巡回トーナメント問題に対する発見的解法の研究

鈴木　翔太　（沼田　一道　教授，松浦　隆文　助教）

1　はじめに

複数の個人やチームが他期間に渡って試合を行い優劣を競うリーグ戦においては，試合スケジュールの作成は不可欠である．スケジュールを作成するときには様々な条件を考慮し，良いスケジュールを作成することが求められる．しかし，条件が複雑になるにつれ人の手によりスケジュールを作成することは困難となる．そこで，近年プロスポーツのスケジュールを数理的な手法を用いて作成することが増えてきている．スポーツ競技において，試合日程，競技場，ホーム＆アウェイテーブル（以下 HAT）などを決める問題をスポーツスケジューリングという．特に競技場間の移動距離は移動コストと密接に関係するものであり，アメリカなど国土が広大な国では移動距離を減らすことでリーグ運営コストの削減を目指している．

J リーグのように複数のチームがお互いのホームとアウェイで一度ずつ対戦するリーグを 2 重総当たり戦と呼ぶ．2 重総当たり戦に対して全チームの総移動距離を最短とするスケジュールを求める問題を巡回トーナメント問題 (Traveling Tournament Problem，以下 TTP) と呼ぶ．TTP は 2001 年に Easton ら [1] により提起された．TTP は非常に難しい問題であることが知られており，分枝価格法に基づく厳密解法 [2]，シミュレーテッドアニーリングを用いた発見的解法 [3] などが提案されている．厳密解は 8 チームのいくつかの問題例まで求められているが，それ以上のチーム数では求まっていない．本研究では TTP に対し，新たな発見的解法を提案し，その性能を評価する．

2　問題

2.1　2 重総当たり戦の定義

2 重総当たり戦は各チームがそれぞれのチームと 2 回対戦する．すべてのチームは同一日に一度対戦を行い $2(n-1)$ 日でリーグが終了する．ただし n はチームの数で偶数である．対戦日のことを以後スロットと呼ぶ．2 重総当たり戦の満たすべき制約を以下に記す．

(CS1)　各チームは他のチームとホームで一度，アウェイで一度ずつ対戦をする．

(CS2)　各チームは各スロットで一度対戦をする．

(CS3)　各試合は対戦する 2 チームのどちらかの本拠地で行う．

表 1　2 重総当たり戦の例（@:アウェイ）

	1	2	3	4	5	6
1	3	2	@4	@2	@3	4
2	4	@1	3	1	@4	@3
3	@1	@4	2	4	1	@2
4	@2	3	@1	@3	2	1

表 1 の各枠について無印 (H) か@印 (A) だけを指定したものを HAT と呼ぶ (対戦相手は未定).

2.2　問題条件

各チームは自分の本拠地から出発しリーグ戦が終了したあと自分の本拠地に戻ってくる．アウェイの試合が連続するときはアウェイ間を直接移動する．本研究では，TTP の中でも特に，

(TTP1)　各チームのホームゲームの連続は高々 3 回まで．

(TTP2)　各チームのアウェイゲームの連続は高々 3 回まで．

(TTP3)　各チーム対は連続して戦わない (i 対 j の試合が行われたスロットの直後のスロットでの j 対 i の試合は禁止).

という制約を加えた問題例を扱う．

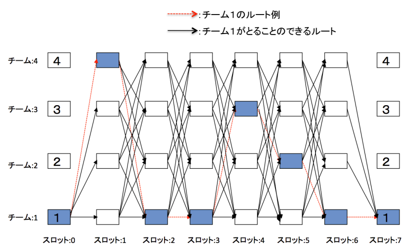

図 1　ルート例

3　定式化

TTP は巡回セールスマン問題と，HAT 許容性問題の二つの問題の要素を含んでいる．HAT 許容性問題とは，HAT から実行可能なスケジュールを作成できるかどうかを判定する問題である．全てのチームの本拠地に訪問するツアーのうち最短経路を求めることは巡回セールスマン問題に似ている．実行可能なスケジュールを作成するところは HAT 許容性問題に似ている．そこで本研究では TTP を二つの問題に分けて考える．まず，HAT を固定したもとでの最短移動距離スケジュールを求める問題について定式化を行う．

