

家庭訪問日程計画に関する研究

河野 高士（沼田 一道 准教授）

1. はじめに

1.1 研究背景

文部科学省は、学校教育において教師と家庭との連携を重要視し、教師が担当クラスの児童の家を訪問するよう指示している。教育活動としての家庭訪問は、質問紙調査などでは知ることが難しい親子関係、家庭の雰囲気、子供の校外における生活などを、教師が的確に把握する機会となっている。さらには、家庭という寛いだ場で子供に関する情報や意見を交換することによって、教師と父母の協力、連携を図ることが期待される。家庭訪問は、一般に、新学年が始まった直後の4月下旬から5月上旬にかけて実施されている[1]。

家庭訪問の実施にあたり、1日に訪問できる家庭数には限度があるので、何日かに分けて訪問する必要がある。教師は各実施日に、予め調査した訪問受入れ可能な家庭の中からいくつかの家庭を選び、それらを訪問して学校へ戻る。このとき、各実施日の制限時間の下で、対象全家庭をできるだけ少ない移動時間で訪問するには、どのような組合せで、どのような順番で訪問したらよいか問題となる。これを「家庭訪問日程計画問題」と呼ぶ。家庭訪問日程計画問題はVRP（配送計画問題）の一種とみなすこともできる。

1.2 研究目的

実際の規模のVRPを厳密に解くのは困難であり、発見的解法を用いて出来るだけ良い解を短時間で求めるのが普通である。しかし、家庭訪問日程計画問題の場合には、比較的問題の規模が小さく一日に訪問できる家庭数も少ないので、厳密解法を適用できる可能性がある。本研究では家庭訪問日程計画問題を0-1整数計画問題として表現し、それを汎用ソルバーに入力して厳密解を得るアプローチを試みる。具体的な定式化における列数は極めて多いが、本研究では0-1計画なのであらかじめ必要な全ての列を生成することで対処する。このアプローチによりどの程度の問題例まで厳密に扱えるかを明らかにする。

2. 問題

教師は家庭訪問の期間内に担任クラスの生徒全員に対して家庭訪問を行う。また、教師は家庭訪問を実施する前に各家庭に対して訪問可能な日にちを調査し、それを考慮してスケジュールを作成する。

教師はスケジュールを作成する際、その総コスト（移動時間と滞在時間の和）を最小にしたいと考える。訪問スケジュールとは、「教師が何日目どういう順序でどの家庭を訪れるのか」を定めるものである。本研究では1クラスの生徒数を約30人程度と想定する。

本研究で扱う問題の前提条件は以下の通りである。

教師は学校から出発し最後に学校に帰ってくる。教師が移動するときの速度は一定とする。滞在時間と移動時間の合計は制限時間を越えてはならない。各家庭における滞在時間は分かっているものとする。各家庭の訪問可能日は分かっているものとする。期間は決まっているものとする。

問題の構成要素を以下のような記号で表す。

家庭の集合を $N = (0, 1, \dots, n)$ とする。(ただし, $1, \dots, n$ は家庭を表し, 0 は学校を表す) 家庭訪問の各日の集合 $K = (1, \dots, m)$ とする. 家庭 i, j 間の移動時間を d_{ij} とする. 各家庭の滞在時間を s とする. 1日の制限時間を T とする.

3. 定式化

本研究では家庭訪問日程計画問題を, 家庭の集合に対する巡回路のカバー問題として考える. 具体的には, 各訪問日に実行可能なルートを生きて生成し, 全家庭をカバーするように各訪問日のルート集合から1つずつルートを選んで訪問スケジュールを構成する.

k 日目において可能な i 番目のルートを $r_i^{(k)}$ で表す. その総コスト(移動時間と滞在時間の和)を $c_i^{(k)}$ とする. k 日目に実行可能なルートの総数を $p^{(k)}$ で表す. $r_i^{(k)}$ が家庭 j を通過している(1)か否(0)かは定数 $a_{ij}^{(k)}$ で与える. k 日目にルート $r_i^{(k)}$ を採用する(1)か否(0)かを決定変数 x_{ik} で表す. これらの記号を用いると, 教師の総コストを最小にする家庭訪問日程計画問題は以下のように定式化される.

$$\begin{array}{ll} \min & \sum_{k=1}^m \sum_{i=1}^{p^{(k)}} c_i^{(k)} x_{ik} & (1) \\ \text{st} & \sum_{k=1}^m \sum_{i=1}^{p^{(k)}} a_{ij}^{(k)} x_{ik} = 1 \quad (i=1, 2, \dots, p^{(k)} \quad j=1, \dots, n) & (2) \\ & \sum_{i=1}^{p^{(k)}} x_{ik} \leq 1 \quad (k=1, 2, \dots, m) & (3) \\ & x_{ik} \in \{0, 1\} \quad (i=1, 2, \dots, p^{(k)} \quad k=1, 2, \dots, m) & (4) \end{array}$$

目的関数は(1)教師の総コスト(移動時間と滞在時間の和)であり, (2)~(4)の制約のもとでこれを最小化する.(2)は全ての家庭を必ず1度訪問することを示している.(3)は k 日目に採用されるルートは1以下になることを示している.(4)は x_{ik} が0-1型の整数変数であることを示している.



