アーク費用スケーリング法とMIP近傍探索法を用いた容量制約をもつネットワーク設計問題のマスヒューリスティクス

Matheuristics for Capacitated Network Design Problems using Arc Cost Scaling and MIP Neighbourhood Search

片山　直登　流通経済大学　流通情報学部

1　はじめに

ネットワーク設計問題 (Magnanti and Wong 1984, Wong 1984, Minoux 1989, Crainic et al. 2021) は，ネットワーク全体の費用を最小化するようにノードまたはアークを選択してネットワークを形成し，ネットワーク上のフローを求める問題である．

ネットワーク設計問題の中で，多品種の需要を考慮しアークが固定費用と処理容量をもつ問題は，容量制約をもつネットワーク設計問題 (CND) または固定費付きの多品種フローネットワーク設計問題とよばれ，NP 困難な問題 (Magnanti et al. 1986) であることが知られている．CND に対して，Wong (1984, 1985), Minoux (1989), Balakrishnan et al. (1997), Gendron et al. (1997), Crainic (2003), Ghamlouche et al. (2003), Costa (2005), Crainic et al. (2006), Hewitt et al. (2010), Yaghini and Rahbar (2012) など，今日までに多くの研究が行われてきた．

近年の研究として，Kazemzadeh et al. (2022) はノードに基づく定式化に対するラグランジュ緩和を用いた近似解法を提案した．また，Rocca et al. (2024) は教師あり学習によって訓練されたモデルを活用した近似解法を開発した．Shibasaki et al. (2024) はボリュームベースの分枝カット法とラグランジュ・ポンプ法を組み合わせた近似解法を提案した．

CND に対して，フロー費用，アーク費用やアーク容量を順次変更して近似解を求めるスケーリング法が開発されている．Crainic et al. (2004) は，アーク変数をフロー変数に変換した多品種フロー問題を用いるスロープスケーリング法を適用した．スロープスケーリング法は，多品種フロー問題のフロー解を用いてフロー費用をスケーリングした問題を反復的に解き，近似解を求める解法である．Gendron et al. (2018) は，スロープスケーリング法によりアーク変数の取りうる領域を限定し，この限定された問題を最適化ソルバーを用いて解くマスヒューリスティクスを提案した．Yaghini et al. (2024) は，線形緩和問題のフロー解を利用してアーク費用を修正した問題を反復的に解くことによりアーク変数の領域を

限定し，この限定された問題を最適化ソルバーを用いて解く2段階LPヒューリスティック法を提案した．これらの解法では，パスフローによる定式化ではなく，アークフローによる定式化を用いている．

一方，Katayama et al. (2009) はアーク容量をスケーリングすることにより，アーク変数の収束解を求めるアーク容量スケーリング法を開発した．Katayama (2015) はアーク容量スケーリング法によりアーク変数の取りうる領域を限定し，この限定された問題を局所分枝法により解くマスヒューリスティクスを提案し，Katayama (2020) は限定された問題をMIP近傍探索法により解くマスヒューリスティクスを提案した．これらの解法では，強い強制制約式を含むパスフローによる定式化の線形緩和問題を列生成法を用いて解いている．

本論文では，固定費用であるアーク費用に対する新たなスケーリング法を示し，MIP近傍探索法と組み合わせた高精度なマスヒューリスティクスを提案する．また，中規模のベンチマーク問題を用いて数値実験を行い，従来の解法と比較する．

2　問題の定式化

ノード集合 N，向きをもつアーク集合 A，ネットワーク上を流れる品種集合 K，ノード i，j 間にアーク (i,j) を設置したときに発生する固定費用であるアーク費用 f_{ij}，アーク (i,j) 上を流れる品種 k の単位あたりのフロー費用 c_{ij}^k，アーク (i,j) で処理できるフロー量の上限であるアーク容量 C_{ij}，品種 k の始点 O^k と終点 D^k 間を流れる品種 k の需要 d^k が与えられる．

アーク (i,j) 上を流れる品種 k のフロー量を表す連続変数であるアークフロー変数を x_{ij}^k，アーク (i,j) を設置するとき1，そうでないとき0である0–1離散変数であるアーク変数を y_{ij} とする．また，目的関数値を Φ とする．

このとき，CND のアークフローにより定式化された問題 $CNDA$ は次のように表すことができる．

$CNDA$

$$minimize \ \ \Phi = \sum_{(i,j)\in A}\sum_{k\in K} c_{ij}^k x_{ij}^k + \sum_{(i,j)\in A} f_{ij}y_{ij} \tag{1}$$

$$subject\ to \ \ \sum_{i\in N_n^+} x_{in}^k - \sum_{j\in N_n^-} x_{nj}^k = \begin{cases} -d^k & if\ n = O^k \\ d^k & if\ n = D^k \\ 0 & otherwise \end{cases} \quad \forall n \in N,\ k \in K, \tag{2}$$

$$\sum_{k\in K} x_{ij}^k \le C_{ij}y_{ij} \quad \forall (i,j) \in A, \tag{3}$$

$$x_{ij}^k \le d^k y_{ij} \quad \forall k \in K, (i,j) \in A, \tag{4}$$

$$x_{ij}^k \ge 0 \quad \forall k \in K, (i,j) \in A, \tag{5}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \tag{6}$$

(1) 式は目的関数であり，アークのフロー費用とアーク費用の総和を最小化する．(2) 式はアークフロー保存式であり，ノードに流入するフローと流出するフローの差が，品種 k の始点であれば $-d^k$，終点であれば d^k，その他のノードであれば 0 であることを表す．(3) 式は，アーク容量制約式であり，アーク (i,j) が設置されるときにアーク上を流れるフロー量の合計はアーク容量以下であり，設置されないときに 0 であることを表す．(4) 式は，アーク上の品種とその需要に関する強制制約式であり，アーク (i,j) が設置されるときに品種 k のフローが品種 k の需要量まで流れることができ，設置されないときには流れることができないことを表す．(5) 式はアークフロー変数の非負条件，(6) 式はアーク変数の 0–1 条件である．

