

# 遺伝的アルゴリズムを用いた 個人の行動スケジュール自動調整手法

遠藤宏太<sup>†</sup> 鈴木優<sup>‡</sup> 川越恭二<sup>‡</sup>

<sup>†</sup>立命館大学大学院理工学研究科

<sup>‡</sup>立命館大学情報理工学部

## 1 はじめに

近年、時間を効率的に使いたい個人が増加している。時間は貴重な資源であるが、多忙な社会人は自分のスキルアップや目標達成するための時間、自分の趣味の時間を確保するために、限られた時間を効率的に使わなければならない。そこで本研究では、個人に適したスケジュールを自動作成し、自動調整するシステムを取り上げる。

既存のスケジュール手法 [1] では、複数のタスクが存在する場合、締切が近いタスクを優先し、タスクを前倒しでスケジュールに組込む。しかし、重要度の高いタスクの締切が重要度の低いタスクよりも後の場合、優先順位が低くなるという状況が発生するため、利用者に最適なスケジュールを提示することは難しい。さらに個人を対象としたスケジュール管理を行うためには、タスクの重要性やタスクの進捗が個人によって異なる点やタスク実行場所も考える必要がある。

そこで本稿では、重要度と締切時間からなるベクトルにより優先順位を表現し、優先順位を動的に変化させる。そして、遺伝的アルゴリズムを利用し、タスク実行場所とタスクの優先順位とタスクの進捗状況を基に、利用者のスケジュールの傾向を反映させた個人に適したスケジュールを提示する手法を提案する。

## 2 スケジュール自動調整手法の提案

### 2.1 定義

以下では、本研究で利用する用語を定義する。スケジュール  $S$  はタスク  $T_i$  のシーケンスであり、 $S = \langle T_0, \dots, T_m \rangle$  で表す。同様に、利用者が日常行っているスケジュールを意味するスケジュールパターン  $S'$  もタスク  $T'_i$  のシーケンスであり、 $S' = \langle T'_0, \dots, T'_m \rangle$  で表す。そして、 $T_i, T'_i$  は各タスクの集合の要素であり、具体的なタスクである。スケジュール  $S, S'$  に何も予定されない時間帯がある場合に、特殊な空きタスクを割り当てるものとする。

以降では、あるタスク  $T_k$  の締切を  $d_k$ 、初期重要度を  $e_k$ 、タスク実行場所を  $l_k$ 、タスクの進捗集合を  $P_k$  と記述し、 $P_k = \{p_{k,0}, \dots, p_{k,d_k}\}$  と定義する。 $p_{k,i}$  はタスク追加日から  $i$  日後の進捗を表す。ここで進捗は、

Automatic Schedule Adjustment for Individual Action with Genetic Algorithms

Kota ENDO<sup>†</sup>, Yu SUZUKI<sup>‡</sup> and Kyoji KAWAGOE<sup>‡</sup>

<sup>†</sup>Graduate School of Science and Engineering, Ritsumeikan University <sup>‡</sup>Faculty of Information Science and Engineering, Ritsumeikan University

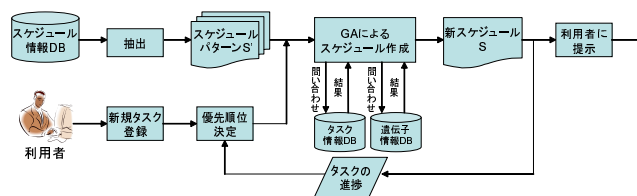


図 1: システム概要

どれだけタスクに時間を要したのかをいう。

### 2.2 スケジュール自動調整手法の概要

提案するシステム概要を図 1 に示す。スケジュールの自動調整を行うためには、タスクをどの順番で着手すべきか、また、一日に割り振るタスクの時間を考える必要がある。まず、利用者の入力する重要度と締切により、優先順位を決定する。そして、過去の進捗状況から、利用者が一日にタスクに費やす時間を求める。そしてスケジュールパターン  $S'$  にタスクを割り振り、利用者に調整後のスケジュールを提示する。

