1K-6

配送計画問題に対する差分近似アルゴリズムの実験的評価

一川 直弥 名古屋 孝幸

鳥取環境大学 環境情報学部 情報システム学科

1 はじめに

配送計画問題とは，デポを出発した複数の車両が各顧客に荷物を配送しデポに戻ってくる経路で，距離の総和が最小となる経路を求める問題である．この問題は NP-困難であることが知られており，近似解を高速に求めるための近似アルゴリズムの研究が盛んになされている．Nagoya[2]は車両の積載量に制限がある容量制約付き配送計画問題に対する差分近似アルゴリズムを提案した．本研究では，そのアルゴリズムに対する計算機実験を行い，その実際的な性能を評価する．

2 問題の定式化

辺重み付き完全無向グラフ $G=(V,E)$， $V=\{0,1,\ldots,n\}$ を考える．頂点 0 はデポと呼ばれる特殊な頂点を表し，その他の頂点は顧客を表す．ルートとは，デポを含む閉路または長さ 2 の歩道 $(0,i,0), i \neq 0$ のことをいう．本研究では，各車両が訪問できる顧客数が k に制限された容量制約付き配送計画問題を考える．この問題の目的は，次の 3 つの条件 $\forall i \in \{1,\ldots,r\}: |C_i| \le k+1$， $\forall i: j \in \{1,\ldots,r\}$， $V(C_i) \cap V(C_j)$， および， $\cup V(C_i)=V(G)$ を満たすルートの集合 $\{C_1,\ldots,C_r\}$ で，辺の重みの総和が最小となるものを求めることである．

3 差分近似率

近似アルゴリズムの性能保証としては，近似比が多く用いられる．すなわち，すべてのインスタンス I に対して， $apx/opt \le \alpha$ を満たす α を近似率として用いる方法である．ここで， opt はインスタンス I に対する最適解のコストであり， apx はインスタンス I に対してアルゴリズムが出力する解のコストである．しかしながら，最悪の解と最適解のコストが近い場合，この尺度では有意な保証が得られない場合がある．そこで別の尺度として，Ausiello ら[1]によって差分近似率が提案された．すなわち，全てのインスタンス I に対して，

$$\frac{wor-apx}{wor-opt} \ge \alpha$$

を満たす α を近似率として用いる方法である．ここで， wor は入力 I に対する最悪の解のコストである．

4 配送計画問題に対する差分近似アルゴリズム

この節では，文献[2]で提案されたアルゴリズムについて述べる．配送計画問題に対する入力グラフを $G=(V,E)$ とし，各車両が訪問できる顧客数の上限を k とする．アルゴリズムでは，まず実行可能解のコストの下界 LB を求める．次に、コスト $LB+\delta_{good}$ の実行可能解 sol_{good} と，コスト $LB+\delta_{bad}$ の実行可能解 sol_{bad} で， $\delta_{good} < \delta_{bad}$ を満たすものを計算する．すると， sol_{good} をアルゴリズムの出力とし， LB と δ_{bad} をそれぞれ最適解と最悪の解のコストの下界として用いることで，差分近似率を次式で評価することができる．

$$\alpha \ge \frac{wor-apx}{wor-opt} \ge 1-\frac{\delta_{good}}{\delta_{bad}}$$

よって， sol_{good} を構成したときの LB からのコストの増分 δ_{good} をできるだけ小さく， sol_{bad} を構成したときのコストの増分 δ_{bad} をできるだけ大きくすることで，上記の式の値を 1 に近づけることができる．文献[2]で提案されたアルゴリズムは， $\delta_{good} \le \max\{1/3, 2/(k+1)\}\delta_{bad}$ を満たす sol_{good} と sol_{bad} を求めることで，差分近似率 $\min\{2/3,(k-1)/(k+1)\}$ を実現している．

以下では， LB ， sol_{good} ，および sol_{bad} の計算方法を説明する．

4.1 LB の計算

実行可能解のコストの下界 LB を求めるために，最小重みサイクルカバーを利用する．グラフ G の部分グラフ H がサイクルカバーであるとは， H の各頂点の次数が 2 であるときをいう．最小重みサイクルカバーのへの入力 $G'=(V',E')$ を，次に従って作成する．