ここで，アーク集合が A であるときの $CNDA$ を $CNDA(A)$，アーク集合が A でかつアーク費用が f であるときの $CNDA$ を $CNDA(A,f)$ とし，これらの線形緩和問題を $CNDAL(A)$ および $CNDAL(A,f)$ とする．また，(4) 式を含まないで定式化された問題を $CNDAW$，アーク集合が A であるときの $CNDAW$ を $CNDAW(A)$ とし，$CNDAW$ の線形緩和問題を $CNDAWL$ とする．

一方，品種 k の取りうるパス集合 P^k が与えられる．品種 k がパス p 上を流れる量を表す連続変数であるパスフロー変数を z_p^k とし，パス p にアーク (i,j) が含まれるとき 1，そうでないとき 0 を表す定数を δ_{ij}^p とする．

CND のパスフローにより定式化された問題 $CNDP$ は次のように表すことができる．

$CNDP$

$$minimize \quad \sum_{(i,j)\in A}\sum_{k\in K} c_{ij}^k \sum_{p\in P^k} \delta_{ij}^p z_p^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{7}$$

$$subject\ to \quad \sum_{p\in P^k} z_p^k = d^k \quad \forall k \in K, \tag{8}$$

$$\sum_{k\in K}\sum_{p\in P^k} \delta_{ij}^p z_p^k \le C_{ij} y_{ij} \quad \forall (i,j) \in A, \tag{9}$$

$$\sum_{p\in P^k} \delta_{ij}^p z_p^k \le d^k y_{ij} \quad \forall k \in K, (i,j) \in A, \tag{10}$$

$$z_p^k \ge 0 \quad \forall p \in P^k, k \in K, \tag{11}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \tag{12}$$

(7) 式は目的関数であり，パスフロー費用とアーク費用の総和を最小化する．(8) 式は，品種 k のパスフロー量の合計が需要量になることを表すパスフロー保存式である．(9) 式はアーク容量制約式であり，(10) 式は強制制約式である．(11) 式はパスフロー変数の非負条件であり，(12) 式はアーク変数の 0–1 条件である．

強制制約式である (4) 式または (10) 式を定式化に加えることで下界値が強化されるため，これらの強制制約式を含む定式化を強い定式化とよび，これらの強制制約式を定式化に含まない定式化を弱い定式化とよぶ．

ここで，アーク集合 A，パス集合 P，アーク費用 f が与えられたときのパスフローにより定式化された問題を $CNDP(A,P,f)$，アーク集合 A，パス集合 P，容量

費用Cが与えられたときの問題を$CNDP(A,P,C)$とする．また，これらの線形緩和問題を$CNDPL(A,P,f)$および$CNDPL(A,P,C)$とする．

3　スケーリング法

ここでは，CNDに対する従来の3つのスケーリング法を示した後，本研究で提案するアーク費用スケーリング法を示す.

3.1　スロープスケーリング法

スロープスケーリング法 (Crainic et al. 2004) は，弱い定式化の線形緩和問題$CNDAWL$においてアーク変数をフロー変数に変換した多品種フロー問題を生成し，このフロー問題を解き，得られたアーク変数解によりアーク費用を修正することを反復する方法である．

l回目の反復における多品種フロー問題$MCFA(\rho^l)$を示す.

$MCFA(\rho^l)$

$$minimize \quad \sum_{(i,j)\in A} \quad \sum_{k\in K} \left(c_{ij}^k + \rho_{ij}^l\right) x_{ij}^k \tag{13}$$

$$subject\ to \quad \sum_{i\in N_n^+} x_{in}^k - \sum_{j\in N_n^-} x_{nj}^k = \begin{cases} -d^k & if\ n = O^k \\ d^k & if\ n = D^k \\ 0 & otherwise \end{cases} \quad \forall n \in N,\ k \in K, \tag{14}$$

$$\sum_{k\in K} x_{ij}^k \le C_{ij} \quad \forall (i,j) \in A, \tag{15}$$

$$x_{ij}^k \ge 0 \quad \forall k \in K, (i,j) \in A. \tag{16}$$

ここで，ρ_{ij}^lはl回目の反復におけるアーク(i,j)のアーク費用に相当し，次のように定義する．

$$\rho_{ij}^l := \begin{cases} f_{ij} / \sum_{k\in K} \bar{x}_{ij}^k & if\ \sum_{k\in K} \bar{x}_{ij}^k > 0 \\ \rho_{ij}^{l-1} & otherwise. \end{cases} \quad \forall (i,j) \in A. \tag{17}$$

ここで，$\bar{x}_{ij}^k$は$l-1$回目の反復における多品種フロー問題$MCFA(\rho^{l-1})$のフロー変数の最適解である．また，次のように初期値ρ_{ij}^0を定義する．

$$\rho_{ij}^0 := f_{ij}/C_{ij} \quad \forall (i,j) \in A. \tag{18}$$

$MCFA^l$のフロー変数の最適解$\bar{x}$と容量Cから，次のようにCNDの実行可能解$\bar{y}$を算出する．

$$\bar{y}_{ij} = \left\lceil \sum_{k\in K} \bar{x}_{ij}^k / C_{ij} \right\rceil \quad \forall (i,j) \in A. \tag{19}$$

Gendron et al. (2018) は，スロープスケーリング法により探索するアークを限定し，2 つの限定したアーク集合を対象に最適化ソルバーで解を求めている．ここで，次のような 2 つのアーク集合を定義する．

$$A^0(\bar{y}) = \{(i,j) \in A | \bar{y}_{ij} = 0\}, \tag{20}$$

$$A^1(\bar{y}) = \{(i,j) \in A | \bar{y}_{ij} = 1\}. \tag{21}$$

続いて，次のような 2 種類の選択規則を定義し，いずれかの定義を用いる．

$$S^0 = A^0(\bar{y}) \; and \; S^1 = A^1(\bar{y}), \tag{22}$$

$$S^0 = A^0(\bar{y}) \; and \; S^1 = \varnothing. \tag{23}$$

これらの S^0 と S^1 を用いて，l 回目の反復において，

$$\rho_{ij}^l := \begin{cases} \infty & if \; (i,j) \in S^0 \\ 0 & if \; (i,j) \in S^1 \end{cases} \quad \forall (i,j) \in A \tag{24}$$

