

Host-Parasite 型共進化を用いたスケジューリング

北海道文理科短大 ○高取則彦, 札幌学院大学 皆川雅章, 北海道大学 嘉数侑昇

要 旨

Host-Parasite 型共進化をスケジューリング問題の解法に応用する. Host はスケジュールを表し, Parasite はその中に存在する機械の遊休時間を調べる. これらが相互に影響しながら進化することにより, 解の探索が進められる. 共進化計算モデルの形式的記述を行い, さらに具体的モデルについて述べる.

1. はじめに

生物学では, 共進化とは 2 つ以上の種が互いに影響を及ぼし合いながら同時に進化していくことを意味する. その種間の相互作用にはいくつかの形式が知られており '寄主-寄生者 (Host-Parasite)' の関係はその 1 つである. この関係では, 寄生者側の種が寄主側の種を利用して一方的に利益を得, 寄主側はそれにより損害を被る. 寄主, 寄生者とも自種の存続のため相手種の進化に対抗するように変化し, 結果として両者はともに進化していく. この現象を最適化問題や制約充足問題などの解法に応用する研究がなされ, 興味深い結果が得られている [1][2]. 本報告では, この共進化をスケジューリング問題の解法に応用する. 共進化計算モデルの形式的記述を試み, スケジューリング問題のための具体的モデルについて述べる.

2. 共進化モデルの形式的表現

ここでは共進化モデルを次の 5 項組で表す.

$$CM = (SP, S, \Psi, \Gamma, \tau) \quad (1)$$

各項目は次のとおりである.

SP : スケジューリング問題.

$S = \{S_i \mid i=1, \dots, ns\}$: 種の集合.

$S_i = (I_i, F_i, \Omega_i, n_i)$: 第 i 種.

I_i : 第 i 種の個体空間.

$F_i: I_1 \times \dots \times I_{ns} \rightarrow \mathfrak{R}$: 適応度関数.

$\Omega_i = \{\omega_{ik} \mid \omega_{ik}: I_i^{n_i} \rightarrow I_i^{n_i}; k=1, \dots, no_i\}$: 第 i 種に適用される遺伝オペレータの集合.

n_i : 集団のサイズ.

$\Psi: S \times S \rightarrow IS$: 種間相互作用の形式.

IS : 種間関係空間.

Γ : 世代遷移関数

$$\Gamma: I_i^{n_i} \rightarrow I_i^{n_i}$$

τ : 終了条件

$$\tau: I_1^{n_1} \times \dots \times I_i^{n_i} \times \dots \times I_{ns}^{n_{ns}} \rightarrow \{true, false\}$$

共進化では個体の適応度は他の種との関わりの中

で決められるので, 第 i 種の適応度関数 F_i は I_1, \dots, I_{ns} の直積集合から実数全体の集合への写像となる. 種は似た特徴をもつ互いに交配可能な個体の集まりと考えているので, 進化計算のオペレータは種ごとに定める.

2 種間の相互作用形式のうち, ここでは寄生関係 $parasitism \in IS$ を対象とする. 2 つの種 S_i, S_j の各個体について, 相手種の個体との相互作用から導かれる状態量をそれぞれ E_i, E_j とする.

$$E_i: S_i \times S_j \rightarrow \mathfrak{R}$$

$$E_j: S_i \times S_j \rightarrow \mathfrak{R}$$

この状態量の変化 $\Delta E_i, \Delta E_j$ とそれぞれの種の個体の適応度の変化 $\Delta F_i, \Delta F_j$ との関係から, 2 種間の関係を定義する. すなわち, $\forall \xi_i \in S_i, \forall \xi_j \in S_j$ について

$$\Delta E_i(\xi_i, \xi_j) > 0 \Rightarrow \Delta F_j(\dots, \xi_i, \dots, \xi_j, \dots) > 0$$

$$\Delta E_j(\xi_i, \xi_j) > 0 \Rightarrow \Delta F_i(\dots, \xi_i, \dots, \xi_j, \dots) < 0$$

が満たされるとき, S_i と S_j の関係を $parasitism$ と呼ぶことにする.

2 種間の相互作用の形式として, 他に $competition$ (競争) や $symbiosis$ (共生) などが考えられるが, ここでは省略する.

3. モデルの具体的記述

式(1)の問題 SP として, 単純なジョブショップ・スケジューリング問題を考え, C_{max} の最小化問題を対象とする [3]. 以下が与えられているものとする.

ジョブの集合: $\{J_j \mid j=1, \dots, nj\}$

機械の集合: $\{M_m \mid m=1, \dots, nm\}$

オペレーションの集合: $\{o_{j,m} \mid j=1, \dots, nj; m=1, \dots, nm\}$

$o_{j,m}$ は機械 M_m で処理するジョブ J_j のオペレーションである. $o_{j,m}$ の処理時間 $p_{j,m}$ も既知とする. この問題の解は, 全体の総処理時間が最小となる $o_{j,m}$ の開始時刻 $s_{j,m}$ である.

