Jiafu Tang, Yang Yu, Jia Li: An exact algorithm for the multi-trip vehicle routing and scheduling problem of pickup and delivery of customers to the airport, *Transportation Research Part E*, Vol. 73, pp. 114-132, 2015.

2016/4/27

理論談話会#1

M1 三木真理子

目次

- 導入
- TCO-SPモデルの記述
- アルゴリズムの紹介
- 数値計算例
- まとめ

目的
：計算は大変だけどやっぱり計算したいから
　がんばって工夫しよう！
という思いをもつこと

Introduction

サービス産業

：operation costとcustomer satisfactionが大事

→交通サービスにおいても同じ

■本論文の検討対象

航空会社による，自宅から空港までの送迎サービス

■既往研究

・single-trip(デポ→顧客→空港)のみを扱う

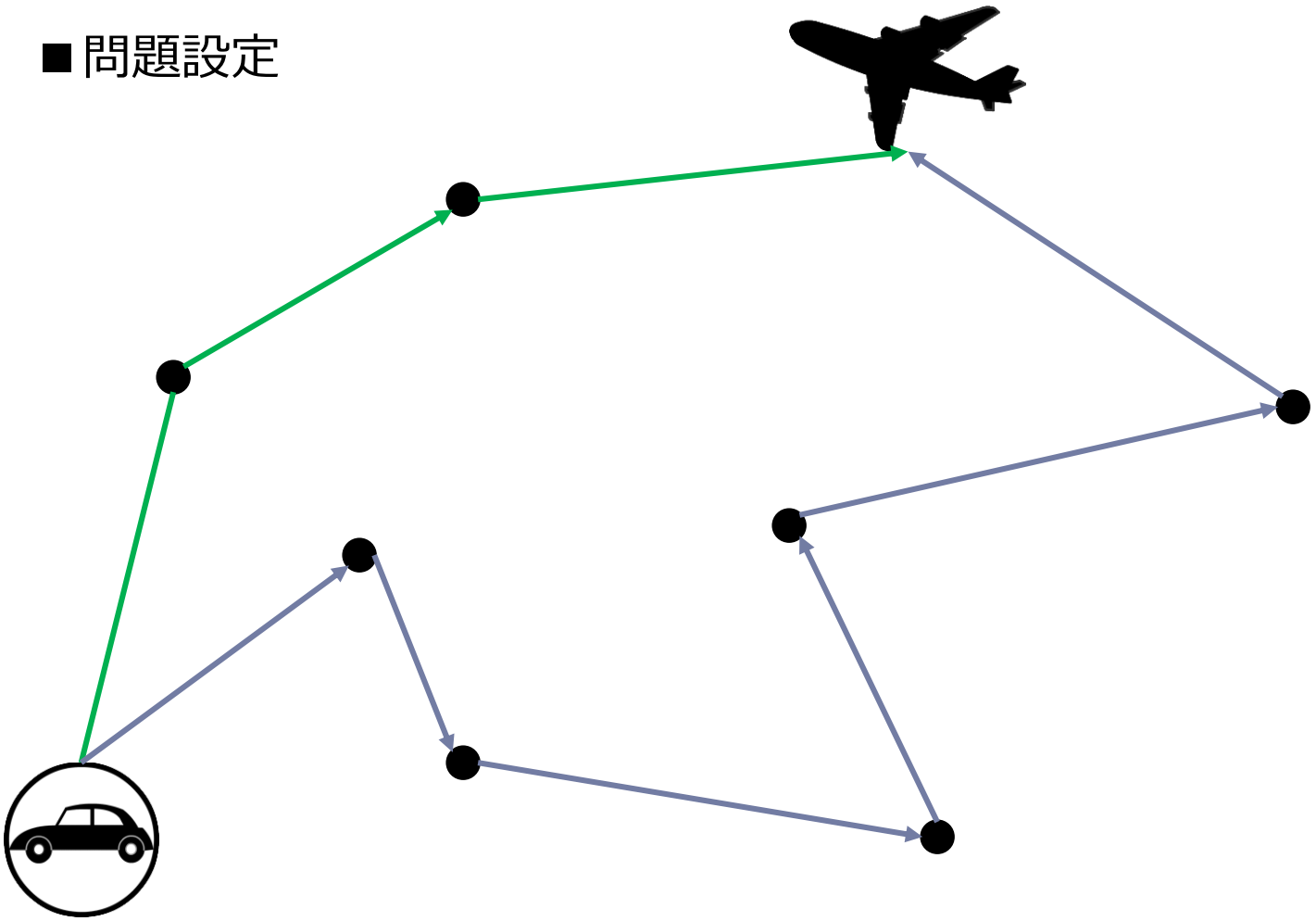
■本論文

・multi-trip(デポ→顧客→空港→顧客・・・→空港)を扱う

：MTM-D2PDCA
(multi-trip mode door-to-door picking up and delivering customers to the airport)

Introduction

■問題設定



Introduction

■MTM-D2PDCA

：Vehicle routing problem (VRP)の一部

VRPについて

・目的関数

$\min$(運行コスト,車両維持費, 遅れ時間etc)

・制約条件

車両1台あたりの運行時間上限

顧客のtime-window制約

・解法

ヒューリスティクスがメイン.

厳密解を求めるには問題の規模の限定が必要

Introduction

■MTM-D2PDCAの特徴

①すべての車両が同一ノード(デポ)から出発し，
　顧客ノードと空港ノードを往復してデポに戻る

②車両容量は小さい(自動車を想定)

③車両の運行時間に上限を設ける

④顧客の希望到着時刻を叶える

⑤顧客の乗車時間に上限を設ける

→VRPベースの定式化で最適解を得るのは困難.

