

## グラフに対する最大供給分割問題解法の性能評価

渡辺 一哉\* 高藤 大介\* 田岡 智志\* 渡邊 敏正\*

**[概要]** 需要点集合  $V_d$  と供給点集合  $V_s$  の和集合である点集合, および無向辺集合  $E$  を持つグラフ  $G = (V_d \cup V_s, E)$ , さらに各需要点  $v$  の需要量  $d(v) > 0$ , 各供給点  $u$  の供給量  $s(u) > 0$  が与えられたとする. 各供給点  $u_k$  ( $1 \leq k \leq |V_s|$ ) に対して,  $\sum_{v \in V_{d_k}} d(v) \leq s(u_k)$  かつ  $V_{d_k} \cup \{u_k\}$  の  $G$  での誘導部分グラフが連結であるような, 互いに素な需要点集合  $V_{d_k}$  の族  $\{V_{d_k} | 1 \leq k \leq |V_s|\}$  を求める. これらの需要点集合に属する需要点の需要量総和が最大となるような需要点集合族を求める問題を”グラフの最大供給分割問題”と呼ぶ. 本稿では, グラフの最大供給分割問題に対する既存の発見的解法と最適解法に着目し, 提案する改良手法と共に計算機上で実装して, これらの性能を計算機実験により比較評価する.

## Experimental Evaluation of Algorithms for Maximum-Supply Partitioning of a Demand-Supply Graph

Kazuya Watanabe\*, Daisuke Takafuji\*, Satoshi Taoka\* and Toshimasa Watanabe\*

**[abstract]** Let  $G = (V_d \cup V_s, E)$  be an undirected graph with two disjoint vertex sets:  $V_d$  of demand vertices and  $V_s$  of supply ones. Suppose that a positive value  $d(v)$  or  $s(v)$  is assigned to each demand or supply vertex  $v$ , respectively. Let  $V_{d_1}, \dots, V_{d_{|V_s|}}$  be a partition of  $V_d$  such that, for each  $u_k \in V_s$  ( $1 \leq k \leq |V_s|$ ), (i) the subgraph  $G[V_{d_k} \cup \{u_k\}]$  induced by  $V_{d_k} \cup \{u_k\}$  of  $G$  is connected, and (ii)  $\sum_{v \in V_{d_k}} d(v) \leq s(u_k)$ . The maximum-supply partition problem for a given demand-supply graph is to find such a partition with  $\sum_{1 \leq k \leq |V_s|} d(V_{d_k})$  being maximum, where  $d(V_{d_k}) = \sum_{v \in V_{d_k}} d(v)$ . This paper proposes improved versions of several known algorithms: one of them gives an optimum solution and all others produce heuristic ones. Their performance is evaluated based on experimental results.

### 1 はじめに

最大供給分割問題の入力グラフ  $G = (V_d \cup V_s, E)$  は, 需要点集合  $V_d$  と供給点集合  $V_s$  ( $V_d \cap V_s = \emptyset$ ) からなる点集合と無向辺集合  $E$  からなり, さらに各需要点  $v \in V_d$  には需要量  $d(v) (> 0)$ , 各供給点  $u \in V_s$  には供給量  $s(u) (> 0)$  が与えられている. 需要点集合  $V_{d_1} \subseteq V_d$  が供給点  $u_1 \in V_s$  から供給可能であるとは,  $V_{d_1} \cup \{u_1\}$  による  $G$  での誘導部分グラフ  $G[V_{d_1} \cup \{u_1\}]$  が連結でかつ,  $s(u_1) \geq \sum_{v \in V_{d_1}} d(v)$  のとき, およびそのときのみである.  $V_{d_1}$  の需要量を, それに含まれる点の需要量の総和とし,  $d(V_{d_1})$  と記す. 供給点集合  $V_{s_1} \subseteq V_s$  についても同様に  $s(V_{s_1})$  を定義する. グラフの最大供給分割問題とは, 次のように定義される: “需要点と供給点からなるグラフ  $G$  が与えられたとき, 各  $V_{d_k}$  が  $u_k \in V_s$  ( $1 \leq k \leq |V_s|$ ) の供給可能な需要点集合で,  $V_{d_{k_1}} \cap V_{d_{k_2}} = \emptyset$  ( $1 \leq k_1, k_2 \leq |V_s|, k_1 \neq k_2$ ), かつ  $\sum_{k=1}^{|V_s|} d(V_{d_k})$  が最大となる需要点集合の族  $\{V_{d_k} | 1 \leq k \leq |V_s|\}$  を求めよ.” 得られた解に対し,  $\sum_{k=1}^{|V_s|} d(V_{d_k})$  を ( $G$  の) 受給量と呼び,  $\sum_{k=1}^{|V_s|} d(V_{d_k})/d(V_d)$  を受給率と呼ぶ.

グラフの最大供給分割問題は, グラフが木であっても NP 困難であり [1], 表 1 に示す通り発見的解法が提案されている [2, 3, 4, 5]. また, グラフが木でありかつ各点の需要量, 供給量が全て整数の場合には, 擬多項式時間で最大受給量を求めるアルゴリズムが [6, 1] で提案されている. さらに木の場合の最大受給量を求める近似アルゴリズムも [6] で提案されている (表 2 参照). 表 2 等における近似比とは, (得られた解の受給率)/(最適解の受給率) の最小値である. 近似比が 1 である解は最適解である.

