

ウイルス感染を用いた進化戦略による 学園祭の人員配置問題に対する解法

新見 雄亮[†] 狩野 均^{††}

本論文では、筑波大学学園祭の人員配置問題を例にあげ、ウイルス感染を用いた進化戦略による解法を紹介する。対象問題では、複数の学園祭実行委員に仕事を割り当てるのが問題となるが、この割当ては強い制約を必ず満たす範囲内で弱い制約をできる限り満たすことが重要となる。弱い制約を制約条件と部分解列挙型の制約に分類し、前者で仕事を多くの人に分散させ、後者で特定の人に仕事を集中させる。本論文は部分解列挙型の制約をウイルスとして定義し、進化戦略と組み合わせることによって従来手法と比べて高速に実用的なスケジュールが編成できることを示す。

Solution to a Personnel Timetabling Problem of Campus Festival Using an Evolution Strategy Adopting Viral Infection

YUSUKE SHINMI[†] and HITOSHI KANO^{††}

This paper discusses a solution to a personnel timetabling problem of campus festival of University of Tsukuba. This problem is a search problem to assign jobs to members of the executive committee of the festival so as to minimize the total penalty for constraint violation. The constraints treated are classified as general constraints or partial solutions. The former decentralizes jobs to many persons and the later centralizes jobs on the particular persons. The proposed method uses an evolution strategy adopting viral infection. The method aims to improve the rate of search by giving the direction to evolution using infection of partial solutions as viruses. Experiments prove that the present method is more effective than conventional techniques.

1. はじめに

スケジューリング問題はプランニングの段階でまとまった複数の仕事の日程計画を立てようとするときに必然的に起こる問題である¹⁾。現実的なスケジューリング問題はNP-困難な問題であり、大規模なスケジューリング問題を適当な時間内で解くための一般的な解法を開発することは非常に難しいといえる²⁾。

遺伝的アルゴリズム(GA: Genetic Algorithm)は解を個体と呼ばれるデータ構造で表し、2つの個体から新しい個体を生成する交叉と、個体の集団による多

点探索を特徴とする^{3),4)}。また問題に対する専門知識を十分持っていないでも比較的高い探索性能を得ることができる。しかしGAは問題ごとに、実行不可能解を回避し、かつ形質遺伝に優れたコード化・交叉を設計する必要があり、その設計が非常に難しいことが知られている⁵⁾。

GAと同じ進化的計算法(EC: Evolutionary Computation)の分野に属している進化戦略(ES: Evolution Strategy)は実数値最適化問題に対する手法として発達してきた⁶⁾。ESは1個体による突然変異を遺伝的演算として使い、確定的な選択を行うことが特徴である。GAとESは独立して発展してきたが、近年は相互に影響を与えながら発展している。その結果、EC内における各手法の垣根は徐々に意味のないものになっている。

GAやESなどのメタ・ヒューリスティクスはハイブリッド化が容易であるという利点がある⁷⁾。そのためウイルス感染と併用したGAが多数提案されている⁸⁾⁻¹⁰⁾。ウイルス感染は個体の遺伝情報の一部をウイルスとして扱い、それを個体に感染させる手法であ

[†] 筑波大学大学院理工学研究科

Graduate School of Sciences and Engineering, University of Tsukuba

^{††} 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

現在、ソニーデジタルネットワークアプリケーションズ株式会社
Presently with Sony Digital Network Applications Inc.

る．それによって探索に人間の意図を介在させ，探索に方向性を持たせることができる．

本論文では実例として，筑波大学学園祭の人員配置問題を取り上げ，ウイルス感染を用いた ES による解法を紹介する．ES は交叉を設計しない分，GA よりも設計が容易であると考えている．またウイルスを感染させることにより，個体を有用な形質が得られる方向に比較的大きく変化させることができる．その結果，進化の高速化を図るものである．

以下では，まず 2 章で研究分野の概要と従来手法の問題点，および提案手法の基本方針について述べる．3 章では提案手法の具体的なアルゴリズムを示す．4 章では実験により提案手法の有効性を確認する．

2. 研究分野の概要

2.1 対象問題

本論文では現実のスケジューリング問題として筑波大学学園祭実行委員会の人員配置問題を扱う．具体的には m 個の仕事を n 人に重複を許して割り当てる問題である．各人・各仕事は各々異なった属性を持っており，まったく同じ人・仕事は存在しない．そのため探索は人と仕事の組合せを決定することが問題となる．問題の規模はスタッフ数約 100，仕事数約 110，仕事の要求人数のべ 250，探索空間は約 10^{500} となる．

