

# Artificial Bee Colony による時間割編成問題の解法

呉東衛<sup>†1</sup> アランニャ・クラウス<sup>†2</sup> 狩野均<sup>†2</sup>

概要：本論文では、Artificial Bee Colony (ABC) を大学の時間割編成問題に適用する手法を提案する。ABC は、近年、連続関数最適化手法として組合せ最適化問題や制約充足問題への応用が研究されている。本稿では、ABC を制約充足問題の一種である時間割編成問題に適用する方法を提案する。それを加えて、離散値最適化問題における ABC の改良として、局所探索における解の更新方法と scout bee での解の生成方法を改良した。評価実験では、2018 年度筑波大学情報学群情報科学類の時間割を対象として、本手法と PSO, GA, ES との比較を行った。その結果、本手法の違反点数において従来手法よりも高い性能を得ることを示す。

キーワード：Artificial Bee Colony, 群知能, 時間割編成問題

## 1. はじめに

Artificial Bee Colony (ABC) は、蜂の群れの採餌行動をモデルとして D. Karaboga らによって 2007 年に提案された [1,2]。ABC は連続関数最適化問題において優れた性能を示している [3]。近年では、ABC を組合せ最適化問題や制約充足問題に適用するための研究が行われている [4,5]。ABC の解の更新式を離散変数に適用した「離散 ABC」はグラフ色塗り問題に有効であることが分かっている [6]。

従来、進化計算手法や群知能を用いた時間割編成問題の解法が提案されている。International Timetabling Competition 2007 (ITC2007) [7] を対象問題として、Bolaji らは解の更新方法として交換と移動を用いた ABC を提案した [8]。また、Alzaqebah らは違反の多い科目を優先して更新する方法を用いた ABC を提案した [9]。しかし、これらの ABC の解の更新方法は集団個体間の相互作用が考慮されていないため、ABC の集団探索の特長を生かしきれていない。また、遺伝的アルゴリズム (GA) [10] や Particle Swarm Optimization (PSO) [11] を用いた実用的な時間割を編成する手法も提案されていたが、性能は十分ではない。

本論文では、離散化 ABC を時間割編成問題に適用する方法を提案する。この方法は、離散値最適化問題に対する ABC の改良として、局所探索における解の更新方法と scout bee での解の生成方法を改良したものである。提案手法は従来手法よりも高い性能が得られることを示す。

以下では、まず時間割編成問題と ABC の概要、関連研究について述べる。次に提案手法のアルゴリズムについて説明し、最後に 2018 年度筑波大学情報学群情報科学類の時間割 [12] を対象として評価実験を行い、従来手法である PSO, GA, ES (進化戦略) [11] との比較実験を行い、本手法の有用性を評価する。

## 2. 研究分野の概要

### 2.1 ABC の概要

ABC [1] は蜂の群れの採餌行動をモデルとした探索手法である。ABC は三種類の蜂 employed bee, onlooker bee, scout bee が役割分担することにより、効率的に探索を行う。ABC のそれぞれの蜂の役割は employed bee, onlooker bee が局所探索, scout bee が大域探索を行うことである。

#### (1) ABC のアルゴリズム

Step1. ABC の解候補をランダムに初期化する。

Step2. employed bee と onlooker bee により局所探索を行う。始めに更新する解候補  $i$  と変更利用する解候補  $k$  を選択する。次にランダムに 1 要素を選択し、解の更新式 (式(1)) を適用し、解候補の更新を行う。解候補の更新後、解候補の適応度が低下した場合、解候補の更新を取り消す。