とし，問題の規模を縮小する．

$y_{ij} = 1, \forall (i,j) \in S^1$ を制約式として加え，アーク変数の領域を制限した問題 $CNDAW(A \backslash A^0)$ を最適化ソルバーで解く．

3.2 2 段階 LP ヒューリスティック法

2 段階 LP ヒューリスティック法 (Yaghini et al. 2024) は，アーク変数を用いた線形緩和問題 $CNDAL$ において，アーク変数解を用いてアーク費用を修正することを反復するスケーリングを行う方法である．

l 回目の反復における修正アーク費用 f_{ij}^l を次のように定義する．

$$f_{ij}^l := f_{ij}^{l-1} / \bar{y}_{ij} \quad \forall (i,j) \in A. \tag{25}$$

ここで，$\bar{y}_{ij}$ は $l-1$ 回目の反復における線形緩和問題 $CNDAL(A, f^{l-1})$ のアーク変数の最適解である．なお，この論文では $\bar{y}_{ij} = 0$ の場合の対処が記述されていない．また，次のように初期値 f_{ij}^0 を定義する．

$$f_{ij}^0 := f_{ij} \quad \forall (i,j) \in A. \tag{26}$$

反復的に得られる $CNDAL(A, f^l)$ の解で変数値が非負になったことのあるアーク集合を $\bar{A}$ とし，$CNDA(\bar{A})$ であるアーク変数の領域を制限した問題を最適化ソルバーで解く．

3.3 容量スケーリング法

容量スケーリング法 (Katayama et al. 2009) は，アーク変数の解とスケーリングパラメータに従ってアーク容量を変化させ，線形緩和問題 $CNDPL$ を反復的に解き，0–1 のアーク変数解を導出する方法である．

l 回目の反復におけるアーク容量を C^l，$CNDPL(A,P,C^{l-1})$ のアーク変数の最適解を $\bar{y}_{ij}$ とする．このとき，アーク (i,j) のアーク容量 C^l_{ij} を次のように定義する．

$$C^l_{ij} := \lambda C^{l-1}_{ij}\bar{y}_{ij} + (1-\lambda)C^{l-1}_{ij} \quad \forall (i,j) \in A. \tag{27}$$

ここで，λ はスケーリングパラメータであり，0 から 1 の定数とする．

0 に収束したアーク集合を A^0，容量スケーリングの際に列生成法により生成されたパス集合を $\bar{P}$ とし，アーク変数とパスを制限した問題 $CNDP(\bar{A},\bar{P},C)$ を最適化ソルバーで解く．

3.4 アーク費用スケーリング法

本研究で提案するアーク費用スケーリング法は，2 段階 LP ヒューリスティック法におけるアーク費用の修正に，1 回前の反復における修正アーク費用も考慮した方法である．

スロープスケーリング法や 2 段階 LP ヒューリスティック法では，アーク変数解やアークを流れるフロー変数解が 0 に近い場合，次の反復における修正アーク費用が極端に大きな値をとる．これを防ぐために，$l-1$ 回目の反復における修正アーク費用と新たに得られた修正アーク費用との凸結合によって，次のように l 回目の反復における修正アーク費用を変更する．

$$f^l_{ij} := \begin{cases} \alpha f^{l-1}_{ij}/\bar{y}_{ij} + (1-\alpha)f^{l-1}_{ij} & if\ \bar{y}_{ij} > 0 \\ \beta f^{l-1}_{ij} & otherwise \end{cases} \quad \forall (i,j) \in A. \tag{28}$$

ここで，α および β は，$0 < \alpha \leq 1$，$\beta \geq 1$ である定数，$\bar{y}_{ij}$ は $l-1$ 回目の反復における $CNDPL(A,\bar{P},f^{l-1})$ のアーク変数の最適解である．また，次のように初期値 f^0_{ij} を定義する．

$$f^0_{ij} := f_{ij} \quad \forall (i,j) \in A. \tag{29}$$

このように，緩和問題のアーク変数解が 1 の場合はアーク費用は変化させず，1 より小さな場合は値に応じてアーク費用を増加させる．また，アーク変数値が 0 の場合は，一定の比率でアーク費用を増加させる．

この修正アーク費用をもつ $CNDPL(A,P,f^l)$ を反復的に解くことにより収束解を求め，収束解をもとに近似解を求める．なお，パス変数は列生成法により生成し，必要な強制制約式を行生成法により生成する．

なお，0 に収束していないアーク変数の数が一定数以下となった場合は，アルゴリズムを終了する．

3.5 近似解法

アーク費用スケーリング法により得られた緩和解から，限定した分枝限定法とMIP近傍探索法により近似解を算出する．

3.6 限定分枝限定法

アーク費用スケーリングを一定回数だけ反復した後，0に収束したアーク変数および当該アークを流れるパス変数を定式化から削除した問題を作成する．この処理により，問題規模を縮小する．

アーク費用スケーリング法により求めた0に収束していないアーク集合を$\bar{A}$とする．また，列生成法によって生成されたパス集合で$\bar{A}$のみを流れるパス集合を$\bar{P}$とする．このとき，$CNDP(\bar{A}, \bar{P}, f)$は，相対的に小規模な問題になるため，計算時間の上限を設けて，最適化ソルバーで解くことにより近似解を算出し，得られた近似解を$\hat{y}$とする．続いて，アーク集合が$\bar{A}$に限定されたアークフロー変数を用いた強い定式化$CNDA(\bar{A})$に$\hat{y}$を代入し，アークフロー変数解と上界値を求める．

3.7 MIP近傍探索法

MIP近傍探索法(Katayama 2020)は，分枝限定法において0–1変数の探索範囲である近傍を縮小しながら，0–1解を求める近似解法である．定式化$CNDA(\bar{A})$において，限定分枝限定法で求めた近似解$\hat{y}$を$\bar{y}$の初期解として，MIP近傍探索法を適用する．

