高いと考えられる．

今後の課題として，本研究では設計することができなかったチーム数$6k+2$の問題へのP_3-Packing の適用が挙げられる．

表 4: 過去最良解と自己最良解の比較

問題例	過去最良解	自己最良解	増減
Galaxy16	14900 *1	14774	-0.85%
Galaxy18	20907 *1	20743	-0.79%
Galaxy22	33901 *2	34074	0.51%
Galaxy24	45554 *3	44101	-3.29%
Galaxy28	75276 *2	75348	0.10%
Galaxy30	96580 *3	94444	-2.26%
Galaxy34	143298 *2	143710	0.29%
Galaxy36	173458 *3	167910	-3.30%
Galaxy40	241908 *2	241367	-0.22%

*1: Langford (2010)

*2: Goerigk, Hoshino, Kawarabayashi,and Westphal (2013)

*3: Hosoe and Imahori (2012)

参考文献

[1] K. Easton, G. L. Nemhauser, M. A. Trick: The Traveling Tournament Problem: Description and Benchmarks, Lecture Notes in Computer Science 2239, Springer, pp.580-585(2001).

[2] M. A. Trick: Challenge Traveling Tournament Instances http://mat.tepper.cmu.edu/TOURN/

[3] 細江卓矢　：リーグ戦スケジュールの構成法―移動距離の最小化を目指して―，名古屋大学大学院工学研究科計算理工学専攻，修士論文(2012).

[4] M. Goerigk, R. Hoshino, K. Kawarabayashi, S Westphal: Solving the Traveling Tournament Problem by Packing Three-Vertex Paths, AAAI, pp.2271-2277(2014).

[5] A. Anagnostopoulos, L. Michel, P. Van Hentenryck, Y. Vergados: A simulated annealing approach to the traveling tournament problem, Journal of Scheduling 9, pp. 177‐193 (2006).