Amir Ahmadi Javid, Nader Azad:
Incorporating location, routing and inventory decisions in supply chain network design,
Transportation Research Part E,
pp. 582–597, 2010.

今泉孝章 M2
理論輪読会 6/27(Fri)

発表概要

〇サプライチェーンにおける意思決定過程の説明

〇モデル化の前提となる仮定

〇モデルの目的関数の設定

〇厳密解法による最適値の求解

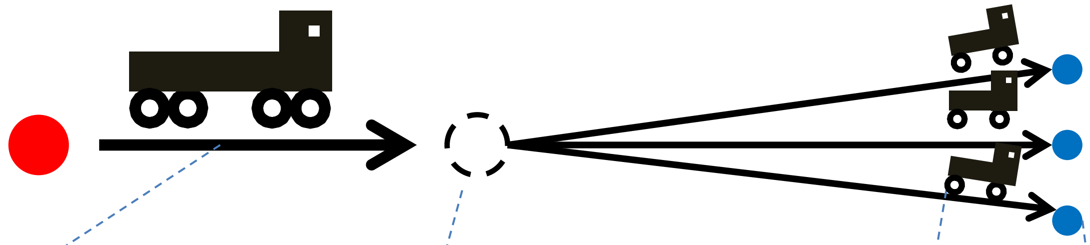
〇ヒューリスティクスによる近似解の求解

〇ヒューリスティクス手法の解の検証

導入

サプライチェーン

サプライヤー



顧客

ディストリビューションセンター(DC)

どれくらいの量を発注するか
→在庫発注量問題

どこに配置するか
→施設配置問題

どのように配送するか
→配送計画問題

どのDCに割り当てるか
→施設割り当て問題

サプライチェーンネットワーク全体を考慮すると様々な問題があるが，
従前の研究ではこれら全てを統合して表現したモデルは存在しない

モデル化の前提条件

1. 各顧客は正規分布に従う, 不確実な需要をもち顧客間の需要は独立である.
2. 全てのDCの考えられるキャパシティはわかっており, DC開設の場合はキャパシティに応じて固定費用がかかる.
3. オーダーごとの発注費用, 品目ごとの在庫保管費用がかかる.
4. 各DC j は在庫発注の際 (Q_j, R_j) 戦略に従う.
   在庫水準 が R_j を下回った場合に Q_j だけ発注する.
   このときDCはリードタイムの間, 品切れに備え安全在庫をもつ.
5. 車両のキャパシティと品目特性は同一である.

モデルの決定変数

DCのロケーション, キャパシティ, 顧客割り当て選択

いくつのDCをどこに開設するのか, 各DCのキャパシティはどれくらいか, 顧客をどのように割り当てるか.

顧客への配送ルート選択

割り当てられた顧客に対してどのように配送を行うか.

在庫発注量選択

どれくらい頻度で再注文をするか, どれくらいの安全在庫をもつのか.

目的関数の設定

〇DCの固定費用

$$\boxed{\sum_{j \in J} \sum_{n \in N_j} f_j^n U_j^n}$$

J :DCの開設候補ノード

N_j :DC j のキャパシティ

U_j^n :DC j がキャパシティ n で開設されていれば1, そうでなければ0の変数.

f_j^n :キャパシティ n のDC j の年間固定費用とオペレーティング費用の和

〇配送費用

$$\boxed{\sum_{v \in V} \sum_{k \in M} \sum_{l \in M} d_k R_{klv}}$$

V :車両の集合

M:顧客ノードとDC候補ノードの集合

R_{klv}: 車両 v での配送において, ノード k がノード l より先の場合1, そうでなければ0の変数.

目的関数の設定

〇在庫発注, 保管費用

$$p_j \frac{D_j}{Q_j} + h_j \frac{Q_j}{2} + h_j z_\alpha \sqrt{lt_j \sum_{k \in K} \sigma_k^2 Y_{jk}}$$

p_j :DC j の品目当たりの年間発注費用
D_j :DC j の年間期待需要量
Q_j : DCj の一回当たりの発注量
h_j : DCj の品目当たりの年間保管費用
α : 顧客の発注量をどの程度カバーするかという比率
z_α : 正規分布の左側パーセンタイル値. 満足率　$P(Z \le z_\alpha) = \alpha$
lt_j : DC j の年間のリードタイム
K : 顧客ノードの集合
σ_k^2 : 顧客 k の年間需要の分散値
Y_{jk} : 顧客 k がDC j に割り当てられていれば1, そうでなければ0

