

特別研究報告書

遺伝的プログラミングを用いた
分散型ジョブマッチング

指導教員 松原 繁夫 准教授

京都大学工学部情報学科

湯月 亮平

平成 25 年 2 月 1 日

遺伝的プログラミングを用いた分散型ジョブマッチング

湯月 亮平

内容梗概

ジョブマッチングは、企業と求職者との組を見つける問題であり、また、どのような採用・求職戦略が最善かを調べる問題である。この問題は、Gale と Shapley による研究を端緒として、2 サイドマッチングとしてゲーム理論分野において盛んに研究されている。また、アルゴリズムの側面から計算機科学分野でも盛んに研究されている。米国や日本において、研修医の病院配属決定に Gale-Shapley アルゴリズムが用いられるなど、実応用という点でも興味深い。

しかし、その応用場面はまだ限られている。この理由の一つは、モデルに強い仮定が置かれていることである。例えば、研修医病院配属問題ではマッチング過程の進行に伴い、研修医や病院が戦略を変更することは考慮されていない。しかし、日本の新卒就職市場などを見れば、周囲の内定状況に応じて、基準を変化させると考えられる。これは、ゲーム理論の用語で言えば、逐次ゲームとして表現されるべき状況である。この問題に対して、Haeringer らは逐次ゲームとしての分散型ジョブマッチングモデルを提案し、均衡分析を行った。企業が求職者に内定通知を出し、求職者が受諾・拒否を決定するモデルである。しかし、ここでも、参加者が他者の正確な選好を持つことを仮定するなど、分析のために別の強い仮定が置かれており、広範な領域での応用の障壁となっている。

この問題に対して、本研究では進化的計算手法の一つである遺伝的プログラミング (GP) の適用を考える。遺伝的プログラミングとは、解候補集団に対して評価、選択、交叉、突然変異という一連の操作を行うことで最適解を求める手法である。仮定が緩和された問題に対して良いマッチング結果を与える戦略が見つかれば、そこから様々な考察・議論が可能となる。一方で、マッチング問題はゲーム的状况であり、均衡解であると保証されない場合、解の善し悪しの議論に関して、その妥当性が問われる。そこで、本研究では以下の課題に取り組む。

遺伝的プログラミングとしての問題表現 遺伝的プログラミングを用いるには、交叉や突然変異といった操作を加えて新たに生成された個体が採用・求職戦略として意味のあるものであることが必要である。また、突然変異率の設定などについても検討が必要である。

正確な選好情報を持たない場合の戦略 上述のように分散型ジョブマッチングでは個々の参加者が他者の正確な選好を知っているという仮定がなされているが、この仮定は現実的ではない。選好に関する正確な情報を持たない場合に、どのような戦略が有力となるのかを調べる必要がある。

本課題を解決するために、まず、Haeringer らの分散ジョブマッチングと同じ設定の元で、GP を用いて採用・求職戦略を求める。ここで、ゲーム理論での均衡解と同じ戦略が得られれば、GP による問題表現法の妥当性を示す一つの証左を与えることになる。つぎに、正確な選好情報を持たない場合に GP を適用する。ここでは、選好情報の精度に関する設定が必要である。本研究では、個々の求職者は個別の企業に関する他の求職者の選好を知らないが、どの業種を希望しているかの情報を知っていると仮定する。ここで、業種を電機、運輸といった粗い分類と考えれば、低精度の選好情報を持つことになり、一方、重電、家電、鉄道、航空といったより詳細な分類と考えれば高精度の選好情報を持つことになる。このときに、高精度・中精度・低精度の3つの場合を設定し、各場合における求職者の戦略を求め、分析する。本研究の貢献は以下の通りである。

遺伝的プログラミングとしての表現法の提案 企業の採用戦略としては、求職者を評価順に並べた際に、当該求職者が第何位であるか、他の企業からの内定通知を得ているかを要素として、それらの IF-THEN 結合で戦略を表現した。また、求職者の求職戦略としては、当該企業が第何希望であるか、また、マッチング実現性の高い企業を基準とした相対順位を要素として、それらの IF-THEN 結合で戦略を表現した。Haeringer らの分散ジョブマッチングと同じ設定の元で GP を用いて均衡解が得られることを確認した。

正確な選好情報を持たない場合の戦略の発見 (1) 高精度の選好情報を持つときは、マッチング実現性の高い企業か、それ以上の希望順位の企業からの内定通知であれば承諾する、(2) 低精度の選好情報を持つときは、どの企業からの内定通知でも承諾する、(3) 中精度の選好情報を持つときは、各時点でのマッチング実現性の高い企業を基準として、それ以上の希望順位の企業からの内定通知であれば承諾する、つまり、マッチング過程の進行に伴い、他者の内定が決まるにつれ、承諾基準を引き下げるという戦略が得られた。

Decentralized Job Matching Using Genetic Programming

Ryohei YUZUKI

Abstract

Job Matching Problem addresses finding pairs of firms and workers, and optimal strategy. This problem has been extensively studied in Game theory as 2-side matching. It has also been extensively studied in the field of computer science aspects of the algorithm. It is used in assignment hospital resident physician in America and Japan. It is also interesting in terms of practical applications.

However, Its application scene is still limited. One of the reason is that strong assumption is placed in the model. For example, in the Assignment problem in hospital medicine training, it is not considered that trainee doctors and hospitals change thier strategies in the process of matching. However, in Job Market in Japan, players can change their criteria depending on the circumstances of the prospective range. This situation should be represented as a sequential game. Haeringer proposed Decentralized Job Matching model as sequential game, and analyzed the equilibrium, which firms make offers to workers and workers decide accept ot decline it. However, This problem has also the strong assumption that all players know all other player's preference, and it inhibits a wide range of applications in the area.

For this problem, this study considers application of genetic programming (GP), which is one of evolutionary computation. GP is the method of find the optimum solution by applying evaluation, selection, crossover, mutation to candidate solutions group. If the strategy that gives good results for matching assumption is found, we can improve various considerations and discussions. On the other hand, matching problem is a game situation, and if it is not guaranteed to be the equilibrium solution, for a discussion of the quality of the solution, its validity is questioned. Therefore this study addresses the following issues.

Problem of representation as GP For using GP, individuals which are generated by crossover or mutation are required to be meaningful as job hunting strategies. It is also required to consider about configuration like mu-

tation rate.

Strategy in the case which preference is incorrect The assumption of this model is not realistic. It is necessary to determine what strategy is influential in the case which players don't have accurate information about the preferences.

To solve these problems, First, I try to find strategy in the same case of Haeringer's decentralized job matching model. If the same equilibrium solution is obtained, it give a proof of validity of the method of GP. Second, I applied GP to the case which all players don't have accurate information about preference. Configuration is necessary on the accuracy of preference information. This study assumed that each worker knows about preference of type of industry. If classification is rough, they have low-accurate information. Conversely, If classification is detailed, they have high-accurate information. I examine the strategies in each case and analyze them. Contribution of this study are as follows.

Proposition the method of representation For firm strategy, I represented it as IF-THEN, that is composed of elements of the rank of worker, and the number of offers which he has. For worker strategy, I represented it as IF-THEN, that is composed of elements of the rank of firm, and rank of optimal firm. I can obtain the same equilibrium solution that Haeringer proposed.

Finding out the strategy when preference is incorrect I obtained the following strategies.

(1)In the case which players have high-accurate information, "If worker receives an offer from his optimal mate or more preferred mate, then accepts it". (2)In the case which players have low-accurate information, "If worker receives an offer, then always accepts it". (3)In the case which players have medium-accurate information, "At first, if worker receive an offer from his optimal mate or more preferred mate, then accepts it. Later, worker come to compromise with the progress of the matching process".

