整数線形計画法に対する変数削減による高速化とEMVRPへの応用

髙島 康裕[1,a)]　鷹野 芙美代[2]　中村 祐一[2]

概要：本稿では，整数線形計画法に対する高速化の一手法として，高速に解ける部分問題を解いた後，その解から，不要である変数を選択して削除する手法を提案する．また，その手法を，ドローンでの配送計画等に利用される EMVRP に適用し，実験を行なった．その結果，ほぼ同等の解を得るのに 100 倍以上高速化することができ，提案手法の有効性が確認できた．

キーワード：整数線形計画法，変数削減，EMVRP

1. はじめに

計算機の計算能力は，近年においても，増大の傾向にある．これは，計算を行なうプロセッサの性能向上だけでなく，アルゴリズムの性能向上による部分が大きな要因として上げられる．特に汎用アルゴリズムである整数線形計画法 [1,2] や，SAT ソルバ [3] 等の進歩は目覚しい．本稿では，混合整数線形計画法に注目する．

一般に, 混合整数線形計画法は, NP 困難問題である [1,2]. そのため，ある程度以上の規模の問題で最適解を得ることは，実際上困難であると考えられる．しかし，その限界を少しでも上げるための方策がいくつか提案されている．特に有効なのは，変数や制約を減らすことである．しかし，入力に依存せずに，変数や制約を減らすことはとても難しい．そこで，本稿では，与えられた入力に対し，高速に求解可能な部分問題を考え，その結果から，変数を削減する手法を提案する．また，その手法を EMVRP に適用した．その結果，従来，20 点程度の問題しか解けないとされてきた問題に対し，50 点程度の問題でも高速に良質な解が得られることが確認できた．

本稿の構成は以下の通りである: 2 節において，整数線形計画法における高速化の紹介と，本稿での提案手法を述べる; 3 節では，提案手法をドローン等の配送計画問題となる EMVRP への適用について，記述する; 4 節で，EMVRP への適用における効果を実験での確認したので，報告する; そして，最後に 5 節で本稿をまとめる．

1 北九州市立大学
University of Kitakyushu
2 NEC
NEC
a) takasima@kitakyu-u.ac.jp

2. 整数線形計画法の高速化と提案手法

整数線形計画法は NP 困難であることが知られており, 本質的に高速に求解することは困難であると考えられる [1,2]. しかし，これまでの数多くの研究により，ある程度の高速化は実現できることもわかっている．最も重要な方法は，緩和問題の導入である．緩和問題とは，考慮している数理計画法に対し，一部の制約を外すことによって，求解をより容易にする手法である．一般に，緩和問題は，原問題の解を実行可能解として含むため，緩和問題の最適解は，原問題の良い限界となる．

混合整数線形計画法においては，変数の一部，もしくは，全てに，整数制約が課せられている．そこで，その制約を緩和し，線形計画法として考える [4]．線形計画法は，多項式時間で解くことが可能である．しかし，緩和問題では，変数の整数性は保証できないため，最適解において，実数となる変数が存在する可能性がある．それに対しては，平面切除法と呼ばれる手法が提案されている．これは，緩和問題での解において，整数制約を違反している変数 x_i に着目する．ここでの x_i の値が非整数 α であったとすると，緩和問題の制約に，$x_i \leq \lfloor \alpha \rfloor$ という制約を追加した問題と，$x_i \geq \lceil \alpha \rceil$ という制約を追加した問題とを考え，それぞれ独立に解く．ここで追加した制約を妥当不等式と呼ぶ．そして，この処理を整数制約を満たすまでくりかえす手法である．もちろん，このままでは，最適化処理の高速化には結びつかないが，ここで，先程述べた緩和問題が原問題の良い限界を示すことに着目し，これまでの暫定解とこの緩和問題の解とを比較し，緩和問題が暫定解を更新する可