目的関数の設定

○在庫発注, 保管費用

$$\boxed{p_j \frac{D_j}{Q_j}} + h_j \frac{Q_j}{2} + h_j z_\alpha \sqrt{lt_j \sum_{k \in K} \sigma_k^2 Y_{jk}}$$

$$\text{品目当たりの発注費用} \times \frac{\text{年間期待需要量}}{\text{一回当たり発注量}} = \text{年間期待発注費用}$$

一年間の期待発注回数

p_j :DC j の品目当たりの年間発注費用
D_j :DC j の年間期待需要量
Q_j : DCj の一回当たりの発注量
h_j : DCj の品目当たりの年間保管費用
α : 顧客の発注量をどの程度カバーするかという比率
z_α : 正規分布の左側パーセンタイル値. 満足率 $P(Z \le z_\alpha) = \alpha$
lt_j : DC j の年間のリードタイム
K : 顧客ノードの集合
σ_k^2 : 顧客 k の年間需要の分散値
Y_{jk} : 顧客 k がDC j に割り当てられていれば1, そうでなければ0

目的関数の設定

○在庫発注, 保管費用

$$p_j \frac{D_j}{Q_j} + h_j \frac{Q_j}{2} + h_j z_\alpha \sqrt{lt_j \sum_{k \in K} \sigma_k^2 Y_{jk}}$$

品目当たりの年間保管費用 × 平均在庫量 = 年間保管費用

p_j :DC j の品目当たりの年間発注費用
D_j :DC j の年間期待需要量
Q_j : DCj の一回当たりの発注量
h_j : DCj の品目当たりの年間保管費用
α : 顧客の発注量をどの程度カバーするかという比率
z_α : 正規分布の左側パーセンタイル値. 満足率 $P(Z \le z_\alpha) = \alpha$
lt_j : DC j の年間のリードタイム
K : 顧客ノードの集合
σ_k^2 : 顧客 k の年間需要の分散値
Y_{jk} : 顧客 k がDC j に割り当てられていれば1, そうでなければ0

