日本オペレーションズ・リサーチ学会
2023 年春季研究発表会

列生成法を用いた分割配送経路問題の求解高速化

非会員　沖電気工業株式会社　　玉井　秀明　　TAMAI Hideaki
非会員　沖電気工業株式会社　*大矢　祐輝　　OOYA Yuuki

1. はじめに

分割配送経路問題[1]を解くことで得られる配送ルートは、通常の配送経路問題に比べより効率が高い反面、求解時間は増加する傾向にある。実際の配車業務への適用を考慮すると、求解の高速化が課題となっている。

今回、我々はヒューリスティック解法の一つである列生成法を時間窓付き複数回転分割配送経路問題(Multi-Trip Split Delivery Vehicle Routing Problem with Time Window : MTSDVRPTW)に適用し、計算時間の短縮を図ったので報告する。

2. MTSDVRPTW の定式化

車両がデポを出発し店舗を訪問して再びデポに帰着する一回転を「便」と定義し、ある車両に割り当てられた複数の便を「ルート」と定義する。文献[2]に記載の方法をベースとした以下の手続きによって MTSDVRPTW の求解を行う。

Step 1.　制約を一部緩和した主問題(【RelMAIN】)の実行可能解が得られるk_0本のルートからなる集合を用意し、これをR_{k_0}とする
Step 2.　車両の最大積載量をg通り変化させ、Step 3. ～Step 5. をループする(列生成法)
Step 3.　制約R_kのみで構成された【RelMAIN】の双対問題(【TEMP-k】) を作る
Step 4.　【TEMP-k】の最適解λ_i^{*k}を得る
Step 5.　λ_i^{*k}を入力としてルート生成子問題(【SUB】)を解く
　Step 5-1.　【SUB】の最適値が 0 以上ならば終了する
　Step 5-2.　そうでないならば、R_kに【SUB】の最適解に相当するルートを加えたものをR_{k+1}とし、kを 1 増やし、Step 3. へ移る
Step 6.　列生成法終了後に、生成されたルートだけを用いて主問題(【MAIN】)を解く

(定数)

- m：車両台数
- n：店舗数
- b：1 ルートあたりの最大便数
- h_i：店舗iに訪問できる最大車両数
- i,j：店舗を表す添え字(0 はデポ、その他は店舗)
- v：車両を表す添え字
- f：便を表す添え字
- r：ルートを表す添え字
- d_i：店舗iの需要量
- a_{ri}:ルートrに店舗iが含まれているか否かを表す 0-1 定数
- a_{rif}:ルートrのf便目に店舗iが含まれているか否かを表す 0-1 定数
- c_{ij}:店舗iからjへ移動する際のコスト
- c_r:ルートrに沿って移動する際のコスト
- e_i:店舗iの最早到着時刻
- l_i:店舗iの最遅到着時刻
- t_{ij}:店舗iからjへ移動する際の所要時間
- q　：車両の最大積載量
- q_p：p回目のループにおける車両の最大積載量

(変数)

- s_r：ルートrを採用したか否かを表す 0-1 変数
- s_{rv}：車両vがルートrを採用したか否かを表す 0-1 変数
- u_{if}:ある車両がf便目に店舗iに到着する時刻を表す実数変数
- w_f：f便目を使って配送するか否かを表す 0-1 変数
- x_{ijf}：ある車両がf便目に、店舗iからjへ移動したか否かを表す 0-1 変数
- y_{if}：ある車両がf便目に店舗iを訪れたか否かを表す 0-1 変数
- y_{ivf}:車両vがf便目に店舗iを訪れたか否かを表す 0-1 変数
- z_{ivf}:車両vがf便目に店舗iへ配送する荷物量を示す整数変数

【RelMAIN】

(目的関数)

$$min \sum_r c_r \cdot s_r$$

(制約条件)

$$\sum_r a_{ri} \cdot s_r \geq 1 \quad (i = 1 \sim n)$$

$$0 \leq s_r \leq 1$$

【TEMP-k】

(目的関数)

$$max \sum_{i} \lambda_i$$

(制約条件)

$$\sum_{i} a_{ri} \cdot \lambda_i \geq c_r \quad (r = 1 \sim k)$$

$$0 \leq \lambda_i \quad (i = 1 \sim n)$$

【SUB】

(目的関数)

$$min \sum_{i,j,f} c_{ij} \cdot x_{ijf} \quad - \sum_{i,f} y_{if} \cdot {\lambda_i}^{*k}$$