遺伝的プログラミングを用いた分散型ジョブマッチング

目次

第1章	はじめに	1
第2章	関連研究	3
2.1	分散型ジョブマッチング	3
2.1.1	Offering stage と Responding stage	3
2.1.2	意志決定の遅延不可性と不可逆性	3
2.1.3	企業の戦略	4
2.1.4	求職者の戦略	4
2.1.5	最適解と最適戦略	5
2.2	遺伝的プログラミング	6
2.2.1	初期集団生成	7
2.2.2	評価	7
2.2.3	進化	7
第3章	モデル化の手法	9
3.1	戦略の表現	9
3.2	変数ノード	10
第4章	実験	11
4.1	準備	11
4.1.1	定義	11
4.1.2	初期化	11
4.1.3	マッチング	12
4.1.4	評価	13
4.1.5	進化	14
4.2	関連研究に基づいた実験	15
4.2.1	企業が戦略的でなく，求職者が逐次手番の場合	15
4.2.2	企業が戦略的であり，求職者が逐次手番の場合	16
4.2.3	企業が戦略的であり，求職者が同時手番の場合	16
4.3	拡張したモデルでの実験	17
4.3.1	予測	17

4.3.2	結果	18
第5章	考察	20
第6章	まとめ	25
	謝辞	25
	参考文献	25

第1章 はじめに

2サイドマッチングは、ある2つのグループの要素に対して安定したマッチングを与える手法として長年盛んに研究が行われおり、実際に研修医配属にも応用されている手法である。ある企業 f, f' 、ある求職者 w, w' について、以下の条件をすべて満たしている場合に企業と求職者の組 (f, w) をマッチングにおけるブロッキングペアと呼ぶ。

- f はの選好順序の上で、現在マッチングしている w' よりも w の方を好む。
 - w はの選好順序の上で、現在マッチングしている f' よりも f の方を好む。
- つまり、 f と w は共謀してマッチングを破壊することができる。安定したマッチングにおいてはこのようなブロッキングペアが存在しないことが絶対条件となる。

Gale と Shapley によって提案された遅延許諾アルゴリズム (DA アルゴリズム) は安定したマッチングを合理的な計算時間で得られ、多くのマッチング手法の基礎となっている。DA アルゴリズムの具体的な流れを簡潔に説明すると以下のようなになる。

1. 各求職者 (企業) はオファー可能な企業 (求職者) の中から自身の選好順序の上で最も高い相手にオファーを出していく。もし、オファーを出せる相手がいなければ終了する。
2. 各企業 (求職者) は自身が受けた求職者 (企業) からのオファーの中で選好順序が最も上の相手からのオファーを承諾し、残りを拒否する。各求職者 (企業) は拒否された相手に再びオファーを出すことはできない。
3. オファーを出せる相手がいなくなるか、全ての企業と求職者がちょうど1対1でマッチングするまで1, 2を繰り返す。

求職者からオファーを出す場合に DA アルゴリズムを適用すると、求職者にとって最適な (W-optimal) マッチングが得られ、同様に企業からオファーを出せば企業にとって最適な (F-optimal) マッチングが得られることが知られている。しかも企業・求職者共に他の相手の選好順序を知らなくてもこの手法は適用できる。

しかし、企業からオファーを出す場合に、DA アルゴリズムの下でマッチングを行うと、オファーを出された求職者は、少なくとも一時的には自分にとって選考順序の低い相手と仮マッチをすることになり、またそれを自分の意思では解消

できず、現在の仮マッチ相手より選考順序の高い相手からオファーを受けるとで解消できない。また、企業は過程の中で既に仮マッチしている求職者に対してオファーをするという状況が出てくることがある。

求職者は志望の低い企業からのオファーを断るだろうし、企業は既に内定が決まっている求職者にはオファーを差し控えるかもしれない。つまり DA アルゴリズムは必ずしも現実の就職活動市場における企業・求職者の意思を反映しているわけではない。

Guillaume Haeringer ・ Myrna Wooders は上記で述べた企業・求職者の意思を反映するために分散型ジョブマッチング市場のモデルを提案した。彼らはそのモデルをゲーム理論における逐次手番ゲームとして表現し、解析を行った。その結果「選好順序が W -optimal の相手より下の企業のオファーを断り続け、 W -optimal の相手からのオファーが来たらそれを承諾する」ことが部分ゲーム完全均衡における最適戦略となり、その結果最適解として W -optimal なマッチングが得られることが明らかになった。しかし、それは各求職者が他の相手の選好順序を完全に把握していなければ得ることが非常に困難なものであり、現実の就職活動市場では相手の選好順序を完全に把握することはまず無理であろう。しかし、求職者同士の交流によって相手がどこの業種を希望しているのか、その中で人気の高い企業はどこかという情報を得ることはさほど困難ではない。例えば、企業 f_1, f_2, f_3 を自動車産業業界の企業、企業 f_4, f_5, f_6 を IT 企業とすると、ある求職者 w_1 が希望している業種が自動車産業である場合、 w_1 の選考順序において f_1, f_2, f_3 が上位にあり、 f_4, f_5, f_6 は中位、下位であることが推測できる。その情報を用いて、他の求職者の選好順序を（完全でなくとも）ある程度推測することができれば、関連研究で得られた手法を現実の就職活動市場に適用できると考えられる。

これらの議論を進めるためには、まず関連研究のモデルの妥当性を検証する必要がある。そこでこのモデルを別の手法で表現し、同様の議論ができるかどうかを検証する。この手の問題を理論的な手法で扱おうとすると、現実では到底実現不可能な強い前提条件を設定しなければならないが、遺伝的アルゴリズム [5]、遺伝的プログラミングなどの進化的アルゴリズムなら理論的に定式化できない問題を扱うことが可能である。特に 3 章の理由により、本研究では遺伝的プログラミングを用いた。

そして、その表現手法の妥当性を検証した後、各求職者が他の相手の選考順序を推測しながら行動をするという仮定を付加したモデルについて同様の検証を行う。

まず、第2章で分散型ジョブマッチング、遺伝的プログラミングを紹介し、第3章でモデル化の手法を述べる。第4章でそれに基づいた実験を行い、第5章で考察、第6章でまとめを述べる。

第2章 関連研究

2.1 分散型ジョブマッチング

分散型とは企業・求職者がそれぞれ他の意思決定とは独立して意思決定を行うことを意味している。特に Guillaume Haeringer・Myrna Wooders のモデルでは大きく分けて4つの仮定が設定されており、その条件の元で得られる最適解及び最適戦略が証明されている（本節中に用いられている例は全て関連論文に即したものを使用している）。

2.1.1 Offering stage と Responding stage

分散型ジョブマッチングは企業が1人の求職者にオファーを出す Offering stage、求職者が自分に来たオファーを処理する Responding stage に分かれており、Offering stage と Responding stage を交互に繰り返しながら展開されていく。

2.1.2 意志決定の遅延不可性と不可逆性

各求職者は企業からのオファーを「保留」することができない、すなわち企業からオファーが来たら必ずそれを「承諾する」か「拒否する」かのどちらかを選択しなければならない。もしオファーを承諾したらその求職者と企業はマッチングを直ちに終了しなければならない。また、一度決定したら、その決定を後で覆すことはできない。すなわち企業は一度断られた求職者には二度とオファーを出すことができず、求職者は一度断った企業から二度とオファーを受けることができない（同様に一度承諾した企業と決裂することはできない）。また企業・求職者共に既に終了している相手とはマッチングできない。

現実社会における意思決定の遅延不可性について ある求職者 w がある企業 f からオファーを受けた場合、一般的にはまず電話か電子メールによるオファーを受け、口頭または電子メールへ返信するという状況が考えられる。いずれの場合も返事をする前に、おおよそ2週間程度の期間があると考えられる。

言い換えれば意思決定には2週間程度しか猶予がなく、それ以上は f は待つてくれない。もし2週間以上たっても意思が固まらない場合、 f は w に自分のオファーを承諾する意思はないとみなし、 w を諦めて別の求職者 w' へオファーを出すことになる。逆にそのオファーを承諾した場合、 w はその旨を f に電話または電子メールで伝える。その後時間をかけて所定の手続きを済ませた時、 w は正式に f からのオファーを承諾し、 f, w の契約が正式に決定するのだが、 w は f に承諾の旨を伝えた時点で、まだ正式決定していなくてもそれ以降就職活動をするのではないと考えられる。