問題を構成する諸要素を以下の記号で表す．球場 i,j 間の距離を d_{ij}, チーム数を n(偶数), チームの集合を $T=\{1,2,...,n\}$, スロットの集合を $S=\{0,1,...,2(n-1)\}$ とする．今回の定式化では予め HAT を与えるので, チーム t のホームゲームスロットの集合:$HD(t)=\{hs^t_1,...,hs^t_{n-1}\}$, チーム t のアウェイゲームスロットの集合:$AD(t)=\{as^t_1,...,as^t_{n-1}\}$, スロット s のホームチームの集合:$HT(s)=\{ht^s_1,...,ht^s_{n/2}\}$, スロット s のアウェイチームの集合:$AT(s)=\{at^s_1,...,at^s_{n/2}\}$ とする．次に変数の定義を行う．x^t_{ijs} は，チーム t がスロット s で球場 i にいて，スロット $s+1$ で球場 j に移動するとき 1，そうでないときは 0 をとる決定変数である．y^t_{is} は，チーム t がスロット s で球場 i にいるとき 1，そうでないとき 0 をとる決定変数である．τ はチーム t がスロット s で対戦するチームである．以上の記号を用いて HAT を固定したもとでの最短移動距離スケジュールを求める問題を定式化すると，以下のようになる．

$$
(\mathrm{TTP}')\quad
\left|
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{t,i,j\in T}\sum_{s\in S} d_{ij}x^t_{ijs} & (1)\\
\text{subject to} & \displaystyle\sum_{t\in AT(s)} y^t_{is}=1 \quad \forall i\in HT(s),\forall s\in S & (2)\\
& \displaystyle\sum_{s\in HD(t)} y^\tau_{ts}=1 \quad \forall \tau\in T\setminus\{t\},\forall t\in T & (3)\\
& \displaystyle\sum_{s\in AD(t)} y^t_{\tau s}=1 \quad \forall \tau\in T\setminus\{t\},\forall t\in T & (4)\\
& \displaystyle\sum_{j\in T} x^t_{ijs}=y^t_{is} & (5)\\
& \displaystyle\sum_{i\in T} x^t_{ijs}=y^t_{j,s+1} & (6)\\
& y^t_{is}+y^i_{t,s+1}\le 1 \quad \forall s\in AD(t),\forall i,t\in T(i\neq t) & (7)\\
& y^t_{is}=\begin{cases}0 & i\neq t\\ 1 & i=t\end{cases} \quad \forall s\in HD(t),\forall t\in T & (8)\\
& y^t_{is}=0 \quad i=t,\ \forall s\in AD(t),\forall t\in T & (9)\\
& \displaystyle\sum_{i\in HT(s)} y^t_{is}=1 \quad \forall s\in AD(t),\forall t\in T & (10)
\end{array}
\right.
$$

(1) 式は全てのチームの移動距離の和を最小化する目的関数．(2) 式はスロット s でアウェイのチームが，スロット s でホームの球場のいずれかにいることを表している．(3), (4) 式はあるチームが, 各チームとホーム，アウェイで一度ずつ対戦することを表している．(5)，(6) 式はチームの移動を表している．(7) 式は各チーム対が連続して戦うことを禁止している．(8) 式はチーム t がホームにいるスロットでは，i と t の値が等しいことを表している．(9),(10) 式はチーム t がアウェイのスロットでは，同じスロットでホームとなるチームのどこかの球場におり，チーム t の本拠地にはいないことを表している．上記の定式化を汎用ソルバー Gurobi に入力して解いた結果を表 2 に示す．実験には，TTP のベンチマーク問題 [4] を用いた．6，8 チームについては比較的短い時間で厳密解が求まることがわかった．

表 2　Gurobi による (TTP′) の求解時間 (s)