本論文では対象問題を制約充足問題 (CSP: Constraint Satisfaction Problem) として定式化する．CSP とは離散値をとるいくつかの変数に割り当てられる値の組合せのうち，与えられた制約をすべて満たす組合せを発見する探索問題である¹¹⁾．本論文では対象問題を以下のように定式化する．

$$X = person_1, \dots, person_n$$

$$D = job_1, \dots, job_m$$

$$C = c_1, \dots, c_p$$

$$W = w_1, \dots, w_p$$

ただし，各記号は次のとおりである．

X : 仕事を行う人

D : 処理されるべき仕事

C : 対象問題の制約

W : 各制約に対する違反点数

n : 人の数

m : 仕事の数

p : 弱い制約の数

対象問題において人は複数の仕事を受け持たなければならない．そのため $person_i$ は複数個の job をとりうる．

次に $person_i$, job_j の持つ属性を以下に示す．括弧

内はその変数のとりうる値を示している．

$person_i = \{person_name: \text{人名}$

$person_code: \text{人のコード}$

$grade: \text{学年 (1,2,3,4)}$

$person_sex: \text{性別 (man,woman)}$

$person_time(t): \text{作業可能時間 (1,0)}$

$job_j = \{job_name: \text{仕事名}$

$job_code: \text{仕事のコード}$

$numter: \text{要求人数}$

$job_sex: \text{要求する性別 (man,woman,NA)}$

$job_time(t): \text{拘束時間 (1,0)}$

$person_time(t)$ と $job_time(t)$ は一定時間を 1 コマとして処理する．本論文では 30 分を 1 コマとし，8:00 から 23:00 までの 30 コマに対して編成を行う．また， $person_i$ の $grade$ の値が 3 以上の人を OB とする．

現実のスケジューリング問題を解く場合，制約を強い制約と弱い制約に分けることが一般的である¹²⁾．強い制約とは違反することで実行不可能解を生成してしまう制約を指し，弱い制約とはできうる限り満たしたい制約を指す．以下に対象問題の強い制約を示す．

- 仕事から指定されている人は必ずその仕事を割り当てられなければならない．
- 人に割り当てられている仕事の作業時間が重複してはならない．

対象問題を解く場合，上記の強い制約を必ず満たす範囲内で弱い制約をできうる限り満たすことが重要となる．本論文では，対象問題の弱い制約を通常の制約条件と部分解列挙型の制約に分けて考え，前者を制約 A，後者は制約 B とする．制約 A は人と仕事の組合せを制限する制約で，複数の人と仕事の組合せに違反が発生することが多い．表 1 にその例を示す．表の数字は違反点数を表している．また制約 B は望ましい人と仕事の組合せに許可を与える制約で，特定の人と特定の仕事に関する局所的な制約である．表 2 にその例を示す．制約 B は満たされた部分解の割合に対して違反点数を加点している．制約 A と制約 B の中には，互いに矛盾する制約がいくつかあり，すべての制約を満たす解は存在しない．

2.2 関連研究

現実のスケジューリング問題を解くための研究は多数行われている^{13),14)}．またスケジューリング問題に対するメタ・ヒューリスティックスの特徴を調査した研究も報告されている¹⁵⁾．進化的計算法の分野に限定してみると，遺伝的アルゴリズム (GA) による解法が多い^{16)~20)}．Tanev らは直接的コード化と間接的

表 1 制約 A (制約条件) の例
Table 1 Examples of constraint A.

制約条件		
c_1	都合の悪い時間帯に仕事を割り当てない	10
c_2	巡回時の各案内所の人員確保	9
c_3	1 年生にいろいろな仕事を体験してもらう	6

表 2 制約 B (部分解列挙型の制約) の例
Table 2 Examples of constraint B.

仕事	人
案内所	田中
ステージ運営	山田
きぐるみ	吉井

コード化について述べている¹⁸⁾。前者の染色体は表現型、もしくはそれに近いデータ構造で表され^{5),17)}、後者は仕事が処理される順番といった数値の列で表される^{16),18),20)-22)}。仙石らは間接的コード化の一種として接続優先順位を提案した²²⁾。これは仕事の優先順位を順序表現で表したコード化を採用しており、交叉・突然変異を行っても親個体の形質が子個体に引き継がれると述べている。

進化戦略 (ES) を用いた解法も提案されている。Homberger らは (μ, λ) -ES とタブーサーチを用いた配送車経路問題 (VRPTW) の解法を提案している⁷⁾。これは ES で配送車の数を最小化し、その後タブーサーチによって配送経路を最適化している。また Mester らは VRPTW に対して $(1+1)$ -ES を局所探索として用いている²³⁾。これは制約が複数の変数に関連して、制約違反している変数を一意に決められないため、山登り法¹¹⁾の代わりに用いている。