→trip-chain-oriented set-partitioning(TCO-SP)model
　をベースに定式化を行い，厳密に解く

TCO-SP modelの定式化

$$\min \sum_{k \in K} \left(c_d \times D_k + c_f \right) x_k$$

s. t.

$$\sum_{k \in K} a_{wk} \cdot x_k = 1 \ \forall w \in W$$

$$x_k = \begin{cases} 1 \ (if \ vehicle \ k \ is \ used) \\ 0 \qquad (otherwise) \end{cases}$$

$$a_{wk} = \begin{cases} 1 \ (w \in W \ is \ visited \ by \ vehicle \ k) \\ 0 \qquad (otherwise) \end{cases}$$

c_d :単位距離あたり運行コスト　　c_f :車両1台あたりの維持費

D_k :trip-chain k の総距離

Assumption

①デポと空港はそれぞれ1つ

②同質の車両が十分な数あり，
　どの車両も同じ速さで動く

③運転手の遅れ時間と待ち時間は無視する

④到着時刻に関するtime windowと乗車場所が同一の顧客を，同一の顧客ノードにいるとして扱う．
ただし，1つの顧客ノードにいる顧客の人数は，車両の容量を超えない．超える場合は，新たにノードを追加する．

⑤各トリップの走行時間は，車両の運行時間上限を超えない．

Description of the MTM-D2PDCA

- デポ(node0)にいる車両の集合をKとする
- 車両の容量Q，速さvとする
- 車両は顧客ノードの集合Wから顧客を拾い，空港ノード(node a)へ送り届ける.
- 車両の1日のスケジュールについて
  : (first trip)デポ→空港
  (subsequent trips) 空港→空港
- 車両の運行時間上限をVDとする
- 顧客の特徴量は，到着時刻と乗車場所で与える

Description of the MTM-D2PDCA

■顧客全員を輸送する条件下のコスト最小化問題

①どの顧客を同じ車両で輸送するか

②顧客を拾う順序は？

③車両のデポ出発時刻，顧客乗車時刻，空港到着時刻をいつにするか

■輸送のTW以外の制約

①同乗する顧客の人数は車両の容量を超えない

②顧客の乗車時間は上限を超えない

③各車両の運行時間は上限を超えない

時間制約と満足度の表現

t_w : node wの顧客が空港に到着する時刻

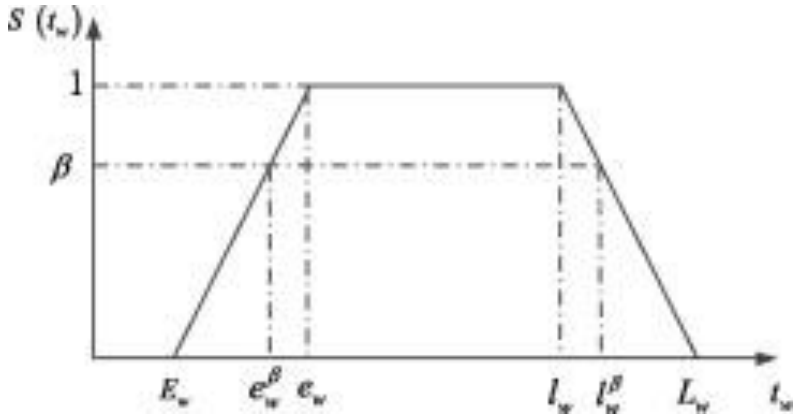
$[e_w, l_w]$: node wの顧客が空港に到着する時刻に対する緩いtime window
t_wがこの中にあれば，顧客の満足度は100%

$[E_w, L_w]$: node wの顧客が空港に到着する時刻に対する強いtime window
t_wがこれを破る場合，顧客の満足度は0%(unacceptable)

β : 顧客満足度の程度に関する閾値

■顧客満足度を表す区分的連続線形関数$S(t_w)$を定義

$$S(t_w) = \begin{cases} 1 & t_w \in [e_w, l_w] \\ \dfrac{E_w - t_w}{E_w - e_w} & t_w \in [E_w, e_w] \\ \dfrac{L_w - t_w}{L_w - l_w} & t_w \in [l_w, L_w] \\ 0 & t_w \notin [E_w, L_w] \end{cases}$$



時間制約と満足度の表現

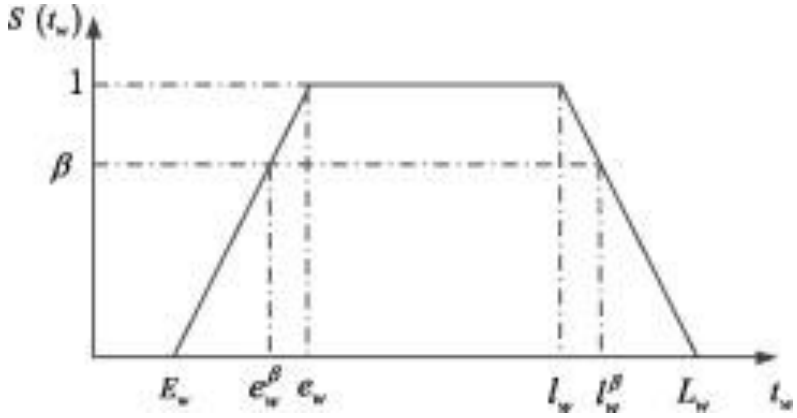
■顧客の満足度に関する制約

$$S(t_w) \geq \beta \quad \forall w \in W$$

これにより，βが決まると，顧客満足度の制約条件は，強いtime window制約に次のように変換できる．

$$e_w^\beta \leq t_w \leq l_w^\beta$$

$$S(t_w) = \begin{cases} 1 & t_w \in [e_w, l_w] \\ \dfrac{E_w - t_w}{E_w - e_w} & t_w \in [E_w, e_w] \\ \dfrac{L_w - t_w}{L_w - l_w} & t_w \in [l_w, L_w] \\ 0 & t_w \notin [E_w, L_w] \end{cases}$$



