数理解析研究所講究録
第 1981 巻 2016 年 27-44

時空間ネットワークを用いた 大都市近郊区間における最長経路探索

Solving the longest oneway route in Tokyo suburb area using a time-space network

名古屋大学大学院 工学研究科 *犬塚 幹夫 田地 宏一

INUZUKA Mikio and TAJI Kouichi

Department of Mechanical Science and Engineering, Graduate School of Engineering, NAGOYA UNIVERSITY

Abstract There is the section called ‘Tokyo suburb area’ in the JR line of Kanto region. In this section, passengers are allowed to take any route between two stations if the route satisfies the conditions such that the route is traversable, that the route does not go through a station in the route more than once and that the route can be passed through in one day. The longest such a route between adjacent two stations is often called ‘detour in Tokyo suburb area.’ In this paper, by using time-space network incorporated with an integer linear programming, we propose a solution method to find the longest detour in Tokyo suburb area passed through in one day, and some computational experiments show the validity and effectivity of the method.

1. はじめに

図 1[1] に示すのは，関東地方において定められている大都市近郊区間の路線図である．この区間では JR 東日本の旅客営業規則第 157 条 2 に示されている規則が適用される [2]．その規則には,「大都市近郊区間内を相互発着する普通乗車券を所持する旅客は，その区間内においてはその乗車券の券面に示された経路にかかわらず，同区間内の他の経路を選択して乗車することができる」とある．この規則を読み替えると，乗車経路の重複や途中下車などの規則に違反しない限り，すなわち，一筆書きの経路であれば大都市近郊区間内のどのような経路も乗車することが可能となると考えることができる．このような規則を利用して，隣接した駅に最短経路を利用せず，規則内で利用できる任意の経路を利用する乗車方法を大都市近郊区間大回り乗車と呼ぶ．本論文では，大都市近郊区間の中の東京近郊区間において，規則に違反せず，1 区間の乗車区間に対して 1 日で乗車できる最長経路を求める問題を考える．

本論文に関連する研究を 2 件紹介する．宮代ら [3][4] は，日本中に張り巡らされている JR 各社の鉄道網に対して，乗車経路が重複することなく最長の距離を乗車することができる問題に対し，整数計画法を用いて定式化することにより求めた．また，堀山ら [5] は，従来から提案されている整数計画法による定式化ではなく，ZDD と呼ばれる有向非巡回グラフによる組み合わせ集合の表現法を用い，東京近郊の大都市近郊区間において，通過する単線区間の距離が最大となる経路,「関東の駅百選」に指定された駅の通過数が最大となる経路を求めている．

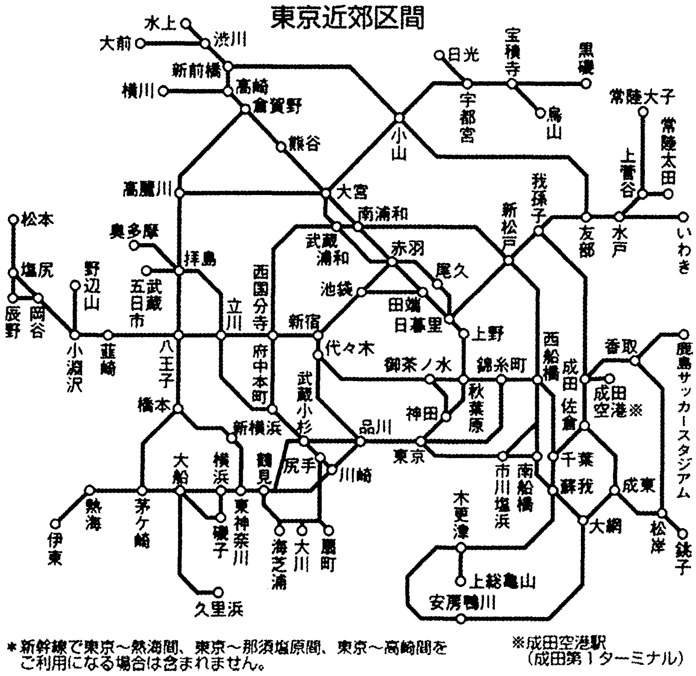


図 1: 東京近郊区間（2014 年時点）

本論文では，出発駅と終着駅が隣接した場合，すなわち，最低運賃の場合[1]の大都市近郊区間（東京近郊区間）大回り乗車の経路を整数計画問題として探索する．まず，宮代ら [4] の方法を用いて列車の運行時間を考慮しない場合の，単純な最長経路を探索した．しかし，得られた経路は 1 日で乗車することが不可能である．そこで，大都市近郊区間内を走行する列車を時空間ネットワーク上に表現し，時刻の制約条件を考慮して探索を行う．なお，取り扱う時刻表データは 2014 年 7 月時点のもの [14] を使用する．また，本論文では大都市近郊区間の中でも東京近郊区間のみを取り扱うため，以降は東京近郊区間と表記する．さらに，本論文で掲載する路線図は図 1 に示したものを参考に作成したため，必要に応じて参照されたい．

本論文の構成は以下の通りである．2 節では，東京近郊区間の紹介も兼ね，区間変更の歴史について触れる．3 節では，はじめに本論文で取り扱う東京近郊区間大回り乗車のルールを確認する．次に 2 節にて述べた各時点における東京近郊区間に対する時刻の制約を考慮しない場合の最長経路を探索する．4 節では，時刻制約を考慮した場合の最長経路探索の概要を述べ，東京近郊区間における最長経路を求めるための問題を，時空間ネットワークを用いた整数計画問題として定式化する．そして，その問題を実際に解き，結果を述べる．さらに，5 節では，乗換に必要な時間（待ち時間）を考慮に入れる場合に必要な制約式を追加する．これは，東京近郊区間には規模が大きな駅が数多く存在し，乗換に時間を要するため，実際に大回り乗車を行うためには必要な条件といえる．最後に 6 節で，まとめと今後の課題について述べる．

---

[1] 2015 年現在 [15]，最低運賃は山手線内，もしくは，電車特定区間内において，営業キロが 1～3km の場合に相当する．

2. 東京近郊区間の変遷

本節では，対象とする東京近郊区間の変遷を示す．

■ 1973 年〜1998 年 [9]

東京近郊区間は 1973 年に，それまで定められていた電車特定区間をベースに，1978 年の武蔵野線，1990 年の京葉線の開業により区間が拡大され，1998 年時点では，図 2 に示す区間となった．