ウイルス進化論とは、進化は世代間における通常の遺伝 (垂直遺伝) だけではなく、同世代での遺伝子の交流 (水平遺伝) がウイルス感染によって起こると考える進化仮説である。それを工学的に応用し解の一部をウイルスと見なし個体に組み込むヒューリスティクスをウイルス感染⁹⁾と呼んでいる。久保田らの手法はウイルス集団を生成し、ウイルスに生命力を持たせるなどウイルスを生物としてとらえた手法である¹⁰⁾。一方、Kanoh らは部分解列挙型の制約充足問題に対するウイルス感染を提案している⁸⁾。部分解の候補をウイルスとし、突然変異の代わりにウイルス感染を起こすことで進化に方向性を持たせている。また Kanoh らは時間割編成問題に対するウイルス感染も提案している¹⁷⁾。これは過去の時間割からウイルスを生成するほかに、ユーザの希望もウイルスとして生成し、人間の知識と希望を時間割に反映している。Kanoh らの手法は、ウイルスを用いることで人間の知識を探索に

反映させ、結果的に探索の方向性を決定し、探索の効率化・精度の向上を図っているといえる。

2.3 従来手法とその問題点

GA を問題に適用する場合、問題ごとに形質遺伝に優れたコード化・交叉の設計が必要不可欠である⁵⁾。直接的コード化の場合は染色体に対して直接遺伝的演算を行える反面、実行不可能解を生成する危険性が高い。そのため、それを避けることのできる交叉の設計を行わなければならない。間接的コード化の場合は、染色体から表現型にデコードする際に実行不可能解の生成を避ける処理を行うことが基本となるため、交叉の設計は容易である。しかし間接的にしか染色体を操作できず、コード化が難しい。つまり GA において、2 つの親個体の交叉から、親個体の形質を受け継いだ子個体を生成するメカニズムの設計が必要不可欠であり、その設計の難しさが GA の応用範囲を狭めていると考えられる。

ES は突然変異によって生成された子個体から適度の高い個体を確定的に選択する手法である。突然変異はランダム探索そのものであり、個体内に局所的な変化を起こすものである。したがって、ES は初期集団として生成された個体が存在する探索空間から、個体の周辺をランダムに探索していることになる。その結果、探索に無駄が多く、収束が遅いという問題点がある。

2.4 本手法の基本戦略

本手法ではウイルス感染を用いた ES によって 2.3 節で述べた問題点を解決する。

- 直接的コード化を用いた ES を用いることでコード化・交叉の設計という負担を軽減する。
- ウイルスは複数個感染させることを基本とし、それにより個体を有用な方向に大きく変化させる。その結果、収束性の向上を図る。

本手法は直接的コード化を用いるため、遺伝子を手直接操作できる。そのため突然変異は個体内の形質が局所的に変化するように設計することで、親個体の大部分の形質を保存したまま子個体を生成することができる。この設計は個体内での変化を定義するものなので、交叉の設計に比べて設計者の負担は軽減されていると考える。

大規模な問題では大域的探索の成否が重要であるが、突然変異は個体に局所的な変化を起こすものであるため、探索は現在の探索点の周辺に限定されてしまう。本手法は部分解列挙型の制約をウイルスとして複数個感染させることで、個体を有用な形質を得られる方向に比較的大きく変化させることができる。

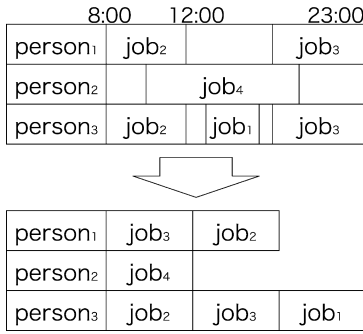


図 1 コード化の例 (上：表現型，下：遺伝子型)
Fig.1 An example of coding.

3. 提案手法

3.1 コード化

対象問題は各人に対する仕事の割り当てを決定する問題である。各仕事はそれぞれ異なった属性を持っているため、特定の属性を持つ人との組合せに強い依存関係が存在する。また部分解列挙型の制約も人と仕事の組合せである。したがって、人と仕事を持つ属性の繋がりを形質と考え、これが両親から子へと継承することを考えるのが妥当であると考えた。

以上をふまえ、本手法のコード化はスケジュールから時間の情報を削除して、各人に仕事を指定する方法を採用する。染色体は表現型に近い形であるため、直接スケジュールを操作できる。また、余計な情報を削ぎ落としたため探索空間を縮小している。コード化の例を図 1 に示す。なお本手法のコード化では、染色体は人に仕事が割り当てられているという情報のみを保持しているため、仕事の並び方は問題とならない。