問題	min	mean	std
NL6	0.02	0.074	0.059
NL8	0.66	2.322	2.929
NL10	85.67	196.23	91.386

4　提案解法

4.1　方針

実行可能な HAT を生成する問題と，ある HAT のもとで総移動距離を最短とするスケジュールを求める問題は，チーム数が少ない場合にはどちらも比較的短い時間で解くことができる．そこで先に述べたように TTP を 2 つの問題に分割して考える．また，HAT について直接的な局所探索を行うことは難しいので，{1,2,...,$2n(n-1)$} の順列を HAT に対応させ，順列に対して局所探索を行う．順列空間の探索においては，局所解に陥ることを防ぐためにタブーサーチを用いて解の改善を行う．

4.2　順列から HAT への写像

順列$\boldsymbol{\sigma}$には 1〜$2n(n-1)$ の番号が格納されており，順列$\boldsymbol{\sigma}$を用いて HAT を表す (図 2)．このとき満たすべき制約は，以下の通りである．

- 各チームはホーム，アウェイが $n-1$ ずつ．
- 各スロットはホーム，アウェイが $n/2$ ずつ．
- ホーム (アウェイ) の連続は 3 スロットまで．

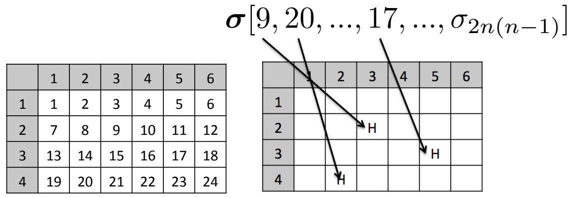

図 2　順列から HAT への変換

上記の制約を満たすよう順列$\boldsymbol{\sigma}$の先頭から HAT の対応する番号へホームを割り当てていく．制約を満たさない場合は次の番号へ進み，順列の最後まで到達したとき，HAT を出力する．HAT ができなかったときには，$\boldsymbol{\sigma}$をランダムに並べ替え再度 HAT への写像を行う．

4.3　解法の流れ

解法の説明に使う記号，関数の定義を行う．

$best\_l$: 最短移動距離　　len: 移動距離　　$count$: 近傍探索回数

Search($\boldsymbol{\sigma}$): 順列$\boldsymbol{\sigma}$の互換近傍探索を行い，最小の値を返す関数．

提案解法の流れを以下に記す．

Step0: (初期化)$best\_l = +\infty$，$count = 0$，初期順列$\boldsymbol{\sigma}$を生成する．

Step1: (HAT への写像)$\boldsymbol{\sigma}$の先頭から対応する枠に制約を満たすようホームを割り当てる．

Step2: (距離の計算)Step1 で得られた HAT に対して (TTP$'$) を解き，最短移動距離スケジュールを求める．

Step3: (近傍探索) 順列$\boldsymbol{\sigma}$に対して互換近傍探索を行い，len =Search($\boldsymbol{\sigma}$) とする．

Step4: (条件判定)$count = count + 1$．$best\_l > len$ の時，$best\_l = len$ とする．

Step5: (終了条件)$count = 100$ ならば探索を終了．そうでなければ Step3 へ．

4.4　互換近傍探索

現在の順列$\boldsymbol{\sigma}$の近傍の探索を行う．近傍は順列$\boldsymbol{\sigma}$の任意の 2 数を入れ替えた順列$\boldsymbol{\sigma}'$とする．順列$\boldsymbol{\sigma}'$から HAT への写像を行い，生成された HAT が現在の HAT と異なる時 (TTP$'$) を解く．順列$\boldsymbol{\sigma}$の近傍のうちタブーリストに含まれず，最も移動距離が短くなった順列$\boldsymbol{\sigma}^*$へ遷移する．次に$\boldsymbol{\sigma}^*$により生成された HAT の各チーム対を交換した場合について (TTP$'$) を解く．$\boldsymbol{\sigma}^*$により生成された HAT をタブーリストに追加する．タブーリストが埋まっているときは最も古いものを消去する．総移動距離が最短の値を返す．