Definition

■Feasible trip of W_q：$0 \rightarrow S_{qh} \rightarrow a \ \ or \ \ a \rightarrow S_{qh} \rightarrow a$

デポ(0)または空港(a)を出発し，制約条件をすべて満たした上で，集合W_q内の顧客全員を空港に送るトリップ

■local optimal trip of W_q：$ls_q^0,\ ls_q^a$

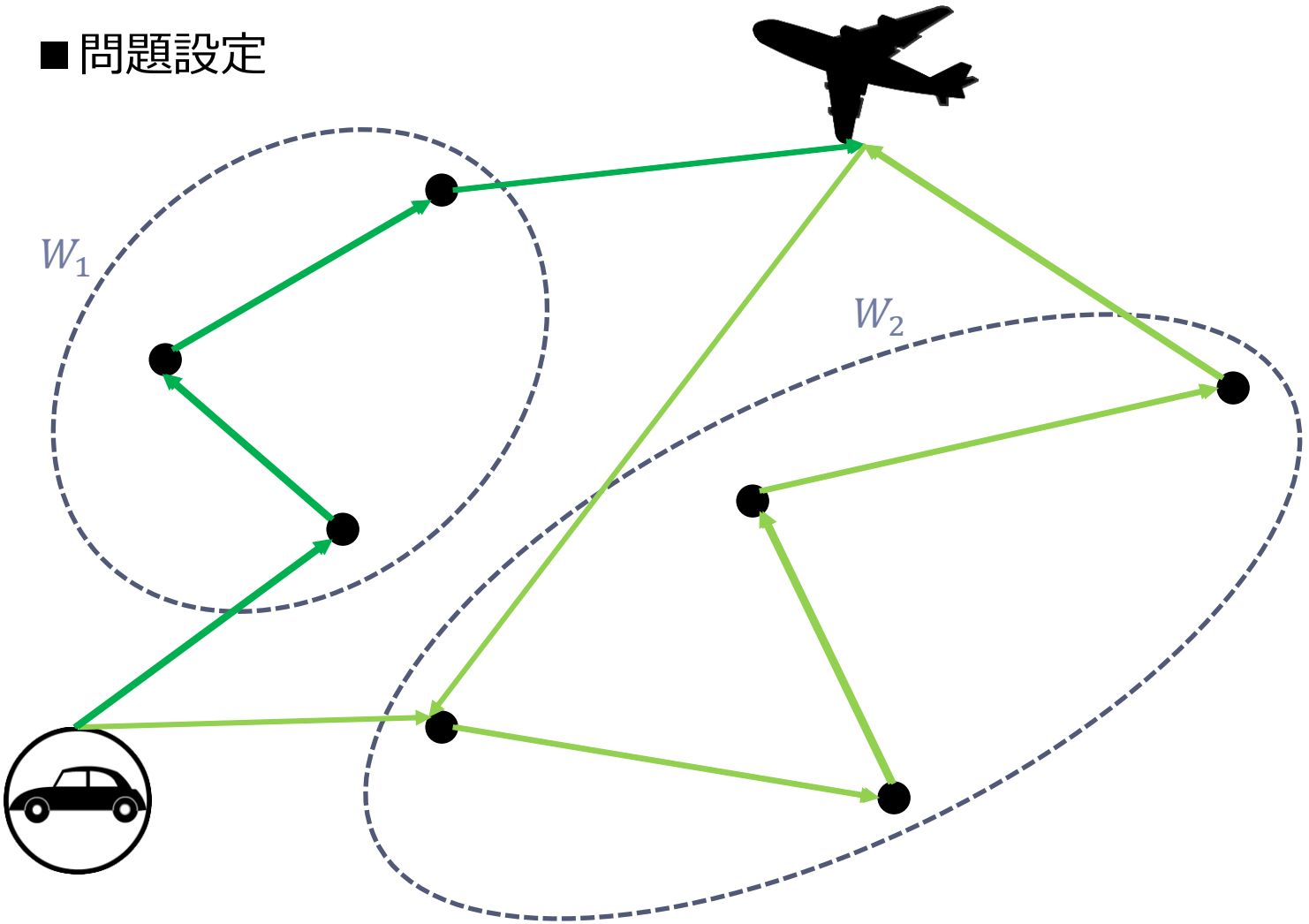
集合W_qについての実行可能トリップのうち，移動距離が最小となるトリップ．発ノードが2種類あるため，同一の集合W_qに関して2種類ある．

■Trip-chain ：$\langle ls_q^0 \rangle [ls_{q1}^a][ls_{q2}^a] \ldots [ls_{qr}^a], r \in N$

同一の車両によってなされる1つ以上のlocal optimal trip(s)の連なり．発ノードはデポとなる．

定義の確認

■問題設定



TCO-SP modelの定式化

$$\min \sum_{k \in K} \left(c_d \times D_k + c_f \right) x_k$$

$s.\ t.$

$$\sum_{k \in K} a_{wk} \cdot x_k = 1 \ \forall w \in W$$

$$x_k = \begin{cases} 1\ (if\ vehicle\ k\ is\ used) \\ 0 \qquad (otherwise) \end{cases}$$

$$a_{wk} = \begin{cases} 1\ (w \in W\ is\ visited\ by\ vehicle\ k) \\ 0 \qquad\qquad (otherwise) \end{cases}$$

c_d :単位距離あたり運行コスト　　c_f :車両1台あたりの維持費

D_k :trip-chain k の総距離

Local optimal trip数の性質

■Theorem 1

$|W|$:顧客ノード数，Q:車両容量とすると，local optimal trip数の上限は

$$2 \cdot \left(C_{|W|}^{1} + C_{|W|}^{2} + \cdots + C_{|W|}^{Q}\right), \qquad \text{ただし} C_{|W|}^{Q} = \frac{|W|!}{Q!\,(|W| - Q)!}$$

（証明）

上限になるのは，各顧客ノードに顧客が1人，任意のQ人を同じ車両に同乗させられるとき．

このとき，車両にn人同乗する場合のlocal optimal tripの上限は，

発ノードが空港の場合もデポの場合も$C_{|W|}^{n}$なのであわせて$2 \cdot C_{|W|}^{n}$.