3.2 個体の評価方法

3.2.1 制約 A (制約条件) の違反点数

制約 A は複数の人と仕事を持つ属性の組合せに依存しており、満たすことで仕事を人に分散させる役割を果たすことが多い。制約 A はその重要度に応じて違反点数が設定されており、重要な制約条件ほど大きい点数が与えられている。制約 A は人に加点するものと仕事に加点するものの 2 種類が存在し、その合計値を制約 A の違反点数とする。

$$W_{violation} = \sum_{i=1}^n W_i + \sum_{j=1}^m W_j \quad (1)$$

$$W_i = \sum_{k=1}^l conf(i, c_k) \quad (2)$$

表 3 主な制約 A
Table 3 Constraint A.

c_k	制約条件	対象	w_k
人に違反点数を加点する制約			
c_1	不都合な時間に仕事を割り当てない	全員	10
c_2	巡回時の各案内所の人員確保	全員	7
c_3	OB はあまり仕事を割り当てない	OB	3
仕事に違反点数を加点する制約			
c_{12}	仕事からの性別の希望に応える	全員	9
c_{17}	最終日の撤収は OB を割り当てる	案内所	10
c_{19}	夜は男性を割り当てる	交通規制	9
c_{24}	2 日連続で割り当ててはならない	徹夜警備	7

$$W_j = \sum_{k=l+1}^{p-1} conf(j, c_k) \quad (3)$$

ただし、各記号は次のとおりである。

$W_{violation}$: 制約 A の違反点数

W_i : $person_i$ の違反点数の合計値

W_j : job_j の違反点数の合計値

l : 「人に違反点数を加算する制約条件」の数

主な制約 A を表 3 に示し、代表的な制約 A の計算方法を説明する。

C_1 : 不都合な時間に仕事を割り当てない

すべての t に対して、 $person_time_i(t) = 0$ 、かつその割り当てられているすべての $job_time_j(t) = 1$ の場合に加点する。

C_9 : OB にあまり仕事を割り当てない

$person_grade_i$ が 3 の場合、4 つ以上の job_j が割り当てられていたら加点、4 の場合は 3 つ以上の job_j で加点する。どちらの場合もオーバした job_j の数に応じて加点する。

C_{12} : 仕事からの性別の希望に応える

job_sex_j が「man」もしくは「woman」の場合、割り当て先の $person_sex_i$ が同値でなければ加点する。

3.2.2 制約 B (部分解列挙型の制約) の違反点数

対象問題の仕事の中には、処理するためにその仕事に対するスキルや経験が必要とされるものがある。そのため特定の仕事に対して、その仕事を経験したことのある人、経験を積ませたい人、もしくは信用できる人に処理させたいという要求が生じる。本手法ではその要求を満たすために制約 B を用いる。これは望ましい組合せに許可を与える局所制約であり、仕事と人の組合せで表す。すべての部分解のうち、満たされた部分解の割合に応じて違反点数を加点する。また本論文ではその割合を部分解含有率と呼ぶ。なお、本研究の制約は全部で p 個あり、 $p - 1$ 番目までは制約 A、 p 番目を制約 B としている。

$$V_p = (1 - V)w_p \quad (4)$$

ただし、各記号は次のとおりである。

W_p : 制約 B の違反点数

V : 部分解含有率

3.2.3 総合違反点数

制約 A と制約 B の違反点数の合計値を総合違反点数とする。この値を最小化することが本研究の目的となる。

$$W_{total} = W_{violation} + W_p \quad (5)$$

ただし、各記号は次のとおりである。

W_{total} : 総合違反点数

$W_{violation}$: 制約 A の違反点数

W_p : 制約 B の違反点数

3.3 提案手法のアルゴリズム

本手法の全体のアルゴリズムを示す。本手法は一般的に用いられている (μ, λ) -ES を基に設計している。

Step 1. 部分解列挙型の制約をウイルスとし、ウイルス集団を生成する。

Step 2. 初期個体集団をランダムに生成する。

Step 3. λ 個の子個体を生成するまで a~d の遺伝的演算を行う。

- a. ランダムに 1 個体選択する。
- b. ランダムに突然変異を行う。
- c. ランダムに複数個のウイルスを選択、感染させる。
- d. 個体の評価を行う。

Step 4. 違反点数の小さい上位 μ 個体を次世代集団とする。

Step 5. 一定世代の間 Step 3, Step 4 を繰り返す。

以下の節でアルゴリズムの詳細を示す。これらの処理はすべて強い制約 (2.1 節参照) を必ず満たす範囲内で行っている。

3.4 ウイルス集団生成

部分解列挙型の制約をウイルスとして用いる。ウイルスの属性を以下に示す。

$viruss = job_code, person_code$

job_code : 仕事のコード

$person_code$: 仕事に割り当てたい人のコード

3.5 突然変異

本手法では突然変異として仕事の移動・交換・人の交換の 3 つを設計した。これは人がスケジュールを修正する際に用いている演算をモデルとしており、遺伝的演算として効率が良いと判断した。また探索空間上のすべての解を探索することができる。以下にそれぞれの処理の詳細を記述し、その例をそれぞれ図 2, 図 3, 図 4 に示す。

person ₁	job ₃	job ₂	
person ₂	job ₄		
person ₃	job ₂	job ₃	job ₁

図 2 仕事の移動の例

Fig. 2 An example of moving job.

