

# 時空間ネットワークを用いたフライトパターンの列挙について

2009SE004 秋田大輔 2009SE100 加藤遥 2009SE175 森口元気

指導教員：鈴木敦夫

## 1 はじめに

### 1.1 背景と研究目的

クルーのスケジューリング方法には、1日または数日のスケジュールパターンを作成し、それをクルーに割り当てる方法がよく用いられる。このスケジュールパターンとは、1日または数日間において、クルーが基点となる場所から出発し、休憩をはさみながら基点となる場所に戻るまでの一連の動きを表したものである。近年までに、このスケジュールパターンを用いたクルーのスケジューリングは多数発表されている。その一部を以下に示す。

- 鉄道乗務員スケジューリング問題に対する解法とその評価 [1]
- 鉄道乗務員スケジューリング問題に対する列生成法によるアプローチ [2]
- 不定期便に対応したパイロット乗務スケジューリング [4]

これらの研究の伝統的なアプローチには、集合被覆問題 (Set Covering Problem, SCP) もしくは集合分割問題 (Set Partitioning Problem, SPP) などが一般的である。しかし、これらの解法では最適解を求められなかったり、計算に時間がかかってしまう場合がある。そこで、スケジュールパターンの作成とクルーへの割り当てを別々に行うことで計算時間を短縮できると考える。本研究では、その中でもスケジュールパターンの作成に着目する。

スケジュールパターンの作成にあたり、新しいアプローチを試みるため、時空間ネットワークを用いる。時空間ネットワークは、2005年に中央大学の田口東教授によって考案された、航空網や鉄道網の時刻表を3次元のネットワークで表現したものである [3]。鉄道網などの評価や分析を行う際に有用な手段として利用されている。しかし、スケジュールリング問題への適用例はまだ発表されていない。そこで、本研究では時空間ネットワークの新しい適用分野としてスケジュールパターンの自動作成を試みる。

スケジュールパターンを作成するスケジューリング方法は、航空会社に限らず鉄道会社なども類似した方法を用いている。スケジュールパターンを自動的にかつ短時間で列挙することができれば、航空会社や鉄道会社のスケジュール作成にかかる時間を短縮できる。

### 1.2 現状と改善案

本研究では、ある地域航空会社に焦点をあてて研究を行う。地域航空会社は、近距離の2点を中心に小型航空機を用いて航空輸送サービスを行っている。小規模な航空輸送サービスを行う地域航空会社では、大規模な航空会社が

用いるような大規模で高価なシステムを導入するのではなく、より安価なシステムを求めている。よって、本研究では地域航空会社のように小規模の航空会社に向けたシステムを考える。

現在、この航空会社では、目視と手作業によってクルーのスケジュールパターンを作成している。クルーごとに勤務時間、フライト時間などを考慮しながら、便ごとにクルーの人数などの条件を満たさなければならないため、手作業では膨大な時間がかかる。そこで本研究では、時空間ネットワークを基に以下の方法でスケジュールパターンの列挙の自動化を行う。

第1の方法では、数理計画問題としてモデル化し、最短経路を繰り返し求める。

第2の方法では、ネットワーク算法を用いる。グラフ探索のアルゴリズムを実現するプログラムを作成する。

この2つの方法でスケジュールパターンを列挙し、比較を行う。

## 2 時空間ネットワーク

### 2.1 時空間ネットワークの概要

時空間ネットワークとは、航空網や鉄道網などの2次元ネットワークに時間軸を加え拡張した3次元のネットワークである。2005年に中央大学の田口東教授によって考案された。時刻表や移動時間などを用いて、移動と時間の経過を同時に表現できる。2次元ではできない、より現実的なシミュレーションが可能となる。

通常の航空ネットワークでは、空港をノードと定義し、空港間にリンクを張った2次元ネットワークで表現される。しかし、本研究で用いる時空間ネットワークでは航空機の移動とクルーの乗り換えをリンクで表現し、各空港における航空機の発着ごとにノードを定義することで航空ネットワークを時間軸方向に拡張する (図1)。また、時空間ネットワーク上では、時刻表の時間を全て [分] 単位で扱う。