図1 巡回路カバー問題のイメージ

4. 解法

解法の手順は以下の通りである. , , についてはプログラムを作成して対処する. は GLPK(GNU Linear Programming Kit[3])を利用する.

学校と各家庭の位置座標, 各家庭の訪問可能日を入力する.

全ての $r_i^{(k)}, c_i^{(k)}, a_{ij}^{(k)}$ を生成する.

生成した $c_i^{(k)}, a_{ij}^{(k)}$ を入力し, コストが最小となるルートの組合せを求める.

の出力結果を入力し訪問ルートと移動時間, 滞在時間, 総コストを出力する.

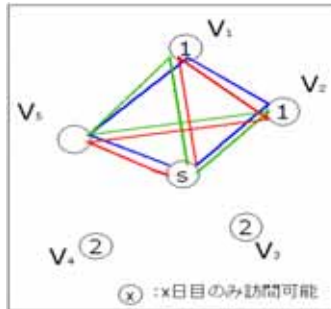
4.1 ルートの生成方法

k 日目において訪問可能な家庭の集合を $G^{(k)}, G^{(k)}$ の部分集合全体の集合の i 番目の要素(部分集合)を $g_i^{(k)}$ とする.

case1 ($|g_i^{(k)}| = 2$ の場合) : 起点に属する点を全て訪問するルートは唯一つなので、それを解とする。

case2 ($|g_i^{(k)}| > 2$ の場合) : 起点を出発して $g_i^{(k)}$ に属する点を全て訪問するルートは $n!/2$ 通り存在するのでその組合せを全て列挙する。その中で、 $s \cdot |g_i^{(k)}| + d_{ij} < T$ という制約を満たし総コストが最小のルートを選出する。

V : 家庭
 $G^{(k)}$: k 日目において訪問可能な家庭の集合
 $g_i^{(k)}$: $G^{(k)}$ の部分集合全体の集合の要素
 $r_i^{(k)}$: $g_i^{(k)}$ と s (起点) で構成される最短のルート



$G^{(1)} = \{V_1, V_2, V_5\}$ 出発して $g_i^{(k)}$
 $2^{G^1} = \{\{V_1\}, \{V_2\}, \{V_5\}, \{V_1, V_2\}, \dots\}$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $g_1^{(1)} \quad g_2^{(1)} \quad g_3^{(1)} \quad g_4^{(1)}$
 $\{V_1, V_5\}, \{V_2, V_5\}, \{V_1, V_2, V_5\}$
 $\downarrow \quad \downarrow \quad \downarrow$
 $g_5^{(1)} \quad g_6^{(1)} \quad g_7^{(1)}$
 $g_7^{(1)}$ に関しては3つのルートが存在するが、最短となる、
 $S \rightarrow V_2 \rightarrow V_1 \rightarrow V_5 \rightarrow S$
 を採用し $r_7^{(1)}$ とする。

図2 ルート生成の例

4.2 GLPK について

GLPK を用いて最適解を求める。GLPK は、線形 (整数) 計画問題を解くためのフリーのソルバーである。GLPK では、問題の記述をモデルファイルとデータファイルに分割して行えるので、データファイルのみを生成する。

5. 実験

5.1 実験概要

前節で述べた方法で、GLPK により最適解を計算する。神奈川県海老名市立上星小学校のあるクラスを対象として行う。1 クラスの人数は 30 人とする。各家庭における滞在時間はすべて等しく 10 分とする。教師は時速 4km で移動するものとして移動時間を算出した。教師が 1 日に訪問できる件数は 7 件までとする。期間は 5 日間とする。1 日の制限時間 (T) を 135 分とする。訪問可能であると提出してくる家庭の件数を様々な値に変化させて解き、どの規模になると解けなくなるのかを見た。

GLPK に入力したモデルファイルは 3 節で定式化を行った式である。データファイルには $c_{ij}^{(k)}$, $a_{ij}^{(k)}$ を入力した。訪問可能であると提出してくる家庭の件数を 1 日当たり 15 件としたときのデータサイズは