車両容量制約より，$1 \leq n \leq Q$なので，local optimal tripの数の上限は

$$2 \cdot \left(C_{|W|}^{1} + C_{|W|}^{2} + \cdots + C_{|W|}^{Q}\right)$$

Trip chain数の性質

■Theorem 2

Local optimal trip数がnのとき，trip chain数の上限は

$$2^{\frac{n}{2}-2} \cdot n$$

（証明）

Local optimal trip数がn⇒デポ発が$\frac{n}{2}$，空港発が$\frac{n}{2}$ある．

Trip-chain内のtrip数がkのとき，

・最初のトリップ：$\frac{n}{2}$通り

・次以降の(k−1)トリップの組合せ：$C_{n/2-1}^{k-1}$

$1 \leq k \leq \frac{n}{2}$より，trip-chainの組合せ数は

$$\frac{n}{2} \cdot \sum_{k=1}^{\frac{n}{2}} C_{n/2-1}^{k-1} = \frac{n}{2} \cdot 2^{\frac{n}{2}-1} = 2^{\frac{n}{2}-2} \cdot n$$

Trip chain数の性質

■Theorem 3

Local optimal trip総数がn，trip-chainのトリップ数の上限がNのとき，trip-chainの総数は

$$C_{n/2}^{1}(C_{n/2-1}^{0} + C_{n/2-1}^{1} + \cdots + C_{n/2-1}^{N-1})$$

以上の定理により，trip-chain数は
実際にはかなり少ないことが予想できる

3. The exact algorithm

3段階からなる厳密解を求めるアルゴリズムを設計.
⇒3P-TCO-SP algorithm

1. Local optimal trips are obtained
2. All feasible trip chains are constructed
3. TCO-SP model is solved using CPLEX

Phase 1: generating local optimal trips

実行可能なすべてのトリップを比較することでlocal optimal tripsを生成する.

[決定変数]

- どの顧客を一緒に送っていくか
- どの順に顧客ノードを回るか
- 車両が各ノードに到着する時刻はいつか

[トリップの実行可能条件]

- 同乗する顧客の総数が車両容量を超えない
- 顧客の乗車時間が閾値を超えない
- 顧客の空港到着時刻に関するtime windowsは重なっていなければならない(must overlap)

Phase1 generating local optimal trips

Local optimal tripsの生成を2段階に分けて行う

1. 顧客ノードの部分集合W_qそれぞれについて，
   集合内の顧客ノードの順番を入れ替えることで
   実行可能なすべてのトリップを作る
2. 部分集合内のすべての実行可能トリップについて
   その走行距離を比較し，local optimal tripsを得る

※単位距離あたりの運行コストは一定

⇒オペレーティングコスト最小となるトリップの必要条件
：走行距離最小のトリップであること

Phase1 generating local optimal trips

Input: the set of the customer nodes (W)
Output: the set of local optimal trips of W (LS)
Step (1) Initialization:
$SW \leftarrow \varnothing$
for all $w \in W$
do *TimeWindow*$(e_w, l_w, E_w, L_w, \alpha)$
Sort (e_w^{α}, w)
Step (2) Construction of all subsets of W in which the customers have the same arrival time at the airport:
Repeat
Choose $w \in W$
do $W_q \leftarrow \varnothing$
$W_q \leftarrow W_q \cup \{w\}$
$SW \leftarrow SW \cup W_q$
for all $u \in W/\{w\}$
If $(e_u^{\alpha} \leqslant l_w^{\alpha})$
$W_q \leftarrow W_q \cup \{u\}$
$SW \leftarrow SW \cup W_q$
$W \leftarrow W/\{u\}$

Reduction of W
$W \leftarrow W/\{w\}$
Until $W = \varnothing$

顧客ノードwの到着時刻区間を計算し，
到着時刻上限の早い順にwを並び替える

到着時刻区間の重なっているwをまとめて
部分集合Wqをつくる

Phase1 generating local optimal trips

Step (3) Construction of all feasible trips of W_q:

Repeat

Choose $W_q \in SW$

If (!$CapacityCheck(W_q)$)

$S_q \leftarrow ConstructSequence(W_q)$

If (!$DetourRestrictionCheck(S_q)$)

$SW \leftarrow SW/\{W_q\}$

Until $SW = \varnothing$

do $FT_q^o \leftarrow ConstructFeasibleTripsD(S_q)$

$FT_q^a \leftarrow ConstructFeasibleTripsA(S_q)$

Step (4) Selection of local optimal trips:

Repeat

Choose $ft_{qn}^o \in FT_q^o$

do $d_{qn}^o \leftarrow Distance\left(ft_{qn}^o\right)$

$D_q^o \leftarrow \varnothing$

$D_q^o \leftarrow D_q^o \cup \left\{d_{qn}^o\right\}$

$ls_q^0 \leftarrow CompareDistance\left(D_q^o\right)$

$LS^0 \leftarrow LS^0 \cup \left\{ls_a^0\right\}$

Reduction of FT_q^o:

$FT_q^o \leftarrow FT_q^o / \left\{ft_{qn}^o\right\}$

Until $FT_q^o = \varnothing$

部分集合Wqの順列を生成

デポ発のトリップのうち,
すべてを比較することで
Local optimal tripを探索

Phase1 generating local optimal trips

Repeat

Choose $ft_{qn}^{a} \in FT_{q}^{a}$

do $d_{qn}^{a} \leftarrow Distance\left(ft_{qn}^{a}\right)$

$D_{q}^{a} \leftarrow \varnothing$

$D_{q}^{a} \leftarrow D_{q}^{a} \cup \left\{d_{qn}^{a}\right\}$

$ls_{q}^{a} \leftarrow CompareDistance\left(D_{q}^{a}\right)$

$LS^{a} \leftarrow LS^{a} \cup \left\{ls_{q}^{a}\right\}$

Reduction of FT_{q}^{a}

Until $FT_{q}^{a} = \varnothing$

do $LS \leftarrow LS^{0} \cup LS^{a}$

Step (5) Output the set of local optimal trips of W (LS).

空港発のトリップのうち，
すべてを比較することで
Local optimal tripを探索

Phase2 building feasible trip-chains

Local optimal tripを組み合わせてtrip chainをつくる

[決定変数]

・1台の車両が担当するlocal optimal tripの組合せ

・各tripの出発時刻

[制約条件]

・どの顧客ノードも1台の車両によって訪れられる

・車両運行時間は上限を超えない

・各車両は次のトリップを開始するよりも前に空港に到着する

高速に構成するため，改良版ラベル修正法を用いる

ラベルの定義

■trip-chain k のラベルを以下に定義する

$$R_k = \{t_{ku}, t_{kd}, T_{ku}, T_{kd}, D_k, W_k, V_k\}$$

t_{ku}, t_{kd}: 出発時刻の上限と下限
T_{ku}, T_{kd}: 到着時刻の上限と下限
D_k: 走行距離

$$W_k = \{W_k^1, \dots, W_k^w, \dots, W_k^m\}$$

: 顧客ノード$w \in W = \{1,2, \dots, m\}$を含むなら$W_k^w = 1$, それ以外なら$0$

$$V_k = \{V_k^1, \dots, V_k^p, \dots, V_k^n\}$$

: $local\ optimal\ trip\ p$に到達可能なら$V_k^p = 1$, それ以外なら0

ラベルの修正

■trip-chain k= trip-chain k’（trip i’→･･･)+ trip i を考える

$$t_{ku} = \delta^{u}_{i's}\ , \qquad t_{kd} = \delta^{d}_{i's}\ , \qquad T_{ku} = \delta^{u}_{ia}\ ,$$

$$T_{kd} = \begin{cases} T_{k'd} + \frac{d_i}{v} & (T_{k'd} > \delta^{d}_{is}) \\ \delta^{d}_{ia} & (T_{k'd} \le \delta^{d}_{is}) \end{cases}, \qquad D_k = \sum_{p=i'}^{i} d_p$$

$\delta^{u}_{i's}, \delta^{d}_{i's}$: $trip\ i'$の出発時刻の上限，下限

$\delta^{u}_{ia}, \delta^{d}_{ia}$: $trip\ i$ の到着時刻の上限，下限

δ^{d}_{is}: $trip\ i$ の出発時刻の下限

$T_{k'd}$: $trip\ chain\ k'$の到着時刻の下限

d_i: $local\ optimal\ trip\ i$の走行距離

繋がらないトリップの識別

■trip j is unreachable for trip-chain k となるのは,

①jとkで共有する顧客ノードがある

②iの到着時刻の下限がjの出発時刻の上限よりも大きい

$$T_{kd} > \delta_{js}^{u}$$

③trip-chain kの出発時刻の下限とtrip jの到着時刻の上限の間が車両運行時間VDより大きい

$$\delta_{ja}^{u} - t_{kd} > VD$$

Phase2 building feasible trip-chains

Local optimal tripを組み合わせてtrip chainをつくる

[決定変数]

・1台の車両が担当するlocal optimal tripの組合せ

・各tripの出発時刻

[制約条件]

・どの顧客ノードも1台の車両によって訪れられる

・車両運行時間は上限を超えない

・各車両は次のトリップを開始するよりも前に空港に到着する

Phase2

Procedure *Label Algorithm* (*LA*)
Input: the set of local optimal trips (*LS*)
Output: the set of labels of all the feasible trip-chains (*L*)
Step (1) Initialization:
for all $k_0^i \in K_0^i$
do $L_i \leftarrow \varnothing$
$K^i = K_0^i$
$E = K^i$
Step (2) Exploration of the successors of each trip-chain:
Repeat
Choose $K^i \in E$
for all $k^i \in K^i$
for all $j \in Succ(k^i)$
do $F_{ij} \leftarrow \varnothing$
for all $R_k = \{t_{ku}, t_{kd}, T_{ku}, T_{kd}, D_k, W_k, V_k\} \in L_i$
do if $V_k^j = 0$
$Extend(R_k, j)$
If (NULL! = $Extend(R_k, j)$)
$F_{ij} \leftarrow F_{ij} \cup Extend(R_k, j)$
$L_j \leftarrow L_j \cup F_{ij}$
if L_j has changed
$K^j = K^j \cup \{k^j\}$
$E \leftarrow E \cup K^j$
$E \leftarrow E/\{K^i\}$
$L \leftarrow L \cup L_j$
Until $E = \varnothing$
Step (3) Output the set of labels of all the feasible trip-chains (*L*).