ⓒ 2020 Information Processing Society of Japan

能性が無い場合は，その探索を打ち切る分枝限定法に利用することで，最適化処理の高速化を実現している．このように，整数制約を外す緩和問題は，その求解が高速であることが保証できる反面，暫定解が構成できないと最適化処理の短縮は実現できない．そのため，できるだけ早く，良質の実行可能解を暫定解として得ることが重要である．

別の緩和手法として，Lazy Constraints と呼ばれる手法が提案されている [5]．これは，緩和問題を解き，解を得た後，その解が外した制約を違反するかを確認し，その結果，違反が無ければ，出てきた解を原問題の解とする．そうでなく，違反している制約があったら，その制約を緩和問題に追加し，再度問題を解く．この作業を実行可能解が出るまで繰り返す手法である．この手法は，定式化中の制約のうち，違反をするような制約があまり多くない場合に効果が大きい．逆に言えば，この手法は，定式化によっては，ほとんど効果が無いこともありえる．

また，緩和ではない高速化については，列生成法と呼ばれる手法が存在している [6]．これは，利用する変数を一時的に削除し，その状況で問題を解く．そして，そこで構成した解に対し，不活性な制約において，改良できる余地がある変数を追加して，再度問題を解くという処理を繰り返し行なう手法である．名前の「列」とは，制約を行列表現したときに，変数が列に相当することに由来する．この手法は，変数が非常に多い反面，目的関数に寄与する変数が少ない場合に有効である．しかし，どの変数を削除すべきかの指針が難しい．

そこで，本稿では，考慮している問題に対し，

(1) 高速に解ける部分問題を構成する．
(2) その部分問題の解を暫定解とし，また，その状況から利用する変数を削減する．
(3) 変数削減を行なった問題にたいし，最適解を得る．

手法を提案する．この手法により，整数線形計画法として，a) 緩和問題で発生する暫定解を得ることの困難さが解消; b) 必要そうな変数を適切に選ぶことにより，高速に問題を解くことが期待することが可能となる．ここで，重要なのが，1) 部分問題の選択，と，2) 変数の選択，となる．この部分は，問題とその実装に依存すると考えられる．

3. EMVRP への適用

さて，本稿では，EMVRP [7,8] を取り上げる．Energy Minimizing Vehicle Routing Problem(EMVRP) とは，基地を出発した Vehicle が各都市を荷物を配送しながら，訪問する問題である．各都市には，配送する荷物が定義されており，それぞれ重量が規定されている．この移動に際し，エネルギーの消費を考える．各都市間の移動エネルギーは，その都市間での重量と都市間の距離の積で表現され，また重量は，車両の重量と，そのときの搭載している荷物の総重量の合計となる．そして，目的は総移動エネルギーの最小化であり，制約は各都市は丁度 1 回だけ訪問することである．

Energy Minimizing Vehicle Routing Problem (EMVRP)

- 入力: 都市集合 $C = \{c_i\}$ (基地: c_0)，車両重量 W_0，各都市への荷物の重量 w_i，都市間 $\{c_i, c_j\}$ の距離 $d_{i,j}$
- 出力: 配送経路
- 目的: 総移動エネルギー最小化
- 制約: 各都市は丁度 1 回訪問

この問題に対し，[7] では，整数線形計画法での定式化を提案している．また，[8] においては，動的計画法による厳密解法と，高速な発見的手法を提案している．ここで，[8] の報告によると，動的計画法による厳密解法では，20 都市程度が実用的な時間で解ける限界であるとしている．

本稿では，[7] が発表されたときからの整数線形計画法のソルバの性能向上を鑑み，その定式化を実装し，gurobi [5] により，求解した．その結果は，20 都市は解けるが，22 都市程度で，計算時間が実用的な時間では難しくなった．そこで，この [7] の定式化を基に，提案手法の適用を検討した．まず，[7] での定式化について，ここで紹介する．

[7] においては，0-1 変数 $x_{i,j}$ として，

$$x_{i,j} = \begin{cases} 1 & \text{都市間 } (c_i, c_j) \text{ を通過するとき} \\ 0 & \text{それ以外} \end{cases}$$