本稿では, これらのいくつかの発見的解法および最適解を求める解法に対し, 次のような改良版を提案する. 発見的解法 *Simple*[2] に対し, 計算時間の短縮を図った解法 *Simple<sub>all</sub>* を, *Fuzzy*[2] については無限ループに陥る可能性があったため, それを回避した解法 *Fuzzy<sub>m</sub>* を, 計算時間の短縮を図った解法 *Fuzzy<sub>l</sub>* ( $l = 1, \dots, 5$ ) を, 別の観点から計算時間の短縮を図った解法 *Fuzzy<sub>m</sub>*, *Fuzzy<sub>l</sub>* ( $l = 1, \dots, 5$ ) をそれぞれ提案する. 発見的解法 *GA*[5] についてはアルゴリズムの停止条件が明記されてなかったため, 2 種類の停止条件を追加して解法 *GA<sub>1</sub>*, *GA<sub>2</sub>* を提案する. 木の場合に最適解を求める解法 *KZN*[1] については, 最適解の受給率しか求まらないため, 解である供給

\*広島大学大学院工学研究科 / Graduate school of Engineering, Hiroshima University

可能な需要点集合族も求めるように修正する。

さらにこれらに対し、計算機実験による性能の比較評価を報告する。

表 1: 一般グラフに対する既存発見的解法のみまとめ

| 解法     | 文献  | 解法  | 文献  | 解法     | 文献  |
|--------|-----|-----|-----|--------|-----|
| Simple | [2] | GA  | [5] | Switch | [3] |
| Fuzzy  | [2] | HNN | [4] |        |     |

表 2: 木に対する解法のみまとめ

| 解法             | 計算時間                              | 近似比            | 文献  |
|----------------|-----------------------------------|----------------|-----|
| KZN            | $O(m_s^2( V_d  +  V_s ))$         | 1              | [1] |
| $IZN_{opt}$    | $O(F^2( V_d  +  V_s ))$           | 1              | [6] |
| $IZN_\epsilon$ | $O(( V_d  +  V_s )^5/\epsilon^2)$ | $1 - \epsilon$ | [6] |

注:  $m_s = \max\{s(u) | u \in V_s\}$ ,  
 $F = \min\{d(V_d), s(V_s)\}$ ,  $0 < \epsilon < 1$

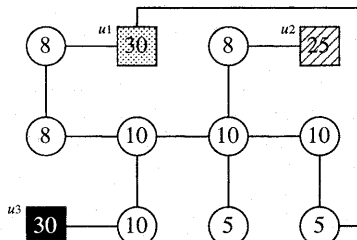


図 1: グラフ  $G = (V_d \cup V_s, E)$  および需要量と供給量 (○が需要点, □が供給点で, 内部の数値が各量を表す)。

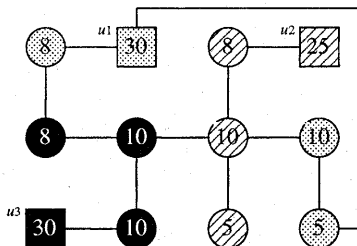


図 2: 図 1 のグラフに対する最適解。(各供給点により供給可能である需要点を同じ模様で示す。)

## 2 諸定義

グラフ  $G = (V_d \cup V_s, E)$  は,  $V_d \cap V_s = \emptyset$  なる需要点集合  $V_d$  と供給点集合  $V_s$  の和集合である点集合と無向辺集合  $E$  からなる。  $V = V_d \cup V_s$  とおく。各需要点  $v \in V_d$  には需要量  $d(v) (> 0)$ , 各供給点  $u \in V_s$  には供給量  $s(u) (> 0)$  が与えられている。点集合  $W \subset V$  に対し,  $N(W) = \{v \in V_d - W | (v, v') \in E \text{ なる } v' \in W \text{ がある}\}$  とおく。需要点集合  $V_{d_1} \subseteq V_d$  あるいは供給点集合  $V_{s_1} \subseteq V_s$  の需要量あるいは供給

量を, それに含まれる点の需要量, 供給量の総和とし, それぞれ  $d(V_{d_1}), s(V_{s_1})$  と記す。  $V' \subseteq V, E' \subseteq E$  なるグラフ  $G' = (V', E')$  を  $G$  の部分グラフと呼び,  $G' \subseteq G$  と表記する。需要点集合  $V_{d_1} \subseteq V_d$  が供給点  $u_1 \in V_s$  から供給可能であるとは,  $V_{d_1} \cup \{u_1\}$  による  $G$  での誘導部分グラフ  $G[V_{d_1} \cup \{u_1\}]$  が連結でかつ,  $s(u_1) \geq d(V_{d_1})$  のとき, およびそのときのみである。このとき,  $v \in V_{d_1}$  と  $u_1$  の組  $\{v, u_1\}$  を ( $u_1$  から) 供給可能な組と呼ぶ。さらに「需要点  $v \in V_{d_1}$  は供給点  $u_1$  により供給される」, または「供給点  $u_1$  は需要点  $v$  に供給する」という。さらに,  $\{u_1\} \cup V_{d_1}$  を供給点  $u_1$  の供給領域,  $p(u_1) = s(u_1) - d(V_{d_1})$  を供給点  $u_1$  の余剰供給力と呼び, 各  $v \in V_{d_1}$  に対して,  $u_1$  を  $v$  の供給源と呼ぶ。各供給点の供給領域の和集合を被供給領域と呼び, 供給源を持たない需要点を, 未供給需要点と呼ぶ。