まず、新規タスクが追加された場合の具体的な処理手順を述べる。

**Step1** 新しいタスク  $T_{new}$  を追加する。利用者は初期重要度  $e_{new}$ 、締切  $d_{new}$ 、タスク実行場所  $l_{new}$  を入力する。

**Step2** システムは以下で述べるタスクごとの優先順位を基に、本日を  $now$  として、進捗  $p_{new,now}$ 、タスク実行場所  $l_{new}$  を考慮し、タスク  $T_{new}$  を  $T'_i$  と置き換え、新しいスケジュール  $S$  を作成する。ここで、 $T'_i$  はスケジュールパターン  $S'$  を構成するタスク中の空きタスク、もしくは優先順位の低いタスクを表す。このとき、効果的にスケジュール作成が可能である遺伝的アルゴリズムを利用する。

**Step3** 作成したスケジュールを利用者に提示する。

次に、本日のスケジュールから次の日のスケジュールを作成する具体的な処理手順を述べる。

**Step1** 利用者は本日行ったタスクの進捗  $p_{new,now}$  を入力する。

**Step2** システムはタスクの進捗  $p_{new,now}$  を基に優先順位を新たに決定し、遺伝的アルゴリズムを利用して、 $p_{new,now}$  から次の日の進捗  $p_{new,now+1}$  を過去の進捗から求め、タスク  $T_{new}$  を  $T'_i$  と置き換え、新しいスケジュール  $S$  を作成する。

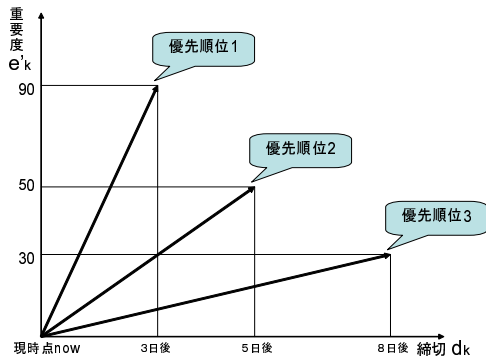


図 2: タスクの優先順位

Step3 作成したスケジュールを利用者に提示する。

### 2.3 スケジュールのパターン化

利用者のスケジュール履歴が格納されているデータベースから、頻出する複数個のタスクを抽出し、スケジュールパターン  $S'$  内のタスクとする。まず、利用者が普段いる場所を特定するため、曜日ごと時間ごとに割合が多い場所を抽出し、タスク実行場所  $l_k$  を決定する。次に、頻出タスクを抽出する操作を行い、曜日ごと時間ごとに毎回行われるような日常的なタスクを固定タスクとして抽出する。例えば、毎日0時から7時まで就寝の場合、就寝タスクが固定タスクとなる。

### 2.4 タスクの優先順位

スケジュールを作成する際、複数のタスクが存在する場合、どのタスクを優先すべきか分からない。そのため、タスクの優先順位を考える必要がある。本研究では、タスクの優先順位を図 2 に示すベクトルの角度を利用して求める。重要度が高く締切に近いタスクはベクトルの角度は大きく、重要度が低く締切が先のタスクはベクトルの角度は小さくなる。ベクトルの角度が等しくなる場合はベクトルの短いタスクを優先する。

このように本手法では、時間軸を利用しているため、時間の経過によりベクトルの角度は変化する。そして、タスクの進捗状況によって利用者のタスクに対する重要度は異なると考え、(1) 式で示すように、タスク  $T_k$  の初期重要度  $e_k$  を本日の進捗  $p_{k,now}$  により減少させ、ベクトルの角度を動的に変化させることにより、新たな優先順位を決定する。ここで、タスク  $T_k$  の重要度を  $e'_k$ 、最小必要時間を  $n_k$ 、消費時間を  $w_k$  と表す。最小必要時間は、過去に同一タスクに要した時間をいい、消費時間は現時点におけるタスクの所要時間をいう。

$$\begin{aligned} e'_k &= e_k(1 - p_{k,now}) \\ p_{k,now} &= \frac{w_k}{n_k} \end{aligned} \quad (1)$$