目的関数の設定

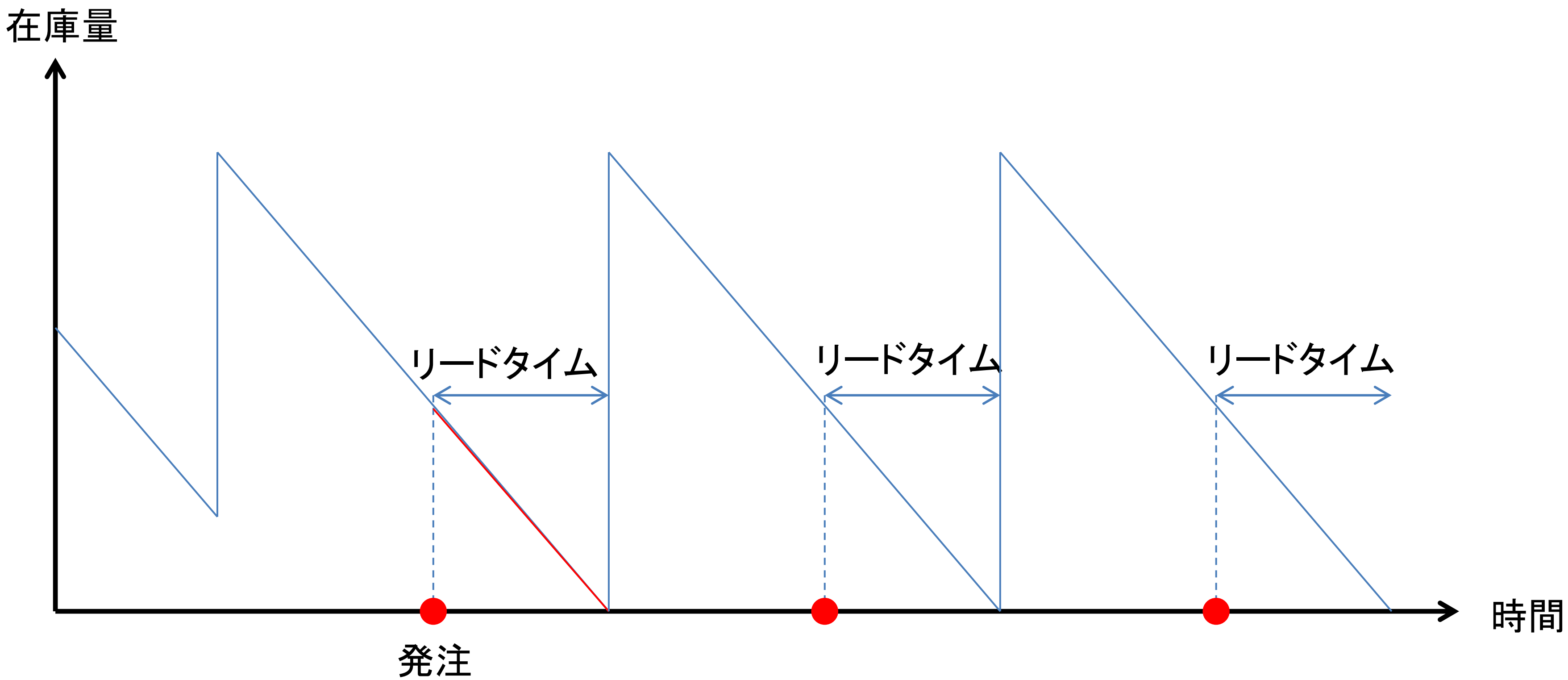
○在庫発注, 保管費用

$$p_j \frac{D_j}{Q_j} + h_j \frac{Q_j}{2} + \boxed{h_j z_\alpha \sqrt{lt_j \sum_{k \in K} \sigma_k^2 Y_{jk}}}$$

品目当たりの年間保管費用 × 満足率 × リードタイムの分散
=安全在庫保管費用

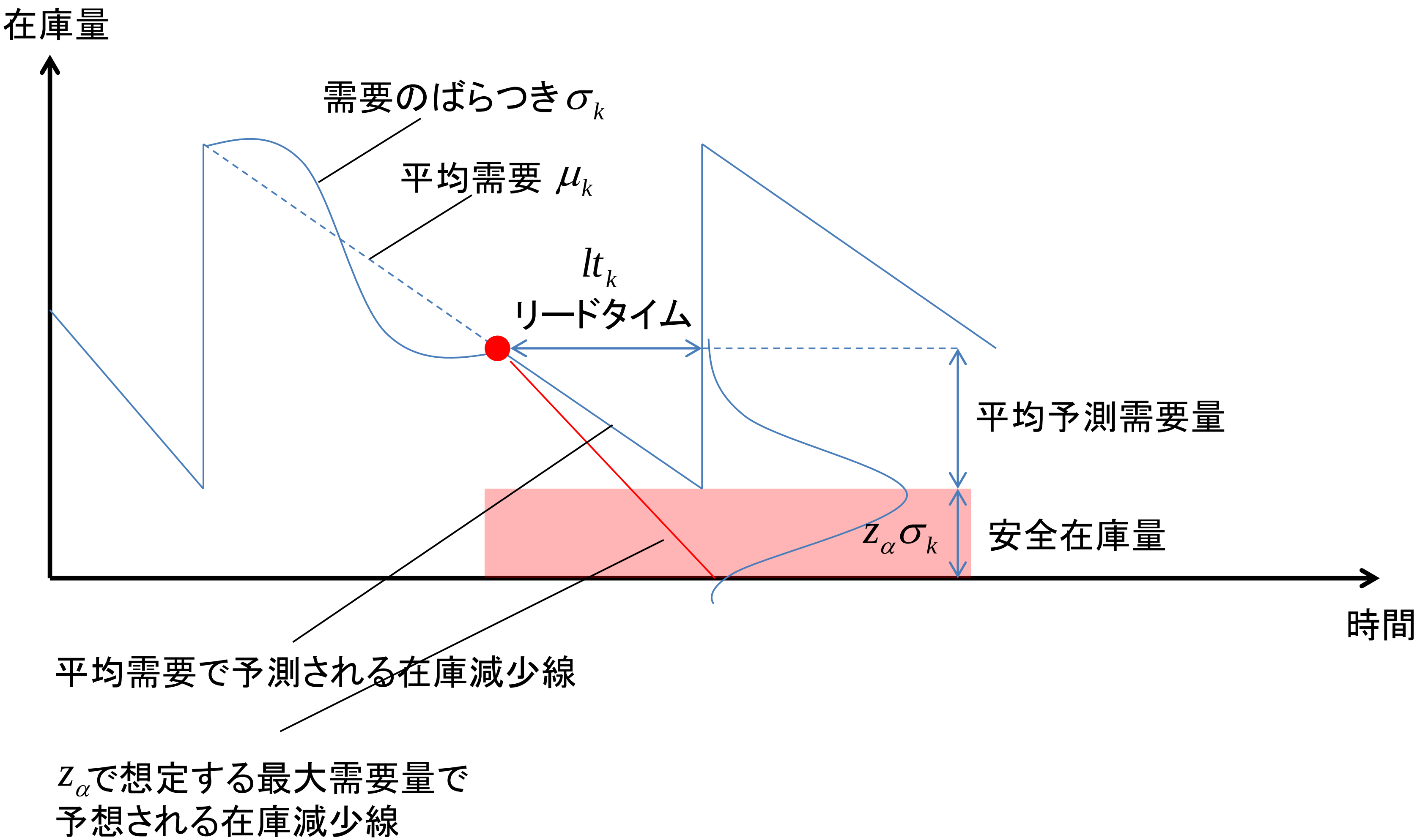
p_j :DC j の品目当たりの年間発注費用
D_j :DC j の年間期待需要量
Q_j : DCj の一回当たりの発注量
h_j : DCj の品目当たりの年間保管費用
α : 顧客の発注量をどの程度カバーするかという比率
z_α : 正規分布の左側パーセンタイル値. 満足率 $P(Z \le z_\alpha) = \alpha$
lt_j : DC j の年間のリードタイム
K : 顧客ノードの集合
σ_k^2 : 顧客 k の年間需要の分散値
Y_{jk} : 顧客 k がDC j に割り当てられていれば1, そうでなければ0