時空間ネットワーク上では場所・時間が同時に表現できる。そのため、最初に出発する場所・時刻から、目的となる場所・時刻までの移動経路がリンクを辿ることで求められる。

### 2.2 時空間ネットワークの構築

本研究で用いる航空網の時空間ネットワークでは、クルーが基点となる空港で搭乗してからその空港に戻るまでの行動を以下のように分類する。

- 航空機に搭乗して空港間を移動する
- 搭乗するべき便の出発を待つ

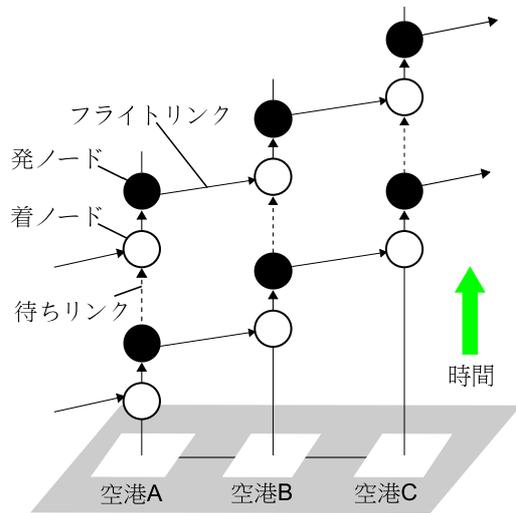


図1 時空間ネットワーク

これらの行動をノードとリンクとして以下のように定義して、時空間ネットワークを構築する。

- 着ノード：各空港における各航空機の到着
- 発ノード：各空港における各航空機の出発
- フライトリンク：航空機で次の空港へ移動する行動
- 待ちリンク：搭乗するべき便を待つ行動

### 2.2.1 フライトリンク

フライトリンクとは、クルーが航空機に搭乗して次の空港へ移動する行動を指す。発ノードから着ノードをつなぐリンクである(図2)。

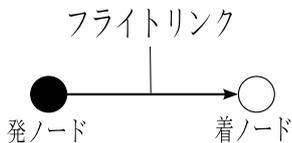


図2 フライトリンク

### 2.2.2 待ちリンク

待ちリンクとはクルーが空港で搭乗するべき便を待つ行動を指す(図3)。

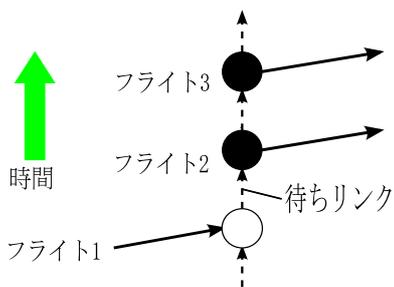


図3 待ちリンク

## 2.3 航空網の時空間ネットワーク

本研究では、対象航空会社の時刻表データ [5] を用いて時空間ネットワークを作成する。この航空会社は10ヶ所の空港を使用しており、1日に40便運航している。スケジュールパターンの基点となる空港としては、富士山静岡空港(以下、静岡)と名古屋小牧空港(以下、小牧)を使用している。空港間の距離データに関しては、実際の空港の座標を取得し、各空港間の直線距離を算出したデータを使用する。この距離データは [km] 単位で表す。時空間ネットワーク(図4)の規模を表すノード・リンク数を表1に示す。



図4 時空間ネットワーク(1日分)

表1 時空間ネットワークの規模(1日分)

ノード数	80
リンク数	110
フライトリンク数	40
待ちリンク数	70

## 3 数理計画モデル

本研究では、全ての経路を列挙するために最短経路問題として定式化し、繰り返し解く。各リンクに実際の距離を重みとして与え、ネットワークに1フロー流すことを仮定して定式化する。

### 3.1 記号の定義

添字集合、定数、変数を以下のように定義する。

添字集合

$N$ : ノードの集合

定数

$l_{ij}$ : ノード  $i \in N$  からノード  $j \in N$  への経路の距離

$s$  : 経路の始点, ただし  $s \in N$

$t$  : 経路の終点, ただし  $t \in N$

$a$  : 1日のフライト回数制限

$b$  : 1日のフライト時間制限

$F_{ij}$  : ノード  $i \in N$  からノード  $j \in N$  への経路がフライトリンクなら 1, 待ちリンクなら 0

$T_{ij}$  : ノード  $i \in N$  からノード  $j \in N$  へ移動するのにかかる時間

#### 変数

$x_{ij}$  : ノード  $i \in N$  からノード  $j \in N$  への経路を使用するなら 1, 使用しないなら 0

#### その他

1 回前の目的関数値よりも大きいという条件を加えて繰り返し解くことを前提としている. よって 1 回前の目的関数値を格納するための文字を定義する必要がある.