現実社会における意思決定の不可逆性について その求職者 w が f にオファーを承諾することを伝えてからしばらくたって、 w の元により選好順序の高い企業 f' からオファーが来たとする。すると w は喜んで承諾し、 f にオファー承諾を取り消すことを電話または電子メールで伝えることになる。 f とはまだ正式な契約をしていないため w が f のオファーを取り消すことは全くの合法であるが、そうした場合、 f と w との間でトラブルが発生し、 w の今後の就職活動に支障をきたす可能性がある（実際、日本では企業が内定辞退者に内定にかかった費用を請求する事例や企業が内定辞退者の別の内定先の企業に連絡をし、彼の評判を下げようとする事例など、企業と内定辞退者間の金銭・人間関係に関するトラブルは数多く報告されている）。従って現実社会において、リスク無しで決定を覆すことは難しいと考えられる。

2.1.3 企業の戦略

企業の行動は「戦略的でない場合」または「戦略的である場合」の2種類に分類される。

戦略的でない場合 企業は自身の選好順序に従って逐次オファーを出していく。

戦略的である場合 企業は自身の選好順序に関係なく望む順番でオファーを出してもよい。すなわち、第1希望の求職者に他の企業がオファーを出していれば、敢えてそこを避けるといった戦略をとることができる。

2.1.4 求職者の戦略

求職者の行動は「逐次手番」または「同時手番」の2種類に分類される。

逐次手番 各求職者は1人ずつ順番に行動する。すなわち各求職者は自分の番が来る前に行動した他の求職者の行動を観察することができる。

同時手番 各求職者は同時に行動する。すなわち各求職者は自分が行動している間、他の求職者の行動を観察することができない。

2.1.5 最適解と最適戦略

上記の条件の下でマッチングを行うと以下のような結果が得られる。

- 企業が戦略的でなく、求職者が逐次手番の場合、求職者にとって最善のマッチングが得られ、「最善の相手より下の相手からのオファーを断り続ける」という戦略が唯一の均衡戦略として現れる。
- 企業が戦略的で、求職者が逐次手番の場合、求職者にとって最善のマッチングが得られるが、均衡戦略は複数ある場合がある。例えば企業 f_1, f_2 、求職者 w_1, w_2 について、選好順序が以下のように与えられているとする。

$$P_{f_1} = (w_2, w_1)$$

$$P_{f_2} = (w_1, w_2)$$

$$P_{w_1} = (f_1, f_2)$$

$$P_{w_2} = (f_2, f_1)$$

この時 W-optimal なマッチングは $\{(f_1, w_1), (f_2, w_2)\}$, F-optimal なマッチングは $\{(f_1, w_2), (f_2, w_1)\}$ となる。このとき W-optimal なマッチングを得る戦略として

- 企業が「F-optimal にオファーを出す」という戦略をとり、かつ求職者が「最初に来たオファーを拒否する」という戦略をとる場合
- 企業が「W-optimal にオファーを出す」という戦略をとり、かつ求職者が「最初に来たオファーを承諾する」という戦略をとる場合

の2通りの均衡戦略が存在する。

- 企業が戦略的で、求職者が同時手番の場合、不安定な解が得られる可能性がある。同時手番の場合、各求職者は行動中に他の求職者の行動を知ることができない。ゆえに全ての企業が誰とも重複することなくオファーを出した場合、それを断ることで自分だけがマッチングに失敗することを恐れるため、最初に自分に来たオファーを承諾することが最適戦略となる。例えば企業 f_1, f_2, f_3 、求職者 w_1, w_2, w_3 について、選好順序が以下のように与えられているとする。

$$P_{f_1} = (w_1, w_3, w_2)$$

$$P_{f_2} = (w_3, w_2, w_1)$$

$$P_{f_3} = (w_3, w_2, w_1)$$

$$P_{w_1} = (f_3, f_2, f_1)$$

$$P_{w_2} = (f_2, f_1, f_3)$$

$$P_{w_3} = (f_1, f_3, f_2)$$

この時 W-optimal なマッチングは $\{(f_1, w_3), (f_2, w_2), (f_3, w_1)\}$, F-optimal なマッチングは $\{(f_1, w_1), (f_2, w_2), (f_3, w_3)\}$ となる。まず, f_1 が w_1 にオファーを出し, 続いて f_2 が w_3, f_3 が w_2 へそれぞれオファーを出したとする。各求職者は「もし自分が今のオファーを拒否すれば, 他の求職者がオファーを承諾していた場合, 自分だけマッチングに失敗してしまう。」と考える。よって全ての求職者は今自分に来ているオファーを承諾することが最適戦略となり, その結果, 解として $\{(f_1, w_1), (f_2, w_3), (f_3, w_2)\}$ が得られるが, これは (f_3, w_3) がブロッキングペアとなっており不安定なマッチングである。

2.2 遺伝的プログラミング

遺伝的プログラミング (以下 GP) とは John Koza[3] によって提案された進化的アルゴリズムであり, 主に関数同定 (入力データの集合と出力データの集合から双方の関係を表す関数を導出する手法) に用いられる。GP の最大の特徴は木構造で表現されたデータを扱うことである。例えば, 以下の数式 (1), (2) を GP で扱うための表現の一例として図 1 のようなものが考えられる。

$$x^3 + 2y \tag{1}$$

$$(x - 1) \div (y + 4) \tag{2}$$

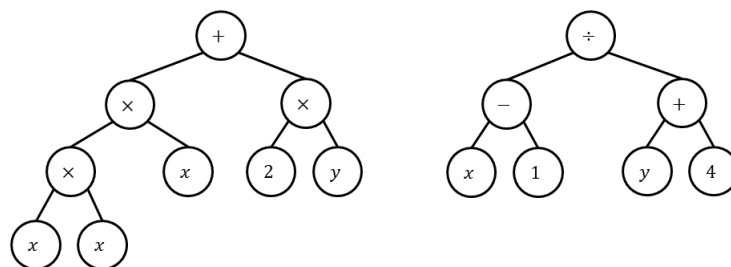


図 1: 木構造データの 1 例

一般的に以下のアルゴリズムを最適解が得られるか, または最大世代に到達するまで繰り返し行う。

2.2.1 初期集団生成

木構造で表現されたデータを無作為に多数生成し，初期集団とする．

2.2.2 評価

各データについて適合度を計算し，評価する．適合度は問題によって定義が異なる．

2.2.3 進化

各データを進化させ，現在の集団の中に含まれていない全く新しいデータを生成し，それらのデータを用いて次世代の集団を生成する．主な進化の手法として選択・交叉・突然変異があり，上記のフローを最適解が得られるか，または最大世代に到達するまで繰り返し行う．

選択 適合度の高いデータに対して，一定のペアを取り出し，そのまま次世代へ引き継ぐ．

交叉 各データから（適合度を元に）2つのペアを選び，その2つのデータから交叉点をそれぞれ1点ずつ無作為に選ぶ．その後その交叉点对应するそれぞれの部分木を交換する．これによって今までに無い全く新しいデータが生成される．図2は式(1),(2)に交叉を適用した場合の一例であり，赤印の部分が入れ替わっている部分である．

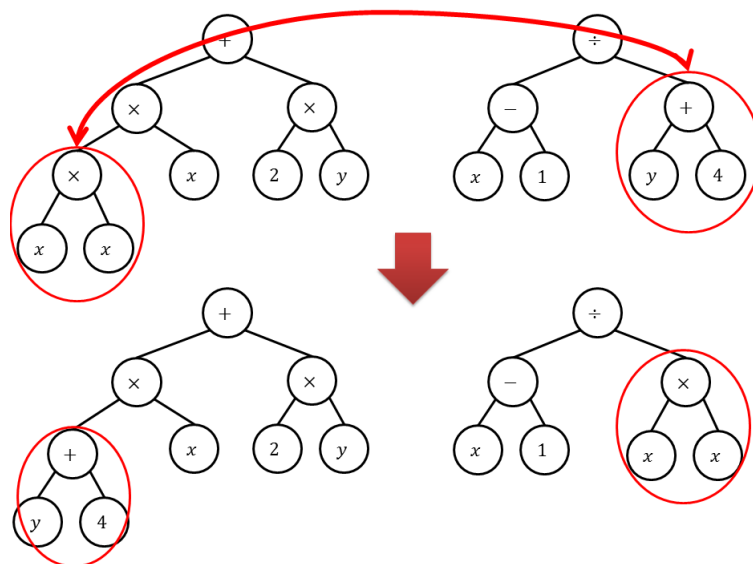


図2: 交叉

この場合，交叉によって

$$x(y + 4) + 2y \quad (3)$$

$$(x - 1) \div x^2 \quad (4)$$

という新たなデータが生成されることを表している．

突然変異 あるデータに対して，突然変異点を無作為に1点選び，その突然変異点に対応する部分木を別の部分木に変化させる．通常 GP では適合度の高いデータの近傍で収束していくため，局所最適解に陥る可能性がある．そこで，突然変異を適用することで近傍から離れたデータを生成し，これによって局所最適解に陥ることを防ぐことができる．図3は式(1)に突然変異を適用した場合の一例である．