グラフの最大供給分割問題とは, 次のように定義される: “需要点と供給点からなるグラフ  $G = (V_d \cup V_s, E)$  が与えられたとき, 各  $V_{d_k}$  がある  $u_k \in V_s (1 \leq k \leq |V_s|)$  の供給可能な需要点集合で,  $V_{d_{k_1}} \cap V_{d_{k_2}} = \emptyset (1 \leq k_1, k_2 \leq |V_s|, k_1 \neq k_2)$  でありかつ,  $\sum_{k=1}^{|V_s|} d(V_{d_k})$  が最大となる需要点集合  $V_{d_k}$  の族  $\{V_{d_k} | 1 \leq k \leq |V_s|\}$  を求めよ。” このような族  $\{V_{d_k} | 1 \leq k \leq |V_s|\}$  を最適解と呼ぶ。なお,  $\sum_{k=1}^{|V_s|} d(V_{d_k})$  が最大とは限らないが, その他の条件は同様に満たす場合には発見的解と呼ぶ。得られた解に対し,  $\sum_{k=1}^{|V_s|} d(V_{d_k})$  を ( $G$ ) 受給量と呼び,  $\sum_{k=1}^{|V_s|} d(V_{d_k})/d(V_d)$  を受給率と呼ぶ。

どの需要点  $v \in V_d$  が, どの集合  $V_{d_k} (1 \leq k \leq |V_s|)$  に含まれるかあるいはいずれにも含まれないかを, ( $G$ ) の供給関係と呼ぶ。以下の 3 つを供給条件と呼ぶ。

- 供給条件 1: 各需要点について, 供給源は高々 1 つ。
- 供給条件 2: 各  $u_k (1 \leq k \leq |V_s|)$  を供給源とする需要点からなる集合  $V_{d_k}$  について,  $G$  での誘導部分グラフ  $G[V_{d_k} \cup \{u_k\}]$  が連結。
- 供給条件 3: 各  $u_k (1 \leq k \leq |V_s|)$  を供給源とする需要点からなる集合  $V_{d_k}$  に対して,  $s(u_k) \geq d(V_{d_k})$ 。

グラフの最大供給分割問題は, “供給条件を全て満たし, 供給源を持つ需要点の需要量総和が最大となるような, 各需要点の供給源を求めよ” と等価な問題となる。

## 3 発見的解法

本研究で実装した既存の発見的アルゴリズムおよびその改良版を示す。以下の全てのアルゴリズムの入力および出力は以下の通り。

入力: グラフ  $G = (V_d \cup V_s, E)$ , 需要量  $d(v) > 0 (v \in V_d)$ , 供給量  $s(u) > 0 (u \in V_s)$ 。

出力:  $G$  に対する最大供給分割問題の発見的解  $\{V_{d_k} \subseteq V_d | 1 \leq k \leq |V_s|\}$ 。

### 3.1 Simple[2] とその改良

被供給領域に隣接する未供給の需要点の中から (ある供給点から供給可能で) 需要量が最大の点を選び、隣接する供給点からそこに供給する. このような需要点と供給点の組がなくなるまで繰り返す手法が *Simple* である.

#### 3.1.1 Simple の詳細

- step 1:**  $V_{d_k} \leftarrow \emptyset, p(u_k) \leftarrow s(u_k) (1 \leq k \leq |V_s|)$ .  
**step 2:**  $V_c \leftarrow V_s$ .  
**step 3:** 次の条件 (1),(2) を満たす  $u_{k_1} \in V_s$  と,  $v_{i_1} \in N(V_c)$  の組を探し, 複数存在する場合は, そのうち任意の 1 組  $\{v_{i_1}, u_{k_1}\}$  を選ぶ.  
 (1)  $v_{i_1} \in N(V_{d_{k_1}} \cup \{u_{k_1}\})$  は未供給の需要点で,  $d(v) \leq p(u)$  なる  $u \in V_s$  を持つ  $v \in N(V_c)$  の中で需要量が最大である.  
 (2)  $p(u_{k_1})$  が最大.  
**step 4:** step 3 で条件を満たす組  $\{v_{i_1}, u_{k_1}\}$  が存在しないならば停止する. そうでないならば  $V_{d_{k_1}} \leftarrow V_{d_{k_1}} \cup \{v_{i_1}\}, V_c \leftarrow V_c \cup \{v_{i_1}\}, p(u_{k_1}) \leftarrow p(u_{k_1}) - d(v_{i_1})$  とし, step 3 へ.

#### 3.1.2 改良手法 Simple<sub>k</sub>

改良のポイントは, 各供給点  $u_k$  に対し,  $N(V_{d_k} \cup \{u_k\})$  の中から未供給の需要点で需要量最大の点を選択することにより, 1 ステップで最大  $|V_s|$  個の需要点に供給することである. アルゴリズム記述は以下の通り.

