Fukasawa, R., Longo, H., Lysgchoa, E., Werneck, RF. : Robust branch-and-cut-and-price for the capacitated vehicle routing problem, Mathematical Programming, Vol. 106, No.3, pp. 491-511, 2006.

D1 浦田淳司

2013年6月28日

論文ゼミ＃10

Capacitated Vehicle Routing Problem

基本定義

•ネットワークG=(V, E)
 - V = {0, 1,…,i,…, n} : Vertex, 0はデポ
 - E = {1, 2,…,e,…,m} : Edge
•頂点iは顧客を示し，需要diを持つ
•辺eは長さleをもつ（le > 0）
•配送車両はK台，車載容量はC

配送ルール

•デポが車両の起点，終点となる
•各顧客には車両1台で配送（ルートは交差しない）
•各ルートの顧客需要の最大値はC以下

全ルートの距離の合計値を最小化

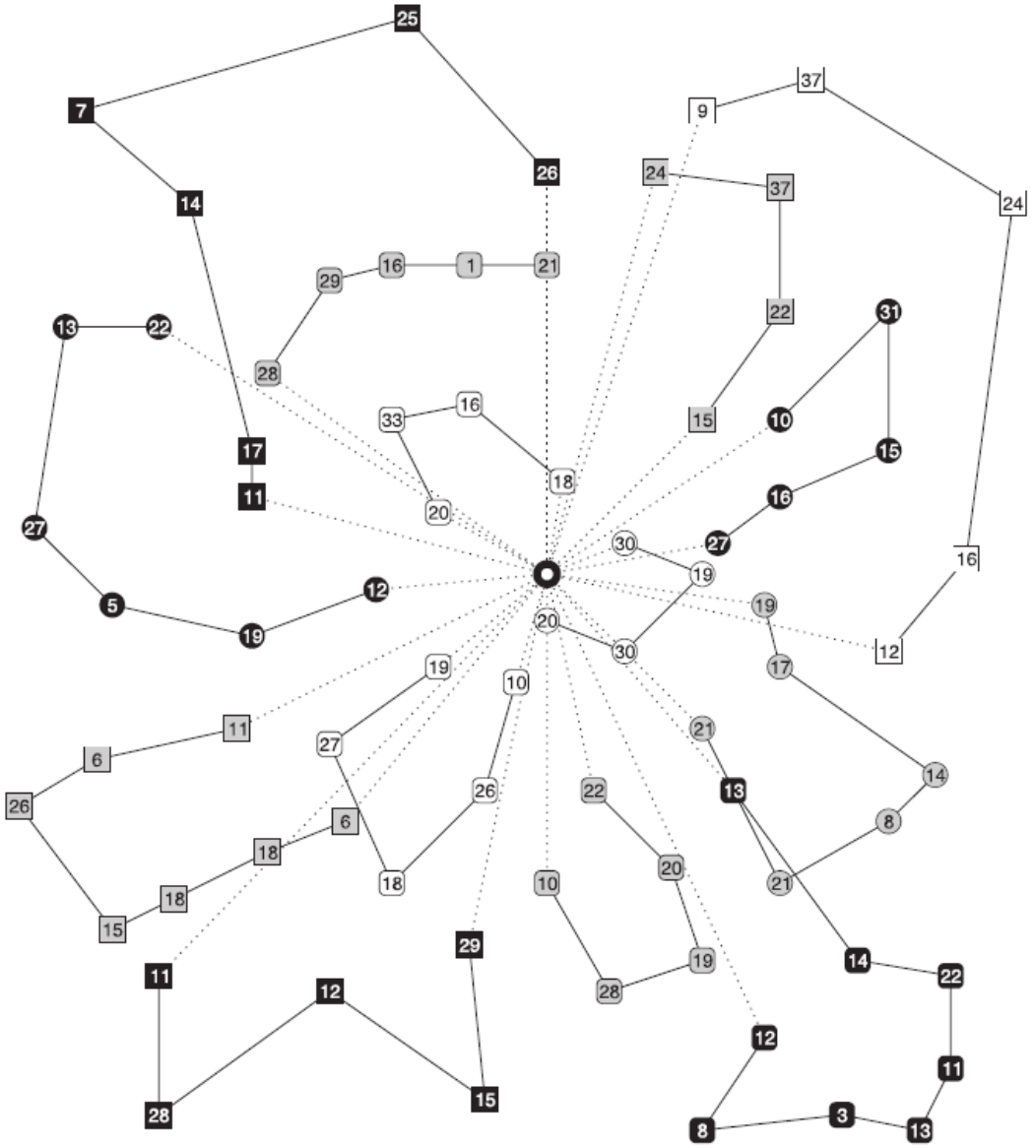


Fig. 3. Optimal solution to E-n76-k14 (vehicle capacity = 100)

次数・容量制約

$$L_1 = \min_{x \in P_1} \sum_{e \in E} l_e x_e$$

$$P_1 = \begin{cases} \sum_{e \in \delta(\{i\})} x_e = 2 \ \ \forall i \in V_+ & (1) \\ \quad \text{頂点iは2本の辺（出入）} \\ \sum_{e \in \delta(\{0\})} x_e = 2K & (2) \\ \quad \text{デポは車両数×2本の辺} \\ \sum_{e \in \delta(\{S\})} x_e \geq 2 \cdot k(S) \ \ \forall S \subseteq V_+ & (3) \\ \quad \text{必要台数×2本の辺} \\ x_e \leq 1 \ \ \forall e \in E \setminus \delta(\{0\}) & (4) \\ \quad \text{1辺の通過回数（折り返しなし）} \\ x_e \geq 0 \ \ \forall e \in E \end{cases}$$

x_e : 辺eの車両通過有無

$V_+ = \{1, \ldots, n\}$: 頂点

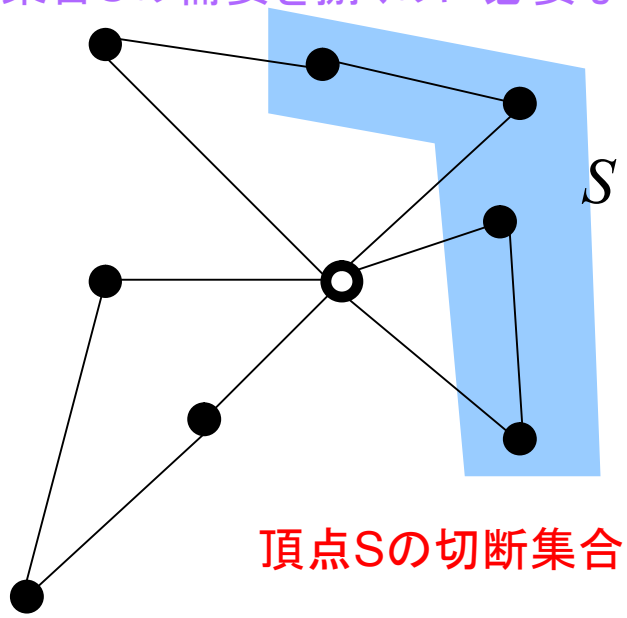
$S \subseteq V_+$: 頂点の部分集合

$d(S)$: 部分集合Sの需要の合計

$\delta(S)$: 部分集合Sの切断集合

$k(S) = d(S)/C$: 車載容量とSの需要の比

集合Sの需要を捌くのに必要な台数



頂点Sの切断集合 $\delta(S)$

ルート・台数制約

$$L_2 = \min_{x \in P_2} \sum_{e \in E} l_e x_e$$

$$P_2 = \begin{cases} \sum_{j=1}^{p} q_j^e \lambda_j - x_e = 0 \ \ \forall e \in E & (5) \\ \sum_{j=1}^{p} \lambda_j = K & (6) \\ \sum_{e \in \delta(\{i\})} x_e = 2 \ \ \forall i \in V_+ & (1) \\ x_e \geq 0 \ \ \forall e \in E \\ \lambda_j \geq 0 \ \ \forall j \in \{1, \ldots, p\} \end{cases}$$