デポ発1トリップからなるTrip-chainを生成

トリップiで終わるtrip-chainについて，
Chainにないトリップjのすべてを探索.
繋がる場合はつなげて，
j終わりのtrip-chain集合に追加
探索対象からi終わりのtrip-chainを削除
→検索対象がなくなるまで反復

最適解に関する定理

■Theorem4

MTM-D2PDCAの最適解では，すべてのtrip-chainは
local optimal tripから構成される

（検証）

Computational results of *CLOT* and simple enumeration (*SE*) in phase 1.

Instance name	Algorithm	NT	NFTC	CT_1 (ms)	CT_2 (ms)	CT_3 (s)	MTOC
R40R80	CLOT	120	10850	37	51	0.2	2512.03
R40R80	SE	124	11572	40	84	0.41	2512.03
R50C100	CLOT	196	276551	86	1266	7.29	2704.75
R50C100	SE	204	287821	125	3499	24.98	2704.75
R120R300	CLOT	380	168240	283	4052	88.36	8965.9
R120R300	SE	390	173355	829	4215	85.5	8965.9

NT is the number of trips. In CLOT, the trip is all the local optimal trip. In SE, the trip is all the feasible trips.

アルゴリズムの効率性評価

■車両容量Qによる比較

定理1より，local optimal trip数の上限はQに関して指数的に増加する

→容量の大きな車両(6人以上)を用いる場合，
　local optimal tripは計算困難

Phase2 で，trip-chain数の上限はlocal optimal tripに関して指数的に増加するため，やはり容量が大きいと解を得るのが難しい.

4.Computational Result

①トリップチェインの数や特徴を把握する

②single-tripとmulti-tripをコスト面から比較する

③遅れペナルティや車両運行時間が最適解に与える影響を見る

■サンプルの構成要素

・顧客ノードの地理的な分布

・顧客ノード数

・time-window分布(Clustered(9h) or Random(14h))

・顧客数

The basic parameters of an instance.

Parameter name	Value
Maximum duration of time window	30 min
Speed of the vehicles	40
Ride time coefficient	1.5
Capacity of the vehicles (except the drivers)	4
Vertical and horizontal coordinates of the depot	(20,30)
Vertical and horizontal coordinates of the airport	(50,30)

計算結果–運行時間制約–

Number of trip-chains for various instances and constraints.

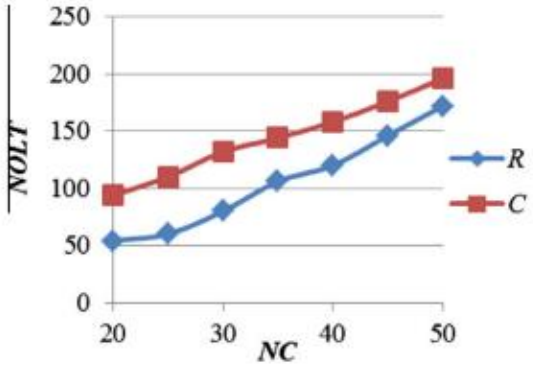
Instance name	NC	NLOT	NTC	NFTC without VD	NFTC	PI (%)
R20R40	20	54	7.82E+04	36608	450	98.77
R25R50	25	60	2.46E+05	191219	1242	99.35
R30R60	30	80	1.05E+07	1400951	2643	99.81
R35R70	35	106	1.26E+09	2308882	4080	99.82
R40R80	40	120	3.46E+19	–	10850	–
R45R90	45	146	3.45E+23	–	20061	–
R50R100	50	172	3.33E+27	–	19072	–
R20C40	20	94	1.39E+08	448	448	0.00
R25C50	25	110	2.61E+09	2214	2214	0.00
R30C60	30	132	1.42E+11	6029	6029	0.00
R35C70	35	144	1.24E+2	35246	35246	0.00
R40C80	40	158	1.54E+13	113594	113594	0.00
R45C90	45	176	3.87E+14	193088	167256	13.38
R50C100	50	196	1.38E+16	667678	276551	58.58

※R20C40：位置がランダム，顧客ノード20，TWクラスター，顧客数40
　顧客ノード1つあたり平均2人，$\beta = 70\%$，車両運行上限時間6h

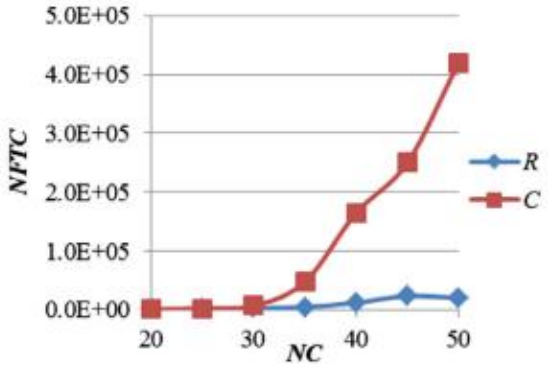
NC:顧客総数，NLOT:local optimal trip総数，NTC:trip-chain数上限
NFTC：feasible trip-chain数，PI：運行時間制約の影響度

NTCは大きいが，NFTCははるかに少ない．（制約の効果）
車両運行時間上限が大きくなるほど，NFTCは急激に大きくなる

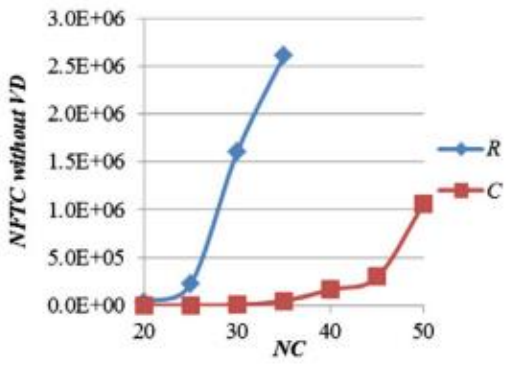
計算結果–顧客数とTW分布–



(a) *NLOT*



(b) *NFTC without VD*



(c) *NFTC.*

■VDの効果
・NFTC << NFTC without VD
■TW分布で比較(R or C)
・NLOTはCのほうがRより大きい
→Cのほうが同乗しやすいから
・NFTCwithoutVDはCのほうが圧倒的に少ない
・NFTCはRのほうが少ない
→TW分布がランダムの場合,
制約を満たすようなtrip-chainが作りにくい

計算結果–実行可能解と最適解–

Proportions of feasible and optimal trip-chains with various numbers of local optimal trips.