- step 1:**  $V_{d_k} \leftarrow \emptyset, p(u_k) \leftarrow s(u_k) (1 \leq k \leq |V_s|)$ ,  $Pair \leftarrow \emptyset$ . /\*  $Pair$  は  $v \in V_d, u \in V_s$  なる組  $\{v, u\}$  を保持する\*/  
**step 2:** 各  $k (1 \leq k \leq |V_s|)$  に対し, 次の条件 (1) を満たす  $u_k \in V_s$  と,  $v_{i_k} \in N(V_{d_k} \cup \{u_k\})$  の組を探し, 複数存在する場合は, そのうち任意の 1 組  $\{v_{i_k}, u_k\}$  をそれぞれ選び,  $Pair$  に追加する.  
 (1)  $v_{i_k} \in N(V_{d_k} \cup \{u_k\})$  は未供給の需要点で,  $d(v) \leq p(u_k)$  なる  $v \in N(V_{d_k} \cup \{u_k\})$  の中で需要量が最大である.  
**step 3:**  $k_1 \neq k_2$  かつ  $p(u_{k_1}) \geq p(u_{k_2})$  なる  $\{v_{i_1}, u_{k_1}\}, \{v_{i_2}, u_{k_2}\} \in Pair$  が存在する場合には,  $\{v_{i_1}, u_{k_2}\}$  を  $Pair$  から取り除く.  
**step 4:**  $Pair = \emptyset$  ならば停止する.  $Pair \neq \emptyset$  ならば以下の操作を反復し,  $Pair = \emptyset$  となったとき step 2 へ: 各  $\{v_{i_k}, u_k\} \in Pair$  に対し,  $V_{d_k} \leftarrow V_{d_k} \cup \{v_{i_k}\}, p(u_k) \leftarrow p(u_k) - d(v_{i_k})$   $Pair \leftarrow Pair - \{v_{i_k}, u_k\}$ .

### 3.2 Fuzzy[2] とその改良

全ての供給可能な組に対して, どの供給点からどの需要点に供給すれば最適解に近づく可能性が高いかを表すことを意図した「供給妥当性」なる指標を

導入し, 供給妥当性の最も大きい需要点を持つ組を選び, その中の供給点から需要点へ供給する. これを, 供給可能な組がなくなるまで繰り返す手法が *Fuzzy* である. 基本アルゴリズムは以下の通り.

- step 1:**  $V_{d_k} \leftarrow \emptyset, p(u_k) \leftarrow s(u_k) (1 \leq k \leq |V_s|)$ .  
**step 2:**  $V_c \leftarrow V_s$ .  
**step 3:** 供給能力  $P_{ik} \leftarrow 0$ , 供給妥当性  $A_{ik} \leftarrow 0 (1 \leq i \leq |V_d|, 1 \leq k \leq |V_s|)$ .  
**step 4:**  $u_k \in V_s, v_i \in V_d - \bigcup_{1 \leq k \leq |V_s|} V_{d_k}$  に対して  $P_{ik}$  を求める. /\*  $P_{ik}$  の具体的計算法は後述\*/  
**step 5:**  $A_{ik} \leftarrow P_{ik} / \max\{p(u) | u \in V_s\} (1 \leq i \leq |V_d|, 1 \leq k \leq |V_s|)$ .  
**step 6:** 各  $v_i \in V_d - \bigcup_{1 \leq k \leq |V_s|} V_{d_k}$  に対し, 以下の操作を行う.  $A_{ik_1} \leftarrow \max\{A_{ik} | 1 \leq k \leq |V_s|\}$  かつ,  $A_{ik_2} \leftarrow \max\{A_{ik'} | 1 \leq k' \leq |V_s| \text{ かつ } k' \neq k\}$  とし,  $A_{ik_1} - A_{ik_2} \geq 0.01$  かつ  $V_{d_{k_1}} \leftarrow V_{d_{k_1}} \cup \{v_i\}$  としても供給条件が成り立つならば  $V_{d_{k_1}} \leftarrow V_{d_{k_1}} \cup \{v_i\}$  とし,  $p(u_{k_1}) \leftarrow s(u_{k_1}) - d(V_{d_{k_1}})$  かつ  $V_c \leftarrow V_c \cup \{v_i\}$  とする.  
**step 7:** step 6 で更新が生じた  $V_{d_k}$  がある場合は step 3 へ.  
**step 8:** 次の条件 (1),(2) を満たす  $u_k \in V_s$  と,  $v_i \in N(V_c)$  の組を探し, 複数存在する場合は, そのうち任意の 1 組を  $\{v_i, u_k\}$  として選ぶ.  
 (1)  $v_i \in N(V_{d_k} \cup \{u_k\})$  は未供給の需要点で,  $d(v) \leq p(u)$  なる  $u \in V_s$  を持つ  $v \in N(V_c)$  の中で需要量が最小である.  
 (2)  $p(u_k)$  が最大.  
**step 9:** step 8 で  $\{v_i, u_k\}$  の組が存在しないならば停止. そうでないならば  $V_{d_k} \leftarrow V_{d_k} \cup \{v_i\}$  とし step 3 へ.

