

分散遺伝的アルゴリズムとローカルサーチを併用した 大学の時間割作成システム

鄭 信雨, 平井 敏之, 村尾 了, 新妻 真輔, 品野 勇治

東京農工大学

本研究は平成 20 年度後期, 東京農工大学情報工学科 3 年生の講義内での研究・開発をまとめたものであり, 大学を対象とした時間割作成問題について述べる。時間割作成問題には, 教員の担当可能時間や講義の重複禁止などの制約条件が数多く存在する。したがって単純な GA を応用した解法の場合, 実行可能な時間割を得ることは困難である。そこで本稿では, 分散遺伝的アルゴリズムとローカルサーチを組み込んだ Memetic Algorithm を提案し, 東京農工大学情報工学科の時間割をもとに数値実験を行い結果を考察する。また各パラメータの設定と時間割の作成, 結果の表示を行う時間割作成システムの開発も行った。

Development of The Timetabling System for Universities with DGA and Local Search

Jung Sinyu, Toshiyuki Hirai, Ryo Murao, Shinsuke Niizuma, Yuji Shinano

Tokyo University of Agriculture and Technology

This study is motivated by the lecture for 3rd grade students of *Tokyo university of Agriculture and Technology, Department of computer and information sciences*. In this paper, we discuss the timetabling problem of universities. Because of many constraints, coming from working time of teachers the prohibition of duplication of lectures and so on, it is difficult to find a feasible solution with simple genetic algorithms. Therefore we propose the memetic algorithm that uses Distributed Genetic Algorithm and local search. Through the computational simulation with actual data of our department, we evaluate the effect of our method. Additionally we develop a timetabling system that can setting some parameters, solving and display obtained timetable.

1 はじめに

本研究は平成 20 年度後期, 東京農工大学情報工学科 3 年生対象の「システム製作実験」内で研究・開発したものである。並列計算またはアルゴリズムの可視化をテーマに学生が調査, 設計, 開発を行った。

研究主題は大学を対象とした時間割作成問題である。時間割作成問題には, 教員の勤務可能時間や講義の重複禁止などの制約条件が数多く存在する。時間割作成問題の目的は, それらすべての条件を満たした上で, 時間割の良さの尺度として設計した評価値が高くなるように講義, 教員, 教室の配置を決定することである。ただし, 本稿では時間割を 1 つの学科に限定するため実験など特殊な教室を使用する講義以外の教室の割当は考慮しない。時間割作成問題は, 教員数, 講義数によって探索空間が指数的に増加するため, 様々な研究がなされており, メタヒューリスティック解法が効果的であることが知られている [1]-[9]。

遺伝的アルゴリズム (以下 GA) は生物の進化の過程にアナロジーを持つメタヒューリスティック解法であるが, 時間割作成問題へ適応した場合, 交叉等の遺伝子操作の際に講義が消失したり, 重複するなど制約に違反する可能性が高い。これを解決するため様々な方法が議論されている [7]-[9]。一般的な時間割作成問題への GA の適応例には, 不当に削除された講義の追加や重複する講義の

削除を行う Filtering 操作を実装した解法 [8], あらかじめ制約の厳しい講義を配置した後に, 残りの講義を配置するハイブリッド探索手法 [7], 講義の時間割への配置後に教室の割り当てを行う分割 GA [9] 等がある。

本稿では上記の問題点を踏まえた上で, より大域的な探索を可能にする Memetic Algorithm を提案する。提案手法は大域探索に分散遺伝的アルゴリズム [10] (以下 DGA) を実装し, 遺伝子操作による制約違反に対しては, ローカルサーチを組み込むことで解決する。また, データ入力やパラメータ設定, 時間割の作成, 結果の出力を行う時間割作成システムも開発した。

本論文は, 以下の順序で構成されている。第 2 章では本研究で対象とする時間割作成問題のモデルを定義し, 入出力や制約条件, 時間割の評価について解説する。第 3 章では提案手法, 第 4 章では開発した時間割作成システムについて述べる。第 5 章では, 平成 20 年度前期の東京農工大学情報工学科の時間割をもとに実験を行い, 結果を考察する。最後に第 6 章でまとめと今後の課題を述べる。

2 時間割モデル

本章では対象とする時間割作成問題のモデルを定義し, 制約条件や評価関数について解説する。まず以下に本研究で対象とする時間割モデルを示す。

- 1 日の講義時限数は 5 時限

- 1週間の講義日数は5日
- 学年は4つ
- 同一講義の1週間の講義回数は1回
- 実験は専用の端末室で行う
- 各教員の担当講義は決定済み

本研究で扱う時間割作成問題の入力データは、教員データと講義データから成る。

教員データ：

氏名：教員名
担当希望時間：3段階(0可能, 1不明, 2不可能)*1

講義データ：

講義名：講義の名前
対象学年：1年...4年
区分：必修, 選択...
単位コマ数
教室：実験のみ記入
担当教員：講義を担当する教員名

また、本研究で考慮する制約条件は以下の通り。

- C1 教員の勤務不可能時間には講義を割り当てない
- C2 同じ教員が同じ時間に2つ以上の講義を行わない
- C3 同じ時間に同じ実験室は使用不可能
- C4 ある学年の授業はその学年の時間割に存在