$z$  : 1 回前の目的関数値 + 0.01 (目的関数値の初期値は 0)

### 3.1.1 定式化

#### 目的関数

$$\min \sum_{i \in N} \sum_{j \in N} l_{ij} x_{ij} + 0.001 \sum_{i \in N} \sum_{j \in N} x_{ij} \quad (1)$$

#### 制約条件

$$\sum_{j \in N} x_{sj} = 1 \quad (2)$$

$$\sum_{i \in N} x_{it} = 1 \quad (3)$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = 0 \quad (i \in N \setminus \{s, t\}) \quad (4)$$

$$\sum_{i \in N} \sum_{j \in N} F_{ij} x_{ij} \leq a \quad (5)$$

$$\sum_{i \in N} \sum_{j \in N} F_{ij} T_{ij} x_{ij} \leq b \quad (6)$$

$$\sum_{i \in N} \sum_{j \in N} l_{ij} x_{ij} \geq z \quad (7)$$

$$x_{ij} = 0, 1 \quad (i \in N, j \in N) \quad (8)$$

#### 目的関数の説明

式 (1) : 使用する経路の総距離を最小化する関数 (ただし, 総距離の等しい経路を無くすために, 経路数を微小な値として加えている)

#### 制約条件の説明

式 (2) :  $s$  から出る経路はただ 1 つであるという制約

式 (3) :  $t$  に入る経路はただ 1 つであるという制約

式 (4) :  $i \in N \setminus \{s, t\}$  において,  $i$  から出る経路の数から  $i$  に入る経路の数を引くと 0 である制約 (流量保存則)

式 (5) : 1 日のフライト回数は  $a$  以下であるという制約

式 (6) : 1 日のフライト時間は  $b$  以下であるという制約

式 (7) : 目的関数値は 1 回前の目的関数値よりも大きい制約 ( $z$  の値は実行するごとに更新される)

式 (8) : 経路を使用するかしないかのバイナリー変数制約

## 3.2 計算するにあたって

### 3.2.1 使用したソフトウェア

最適化計算には, LINDO System Inc. 社の LINGO11.0 を使用する. データの入出力や解の列挙にあたり, Microsoft 社の Excel に搭載されている Visual Basic for Applications (以下 Excel VBA とする) を用いる.

LINGO は Excel VBA を用いて実行することができるので, 今回のように新たなデータを加えながら繰り返し実行させ, Excel のシート上に列挙する際に利用しやすい. よって今回はこのソフトウェアを使用した.

### 3.2.2 データの処理

計算するにあたって, 各ノードに番号を与える. 時間軸方向に逆行するリンクがないように, 各ノードに関して時刻の昇順にソートを行い, その順に番号を指定する.

始点と終点である  $s, t$  の値は, 基点となる空港の最初のノード番号と最後のノード番号となる. 基点となる空港は複数あるため, その空港ごとに計算を行う.

各ノード間の距離は以下のように指定する.

- 待ちリンクでの移動距離 :  $[0, 100)$  の一様乱数

- フライトリンクでの移動距離 : 空港間距離

- リンクのないノード間の距離 : 十分に大きな値

待ちリンクに乱数を加えることで, 等距離の経路を無くし, 全ての経路を列挙できるように設定する.

## 3.3 スケジュールパターンの列挙

### 3.3.1 LINGO による計算結果

#### 1. 小牧空港を基点としたスケジュールパターン

始点  $s=1$ , 終点  $t=80$  で計算を行う.

1 回目の計算結果を以下の表 2 に示す. ただし, 着発に関しては, 0 が着ノード, 1 が発ノードを表している. 空港名に変化がない経路は, 空港内での待ちリンクを表す. また, 便名に変化がない経路は, フライトリンクを表す.

表 2 ノード 1 からノード 80 の 1 回目の経路

ノード	空港	時刻	着発	便名
1	小牧	7:30	1	351
2	小牧	7:50	1	371
7	新潟	8:50	0	371
10	新潟	9:20	1	502
35	新潟	13:00	0	373
38	新潟	13:30	1	374
69	新潟	19:30	0	507
74	新潟	20:00	1	376
78	小牧	21:05	0	376
79	小牧	21:20	0	314
80	小牧	21:30	0	368

計算時間は 2 秒未満で, 目的関数値は 1040.637 である.