[2] の *Fuzzy* の記述では無限ループ発生の可能性があり, それを避けるために, 上記アルゴリズムでは step 8, step 9 の操作を追加している.

step 4 における  $P_{ik}$  の求め方によって, 以下の 4 手法があるが, そのうち *Fuzzy<sub>m</sub>* は [2] の提案手法であり, 他の 3 手法は本研究での提案である.

#### 3.2.1 Fuzzy<sub>m</sub>

*Fuzzy* の step 4 の  $P_{ik}$  を以下の操作により求めるアルゴリズムが *Fuzzy<sub>m</sub>* である.

- step4-1:** まず, キュー  $Q \leftarrow \emptyset$  とする. 次に未供給の各需要点  $v_i$  に対し, 供給可能な供給点数  $Num(i)$  を求め,  $Q$  に  $u_k$  を追加する.  
**step4-2:**  $Q$  から 1 点を取り出す. 取り出した点が需要点  $v_p$  ならば,  $v_p$  に隣接する未供給の需要点  $v_{i'}$  で,  $P_{pk} - d(v_p) \geq d(v_{i'})$  かつ  $P_{pk} - d(v_p) \geq P_{i'k}$  となる需要点を  $v_{i'_1}, \dots, v_{i'_n}$  とする. 取り出した点が供給点  $u_k$  ならば,  $N(V_{d_k} \cup \{u_k\})$  内の未供給の需要点  $v_{i'}$  で,  $p(u_k) \geq d(v_{i'})$  かつ  $p(u_k) \geq P_{i'k}$  となる需要点を  $v_{i'_1}, \dots, v_{i'_n}$  とする.

step4-3:  $n = 0$  なら step 4-8 へ,  $n = 1$  なら step 4-4 へ,  $n \geq 2$  なら step 4-5 へ.

step4-4:  $P_{i',k} \leftarrow P_{pk} - d(v_p)$  (ただし step 4-2 で取り出した点が供給点  $u_k$  のときには  $P_{i',k} \leftarrow p(u_k)$ ) とし,  $Q$  に  $v_{i'}$  を追加して step4-2 へ.

/\*  $P_{i',k} \leftarrow p(u_k)$  が供給能力  $P_{i',k}$  の実質的初期値である \*/

step4-5: 各  $v_{i_j}$  ( $1 \leq j \leq n$ ) に対し, 需要点集合  $L_j$  を求める. ただし  $L_j$  は,  $v_{i_j}$  から  $v'$  へのパス上の点が全て未供給の需要点で, かつそのパス上の需要点の需要量総和が  $P_{pk} - d(v_p)$  以下 (ただし step 4-2 で取り出した点が供給点  $u_k$  のときには  $p(u_k)$ ) であるパスが存在するような需要点  $v'$  からなる集合である.

step4-6:  $1 \leq j \leq n$  なる各  $j$  に対し, 期待需要量  $D_j = \sum_{v_{i''} \in L_j} (d(v_{i''}) / \text{Num}(i''))$  を求める.

step4-7:  $1 \leq j \leq n$  なる各  $j$  に対し,  $(P_{pk} - d(v_p)) \times F(D_j / \sum_{j=1}^n D_j) \geq P_{i',k}$  ならば,  $P_{i',k} \leftarrow (P_{pk} - d(v_p)) \times F(D_j / \sum_{j=1}^n D_j)$  とし (ただし step 4-2 で取り出した点が供給点  $u_k$  のときには,  $p(u_k) \times F(D_j / \sum_{j=1}^n D_j) \geq P_{i',k}$  ならば  $P_{i',k} \leftarrow p(u_k) \times F(D_j / \sum_{j=1}^n D_j)$  とし),  $Q$  に  $v_{i_j}$  を追加する. ここで,  $F(x) = \min\{\max\{0, (x - \alpha) / (1 - 2\alpha)\}, 1\}$  かつ  $\alpha = 0.05$  である.

step4-8:  $Q = \emptyset$  ならば step 4 終了.  $Q \neq \emptyset$  ならば step 4-2 へ.

### 3.2.2 Fuzzy<sub>e</sub>

Fuzzy<sub>m</sub> による供給能力  $P_{ik}$  の計算時間減少のため, step 4-5 を以下の step 4-5' に置き換えた手法が Fuzzy<sub>e</sub> ( $\ell = 1, \dots, 5$ ) である.

step4-5': 各  $v_{i_j}$  ( $1 \leq j \leq n$ ) に対し, 需要点集合  $L_j$  を求める. ただし  $L_j$  は,  $v_{i_j}$  から  $v'$  へのパス上の全ての需要点が未供給で, かつそのパス上の需要点の需要量総和が  $P_{pk} - d(v_p)$  以下 (ただし step 4-2 で取り出した点が供給点  $u_k$  のときには  $p(u_k)$  以下) で, しかも長さ (辺の本数) が  $(\ell - 1)$  以下であるパスが存在するような需要点  $v'$  からなる集合である.

### 3.2.3 Fuzzy<sub>m</sub><sup>b</sup>

Fuzzy<sub>m</sub> の step 4-1 で, 各需要点  $v_i$  に対し供給可能な供給点数  $\text{Num}(i)$  を求めているが, この計算を省略し, 常に  $\text{Num}(i) = 1$  ( $1 \leq i \leq |V_d|$ ) として step 3-6 を計算することにより, 計算時間短縮を図った手法が Fuzzy<sub>m</sub><sup>b</sup> である.