辺eの利用有無xはルートjの辺eの包含有無とルートjの利用有無で記述可能

利用ルート数と車両数は一致

行列Q=

	Edge e 1	2	3	…	e	…	m	
	0	0	1	…	0	…	1	0	1
	1	0	0	…	1	…	0	0	2
	0	1	0	…	0	…	1	1	3
	:								
ルート候補	1	0	1	…	1	…	0	1	j
	:								
	0	1	0	…	0	…	1	0	p

q_j^e : 行列Qの(e, j)成分

λ_j : ルートjの利用有無

最適化計算の合成

(1)~(4)に(5)式を代入

$$L_3 = \min_{x \in P_1} \sum_{e \in E} l_e x_e = \min \sum_{j=1}^{p} \sum_{e \in E} l_e q_e^j \lambda_j$$

s.t.

$$\sum_{j=1}^{p} \sum_{e \in \delta(\{i\})} q_j^e \lambda_j = 2 \quad \forall i \in V_+ \qquad (8)$$

$$\sum_{j=1}^{p} \sum_{e \in \delta(\{0\})} q_j^e \lambda_j = 2K \qquad (9)$$

$$\sum_{j=1}^{p} \sum_{e \in \delta(\{S\})} q_j^e \lambda_j \geq 2 \cdot k(S) \quad \forall S \subseteq V_+ \qquad (10)$$

$$\sum_{j=1}^{p} q_j^e \lambda_j \leq 1 \quad \forall e \in E \setminus \delta(\{0\}) \qquad (11)$$

$$\lambda_j \geq 0 \quad \forall j \in \{1, \ldots, p\}$$

一応，ラグラジアンの未定乗数法で解ける

Column Generation I (問題の設定)

双対変数の対応　(8), (9), (10), (11) : μ , v, π , ω

ラグランジュ関数を $x_e = \sum_{j=1}^{p} q_j^e \lambda_j$ で微分して求めた辺eによる減少コストは次となる

$$\bar{c}_e = \begin{cases} l_e - \mu_i - \mu_j - \sum_{S \mid \delta(S) \in e} \pi_S - \omega_e & e = \{i, j\} \in E \setminus \delta(\{0\}) \\ l_e - v - \mu_j - \sum_{S \mid \delta(S) \in e} \pi_S & e = \{0, j\} \in \delta(\{0\}) \end{cases}$$

Pricing Subproblem

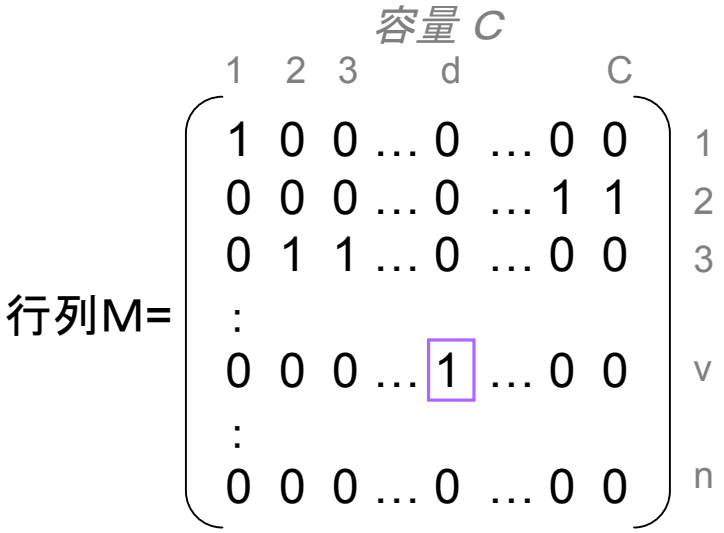
上位問題：K台の選択ルートの距離和の最小化（全顧客への配送制約付き）

q本のルートの中から制約条件を満たし，最小化するK本のルートを抽出

下位問題：距離の短いルート群の生成 (Column Generation)

NP-hard :
与えられた解の真偽でさえ，多項式時間以内に判定できない問題
（NP (Non-deterministic Polynomial time) : ある解が与えられたときに，
その解が正しいかどうかを多項式時間内に解くことができる問題）
⇒近似解探索戦略が必要(メタヒューリスティック)

