時間割最適化

藤井綾香　(指導教員：工藤和恵)

1　はじめに

大学の時間割は、必修科目、教職科目、語学など、同じ時間帯に重なってほしくない科目が複数あり、時間割を考えるのは難しい。そこで本研究は、重なってほしくない科目が重ならないような時間割を簡単に作成するための方法を考案することを目的とする。

2　時間割ネットワーク

まず、複雑な時間割をグラフで表す。科目を頂点とし、同じ時間帯に重なってはいけない科目を辺で結んでできるネットワークを、時間割ネットワークと呼ぶことにする。2013 年度の情報科学科前期の時間割ネットワークは図 1 のようになる。

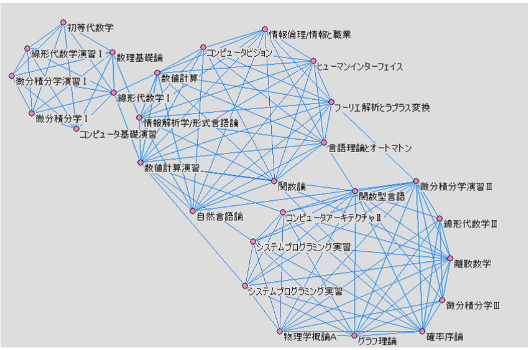
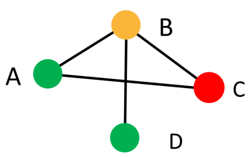


図 1: 情報科学科前期の時間割ネットワーク。頂点が講義、重なってほしくない授業を辺で結んだ。この場合、対象学年が同じ科目同士、同じ先生の科目同士、計算機室を使う授業同士を結んだ。

3　時間割を組む方法

時間割ネットワークを無向グラフとし、頂点彩色をする。まず、コマ (時間帯) それぞれに色を振り分ける。例えば、赤は月曜日 1 限、緑は火曜日 2 限などといったようにする。その後頂点彩色をすると、つながれた頂点同士は違う色で塗り分けられるので、重なってはいけない科目が重ならずに時間割が組める (図 2)。しかし、頂点彩色での時間割作成には問題点がある。まず、頂点彩色問題は NP 完全問題である。つまり任意のネットワークに対して多項式時間でこの問題を解けるアルゴリズムは存在しない。また、時間割には頂点彩色では解けない制約がある。例えば、実習科目は午後がいい、などである。特に後者の問題点を解決するために、時間割ネットワークに関してエネルギーを定義し、最適化問題を解く方法を採用する。



1 限	C
2 限	A,D
3 限	B

図 2: 頂点彩色後の例

4　モデル

時間割ネットワーク上の反強磁性的ポッツ模型のエネルギーを次のように定義する。

$$E = \sum_{(i,j)} J_{ij}\delta_{q_i,q_j} + \sum_{k=1}^{n\_slot}\sum_{i=1}^{n\_node} w_{ik}\delta_{q_i,k}$$

第 1 項は辺で結ばれた頂点 (i,j) の間に生じるエネルギーである。

- i,j: 頂点 (科目)
- q_i,q_j: それぞれの頂点の色 (それぞれの科目のコマ)
- J_{ij}: 枝の重み、今は $J_{ij}=1$ とする

$q_i=q_j$ ならば、$\delta_{q_i,q_j}=1$、$q_i \neq q_j$ ならば、$\delta_{q_i,q_j}=0$ である。つまり、結ばれた頂点同士の色が同じ色ならば、1 であり、違う色ならば 0 である。したがって、エネルギーの合計が 0 になれば、重なってはいけない科目が重ならずに時間割が組める。

第 2 項は各コマの色に関する制約を表している。

- i: 頂点 (科目)
- k: 色 (コマ)
- q_i: 頂点 i の色 (科目が入れられているコマ)
- n_slot: 色の数、n_node: 頂点数

頂点 (科目)i が避けたい色 (コマ)k に対して、正の重み $(w_{ik}>0)$ を、塗りたい色 (コマ)k に対して、負の重み $(w_{ik}<0)$ をあたえる。よって、頂点の重みが最小になるように色を塗っていけば、制約を満たせる。

このエネルギー E が最小のとき、時間割は最適な状態である。E が最小である状態を探すには、どうすればよいか。考えられるすべての色配置について、それぞれ E を計算し、その中から最小のものを選んでくるというやり方では、とても時間がかかってしまう。そこで、本研究ではシミュレーテッド・アニーリング法を用いる。シミュレーテッド・アニーリング法は、モンテカルロ法の一種であるメトロポリス法を発展させた方法であり、近似解を求める一般的な方法である。

5　アルゴリズム

シミュレーテッド・アニーリング法を使った時間割最適化アルゴリズムは、以下の通りである [1]。

ステップ 0　初期値設定
すべての頂点にランダムに色を塗る。

ステップ 1　次の色配置を設定する
ランダムに頂点を一つ選び、ランダムに色を塗り替える。塗り替える前の状態を $X^{(t)}$、後の状態を X' とする。

ステップ２ エネルギー変化率の計算

エネルギー変化率 : $\Delta = \exp(\frac{E-E'}{T})$

E: 状態 $X^{(t)}$ のときのエネルギー

E': 状態 X' のときのエネルギー

T: パラメーター (温度に対応、シミュレーテッドアニーリング法の特徴)