### 3.2.4 Fuzzy<sub>e</sub><sup>b</sup>

Fuzzy<sub>m</sub> の step 4-1 には, 上記 Fuzzy<sub>m</sub><sup>b</sup> の手法を用い, かつ step 4-5 は Fuzzy<sub>e</sub> の step 4-5' の手法を用いることにより, さらに計算時間の短縮を図った手法が Fuzzy<sub>e</sub><sup>b</sup> ( $\ell = 1, \dots, 5$ ) である.

## 3.3 GA[5] とその改良

GA は, 遺伝的アルゴリズムに基づく手法である. 遺伝体は  $|V_d|$  個の遺伝子座を持ち,  $i$  ( $1 \leq i \leq |V_d|$ ) 番目の遺伝子として,  $v_i \in V_d$  の供給の親ノードを持つ.  $v_i$  の「供給の親ノード」とは,  $u_k \in V_s$  から  $v_i$  までのパス上の点で,  $v_i$  に隣接する点のことである. ここで, パス上の全ての点は  $u_k$  の供給領域に属する. これにより, 供給関係は 1 個の遺伝体によって表すことができる. 初期遺伝体を構成し, 自然淘汰, 交叉, 突然変異などの各遺伝操作を繰り返し行うことにより解を求める. アルゴリズムの詳細は [5] 参照.

[5] ではアルゴリズムの停止条件が明記されてなかったため, 本研究では [5] の提案手法に 2 種類の停止条件を追加し, その停止条件によって次の GA<sub>1</sub>, GA<sub>2</sub> に分ける.

### 3.3.1 GA<sub>1</sub>

停止条件を, 「遺伝体の評価関数値平均値に対する評価関数値の最大値の比が 1.001 以下である」とする. (つまり「評価関数値が平準化されたとき」である.)

### 3.3.2 GA<sub>2</sub>

停止条件を, 「各遺伝操作の試行回数が  $|V_d|/2$  回である」とする.

## 4 木に対する最適解法 KZN[1]

本節では, グラフが木の場合に最適解を与える解法 KZN を説明する. 入力および出力は以下の通り.

入力: 木  $T = (V_d \cup V_s, E)$ , 全て整数値である需要量  $d(v)$  ( $v \in V_d$ ) および供給量  $s(u)$  ( $u \in V_s$ )  
/\*  $V = V_d \cup V_s$  \*/

出力:  $T$  に対する最大供給分割問題の最適解  $\{V_{dk} \subseteq V_d | 1 \leq k \leq |V_s|\}$ .

### 4.1 概要

KZN は, 入力グラフが木  $T$  であり, しかも全ての需要量, 供給量が整数の場合に限り適用できる手法であり, 計算時間は  $O(m_s^2(|V_d| + |V_s|))$  ( $m_s = \max\{s(u) | u \in V_s\}$ ) である.

$v$  を根とする部分木  $T_v \subseteq T$  について, 根  $v$  に  $s(u_0) = x$  なる供給点  $u_0$  (これを仮想供給点と呼ぶ) を接続し  $u_0$  から  $v$  に供給する. このとき  $T_v$  の受給量が最大となり, かつ  $T_v$  内の供給点あるいは  $u_0$  から供給可能な需要点集合を  $F_{in}(v, x)$  と表し,  $f_{in}(v, x) = d(F_{in}(v, x))$  とおく. ただし,  $v$  が  $u_0$  により供給されない, あるいは  $v \in V_s$  ならば  $F_{in}(v, x) = \emptyset$ ,  $f_{in}(v, x) = -\infty$  とする. 逆に, 根  $v$  に  $d(v_0) = x$  なる需要点  $v_0$  (これを仮想需要点と呼ぶ) を接続し  $T_v$  内の供給点から  $v_0$  に供給する. このとき  $T_v$  の受給量が最大となり, かつ  $T_v$  内の供給点から供給可能な ( $v_0$  を除く) 需要点集合を  $F_{out}(v, x)$  と表し,

$f_{out}(v, x) = d(F_{out}(v, x))$  とおく。なお,  $x > 0$  なら, 仮想需要点  $v_0$  はある一つの供給源, これを  $u_0(v, x)$  と表す, から供給されていなければならない。さらに  $v$  自身が需要点ならば  $v$  も供給されてなければならない。ただし, いずれかの点が供給されないならば,  $F_{out}(v, x) = \emptyset, f_{out}(v, x) = -\infty, u_0(v, x) = nil$  とする。  $F_{in}$  あるいは  $F_{out}$  の要素を  $v_i(u_k)$  と表現する。これは需要点  $v_i$  の供給源が  $u_k$  であることを表す。供給源が不明な場合は  $v_i(\cdot)$  と表記する。なお,  $x$  は  $0 \leq x \leq m_s$  なる整数である。