Instance name	NFTC	RFL (%)			AUV	ROL (%)		
		1–3	4–6	7–		1–3	4–6	7–
R20R40	450	62.2	37.8	0.0	5	60.0	40.0	0.0
R25R50	1242	43.4	56.6	0.0	6	66.7	33.3	0.0
R30R60	2643	36.7	62.7	0.6	6	33.3	66.7	0.0
R35R70	4080	31.5	67.6	0.9	9	77.8	22.2	0.0
R40R80	10850	20.9	77.3	1.8	9	55.6	44.4	0.0
R45R90	20061	15.2	78.2	6.6	9	44.4	55.6	0.0
R50R100	19072	18.7	79.4	1.9	11	72.7	27.3	0.0
R20C40	448	64.5	35.5	0.0	7	100.0	0.0	0.0
R25C50	2214	34.2	65.8	0.0	8	100.0	0.0	0.0
R30C60	6029	22.8	75.0	2.2	9	100.0	0.0	0.0
R35C70	35246	8.1	78.4	13.5	8	75.0	25.0	0.0
R40C80	113594	4.1	69.8	26.1	8	62.5	37.5	0.0
R45C90	167256	3.8	70.4	25.8	8	25.0	75.0	0.0
R50C100	276551	3.1	67.9	29.0	9	66.7	33.3	0.0

※VD=6h
AUV:使用車両台数
RFL:ある特定のトリップ数からなるfeasible trip chainの割合
ROL:ある特定のトリップ数からなるoptimal trip chainの割合

7以上のtripからなるtrip-chainはかなり少ない
→次善の解を見つけるときには,
　trip-chainの構成するtrip数を少なくしても十分だといえる

計算結果−single or multi−trip−

Results for various instances with $VD = 6$, $\beta = 70\%$.

Instance name	NLOT	NFTC	CT_1 (ms)	CT_2 (ms)	CT_3 (s)	MTOC	STOC
R20R40-	54	450	25	6	0.05	1354.41	1418.14
R25R50-	60	1242	24	9	0.08	1724.91	1991.67
R30R60-	80	2643	43	18	0.06	1799.22	2284.37
R35R70-	106	4080	46	25	0.13	2414.69	2475.01
R40R80-	120	10850	37	51	0.2	2512.03	2850.39
R45R90-	146	20061	50	99	1.2	2679.25	3250.99
R50R100-	172	19072	65	88	0.17	3470.75	3588.25
R20C40-	94	448	24	6	0.03	1764.07	1195.89
R25C50-	110	2214	36	11	0.05	2068.43	1501.31
R30C60-	132	6029	27	25	0.06	2291.64	1744.82
R35C70-	144	35246	27	93	0.4	2178.49	2265.21
R40C80-	158	113594	56	387	1.49	2281.77	2652.37
R45C90-	176	167256	54	635	3.32	2406.79	3048.71
R50C100	196	276551	86	1266	7.29	2704.75	3412.6

※single trip ： $c_d = 2, c_f = 10(RMB)$ taxiベース

multi trip : $c_d = 0.6, c_f = 200(RMB)$ rental car ベース

CT:計算時間(s),

MTOC：multi−tripの総コスト，STOC:single tripの総コスト

計算結果-single or multi-trip-

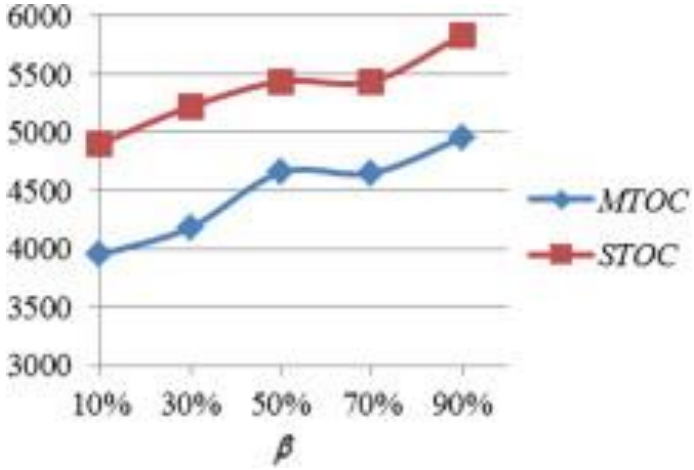
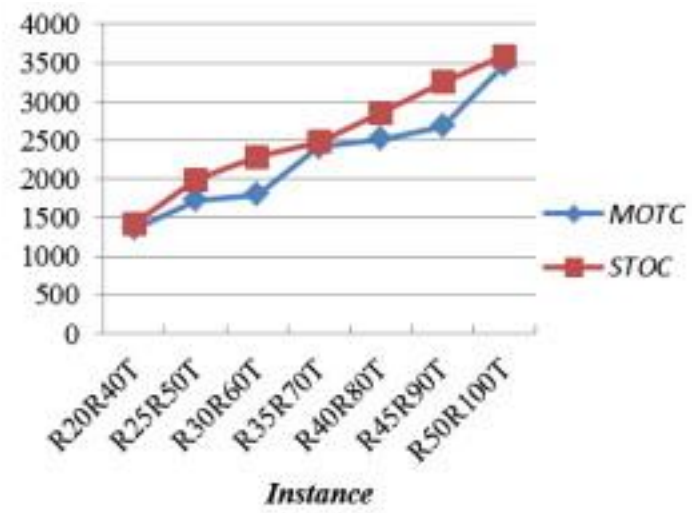
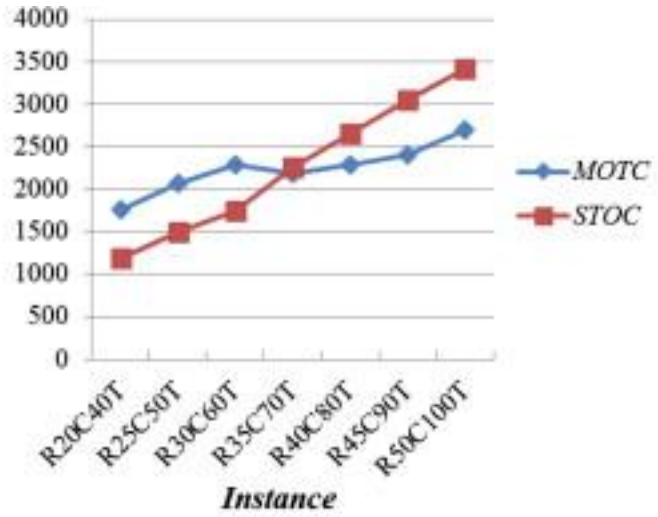