person ₁	job ₃	job ₂	
person ₂	job ₄		
person ₃	job ₂	job ₃	job ₁

図 3 仕事の交換の例

Fig. 3 An example of exchange of jobs.

person ₁	job ₃	job ₂	
person ₂	job ₄		
person ₃	job ₂	job ₃	job ₁

図 4 人の交換の例

Fig. 4 An example of exchange of persons.

3.5.1 仕事の移動

Step 1. ランダムに job_j を選択。

Step 2. job_j を保持する $person_j$ を発見。

Step 3. ランダムに $person_i$ を決定。

Step 4. 強い制約を満たせるかチェックする。

Step 5. 満たせる場合は job_j を $person_i$ に移動。満たせない場合は Step 3 に戻る。

3.5.2 仕事の交換

Step 1. ランダムに job_j, job_i を選択。

Step 2. それぞれを保持する $person_j, person_i$ の発見。

Step 3. 強い制約を満たせるかのチェックをする。

Step 4. 満たせる場合は job_j を $person_i$ に、 job_i を $person_j$ に割り当て。満たせない場合は Step 1 に戻る。

3.5.3 人の交換

Step 1. ランダムに $person_i, person_j$ を選択。

Step 2. 強い制約を満たせるかチェックする。

Step 3. 満たせる場合は 2 人が保持するすべての仕事を交換する。

3.6 ウイルス感染

本手法のウイルスは部分解列挙型の制約そのものであるため、感染させることで高い確率で違反点数を下げることができる。そのため毎世代突然変異で生成された子個体に対して一定確率でウイルス感染を行う。

表 5 実験条件
Table 5 Experiment conditions.

	IHC	GA1	GA2	GA3	ES	本手法
個体数	1	30	30	100	30	30
子個体数	1	26	26	90	150	150
交叉率		0.8	0.8	0.8		
突然変異率	0.8	0.0	0.1	0.1	0.8	0.8
ウイルス感染率						0.6
エリート数		4	4	10		
評価回数	375,066	375,054	375,054	375,040	375,030	375,030

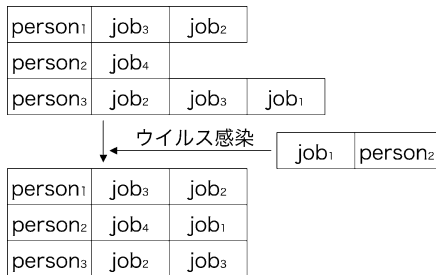


図 5 ウイルス感染の例

Fig. 5 An example of viral infection.

以下にその手順を示し、図 5 に例を示す。

- Step 1. 感染させるウイルスの数をランダムに決定。
- Step 2. ウイルス $virus_s$ 選択。
- Step 3. $virus_s$ が持つ job_j を取得。
- Step 4. job_j を保持する $person_j$ の発見。
- Step 5. $virus_s$ が持つ $person_i$ を取得。
- Step 6. 強い制約を満たせる場合は job_j を $person_i$ に割り当て、 $person_j$ から job_j を削除。
- Step 7. 感染するウイルス数に満たない場合は Step 2 へ。

4. 評価実験

4.1 評価方法

提案手法の有効性を確認するために評価実験を行った。対象データは 2003 年筑波大学学園祭「第 29 回 雙峰祭」初日、2 日目のデータを用いた。評価指標は総合違反点数を用いるが、より詳しい挙動を確認するために部分解含有率 (3.2.2 項参照) も示す。実験は 20 回試行し、実験結果は各試行における総合違反点数が最も小さい個体 (最良個体) に対して集計を行っている。表中の min は 20 回試行中で生成された最も良い個体の総合違反点数、ave は最良個体の総合違反点数の平均値、SD はその標準偏差を示している。ただし、部分解含有率の表中の max は 20 試行中の総合違反点数最良個体の部分解含有率を示している。

表 4 比較手法

Table 4 Comparison methods.