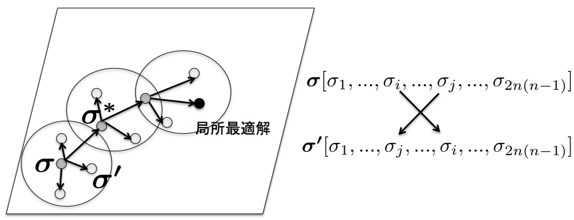

図 3　互換近傍探索

5　実験と考察

チーム数が 6 のベンチマーク問題 NL6[4] を用いて数値実験を行った．プログラムは c++ で作成し，(TTP′) の計算は，Gurobi Version3.0.2 を用いて解いた．NL6 について，タブーリスト長を 7，12，20 として各 10 回ずつ解いた．提案法の結果を表 3 に示す．なお時間は 10 回の平均値，最適値は 23916 である．

表 3　タブーリストの比較

タブーリスト	min	max	mean	std	Time(s)
7	24101	24728	24567	177.85	1518.8
12	23916	24672	26958	298.12	1561.1
20	23916	24672	24266	283.63	1360.6

本実験ではタブーリスト長が 12 と 20 のときに最適値が求まった．タブーリスト長が長くなると近傍の探索回数が減るので時間は短くなる傾向にある．最も最適値が求まる回数が多かったのはタブーリスト長が 20 のときであった．このときの平均値が最も小さく，タブーリスト長はある程度の長さを持たせた方がいい結果が得られると考えられる．

次にタブーリスト長を 20 とし，他の問題例について提案法を用いた結果を表 4 に示す．表中の回数とは，最適解が求められた回数のことである．

表 4　提案法の結果

問題	opt	min	max	mean	回数	Time(s)
NL6	23916	23916	24672	24266.4	3	1360.6
circ6	64	64	66	64.5	7	1353.5
galaxy6	1365	1365	1397	1376.4	4	1315.1

表 4 より，どの問題に対しても最適解を求めることができた．特に circ6 については最も良い結果が得られた．circ6 は円周上に開催地が分布しているモデルであり，このようなときに提案解法は有効であると考えられる．

提案解法では最適解は数回に一度しか求まらなかった．これは HAT を順列の写像を行うことにより生成しているので，順列に対して近傍探索を行うことが必ずしも HAT の近傍を探索していることにはならないためだと考えられる．また，6 チームの HAT パターンは膨大であり探索回数が少なく十分に解空間の探索ができていないとも考えられる．

6　まとめと今後の課題

本研究では，問題を二つに分割して考える独自の定式化を基本とする発見的解法を提案した．HAT が固定されたもとでは移動距離最短となるスケジュールは比較的早い時間で求まることがわかった．今後は最適解が求まる頻度をあげるためによりよい近傍の探索方法を考える必要がある．提案した方法では扱える問題のサイズがチーム数 6 までであり，それ以上のチーム数になると現実的な時間内に求まらないので，アプローチを含め提案解法を見直す必要がある．

参考文献

[1] Kelly Easton, George Nemhauser and Micheal Trick:“The Traveling Tournament Problem Description and Benchmarks”, In Seventh International Conference on the Principles and Practice of Constraint Programming(CP’01),pages 580-589,Paphos,Cyprus,2001.Springer-Verlag,LNCS 2239

[2] Stefan Irnich:“A new branch-and-price algorithm for the traveling tournament problem”,European Journal of Operational Research vol.204,pp.218-228(2010)

[3] Pascal Van Hentenryck and Yannis Vergados:“Traveling Touunament Scheduling: A Systematic Evaluation of Simulated Annealing”, Journal of Scheduling, 9(2),pp. 177-193(2006)

[4] Challenge Traveling Tournament Problems, URL http://mat.gsia.cmu.edu/TOURN/, 最終閲覧日 2010 年 12 月 23 日