この経路は小牧空港と新潟空港を往復する経路である。

この結果を用いて 2 回目の実行を行う。1 回目の目的関数値 (つまり 1040.637) を  $z$  に代入し、再び同じ式を用いて計算させる。これにより、2 番めに距離が小さい経路を求めることができる。このような順序で、目的関数値が 100000 より小さい範囲で繰り返し計算させる。Excel VBA でループさせ、自動列挙を行ったところ、計算時間は約 8 分で 169 通りの経路を求めることができた。

## 2. 静岡空港を基点としたスケジュールパターン

始点  $s=3$ 、終点  $t=72$  で計算を行う。小牧を基点とする場合と同様の処理を行った結果、計算時間約 2 分で 17 通りの経路を求めることができた。

以上より、このモデルを計算した結果、スケジュールパターンは以下の表 3 の通りであるとわかる。

表 3 スケジュールパターンの合計

基点となる空港	パターン数
小牧空港	169 通り
静岡空港	17 通り
合計	186 通り

### 3.3.2 条件の追加

3.3.1 のスケジュールパターンには、以下のような現実的でない経路も含まれている。

- 待ちリンクのみを辿る経路
- フライトの間隔が 1 時間を超える経路
- 勤務時間が 14 時間を超える経路

これらを省くために、新たに勤務時間制約とフライト間隔制約を追加すると、線形モデルでなくなり、1 回の計算時間が膨大になってしまう。そのため、定式化を行うのではなく、以下の条件に当てはまる場合のみ Excel VBA で書き込みを行うようにモデル化する。

このとき加えた条件は以下の 2 つである。

- (1) 1 日の勤務時間は 14 時間以内
- (2) フライト間隔は 25 分~60 分

(1) の勤務時間の条件において、出勤時間を最初のフライトの 1 時間前、退勤時間を最後のフライトの 1 時間後とした。

この 2 つの条件を加えた場合、スケジュールパターンの数は表 4 のようになる。

表 4 スケジュールパターンの合計 (条件追加後)

基点となる空港	パターン数
小牧空港	16 通り
静岡空港	8 通り
合計	24 通り

これらのスケジュールパターンは、1 日で考えられる制約・条件を全て満たしている経路なので、現実的に実行可能である。

### 3.3.3 ガントチャートでの表示

前述したように、経路の列挙は可能であったが、表 2 では他の経路との比較が困難である。そのため、ガントチャートを作成し、視覚的に経路の比較を容易にする。

ガントチャートとは、作業計画およびスケジュールを横型棒グラフで示した工程管理図である。一般的には、プロジェクト管理や生産管理などで使われるもので、縦軸に作業をおき、横軸に時間 (期間) をとることで各作業の所要時間を横棒で示す。本研究では、ガントチャートの縦軸にスケジュールパターンを、横軸に時刻をとり、フライト時間を横棒で示す。

図 5 はガントチャートの一部であり、Excel シート 1 枚で表せている。

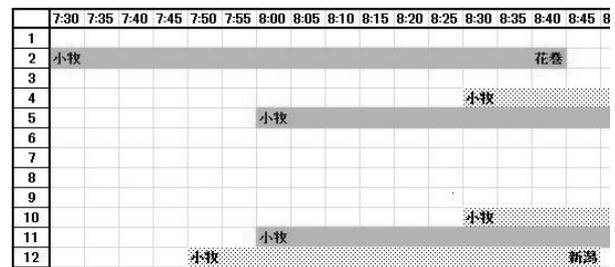


図 5 ガントチャート (一部)

このように表示するにあたり、LINGO の実行からガントチャートへの表示までを 1 回のクリックで繰り返し行えるようにした。ガントチャートを横に見ていくことで、搭乗すべき空港や時間、フライト間の時間が視覚的にわかる。

## 4 ネットワーク算法

### 4.1 概要

グラフ探索のアルゴリズムを用いてスケジュールパターンを列挙する。

プログラムは、Microsoft Visual C++ 2010 Express を用いて作成した。距離データ、空港データ、フライトデータを読み込み、経路探索を行う。