人手	スケジュール編成担当者とその経験者である OB が約 2 週間かけて編成し、学園祭で実際に使用したもの。
IHC	反復を行う山登り法。対象問題は複数の変数の組合せに制約違反が生じるため、一意に違反している変数を決定できない。そのためここでは (1+1)-ES を使用している。
GA	接続優先順位を順序表現で表したコード化と一点交叉を用いた遺伝的アルゴリズム。世代交代はエリート個体を保存し、親個体からルーレット戦略で 2 個体選択し子個体を生成する。一定回数繰り返し、生成された子個体とエリート個体の集団をそのまま次世代集団とする SimpleGA を用いる。
ES	一般的に用いられ性能が高いとされる (μ, λ) -ES。 μ 個の親個体からランダムに選択し、子個体を生成する。それを繰り返し、 λ 個の子個体から上位 μ 個を次世代集団とする。

4.2 実験条件

本手法の有効性を確認するため、人手による解、山登り法 (IHC: Iterated Hill-Climbing), GA, ES, 本手法のそれぞれの手法に対して実験を行った。表 4 に各手法の概要を、表 5 に実験条件を示す。

IHC と比較することで集団を有する多点探索の有効性を、GA と比較することで形質遺伝に優れたコード化・遺伝的演算を設計することの有効性を確認する。GA のコード化は接続優先順位を用いた²²⁾。GA1 と GA2, GA3 の違いは突然変異の有無であり、GA3 は個体数を 100 とした。GA の突然変異は本手法のものを用いた。これは順序表現に対して突然変異を行った場合、突然変異を行った遺伝子以降の遺伝子が大きく変化するため⁴⁾ 不適切と考えた。

また、ウイルス感染の有効性を確認するため ES と本手法 (ウイルス感染を行う ES) との比較を行う。

なお、本論文での突然変異率、交叉率、ウイルス感染率はすべて 1 個体に対してそれぞれの操作を行う確率を表すものである。

4.3 他手法との比較

ES と他手法との比較実験を行った。総合違反点数

表 6 実験結果 (総合違反点数)
Table 6 Experimental results (total constraint).

	学園祭初日			学園祭 2 日目		
	min	ave	SD	min	ave	SD
人手	1133		1194			
IHC	696	755	38	635	689	32
GA1	3745	3992	163	3617	3984	192
GA2	555	631	52	546	682	70
GA3	1086	1207	74	1118	1237	58
ES	357	383	14	302	341	19
本手法	347	367	12	278	303	22

表 7 実験結果 (部分解含有率)
Table 7 Experimental result (partial solution content rate).

	学園祭初日			学園祭 2 日目		
	max	ave	SD	max	ave	SD
人手	0.92		1.0			
IHC	0.69	0.63	0.04	0.73	0.68	0.04
GA1	0.11	0.06	0.03	0.15	0.08	0.03
GA2	0.78	0.72	0.04	0.8	0.7	0.07
GA3	0.5	0.41	0.04	0.55	0.44	0.04
ES	0.91	0.87	0.02	1.0	0.94	0.02
本手法	0.92	0.89	0.02	1.0	0.97	0.02

を表 6 に、部分解含有率を表 7 に示す。

表 6 から以下のことが分かる。ES が最も良い結果を出している。人手による解の違反点数と比べると、初日は約 800 点、2 日目は約 900 点の改善をみることができる。ES と IHC を比べると初日、2 日目ともに約 330 点改善している。これは ES が複数個体による多点探索であることと、1 世代で集団数よりも多くの子個体を生成し選択を行うことにより、より大域的な探索を行っているためであると考えられる。

GA1, GA2, GA3 を比較すると GA1 が最も違反点数が高い。これは進化の初期の段階で全個体の違反点数が同一になっていることから、集団が一様に収束してしまったと考えられる。また GA1 に突然変異を加えた GA2 は IHC, GA1 よりもよい性能を示している。GA 2 と GA3 を比較すると GA2 の方がよい結果を示している。評価回数を同一にしているため、集団サイズの大きい GA3 は収束する前に探索を終えているためであると考えられる。

次に表 7 から部分解含有率に対する挙動として、人手による解が最も含有率が高いことがわかる。これは実行委員会のスケジュール編成担当者がスケジュール編成を行う際、最初に制約 B で許可を与えられている人にその仕事を割り当てる、という編成方法のためであると考えられる。しかし人手による総合違反点数が 1194 という結果を考えると、制約 B を満たしたうえで、同時に制約 A を満たせていないと考えられる。制約 A は複数の人と仕事に関係して制約違反が発生

するため、人間にとって満たすことは非常に難しいためであると考えられる。一方、ES は総合違反点数では人手を大きく上回っているが、ウイルス感染率の平均値は約 6%低い。

4.4 ウイルス感染の有効性

ウイルス感染の有効性を確認するため、ES と本手法との比較を行った。その総合違反点数を表 6 に、部分解含有率を表 7 に示す。

表 6 から ES にウイルス感染を行うことで初日のデータで約 5%、2 日目データで約 15%の性能向上を見ることが出来る。また表 7 から部分解含有率も初日のデータで約 2%、2 日目で約 3%の性能向上が見られる。