次のような制約式を追加して近傍の範囲を設定し，最適化ソルバーを用いて解を探索する．

$$\sum_{(i,j)\in\bar{A}|\bar{y}_{ij}=1} y_{ij} \geq L - M, \tag{30}$$

$$\sum_{(i,j)\in\bar{A}|\bar{y}_{ij}=1} y_{ij} \leq L - 1, \tag{31}$$

$$\Phi \quad < UB. \tag{32}$$

ここで，Lは現在の解における$\bar{y}_{ij} = 1$であるアーク変数の数，Mは削除的な近傍の範囲，UBは現在までの最良値である．(30)式は，現在の解で選択されているアークから高々M本までのアークをネットワークから取り除くことを表し，M近傍までを探索することを表す．(31)式は，現在の解を排除する式である．(32)式により，現在までの最良値UBをもつ解を排除する．

本研究では，次のような近傍制約式を追加する．

$$\sum_{(i,j)\in\bar{A}|\bar{y}_{ij}=0} y_{ij} \leq Q. \tag{33}$$

ここで，Q は追加的な近傍の範囲である．(33) 式は，現在の解で選択されていない高々Q 本までのアークをネットワークに追加することを表す．(30) 式と (33) 式により，M・Q 近傍までを探索する．

また，(31) 式の代わりに次式を用いることができる (Gendron et al. 2018)．

$$\sum_{(i,j)\in\bar{A}|\bar{y}_{ij}=0} y_{ij} + \sum_{(i,j)\in\bar{A}|\bar{y}_{ij}=1} (1 - y_{ij}) \quad \geq 1. \tag{34}$$

$CNDA(\bar{A})$ に，(30) 式から (32) 式の 3 本の式，または (30) 式から (33) 式の 4 本の式を追加し，最適化ソルバーを用いて，一定の計算時間内で解を求める．現在の解より良い解が求められた場合，現在の解を更新する．続いて，追加した制約式を削除して，更新された解に対応する制約式を追加し，近傍探索を繰り返す．

M・Q 近傍の探索において，問題が実行不可能または現在の解よりも良い解が無いと判断できた場合は，探索を終了する．また，計算時間の制限により，現在の解より良い解を算出することができない場合は，$M := \lfloor M/\gamma \rfloor$ として M を減少させて探索範囲を狭め，探索を繰り返す．ここで，γ は $\gamma > 1$ である M の変更基準である．同様に，$Q := \lfloor M/\sigma \rfloor$ として Q を減少させる．σ は $\sigma > 1$ である Q の変更基準である．

アーク費用スケーリング法と MIP 近傍探索法を組み合わせた解法のアルゴリズムを Algorithm1 に示す．

4 数値実験

容量制約をもつネットワーク設計問題で用いられている中規模のベンチマーク問題である C 問題の 37 インスタンス (Crainic et al. 2001) に対して，数値実験を行った．

数値実験で使用した機器等は以下の通りである．

- 使用 OS および言語：UBUNTU 24，C++
- 最適化ソルバー：Gurobi 11
- CPU AMD Ryzen9-3950X，16Cores，3.5GHz，RAM 64GByte
- 使用コア数：容量スケーリング 1 コア，MIP 近傍探索 16 コア

また，数値実験で使用した設定したパラメータは以下の通りである．

- スケーリングの反復回数 ITE：20
- スケーリングパラメータ α：0.025〜0.300
- スケーリング法の終了判定アーク数 $ArcNum$：200

- 近傍 M の初期値：50
- 近傍 M の変更基準 γ：2
- 近傍 Q：10
- 近傍 Q の変更基準 σ：1
- 1 回の近傍探索における最適化ソルバー計算時間の上限 T：100 秒，500 秒，1000 秒

近似解の誤差を算出するために，最適化ソルバーを用いて得られた下界値または最適値を使用した．これらの値は，Gurobi を用いて計算時間の上限を 250 時間として，$CNDA$ を解いて得たものである．

C 問題に対しては多くの研究が行われ，その結果が公開されている．ここでは，近年に公開された結果と比較する．

- Gurobi(G250H)Gurobi 250 時間
- 容量スケーリング・局所分枝法 (CSLB)(Katayama 2015)
- 遺伝的アルゴリズム・近傍探索法 (GANS)(Momeni and Sarmadi 2016)
- 反復 LP ヒューリスティック法 (ITLP)(Gendron et al. 2016)
- 切除平面・局所分枝法 (CPLB)(Yaghini et al. 2016)
- 並列局所探索法 (PALS)(Munguía et al. 2017)
- スロープスケーリング・反復 LP 法 (SILP)(Gendron et al. 2018)
- 容量スケーリング・MIP 近傍探索法 (CSMP)(Katayama 2020)
- ノードベースラグランジュ緩和法 (NBLR)(Kazemzadeh et al. 2022)
- 2 段階 LP ヒューリスティック法 (TWLH)(Yaghini et al. 2024)
- ボリューム・分枝カット法 (VBCT)(Shibasaki et al. 2024)

これらに加え，最適化ソルバー計算時間の上限を 100 秒，500 秒，1000 秒としたアーク費用スケーリング･MIP 近傍探索法 (FSM01, FSM05, FSM10) の結果を示す．

誤差 (Gaps) は「(各解法の上界値 − 下界値)/下界値」とし，平均誤差はこれらの平均値である．

表 1 に C 問題に対する上界値の平均誤差を示す．従来の研究では，遺伝的アルゴリズム・近傍探索法 (GANS) の平均誤差が 0.415%，反復 LP ヒューリスティック法 (ITLP) が 0.570%，切除平面・局所分枝法 (CPLB) が 0.157%，並列局所探索法 (PALS) が 0.276%，スロープスケーリング・反復 LP 法 (SILP) が 0.483% と大きく，0.15%を超えている．なお，ノードベースラグランジュ緩和法は C 問題の中の困難なインスタンスについて計算結果を示しているのみであるため，その他の誤差を 0.0%とし，全体の平均誤差を 0.371%とした．一方，容量スケーリング・局所分枝法 (CSLB) の平均誤差が 0.098%，容量スケーリング・MIP 近傍探索法 (CSMP) と 2 段階 LP ヒューリスティック法 (TWLH) が 0.080%と 0.081%，ボリューム・分枝カット法 (VBCT) が 0.074% と 0.1%以下となっている．