作成した時間割は以下の項目と値から評価する。

- E1 教員の講義希望時間が実現：+50
- E2 各教員の1日の授業時間が2回以上：分散値*2
- E3 各学年の必修科目が同時に開講されている：-10
- E4 講義が連続していない：-150
- E5 必修科目が連続している：連続回数の指数倍*3
- E6 1限に講義がある：-20
- E7 5限に講義がある：-150

すべての講義を配置した時間割が時間割作成問題における出力となる。上記の制約条件をすべて満たした上で、評価値の高い時間割を作成することが目的である。

3 提案手法

本章では大学の時間割作成問題の解法として提案するDGAとローカルサーチを組み合わせたMemetic Algorithmについて解説する。3.1節では提案手法の概要を、3.2節ではローカルサーチ、3.3節では各プロセッサで動作するGAについて解説する。なおDGAの詳細については参考文献[10]を参照されたい。

3.1 提案手法の概要

DGAは、大きな母集団をより小さなサブ母集団に分割し、各サブ母集団内で従来のGAの処理を並列に行う。また、各サブ母集団間では移住操作を行う。(図1参照)これは、あるサブ母集団内の個体を別のサブ母集団に送る操作である。このため、移住間隔と移住率というパラメータを導入する。移住間隔は、移住から次の移住までの世代数であり、移住率は、サブ母集団に属する個体全体のうち、移住する個体の割合を示す。

このDGAにローカルサーチを組み込んだ提案手法の概要を以下に示す。

- 1. ランダムに個体数×プロセッサ数分の時間割を作成

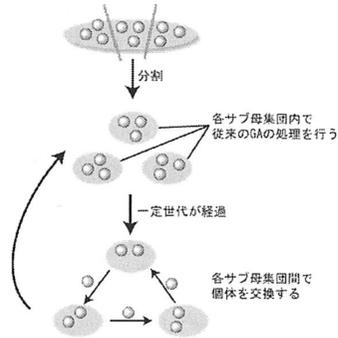


図1 DGA

- 2. ローカルサーチを用いて実行可能な母集団を生成
- 3. 母集団を分割し、各プロセッサに送信
- 4. 各プロセッサ内でGAを適用
- 5. 各プロセッサ内で設定確率に基づきローカルサーチを適応
- 6. 一定世代でプロセッサ間の個体交換
- 7. 終了条件(世代数, 評価値)ならば終了, それ以外は4へ

なお、並列化の際にはRubyのライブラリである、dRuby[11]を用いた。

3.2 ローカルサーチ

ローカルサーチの役割は、実行可能化と評価関数値の向上の2つである。これを可能にするため以下のように近傍を定義する。

- Swap: 2つの講義を交換
- Move: 講義と空き時間を交換

実行可能化は制約違反している講義について、違反がなくなるまでSwapとMoveを繰り返すことで可能になる。

評価関数値の向上は、曜日をまたいだSwap、Moveを考慮すると近傍が膨大になり、評価にも時間がかかるため扱わない。同一曜日内でのSwapは、E1, E3, E5のみを考慮すればよく、MoveではE4, E6, E7の考慮すればよいので、迅速な探索が可能となる。

3.3 各プロセッサで動作するGA

個体 個体は講義数の遺伝子を持ち、各遺伝子は

$$5 \times (\text{曜日} - 1) + (\text{時限} - 1)$$

(0:月1限, 1:月2限, ..., 24:金5限)の値をとる。

評価 評価は第2章時間割モデルで示したものをを用いる。

選択 選択には適応度によるルーレット選択を用いる。

交叉 交叉には2点交叉法を用いる。

突然変異 ある確率で個体内の遺伝子一つを変化

終了条件 あらかじめ定めた世代数, 評価値になり次第終了する。

4 時間割作成システム

本章では、作成した時間割作成システムについて述べる。開発言語にはRubyのライブラリであるVisualu-Ruby(仮称)[12]を用いた。メイン画面を図2に示す。