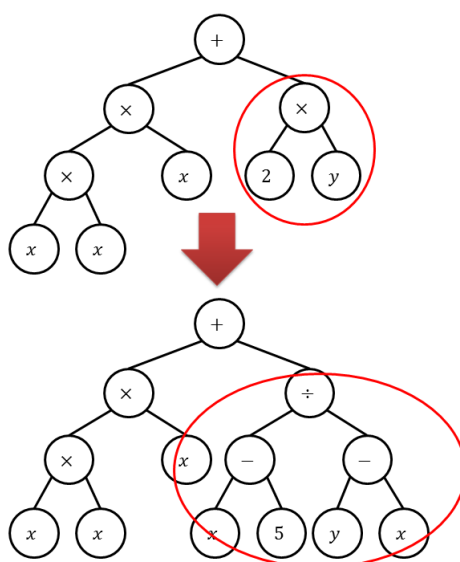


図3: 突然変異

この場合，突然変異によって

$$x^3 + \{(x - 5) \div (y - x)\} \quad (5)$$

という新たなデータが生成されことを表している．通常突然変異を適用すると，単にノードの値が変化するだけでなく，図3のように木が伸縮する可能性がある．

第3章 モデル化の手法

3.1 戦略の表現

戦略を GP で扱えるようにするために、まずは戦略を木構造で表現する手法について述べる。戦略とは四則演算等を用いた数式で表現するよりも、「もし A ならば B, そうでなければ C」のような文章、つまり構造をもった形式で表現するほうがよい。そこでまずは木構造を表現するためにノードを定義する。ノードは値、または関数、子ノードを持っており、関数を持つノードの値は、その子ノードを引数として関数に適用し、得られた値に等しい。次章で使用する各ノードを以下のように定義する。

定数ノード 定数ノードは 0 以上の整数の値をもっており、その値をノードの値とする。

変数ノード 変数ノードは-1 以上の整数で表されたある引数をもっており、その引数の値をノードの値とする。

真偽ノード 真偽ノードは真偽値を表わす整数をもっており、真ノードは 1, 偽ノードは-1 をノードの値とする。

> ノード > ノードは比較関数と 2 個の子ノード child1, child2 をもっており、child1 > child2 ならば 1, そうでなければ-1 をノードの値とする。図 4 の場合、> ノードは $x > 5$ ならば 1, $x \leq 5$ ならば-1 を返す

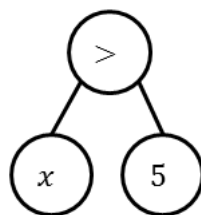


図 4: > ノード

IF ノード IF ノードは条件分岐関数と 3 個の子ノード child1, child2, child3 をもっており、child1 > 0 ならば child2, そうでなければ child3 の値をノードの値とする。図 5 の場合、IF ノードは $x > 0$ ならば 4, $x \leq 0$ ならば 7 を返す

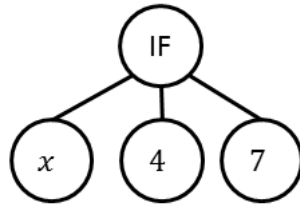


図 5: IF ノード

次に戦略ノードを以下のように定義する．

- 戦略ノードを 1 個以上のノードで表現された木構造データとする．
- 戦略ノードの値が 0 以下ならば偽，0 より大きければ真とする．

定義により，全ての戦略ノードは真か偽のいずれかの値をもつため，どのように交叉・突然変異を適用したとしてもその妥当性は失われない．

3.2 変数ノード

各変数ノードを以下のように設定する．

企業の変数ノード 企業の戦略ノードで用いられる変数ノードを以下に示す．実験の際に使用される変数ノードは実験の設定によって異なる．

RANK ノード 対象となる求職者の，選好順序における順位を引数とする．例えば，企業 f の選好順序が $P_f = (w_3, w_2, w_1)$ で与えられているとする．このとき， f が求職者 w_1 に対してオファーを出すかどうかを判断する場合，RANK ノードの値は 3 となる．同様に w_3 に対しては 1 となる．

POP ノード 対象となる求職者に来ているオファーの数を引数とする．例えば，企業 f が求職者 w に対してオファーを出すかどうかを判断する場合， w のもとに既に別の企業 f' と f'' からオファーが来ていれば，POP ノードの値は 2 となる．同様に w がどの企業からもオファーが来ていなければ 0 となる．このノードは企業が戦略的でない場合は使用しない．

求職者の変数ノード 求職者の戦略ノードで用いられる変数ノードを以下に示す．実験の際に使用される変数ノードは実験の設定によって異なる．

RANK ノード 企業の場合と全く同様に，対象となる企業の，選好順序における順位を引数とする．

OPT ノード 求職者にとって最適なマッチングにおける相手(と推測される)の企業の, 選好順序における順位を引数とする.

REM ノード 現在オファーを受けている企業よりも, 選好順序が上の企業からオファーが来る可能性があれば1, そうでなければ-1をノードの値とする. 例えば, 求職者 w の選好順序が $P_w = (f_1, f_2, f_3)$ で与えられているとする. 今, w の元に企業 f_3 からオファーが来て, そのオファーをどう処理するかを判断する場合, f_3 より選好順序が高い企業 f_1, f_2 がそれぞれ以下の条件を1つ以上満たしていればREMノードの値は-1となり, f_1, f_2 の中で以下の条件を1つも満たさない企業が少なくとも1つ存在する場合は1となる.

- 既に w 以外の別の求職者 w' に対してオファーを出していて, 且つまだ w' からオファーを拒否されていない
- 過去に w にオファーを出したが拒否された
- 既にマッチングを終了している

第4章 実験

4.1 準備

以下, おおまかな実験の流れについて説明する.

4.1.1 定義

説明の準備として, 各種用語・数式を以下のように定義する.

- 企業・求職者の集団をそれぞれ F, W とし, 各プレイヤー $v \in F \cup W$ の選好順序を P_v とする.
- 企業の戦略ノードを S_F とし, S_F の元である企業 $f_i \in F$ がある求職者 $w_j \in W$ にオファーを出すべきかどうかを $S_F[f_i, w_j]$ と表す. 全く同様に, 求職者の戦略ノード S_W の元で求職者 w_j が企業 f_i からのオファーを承諾すべきかどうかを $S_W[w_j, f_i]$ と表す.

4.1.2 初期化

まず始めに, 3つのプレイヤーセット $F_1 \cup W_1, F_2 \cup W_2, F_3 \cup W_3$ を生成する. マッチングはそれぞれのセットについて独立して行うものとする. 次に世代 t における戦略ノードの集団を $G(t)$ とする. 企業, 求職者についてそれぞれ500個の戦略ノード $S_F(k) (k = 1 \dots 500), S_W(k) (k = 1 \dots 500)$ を無作為に生成し, さらに

その戦略ノードの組 $Pair_{strategy}(k) = (S_F(k), S_W(k))$ を生成する. そのようにしてできた 500 組の戦略ノードの組の集合を初期集団 $G(0)$ とする.

4.1.3 マッチング

各プレイヤーセットについて, 2.1 節で仮定された条件の下で前節の戦略ノードを用いてマッチングを行う.