$$x'_{ij} = x_{ij} + \varphi(x_{ij} - x_{kj}) \quad (1)$$

$\varphi$  : [-1,1] の一様乱数

$k$  : 利用される解の番号

$i$  : 更新する解候補の番号

$j$  : 解候補における要素の番号

Step3. scout bee が  $limit$  回数連続で更新されなかった解候補をランダムに初期化する。

Step4. 終了条件を満たしたら、探索を終了する。満たさない場合は Step2 に戻る。

#### (2) 離散 ABC

制約充足問題 (CSP) は離散値問題であるため、連続値問題を最適化する手法としての ABC をそのまま適用することはできない。富樫らは ABC の更新式を離散化したことで CSP に適用できるようにした [6]。解の更新式は、次のように行っている。更新する解  $x_i$  と更新に利用する解  $x_k$  を選択し、ランダムに選んだ複数の要素に対して、 $x_i$  を  $x_k$  と同じ変数の値にする。

<sup>†1</sup> 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻  
Department of Computer Science, Graduate School of Systems and  
Information Engineering, University of Tsukuba

<sup>†2</sup> 筑波大学システム情報系情報工学科  
Division of Information Engineering, Faculty of Engineering, Information and  
Systems, University of Tsukuba

## 2.2 時間割編成問題

時間割編成問題とは、開講される全科目に対して、実施される時間帯（コマ）を割り当てる問題である[10]。制約条件をハード制約とソフト制約に分類し、各ソフト制約の重要度に応じて違反点数を設定する。本研究では、2018年度の筑波大学情報科学類の時間割を対象問題として、全ハード制約を満たしつつ、各ソフト制約の違反点数の合計が最小となるように時間割を編成するものとする。

### (1) 時間割編成問題の特徴

時間割編成問題には以下の特徴がある。

- ・探索空間が大きく、厳密解法で実用的な時間内で解くことは難しい。
- ・様々な制約があり、それらの制約をすべての満たすことは難しい。

### (2) 時間割編成問題の構成要素

時間割編成問題は、科目の集合、コマの集合、制約の集合から構成される。

### (3) 時間割編成問題の用語

本問題に関する用語を表1に示す。

### (4) 科目の集合

科目の集合は、科目名とその属性からなる。科目の属性とは、（グループ、コード、学年、必要コマ数、必修、専攻、必要学期数、講義、実験、コンピュータールーム、教員）の組のことである。ある科目の必要コマ数が  $n$  であるとき、登録する科目は  $n$  科目となる。

### (5) コマの集合

コマの集合は、（学年、学期、曜日、時限）の組からなる。学年については、科目ごとに指定された「学年」で決定されるので、科目に「コマ」を割り当てることは（学期、曜日、時限）の組を割り当てることになる。従って、「コマ」の集合は（学期、曜日、時限）の組として定義される。また、筑波大学は春 AB、春 C、秋 AB、秋 C、4 学期制である。本研究では、「科目」に「コマ」を割り当てることで時間割の編成を行う。

表1 時間割編成問題の用語

用語	用語の説明
グループ	関連科目をまとめたもの
学期連続科目	複数の学期に渡って開講される科目
時限連続科目	同学期に開講される2コマ続きの科目
主専攻科目	各主専攻で開講されている科目
全学科科目	すべての学部を対象として開講されている科目
情報科学類科目	情報科学類のみ開講されている科目
固定時間割	あらかじめ定められている時間割
科目の裏番組	同学年で同コマに実施する科目どうし

### (6) 制約の集合

制約をハード制約とソフト制約に分類する。

#### (a) ハード制約

編成上必ず満たさなければならない制約のことである。本研究で考慮するハード制約を表2に示す。

#### (b) ソフト制約

実用上満たすことが望ましい制約のことである。各制約において重要度が異なるため、それぞれの制約の重要度に応じて違反点数を設定する。ソフト制約と違反点数を表3に示す。本研究ではソフト制約の内容とそれぞれの違反点数の値は、学生アンケートにより設定した。また、S2の適用範囲は一年生、二年生の春 AB、秋 AB 学期となる。S7の適用範囲は一年生、二年生全学期となる。他のソフト制約は全学年に適用する。