1. $2n$ 頂点からなる完全グラフ V_0 を用意する． V_0 の任意の 2 頂点間の距離は 0 とする．
2. デポ 0 を V_0 で置き換える．
3. V_0 の頂点 i と， V の頂点 j の間の距離は，グラフ G における 0 と j の距離と同じであるものとする．

すなわち， $G'=(V',E')$ を $V'=(V \setminus \{v_0\}) \cup V_0$， $E'=E \cup \{(i,j) \mid i \in V_0, j \in V\} \cup \{(i,j) \mid i,j \in V_0\}$ で定める．このようにして得られたグラフ G' に対して，最小重みサイクルカバー M' を求める．後述するように，最小重みサイクルカバーは効率的に求めることができる．

上記で得られた G' と M' に対して， V_0 の頂点を同一視することで，グラフ G の頂点集合を被覆するサイクルの集合 M が得られる．このとき， M' のコストと M のコストは同じである．また M' のコストを LB とすると， LB は実行可能解のコストの下界となることが示せる．

4.2 最小重みサイクルカバーの計算

最小重みサイクルカバー問題は，最小重み完全マッチング問題に還元することで多項式時間で解ける．まず，4.1 で構成した G' から最小重み完全マッチング問題に対する入力 G'' を次のように構成する．

An Experimental Study of a Differential Approximation Algorithm for the Vehicle Routing Problem
Naoya ICHIKAWA , Takayuki NAGOYA, Department of Information Systems, Tottori University of Environmental Studies

1. グラフ G' の各辺 (v_i, v_j) に対して，新しい頂点 (v_s, v_t) を用意し辺 (v_i, v_j) を長さ4の道 (v_i, v_s, v_t, v_j) に置き換える．辺 (v_i, v_s)，(v_t, v_j) の重みはそれぞれ (v_i, v_j) の重みの半分とし，辺 (v_s, v_t) の重みは0とする．
2. グラフ G' の各頂点 v_i に対して，新しい頂点 v'_i を用意し，グラフ G'' に加える．また，頂点 v'_i の近傍は，G における頂点 v_i の近傍と同じものとす

上記により得られた G'' の最小重み完全マッチング M'' を計算する．最小重み完全マッチングは効率的に計算できることが知られている．

次に，最小重み完全マッチング M'' から，グラフ G' の最小重みサイクルカバー M' を以下のように計算する．

1. G' の任意の 2 頂点 v_i，v_j に対して，G'' における v_i から v_j への長さ4の道を (v_i, v_s, v_t, v_j) とする．このとき $(v_s, v_t) \notin M''$ であるとき，かつそのときに限り辺 (v_i, v_j) を M' に含める．

これにより得られる M' は，G' の最小重みサイクルカバーであることが証明できる．

4.3 sol_{good}，sol_{bad} の計算

4.1節で計算したサイクルの集合 M は，実行可能解とは限らない．具体的には，C_i を M に含まれる任意のサイクルとすると，1) $|C_i| \le k$ かつ C_i が頂点 0 を含まない場合，2) $|C_i| > k$ かつ C_i が頂点 0 を含む場合，および 3) $|C_i| > k$ かつ C_i が頂点 0 を含まない場合，が存在する．各 C_i をルートの集合に修正することで，M から2つの実行可能解 sol_{good} と sol_{bad} を構成する．すなわち，上記の3つの場合に対して，C_i をルートの集合 $sol_{good,i}$ と $sol_{bad,i}$ に変換し，全ての i に対する $sol_{good,i}$ と $sol_{bad,i}$ の和集合をとることで，それぞれ sol_{good} と sol_{bad} を構成する．このとき，M から sol_{good} を構成したときのコストの増分 δ_{good} をできるだけ小さく，M から δ_{bad} を構成したときのコストの増分 δ_{bad} をできるだけ大きくするように構成する．まず，各 $C_i = (1, ..., m_t)$ に対して，長さ 2 の歩道の集合 $(0,1,0), (0,2,0), ..., (0, m_t, 0)$ を求め，それを $sol_{bad,i}$ とする．以下では，sol_{good} の構成方法を，上記の3つの場合に分けて説明する．