- 各企業 $f_i \in F_n$ は $S_F(k)$, 各求職者 $w_i \in W_n$ は $S_W(k)$ を用いて行動する.
- 与えられたインデックス番号に従って, 企業が $S_F(k)$ を用いてオファー先を決定していく (図 6:Offering stage)
 - 行動する企業を f とすると, f は残っている (まだマッチングを終了していないか, 過去に拒否されたことのない) 求職者の内, P_f 上で最も高い求職者 w_{i_1} にオファーを出すべきかどうかを $S_F(k)$ に問い合わせる.
 - $S_F(k)[f, w_{i_1}] = true$ なら w_{i_1} にオファーを出す.
 - $S_F(k)[f, w_{i_1}] = false$ なら次に選好の高い求職者 w_{i_2} にオファーを出すべきかどうかを $S_F(k)$ に問い合わせる. 全ての求職者 $w \in W_n$ について $S_F(k)[f, w] = false$ ならば, f はマッチングを終了する.

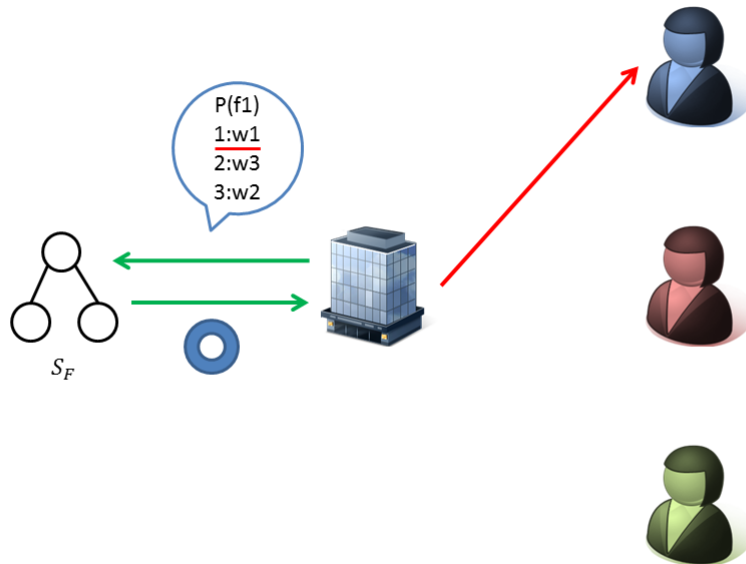


図 6: Offering stage における企業の行動

- 与えられたインデックス番号に従って, 求職者が $S_W(k)$ を用いて自分に来たオファーを処理していく. (図 7: Responding stage)
 - 行動する求職者を w とすると, w は自分に来た企業 $f \in F_n$ からのオファーの内, 承諾できるオファーがあるかどうかを $S_W(k)$ に問い合わせる.
 - $S_W(k)[w, f] = false$ なら f からのオファーを拒否する.
 - $S_W(k)[w, f] = true$ となった企業の内, P_w 上で最も高い企業 f_{max} からのオファーを承諾し, 残りのオファーを拒否する. w, f_{max} はマッチングを終了する.

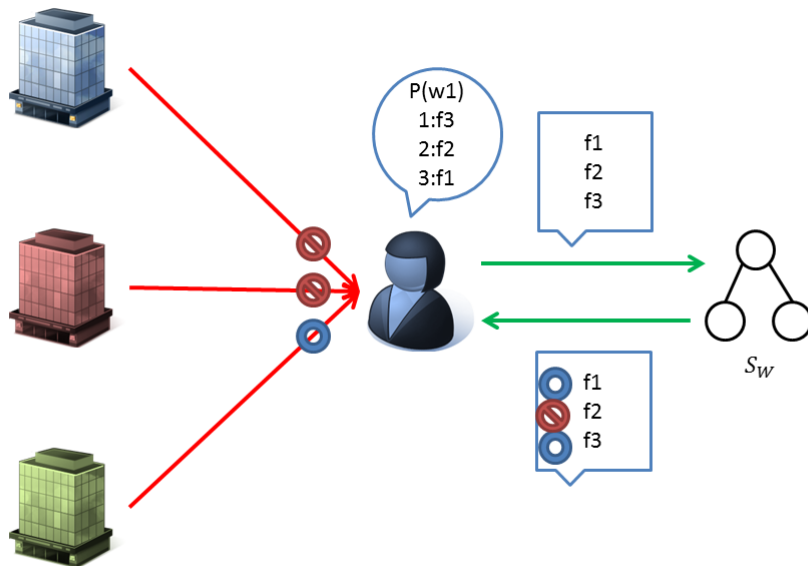


図 7: responding stage における求職者の行動

- 全ての企業が終了するまで上記の手順を繰り返す.

4.1.4 評価

プレイヤーセット $F_n \cup W_n$ に戦略ノードの組 $Pair_{strategy}(k)$ を用いてマッチングを行った結果得られたマッチングを $Matching(n, k)$ とし, $Matching(n, k)$ における各プレイヤー $v \in F_n \cup W_n$ のマッチング相手を $\mu(v)$ とする. さらに, v の選考順序 P_v に基づいた v の $\mu(v)$ に対する満足度を $sat_v(\mu(v))$ とする. 例えば,

求職者 w_1 の選考順序が $P_{w_1} = (f_1, f_2, f_3, f_4)$ で与えられている時, w_1 の f_1 に対する満足度は $sat_{w_1}(f_1) = 4$, 同様に f_3 に対する満足度は $sat_{w_1}(f_3) = 2$ となる .

この時 , $Matching(n, k)$ の評価値 $Eval_{matching(n, k)}$ を以下の式で定義する .

$$Eval_{matching(n, k)} = \sum_{i=1}^N sat_{w_i}(\mu(w_i)) - NN' - \alpha_F - \alpha_W \quad (6)$$

N はプレイヤーセット $F_n \cup W_n$ に含まれるプレイヤーの人数を表し , N' はその中で $Matching(n, k)$ における不安定なプレイヤー , すなわちブロッキングペアの相手が存在するプレイヤー人数を表している (よって $N' \leq N$ が成り立つ) . α_F, α_W はそれぞれマッチングの際に適用した戦略ノード $S_F(k), S_W(k)$ に含まれるノード数の合計が一定値を超えていれば 0.1 , そうでなければ 0 となる . 上式より , 各求職者 $w_i \in W_n$ の満足度の合計が高くで , かつ不安定なプレイヤーが少ないマッチングほど評価値が高く , 複数の戦略組でマッチングの評価値が同点となった場合はその中でノード数が少ない戦略を選択することになる .

最後に, $F_n \cup W_n$ における求職者にとって最適なマッチングを $W - optimal(n)$ とする . このとき戦略ノードの組 $Pair_{strategy}(k)$ の評価値 $Eval_{Pair_{strategy}(k)}$ を以下の式で定義する .

$$Eval_{Pair_{strategy}(k)} = \sum_{i=1}^n |Eval_{matching(i, k)} - Eval_{W-optimal(i)}| \quad (7)$$

n はプレイヤーセットのインデックス番号を表している (この実験では $n = 1, 2, 3$ となる) . すなわち, 全てのプレイヤーセットについて $Pair_{strategy}(k)$ を用いて得られたマッチングの評価値と最適解の評価値との差が 0 であるような $Pair_{strategy}(k)$ が最適戦略であるとみなすことができる .

4.1.5 進化

全ての戦略ノードの組 $Pair_{strategy}(k) \in G(t)$ について評価値 $Eval_{Pair_{strategy}(k)}$ を計算し, 適合度とする . 適合度を用いて以下の手順で戦略ノードを進化させ , 次世代の集団 $G(t + 1)$ を生成する .

選択 $Pair_{strategy}(k) \in G(t)$ の内, 適合度が最大, すなわち $Eval_{Pair_{strategy}(k)}$ が最小となる $Pair_{strategy}(k)$ を 2 組選択し, $G(t + 1)$ に加える .