■ 1999 年〜2003 年 [10]

1999 年に大幅な変更が加えられ，高崎駅周辺の高崎線・八高線，両毛線，水戸線などが追加された．さらに，2001 年 12 月には湘南新宿ラインの運行が開始され，図 3 に示すとおりとなった．

■ 2004 年〜2007 年 [11]

2004 年に，上越線，東北本線，中央本線，伊東線，常磐線，外房線の盲腸線部[2]が新たに加えられ，図 4 に示すようになった．

■ 2008 年 [12]

信越線，日光線，常磐線の盲腸線部が新たに加えられた（図 4）．

■ 2009 年〜2013 年 [13]

2008 年に内房線，外房線，成田線，総武本線のほかに，上越線，烏山線，中央本線，常磐線，久留里線，鹿島線の盲腸線部が加えられた．さらに，2010 年には横須賀線の武蔵小杉駅が開業した．(図 5)

■ 2014 年〜現在 [14]

図 6 に示した 2013 年までの区間に対し，吾妻線，中央本線，篠ノ井線，水郡線，小海線の盲腸線部が新たに加えられた（図 7）．

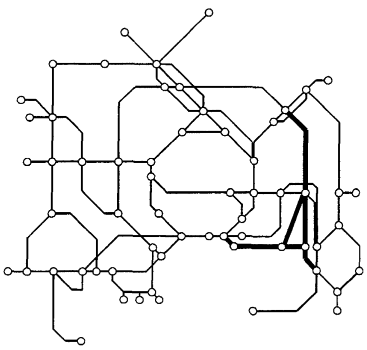


図 2: 東京近郊区間 (1998 年)

表 1: 1998 年までの東京近郊区間の歴史

年度	路線	区間	状況
1973	武蔵野線	府中本町-新松戸	開業
1975	京葉線	千葉貨タ[3]-蘇我	開業
1978	武蔵野線	新松戸-西船橋	開業
1986	京葉線	新木場以東	開業
1990	京葉線	東京-新木場	開業

[2]他の路線に接続しておらず，行き止まりの路線のこと．
[3]千葉貨物ターミナル

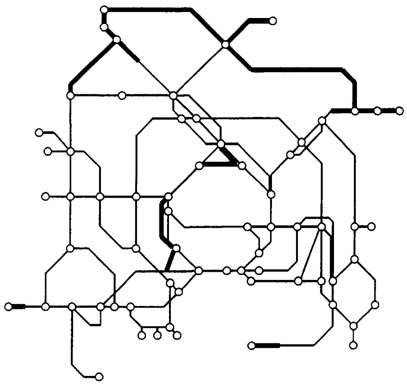


図 3: 東京近郊区間 (2003 年)

表 2: 2003 年までの東京近郊区間の歴史

年度	路線	区間	状況
1999	高崎線	熊谷-高崎	編入
1999	両毛線	高崎-小山	編入
1999	水戸線	小山-友部	編入
1999	常磐線	土浦-勝田	編入
1999	宇都宮線	小山-宇都宮	編入
1999	東海道線	平塚-熱海	編入
1999	内房線	木更津-君津	編入
2001	湘南新宿ライン	全線	開業

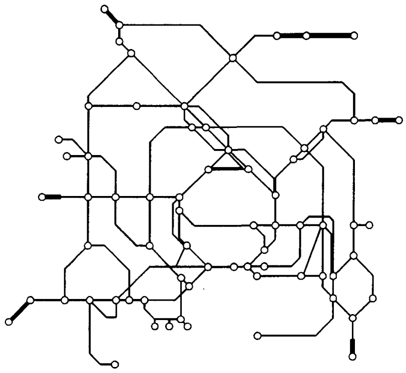


図 4: 東京近郊区間 (2007 年)

表 3: 2007 年までの東京近郊区間の歴史

年度	路線	区間	状況
2004	上越線	新前橋-渋川	編入
2004	宇都宮線	小山-黒磯	編入
2004	中央本線	大月-韮崎	編入
2004	常磐線	勝田-日立	編入
2004	伊東線	熱海-伊東	編入
2004	外房線	茂原-大原	編入

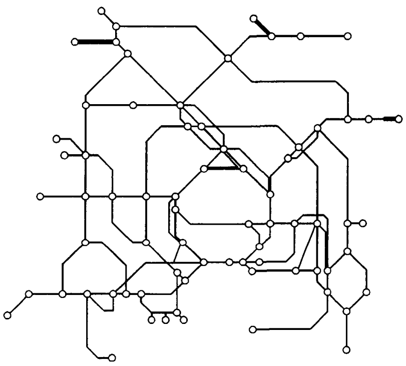


図 5: 東京近郊区間 (2008 年)

表 4: 2008 年までの東京近郊区間の歴史

年度	路線	区間	状況
2008	信越線	高崎-横川	編入
2008	日光線	宇都宮-日光	編入

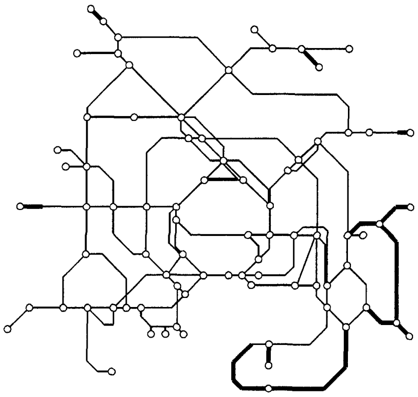


図 6: 東京近郊区間 (2013 年)

表 5: 2013 年までの東京近郊区間の歴史

年度	路線	区間	状況
2009	上越線	渋川-水上	編入
2009	烏山線	宝積寺-烏山	編入
2009	常磐線	日立-いわき	編入
2009	中央本線	韮崎-小淵沢	編入
2009	成田線	成田-松岸	編入
2009	総武本線	成東-銚子	編入
2009	鹿島線	香取-鹿島 SS[4]	編入
2009	内房線	君津-安房鴨川	編入
2009	外房線	大原-安房鴨川	編入
2009	久留里線	木更津-上総亀山	編入
2010	品鶴線	武蔵小杉駅	開業

