

## ウイルス感染を用いた進化戦略によるスケジューリング問題の解法

新見 雄亮<sup>†</sup> 狩野 均<sup>‡</sup>

本論文では、スケジューリング問題に対してウイルス感染を用いた進化戦略による解法を提案する。近年、GAとESを実用問題へ応用する研究が行われている。GAは問題ごとに形質遺伝に優れたコード化・交叉方法を設計する必要があり、その設計は問題に強く依存しており非常に難しいことが知られている。ESは交叉を設計する必要がないものの、突然変異で探索を行うため効率が悪い。本研究はESに対して、有用な形質をウイルスとして定義し、それを一度に複数個感染させることで、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手法を提案する。現実存在するスケジューリング問題を対象に、従来手法と本手法を比較し、本手法の有効性を示す。

### Solution to scheduling problems using an evolution strategy adopting viral infection

YUSUKE SHINMI<sup>†</sup> and HITOSHI KANO<sup>‡</sup>

In this paper, we present a solution to real-world scheduling problems using evolution strategies (ESs) adopting viral infection. Crossover, in Genetic Algorithms (GAs), should preserve genes of good characteristics between parents and their children. It can be difficult, however, to design the appropriate procedure of crossover depending on individual problems. In ESs, the mutation is regarded as a main operation. Consequently, the rate of search in ESs is not too high. The present method aims to improve the rate of search by giving the direction to evolution using infection of partial solutions. Experiments using a person timetable problem 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1つ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。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。

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。また問題に対する専門知識を十分持っていないながらも比較的ロバストな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。このことがGAのロバスト性を狭める要因となっているといえる。

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。ESは1個体による突然変異を遺伝的演算として用い、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無いものになっている。

本論文では、現実のスケジューリング問題を対象にウイルス感染を用いたESを提案する。スケジューリング問題にESを適用させた研究は報告されている[1][2]。また、ウイルス感染も提案された手法である[3]。したがって本論文の主題は現実存在するスケジューリング問題へのウイルス感染を用いたESの適用、および評価となる。

#### 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学園祭当日のスケジュール編成問題を扱う。具体的には $n$ 個の仕事をも $m$ 人に重複を許して割り当てる問題である。各人・各仕事は各々異なる属性を持っており、全く同じ人・仕事は存在しない。そのため探索は人と仕事の組み合わせを決定す

<sup>†</sup>筑波大学大学院 理工学研究科

Graduate School of Sciences and Engineering, University of Tsukuba

<sup>‡</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.

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。

## 2.2. 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。コード化の方法として直接的コード化と間接的コード化がある。前者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。後者は染色体から表現型にデコード化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[4]。

ES は突然変異によって生成された子個体から適応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。従って 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。

## 2.3. 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 2.2 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。

## 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み合わ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み合わせである。従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。そこで、本

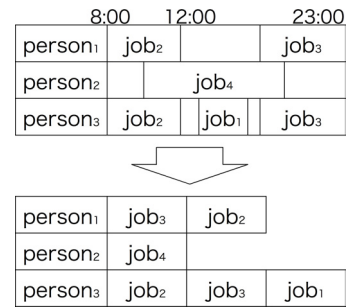


図 1 コード化の例

(上：表現型, 下：染色体)

表 1 制約条件 (制約 A) の例

| $\alpha_k$ | 制約条件                | $w_k$ |
|------------|---------------------|-------|
| $C_1$      | 都合の悪い時間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経験してもらう | 6     |

表 2 部分解列挙型の制約 (制約 B) の例

| 仕事     | 人  |
|--------|----|
| 案内所    | 田中 |
| ステージ運営 | 山田 |
| きぐるみ   | 吉井 |

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。

### 3.2. 個体の評価方法

本論文では、対象問題の制約を一般的な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 は人と仕事の組み合わせを制限する制約で、複数の人と仕事の組み合わ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。これは、制約の重要度に応じて設定され、重要な制約ほど大きな点数が設定されている。また制約 B は望ましい人と仕事の組み合わ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。また本論文では、その割合を部分解含有率

と呼ぶ。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。本研究の目的は総合違反点数を最小化することとなる。

### 3.3. アルゴリズム

本手法のアルゴリズムを図 2 に示す。本手法は一般的に用いられている  $(\mu, \lambda)$ -ES を基に設計している。以下の節ではアルゴリズムの詳細について述べる。

### 3.4. ウイルス集団生成

テキストファイルに記された仕事と人の組み合わせを読み込み、それをウイルスとする。ウイルスは部分解列挙型の制約としても用いる。ウイルスの染色体を図 3 に示す。

### 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。仕事の移動、交換、人の交換の例をそれぞれ図 4 図 5 図 6 に示す。

### 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。ウイルス感染はランダムにウイルスを複数個選択し、感染を行う。その例を図 7 に示す。

## 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部分解含有率も示す。部分解含有率とは部分解列挙方の制約を満たしている割合である。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）の平均値を示している。