図 6 と図 7 に世代に対する総合違反点数を 1000 世代まで示す。これらからウイルス感染を行うことで進化の速度が約 2 倍になっていることが分かる。特に探索初期の進化が速いことが分かる。

さらに詳しく見てみる。初日のデータでは、人手による解の違反点数を ES は 303 世代目、ES + ウイルス感染は 167 世代目で上回っている。2 日目では、人手による解の違反点数を ES は 280 世代目、ES + ウイルス感染は 128 世代目で上回っている。

4.5 人手による解との比較

本節では、人手による解と本手法で編成した解を比較する。違反点数は本手法のほうが優れているのに対して、部分解含有率は人手による解のほうが優れている。制約 A が仕事を分散させる役割を担うものが多

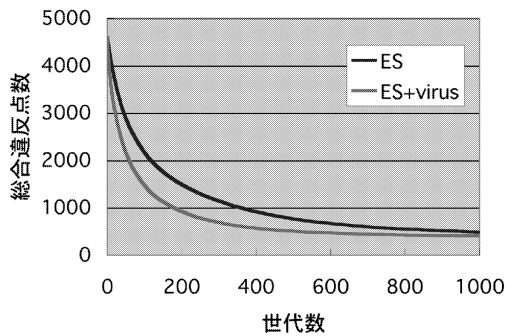


図 6 各世代における最小総合違反点数の推移 (学園祭初日)
Fig. 6 Minimum violation in each generation (first day).

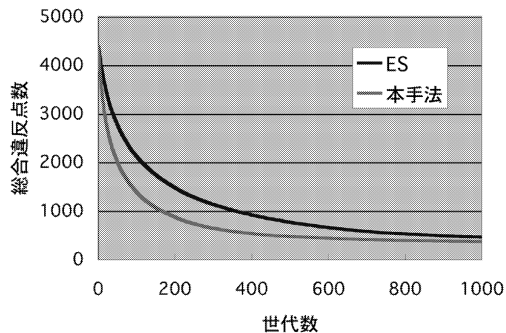


図 7 各世代における最小総合違反点数の推移 (学園祭 2 日目)
Fig. 7 Minimum violation in each generation (second day).

く、制約 B が人に仕事を集中する役割を担っていることを考えると、人手による解は仕事の分散がうまくできていないと考えられる。実際に人手による解と本手法で編成されたスケジュールを見比べてみると、人手による解では特定の人に仕事が集中する傾向があることを確認できた。具体的に述べると、人手による解は制約 A 「1 年生にはさまざまな仕事を体験してもらう」を満たされた 1 年生が少ない。それに対し、本手法で編成されたスケジュールは仕事の分散がうまくできており、特に 1 年生に種類の違う仕事が割り当てられている。提案手法は制約 A と制約 B をともに満たすスケジュール編成に成功しているといえる。

5. おわりに

本論文ではウイルス感染を用いた進化戦略を提案し、筑波大学学園祭実行委員会のスケジュールを有効に編成できることを示した。本手法は進化戦略を用いることでコード化・交叉の設計の負担を軽減し、遺伝子を局所的に操作することで形質遺伝に優れていると考えている。また複数個のウイルスを一度に感染させることで、有用な形質が得られる方向に比較的大きく変化

させることができ、探索の効率化が図れた。

本手法は、時間割編成問題や室内レイアウト問題などの CSP で定式化できる問題に対しても適用できる。今後の課題としては、本手法を用いて実用に耐えうるシステムを構築することがあげられる。その際にいくつか考慮すべき項目が残されているのも事実である。本手法は確率的探索手法であるため、解のいくつかの変数に制約違反が残ってしまう場合がある。また直前になっての仕事の作業時間、要求人数の変更などがある。しかし、前者の問題点は本手法に GUI を用いた問合せ機能を追加することで解決することができる。また、後者は仕事のデータを変更したうえで、編成済みのスケジュールを再編成することで解決できる。約 500 世代で進化が収束しはじめていることから、両者を組み合わせて、GUI を用いてインタラクティブに編成することで、実用化できると考えている。

謝辞 データを提供していただいた 2003 年筑波大学学園祭実行委員会、貴重なご意見をいただいた同委員会スケジュール編成担当、福田英臣氏、2001 年同担当、佐藤省吾氏に感謝の意を表します。