表2 ハード制約

番号	制約内容
H1	関連科目において、履修順番の通りに科目を割り当てなければならない。
H2	1年次・2年次・同主専攻の時間割において、科目の裏番組を作ってはいけない。
H3	時限連続科目において、連続コマとなるように科目を割り当てなければならない。
H4	春 AB、秋 AB 学期に割り当てる科目を春 C、秋 C 学期に割り当ててはいけない。

表3 ソフト制約

番号	制約内容	違反点数
S1	学期連続科目の割り当てにおいて、同じ曜日の同じ時限となるように割り当てる。	10点
S2	特定の曜日に集中して科目を割り当てない。また、ある曜日のコマ数（授業の数）が2コマ以下であってはいけない。	5~15点
S3	一日の中で科目を連続的に割り当てる（空コマを作らない）。	10点
S4	週複数回開講されている科目を同じ曜日に割り当てない。	10点
S5	時限連続科目において、昼休みを挟んだコマの割り当てを避ける。	2点
S6	6時限目に科目を割り当てない。	1点
S7	ある学年の必修科目と別の学年の必修科目を同じ時間帯に割り当てない。	1点

### 2.3 関連研究

坂本らは筑波大学の時間割を対象に、知識に基づく GA を用いて編成することを提案した[10]. GA は選択・交叉・突然変異といった生物の進化過程をモデル化したアルゴリズムである[13]. GA の探索能力を向上させるために狩野らが提案しているウイルス進化論に基づく GA[14]を用いて性能向上と時間割編成担当者の意図の反映を実現している.

陳らは遷移確率を用いた PSO による 2010 年度筑波大学情報科学類の時間割を対象問題として編成することを提案した[11,15]. この手法は PSO の速度と位置の更新式を更新し, PSO に遷移確率を導入することで離散化 PSO を時間割編成問題に適した形式へと拡張した. しかし, この手法で用いている遷移確率を直接 ABC に適用することは難しい.

### 3. 提案手法

本手法では ABC の時間割編成問題への適用を行った. また, ABC の解の多様性と scout bee における大域探索の再現方法を考える上で, ABC の局所探索における解の更新方法と scout bee での解の生成方法を改良した.

#### 3.1 解のコード化

年・専攻ごとにユニットは「1 学年」・「2 年生」・3 年次の時間割においては「ソフトウェアサイエンス専攻」・「情報システム専攻」・「知能情報メディア専攻」の合計 5 つの時間割を編成するので, ユニットの 5 つ用意する. 各ユニットにおいて, 変数(アルゴリズムの探索個体の位置)は「科目」に対応し, 変数の値(探索個体の位置座標)は「コマ」に対応する. 本研究で対象とする時間割編成問題の科目数は 75, コマ数は 170, 各 5 ユニットの編成した時間割数は 20 となる. 解のコード化を図 1 に示す.

#### 3.2 合計違反点数の計算

探索個体がソフト制約を満たすかどうかをチェックし, 違反した場合は制約の違反点数を加算する. これを合計違反点数(すなわち適応度)とする.

#### 3.3 時間割編成問題における ABC の解の更新式の離散化

富樫らの離散化 ABC を基に, 時間割編成問題における ABC の解の更新式の離散化方法を提案する. 以下に, ABC の解の更新方法 1 を示す.

Step1 更新する解  $x_i$  と更新に利用する解  $x_k$  を選択する.

Step2 ランダムに選んだ複数(定数  $c$  個)の科目要素に対して,  $x_i$  を  $x_k$  と同じ変数の値(学期, 曜日, 時間)にする.