目的関数の設定



需要にばらつきがない場合: 赤の部分が確定的なので,
リードタイムを考慮してちょうど在庫が0になるときに発注すればよく
安全在庫は必要ない.

目的関数の設定



正規分布のどの程度の範囲をカバーするかが $z_\alpha\sqrt{\sigma_k^2}$ で, lt_k 個の正規分布が重なるので, その ときの安全在庫は $z_\alpha\sqrt{lt_k\sigma_k^2}$

目的関数の設定

〇輸送費用

$$\left(g_j + a_j Q_j\right)\frac{D_j}{Q_j}$$

g_j :サプライヤーからDC j までの一回当たりの固定輸送費用

a_j : サプライヤーからDC j までの品目当たりの輸送費用

輸送費用と在庫費用はパラメータを使って重み付けを行い目的関数にいれる

β :輸送費に関する重み係数

θ :在庫保管費に関する重み係数

目的関数の設定

$$
\min : \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q \sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl} R_{klv}
$$

$$
+\sum_{j\in J}\left[\left(\theta p_j + \beta g_j\right) \frac{\sum_{k\in K}\mu_k Y_{jk}}{Q_j} + \beta a_j \sum_{k\in K}\mu_k Y_{jk} + \frac{\theta h_j Q_j}{2} + \theta h_j z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}} \right]
$$

(1)

目的関数 = DCの固定費用 + 配送費用 + 発注費用 + 配送固定費用

+ 配送変動費用 + 在庫保管費用 + 安全在庫費用

制約条件

$$\sum_{v\in V}\sum_{l\in M} R_{klv} = 1 \quad \forall k \in K \qquad (2)$$

顧客は一つの車両によって配送される

$$\sum_{l\in K}\mu_l \sum_{k\in M} R_{klv} \quad \forall v \in V \qquad (3)$$

配送するとき車両のキャパシティを超えない

$$\sum_{l\in M} R_{klv} - \sum_{l\in M} R_{lkv} = 0 \quad \forall k \in M,\ \forall v \in V \qquad (4)$$

車両はノードに入ったら必ず同じノードから出る

$$\sum_{j\in J}\sum_{k\in K} R_{jkv} \le 1 \quad \forall v \in V \qquad (5)$$

一つのルートには一つのDCが含まれる.

$$\sum_{l\in M} R_{klv} + \sum_{l\in M} R_{jlv} - Y_{jk} \le 1 \quad \forall j \in J,\ \forall k \in K,\ \forall v \in V \qquad (6)$$

顧客 k がDC j が割り当てられるとき, 車両 v は DC j から出発して顧客 k を訪れる.

$$\sum_{n\in N_j} U_j^n \le 1 \quad \forall j \in J \qquad (7)$$

一つのDCに一つのキャパシティレベルが割り当てられる.

制約条件

$$\sum_{k\in K}\mu_k Y_{jk} \le \sum_{n\in N_j} b_j^n U_j^n \qquad \forall j\in J$$

(8)

需要はDCのキャパシティを超えない.

$$M_{kv} - M_{lv} + (B\times R_{klv}) \le B+1 \qquad \forall k,l\in K,\ \forall v\in V$$

(9)

サブツアー禁止制約?
(一つのルートには必ずDCを含む)

$$Y_{jk}\in\{0,1\} \qquad \forall j\in J,\ \forall k\in K$$

$$U_j^n\in\{0,1\} \qquad \forall j\in J,\ \forall n\in N_j$$

(10)

$$R_{klv}\in\{0,1\} \qquad \forall k,l\in M,\ \forall v\in V$$

$$M_{kv}\ge 0 \qquad \forall k\in K,\ \forall v\in V$$

(11)

非負制約

$$Q_j > 0 \qquad \forall j\in J$$

厳密解法

$$\min : \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q \sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl} R_{klv}$$

$$+\sum_{j\in J}\left[\left(\theta p_j + \beta g_j\right)\frac{\sum_{k\in K}\mu_k Y_{jk}}{Q_j} + \beta a_j \sum_{k\in K}\mu_k Y_{jk} + \frac{\theta h_j Q_j}{2} + \theta h_j z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}}\right] \qquad \textbf{(1)}$$

目的関数の中には, 決定変数は Q_j しかあらわれていない?????