*1 5日間5時限分

*2 評価値：教員の持つ講義の分散値

*3 評価値：3連続回数

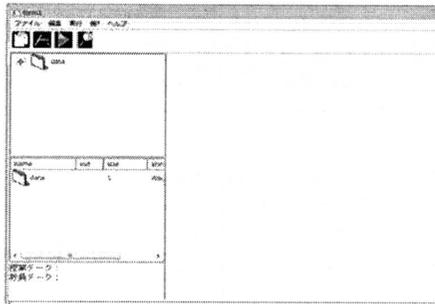


図2 メイン画面

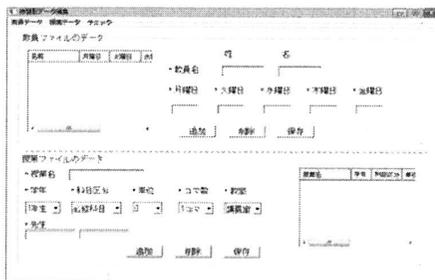


図3 データ編集画面

本システムによる時間割作成は以下の手順からなる。

1. データ作成
2. 前処理
3. 時間割作成
4. 時間割の保存、表示

各処理ステップは以降の節で順に解説する。

4.1 データ作成

データは講義データと教員データからなる。編集画面を図3に示す。子の画面は図2の画面上でプルダウンメニューの編集を選択するなどによって表示される講義データ、教員データともに入力後はそれぞれ名前を付けて保存することができる。

4.2 前処理

前処理は、データの整合性検査と事前講義固定からなる。データ整合性とは講義データに記載された教員のデータが存在するかをチェックするものである。この処理は図3のメニューにあるチェックから行う。事前講義固定とは、すでに開講時限が確定済みの講義を事前に固定する処理のことである。この処理を行うためには図3の画面の左上部ツリーリストから読み込む。データの読み込み完了後図4の画面が表示され、この画面上でユーザが直接時間割編集を行う。固定は、右画面より講義を選択し、クリックにより行う。

4.3 時間割作成

時間割は、時間割の読み込み、整合性チェック、事前講義固定を行った後実行ボタンをクリックすることで作成される。実行は図2の実行アイコン等から行い、計算機が時間割の作成を開始する。実行完了後、作成した時間割が表示される。

4.4 時間割の表示、保存

時間割の保存は図2のメニュー内の「保存」から行う。保存した時間割の表示は同じく図2のメニュー内の「開



図4 事前講義固定

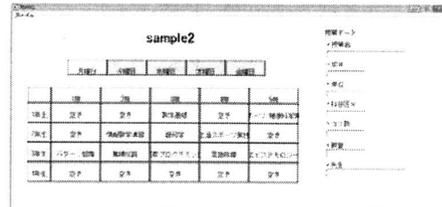


図5 時間割表示例

く」から行う。時間割の表示例を図5として示す。ここでは時間割を曜日ごとに表示し、固定された講義をクリックすることで、講義の詳細を見ることが可能である。

5 数値実験

本章では、5.1節でパラメータチューニング、5.2節では複数台による時間割作成実験の順に解説する。

5.1 パラメータチューニング

実験環境は Intel Pentium 4, 1GB RAM である。考慮するパラメータは、各プロセッサ内の個体数、世代数、突然変異率、移住率、移住間隔、プロセッサ数である。ここでは、デフォルトとして、各パラメータを以下のように固定し、

- 個体数 20
- 世代数 10000
- 突然変異率 0.1
- 移住率 0.1
- 移住間隔 10000 世代
- プロセッサ数 1

一つ一つについて変更した際の評価関数値を7回の実験の平均値により比較した。結果からパラメータは以下のように設定した。

- 個体数 20
- 世代数 20000
- 突然変異率 0.1
- 移住率 0.1
- 移住間隔 10 世代
- プロセッサ数 20

5.2 複数台による時間割作成実験

ここでは20台のプロセッサを用いて時間割の作成を試みた。ただし、教員の担当希望は入手できなかったため、実行可能性を考慮しながらランダムに生成した。また、終了条件は世代数が20000、もしくは評価値が5000とした。得られた時間割を図6に示す。作成した時間割は、制約違反がなく、評価値は4790点であった。評価について結果を考察すると、講義が連続している、1, 5限の講義が少ないことから、少なくともE4,E6,E7の評価項目は満たされていることがわかる。5限よりも1限に

	月	火	水	木	金	
1年	1限	-	Writing Basics	-	-	
	2限	Reading Basics	スポーツ・健康科学理論	情報化社会と職業	プログラミング序論	微分積分学Ⅰ・演習
	3限	科学	基礎ゼミ	プログラミング序論	コンピュータ序論	コンピュータ序論
	4限	線形代数学Ⅰ	初級第二外国語	物理学基礎	数学基礎	スポーツ・健康科学実技
	5限	-	-	-	-	-
2年	1限	Advanced Writing	-	-	-	
	2限	情報理論演習	-	幾何学	生涯スポーツ実技	
	3限	情報理論	情報数学演習	関数論	電磁気学Ⅰ	中級第二外国語
	4限	微分方程式	計算機アーキテクチャ基礎	計算機アーキテクチャ演習	Advanced Reading	情報数学
	5限	-	-	-	-	-
3年	1限	-	実験A	-	-	
	2限	Object指向Program	実験A	関数プログラミング	集積回路	Computer Graphics
	3限	情報社会文化論	実験A	人工知能	画像工学	Epistemology
	4限	パターン認識	Operating System	-	言語処理	Human Interface
	5限	-	-	-	-	-
4年	1限	-	-	-	-	
	2限	論文・文献講読	-	-	-	
	3限	-	-	ソフトウェア工学	-	-
	4限	-	-	-	-	-
	5限	-	-	-	-	-