#### 3.5 ABC の改良

##### (1) scout bee の挙動の改良

ABC は時間割編成問題の解法としては, 局所探索の性能が優れている. しかし, 大域探索の挙動は十分ではない. 大域探索の挙動を改良するために, 今までの探索の最良解(gbest)を導入し, scout bee における

### 3.4 離散 ABC のアルゴリズム (提案手法 1)

```

1 全ハード制約を満たす初期集団を生成する
2 while (終了条件)
3     各探索個体の違反点数を計算する
4     for (すべての解候補) //employed bee
5         集団からランダムに一つの解を選択する
6         解の更新方法 1 による新たな解候補の探索
7         if (更新後の解が更新前の解と同等以上)
8             解候補を更新
9         if (更新後の解が更新前の解と同等以下)
10            limit += 1
11            探索回数 += 1
12    end for
13    for (すべての解候補) // onlooker bee
14        違反点数の低い順にルーレットで更新する解
15        を選択
16        解の更新方法 1 による新たな解候補の探索
17        if (更新後の解が更新前の解と同等以上)
18            解候補を更新
19        if (更新後の解が更新前の解と同等以下)
20            limit += 1
21            探索回数 += 1
22    for (scout bee サイズ) { //scout bee
23        limit を連続して更新に失敗した解を
24        ランダムな解に置き換える
25    end while

```

図 2 提案手法 1 の擬似コード

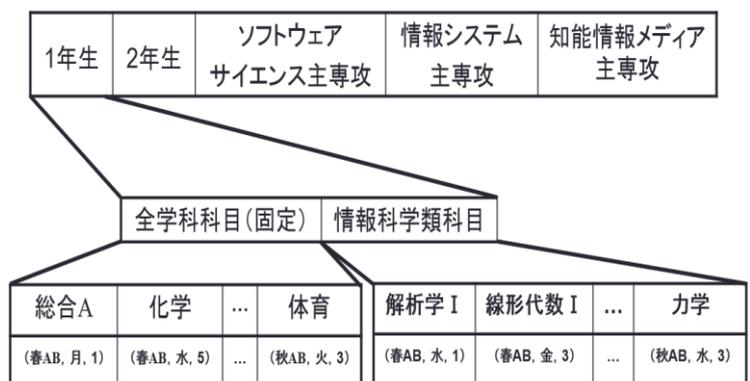


図1 解のコード化

新たな解の生成方法を提案する.

##### (2) 解の更新方法の改良

提案手法 1 の解の更新方法は局所探索に優れていることを示した[6]. しかし, 上記 (1) の改良を踏まて, 解の多様性を高めるため, ランダムに更新する手法を追加する. 提案手法 2 の解の更新方法 2 を示す.

Step1 更新する解 $x_i$ と更新に利用する解 $x_k$ を選択する。  
 Step2 ランダムに選んだ複数(定数 $c$ 個)の科目要素に対して、1/2の確率で以下の更新方法を選択する。

### 3.6 改良 ABC のアルゴリズム (提案手法 2)

```

1 全ハード制約を満たす初期集団を生成する
2 while (終了条件)
3     各探索個体の違反点数を計算する
4     for (すべての解候補) //employed bee
5         集団からランダムに一つの解を選択する
6         解の更新方法 2 による新たな解候補の探索
7         if (更新後の解が更新前の解と同等以上)
8             解候補を更新
9         if (更新後の解が更新前の解と同等以下)
10            limit += 1
11            探索回数 += 1
12        end for
13        for (すべての解候補) // onlooker bee
14            違反点数の低い順にルーレットで更新する解
                を選択
15            解の更新方法 2 による新たな解候補の探索
16            if (更新後の解が更新前の解と同等以上)
17                解候補を更新
18            if (更新後の解が更新前の解と同等以下)
19                limit += 1
20                探索回数 += 1
21        for (scout bee サイズ) { //scout bee
22            limit を連続して更新に失敗した解を
                以下生成方法による新たな解に置き換える
23            元の解を gbest に置き換えてから、 $\alpha$  の確率で科
                目がランダムに設定される
24        end while
    
```