また $Q_j > 0$ のとき式(1)は凸関数となるので, Q_j に関する一階微分=0とおくことで最適解Q^* を得る.

$$Q^* = \sqrt{\frac{2\left(\theta p_j + \beta g_j\right)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}} \qquad \textbf{(12)}$$

(1)へ代入して式(13)を得る. 式(13)を制約式(2)～(11)のもとで解く.

$$\min : \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q \sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl} R_{klv}$$

$$+\sum_{j\in J}\left[\sqrt{2\theta h_j\left(\theta p_j + \beta g_j\right)\sum_{k\in K}\mu_k Y_{jk}} + \beta a_j \sum_{k\in K}\mu_k Y_{jk} + \theta h_j z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}}\right] \qquad \textbf{(13)}$$

厳密解法

$$\min : \sum_{j \in J} \sum_{n \in N_j} f_j^n U_j^n + \beta q \sum_{v \in V} \sum_{k \in M} \sum_{l \in M} d_{kl} R_{klv}$$

$$+ \sum_{j \in J} \left[\sqrt{2\theta h_j (\theta p_j + \beta g_j) \sum\nolimits_{k \in K} \mu_k Y_{jk}} + \beta a_j \sum_{k \in K} \mu_k Y_{jk} + \theta h_j z_\alpha \sqrt{lt_j \sum\nolimits_{k \in K} \sigma_k^2 Y_{jk}} \right] \quad \textbf{(13)}$$

式(13)を連続変数へと緩和した関数は凹関数であり????, 大域的最適解を求めることはできない

そこで, バイナリー変数 Y_{jk} を Y_{jk}^2 と置き換えることで, 関数を凸関数にする.???????

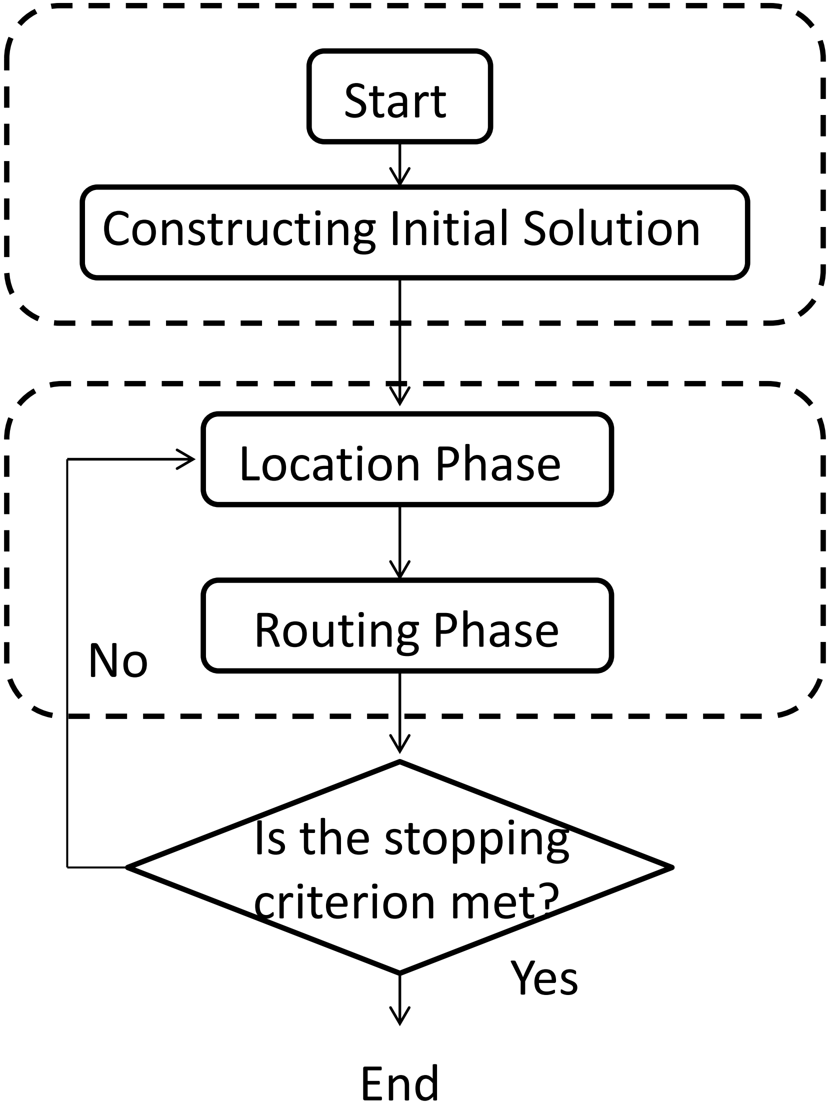
$$\min : \sum_{j \in J} \sum_{n \in N_j} f_j^n U_j^n + \beta q \sum_{v \in V} \sum_{k \in M} \sum_{l \in M} d_{kl} R_{klv}$$