図 6 は木構造を深さ優先探索で辿る順番を表す [6]。

木構造は頂点と辺で 1 周できる部分がなく、全ての頂点がつながっているグラフである。本研究では木構造を作成し、深さ優先探索にて経路を全て列挙する。

深さ優先探索とは、可能な限り左に深く進み、行き止まり (子のないノード) まで探索した後、後戻りして最も近くの探索の終わっていないノードに進みながらノードを探索していく方法である。

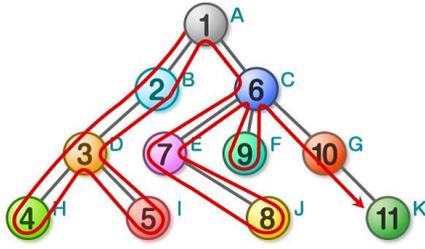


図6 深さ優先探索

## 4.2 アルゴリズムの詳細

注目しているノード(以下、注目ノード)の子は、以下のよう指定する。

- 待ちリンクを辿って到着する場合：左の子
- フライトリンクを辿って到着する場合：右の子

とする。また、探索を行う際のアルゴリズムは、以下のようになっている。

1. 注目ノードを記録する。
2. 左の子の有無を確認する。ある場合は注目ノードを更新してステップ1に戻り、ない場合は次のステップへ進む。
3. 右の子の有無を確認する。ある場合は注目ノードを更新してステップ1に戻り、ない場合は次のステップへ進む。
4. 注目ノードが終点かどうかを判別する。終点の場合は経路を保存し、ステップ5に進む。終点でない場合は、注目ノードのみを経路から除外し、ステップ5に進む。
5. 親から見て注目ノードが左の子である場合、注目ノードを「親」に更新し、ステップ3に戻る。右の子である場合は、注目ノードを「親」に更新し、次のステップに進む。
6. 注目ノードのみを経路から除外し、ステップ5に戻る。

## 4.3 実行結果

### 4.3.1 1日分の経路

以下に、始点を0、終点を79とした時の実行結果を示す。最短経路問題として求めた際には、始点s=1、終点t=80としていたが、プログラムではその性質上、番号が1だけずれることになる。

ノードの番号を表示することで、どのノードからどのノードへの経路なのかを示している。時刻昇順でデータを並べてあるため、ノード番号も昇順になっている。

初めに表示される経路は深さ優先探索によってすべて左の子(待ちリンク)に進んだ結果であるので、総距離が0の経路となる。

0 → 1 → 3 → 4 → 7 → 17 → 19 → 21 → 24 → 25 → 27 → 33 → 40 → 43 → 44 → 49 → 56 → 57 → 58 → 60 →

63 → 75 → 76 → 77 → 78 → 79

このような表示ですべての経路を求めることができた。今回のケースでは、計算時間は1秒以下であり、経路は198通りであった。

この表示では、ノード番号のみを表示しているため、具体的に何時の、どの空港からどの空港への便に搭乗するかがわかりにくく、実際に現場で使用するためには不便が多い。そこで、Excelに「空港名」、「時刻」、「便名」、「着ノードか発ノードか」、「経路の総距離」を以下のような形式で書き込むことで、不便さをある程度解消した(図7)。

67個目の経路									
空港名	小牧	福岡	福岡	福岡	福岡	福岡	福岡	福岡	福岡
時刻	540	620	650	660	690	780	990		
便ナンバー	303	303	202	502	304	305	310		
着or発	1	0	1	0	1	0	1		
距離(累積)	0	622.64	622.64	622.64	622.64	622.64	622.64		

68個目の経路						
空港名	小牧	青森	青森	青森	青森	小牧
時刻	655	735	765	1175	1205	1290
便ナンバー	363	363	364	367	368	368
着or発	1	0	1	0	1	0
距離(累積)	0	692.94	692.94	692.94	692.94	10692.94

図7 Excelでの表示(一部)

また、第3章でも用いた制約及び条件をプログラムにも適用させる。実行した結果は表5のようになった。

表5 1日分の経路に対する実行結果

基点となる空港	小牧	静岡
実行時間(秒)	0.001	0.001
制約なしの場合の経路数	198	21
全ての制約を考慮した経路数	16	8

以上より、1日分のパターン数は2つの基点を合わせて24通りとなり、第3章の数理計画問題として解いた場合と同じパターン数になることが確認できた。

### 4.3.2 2日分の経路

実際は、1日で基点となる空港に戻らないパターンが存在することも考えられる。例を挙げると、

- 1日め：小牧にて出勤し、福岡にて退勤し宿泊する。
- 2日め：福岡にて出勤し、小牧にて退勤する。

といったものがある。また、1日分のパターンでは使用されていない便や空港が存在していた。そこで、連続する2日間に規模を拡張し、更にスケジュールパターンを求めていく。