(制約条件)

$$y_{0f} = w_f \quad (f = 1 \sim b)$$

$$\sum_{f} y_{if} \quad \leq 1 \quad (f = 1 \sim b)$$

$$w_f \geq w_{f+1} \quad (f = 1 \sim b)$$

$$\sum_{i} x_{ijf} \quad = y_{jf} \quad (i = 1 \sim n\,, f = 1 \sim b)$$

$$\sum_{j} x_{ijf} \quad = y_{if} \quad (j = 1 \sim n\,, f = 1 \sim b)$$

$$u_{if} + t_{ij} - M(1 - x_{ijf}) \leq u_{jf}$$

$$(i = 1 \sim n, j = 1 \sim n, f = 1 \sim b, M\text{は巨大数})$$

$$u_{01} = 0$$

$$u_{0f} \geq u_{n+1\,f-1} \quad (f = 2 \sim b)$$

$$e_i \leq u_{if} \leq l_i \quad (i = 1 \sim n\,, f = 1 \sim b)$$

$$\sum_{i} y_{if} \cdot d_i \quad \leq q_p \quad (f = 1 \sim b)$$

【MAIN】

(目的関数)

$$min \sum_{r,v} c_r \cdot s_{rv}$$

(制約条件)

$$\sum_{v,f} y_{ivf} \quad \geq 1 \quad (i = 1 \sim n)$$

$$\sum_{v,f} y_{ivf} \quad \leq h_i \quad (i = 1 \sim n)$$

$$\sum_{v,f} z_{ivf} \quad = d_i \quad (i = 1 \sim n)$$

$$z_{ivf} \leq d_i \cdot y_{ivf} \quad (i = 1 \sim n, v = 1 \sim m, f = 1 \sim b)$$

$$\sum_{i} z_{ivf} \quad \leq q \quad (v = 1 \sim m, f = 1 \sim b)$$

$$\sum_{r} a_{rif} \cdot s_{rv} \quad = y_{ivf}$$

$$(i = 1 \sim n, v = 1 \sim m, f = 1 \sim b)$$

3. 効果の検証

先述のモデルを実装し数理最適化ソルバー(Gurobi Optimizer 10.0)で求解した。入力条件は、実際の配送が行われているエリアの実績をもとに作成した。店舗数を約 50 店とした。

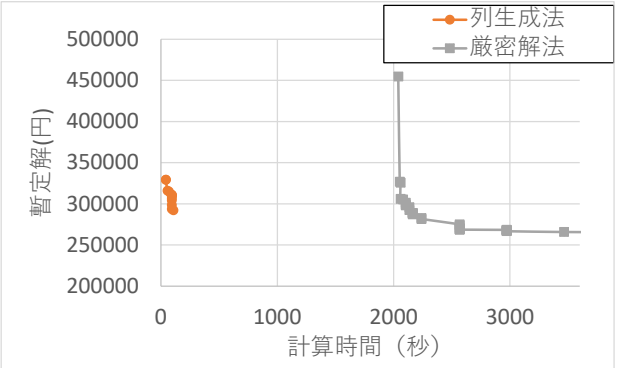

図 1　数値実験結果

数値実験結果を図 1 に示す。初期解を得るのに従来の厳密解法では 2040 秒要したのに対し、列生成法では 31 秒と約 67 倍の高速化を確認した。一方最終的に得られた目的関数値は厳密解法で 264,884 円に対し列生成法では 281,231 円と、約 6%の劣化が見られた。これは今回の手法が近似解法であることに起因する。生成するルート数を増やすと解精度の劣化を抑えることができるが計算時間は増加する。このトレードオフ関係を実際の利用環境に応じて適切に調整することが今後の課題である。

4. おわりに

列生成法を MTSDVRPTW に適用して求解の高速化を検討した。従来の厳密解法に比べ、約 67 倍の計算高速化を確認した。

参考文献

[1] 高橋, 玉井："分割配送路問題を応用した配送業務効率化", 日本オペレーションズ・リサーチ学会 2021 年春季研究発表会, 2-F-3, 2021.

[2] Pedro Munari et al., "A Column Generation Based Heuristic for the Split Delivery Vehicle Routing Problem with Time Windows," SN Operations Research Forum, vol. 1, issue 4, pp. 1-24, 2020.