$c_{ij}^{(k)}$ (16383 × 5), $a_{ij}^{(1)}$ (16383 × 30), $a_{ij}^{(2)}$ (16383 × 30), $a_{ij}^{(3)}$ (16383 × 30), $a_{ij}^{(4)}$ (16383 × 30), $a_{ij}^{(5)}$ (16383 × 30) である。また、プログラムの開発は Borland 社の Delphi6 で作成した。[4]

```
param C : 1 2 3 4 5 :=
1      31 31 41 41 31
2      21 21 25 25 41
      .
      .
      .
16382  155 170 153 157 142
16383  156 166 147 159 146;

param A1 : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 . .
1      10000000000000000000000000000000000000000000000000000
2      01000000000000000000000000000000000000000000000000000
      .
      .
      .
16382  0000000000000000000000100100000011100011
16383  000000000000000000000010100000011100011;
```

図3 データファイル ($c_{ij}^{(k)}$ と $a_{ij}^{(1)}$)

5.2 結果・考察

表1 と図4 は、訪問可能と提出してくる家庭の件数を 1 日当たり 15 件と入力した時の結果である。

表1 より、教師の総コストは 461 分と分かる。1 日目の総コストのみ 29 分と短くばらつきのある結果となっているが、これは目的関数を全体の総コストを最小にすることとしたためである。また、訪問可能であると提出してく

表1 実験結果

日程	訪問ルート	移動時間	滞在時間	総コスト
1日目	0 15 26 0	9分	20分	29分
2日目	0 2 14 29 1 13 8 23 0	33分	70分	103分
3日目	0 7 5 22 18 27 20 9 0	31分	70分	101分
4日目	0 4 19 16 6 24 12 11 0	43分	70分	113分
5日目	0 17 3 25 10 30 21 28 0	45分	70分	115分

る家庭の件数が増えると総コストの和は減少した。これは件数が増えるとその分生成されるルート数も増え、様々な組合せを考慮できるためである。

学校と各家庭の位置座標，各家庭の訪問可能日のデータ入力に約 20 分かった。プログラムが全ての $r_i^{(k)}$ ， $c_i^{(k)}$ ， $a_{ij}^{(k)}$ を生成するのに約 10 分かかる。また，GLPK でコストが最小となるルートの組合せを求めるのに約 1 分かった。GLPK による求解よりもルートの生成等に

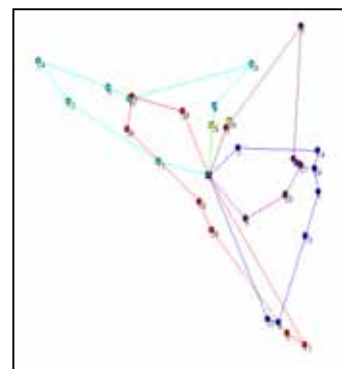


図 4 最短巡回経路の例

要する時間のほうがずっと大きい。GLPK の結果から訪問ルートと移動時間，滞在時間，総コストを出力するプログラムの実行時間は約 1 分である。よって，データの入力時間から結果の出力時間を含め，約 30 分あればスケジュールを作成することができる。

また，訪問可能であると提出してくる家庭の件数を 18 件以上に設定した時は GLPK が解く事ができなかった。これは生成されたルートの組合せ数が膨大に増えてしまい，パソコンのメモリが足りなくなったためである。

6. まとめ・今後の課題

本研究では小学校の家庭訪問のスケジュール作成問題を数理計画問題として定式化し，厳密解法での程度の規模まで解けるのか実験を行った。結果としては，30 人のクラスで訪問可能であると提出してくる家庭の件数を 1 日当たり 17 件としたときまでは厳密解を求めることができた。

何人かの教師にインタビューしたところスケジュールを作成するのにかかる時間は平均 1 時間半程度かかることがわかった。本研究のシステムでは約 30 分で作成できるので，約 1 時間節約することができる。また，教師がスケジュールを手作業で作成する際に，教師の移動コストを厳密に最小なものにするのは困難であることから本研究は十分参考になるものと考えられる。

実際の家庭訪問においては，クラスの児童数（家庭数）や，訪問可能と回答してくる家庭数がより大きな場合も十分考えられる。求解可能な問題の規模を少しでも広げるために，最適性を失わない範囲で生成する列を減らすことを考える必要があるが，それは今後の課題である。

7. 参考文献

[1] 「家庭訪問」，<http://100.yahoo.co.jp/detail/%E5%AE%B6%E5%BA%AD%E8%A8%AA%E5%95%8F/>，
最終閲覧日(2009/12/27)

[2] 「配送計画」，<http://www3.stat.ie.musashi-tech.ac.jp/yokoyama/class/deliveryplan.html>，
最終閲覧日(2009/12/27)

[3] 「GNU Linear Programming Kit」，
<http://www.ibm.com/developerworks/jp/linux/library/l-glpk1/index.html>，
最終閲覧日(2010/01/14)

[4] 掌田 津耶乃：「Delphi パーソナルプログラミング」，毎日コミュニケーションズ，2002