図 3 提案手法 2 の擬似コード

表 4 実験条件

	PSO	GA	ES	提案手法1	提案手法2
評価回数	5×10 <sup>5</sup>				
個体数	25	80	50	50	50
limit	-	-	-	4000	400
$\alpha$	-	-	-	-	1/3
$c$	-	-	-	3	2
$\mu: \lambda$	-	-	1:1	-	-
突然変異率	-	0.3	-	-	-
elite 数	-	10	-	-	-
$w$	0.03	-	-	-	-
$c_1$	4	-	-	-	-
$c_2$	2	-	-	-	-

- $x_i$ を $x_k$ と同じ変数の値(学期, 曜日, 時限)にする.
- $x_i$ を科目の移動できる学期の中でランダムに移動する.

## 4. 実験

### 4.1 実験方法

2018 年度筑波大学情報学群情報科学類の時間割を対象として、PSO[11], GA, ES[11], 提案手法 1, 提案手法 2 の比較実験を行った. GA の選択はルーレット+エリート保存, 交叉は OX とした. ES は通常の  $(\mu + \lambda)$  ES である. 以下に示す実験結果は, 乱数のシードを変えて 30 回実験を行った時の平均値である. また, 実験条件を表 4 に示す. これらの値は予備実験により設定したものである.

### 4.2 実験結果

各手法で編成した時間割の合計違反点数を表 5, 各手法の 30 回実験結果の最良解(最終世代における集団中の最小値)の平均違反点数の推移グラフを図 4 に示す. また, 統計的な検定結果を表 6, ソフト制約の違反状況を表 7 に示す.

表 5, 表 6 について, 得られた結果の有意検定を行った. ここでは, Shapiro-Wilk normality test (95%信頼度 P-value: 0.05) を用いて実験結果(違反点数)の分布が正規分布であるかどうかを検定した. なお, 正規分布を仮定しないデータには, Exact Wilcoxon signed rank test (95%信頼度 P-value: 0.05) を用いて, 従来手法と提案手法の実験結果の間に有意差があるかどうかを検定した.

#### (1) 最良解の合計違反点数の比較

表 5 と表 6 から以下のことがわかる.

- 提案手法 1 と提案手法 2 は PSO, GA, ES より最良解の合計違反点数の最小値と平均値が優れている.
- 提案手法 2 は提案手法 1 と比較して最良解の合計違反点数の平均値が優れている.
- 表 6 より, 提案手法 1 と従来手法の有意差が確認された. また, 提案手法 2 は提案手法 1 より統計的に良い性能を示した.

#### (2) ソフト制約の違反状況の比較

表 7 から, 違反点数が高いソフト制約 (S1, S2, S4) は, すべての手法においてほぼ解消されている. 提案手法 1, 2 と PSO, GA, ES を比較すると, 特に S5, S7 に対して違反点数が大きく改善された.

#### (3) 最良値の進化過程

図 4 は, 違反点数と評価回数の関係を示している. 各手法とも 20 万回の探索でほぼ収束してことがわかる.

表5 最良解の合計違反点数

	PSO	GA	ES	提案手法1	提案手法2
平均値	24.1	29.9	25.8	19.0	17.4
標準偏差	4.7	6.7	3.4	0.9	0.6
最小値	19	20	20	17	17

表6 統計的な検定結果

	Exact Wilcoxon signed rank test: P-value
PSO-提案手法1	$7.4 \times 10^{-7}$
GA-提案手法1	$3.7 \times 10^{-9}$
ES-提案手法1	$1.9 \times 10^{-9}$
提案手法1-提案手法2	$1.6 \times 10^{-7}$

表7 ソフト制約の違反状況(30回の平均値)

番号	PSO	GA	ES	提案手法1	提案手法2
S1	0.0	0.1	0.0	0.0	0.0
S2	0.0	0.0	0.0	0.0	0.0
S3	1.1	1.1	1.1	1.0	1.0
S4	0.0	0.0	0.0	0.0	0.0
S5	0.9	2.1	1.8	0.03	0.0
S6	4.0	4.5	4.5	4.4	4.6
S7	7.6	8.8	7.0	4.6	2.8