参考文献

- 鍋島：スケジューリング理論，森北出版 (1974).
- 宮下：スケジューリング問題へのアプローチ (1) 一人工知能における課題，人工知能学会誌，Vol.10, No.6, pp.880-887 (1995).
- Gen, M. and Cheng, R.: *Genetic Algorithms & Engineering Optimization*, WILEY-INTERSCIENCE (2000).
- 北野 (編)：遺伝的アルゴリズム，産業図書 (1993).
- 小野，小林：Inter-Machine JOX に基づく JSP の進化的解法，人工知能学会誌，Vol.13, No.5, pp.780-790 (1997).
- Fogel, D.B.: *Evolutionary Computation*, IEEE Press (1995).
- Homberger, J. and Gehring, H.: A two-phase hybrid metaheuristic for the vehicle routing problem with time windows, *European Journal of Operational Research*, Vol.162, pp.220-238 (2005).
- Kanoh, Matsumoto, Hasegawa, Kato and Nishihara: Solving Constraint Satisfaction Problem by a Genetic Algorithm Adopting Viral Infection, *Engng Applic. Artif. Intell.*, Vol.10, No.6, pp.531-537 (1997).
- 佐藤，狩野：ウイルス感染を用いた遺伝的アルゴリズムによるニューラルネットワークの学習，情報処理学会論文誌：数理モデル化と応用，Vol.44, No.SIG 7 (TOM 8), pp.82-90 (2003).

- 10) 久保田, 下島, 福田: ウイルス進化論に基づく遺伝的アルゴリズム—ウイルス感染による進化, 第8回自律分散システム・シンポジウム, pp.167–170 (1996).
- 11) 狩野: 制約充足問題の近似解法, 人工知能学会誌, Vol.12, No.3, pp.359–365 (1997).
- 12) Grobner, H., Wilke, P. and Buttcher, S.: A Standard Framework for Timetabling Problems, *Proc. Practice And Theory of Automated Timetabling 2002*, LNCS 2740, pp.24–38 (2003).
- 13) Causmaecker, P.D. and Berghe, G.V.: Relaxation of Coverage Constraints in Hospital Personnel Rostering, *Proc. Practice And Theory of Automated Timetabling 2002*, LNCS 2740, pp.129–147 (2003).
- 14) White, C.A. and White, G.M.: Scheduling Doctors for Clinical Training Unit Rounds Using Tabu Optimization, *Proc. Practice And Theory of Automated Timetabling 2002*, LNCS 2740, pp.120–128 (2003).
- 15) Rossi-Doria, O., et al.: A comparison of the Performance of Different Metaheuristics on the timetabling Problem, *Proc. Practice And Theory of Automated Timetabling 2002*, LNCS 2740, pp.329–351 (2003).
- 16) 杉本, 狩野: ウイルス進化論に基づく GA を用いた人員配置問題の解法—子孫の適応度に基づく選択戦略, 情報処理学会第 60 回全国大会 5V06 (2000).
- 17) Kanoh and Sakamoto: Interactive Timetabling System Using Knowledge-Based Genetic Algorithms, *Proc. International Conference on System, Man and Cybernetics (SMC 2004)*, pp.5852–5857 (2004).
- 18) Tanev, I.T., et al.: Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: Application service provider approach, *Allied Soft Computing*, Vol.5, pp.87–100 (2004).
- 19) Vergara, F.E., et al.: An evolutionary algorithm for optimizing material flow in supply chains, *Computers & Industrial Engineering*, Vol.43, pp.407–421 (2002).
- 20) Käschel, J., et al.: Real-time dynamic shop floor scheduling using Evolutionary Algorithms, *Int. J. Production Economics*, Vol.79, pp.113–120 (2002).
- 21) Wang, K., et al.: A genetic algorithm for online-scheduling of a multiproduct polymer batch plant, *Computers and Chemical Engineering*, Vol.24, pp.393–400 (2000).
- 22) 仙石, 吉原: GA によるヒューリスティック探索の最適化, 情報処理学会論文誌, Vol.37, No.10, pp.1811–1820 (1996).
- 23) Mester, D. and Bräysy, O.: Active guided evolution strategies for large-scale vehicle routing problems with time windows, *Computers & Operations Research*, Vol.32, pp.1593–1614 (2005).

(平成 16 年 11 月 22 日受付)

(平成 17 年 1 月 31 日再受付)

(平成 17 年 3 月 23 日採録)



新見 雄亮

1980 年生。2003 年筑波大学第三学群情報学類卒業。2005 年同大学院大学院理工学研究科修了。同年ソニーデジタルネットワークアプリケーションズ株式会社入社。在学中は進化戦略を中心とした進化的計算法に関する研究に従事。



狩野 均 (正会員)

1978 年筑波大学第一学群自然科学類卒業。1980 年同大学院大学院理工学研究科修了。同年日立電線(株)入社。同社オプトロシステム研究所において人工知能・ニューラルネットワークの応用に関する研究に従事。平成 5 年より筑波大学電子・情報工学系。現在, 同大学院助教授。知識システム・遺伝的アルゴリズム・人工生命の研究に従事。工学博士。1992 年電気学会論文賞, 1999 年 WSC4 論文賞, 2003 年情報処理学会優秀研究報告賞各受賞。人工知能学会, IEEE 等各会員。