Column Generation II (形成アルゴリズム)



$M(d,v)$: 頂点vまでの道(walk)とその際の総需要d

$\bar{c}(M(d,v))$: vまでの道のコスト

■Dynamic Programmingの考え方で道を形成

(最適性原理は「最適解を持つ問題の部分問題の最適解は元の最適解の部分解である」)

w(vの近傍点)をvの次の道に追加することを考える

$\bar{c}(M(d,v)) + \bar{c}_{\{v,w\}} < \bar{c}(M(d+d_w,w))$ $\bar{c} < 0$ が成り立つとき

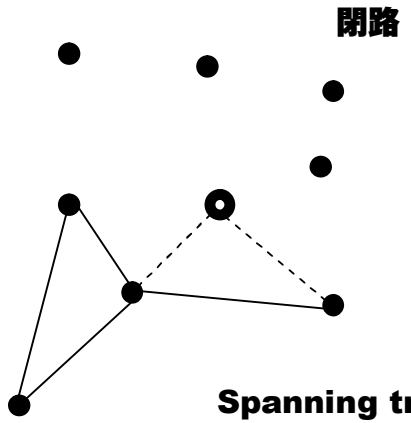
$\bar{c}(M(d+d_w,w))$ に道を延長

行列内の点の数はnC, ルートの本数は最大n(頂点数)なので, O(C*n^2)時間内に算出可能

Column Generation III (高速化)

■Cycle elimination

途中で閉路となっているルートの除去
-記憶量としてsを指定
-s個前までの通過点を記憶し，閉路となっているかを判断
-s=2,3,4をテスト



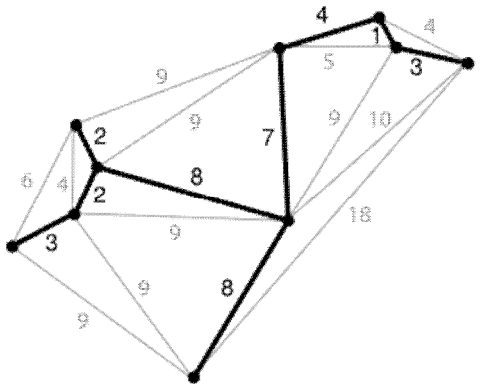
■Heuristic Acceleration (発見的問題解決による高速化)

□scaling :
•容量の基礎単位を1からgに変更する

需要 $d'_v(g) = d_v / g$

容量 $C' = C / g$

□sparsification :
•1つのグラフを5つに分割し，Spanning treeを予め設定
•そのtree上の道を利用して，Dynamic programmingによりルートを形成



ある頂点から他の全ての頂点への移動コストが最小になるような最短経路木．
なお，木とは閉路をもたないグラフのこと．

□bucket pruning (枝刈):
•形成ルートにおける頂点列として，異なる頂点のみで構成されていれば頂点列を保管

Cut Generation

(10)式 $\sum_{j=1}^{p} \sum_{e \in \delta(\{S\})} q_j^e \lambda_j \geq 2 \cdot k(S) \ \ \forall S \subseteq V_+$

の周回容量制約には，最適化計算の制約条件としてではなく，別途アルゴリズムを設定

Rounded Capacity Cuts Separation(M)

$z = \min \sum_{e \in E} \bar{x}_{ij} w_{ij}$ 出入り辺の数の最小化

$s.t.$

$w_{ij} \geq y_j - y_i \ \ \forall \{i, j\} \in E$

$w_{ij} \geq y_i - y_j \ \ \forall \{i, j\} \in E$

切断集合に属するための条件

$\sum_{i \in V_+} d_i y_i \geq (M \cdot C) + 1$ M所与下でのy, Sの抽出

$y_0 = 0, \ \ y_j \in \{0,1\}, \ \ w_{ij} \geq 0$

（$z < 2(M+1)$　ただし，この条件も成立してはならない）

$\bar{x}$ ：分数解(固定)

y_i ：頂点i がSに属していれば1

w_{ij} ：辺ijが切断集合Sに属していれば1

M ：Sに必要な車両台数-1(0以上K-1以下)