$$+ \sum_{j \in J} \left[\sqrt{2\theta h_j (\theta p_j + \beta g_j) \sum\nolimits_{k \in K} \mu_k Y_{jk}^2} + \beta a_j \sum_{k \in K} \mu_k Y_{jk}^2 + \theta h_j z_\alpha \sqrt{lt_j \sum\nolimits_{k \in K} \sigma_k^2 Y_{jk}^2} \right] \quad \textbf{(14)}$$

式(14)のうち非線形なのは[]内の3つの要素である.2つ目の要素は連続変数に緩和すれば
二次関数なので凸. 1,3つ目の要素はユークリッドノルムの場合, 線形に書き直すことができる.
従って, 連続変数へ緩和すると凸計画問題となり, 結果として式(14)は混合整数凸計画問題となる.
→分枝限定法なので, サイズが小さければ厳密解法で解くことができる.

ヒューリスティクスによる解法

タブーサーチとアニーリングのハイブリット(TA-SA法)による解法.



Constructing Stage
実行可能解の生成

Improvement Stage

TA-SAによる解の探索

TS-SA(Tabu Search – Simulated Annealing)

○パラメータ

IT_0 :初期温度

CS :冷却スケジュール(冷却比率)

Ft :凍結温度(SAの収束判定に用いる最終温度)

MNT :各温度における解の最大受理回数(これを超えると冷却スケジュールにより温度を下げる)

nt :各温度における, 解の受理回数を保持しておく変数

X_0 :初期解

X :アルゴリズムにおける, 現在の解

X_{nh} :繰り返しの中での X の近傍の解

X_{best} :アルゴリズムで得られた最善の解

$C(X)$:解 X の目的関数値

TS-SA(Tabu Search – Simulated Annealing)

ステップ1. 初期解 X_0 を得る. $X_{best} = X_0$, $X = X_0$ とする.

ステップ2. X の近傍解 X_{nh} を生成する.

ステップ3. X_{nh}を生成した動きがタブーリストに入っていればステップ4へ,
そうでなければステップ5へ.

ステップ4. もし $C(X_{nh}) \le C(X_{best})$ ならば, $X = X_{nh}$, $X_{best} = X_{nh}$ とし, ステップ6へ,
そうでなければステップ2へ戻り, 別の動きによって近傍をつくり,
タブーリストを更新する

ステップ5. タブーリストを更新し, $\Delta C = C(X_{nh}) - C(X)$ とする.

5.1 $\Delta C \le 0$ ならば $X = X_{nh}$ とし, $C(X_{nh}) < C(X_{best})$ ならば $X_{best} = X_{nh}$

5.2 $\Delta C > 0$ ならば $y \leftarrow U(0,1)$, $y < e^{-\Delta C / T}$ ならば $X = X_{nh}$

ステップ6. 温度 T における解の受理回数が MNT より小さければステップ2へ,
そうでなければステップ7へ

ステップ7. $T = CS \times T$

ステップ8. 収束判定式 $T < FT$ が満たされていれば終了,
そうでなければステップ2へ

実行可能解の生成

ステップ1. 顧客をすべて集合 K' に入れる.

ステップ2.

2-1　K' から顧客をランダムにいくつか選択する

2-2　その顧客を K' から消去する.

ステップ3. DCをランダムに選択する.

ステップ4. 初めてDCが選択された場合は, キャパシティレベルを ランダムに割り当てる.

ステップ5. もし, DCの残りのキャパシティがステップ2で選択された顧客の需要より大きければDCに顧客を割り当てステップ6へ, そうでなければ, ステップ3へいき別のDCを選択する.

ステップ6. K' が空であればステップ7へ, そうでなければステップ2へ.

ステップ7. 開設されたDCごとに顧客配送のルートを近傍探索アルゴリズムによって生成する.

実行可能解の改善

解の改善ステージはロケーション(DCの変更)フェイズとルーティングフェイズ(配送ルートの変更)の二段階に分かれている.