本研究のFSM01の平均誤差が0.089%，FSM05が0.077%，FSM10が0.074%であり，FSM10はボリューム・分枝カット法とともに最も平均誤差が小さい．また，Gurobiで250時間を使って解いた場合の平均誤差は0.071%であった．

表2に，各解法による個別のC問題の上界値を示す．なお，LB/OPTはGurobiで250時間を使って算出した下界値または最適値であり，太文字は最適値，斜体文字は最良値を表している．

Gurobiにより，37インスタンスの内の33インスタンスの最適値が確定しており，現在，最適値が求められていないものは4インスタンスに過ぎない．37インスタンスの内，遺伝的アルゴリズム・近傍探索法(GANS)が16インスタンス，反復LPヒューリスティック法(ITLP)が18インスタンス，切除平面・局所分枝法(CPLB)が20インスタンス，並列局所探索法(PALS)が17インスタンス，スロープスケーリング・反復LP法(SILP)が18インスタンスの最適値を算出している．また，容量スケーリング・局所分枝法(CSLB)が26インスタンスの最適値，容量スケーリング・MIP近傍探索法(CSMP)が27インスタンスの最適値，2段階LPヒューリスティック法(TWLH)が31インスタンスの最適値と1インスタンスの最良値，ボリューム・分枝カット法(VBCT)が30インスタンスの最適値と1インスタンスの最良値を算出している．

本研究のFSM01が27インスタンスの最適値と1インスタンスの最良値，FSM05が31インスタンスの最適値と2インスタンスの最良値，FSM10が32インスタンスの最適値と3インスタンスの最良値を算出している．FSM10が最も多くの最適値または最良値を算出し，2インスタンス以外の35インスタンスの最適値または最良値を算出している．なお，FSM01の2インスタンスでは4つの近傍制約式を用いている．

表3にC問題に対する平均計算時間を示す．従来の研究の計算時間は，各論文に掲載しているものであり，使用しているコンピュータが異なっているため，計算時間を直接比較することはできない．使用しているコンピュータは以下の通りである．

- Gurobi(G250H)：AMD Ryzen7-3950X，16Cores，3.5GHz，RAM64GB
- 容量スケーリング・局所分枝法(CSLB)：Intel Corei7-4770，4Cores，3.4GHz，RAM24GB
- 遺伝的アルゴリズム・近傍探索法(GANS)：2Cores，2.66GHz，RAM4GBs
- 反復LPヒューリスティック法(ITLP)：Intel Corei7-4900MQ，1Cores，2.8GHz，RAM16GB
- 切除平面・局所分枝法(CPLB)：Intel Core2-Duo，2Cores，2.53GHz，RAM4GB
- 並列局所探索法(PALS)：Intel Xeon-X5650 2CPU,6Cores，2.66GHz，RAM24GB，8台，96CPU並列接続
- スロープスケーリング反復LP法(SILP)：Intel i7-4900MQ，4Cores,2.80GHz，RAM16GB

- 容量スケーリング・MIP 近傍探索法 (CSMP)：AMD Ryzen-1800X，8Cores，3.6GHz，RAM16GB
- ノードベースラグランジュ緩和法 (NBLR)：Intel Xeon-X5675，6Cores，3.07GHz
- 2段階LP ヒューリスティック法 (TWLH)：Intel i7-6800k，6Cores，3.4GHz，RAM32GB
- ボリューム・分枝カット法 (VBCT)：Intel Xeon-E5–2687Wv3, 10Cores，3.1GHz，RAM30GB
- FSM01，FSM05，FSM10：AMD Ryzen9-3950X，16Cores，3.5GHz，RAM64GB

平均計算時間は，容量スケーリング・局所分枝法 (CSLB) が 6728 秒，反復 LP ヒューリスティック法 (ITLP) が 3600 秒，ノードベースラグランジュ緩和法 (NBLR) が 8058 秒，2 段階 LP ヒューリスティック法 (TWLH) が 6136 秒と計算時間が長く，1 時間を超えている．なお，ITLP と TWLH では，それぞれ 3600 秒，18000 秒の計算時間の上限を設けている．また，ボリューム・分枝カット法 (VBCT) は，計算時間の上限を 24480 秒としているため 16055 秒と最も長い．スロープスケーリング・反復 LP 法 (SILP) は 1075 秒，遺伝的アルゴリズム・近傍探索法 (GANS) が 292 秒，並列局所探索法 (PALS) が 153 秒，切除平面・局所分枝法 (CPLB) が 1849 秒，容量スケーリング・MIP 近傍探索法 (CSMP) が 467 秒と比較的短い計算時間である．

一方，本研究の FSM01 は 479 秒と短いが，FSM05 が 1720 秒，FSM10 が 2950 秒となっている．なお，Gurobi(G250H) は 250 時間=900000 秒を計算時間の上限としており，平均計算時間は 138849 秒である．また，表 4 に個別の C 問題の計算時間を示す．

5　おわりに

本研究では，アークに容量制約をもつネットワーク設計問題に対して，アーク費用スケーリング法と最適化ソルバーによる近傍探索法を組み合わせた高速で精度の高い MIP 近傍探索法を提案した．また，中規模のベンチマーク問題である C 問題に対して，数値実験を行い，従来の研究との比較を行った．

最新の研究である 2 段階 LP ヒューリスティック法 (TWLH) やボリューム・分枝カット法 (VBCT) と比べて，短時間の計算時間で，同水準の高精度の解を算出することができた．また，FSM10 が最も多くの最適値または最良値を算出し，37 インスタンスの内の 35 インスタンスの最適値または最良値を算出することができた．

本研究を含む，近年に提案された解法により，中規模のベンチマーク問題である C 問題の大半の最適解を求めることができるようになっている．このため，GT 問題のような大規模のインスタンスを用いて数値実験を行う必要がある．

本研究は科学研究費基盤研究 C(課題番号 23K04273) による成果の一部である．

参考文献

A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. Network design. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 311–334. John Wiley & Sons, New York, 1997.

A.M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32:1429 – 1450, 2005.

T.G. Crainic. Long-haul freight transportation. In R.W. Hall, editor, *Handbook of Transportation Science*, pages 451–516. Kluwer Academic Publishers, 2003.