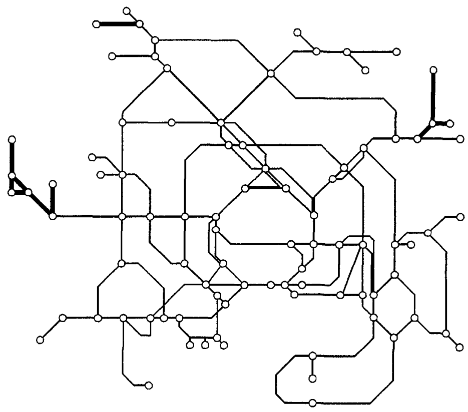


図 7: 東京近郊区間 (2014 年)

表 6: 2014 年までの東京近郊区間の歴史

年度	路線	区間	状況
2014	吾妻線	渋川-大前	編入
2014	水郡線	水戸-常陸大子・常陸太田	編入
2014	中央本線	小淵沢-塩尻	編入
2014	篠ノ井線	塩尻-松本	編入
2014	小海線	小淵沢-野辺山	編入

3. 時刻制約なしの最長経路探索

本節では，図 1 上での単純な最長経路を探索する．

3.1. 大回り乗車の乗車規則

はじめに，大都市近郊区間大回り乗車のルールを確認する．

条件 1 途中下車をしない，一筆書きができる経路を乗車する．すなわち，各路線および各駅は 1 度しか通過してはならないことを意味する．

条件 2 切符の有効期限は「1 日」である．この場合の「1 日」とは，始発から終電までの時間を意味し，24 時を経過することは認められる．本節ではこの条件を考慮に入れないが，4 節では考慮する．

[4]鹿島サッカースタジアム

上記のふたつの条件は，大都市近郊区間大回り乗車を行う際に必ず遵守しなければならない条件である．本論文では，先述の条件に加え，簡単のために次の条件を加えることにする．

条件 3 東京駅を出発駅とし，東京駅に隣接する，神田駅，有楽町駅，新日本橋駅，八丁堀駅のいずれかを到着駅とする．なお，運賃は，紙の切符では 140 円，IC カードでは 133 円である．

本論文ではこれら 3 つの条件に基づいた最長経路探索を行う．

3.2. 対象のネットワーク

東京駅を出発駅とし，その隣接駅を終着駅とする時刻の制約を考慮しない場合の最長経路を，東京近郊区間として指定されたネットワーク（図 2，4，6）に対して探索する．

3.3. 定式化

本節では，東京近郊区間における最長経路探索問題を整数計画問題として定式化する．

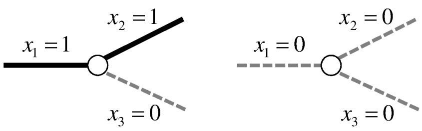
まず，目的関数について述べる．いま，x_i は各リンクを表す 0-1 変数であり，l_i は駅間距離を表すリンクのコストを表すとする．このとき，目的関数は式 (1) となる．

$$\max f(x) = \sum_{i=1} l_i x_i \tag{1}$$

この式は，$x_i = 1$ のときにそのリンクを通過し，$x_i = 0$ のときにそのリンクを通過しないことを表し，総乗車距離を最大化する問題であることを意味する．

次に，制約条件について述べる．3.1 節で述べた条件を遵守するため，以下に述べる 6 つの制約条件が必要である．

まず 1 つめが，通過駅に対して加える条件である．ネットワーク中の通過駅（図 8）では，そのノードに接続されているリンクのうち，使用するリンクの本数は 0 本（通過しない），もしくは 2 本（通過する）でなければならない．そのようなリンク同士の接続関係は，以下の 4 つの不等式 [3][4] を用いて表現できる．



通過する場合　　通過しない場合

図 8: 通過駅の例

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 2 \\ x_1 - x_2 - x_3 &\leq 0 \\ -x_1 + x_2 - x_3 &\leq 0 \\ -x_1 - x_2 + x_3 &\leq 0 \end{aligned} \tag{2}$$

2 つめは，ループの生成を回避するための条件である．式 (1) を目的関数として問題を解いた場合，図 9 のようなループが解に含まれてしまうことがある．そのため，ループを生成しないようにするための制約式を加える．例として，式 (3) に，図 9 に示すような 5 本のリンクにより構成されるループの形成を回避するための制約式を示す．ただし，すべてのループ回避条件を事前に追加すると，制約条件が非常に多くなるため，計算実験では，ループを含んだ解が算出されるごとに対応する条件式を追加し，解きなおすことにする．

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 4 \tag{3}$$

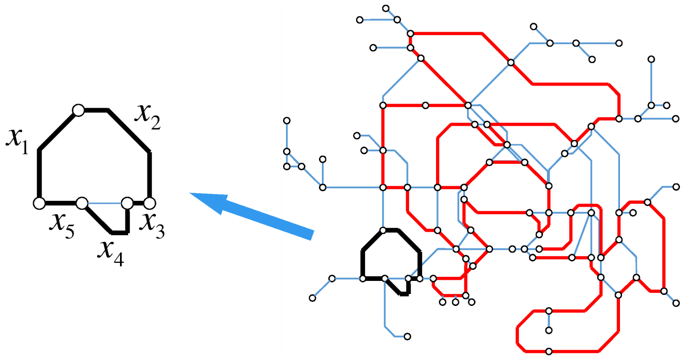


図 9: ループを含んだ解の例

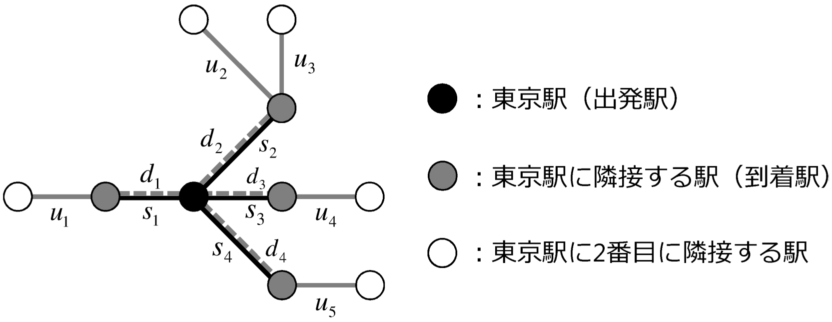


図 10: 出発駅・終着駅周辺のモデル

3 つめは出発駅を指定するための条件である．3.1 節にて述べた「条件 3」の通り，東京駅（図 10 の黒いノード）を出発駅とするため，東京駅に接続するリンクのうちのひとつを使用するという条件を加える．

$$s_1 + s_2 + s_3 + s_4 \leq 4 \tag{4}$$

4 つめは終着駅に関する条件である．終着駅（図 10 のグレーのリンク）では，接続するリンクは 1 つしか使用することができない．なぜなら，接続するリンクを 2 つ使用した場合，出発駅に戻ってきてしまうためである．そこで，図 10 に示すように，距離が 0 のダミーリンク（図 10 の破線）を考え，式 (2) と同様な制約式を加える．

$$\begin{aligned} u_1 + s_1 + d_1 &\leq 2 \\ u_1 - s_1 - d_1 &\leq 0 \\ -u_1 + s_1 - d_1 &\leq 0 \\ -u_1 - s_1 + d_1 &\leq 0 \end{aligned} \tag{5}$$