〇ロケーションフェイズ

DCの数, 場所, キャパシティ, 割り当てる顧客を変化させることで解を改善する.
解の改善の際は以下の3つのMOVからランダムに一つ選び解を得ることとする.

MOV1

開設されたDCをランダムに一つ閉鎖し, そこに割り当てられていた顧客を残りのDCに割り当てし直す. もし, 残りのDCのキャパシティが顧客の需要を超えている場合は, キャパシティが需要を超えるように調整し, 近傍探索によりルートを再生成する.

MOV2

二つのDCをランダムに選択し, 顧客の割り当てをそのまま交換し, 近傍探索によりルートを再生成する. このときキャパシティは需要を超えないように調整される.

MOV3

開設しているDCを一つランダムに閉鎖し, 新たに別のDCをランダムに開設し, 顧客をそのまま割り当てる. このときキャパシティは顧客に合わせて更新される.

実行可能解の改善

○ルーティングフェイズ

現在の顧客への配送ルートを変更することにより解を改善する.

解の改善は以下の2つのMOVからランダムに一つ選び解を改善することとする.

MOV4

DCをランダムに一つ選択し, その中から車両(これが一つの配送ルートとなる)を一つ選択する. 選択されたルートを削除し, ルート上の顧客を別のDCに割り当てる. もし, 顧客の需要を満たすだけのキャパシティがなければ開設中のDCをランダムに一つ選択しキャパシティを増加させる. 最後に新しい顧客を加えて近傍探索によりルートを再生成する.(結局このMOVは解を上手く作れなかったので使わない)

MOV5

車両(ルート)をランダムに二つ選択し, 一方の車両が配送する一つの顧客ともう一方の車両が配送する一つの顧客を交換する. このときキャパシティが顧客需要を満たしうるかをチェックする.

解法の精度比較-数値計算例

〇厳密解とヒューリスティクス手法による解の比較

厳密　　　　ヒューリスティクス

Comparison of optimal and heuristic solution methods.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Optimal method		Heuristic method			
						Average cost	CPU time	Average cost	CV	CPU time	Gap (%)*
1	0.003	0.7	4	2	2	1812.2	9	1812.2	0	1	0.00
2	0.003	0.7	4	3	2	2023.4	29	2023.4	0	1	0.00
3	0.003	0.7	6	2	2	2412.3	150	2412.3	0	2	0.00
4	0.003	0.7	6	3	2	2598.1	379	2598.1	0	3	0.00
5	0.003	0.7	6	4	3	2741.8	587	2741.8	0	4	0.00
6	0.003	0.7	8	3	3	3011.6	19,874	3038.7	0	9	0.90
7	0.003	0.7	9	3	2	3215.1	31,458	3250.3	0	11	1.09
8	0.003	0.7	11	4	3	3684.7	12 h limit	3414.3	0.0001	16	−7.34
9	0.003	0.7	13	4	3	4176.5	12 h limit	3750.6	0.0001	20	−10.20
10	0.003	0.4	20	6	5	4462.8	24 h limit	3804.3	0.0001	34	−14.76
11	0.003	0.7	20	6	5	6800.6	24 h limit	5909.4	0.0001	35	−13.10
12	0.02	0.7	20	6	5	8039.2	24 h limit	7279.1	0.0001	34	−9.45
13	0.003	0.4	40	12	12	7011.2	60 h limit	6093.7	0.0001	55	−13.09
14	0.003	0.7	40	12	12	10393.7	60 h limit	9056.1	0.0001	55	−12.87
15	0.02	0.7	40	12	12	15078.3	60 h limit	13252.9	0.0001	56	−12.11
16	0.002	0.3	50	15	15	9023.4	72 h limit	7743.8	0.0001	66	−14.18
17	0.002	0.7	50	15	15	15006.2	72 h limit	12262.2	0.0001	65	−18.29
18	0.02	0.7	50	15	15	22843.1	72 h limit	18955.1	0.0001	66	−17.02

* Gap (%) = 100 × (Heuristic solution value − LINGO best solution value)/LINGO best solution value.

計算の規模が大きくなるほど厳密解とヒューリスティクスによる解の
誤差は大きくなっている.