部分木  $T_w$  に関して, 根  $w$  に仮想需要点あるいは仮想供給点を接続した時の  $T_w$  に関する上記5種類 ( $F_{in}, f_{in}, F_{out}, f_{out}, u_0$ ) の関数値がわかっているならば,  $T$  における  $w$  の親  $v$  と  $T_w$  を結合して得る部分木  $T_v$  に関して, 上記5種類の関数値を計算する。  $w$  以外に  $v$  の子  $w'$  があれば,  $T_v$  と  $T_{w'}$  を結合した部分木を新しく  $T_v$  とし,  $T_v$  についても同様に上記5種類の関数値を計算する。これを木の葉から根に向かって順次求めていくことにより, 木全体の供給関係を定める。なお, [1] では  $f_{in}, f_{out}$  しか定義されてなかったため, 出力が受給量  $\sum_{k=1}^{|V_s|} d(V_{d_k})$  のみとなっていたが, 本研究では  $F_{in}, F_{out}, u_0$  を新たに定義, 導入することにより, 解である需要点集合族  $\{V_{d_k} \subseteq V_d | 1 \leq k \leq |V_s|\}$  を出力するように修正している。

## 4.2 アルゴリズム記述

アルゴリズムは以下の通りである。

**step 1:** 根  $r$  を指定し,  $d \leftarrow$  (木の最大深さ),  $V_{depth}(d_0)$  を深さ  $d_0$  の点の集合 ( $0 \leq d_0 \leq d$ ),  $m_s \leftarrow \max\{s(u) | u \in V_s\}, V_{d_k} \leftarrow \emptyset (1 \leq k \leq |V_s|)$ .

**step 2:** 任意に1点  $v \in V_{depth}(d)$  を取り出し,  $0 \leq x \leq m_s$  なる各整数  $x$  について,  $v \in V_d$  ならば,

$$\begin{aligned} f_{in}(v, x) &\leftarrow \begin{cases} -\infty & (0 \leq x < d(v)) \\ d(v) & (d(v) \leq x \leq m_s) \end{cases} \\ F_{in}(v, x) &\leftarrow \begin{cases} \emptyset & (0 \leq x < d(v)) \\ \{v(\cdot)\} & (d(v) \leq x \leq m_s) \end{cases} \\ f_{out}(v, x) &\leftarrow \begin{cases} 0 & (x = 0) \\ -\infty & (0 < x \leq m_s) \end{cases} \\ F_{out}(v, x) &\leftarrow \emptyset \\ u_0(v, x) &\leftarrow nil \end{aligned}$$

とし,  $v \in V_s$  ならば,

$$\begin{aligned} f_{in}(v, x) &\leftarrow -\infty \\ F_{in}(v, x) &\leftarrow \emptyset \\ f_{out}(v, x) &\leftarrow \begin{cases} 0 & (0 \leq x \leq s(v)) \\ -\infty & (s(v) < x \leq m_s) \end{cases} \\ F_{out}(v, x) &\leftarrow \emptyset \\ u_0(v, x) &\leftarrow \begin{cases} v & (0 \leq x \leq s(v)) \\ nil & (s(v) < x \leq m_s) \end{cases} \end{aligned}$$

とする。もし  $v$  が葉ならば step 7へ。そうでないならば  $v$  の全ての子の集合を  $V_{son}(v)$  として step 3へ。

**step 3:** 任意に1点  $w \in V_{son}(v)$  を取り出し,  $0 \leq x \leq m_s$  なる各整数  $x$  について,

$$\begin{aligned} M_1 &= f_{in}(v, y_1) + f_{in}(w, x - y_1) \\ &= \max\{f_{in}(v, y) + f_{in}(w, x - y) | 0 \leq y \leq x\} \\ M_2 &= f_{in}(v, x) + f_{out}(w, 0) \\ M_3 &= f_{out}(v, y_2) + f_{in}(w, y_2 - x) \\ &= \max\{f_{out}(v, y) + f_{in}(w, y - x) | x \leq y \leq m_s\} \\ M_4 &= f_{out}(v, x) + f_{out}(w, 0) \\ M_5 &= f_{in}(v, y_3 - x) + f_{out}(w, y_3) \\ &= \max\{f_{in}(v, y - x) + f_{out}(w, y) | x \leq y \leq m_s\} \end{aligned}$$

とおき,  $f'_{in}(v, x) \leftarrow \max\{M_1, M_2\}, f'_{out}(v, x) \leftarrow \max\{M_3, M_4, M_5\}$  とする。

**step 4:**

$$F'_{in}(v, x) \leftarrow \begin{cases} F_{in}(v, y_1) \cup F_{in}(w, x - y_1) \\ \quad (f'_{in}(v, x) = M_1 \neq -\infty \text{ のとき}) \\ F_{in}(v, x) \cup F_{out}(w, 0) \\ \quad (f'_{in}(v, x) = M_2 \neq -\infty \text{ のとき}) \\ \emptyset \quad (f'_{in}(v, x) = -\infty \text{ のとき}) \end{cases}$$

$$F'_{out}(v, x) \leftarrow \begin{cases} F_{out}(v, y_2) \\ \quad \cup \{v'(u') | v'(u') \in F_{in}(w, y_2 - x)\} \\ \quad \cup \{v''(u_0(v, y_2)) | v''(\cdot) \in F_{in}(w, y_2 - x)\} \\ \quad (f'_{out}(v, x) = M_3 \neq -\infty \text{ のとき}) \\ F_{out}(v, x) \cup F_{out}(w, 0) \\ \quad (f'_{out}(v, x) = M_4 \neq -\infty \text{ のとき}) \\ \{v'(u') | v'(u') \in F_{in}(v, y_3 - x)\} \\ \quad \cup \{v''(u_0(w, y_3)) | v''(\cdot) \in F_{in}(v, y_3 - x)\} \\ \quad \cup F_{in}(w, y_3) \\ \quad (f'_{out}(v, x) = M_5 \neq -\infty \text{ のとき}) \\ \emptyset \quad (f'_{out}(v, x) = -\infty \text{ のとき}) \end{cases}$$