5 つめが同一路線扱いの異なる運転系統に対する制約条件である．例えば，図 11 に示す大宮-赤羽間には「宇都宮・高崎線」と「京浜東北線」の 2 系統の路線が乗り入れている．このふたつの路線は両方とも「東北本線」に所属しており，どちらか一方の路線を乗車した場合，「東北本線」を経由したとみなされ，規則上もう一方の路線を利用することはできない．そのため，このような経路を排除するための制約を加える必要がある．図 11 に示す，大宮-赤羽間

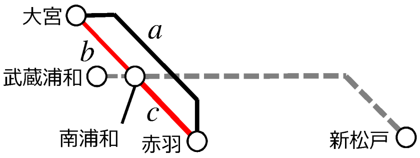


図 11: 南浦和駅周辺の路線のモデル

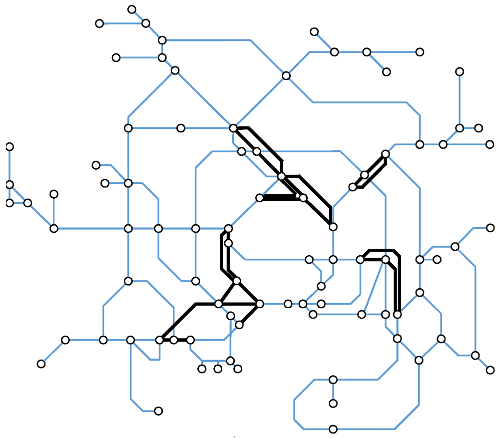


図 12: 制約を加えた路線

の宇都宮・高崎線の系統を a，大宮-南浦和間の京浜東北線の系統を b，南浦和-赤羽間の京浜東北線の系統を c とそれぞれ表す．このとき，該当区間を通過できるのは 1 回以下であること表現する制約式は式 (6) の 2 式であり，この制約式を図 12 に示す各区間に対して加える．

$$a + b \leq 1, \quad a + c \leq 1 \tag{6}$$

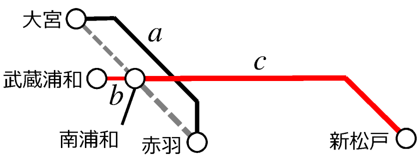


図 13: 南浦和駅周辺の路線のモデル

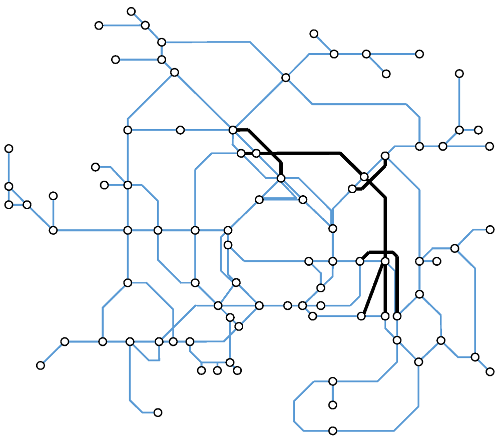


図 14: 制約を加えた箇所

最後に，快速線に対する制約条件を述べる．図 13 中の宇都宮・高崎線 (a) と武蔵野線 (b, c) は両線が南浦和駅において交差している．しかし，宇都宮・高崎線の列車は西浦和駅には停車しないが，通過扱いになるため，a と b, c は同時に使用することができない．そのことを表す制約式 (7) を図 14 に示すように，南浦和駅周辺のほか，新松戸駅，西船橋駅周辺に対しても加える．

$$a + b \leq 1, \quad a + c \leq 1 \tag{7}$$

3.4. 計算結果

本節で定式化した整数計画問題を整数計画ソルバーSCIPを用いて解いた[18][21]．なお，計算にはCPUをcore i7 4770，メモリを16GB搭載した計算機を用いた．

図2，3，6に示すネットワークに対し，時刻制約を考慮しない場合の最長経路を計算した．なお，2013年時点（図6）での場合では，変数は95個，制約式は278本であり，計算時間は1秒未満であった．

■ 1973年～1998年

図2の場合における最長経路は図15に示すとおりとなった．この場合の乗車距離は552.0キロである．この経路は1日で乗車可能である[9]．

■ 1999年～2003年

図3の場合における最長経路は図16に示すとおりとなった．この場合の乗車距離は841.9キロである．この経路は1日での乗車は不可能である[10]．

■ 2009年～2013年

図6の場合における最長経路は図17に示すとおりとなった．この場合の乗車距離は1030.5kmキロである．現在ではこの経路が時刻制約を考慮しない場合の最長経路となる．この経路は1日での乗車が不可能なため，時刻の制約を考慮した最長経路探索を行う必要がある[14]．そのための定式化を4節にて述べる．

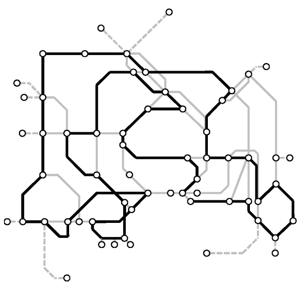


図15: 最長経路(@1998年)

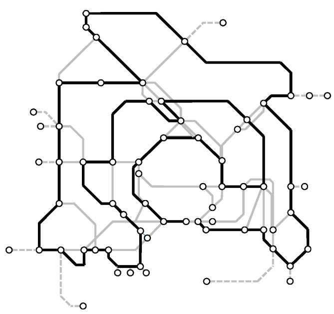


図16: 最長経路(@2003年)