### 2.5 遺伝的アルゴリズムを用いたタスクの割り振り

本研究では、刻々と変化する優先順位を遺伝的アルゴリズム (GA: Genetic Algorithms)[2] に適用し、タスクをスケジュールに割り振る。GA は初期集団を生成し、適応度を評価し、選択、交叉、突然変異を繰り返して最適解を求めるアルゴリズムであり、特徴ある利用者のスケジュール情報を反映できるため、GA を利用

する。ここで過去のタスクの進捗情報を保持しておき、保持した情報を GA の処理で利用する。タスク  $T_k$  と同じ内容の過去に行われたタスクの進捗  $T_k^{old}$  の集合  $P_k^{old}$  を以下のように定義する。  $P_k^{old} = \{p_{k,0}^{old}, \dots, p_{k,d_k}^{old}\}$

まず、2.3 節で抽出したスケジュールパターン  $S'$  内のタスク  $T'_i$  にタスク  $T_{new}$  をランダムに入れたものを複数作成し、初期集団とする。そして本日を  $now$  としたときに、(2) 式に示す評価関数  $Fitness$  を作成し、個々のタスクの適応度  $Ada$  を (3) 式により求める。

$$\begin{aligned} Fitness &= \cdot \cos + \cdot location\_check \\ &+ \cdot (p_{new,now} - p_{new,now}^{old}) \\ &+ \cdot (p_{new,now+1} - p_{new,now+1}^{old}) \end{aligned} \quad (2)$$

$$location\_check = \begin{cases} 1 & (\text{タスク実行場所 } l_{new} \text{ と不一致}) \\ 0 & (\text{タスク実行場所 } l_{new} \text{ と一致}) \end{cases}$$

$$Ada = \frac{1}{Fitness} \quad (3)$$

ここで、 $\cdot$ 、 $\cdot$ 、 $\cdot$  はそれぞれ重み係数を表す。また、締切  $d_k$  と重要度  $e'_k$  を用い、 $\vec{x} = (d_k - now, e'_k)^T$ 、 $\vec{y} = (d_k - now, 0)^T$  で表し、 $\vec{x}$  と  $\vec{y}$  で表される角度をとす。そして  $location\_check$  は、 $S'$  にランダムに割り振られたタスクの場所が利用者の入力したタスク実行場所  $l_{new}$  と一致しているかを表す。さらに、利用者の入力する実際に行われたタスクの進捗  $p_{new,now}$  と過去の進捗  $p_{new,now}^{old}$  を比較する。そして、ランダムに割り振られた次の日のタスクの進捗  $p_{new,now+1}$  と過去の進捗  $p_{new,now+1}^{old}$  を比較する。これにより、 $S'$  に割り振られたタスク  $T_{new}$  の優先順位が高く、タスク実行場所が一致し、実際に行われたタスクの進捗と次の日のタスクの進捗が過去に行われた進捗に類似すれば適応度は高くなる。

次に、選択操作ではタスクが割り振られたスケジュール  $S$  の各タスクの平均適応度が高いスケジュール  $S$  を残す操作を行う。また交叉操作は、適応度の低いタスクを入れ替え、適応度を高くする。さらに、突然変異操作は、進捗  $p_{new,now}^{old}$  では締切  $d_{new}$  を満足できない場合に、例外的に多くのタスクを割り当てる。

以上の選択、交叉、突然変異を最良なスケジュールが作成されるまで繰り返していくことで、利用者のスケジュールの傾向を反映させたスケジュールを作成する。

## 3 おわりに

本研究ではタスクの優先順位を重要度と締切により表現し、タスクの進捗状況を利用者のスケジュールの傾向に合わせることにより、個人に適したスケジュールの自動調整を行う手法を提案した。これにより利用者は、自分のスケジュールの傾向に合ったスケジュールを得ることができるようになる。今後は、試作システムを作成し、評価実験を行う予定である。

## 参考文献

- [1] 大向他. 多様かつ曖昧な個人タスクのための管理システムの提案と実装. エージェント合同シンポジウム (JAWS), pp. 502-509, 2002.
- [2] 伊庭斉志. 遺伝的アルゴリズムの基礎. オーム社, 1994.