### 4.3 考察

表5から、提案手法1はPSO, GA, ESに比べ、合計違反点数の平均値、標準偏差、最小値が小さいことがわかった。この理由は、提案手法1ではemployed beeとonlooker beeによる局所探索のときに解同士の相互作用があるため、探索が効率的に行われたものと思われる。

また、提案手法2が提案手法1に比べて合計違反点数の平均値が向上した理由として、次のことが考えられる。

- employed bee と onlooker bee の探索において、解の更新にランダム性が加味されているので、多様性が向上した。
- scout bee の探索において、gbest (集団中の最良解) の部分解を利用しているため、収束性が高まった。

次に表7から、未解決の制約として、特にS6, S7の違反点数が高いといえる。この理由は次のように考える。

- S6, S7は違反点数(表3)が小さいので、制約違反を解決する優先順位が低く、探索の最後まで残った。
- S6が全科目を対象としているのに対して、S7は必修科目を対象としているので、S6はS7より解決が難しく、探索後の違反点数が最も大きくなってしまった。

これらの制約違反を解決するためには、提案手法と山登り法をハイブリット化して局所探索性能を高めることや、対話型システムを構築して探索終了後の時間割を手動で微調整する方法が考えられる。

### 5. おわりに

本稿では、2018年度筑波大学情報学群情報科学類の時間割を対象としてABCを適用する方法を提案した。また、ABCの改良として、局所探索における解の更新方法と大域探索のためのscout beeにおける解の生成方法を改良した。PSO, GA, ESとの比較実験を行い、提案手法が比較手法より良い性能を示すことを確認した。

ソフト制約条件の違反点数を変更したときの各手法の性能について評価を行う。また、対話型システムの開発を行う予定である。

### 参考文献

- [1] D. Karaboga, B. basturk: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Jornal of Global Optimization*, Vol. 39, No. 3, pp.459-471(2007).
- [2] 伊庭齊志: 進化論的計算手法, オーム社, (2005).
- [3] Karaboga, Dervis, and Bahriye Basturk. "On the performance of artificial bee colony (ABC) algorithm." *Applied soft computing* 8.1 (2008): 687-697.
- [4] Karaboga, Dervis, and Bahriye Akay. "A modified artificial bee colony (ABC) algorithm for constrained optimization problems." *Applied soft computing* 11.3 (2011): 3021-3031.