2日分の時空間ネットワーク(図8)の規模を表6に示す。

次に、以下の制約を追加する。

1. 1日のフライト回数は2回以上である。
2. 1日めの退勤から2日めの出勤までに、10時間以上18時間以下の休養を与える。



図 8 時空間ネットワーク (2 日分)

表 6 時空間ネットワークの規模 (2 日分)

ノード数	160
リンク数	230
フライトリンク数	80
待ちリンク数	150

1 の制約 : 1 日のフライト回数が 0 回 (ある空港に滞在したまま), 1 回 (ある空港から他の空港への片道のみ) というパターンは, 実際に使用されていないため除く.

2 の制約 : クルーの状態を健全に保つ上で重要な制約である. あまりに長い休養時間をとるパターンを除くため, 今回は上限を 18 時間とした.

以上の制約を追加し, 実際に解いた結果は以下の表 7 の通りである.

表 7 連続する 2 日間の経路に対する実行結果

基点となる空港	小牧	静岡
実行時間 (秒)	0.047	0.016
制約なしの場合の経路数	46,211	2,735
全ての制約を考慮した経路数	108	61

連続する 2 日間のパターンは, 小牧基点と静岡基点の両方を合わせて 169 通りとなった. また, 1 日分のスケジュールパターンでは使用されていなかった便や空港も全て使用されていることが確認できた. より現実的な解が得られたと考えられる.

## 5 システムの比較

第 3 章で述べた数理計画モデルによる解法と第 4 章で述べたネットワーク算法による解法の比較を行う.

### 5.1 実行時間について

数理計画モデルは 8 分以下, ネットワーク算法は 1 秒以下であった. これより, 繰り返し実行を行う数理計画モデルよりもネットワーク算法の方が短時間でスケジュールパターンを求めることができるため, スケジュールパターンを列挙する時間の観点ではネットワーク算法が優位といえる.

### 5.2 データの可変性について

フライト回数条件や勤務時間条件などを変更する場合, 数理計画モデルでは, Excel シート上の指定されたセルの値を変えることで条件を変更できる. ネットワーク算法では, プログラム内の各変数の値を変更する必要がある.

よって数理計画モデルの方が視覚的に変更点がわかりやすいため, データの変更のしやすさという点では数理計画モデルが優位といえる.

## 6 おわりに

本研究では, 時空間ネットワーク用いてスケジュールパターンを自動的に列挙させることができた. これより, 目視と手作業で行われていたスケジュールパターン作成の時間短縮が可能となった.

本研究は小規模の地域航空会社を例に挙げたが, 他の航空会社や小規模な鉄道会社にも適用可能であると考えられる. より大きな規模に対応できるようにプログラムを改善し, 2 日分や 3 日分の組み合わせで 1 週間や 1 カ月のパターンを作成することができれば, スケジュールパターンを列挙するにあたって時空間ネットワークが有用な手段であるといえる.

## 参考文献

- [1] 今泉淳: 「鉄道乗務員スケジューリング問題に対する列生成法によるアプローチ」. 東洋大学経営学部『経営論集』69 号, pp. 133-149, 2007.
- [2] 今泉淳, 福村直登, 森戸晋: 「鉄道乗務員スケジューリング問題に対する解法とその評価」. 東洋大学経営学部『経営論集』71 号, pp. 125-139, 2008.
- [3] 田口東: 「首都圏列車ネットワークに対する時間依存通勤配分モデル」. 日本オペレーションズ・リサーチ学会和文論文誌, 48 巻, pp. 85-108, 2005.
- [4] 田島玲, 味園真司: 「不定期便に対応したパイロット乗務員スケジューリング」. 情報処理学会全国大会 第 55 回平成 9 年後期 (1), pp. 85-86, 1997.
- [5] FDA フジドリームエアラインズホームページ.  
<http://www.fujidream.co.jp/>
- [6] IT 情報ポータル「ITmedia」Home.  
<http://www.itmedia.co.jp/>