とし，また，$y_{i,j}$ を都市間 (c_i, c_j) での積荷の総重量とする．すると，制約としては，

- 出枝制約: $\forall i, \sum_j x_{i,j} = 1$
- 入枝制約: $\forall j, \sum_i x_{i,j} = 1$
- 配送制約: $\forall i \neq 0, \sum_k y_{k,i} - \sum_j y_{i,j} = w_i$
- 終着制約: $\forall i \neq 0, y_{i,0} = 0$
  最後は積荷は無し
- 積荷制約:
  - $\forall i \neq 0, y_{k,i} \geq w_i x_{k,i}$
    都市 c_i に来るときは最低でも w_i の荷物を積んでくる
  - $\forall i \neq 0, y_{k,i} \leq (W - w_i) x_{i,j}$
    都市 c_i から出るときは最低でも w_i の荷物を積んでいない
    $W = \sum_i w_i$: 積荷の総重量

また，目的は $\min \sum_i \sum_j \alpha d_{i,j}(W_0 x_{i,j} + y_{i,j})$ である．ここで，α は正の定数である．

ここで，目的関数に着目すると，α は全項にかかる正の係数であるので，無視して，$\sum_i \sum_j d_{i,j}(W_0 x_{i,j} + y_{i,j}) =$

ⓒ 2020 Information Processing Society of Japan

$\sum_i \sum_j d_{i,j} W_0 x_{i,j} + \sum_i \sum_j y_{i,j}$ である．すなわち，第 1 項は Vehicle 自体の移動エネルギーの総量であり，第 2 項は積荷の総移動エネルギーに対応する．ここで，第 1 項においては，W_0 は定数であるので，$\sum_i \sum_j d_{i,j} x_{i,j}$ となり，これは総移動距離の最小化となる．つまり，この項だけに着目すると，トラベリングセールスマン問題 (TSP) と等価である．そこで，部分問題として TSP を考える．この TSP に関しては，gurobi [5] において，*Lazy Constraints* と呼ぶ制約生成法を実現する枠組みを用いて，ある程度のサイズの問題まで，高速に求解可能であることが確認されている．また，今，EMVRP では，都市数は 100 程度であるので，整数線形計画法で十分高速に求解可能である．そこで，EMVRP の部分問題として，TSP を利用することとした．すなわち，EMVRP の解法を図 1 に示す．

step 1: EMVRP の部分問題として，TSP を考え，gurobi の Lazy Constraints の枠組みを用いて，求解する．

step 2: step 1 での解に対し，EMVRP の暫定解を構成し，また，都市間の移動を制限する．ここで，移動を認めない都市間に関しては，対応する変数 $x_{i,j}$ を 0 に固定する．

step 3: step 2 での定式化を用いて，EMVRP を求解する．

図 1 EMVRP に対する提案手法

この提案手法による効果は以下の通りと考えている．a) 部分問題である TSP が高速に求解可能であり，また，その解は EMVRP の実行可能解として，良質であることが期待できる．b) TSP の解で通過する都市間の距離より，ある程度以上の距離の都市間は通過しないという仮定が可能である．ただし，この場合は，最適解を失なう可能性はある．

ここで，削除する都市間の選択基準であるが，以下の 2 通りを考えた．

- 手法 1: TSP での経路中での最大距離を閾値
- 手法 2: (TSP での経路中での最大距離) と (各都市からの移動の距離の定数倍) との小さい方を閾値

以下に，各手法に対する例を示す．まず，図 2 に 4 都市からなる入力の例を示す．ここで，都市は A〜D であり，都市間の距離は対応する枝にラベルとして付している．この入力に対し，TSP での最適解を求めた結果を図 3 に示す．ここで，経路は都市 A から矢印の向きに訪問したとする．そして，この経路で最長の都市間は，$d_{C,D} = 4$ である．そこで，手法 1 では，4 より長い都市間である A-C は移動対象から外す．その状況を図 4 に示す．ここで，枝 $\{A, C\}$ は点線で表現している．また，手法 2 では，更に各都市からの移動の定数倍以上の移動は考慮しない．図 5 にその状況を示す．ここで，閾値を定める定数は 3 としている．すると，$d_{B,C} = 1$ より，都市 B からは最大で 3 までの距離の都市にしか移動しない．すなわち，B → D の移動は考慮しない．ちなみに，逆向きの移動，すなわち D → B に関しては，$d_{D,A} = 2$ より，基準を越えていないので，考慮する．図 5 では，枝 (D, B) は考慮するが，枝 (B, D) は考慮しないことを表現するため，(D, B) には矢印を用いている．