種として Host と Parasite の 2 つを設定し, Host は問題の解, Parasite は解における評価規準や制約条件

の充足度を調べるものとする。そしてHostの個体は評価規準に関して良いものほど高い適応度を与えられ、Parasiteの個体はHost個体の‘悪さ’を見つけることにより適応度を上げるようにする。これらHostとParasiteが相互に影響しながら進化することにより、解の探索が進められる。

・**個体空間と個体の表現** Hostの個体はスケジューリング問題の解すなわちスケジュールである。各機械ごとのタスクの処理順序列を全機械について連結したものとする。したがって個体空間は

$$I_1 = \underbrace{Q \times \dots \times Q}_{nm}$$

となる。ここで Q は集合 $Jl = \{1, 2, \dots, nj\}$ 上の全単射写像 $\pi: Jl \rightarrow Jl$ を元とする集合である。個体は $a_k = (\pi_{k,1}, \dots, \pi_{k,m}, \dots, \pi_{k,nm})$ とする。 $\pi_{k,m}$ は機械 m 上のジョブの処理順序を表し、 $\pi_{k,m}(l)$ は機械 m 上で l 番目に処理するジョブの番号を与える。

Parasiteは指定された機械の遊休時間を調べるものとする。Parasiteの個体空間は

$$I_2 = MI$$

ただし $MI = \{1, 2, \dots, nm\}$ である。個体は $b_q = (r_q)$ 、 $r_q \in MI$ とする。Host個体 a_k が表すスケジュールにおいて機械 r_q の遊休時間 $idle_time(a_k, b_q)$ を次式により計算する。

$$idle_time(a_k, b_q) = \sum_{k=0}^{nj-1} h(s_{u(l+1), r_q} - (s_{u(l), r_q} + p_{u(l), r_q})),$$

$$h(x) = \begin{cases} x; & \text{if } x \geq 0 \\ 0; & \text{if } x < 0 \end{cases} \quad (2)$$

ここで $u(l) = \pi_{k, r_q}(l)$ である。また便宜上 $\pi_{k, m}(0) = 0$ 、 $s_{0, m} = 0$ 、 $p_{0, m} = 0$ とする。

・**個体の評価と適応度** Host個体 a_k の適応度は、Parasite個体 b_q によって算出された遊休時間の総和

$$eval_h(a_k) = \sum_{q=1}^{n_2} idle_time(a_k, b_q) \quad (3)$$

を用いて次式により算出する。

$$fitness_h(a_k) = (eval_h_{\max} - eval_h(a_k))^2,$$

$$eval_h_{\max} = \max_k (eval_h(a_k)) \quad (4)$$

Parasite個体 b_q の適応度は、Host個体 a_k を調べて求めた遊休時間の総和

$$eval_p(b_q) = \sum_{k=1}^{n_1} idle_time(a_k, b_q) \quad (5)$$

を用いて次式により算出する。

$$fitness_p(b_q) = (eval_p(b_q) - eval_p_{\max})^2,$$

$$eval_p_{\max} = \max_q (eval_p(b_q)) \quad (6)$$

・**遺伝オペレータ** Hostでは $\Omega_1 = \{selection, crossover, mutation\}$ とする。

selection: ルーレット選択。

crossover: 順序型一様交差。2個体 a_k 、 a_k' の同じ位置(同じ m)にあるサブストリング $\pi_{k,m}$ 、 $\pi_{k',m}$ に対して適用する。

mutation: 順序型突然変異。ランダムに選んだ l, l' について $\pi_{k,m}(l)$ を $\pi_{k,m}(l')$ を交換する。

Parasiteでは $\Omega_2 = \{selection, mutation\}$ である。

selection: ルーレット選択。

mutation: r_q をランダムに選択した $r_q' \in MI - \{r_q\}$ に変える。

以上を用いたモデルの計算手続きを以下に示す。

```

begin
  t := 0;
  for i := 1 to ns do initialize  $P_i(t) \in I_i^{n_i}$ ;
  evaluate  $P_1(t), \dots, P_{ns}(t)$ ;
  while ( $\tau(P_1(t), \dots, P_{ns}(t)) \neq true$ ) do
    begin
      for i := 1 to ns do
        begin
           $P_i'(t) := crossover(P_i(t))$ ;
           $P_i''(t) := mutate(P_i'(t))$ ;
        end
        evaluate  $P_1''(t), \dots, P_{ns}''(t)$ ;
        for i := 1 to ns do  $P_i(t+1) := select(P_i''(t))$ ;
        t := t + 1;
      end
    end
end.

```

4. おわりに

Host-Parasite型共進化をスケジューリング問題の解法に応用した。Hostはスケジュールを表すものとし、Parasiteはその中に存在する機械の遊休時間を調べるものとした。共進化計算モデルの形式的記述を行い、さらに具体的モデルについて述べた。実験による検証は今後行う。

参考文献

- [1] Hillis, W. D. (1991). *Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure*. In: *Artificial Life II*, (C. G. Langton, et al. (Eds.)), pp.313-324, Addison-Wesley.
- [2] Paredis, J. (1994). *Co-evolutionary Constraint Satisfaction*. In: *Parallel Problem Solving from Nature - PPSN III*, (Y. Davidor, et al. (Eds.)), pp.46-55, Springer.
- [3] Blazewicz, J., Ecker, K. H., Schmidt, G. and Weglarz, J. (1994). *Scheduling in Computer and Manufacturing Systems (2nd Ed.)*. Springer.
- [4] Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford University Press.