交叉 $G(t)$ において上位 20 位以内にある戦略ノードの内 , 2 つの戦略ノードの組を無作為に選択し, それらに対して交叉を適用する . さらに交叉適用後

20%の確率で突然変異を適用させる．交叉を適用する確率を90%とし，交叉は S_F, S_W について独立に行うものとする．こうして得られた新たな戦略ノードの組を $G(t+1)$ に加える．

突然変異 $G(t)$ から1つの戦略ノードの組を無作為に選択し，それに突然変異を適用する．突然変異を適用する確率を10%とする．こうして得られた新たな戦略ノードを $G(t+1)$ に加える．

$Eval_{Pair_{strategy}(k)} = 0$ に収束するか，または $t = 500$ になるまで進化を繰り返す．

4.2 関連研究に基づいた実験

まず，2.1節で紹介した関連研究に基づき3つの場合について実験を行う．

4.2.1 企業が戦略的でなく，求職者が逐次手番の場合

企業が戦略的でない場合，各企業は自身の選考順序に従って，第1希望の求職者から順番にオファーを出していく．つまり，企業の戦略ノード S_F を「常に選考順序に従って第1希望から順番にオファーを出し続ける」という戦略に固定し，進化の処理において，求職者の戦略ノード S_W にのみ進化を適用する．

7種類のプレイヤーセットから無作為に3種類を選択し，それらのプレイヤーセットについて実験を20回行った．その結果いずれの場合においても以下の戦略が得られた．

S_W : 「W-optimal な相手以上ならば承諾する，そうでなければ拒否する」

一方関連研究 [1] において以下の定理が証明されている．

「定理：どのような戦略を用いても，他の求職者が W-optimal な相手より下の相手とマッチングしていない限り，W-optimal な相手より上の相手とはマッチングできない」

この定理によりどの求職者も，W-optimal な相手より上からのオファーを受けることができない．つまり上記の戦略は W-optimal な相手からのオファーが来たときに限り *true* となる．したがって上記の戦略は「W-optimal な相手のみを承諾する」という戦略に等しい．

4.2.2 企業が戦略的であり、求職者が逐次手番の場合

企業が戦略的である場合、企業は必ずしも自身の選考順序に従ってオファーを出すとは限らない（この実験では、あえて第2希望以下からオファーを出す、選考順序に関係なくオファーが集中している求職者を避ける、またはオファーを出すといった戦略があげられる）。

同様に、7種類のプレイヤーセットから無作為に3種類を選択し、それらのプレイヤーセットについて実験を20回行った。その結果いずれの場合においてもおよそ20回中16回の割合で以下の戦略（またはそれに準ずる戦略）組が得られた。

S_F : 「常に選考順序に従って第1希望から順番にオファーを出し続ける」

S_W : 「W-optimalな相手以上ならば承諾する、そうでなければ拒否する」

S_W については前節と全く同様である。一方 S_F は企業が戦略的でない場合と等しい。関連研究では「常に選考順序に従って第1希望から順番にオファーを出し続ける」という戦略が、企業にとって弱支配戦略である、すなわちそのような戦略をとることが企業にとって利得が上がる可能性がある戦略であると示されていた。ある固有のプレイヤーセットのみを用いればそのプレイヤーセット固有の戦略組が得られることも考えられるが、上記の戦略組はどのプレイヤーセットにも適用できる。

4.2.3 企業が戦略的であり、求職者が同時手番の場合

求職者が同時手番である場合、求職者は企業の行動は全て観察できるが、行動中に他の求職者がどう行動したかを知ることができない（全ての求職者が行動を終え、次のOffering stageが来た時に初めて他の求職者の行動結果を知ることができる）。つまり、 S_W に含まれる変数ノードはRANKノードとOPTノードのみでREMノードは含まれない（よりよい相手が残っているかどうかを知ることができない）。

同様に、7種類のプレイヤーセットから無作為に3種類を選択し、それらのプレイヤーセットについて実験を複数回行った。その結果いずれの場合においてもおよそ20回中18回の割合で以下の戦略（またはそれに準ずる戦略）組が得られた。

S_F : 「常に選考順序に従って第1希望から順番にオファーを出し続ける」

S_W :: 「W-optimal な相手以上ならば承諾する, そうでなければ拒否する」

これは前節と全く同様である。2.1節より, 各求職者は行動中に他の求職者の行動を知ることができないため, マッチングに失敗することを恐れて不安定な相手であっても承諾してしまうという理由から得られるマッチングが不安定になる可能性がある」と述べられていた。しかし, この実験では各プレイヤーはそれぞれ共通の戦略を用いて行動するため, 言い換えれば各求職者は他の求職者のとりうる行動を全て知っており, かつ他の行動を絶対にとらないことが保証されていることに等しい。故に厳密には前節とほぼ同じ状況であるといえる。

仮に各求職者がそれぞれ個別の戦略を用いれば, 関連論文における求職者の意思をより厳密に反映できるものと考えられる。

4.3 拡張したモデルでの実験

前節において, 関連研究・および実験の妥当性を検証できた。次に各プレイヤーが他のプレイヤーの選好順序を把握しておらず, 求職者間の情報共有によって推測された仮の選好順序を用いて自身にとっての最適な相手を予測しながら行動していくモデルについて実験を行う。以下,

1. 企業が戦略的でなく, 求職者が逐次手番の場合を case1
 2. 企業が戦略的であり, 求職者が逐次手番の場合を case2
 3. 企業が戦略的であり, 求職者が同時手番の場合を case3
- と表す。

4.3.1 予測

拡張モデルにおける予測の手法について述べる。予測は各求職者間で独立して行うものとする。以下予測をする時点でマッチングを終了していない企業・求職者の集合をそれぞれ \bar{F}, \bar{W} とする。

各求職者 $w_i \in \bar{W}$ はまず各企業 $f_i \in \bar{F}$ の選好順序を, 企業がオファーを出した順序に従い, その企業が先にオファーを出した求職者が選好順序の上で高いと仮定し, 予測する。企業 f_i の仮選好順序を P'_{f_i} とすると, f_i が最初の Offering stage で求職者 w_{i_1} にオファーを出し, 次の stage で求職者 w_{i_2} にオファーを出

し、さらに次の stage で求職者 w_{i_3} にオファーを出した場合、

$$P'_{f_i} = (w_{i_1}, w_{i_2}, w_{i_3}) \quad (8)$$

となる。尚、まだオファーを出していない相手の選好順序は無作為に決定するものとする。

次に各求職者の選好順序が業種別に分布していると仮定し、その分布から他の求職者の選好順序を予測する。求職者 w_i の仮選好順序を P'_{w_i} 、業種別の選好順序を $P_{w_i}^T$ とすると、企業 $f_{i_1}, f_{i_2}, f_{i_3}$ が業種 A、 $f_{j_1}, f_{j_2}, f_{j_3}$ が業種 B、 $f_{k_1}, f_{k_2}, f_{k_3}$ が業種 C にそれぞれ属しており、かつ $P_{w_i}^T = (B, A, C)$ ならば、例として

$$P'_{w_i} = \begin{cases} (f_{j_1}, f_{j_2}, f_{j_3}, f_{i_1}, f_{i_3}, f_{i_2}, f_{k_3}, f_{k_1}, f_{k_2}) \\ (f_{j_3}, f_{j_1}, f_{j_2}, f_{i_1}, f_{i_2}, f_{i_3}, f_{k_3}, f_{k_2}, f_{k_1}) \\ (f_{j_2}, f_{j_3}, f_{j_1}, f_{i_1}, f_{i_2}, f_{i_3}, f_{k_1}, f_{k_2}, f_{k_3}) \end{cases} \quad (9)$$

など、合計 216 通りの選好順序が考えられる。これらの組の中から無作為に 1 つの選好順序を選び、 w_i の仮選好順序とする。各求職者 w_i は $P_{w_i}, P'_{f_i}, P'_{w_j} (f_i \in \bar{F}, w_j \in \bar{W}/\{w_i\})$ に対し、DA アルゴリズムを適用した際に得られるマッチングの下での w_i のマッチング相手 $\mu'(w_i)$ を予測結果とする。そして、 P_{w_i} における $\mu'(w_i)$ の順位を OPT ノードの引数とする。

予測を行うタイミングは求職者の行動が逐次手番か同時手番によって異なる。逐次手番の場合 Responding Stage において、自分が行動する度に予測を行う。同時手番の場合 Responding Stage の開始時に全ての求職者が同時に予測を行う。

4.3.2 結果

関連研究に基づいた実験と同様の条件の下、拡張したモデルで企業・求職者の行動の種類別に実験を行った。実験には以下に設定されたパラメーターをもつプレイヤーセットを 3 つずつ生成しそれらを用いた。