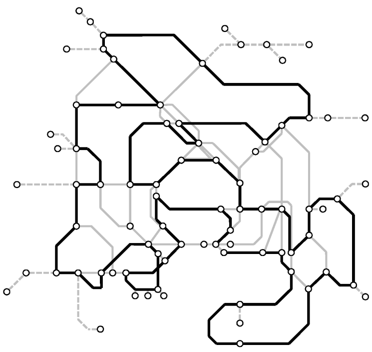


図17: 最長経路(@2013年)

4. 時刻制約を考慮した最長経路探索

前節では，時刻の制約を考慮しない場合における最長経路を求めた．その結果は図17に示したように総営業キロ数は1030.5kmであり，完乗するためには2日間を要する．

本節では，3.1節で述べた，時刻に関する条件,「条件2」を考慮に入れた場合の東京近郊区間における最長経路問題を，0-1整数計画問題として定式化する．

4.1. 時空間ネットワーク

時空間ネットワーク（以下，TSN:Time Space Network）とは，縦軸と横軸の位置に関する2次元の情報のみを持つネットワークに，時間の情報を持つ高さ方向の軸を加えた，3次元のネッ

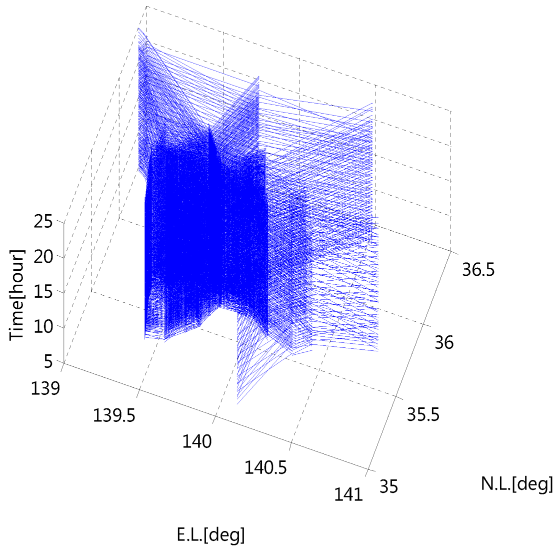


図 18: 対象の時空間ネットワーク

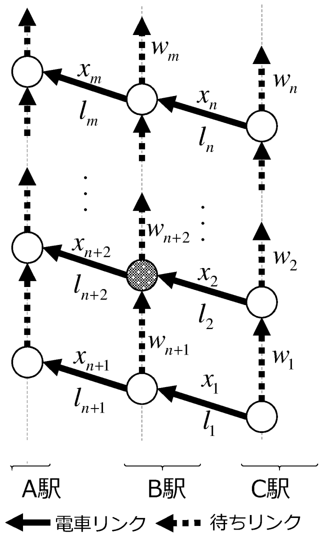


図 19: 各リンクの接続関係

トワークである．TSN を鉄道網に適用した事例は田口 [16][17] のものをはじめとして，多数挙げられる．

本問題で使用する TSN を図 18 に示す．これは，図 6 を時間軸方向に拡張したもので，図 18 には 30925 本のリンクが描かれている．また，TSN の接続関係を模式的に表したものを図 19 に示し，図中の変数の説明を以下に示す．

- $x_n \in \{0, 1\}$
  電車リンク変数：時空間ネットワーク中の列車や旅客の動きを表現する 0-1 変数である．
- l_n
  電車リンクの重み：駅間距離を表す定数である．
- $w_n^A \in \{0, 1\}$
  待ちリンク変数：時空間ネットワーク中の駅での乗換客の待ちを表現する 0-1 変数である．

4.2. 定式化

本節では，時刻の制約を考慮した場合の東京近郊区間における最長経路探索問題を 0-1 整数計画問題として定式化する．

まず，目的関数について述べる．図 19 中の変数を利用し，目的関数を式 (8) のようにとる．この式は式 (1) と同じであるが，リンクの意味が駅間から電車の走行区間に変わっていることに注意する．

$$\max\ f(x) = \sum_i l_i x_i \tag{8}$$

これは，$x_i = 1$ のときにその電車リンクを利用し，$x_i = 0$ のときにその電車リンクを利用しないことを表し，さらに，$w_i = 1$ のときにはその駅で待つ（乗り換える）ことを意味する．したがって，この目的関数は総乗車距離を最大化する問題であることを意味する．

次に，制約条件について述べる．まずひとつめの制約条件は通過駅に対する制約条件である．この条件には流量保存則を用いている．流量保存則とは，あるノードに対して流入する量と流

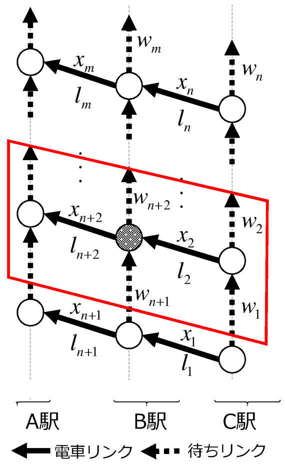


図 20: 通過駅に関する条件

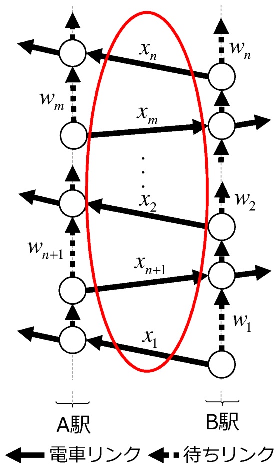


図 21: 一筆書き条件（路線）

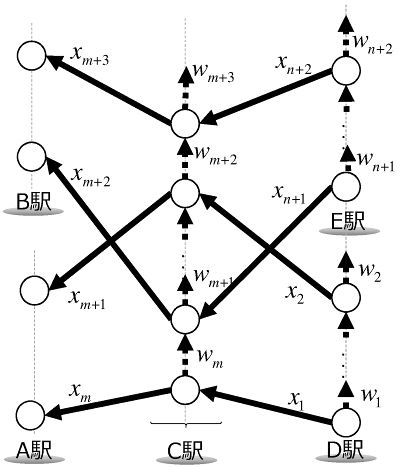


図 22: 一筆書き条件（駅）

出する量が等しいという法則である．

図 20 に示した網掛けにしたノードに着目すると，ノードに流入する量は x_2，w_{n+1} であり，流出する量は x_{n+2}，w_{n+2} である．それぞれに正の符号，負の符号を付し，数式に表すと式 (9) となる．

$$x_2 + w_{n+1} - x_{n+2} - w_{n+2} = 0 \tag{9}$$

この制約式を各ノードに対して加える．

前節の時刻制約を考慮しない場合の最長経路探索では，無向グラフのため，式 (2) にて述べた制約式がそれぞれのノードに対して必要であった．さらに，ネットワークの構造上，解に閉路が含まれる場合もあった．一方，本節で用いる時空間ネットワークは有向グラフのため，その構造から解には巡回路が含まれない．ただし，走行する一つ一つの列車をリンクとして表現するため，変数は大幅に増加する．