T.G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems. *Discrete Applied Mathematics*, 112(1-3):73–99, 2001.

T.G. Crainic, B. Gendron, and G. Hernu. A slope scaling/Lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10:525–545, 2004.

T.G. Crainic, Y. Li, and M. Toulouse. A first multilevel cooperative algorithm for capacitated multicommodity network design. *Computers & Operations Research*, 33:2602–2622, 2006.

T.G. Crainic, M. Gendreau, and B. Gendron, editors. *Network Design with Applications to Transportation and Logistics*. Springer, Cham, Switzerland, 2021.

B. Gendron, T.G. Crainic, and A. Frangioni. Multicommodity capacitated network design. Technical Report CIRRELT-98-14, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Université de Montréal, 1997.

B. Gendron, S. Hanif, and R. Todosijević. An efficient matheuristic for the multicommodity fixed-charge network design problem. *IFAC-PapersOnLine*, 49(12):117–120, 2016.

B. Gendron, S. Hanafi, and R.Todosijević. Matheuristics based on iterative linear programming and slope scaling for multicommodity capacitated fixed charge network design. *European Journal of Operational Research*, 268:70–81, 2018.

I. Ghamlouche, T.G. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, 51:655–667, 2003.

M. Hewitt, G. L. Nemhauser, and M. W. P. Savelsbergh. Combining exact and heuristics approaches for the capacitated fixed charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325, 2010.

N. Katayama. A combined capacity scaling and local branching approach for capacitated multicommodity network design problem. *Far East Journal of Applied Mathematics*, 92(1):1–30, 2015.

N. Katayama. MIP neighborhood search heuristics for a capacitated fixed-charge network design problem. *Asia-Pacific Journal of Operational Research*, 37(3), 2020.

N. Katayama, M.Z. Chen, and M. Kubo. A capacity scaling procedure for the multi-commodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232(2):90–101, 2009.

M.R.A. Kazemzadeh, B. Tolga, T.G. Crainic, A. Frangioni, B. Gendron, and E. Gorgone. Node-based Lagrangian relaxations for multicommodity capacitated fixed-charge network design. *Discrete Applied Mathematics*, 308:255–275, 2022.

T.L. Magnanti and R.T. Wong. Network design and transportation planning : Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

T.L. Magnanti, P. Mireault, and R.T. Wong. Tailoring benders decomposition for uncapacitated network design. *Mathematical Programming Study*, 26:112–155, 1986.

M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360, 1989.

M. Momeni and M. Sarmadi. A genetic algorithm based on relaxation induced neighborhood search in a local branching framework for capacitated multicommodity network design. *Networks and Spatial Economics*, 16(2):447–468, 2016.

L. Munguía, S. Ahmed, D. A. Bader, G. L. Nemhauser, V. Goel, and Y. Shao. A parallel local search frame work for the fixed-charge multicommodity network flow problem. *Computers & Operations Research*, 77:44–57, 2017.

C.R.L. Rocca, J.F. Cordeau, and E. Frejinger. Combining supervised learning and local search for the multicommodity capacitated fixed-charge network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 192:10385, 2024.

R. Shibasaki, M. Baiou, F. Barahona, P. Mahey, and M. Souza. Volume-based branch-and-cut algorithm for large-scale fixed-charge multicommodity network design. *Annals of Operations Research*, 2024. doi: doi.org/10.1007/s10479-024-06303-y.

R.T. Wong. Introduction and recent advances in network design: Models and algorithms. In M. Florian, editor, *Transportation Planning Models*, pages 187–225. Elsevier Science, North Holland, Amsterdam, 1984.

R.T. Wong. Location and network design. In M. O'heEigeartaigh, J. Lenstra, and A. Rinnooykan, editors, *Combinatorial Optimization Annotated Bibliographies*, pages 129–147. John Wiley & Sons, New York, 1985.

M. Yaghini and M. Rahbar. Multicommodity network design problem in rail freight transportation planning. *Procedia - Social and Behavioral Sciences*, 43:728–739, 2012.

M. Yaghini, M. Karimi, M. Rahbar, and M. H. Sharifitabaro. A cutting-plane neighborhood structure for fixed-charge capacitated multicommodity network design problem. *INFORMS Journal on Computing*, 27(1):45–58, 2016.

M. Yaghini, K. Rezaei, and M. Karimi. A two-phase Lp-based heuristic algorithm for the fixed-charge capacitated multicommodity network design problem. *SSRN*, 4979547, 2024.

Algorithm 1: Arc Cost Scaling and MIP Neighborhood Search

Set A, $\bar{P}$, f, α, β, γ, σ, ITE, $ArcNum$, M, Q, T;
Solve $CNDPL(A, \bar{P}, f)$;
$l \leftarrow 0$; $f^l \leftarrow f$;
repeat
  $l \leftarrow l + 1$;
  Add paths to $\bar{P}$ by Column Genaration;
  Solve $CNDPL(A, \bar{P}, f^{l-1})$;
  Get $\bar{y}$ of $CNDPL(A, \bar{P}, f^{l-1})$;
  for $(i, j) \in A$ **do**
    if $\bar{y}_{ij} > 0$ **then**
      $f_{ij}^l \leftarrow \alpha f_{ij}^{l-1} / \bar{y}_{ij} + (1 - \alpha) f_{ij}^{l-1}$;
    else
      $f_{ij}^l \leftarrow \beta f_{ij}^{l-1}$;
    end
  end
  $n \leftarrow 0$;
  for $(i, j) \in A$ **do**
    if $\bar{y}_{ij} > 0$ **then**
      $n \leftarrow n + 1$;
    end
  end
until $l \geq ITE$ *and* $n \leq ArcNum$;
$\bar{A} \leftarrow \varnothing$;
for $(i, j) \in A$ **do**
  if $\bar{y}_{ij} > 0$ **then**
    $\bar{A} \leftarrow \bar{A} \cup \{(i, j)\}$;
  end
end
Solve $CNDP(\bar{A}, \bar{P}, f)$; Get $\hat{y}$;
Solve $CNDA$ with $\hat{y}$; Get UB;
repeat
  Add Neighborhood Equations to $CNDA(\bar{A})$ for $\hat{y}$;
  Solve $CNDA(\bar{A})$ within Time T;
  if $CNDA(\bar{A})$ *has no Feasible Solution* **then**
    break;
  else
    if *Solution* $\tilde{y}$ *of* $CNDA(\bar{A})$ *is Found* **then**
      Get UB; $\hat{y} \leftarrow \tilde{y}$;
    else
      $M \leftarrow \lfloor M/\gamma \rfloor$; $Q \leftarrow \lfloor Q/\sigma \rfloor$;
    end
  end
  Delete Neighborhood Equations;
until $M = 0$;
Return $\hat{y}, UB$;