- (a) 企業数：6 求職者数：6 業種数：3 1つの業種当りに属する企業数：2
- (b) 企業数：6 求職者数：6 業種数：2 1つの業種当りに属する企業数：3
- (c) 企業数：9 求職者数：9 業種数：4 1つの業種当りに属する企業数：2~3

- (d) 企業数：9 求職者数：9 業種数：3 1つの業種あたりに属する企業数：3
- (e) 企業数：9 求職者数：9 業種数：2 1つの業種あたりに属する企業数：4~5
- (f) 企業数：12 求職者数：12 業種数：6 1つの業種あたりに属する企業数：2
- (g) 企業数：12 求職者数：12 業種数：4 1つの業種あたりに属する企業数：3
- (h) 企業数：12 求職者数：12 業種数：3 1つの業種あたりに属する企業数：4
- (i) 企業数：12 求職者数：12 業種数：2 1つの業種あたりに属する企業数：6

その結果，各パラメーターごとに case1, case2, case3 のいずれの場合においても，

1. 関連研究で示されていた最適戦略
2. 「基本的に上位を目指す，上位とのマッチングが不可能であれば妥協する」という戦略
3. 「予測順位に関係なく，常に最初に来たオファーを承諾する」という戦略が解として得られた．以下に代表的な解および特徴を示す．
 - 1つの業種あたりに属する企業数が2，すなわち企業の業種が最も細分化されている (a)(c)(f) ではいずれの場合もほぼ 100 % の確率で収束し，最適戦略が得られた．
 - (b)

S_F : 「常に選考順序に従って第1希望から順番にオファーを出し続ける」

S_W : 「第1希望，または W-optimal な相手の順位を2位以下と予測した場合に承諾する」

また，希に収束し，最適解が現れる。

また， S_F, S_W を用いるとある1つのプレイヤーセットのみ W-optimal なマッチングが得られ，残りの2つのプレイヤーセットは不安定なマッチングが得られる．
 - (d)

S_F : 「常に選考順序に従って第1希望から順番にオファーを出し続ける」

S_W : 「第2希望以上，または第3希望以下でかつ W-optimal な相手の順位

を3位以下と予測した場合に承諾する。」

また、希に最適戦略が現れるが、進化の最中に評価値 $Eval_{Pair_{strategy}(k)}$ が0に収束しない。

また、 S_F, S_W を用いるとある1つのプレイヤーセットのみ W-optimal なマッチングが得られ、残りの2つのプレイヤーセットは不安定なマッチングが得られる。

- (i)

S_F : 「常に選考順序に従って第1希望から順番にオファーを出し続ける」

S_W : 「常に最初に来たオファーを承諾する」

また、いずれのプレイヤーセットでも、

case1, case2, case3 の順にマッチングの評価 $Eval_{matching}$ が低くなる傾向がある。

第5章 考察

最適解は得られるものの、進化中に $Eval_{Pair_{strategy}(k)} = 0$ にならない場合、これは単に予測が外れていることによるものと思われる。業種数: N_T , 1つの業種当たりには属する企業数: N_F の場合、マッチングにおいて一番最初に行動する求職者 w_1 が予測を行う時、他の求職者の仮選好順序 P'_W の候補数は

$$(N_F!)^{N_T} \quad (10)$$

で表される。従って、仮選好順序の候補数は、業種数に比例して増加し、1つの企業当たりには属する企業数に比例して爆発的に増加する。故に人数が多い、または1つの業種当たりには属する企業数が多いプレイヤーセットでは予測の精度が落ちるため、結果 $Eval_{Pair_{strategy}(k)} = 0$ になる確率が極めて低いと考えられる。

最適解が得られない場合、いずれのプレイヤーセットでも、ある特定のプレイヤーセットに対してのみ W-optimal なマッチングを与える戦略 S'_F, S'_W が得られる。

4.14節の式(7)より、1つのプレイヤーセットのマッチングの評価値 $Eval_{matching}$ が0になると全体の評価値 $Eval_{Pair_{strategy}}$ に大きな影響を及ぼす(特に人数が多いプレイヤーセットほど効用が高い)。故に、ある戦略組によって1つでも W-optimal なマッチングに近づくとそれだけであたかも全体の評価が高いかのように評価してしまっている可能性が高いと考えられる。また、進化を適用するのが上位の戦略ノードのみであり、戦略ノードを構成するノードの種類によ

では、最適戦略ノードが交叉・突然変異によって生成される確率が低くなってしまふ場合がある。また、仮に最適戦略ノードが生成されたとしても、予測が外れてしまうとすぐに淘汰されてしまい、最適戦略ノードが次世代に残る確率が低い。以上より高確率で局所最適解に陥っているものと思われる。

例えば、あるプレイヤーセットが以下のように選好順序が企業・求職者共に全て同じである場合について考える。

$$\begin{aligned}
 P_{f_1} &= (w_3, w_2, w_1) \\
 P_{f_2} &= (w_3, w_2, w_1) \\
 P_{f_3} &= (w_3, w_2, w_1) \\
 P_{w_1} &= (f_1, f_2, f_3) \\
 P_{w_2} &= (f_1, f_2, f_3) \\
 P_{w_3} &= (f_1, f_2, f_3)
 \end{aligned} \tag{11}$$

この時 W-optimal なマッチングは $\{(f_1, w_3), (f_2, w_2), (f_3, f_1)\}$ となる。もし企業が「常に選好順序に従って第 1 希望から順番にオファーを出す」という戦略を用いた場合、最初の Responding stage において行動するのは w_3 のみである。この時 w_3 は f_1 からのオファーを承諾してマッチングを終了する。同様に次の Responding stage では w_2 、その次の Responding stage では w_1 のみが行動し、 w_1, w_2 共に常に来たオファーの中で承諾できる（承諾せざるを得ない）オファーが必ず存在する。従って、各求職者は予測をしなくても「常に最初に来たオファーを承諾する」という戦略を用いれば、必ず W-optimal なマッチングが得られる。この戦略を表す戦略ノードを S'_W とすれば、全ての企業 f 、求職者 w において常に $S_W[w, f] = true$ となるので、例えば図 8 のような真ノード 1 つのみ、または 0 より大きい定数ノード 1 つのみで構成されているものが考えられる。

S'_W を使用するとこのプレイヤーセットにおけるマッチングの評価は

$$Eval_{matching} = Eval_{W-optimal} \tag{12}$$

となり、式 (7) における $Eval_{Pair_{strategy}}$ の値を大幅に減少させる。その結果全体の評価値が高いとみなされ、戦略ノードの集団 $G(t)$ における、 S'_W の順位が高くなる可能性が高い。もしそうなれば、 S'_W が交叉・突然変異適用の対象になる頻度が高くなるが、図 8 の S'_W から交叉によって最適戦略ノード S_W^{opt} (図 9) が



図 8: 戦略ノード S'_W の例

生成される可能性は直感的に低いと推測できる（ただし突然変異を適用すれば生成される可能性はある）。

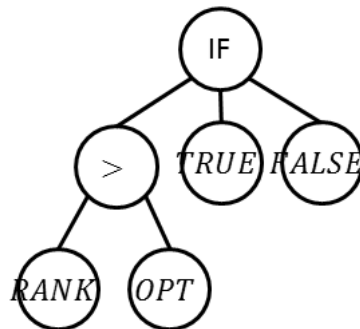


図 9: 最適戦略ノード S_W^{opt} の例

また、進化の過程で S_W^{opt} が生成されたとしても、含まれるノード数の合計・予測の精度による解の不安定性により、 S_W^{opt} が S'_W よりも上位にくる可能性は極めて低い。その結果、 S'_W は最後まで残り続け、最終的に局所最適解として現れる。

考察が正しいかどうかを検証するために、以下のように設定を変更し、再び 4.3 節の実験を行った。

- 進化処理においても S_W^{opt} が $G(t)$ から淘汰された場合、 $G(t+1)$ の上位 3 位の位置に S_W^{opt} を加える。
- 実験毎に使用する 3 つのプレイヤーセットの
 - 企業数・求職者数
 - 業種数または 1 つの業種あたりに属する企業数を固定し、選好順序のみを変えたものを使用する。

- 進化の過程で S_W^{opt} の $G(t)$ における順位の推移を観察する． S_W^{opt} が $G(t)$ 中に現れない場合は最下位として扱う．