2 つの目の制約条件は一筆書きに関する条件である．本論文で考えるネットワークは，2 次元のネットワークを時間軸方向へ拡張した 3 次元のものである．走行するひとつひとつの列車をリンクとする構造上，

- 同じ路線を通るのは 1 回以下
- 同じ駅を通過するのは 1 回以下

という制約が必要である．前者は，図 21 中の A 駅と B 駅の間を走行する双方向の列車を利用するのは 1 回以下であり，式 (10) で表現される．

$$x_1 + x_{n+1} + x_2 + \cdots + x_m + x_n \leq 1 \tag{10}$$

また，後者は，図 22 中の変数を用いると，駅に進入するリンクに対して，

$$x_1 + x_2 + \cdots + x_{n+1} + x_{n+2} \leq 1 \tag{11}$$

という制約式，また駅から出発するリンクに対して，

$$x_m + x_{m+2} + \cdots + x_{m+1} + x_{m+3} \leq 1 \tag{12}$$

という 2 つの制約式で表現できる．

3 つめの制約条件として出発駅を指定するための条件を述べる．本論文では，東京駅を出発する各路線の始発列車を選択すると仮定する．図 23 に示すように，東京駅には隣接駅が 4 駅あり，その各始発列車の変数がそれぞれ s_1，s_2，s_3，s_4 であるとき，以下の制約式を課す．

$$s_1 + s_2 + s_3 + s_4 = 1 \tag{13}$$

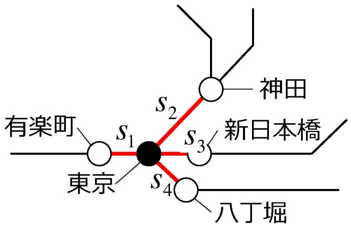


図 23: 出発駅条件

4 つめの制約条件は終着駅を指定するための制約条件である．終着駅では最終の待ちリンクが使用されるが，一筆書きの性質上，すべての駅の最終待ちリンクの中で使用されるものはただひとつとなる．東京駅を出発駅と指定しているため，終着駅は有楽町，神田，新日本橋，八丁堀いずれかである．これらの駅の最終待ちリンク変数がそれぞれ f_1，f_2，f_3，f_4 であるとき，

$$f_1 + f_2 + f_3 + f_4 = 1 \tag{14}$$

となる．

次に 5 つめが同一路線扱いの異なる運転系統に対する制約条件である．これは，式 (6) にて述べた制約式を，時空間ネットワークの構造のとおり，時間軸方向に拡張したものである．これを式に表現すると式 (15) となる．

$$\begin{gathered}\sum_{i}^{n} a_i + \sum_{j}^{n} b_j \leq 1 \\ \sum_{i}^{n} a_i + \sum_{k}^{n} c_k \leq 1\end{gathered} \tag{15}$$

そして最後に，快速線に対する制約条件を述べる．これについても先に述べたように，式 (7) にて述べた制約式を，時間軸方向に拡張したものである．これを式に表現すると式 (16) となる．

$$\begin{gathered}\sum_{i}^{n} a_i + \sum_{j}^{n} b_j \leq 1 \\ \sum_{i}^{n} a_i + \sum_{k}^{n} c_k \leq 1\end{gathered} \tag{16}$$

4.3. 制約式のまとめ

30925 本の電車リンク，30925 本の待ちリンクからなる TSN 上の 61850 個の変数に対し，上で述べた制約式を 31156 本加える．

3.4 節において述べた手法に加え，MATLAB2014a にて提供されている混合整数計画ソルバー「intlinprog」においても同様の数値実験を行った．

4.4. 計算結果

計算結果を表7，8，図24に示す．図24には得られた経路を2次元ネットワークにプロットしたものを，表7には得られた経路の内訳を示す．なお，SCIPによって計算した結果とMATLABによって計算した結果は一致した．

表7より，1節で述べた規則のとおり，一筆書きができる経路を求めることができ，4.2節での定式化は有効であると確認できた．

表 7: 経路の内訳

出発駅	時刻	到着駅	時刻	距離 [km]	乗換時間 [分]
東京	4:55	八丁堀	4:57	1.2	16
八丁堀	5:13	蘇我	6:00	41.8	11
蘇我	6:11	大網	6:56	19.1	3
大網	6:59	成東	7:21	13.8	7
成東	7:28	松岸	8:24	40.4	4
松岸	8:28	成田	9:40	62.3	33
成田	10:13	佐倉	10:27	13.1	12
佐倉	10:39	千葉	10:56	16.1	15
千葉	11:11	錦糸町	11:43	34.4	3
錦糸町	11:46	御茶ノ水	11:55	4.3	7
御茶ノ水	12:02	**西国分寺**	**12:45**	30.2	1
西国分寺	**12:46**	新松戸	13:43	53.6	24
新松戸	14:07	我孫子	14:20	10.6	7
我孫子	14:27	友部	15:41	67.5	8
友部	15:49	小山	16:56	50.2	6
小山	17:02	高崎	18:38	91.7	21
高崎	18:59	高麗川	20:28	65.3	18
高麗川	20:46	八王子	21:31	31.1	3
八王子	21:34	茅ヶ崎	22:50	42.1	5
茅ヶ崎	22:55	大船	23:07	12.1	5
大船	23:12	有楽町	24:21	53.0	-
				合計	753.9