表 1: Average Gap for C-Category Problems(%)

G250H	CSLB	GANS	ITLP	CPLB	PALS	SILP
0.071	0.098	0.415	0.570	0.157	0.276	0.483
CSMP	NBLR*	TWLH	VBCT	FSM01	FSM05	FSM10
0.080	0.371	0.081	0.074	0.089	0.077	0.074

*:Errors not Shown 0.0%

表 2: Results for C-Category Problems

N/A/K/F/C	LB/OP-250H	G250H	CSLB	GANS	ITLP	CPLB	PALS	SILP	CSMP	NBLR	TWLH	VBCT	FSM01	FSM05	FSM10
100/400/10/F/L	23949	**23949**	**23949**	**23949**	**23949**	**23949**	24022	**23949**	**23949**	-	**23949**	**23949**	**23949**	**23949**	**23949**
100/400/10/F/T	63753	**63753**	**63753**	64126	65247	64034	64207	65247	**63753**	63764	**63753**	**63753**	**63753**	**63753**	**63753**
100/400/10/V/L	28423	**28423**	**28423**	**28423**	**28423**	**28423**	28486	**28423**	**28423**	-	**28423**	**28423**	**28423**	**28423**	**28423**
100/400/30/F/L	49018	**49018**	**49018**	49058	**49018**	**49018**	**49018**	**49018**	**49018**	49115	**49018**	**49018**	**49018**	**49018**	**49018**
100/400/30/F/T	134708	*136231*	136803	137845	138784	136621	136861	139177	136250	137243	*136231*	136621	136636*	*136231*	*136231*
100/400/30/V/T	384802	**384802**	**384802**	**384802**	**384802**	**384802**	**384802**	**384802**	**384802**	-	**384802**	**384802**	**384802**	**384802**	**384802**
20/230/040/V/L	423848	**423848**	**423848**	**423848**	**423848**	**423848**	424075	**423848**	**423848**	-	**423848**	**423848**	**423848**	**423848**	**423848**
20/230/040/V/T	371475	**371475**	**371475**	**371475**	**371475**	**371475**	371573	**371475**	**371475**	-	**371475**	**371475**	**371475**	**371475**	**371475**
20/230/040/F/T	643036	**643036**	**643036**	**643036**	**643036**	**643036**	**643036**	**643036**	**643036**	-	**643036**	**643036**	**643036**	**643036**	**643036**
20/230/200/V/L	94213	**94213**	**94213**	**94213**	94218	**94213**	**94213**	**94213**	**94213**	94295	**94213**	**94213**	**94213**	**94213**	**94213**
20/230/200/F/L	137642	**137642**	**137642**	137854	138348	**137642**	**137642**	138169	**137642**	138118	**137642**	**137642**	**137642**	**137642**	**137642**
20/230/200/V/T	97914	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**	**97914**
20/230/200/F/T	135863	**135863**	**135863**	137449	136102	135991	135867	136513	**135863**	**135863**	**135863**	**135863**	**135863**	**135863**	**135863**
20/300/040/V/L	429398	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	-	**429398**	**429398**	**429398**	**429398**	**429398**
20/300/040/F/L	586077	**586077**	**586077**	**586077**	**586077**	**586077**	**586077**	**586077**	**586077**	-	**586077**	**586077**	**586077**	**586077**	**586077**
20/300/040/V/T	464509	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	-	**464509**	**464509**	**464509**	**464509**	**464509**
20/300/040/F/T	604198	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	-	**604198**	**604198**	**604198**	**604198**	**604198**
20/300/200/V/L	74811	**74811**	**74811**	75366	75003	74946	**74811**	74971	**74811**	75097	**74811**	**74811**	**74811***	**74811**	**74811**
20/300/200/F/L	115489	**115489**	115526	115963	116759	115574	115580	116375	115542	115671	**115489**	115525	115539	**115489**	**115489**
20/300/200/V/T	74991	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**	**74991**
20/300/200/F/T	107102	**107102**	107167	**107102**	**107102**	107284	**107102**	107298	**107102**	107760	**107102**	**107102**	**107102**	**107102**	**107102**
30/520/100/V/L	53958	**53958**	**53958**	53974	**53958**	**53958**	53978	**53958**	**53958**	**53958**	**53958**	**53958**	**53958**	**53958**	**53958**
30/520/100/F/L	93922	**93922**	93967	94094	93991	94043	93967	94066	**93922**	93985	**93922**	**93922**	**93922**	**93922**	**93922**
30/520/100/V/T	52046	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**
30/520/100/F/T	97098	**97098**	97107	98333	97711	97361	97862	97404	**97098**	97750	**97098**	**97098**	**97098**	**97098**	**97098**
30/520/400/V/L	112774	**112774**	**112774**	112871	112957	112786	112787	112974	112784	113201	**112774**	**112774**	112784	**112774**	**112774**
30/520/400/F/L	149093	**149093**	149151	150624	151134	149328	149677	149945	**149093**	149943	**149093**	149094	**149093**	**149093**	**149093**
30/520/400/V/T	114640	**114640**	**114640**	114884	114757	**114640**	**114640**	114798	**114640**	114741	114641	**114640**	**114640**	**114640**	**114640**
30/520/400/F/T	151753	*152354*	152477	154044	154859	152745	154137	153856	152451	152976	152476	152414	*152354*	*152354*	*152354*
30/700/100/V/L	47603	**47603**	**47603**	**47603**	**47603**	**47603**	**47603**	**47603**	**47603**	-	**47603**	**47603**	**47603**	**47603**	**47603**
30/700/100/F/L	59958	**59958**	**59958**	60184	60011	59987	60058	60049	**59958**	60056	**59958**	**59958**	60019	59995	**59958**
30/700/100/V/T	45872	**45872**	**45872**	45905	45908	45875	45879	45908	**45872**	45890	**45872**	**45872**	45873	**45872**	**45872**
30/700/100/F/T	54904	**54904**	54912	54925	**54904**	**54904**	**54904**	**54904**	**54904**	-	**54904**	**54904**	**54904**	**54904**	**54904**
30/700/400/V/L	97830	**97830**	97853	98747	98534	97960	98090	98385	97875	97997	**97830**	97845	97860	**97830**	**97830**
30/700/400/F/L	133335	134350	134554	136748	141170	135128	136257	139663	134511	135104	134666	*133976*	134511	134511	134511
30/700/400/V/T	95250	**95250**	**95250**	95704	95863	95321	95651	95773	95265	95457	95268	**95250**	95265	95265	95265
30/700/400/F/T	129396	*129816*	129990	130842	131814	130197	131104	131141	129869	130809	129869	129869	129869	129869	*129816*
Average	176612	**176708**	176744	177111	177309	176806	176947	177191	176721	-	176722	176713	176730	176716	176713