図6 作成した時間割

講義が集中している事は、評価値の設定に沿っているので妥当であるといえる。

6 おわりに

本稿では、大学における時間割作成問題をモデル化し、問題点を理解した上で Memetic Algorithm を提案した。提案手法は東京農工大学情報工学科の時間割を用いた数値実験から妥当な解を導出している。また、時間割の作成に関わる一連の処理が可能なシステムの開発も行った。記載できない部分は発表の際にデモで紹介する。

今後の課題としては、まず第一にデータの入手があげられる。教員の担当希望は制約条件に含まれるため、データ次第で問題の困難さは大きく変化しうる。システムの有用性を考える意味でも、実データでの実験は必須である。次に時間割評価の問題について考える。本稿における評価項目とその値はすべて鄭と平井が一意に決定したものである。幅広い需要にこたえるためには、教員や事務、その他の学生の意見も取り入れる必要がある。最後に拡張性の問題について触れる。本システムは東京農工大学情報工学科を対象に作成したものであるが、当然時間割の作成の際には他学科との兼ね合いや、教室の問題も浮上する。システムとしての完成度を高める意味でも、汎用性、拡張性は重要である。

参考文献

- [1] M.W. Carter and G. Laporte: *Recent Developments in Practical Course Timetabling*, ed. E. Burke and M. Carter *The Practice and theory of automated timetabling* Selected papers. Lecture Notes in Computer Science, Vol. 1408, pp.3-19, Springer-Verlag, Berlin, (1992)
- [2] W. Erben and J. Keppler: *A genetic algorithm solving a weekly course-timetabling problem*, ed. E. Burke and R. Poss *The Practice and theory of automated timetabling* Selected papers. Lecture Notes in Computer Science, Vol. 1153, pp.198-211, Springer-Verlag, Berlin, (1996)
- [3] B. Peachter, R.C. Rankin, and A. Cumming: *Improving a lecture timetabling system for universitywide use*, ed. E. Burke and M. Catter *The Practice and theory of automated timetabling* Selected papers. Lecture Notes in Computer Science, Vol. 1408, pp.156-165, Springer-Verlag, Berlin, (1998)
- [4] M.A.S. Elmohamed, P. Coddington and G. Fox: *A comparison of annealing techniques for academic course scheduling*, ed. E. Burke and M. Catter *The Practice and theory of automated timetabling* Selected papers. Lecture Notes in Computer Science, Vol. 1408, pp.92-112, Springer-Verlag, Berlin, (1998)
- [5] K.A. Dowsland: *Off-the-peg or made-to-measure? Timetabling and scheduling with SA and TS*, ed. E. Burke and M. Catter *The Practice and theory of automated timetabling* Selected papers. Lecture Notes in Computer Science, Vol. 1408, pp.37-52, Springer-Verlag, Berlin, (1998)
- [6] E. Berke and J-P. Newall: *Multi-stage evolutionary algorithm for timetabling problem*, IEEE Trans. Evol. Comput. pp.1085-1092, (1999)
- [7] D.C. Rich: *A smart genetic algorithm for university timetabling*, ed. E. Burke and R. Ross *The Practice and theory of automated timetabling* Selected papers. Lecture Notes in Computer Science, Vol. 1153, pp.181-197, Springer-Verlag, Berlin, (1996)
- [8] 福島 誠, 田中 信二: 学校時間割問題における遺伝アルゴリズムの運用について, 信学論 (D-I), Vol. J82-D-I, No. 6, pp.883-885, (1998)
- [9] 上田 祐彰, 大内 大輔, 高橋 健一, 宮原 哲浩: 遺伝的アルゴリズムの時間割作成問題に関する一考察, 電子情報通信学会論文誌, Vol. J86-D-I, No. 9, pp.691-701, (2003)
- [10] Reiko Tanese: *Distributed Genetic Algorithms*, Proceedings of the Third International Conference on Genetic Algorithms, pp434-439, (1989)
- [11] 関 将俊: dRuby による分散・Web プログラミング, オーム社, (2005)
- [12] Visualu Ruby: <http://vruby.wiki.sourceforge.net/Home-J>