表 8: 計算時間

ソルバー	計算時間 [s]
SCIP 3.1.1	550.8
MATLAB 2014a (intlinprog)	1764.8

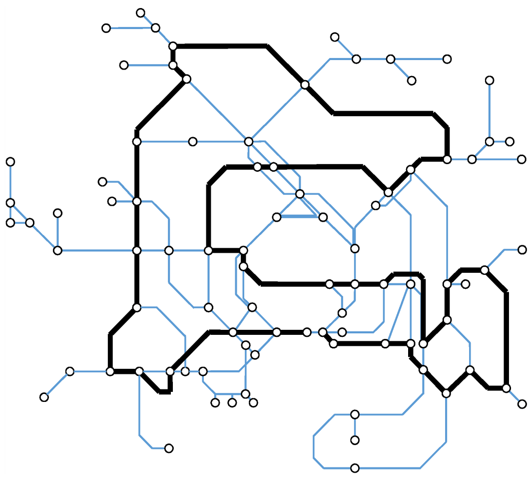


図 24: 2014 年時点での最長経路

5. 待ち時間を考慮した最長経路探索

4.4 節にて述べた結果において，西国分寺駅では 1 分の乗換時間（表 7 の太字）では乗換は困難と考えられる [5]．そこで本章では，東京近郊区間大回り乗車時に考慮が必要となる乗換時間を確保するための定式化を提案する．

5.1. 乗換時間の制約式

乗換時間を考慮するための定式化として求められる条件は次の 2 点が考えられる．

- 到着した列車に乗車したまま駅を通過する場合，待ち時間は考慮に入れなくて良い．
- 到着した列車とは異なる列車に乗車する場合，必要な乗換時間を確保する．

基本的な考え方は，各列車の発車時刻の差を待ち時間とし，その和が必要な乗り換え時間以上となるということである．この考え方の下，上記の 2 つの条件を満たす線形の制約条件について以下に述べる．そのために，以下の 2 つの記号を追加する（図 25）．

- t_n^A
  待ち時間変数：駅 A を出発する各列車の発車時間の差をとった定数である．
- t^{trs_B}
  必要な乗換時間定数：駅 B にて必要とする乗換時間を設定する変数である．

これらを用いると，乗換時間に対する制約式は以下の式 (17)，(18) で表現できる．

$$Mz^j \geq \sum_{i \in I_j} w_i^j \tag{17}$$

$$\sum_{i \in I_j} t_i^j w_i^j \geq t^{trs_j} z^j \tag{18}$$

これは big-M 法 [8] を利用した定式化であり，乗換時間を設定する駅に対して加える．z^j は 0 もしくは 1 をとる補助変数であり，乗換時間を設定する駅数分だけ加える（その分だけ変数が増加する）．M は big-M 法で用いられる十分大きな正の定数とする．

[5]中央線（1 番線・1F）から武蔵野線（4 番線・2F）への乗換

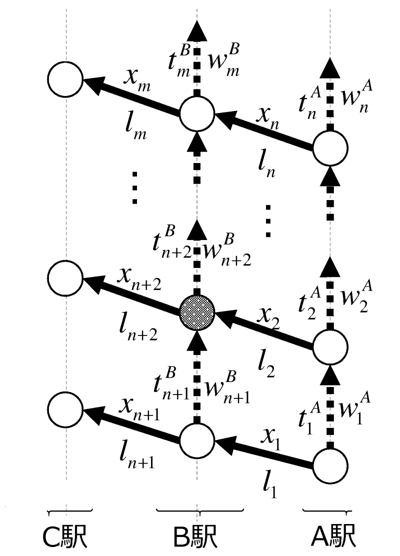


図 25: 接続関係

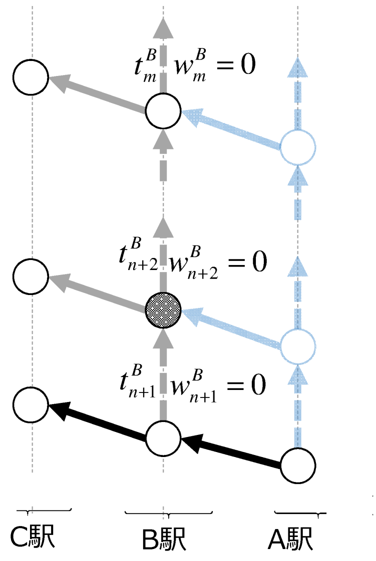


図 26: 定式化 (i)

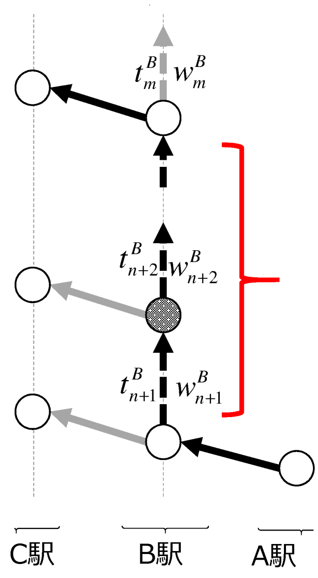


図 27: 定式化 (ii)

式 (17) では，左辺の値によって式の意味が変化する．

1. $\sum_{i\in I_j} w_i^j = 0$ のとき（図 26）
   式 (17) は，
   $$Mz^j \geq \sum_{i\in I_j} w_i^j = 0 \tag{19}$$
   より，
   $$z^B = 0 \tag{20}$$
   となる．従って，式 (18) は，
   $$\sum_{i\in I_j} t_i^j w_i^j = 0 \tag{21}$$
   となる．これは，乗換時間が 0 分で通過する，もしくは，駅自体不通過という取り扱い意味する．
2. $\sum_{i\in I_j} w_i^j \geq 0$ のとき（図 27）
   この場合分けは，駅 B を通過し，列車を乗り換える，もしくは列車の発着時刻に 1 分以上差が生じるときに発生する．この場合では，式 (17) が成立するために，$z^B = 1$ となる．従って，乗換時間は所望の乗換時間 t^{trs_B} 以上となり，題意を満たす．

5.2. 計算結果

計算機実験には CPU を core i7 4770，メモリを 16GB 搭載した計算機を用いた．また，計算に用いたソルバーは，SCIP，MATLAB2014a Optimization Toolbox の混合整数計画ソルバーである intlinprog コマンドである．

乗換時間を設定する駅を西国分寺とした．さらに，乗換時間を $t^{trs_j} = 10$ とし，$M = 20$ とした．

上記の環境・条件にて計算した結果を表 9，10，図 28 に示す．

4 節の結果と比較すると，経路が大きく変化した．表 9 中の下線部において，指定した 10 分の乗換時間が確保されていることが確認できたが，太字部において乗換不可能な解となった．これは式 (17)，(18) では乗換時間が 0 分の場合，制約式がアクティブにならないためであり，今後の課題とする．