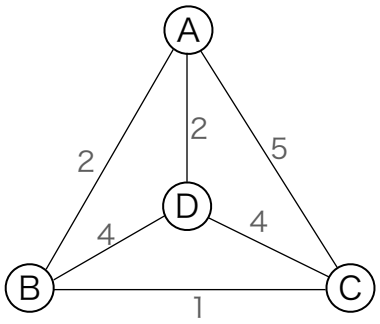

図 2 4 都市からなる入力の例

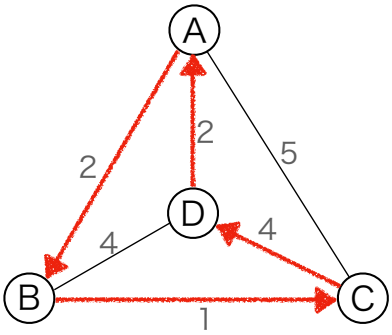

図 3 TSP の結果

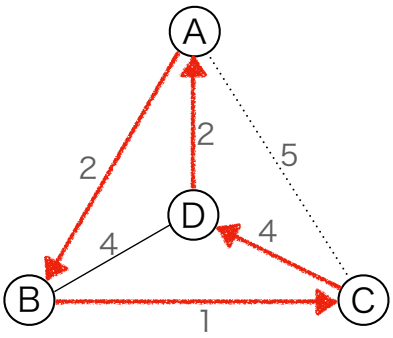

図 4 手法 1 での削除の結果

ⓒ 2020 Information Processing Society of Japan

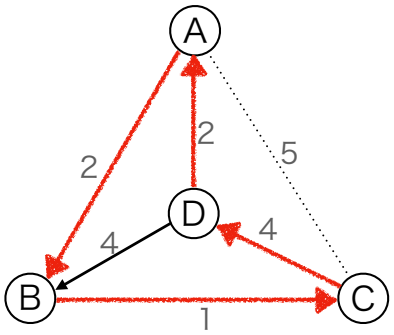

図 5 手法 2 での削除の結果 (定数:3)

4. 実験

提案手法の有効性を確認するため，計算機実装をし，実験を行なった．実験に利用した環境は以下の通りである．

- CPU: Intel Xeon Gold 6126 CPU @ 2.60GHz
- Memory: 196 GB
- OS: CentOS 7.8
- Solver: Gurobi Optimizer version 9.0.0
  - 24 スレッド
- Benchmark: TSPLIB [9] から作成
  - eil7 (7 都市)，eil13(13 都市)，ulysses16(16 都市)，ulysses22(22 都市)，bayg29(29 都市)，att48(48 都市)
  - 基地は最初の都市に設定
  - 各都市への荷物と Vehicle の自重が定義されていないので，追加
- 比較手法
  - 初期解無し ([7] のまま)
  - 初期解有り & 削除無し
  - 初期解有り & 削除 手法 1
  - 初期解有り & 削除 手法 2 (係数は 3.0, att48 では 2.0)

まず，実行時間を表 1 に示す．ここで，各行はベンチマークデータに，各列は比較手法に対応している．また，実行は 1 時間で打ち切った．

表 1 実行時間 (単位:秒)

データ名	初期解無し	初期解有り		
		削除無し	手法 1	手法 2
eil7	0.01	0.01	0.01	0.01
eil13	0.53	0.13	0.10	0.08
ulysses16	5.18	0.90	0.73	0.10
ulysses22	29.79	61.36	38.13	0.27
bayg29	2286.15	—	28.59	30.99
att48	—	—	—	387.32