今回の実験で以下のパラメーターを持つプレイヤーセットをそれぞれ3つずつ生成し，使用する．

1. 企業数：9 求職者数：9 業種数：4 1つの業種あたりに属する企業数：2～3
2. 企業数：12 求職者数：12 業種数：6 1つの業種あたりに属する企業数：2
3. 企業数：9 求職者数：9 業種数：2 1つの業種あたりに属する企業数：4～5
4. 企業数：12 求職者数：12 業種数：2 1つの業種あたりに属する企業数：6

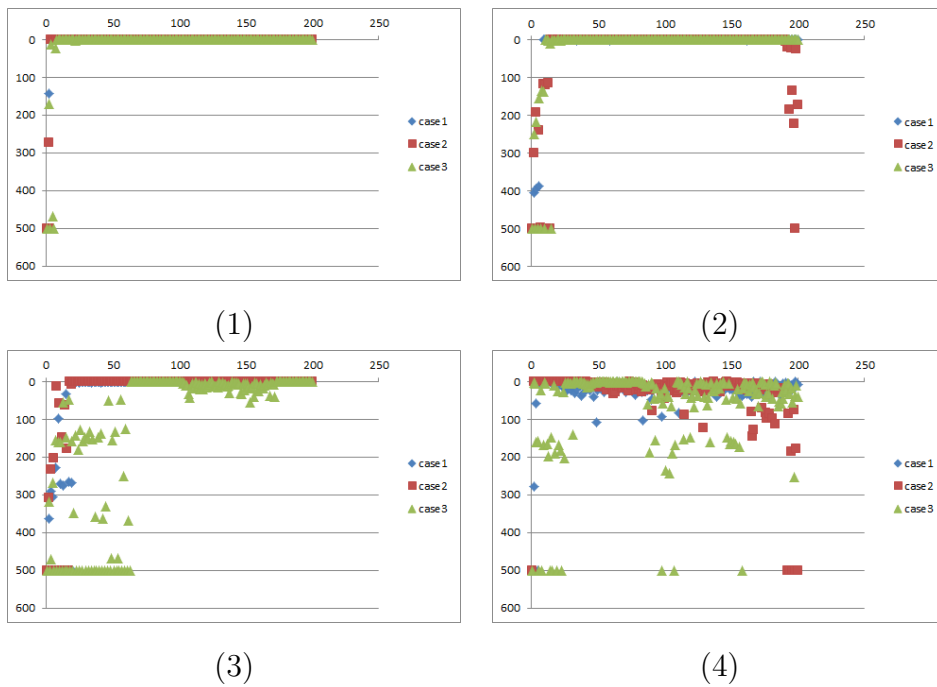


図 10: 各プレイヤーセットにおける S_W^{opt} の分布

図 10 は各プレイヤーセット，企業・求職者の戦略ごとの S_W^{opt} の推移を表している¹⁾．横軸が世代 t ，縦軸が戦略ノードの集団 $G(t)$ における S_W^{opt} の順位を表

¹⁾ これらの図はあくまで 1 例であり，実験を行うたびに様相が大きく変化する．これは進化・予測の処理が完全に無作為であるためである．

している．また，各グラフのマークはそれぞれ以下のように対応している．

case1(青の四角印) 企業が戦略的でなく，求職者が逐次手番の場合

case2(赤の四角印) 企業が戦略的であり，求職者が逐次手番の場合

case3(緑の三角印) 企業が戦略的であり，求職者が同時手番の場合

1つの業種当たりに属する企業数が2，すなわち業種が細かく分類されている場合，各プレイヤーセット 1,3 における仮選好順序 P'_W の候補数はそれぞれ式 (10) より 48,64 となる．一方業種数が2，すなわち業種が2つにしか分類されていない場合，同様に各プレイヤーセット 2,4 における P'_W の候補数はそれぞれ 2880,518400 となる．図よりいずれのプレイヤーセット，戦略行動においても S_W^{opt} が $G(t)$ においてほとんど上位に分布しており，優良な戦略であることが分かる．ただし， P'_W の候補数に比例して予測の精度が悪くなり，それに比例して S_W^{opt} の順位の分散も大きくなる．また，case1,case2 がほぼ同じ分布であるのに対し，case3 の分布はプレイヤーセットによって精度が異なる．なお全体の評価値 $Eval_{Pair, strategy}$ は case1,case2,case3 の順に低くなる傾向がある（各プレイヤーセットの評価値の合計を $S = \sum_{i=1}^N Eval_{W-optimal(i)}$ とすると，case1,case2 の場合では S の 0 50 % の値，case3 の場合では S の 30 50 % の値の間で振動する．）

case1 の場合，各企業は自身にとって最も選好順序の高い求職者から順にオファーを出すため，企業の仮選好順序 P'_F は企業の真の仮選好順序 P_F を正確に反映している．従って， S_W^{opt} の順位は予測において P'_W が P_W を正確に反映しているかどうかのみ依存していると考えられる．

case2 の場合，各企業は選好順序に従ってオファーを出すとは限らないため， P'_F を正確に反映できない場合がある．よって，予測の精度は case1 よりも低く，そのため局所最適解にも陥りやすい．ただし，case1 と同様に第1希望から順番にオファーを出す戦略を用いた場合，その予測の精度は case1 の場合と等しい．従って，もし局所最適解に陥らなければ，他の企業の戦略は淘汰される確率が高いため case1 とほぼ等程度か，またはそれ以上の確率で S_W^{opt} が上位に来ると考えられる．

case3 の場合，case2 と全く同様である．ただし，case3 は他の場合と比べて予測の精度が最も低いため，他の場合に比べるとプレイヤーセットのサイズの影響が大きいと考えられる．

第6章 まとめ

本研究により，GP でジョブマッチング市場を扱うための表現手法の1つが提案され，強い仮定が緩和された場合での議論が可能となった．しかし本研究のGP システムには局所最適解に陥った時にそこから抜けだしにくいという問題もあり，評価値の表現，交叉，突然変異を適用する確率など，より適正なパラメーターを模索する必要があると考えられる．また，残っているプレイヤーの人数，オファーを断った回数など，変数ノードの種類も増やす余地があると考えられる．

分散型ジョブマッチング市場において，各求職者が情報共有によって他の求職者の選好順序を予測し，それを元に自分にとって最適なマッチング相手を推測しながら行動する場合，得られた戦略が最適戦略かどうかという確証こそ得られなかったものの，予測精度が高い場合は関連研究とほぼ等しい戦略が得られ，予測精度が低い場合は「上位とマッチングできる可能性がある内は上位を目指す，上位とのマッチングが不可能であれば妥協をする」という戦略が得られ，特に予測精度が著しく低い場合は「予測順位に関係なく，常に最初に来たオファーを承諾する」という戦略が得られた．今後戦略が切り替わる条件やこれらの戦略によって得られるマッチングの質を検証するために，プレイヤーセットの種類を増やして検証していく必要がある．

謝辞

本研究を行うにあたり，熱心なご指導，ご助言を賜りました松原繁夫准教授に厚くお礼申し上げます．また，お忙しい中様々な御助言，御協力を頂きました石田・松原研究室の皆様から感謝致します．

参考文献

- [1] Guillaume, Haeringer, and Myrna Wooders.: Decentralized job matching. Int J Game Theory (2011) 40:128 DOI 10.1007/s00182-009-0218-x
- [2] 竹本 浩.: 集合知/遺伝的プログラミング. <http://www.pwv.co.jp/~take/TakeWiki/index.php?集合知%2F遺伝的プログラミング> (2012/11/20)
- [3] John R. Koza,: Genetic programming as a means for programming com-

puters by natural selection. *Statistics and Computing*. June 1994, Volume 4, Issue 2, pp 87-112

- [4] 伊庭 斉志.:Cによる探索プログラミング : 基礎から遺伝的アルゴリズムまで.(2008.9)
- [5] 森 直樹.:javaで学ぶ遺伝的アルゴリズム.(2007.11.25)
- [6] 梶井 厚志. 松井 彰彦:ミクロ経済学 戦略的アプローチ.(2009.2.10)
- [7] Osborne MJ,Rubinstein A(1994)A course in game theory. MIT Press,Cambridge