表 9: 経路の内訳

出発駅	時刻	到着駅	時刻	距離 [km]	乗換時間 [分]
東京	4:42	有楽町	4:43	0.8	13
有楽町	4:56	品川	5:05	6.0	5
品川	5:10	横浜	5:28	22.0	4
横浜	5:32	大船	6:02	22.1	2
大船	6:04	茅ヶ崎	6:16	12.1	10
茅ヶ崎	6:26	八王子	7:48	42.1	5
八王子	7:53	立川	8:04	9.9	4
立川	8:08	拝島	8:20	6.9	2
拝島	8:22	高麗川	8:49	21.2	18
高麗川	9:07	倉賀野	10:40	60.9	55
倉賀野	11:35	**大宮**	**12:47**	70.3	**0**
大宮	**12:47**	小山	13:37	50.3	26
小山	14:03	友部	15:08	50.2	7
友部	15:15	我孫子	16:32	67.5	15
我孫子	16:47	成田	17:28	32.9	12
成田	17:40	松岸	18:57	62.3	42
松岸	19:39	千葉	21:28	78.1	6
千葉	21:34	西船橋	21:58	18.6	2
西船橋	22:00	<u>西国分寺</u>	<u>23:12</u>	67.9	<u>24</u>
<u>西国分寺</u>	<u>23:36</u>	新宿	24:10	22.5	9
新宿	24:19	田端	24:37	10.0	5
田端	24:42	神田	24:54	5.8	-
				合計	740.4

表 10: 計算時間

ソルバー	計算時間 [s]
SCIP 3.1.1	450.3
MATLAB 2014a (intlinprog)	889.6

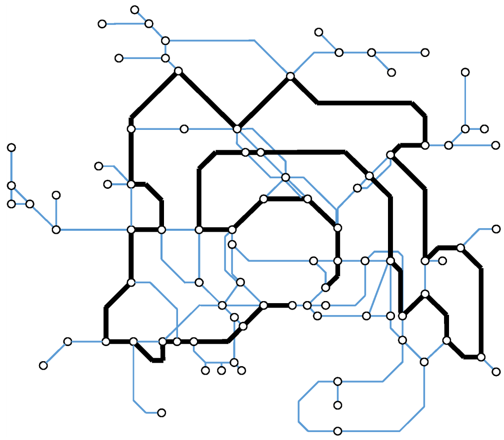


図 28: 乗換時間を考慮した場合の最長経路

6. おわりに

東京近郊区間の時刻制約付き最長経路探索問題を時空間ネットワークに基づく 0-1 整数計画問題として定式化した．4.2 章での定式化を利用することで所望の結果を得ることができ，時空間ネットワークを用いた定式化の有効性を確認することができた．さらに，これまでの成果に加え，乗換時間を考慮するための定式化を提案した．

今後の課題は以下のとおり．本論文では出発駅を東京駅として指定したが，この条件を緩和することが挙げられる．例えば，出発駅の候補として，山手線の始発列車が設定されている田町駅，品川駅，大崎駅，池袋駅が挙げられる．また，乗換時間を確保する制約式として big-M 法による方法を提案したが，現状，最良の方法でないことが計算結果からわかった．これは，乗換の条件には直通の列車で乗換えが必要ないものと，乗換が必要なものとの区別がなされていないためで，さまざまな点に再考・改良の余地があると考える．

参考文献

[1] 「JR 東日本：きっぷに関するご案内」，https://www.jreast.co.jp/kippu/1103.html，2014.12.12 閲覧

[2] 「JR 東日本 旅客営業規則」，https://www.jreast.co.jp/ryokaku/index.html，2014.12.13 閲覧

[3] 宮代，葛西，「最長片道切符」，オペレーションズ・リサーチ：経営の科学 49(1), 15-20, 2004-01-01

[4] 葛西隆也，「最長片道きっぷの経路を求める」，http://www.swa.gr.jp/lop/，2014.12.12 閲覧

[5] 堀山，羽室，「大都市近郊区間の経路の効率的な列挙と検索」，オペレーションズリサーチ：経営の科学 60(10), 3-9, 2015-10-01

[6] 湊 真一，「BDD/ZDD を用いたグラフ列挙索引化技法」，オペレーションズリサーチ：経営の科学 57(11), 10-17,2012-12-01

[7] 川原ら，「ZDD によるパスの列挙」，数理解析研究所講究録，第 1744 巻，2011 年，35-41

[8] 高松瑞代，「バス時刻表の最適化」，オペレーションズ・リサーチ：経営の科学 60(9), 4-8, 2015-09-01

[9] 「マイライン東京時刻表 1998 年 9 月号」，p1072，交通新聞社，1998
[10] 「マイライン東京時刻表 2003 年 4 月号」，p1104，交通新聞社，2003
[11] 「マイライン東京時刻表 2007 年 4 月号」，p1088，交通新聞社，2007
[12] 「マイライン東京時刻表 2008 年 4 月号」，p1088，交通新聞社，2008
[13] 「マイライン東京時刻表 2013 年 4 月号」，p1104，交通新聞社，2013
[14] 「マイライン東京時刻表 2014 年 7 月号」，p1104，交通新聞社，2014
[15] 「マイライン東京時刻表 2015 年 10 月号」，p1104，交通新聞社，2015
[16] 田口東,「首都圏電車ネットワーク上の時間変化する乗客分布の計算 (輸送・交通 (1))」，日本オペレーションズ・リサーチ学会秋季研究発表会アブストラクト集 2003, 30-31, 2003-09-10
[17] 安井ら,「大規模最短路問題に対するダイクストラ法の高速化」，数理解析研究所講究録，第 1726 巻，2011 年，62-72
[18] Tobias Achterberg, SCIP: solving constraint integer programs, Mathematical Programming Computation, volume 1, number 1, pages 141, 2009.
[19] Timo et al.,「SCIP Optimization Suite を利用した混合整数（線形/非線形）計画問題の解法」，ZIB-Report 12-24 (July 2012), https://opus4.kobv.de/opus4-matheon/frontdoor/index/index/docId/1160
[20] 宮代，松井,「ここまで解ける整数計画」，システム/ 制御/情報，Vol.50，No.9，pp.363-368，2006
[21] 宮代隆平，整数計画ソルバー入門，オペレーションズ・リサーチ，57(4) (2012)，pp183-189