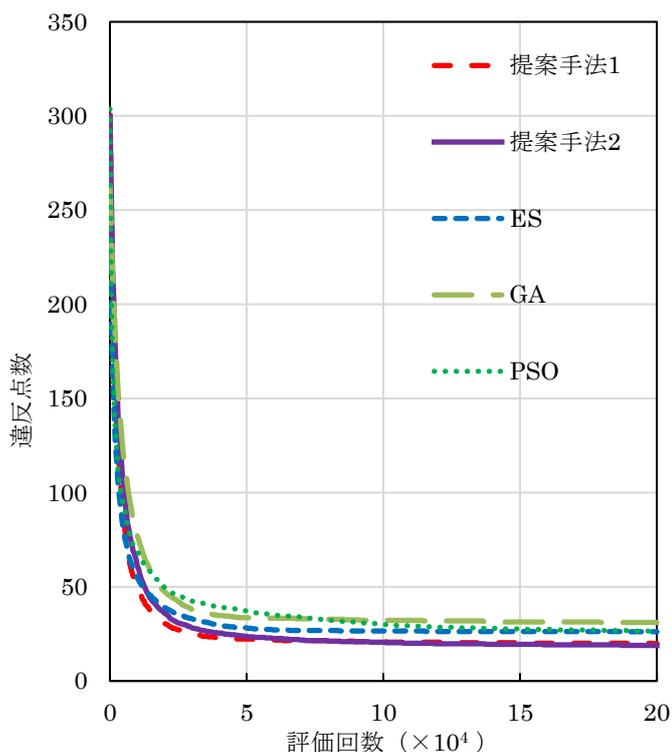


図4 違反点数の推移

- [5] Brajevic, Ivona, and Milan Tuba. "An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems." *Journal of Intelligent Manufacturing* 24.4 (2013): 729-740.
- [6] 富樫 勇哉, アランニャ クラウス, 狩野 均: Artificial Bee Colony 法を用いたグラフ色塗り問題の解法, 報処理学会 第79回全国大会 1R-05 (2017).
- [7] Di Gaspero, Luca, Barry McCollum, and Andrea Schaerf. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0, Queen's University, Belfast, United Kingdom, 2007.
- [8] Bolaji, Asaju La'aro, et al. "An improved artificial bee colony for course timetabling." *Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2011 Sixth International Conference on. IEEE, 2011.
- [9] Alzaqebah, Malek, and Salwani Abdullah. "Artificial bee colony search algorithm for examination timetabling problems." *International Journal of Physical Sciences* 6.17 (2011): 4264-4272.
- [10] H. Kanoh, Y. Sakamoto, "Knowledge-Based Genetic Algorithm for University Course Timetabling Problems", *International Journal of Knowledge-Based and Intelligent Engineering Systems*, Vol.12, No.4, pp. 283-294 (2008).
- [11] Kanoh, Hitoshi, and Satoshi Chen. "Particle swarm optimization with transition probability for timetabling problems." *International Conference on Adaptive and Natural Computing Algorithms*. Springer, Berlin, Heidelberg, 2013.
- [12] <http://www.tsukuba.ac.jp/education/ug-courses/>.
- [13] 北野 宏明: 遺伝的アルゴリズム, 産業図書, 産業図書(1993)
- [14] 長谷川和代, 松本美幸, 狩野均, & 西原清一. (1996). ウイルス進化論に基づく制約充足問題の解法. 全国大会講演論文集, (人工知能と認知科学), 139-140.
- [15] 陳聡志, 狩野均: 遷移確率を用いた PSO による時間割編成問題の解法, 計測自動制御学会 第39回知能システムシンポジウム, pp. 247-252(2012).

### 付録：提案手法で編成した1年生の時間割の例

1年生の春AB学期の時間割

	月	火	水	木	金
1	総合科目II	線形代数I		コンピュータテラシ	
2	総合科目II	線形代数I		コンピュータテラシ	
3	初修外国語	体育	解析学I	初修外国語	論理回路
4	英語	離散構造	解析学I	英語	論理回路
5	コンピュータテラシ	離散構造	化学A		英語
6					

1年生の春C学期の時間割

	月	火	水	木	金
1	総合科目II	情報科学概論I	情報科学概論I	情報科学基礎実験	
2	総合科目II	情報科学概論I	情報科学概論I	情報科学基礎実験	
3	初修外国語	プログラミング入門A	情報科学基礎実験		
4	英語	プログラミング入門A	情報科学基礎実験		
5	情報科学概論I	プログラミング入門A	化学A		英語
6	情報科学概論I				

1年生の秋AB学期の時間割

	月	火	水	木	金
1	総合科目II	情報社会と法制度	線形代数II	プログラミング入門B	
2	総合科目II	情報社会と法制度	線形代数II	プログラミング入門B	
3	初修外国語	体育	力学	初修外国語	解析学II
4	英語		力学	英語	解析学II
5			化学B		英語
6					

1年生の秋C学期の時間割

	月	火	水	木	金
1	総合科目II			プログラミング入門B	
2	総合科目II			プログラミング入門B	
3	初修外国語	コンピュータ数学		初修外国語	
4	英語	コンピュータ数学		英語	
5			化学B		英語
6					