解法の精度比較-数値計算例

〇TS-SAとSAの比較

Comparison of heuristic method based on hybrid TS–SA algorithm with heuristic method based on SA algorithm.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Based on hybrid algorithm			Based on SA algorithm		
						Average cost	CV	CPU time	Average cost	CV	CPU time
1	0.003	0.4	40	12	12	6093.7	0.0001	55	6171.2	0.0002	54
2	0.003	0.7	40	12	12	9056.1	0.0001	55	9154.1	0.0002	54
3	0.02	0.7	40	12	12	13252.9	0.0001	56	13427.5	0.0002	55
4	0.002	0.3	50	15	15	7743.8	0.0001	66	7864.2	0.0002	65
5	0.002	0.7	50	15	15	12262.2	0.0001	65	12469.1	0.0002	63
6	0.02	0.7	50	15	15	18955.1	0.0001	66	19226.8	0.0002	63
7	0.002	0.4	80	18	20	13230.4	0.0001	93	13547.7	0.0003	89
8	0.002	0.6	80	18	20	19133.9	0.0001	92	19635.6	0.0003	90
9	0.01	0.6	80	18	20	28124.3	0.0001	92	28839.7	0.0003	88
10	0.003	0.25	100	20	22	17943.3	0.0001	108	18571.2	0.0003	104
11	0.009	0.5	100	20	22	33516.8	0.0001	109	34775.9	0.0003	104
12	0.003	0.25	150	25	28	24219.5	0.0001	149	25188.6	0.0004	143
13	0.009	0.5	150	25	28	42577.1	0.0001	151	44374.2	0.0004	146
14	0.003	0.25	200	30	35	33847.1	0.0001	196	35446.3	0.0004	189
15	0.009	0.5	200	30	35	54730.4	0.0001	195	57546.1	0.0004	191
16	0.003	0.25	300	40	48	56277.6	0.0002	315	59285.9	0.0005	306
17	0.009	0.5	300	40	48	83964.7	0.0002	318	88947.4	0.0005	308
18	0.003	0.25	400	50	65	82751.2	0.0002	448	88085.6	0.0005	437
19	0.009	0.5	400	50	65	117271.8	0.0002	452	125307.5	0.0005	439

ハイブリッド型の方が平均費用が小さくなっており良い解を求めることができている

解法の精度比較-数値計算例

〇緩和問題の解と原問題の解の比較

Studying quality of lower bound presented in Section 3.2.4 for small instances.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Deviation (%)*
1	0.003	0.7	4	2	2	7.41
2	0.003	0.7	4	3	2	7.53
3	0.003	0.7	6	2	2	6.92
4	0.003	0.7	6	3	2	6.93
5	0.003	0.7	6	4	3	6.85
6	0.003	0.7	8	3	3	6.64
7	0.003	0.7	9	3	2	6.52
8	0.003	0.4	10	4	3	6.46
9	0.003	0.7	10	4	3	6.45
10	0.02	0.7	10	4	3	6.38

* Deviation (%) = 100 × (optimal solution value − lower bound value)/lower bound value.

解法の精度比較-数値計算例

○重みパラメータの違いによる比較

Impacts of weight factors, computational results for 40 customers.

No.	Input				Output		
	β	θ	# Potential DCs	# Vehicles	# Opened DCs	Average cost	CPU time
1	0.003	0.15	12	12	7	4305.1	54
2	0.003	0.4	12	12	6	6093.7	55
3	0.003	0.7	12	12	6	9056.1	55
4	0.003	0.7	12	12	6	9056.1	54
5	0.02	0.7	12	12	7	13252.9	56
6	0.04	0.7	12	12	7	18127.6	55

Impacts of weight factors, computational results for 300 customers.

No.	Input				Output		
	β	θ	# Potential DCs	# Vehicles	# Opened DCs	Average cost	CPU time
1	0.003	0.15	40	48	29	49195.1	316
2	0.003	0.25	40	48	27	56277.6	315
3	0.003	0.5	40	48	23	68461.3	317
4	0.003	0.5	40	48	23	68461.3	317
5	0.009	0.5	40	48	25	83964.7	318
6	0.02	0.5	40	48	28	99761.2	318

輸送の重み β が大きいとDCの開設数は増加する.
在庫の重み θ が大きいとDCの開設数は減少する.

解法の精度比較-数値計算例

〇モデルの違いによる比較

Comparison of proposed model and Shen and Qi's model, computational results for 200 customers.

					Average cost		
No.	β	θ	# Potential DCs	# Vehicles	Proposed model	Shen and Qi's model	Improvement (%)
1	0.003	0.15	30	35	27933.4	31923.3	14.28
2	0.003	0.25	30	35	32933.2	36961.6	12.23
3	0.003	0.5	30	35	43274.6	47953.7	10.81
4	0.003	0.5	30	35	43274.6	47953.7	10.81
5	0.009	0.5	30	35	53375.9	60806.5	13.92
6	0.02	0.5	30	35	70149.8	83767.1	19.41

Shen and Qi's modelではロケーション-顧客割り当てと在庫しか考慮していない
→配送ルートの決定については最適化していない.
　その結果提案したモデルの王がコストが小さくなっているのがわかる.