ここで，—は，時間制限を超過してしまったことを示す．手法 2 の方が手法 1 よりも都市間の移動の削除を多く行なっているので，速くなる傾向にある．ただし，場合によっては，削減数がほとんど変化しないため，同程度の時間がかかる場合もある．また，初期解を与えた方が与えない場合よりも，短かくなる傾向にあるが，これも場合によっては，逆転することもあることがわかった．しかし，削減しないよりはした方が時間が短かくなることは同じ傾向にあり，提案手法が実行時間削減に効果があることがわかる．

次に，総移動エネルギー結果を表 2 に示す．

表 2 総移動エネルギー: 係数 $\alpha = 0.03$

データ名	初期解無し	初期解有り		
		削除無し	手法 1	手法 2
eil7	1119.13	1119.13	1119.13	1119.13
eil13	4234.4	4234.4	4234.4	4234.4
ulysses16	1076.16	1076.16	1076.16	1076.16
ulysses22	1170.7	1170.7	1170.7	1170.8
bayg29	25624	25654.4	25624	25624
att48	4691733.51	4701889.70	4701890.22	4701890.40

ここで，制限時間を超過した状況においては，1 時間のときの途中結果を示している．基本的には，ほとんど同程度の解を出力している．ただし，bayg29 の初期解無しが制限時間内で終了したのに対し，初期解有&削除無しは制限時間を越えている．そのため，総移動エネルギーが大きくなっている．いずれにせよ，提案手法はほとんど解の劣化が無い．よって，提案手法による変数削減は，適切な選択であることが確認できた．

5. おわりに

本稿では，整数線形計画法に対する高速化の一手法として，高速に解ける部分問題を解いた後，その解から，不要である変数を選択して削除する手法を提案した．また，その手法を，EMVRP に適用した．実験により，ほぼ同等の解を得るのに 100 倍以上高速化することができ，提案手法の有効性が確認できた．

今後の課題としは，変数削減の手法はまだ改良の余地があるため，どのように削除変数を選択すべきかという部分を検討することが必要である．

参考文献

[1] 藤江 哲也: 整数計画法による定式化入門, オペレーションズ・リサーチ, 57-4 (2012), pp.190–197.

[2] 宮代 隆平: 整数計画法メモ, http://web.tuat.ac.jp/~miya/ipmemo.html.

[3] 宋 剛秀, 番原 睦則, 田村 直之, 鍋島 英知: SAT ソルバーの最新動向と利用技術, コンピュータソフトウェア, 35 巻 第 4 号, pp. 72–92, https://www.jstage.jst.go.jp/article/jssst/35/4/35_72/_article/-char/ja, 2018.

[4] 宮代 隆平, 松井 知己: ここまで解ける整数計画, システム/制御/情報, vol. 50, pp. 363–368, https://web.tuat.ac.jp/~miya/miyashiro_matsui_ISCIE_rev5.pdf, 2006.

[5] Gurobi Optimization: Gurobi Optimizer Reference Manual Version 9.0, https://www.gurobi.com/documentation/9.0/refman/index.html.

[6] 宮本 裕一郎: はじめての列生成法, オペレーションズ・リサーチ, 57(4), pp. 198–204, https://www.orsj.or.jp/

ⓒ 2020 Information Processing Society of Japan

archive2/or57-04/or57_4_198.pdf, 2012.

[7] Imdat Kara, Bahar Y. Kara, M. Kadri Yetis: Energy Minimizing Vehicle Routing Problem, COCOA 2007, LNCS 4616, pp. 62–71, 2007.

[8] 舟橋 勇佑, 柴田 敦也, 根來 俊輔, 谷口 一徹, 冨山 宏之: 動的計画法に基づくドローンの低消費エネルギー配送計画, 研究報告システムと LSI の設計技術 (SLDM), 2018-SLDM-185 (2), pp. 1–5.

[9] TSPLIB: Capacitated vehicle routing problem (CVRP) data, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/vrp/.

ⓒ 2020 Information Processing Society of Japan