ステップ３ 変化率 Δ と乱数との比較

一様乱数 $r \in [0,1]$ を生成し、以下のように次の状態 $X^{(t+1)}$ を決める。

$$X^{(t+1)} = \begin{cases} X' & r \le \Delta \quad (\text{状態更新}) \\ X^{(t)} & \text{otherwise} \quad (\text{状態更新なし}) \end{cases}$$

ステップ１〜３を何回か繰り返す。

ステップ４ パラメーター T の更新

状態が更新されたら T を下げる: $T = T \cdot dT$

状態が更新されなければいったん T を上げる: $T = T/dT$

(dT は温度変化率、$0 \le dT < 1$)

ステップ５ 繰り返し

ステップ１に戻って、同じ操作を繰り返す。

6 実行

上記のアルゴリズムを使用して、情報科学科前期の授業で実行した。重なってはいけない講義を、「対象学年が同じ学年」、「同じ先生の授業」、「計算機室を使う授業」とする。制約条件としては、「コア科目・語学・教職科目の開講時間を避ける（対象学年が同じのもの）」、「計算室を使う授業は午後」、「専門科目は水曜日を避ける」とする。

実行結果は図３のようになった。重なってはいけない科目が重ならずに組めている。また、制約もすべて満たせている。

	1限	2限	3限	4限	5限
月		微分積分学Ⅰ	微分積分学演習Ⅰ 関数論 関数型言語	線形代数学Ⅲ	
火		物理学概論A 自然言語論	数理基礎論 確率序論	数値計算	数値計算演習
水					
木	離散数学 フーリエ解析とラプラス変換		線形代数学Ⅰ コンピュータアーキテクチャⅡ	線形代数学演習Ⅰ システムプログラミング実習 コンピュータビジョン	システムプログラミング実習
金	グラフ理論 言語理論とオートマトン		微分積分学Ⅲ コンピュータ基礎演習 ヒューマンインターフェース	微分積分学演習Ⅲ 初等代数学 形式言語論	情報倫理

図 3: 実行結果。オレンジ：１年生、青：２年生、赤：3〜４年生対象のもの。

7 制約の数と時間割の組みにくさの関係

時間割アルゴリズムの大きな特徴は、制約をつけられることである。月曜日はいやだ、午後がいいなどのわがままに対応できる。では、どのくらい多くのわがままに対応できるのだろうか。制約をつけるための重みが $w_{ik} \neq 0$ となる i と k の組み合わせの制約の数と、時間割最適化の成功率の統計をとった結果が図４である。T の初期値は、$T_{\max} = 10^3$、最小値は $T_{\min} = 10^{-2}$ に設定する。温度変化率が 0.1 の場合、E が順調に下がっていけば、アルゴリズムの繰り返し回数は５回である。制約の数が 10 から 300 までは、平均繰り返し回数が５程度なので E が順調に下がり、最小値を見つけることができている。制約の数が 400 以上になると、成功率が１未満になった。そこで、温度変化率 dT を大きくし、ゆっくり温度を下げるようにした。制約の数が 600 になると、成功率は大きく下がる。以上をまとめると、制約の数が 10 から 300 程度であれば、時間割が組みやすく、400 以上になると時間割が組みにくくなることがわかった。

制約	温度減少率	成功率	アルゴリズムの平均繰り返し回数
10	0.1	1	4.42150
50	0.1	1	5.00000
100	0.1	1	5.15660
200	0.1	1	5.51290
300	0.1	1	6.02700
400	0.1	0.9969	7.79450
400	0.9	0.9982	40.41510
500	0.9	0.9804	98.04000
600	0.9	0.9069	167.48850

図 4: 10000 回の実行に関する統計。時間割最適化の成功率 ($E = 0$ で実行終了したら成功)、１回の実行でアルゴリズム（ステップ１〜５）を何回繰り返したかの平均を計算したもの。ただし、1000 回を超えると $E \neq 0$ でも、ループをぬける。($w_{ik} = \{0,1\}$ とし、$w_{ik} = 1$ となる i, k はランダムに選ぶ。制約の数は最大で 672 個である。)

8 まとめ

時間割ネットワークの頂点彩色問題を解くことで時間割を組むことを土台とし、時間割ネットワーク上のポッツ模型を導入した。シミュレーテッドアニーリング法を用いたアルゴリズムを使い、ポッツ模型のエネルギーを最小化することで、時間割を組む方法を提案した。この時間割最適化アルゴリズムを、実データで実行した結果、重なってはいけない科目同士が重ならないという条件と、制約を満たすことができた。実験的に制約の数を増やすと、ある値以上で成功率が下がり、計算時間が増えるという結果になった。

今後は、より複雑なネットワークでこのアルゴリズムを使い、研究を発展させたい。

参考文献

[1] 福島孝治,「モンテカルロ法の前線」, 若手研究者・学生向けに最新技術をわかりやすく紹介する講演会「確率的アルゴリズムによる情報処理」講義ノート, (2003)

[2] R.J. ウィルソン (原著), 西関 隆夫 (翻訳)・西関 裕子 (翻訳), 「グラフ理論入門」, 近代科学社, (2001)