### 4.2. 実験結果

本手法と他手法との比較実験を行った。比較手法として人手、反復山登り法、GA[5]、ES を用いた。その総合違反点数を表 3 に、部分解含有率を表 4 に示す。また、ES と本手法における 0, 100, 200, 400, 600 世代における総合違反点数を表 5 に示す。

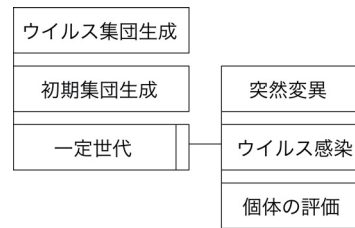


図 2 提案手法のアルゴリズム

$$(job_j, person_i)$$

図 3 ウイルスの染色体

|                     |                  |                  |                  |
|---------------------|------------------|------------------|------------------|
| person <sub>1</sub> | job <sub>3</sub> | job <sub>2</sub> |                  |
| person <sub>2</sub> | job <sub>4</sub> |                  |                  |
| person <sub>3</sub> | job <sub>2</sub> | job <sub>3</sub> | job <sub>1</sub> |

図 4 仕事の移動の例

|                     |                  |                  |                  |
|---------------------|------------------|------------------|------------------|
| person <sub>1</sub> | job <sub>3</sub> | job <sub>2</sub> |                  |
| person <sub>2</sub> | job <sub>4</sub> |                  |                  |
| person <sub>3</sub> | job <sub>2</sub> | job <sub>3</sub> | job <sub>1</sub> |

図 5 仕事の交換の例

|                     |                  |                  |                  |
|---------------------|------------------|------------------|------------------|
| person <sub>1</sub> | job <sub>3</sub> | job <sub>2</sub> |                  |
| person <sub>2</sub> | job <sub>4</sub> |                  |                  |
| person <sub>3</sub> | job <sub>2</sub> | job <sub>3</sub> | job <sub>1</sub> |

図 6 人の交換の例

|                     |                  |                  |                  |
|---------------------|------------------|------------------|------------------|
| person <sub>1</sub> | job <sub>3</sub> | job <sub>2</sub> |                  |
| person <sub>2</sub> | job <sub>4</sub> |                  |                  |
| person <sub>3</sub> | job <sub>2</sub> | job <sub>3</sub> | job <sub>1</sub> |

↓ ウイルス感染

|                     |                  |                  |  |
|---------------------|------------------|------------------|--|
| person <sub>1</sub> | job <sub>3</sub> | job <sub>2</sub> |  |
| person <sub>2</sub> | job <sub>4</sub> | job <sub>1</sub> |  |
| person <sub>3</sub> | job <sub>2</sub> | job <sub>3</sub> |  |

図 7 ウイルス感染の例

表 3 から以下のことがわかる。本手法が最もよい結果となった。人手による解と比較すると初日のデータで約 800 点、2 日目のデータで約 900 点の改善が見られる。

表 4 から次のことがわかる。人手による解が最も良い結果を示した。また、ES にウイルス感染を行うことで初日、二日目とも約 3% の改善が見られる。

表 3 総合違反点数

|     | 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 |
| GA  | 555   | 631 | 52 | 546    | 682 | 70 |
| ES  | 357   | 383 | 14 | 302    | 341 | 19 |
| 本手法 | 347   | 367 | 12 | 278    | 303 | 22 |

表 4 部分解含有率

|     | 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 |
| GA  | 0.78  | 0.72 | 0.04 | 0.8    | 0.70 | 0.07 |
| 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 |

表 5 世代ごとの総合違反点数

| 世代数 | 学園祭初日 |               | 学園祭2日目 |               |
|-----|-------|---------------|--------|---------------|
|     | ES    | ES+<br>ウイルス感染 | ES     | ES+<br>ウイルス感染 |
| 0   | 4600  | 4600          | 4386   | 4386          |
| 100 | 2163  | 1469          | 2129   | 1387          |
| 200 | 1513  | 923           | 1504   | 875           |
| 400 | 913   | 563           | 920    | 538           |
| 600 | 666   | 466           | 662    | 439           |

表 5から次のことがわかる。ES が 200 世代目に達成した違反点数を本手法は 100 世代目で達成している。同様に ES の 400 世代目の値を本手法は 200 世代目で達成している。これらの結果からウイルス感染を行うことで収束性が約 2 倍に改善されていることがわかる。

#### 4.3. 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い、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上手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条件「一年生にはさまざまな仕事を体験してもらう」を満たされた一年生が少ない。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上手くできており、特に一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 を共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。

#### 参考文献

- [1] J.Homberger, H.Gehring: Two evolutionary metaheuristics for the vehicle routing problem with time windows, In: Laporte, G, Sement, F. (Eds.), Metaheuristics for Location and routing Problems. Information Systems and Operational Research 37(Special issue), pp297-318 (1999).
- [2] D.Mester, O.Bräysy: Active guided evolution strategies for large-scale vehicle routing problems with time windows, Computers & Operations Research, (印刷中)
- [3] 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).
- [4] 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).
- [5] 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).