Case1：$|C_i| \le k$ かつ C_i が頂点0を含まない場合，一般性を失わずに $C_i = (0,1,2,...m_i,0)$ と仮定する．このとき，各 $1 \le p \le m_i$ に対して，辺 $(p, p+1)$ を削除し，新しい辺 $(p, 0)$ と $(0, p+1)$ を加えることで2つのルートが得られる．これらの中からコストが最小のものを選び，$sol_{good,i}$ とする．

Case2：$|C_i| > k$ かつ C_i が頂点0を含む場合，一般性を失わずに $C_i = (0,1,2,...m_i,0)$ と仮定する．各 $l, (0 \le l \le k-1)$ に対して，ルートの集合を次のように計算する．各 $p, (1 \le p \le m_i - 1, l \equiv p \bmod k)$ に対して，C_i に含まれる辺 $(p, p+1)$ を削除し，新しい 2 つの辺 $(p, 0)$ と辺 $(0, p+1)$ を加える．これにより得られたルートの集合の中からコストが最小のものを選び $sol_{good,i}$ とする．図 1 に，この場合の計算例を示す．

Case3：$|C_i| > k$ かつ C_i が頂点0を含まない場合，一般性を失わずに $C_i = (1,2,...,m_i,1)$ と仮定する．各

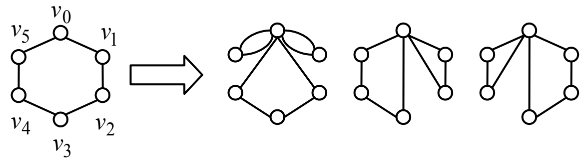

図1　case2 の構成例

$1 \le p \le m_i - 1$ に対して $e_p = (p, p+1)$ とする．各辺 e_p に対して，e_p を除いてそれらの端点を頂点0に接続して得られるルートを考え，それらの中で最もコストの小さいものを C_i' とする．このとき，得られた C_i' に対して，case2 と同様の操作により $sol_{good,i}$ を計算する．

以上のいずれの場合においても，$sol_{good,i}$ と $sol_{bad,i}$ のコストは $\delta_{good,i} \le \max\{1/3, 2/(k+1)\}\delta_{bad,i}$ を満たすことが示せる．以上より，次の定理が成り立つ．

定理 1:([2])容量制約付き配送計画問題に対する差分近似率 $\min\{2/3, (k-1)/(k+1)\}$ の近似アルゴリズムが存在する．

5　計算機実験

計算機実験のための入力データは，VRPLIB URL *http://neo.lcc.uma.es/radi-aeb/WebVRP/* のベンチマークテスト用データを利用し，2 種類の近似率 apx/opt および $\alpha = (wor - apx)/(wor - opt)$ を求めた．ただし，最悪の解を求めることは困難であるため，wor の代わりに δ_{bad} を用いて α を評価した．結果は表 1のようになった．

表 1　各入力に対する実行結果と近似率

頂点数	17	21
k	6	7
opt	2685	3704
apx	2948	3836
apx/opt	1.0979516	1.0356371
α	0.95255277	0.98611111

6　おわりに

文献[2]によって提案された差分近似アルゴリズムを実装し，計算機実験を行った．しかしながら，今回実験した入力データは頂点数が少なく，またデータ数自体もが少ないため，より多くのデータに対して実験をする必要がある．

参考文献

[1] G. Auseillo, A. D'Atri, M. Protasi, Structure preserving reductions among convex optimization problems, J. Comput. System Sci. 21(1980),pp.136-153.

[2] T. Nagoya, New differential approximation algorithm for k-customer vehicle routing problem, Information Processing Letters, 109(2009), pp.405-408.