Fig. 6. Total operating costs for single-trip and multi-trip modes versus customer satisfaction threshold β.



(a) Total operating costs with *R-type DDTW*



(b) Total operating costs with *C-type DDTW*

計算結果–1顧客ノードあたりの顧客数–

Computational results of different number of the customers per customer point.

Instance name	Nc	NLOT	NFTC	CT_1 (ms)	CT_2 (ms)	CT_3 (s)	MTOC	STOC
R40R40	1	146	16186	148	127	0.36	2247.59	2464.02
R40R55	1.375	138	14244	177	111	0.47	2260.04	2543.12
R40R66	1.65	134	12743	84	103	0.42	2260.04	2614.58
R40R80	2	120	10850	37	51	0.2	2512.03	2850.39
R40R100	2.5	102	7140	49	54	0.2	2648.42	3172.00
R40R117	2.925	84	5323	69	38	0.17	2900.15	3547.25
R40R133	3.325	84	5161	33	37	0.14	2917.90	3532.09
R40R160	4	80	4945	36	34	0.11	3120.00	3671.05

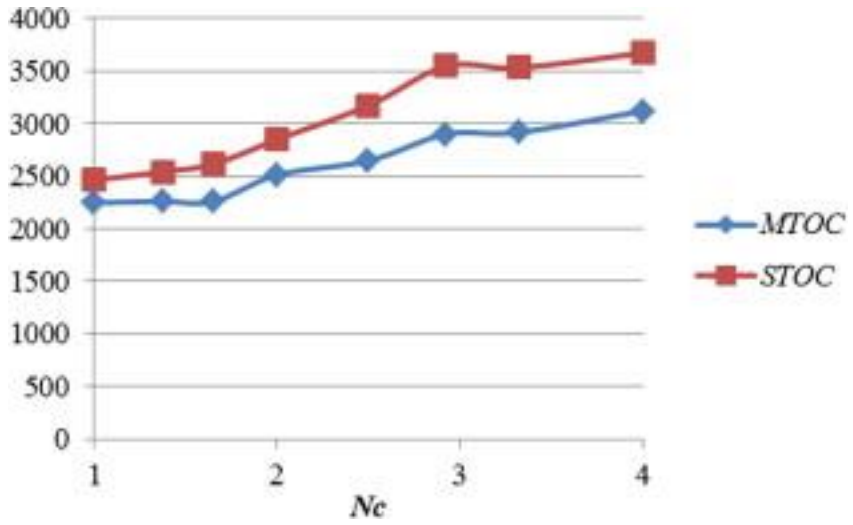


Fig. 8. Comparison of total costs between single-trip and multi-trip modes with different Nc.

顧客ノードあたりの顧客数が増えるほどMultiのほうが有利？

計算結果–計算時間の比較–

■CPLEXで厳密に解こうとしてみる

The running time, the lower and upper bounds and the optimality gaps value of different instances obtained from CPLEX.

Instance name	Running time (s)	Upper bound	Lower bound	Gap (%)
R20R20	2001	1348	942	30
R20R30	1263	1350	928	31
R20R40	954	1358	1010	26
R20R50	5974	1826	1643	10
R20R60	3954	1852	1405	24

$$x_{ijk} = \begin{cases} 1, trip\ j\ is\ finished\ directly\ after\ trip\ i\ by\ vehicle\ k \\ 0, \qquad\qquad otherwise \end{cases}$$

■本論文のアルゴリズムで計算

The results and running times from the proposed approach for the instances in Table 7.

Instance name	CT_1 (ms)	CT_2 (ms)	CT_3 (s)	MTOC
R20R20	142	14	0.05	1345.06
R20R30	42	10	0.02	1346.34
R20R40	25	6	0.05	1354.41
R20R50	24	8	0.02	1813.96
R20R60	39	7	0.03	1835.46

まとめ

■本論文の貢献

・trip-chain-oriented set-partitioning(TCO-SP) model を用いて，MTM-D2PDCAを簡便に厳密に解くアルゴリズムを設計

・trip-chain構成を実行不可能なものを除きながら行う改良ラベル修正法を提案.

・trip-chainの数や性質に関して考察した

・数値計算により，multi-tripによるオペレーションコストの減少を確認するとともに経営上の示唆を得た

■今後の展望

運転手の空港での待ち時間など，他の条件も考慮に入れることが望ましい.