*: Four Neighborhood Constraints

表 3: Average Computation Time for C-Category Problems(Seconds)

G250H	CSLB	GANS	ITLP	CPLB	PALS*	SILP
138849	6728	292	3600	1849	153	1075
CSMP	NBLR	TWLH	VBCT	FSM01	FSM05	FSM10
467	8058	6136	16055	479	1720	2950

*: Time to Best Solutions

表 4: Computation Time for C-Category Problems(Seconds)

N/A/K/F/C	G250H	CSLB	GANS	ITLP	CPLB	PALS*	SILP	CSMP	NBLR	TWLH	VBCT	FSM01	FSM05	FSM10
100/400/10/F/L	9	78	1	3600	786	1	1079	23	-	1	1498	29	28	29
100/400/10/F/T	7537	10423	1	3600	822	53	2758	566	8387	2651	24480	574	2573	4597
100/400/10/V/L	0	2	33	3600	5	3	389	1	-	1	1422	1	1	1
100/400/30/F/L	331	2165	386	3600	367	29	610	397	1547	2032	11579	421	495	499
100/400/30/F/T	900000	18731	239	3600	377	79	1816	1118	10744	18000	24480	940	4217	5288
100/400/30/V/T	10	69	362	3600	31	42	735	40	-	2	24480	35	38	35
20/230/040/V/L	0	1	507	3600	11	2	1	1	-	1	3	1	1	1
20/230/040/V/T	0	3	1	3600	5	1	1	1	-	1	64	2	2	1
20/230/040/F/T	2	10	8	3600	38	10	4	4	-0	1	1288	6	6	5
20/230/200/V/L	952	6061	2	3600	1523	123	2413	520	7663	2343	11473	505	1743	656
20/230/200/F/L	2400	6723	2	3600	2085	144	313	648	9843	10250	19975	609	2220	3934
20/230/200/V/T	475	2706	379	3600	2483	52	240	495	1136	92	3165	462	331	236
20/230/200/F/T	9729	9707	432	3600	1297	240	947	852	10248	18000	24480	850	2862	6309
20/300/040/V/L	0	1	391	3600	4	1	169	1	-	0	4	0	0	0
20/300/040/F/L	3	8	461	3600	56	16	21	4	-	3	67	8	8	8
20/300/040/V/T	1	4	304	3600	3	23	0	1	-	0	134	1	1	1
20/300/040/F/T	1	4	319	3600	3	3	129	1	-	1	31	1	1	1
20/300/200/V/L	5895	10622	273	3600	4392	361	320	606	7728	3808	24480	711	2726	4798
20/300/200/F/L	134279	14044	369	3600	3279	250	2562	754	10800	18000	24480	915	3688	6532
20/300/200/V/T	221	2550	583	3600	1837	58	141	532	7836	75	4588	490	509	392
20/300/200/F/T	20312	8720	452	3600	6083	122	1256	586	4211	693	24480	585	2389	4308
30/520/100/V/L	71	1680	296	3600	329	20	1211	435	1152	409	24480	115	151	97
30/520/100/F/L	12230	7500	571	3600	2004	83	2755	659	8343	9061	24480	888	2854	4304
30/520/100/V/T	231	4049	25	3600	1850	32	1573	424	505	1267	24480	469	517	497
30/520/100/F/T	88120	15792	198	3600	978	158	2786	766	10795	5937	24480	1210	3232	5763
30/520/400/V/L	8589	9502	204	3600	3009	542	791	656	10143	9283	24480	769	3200	6212
30/520/400/F/L	836569	11131	278	3600	4231	463	434	731	10800	18000	24480	bar	3158	5650
30/520/400/V/T	4559	8263	441	3600	4103	461	2405	637	8532	1434	24480	639	2714	4855
30/520/400/F/T	900000	19594	335	3600	5238	288	1538	759	10760	18000	24480	776	3573	7072
30/700/100/V/L	11	64	536	3600	84	179	20	21	-	8	170	33	32	33
30/700/100/F/L	6496	7632	678	3600	1029	111	1549	515	10260	9165	24480	543	2325	4452
30/700/100/V/T	562	8807	381	3600	976	258	820	675	9577	1916	24480	554	2330	1605
30/700/100/F/T	1543	7640	191	3600	2109	173	1842	534	-	4590	24480	751	2280	4269
30/700/400/V/L	57640	13312	105	3600	1938	243	2714	826	9652	18000	24480	781	3290	6462
30/700/400/F/L	900000	16947	524	3600	2419	223	1066	874	10010	18000	24480	798	3592	6849
30/700/400/V/T	338640	10743	331	3600	8327	374	884	835	10800	18000	24480	757	2970	5478
30/700/400/F/T	900000	13656	191	3600	4297	428	1472	792	9973	18000	24480	863	3575	7929
Average	138849	6728	292	3600	1849	153	1075	467	8058	6136	16055	484	1720	2950

*: Time to Best Solutions