$$u'_0(v, x) \leftarrow \begin{cases} u_0(v, y_2) & (f'_{out}(v, x) = M_3 \text{ のとき}) \\ u_0(v, x) & (f'_{out}(v, x) = M_4 \text{ のとき}) \\ u_0(w, y_3) & (f'_{out}(v, x) = M_5 \text{ のとき}) \end{cases}$$

**step 5:**  $f_{in}(v, x) \leftarrow f'_{in}(v, x), F_{in}(v, x) \leftarrow F'_{in}(v, x),$   
 $f_{out}(v, x) \leftarrow f'_{out}(v, x), F_{out}(v, x) \leftarrow F'_{out}(v, x),$   
 $u_0(v, x) \leftarrow u'_0(v, x)$

**step 6:**  $V_{son}(v) \leftarrow V_{son}(v) - \{w\}$  とし,  $V_{son}(v) = \emptyset$  ならば step 7へ。  $V_{son}(v) \neq \emptyset$  ならば step 3へ。

**step 7:**  $V_{depth}(d) \leftarrow V_{depth}(d) - \{v\}$  とし,  $V_{depth}(d) = \emptyset$  ならば step 8へ。  $V_{depth}(d) \neq \emptyset$  ならば step 2へ。

**step 8:**  $d > 0$  ならば  $d \leftarrow d - 1$  とし, step 2へ。  $d = 0$  ならば step 9へ。

**step 9:**  $F_{out}(r, 0) = \emptyset$  となるまで以下の操作を繰り返す:  $F_{out}(r, 0) \leftarrow F_{out}(r, 0) - \{v(u_k)\}$  とし,  $V_{d_k} \leftarrow V_{d_k} \cup \{v\}$  とする。

## 5 計算機実験

### 5.1 実験概要

発見的解法である *Simple*, *Simple<sub>all</sub>*, *Fuzzym*, *Fuzzy<sub>ℓ</sub>*, *Fuzzym<sub>ℓ</sub>*, *Fuzzy<sub>ℓ</sub><sup>b</sup>* ( $ℓ = 1, \dots, 5$ ),  $GA_1$ ,  $GA_2$ , および最適解を求める解法 *KZN* を, 計算機 (OS: FreeBSD 4.5-RELEASE, CPU: Pentium IV 1.7GMz) 上の C 言語により実装した. グラフ  $G$  としては以下の 5 種類を用意した.

**tree A:** 最適解の受給率は 100% となるような木.

**graph A<sup>+</sup>:** tree A と同様な木に, 多重辺とならないように辺を  $|V_d|$  本付加したグラフ.

**tree B:** type A と同様な木で, 全ての供給点が葉に存在.

**tree C:** 最適解であってもその受給率は 100% とは限らない木.

**graph C<sup>+</sup>:** tree C と同様な木に, 多重辺とならないように辺を  $|V_d|$  本付加したグラフ.

どの供給点対も非隣接とし, 各点の需要量および供給量は正の整数とする. 各種類のグラフに対して,  $m_s \in \{200, 2000\}$ ,  $|V_d| \in \{10, 20, 50, 100, 200, 300, 500, 700, 1000, 2000, 3000, 5000\}$ ,  $|V_s| \in \{3, 5, 10, 20, 50, 100, 200, 500\}$  なる 3 項組  $\{m_s, |V_d|, |V_s|\}$  (ただし,  $200 \times |V_s| > |V_d| \geq |V_s|$ ) を 100 個ずつ用意.

### 5.2 実験結果

表 3 に  $m_s$  が 200, 2000 である全データ (いずれも 6600 個) を各手法に適用した際の受給率の平均値を示し, 表 4 に  $\{|V_d|, |V_s|\} = \{500, 20\}$  である 100 個の入力に対する受給率の平均値を示す. 表 5~表 8 に, それぞれ  $\{|V_d|, |V_s|\} = \{500, 20\}, \{500, 100\}, \{1000, 20\}, \{1000, 100\}$  である 100 個の入力を各手法に適用した際の平均計算時間を示す. 解精度の評価には受給率を用いる.

*Simple*, *Simple<sub>all</sub>* は他手法と比べて非常に速いが, *Simple<sub>all</sub>* の方が *Simple* より高速である. 受給率は逆に, *Simple* の方が *Simple<sub>all</sub>* よりも概ねよい. *Fuzzy* については, *Fuzzym* よりも本研究で改良した手法 *Fuzzym<sub>ℓ</sub>*, *Fuzzy<sub>ℓ</sub><sup>b</sup>* のほうが高速である. 特に graph A<sup>+</sup> や graph C<sup>+</sup> では *Fuzzym<sub>ℓ</sub>* や *Fuzzy<sub>ℓ</sub><sup>b</sup>* の高速性が顕著である. それでも *Simple*, *Simple<sub>all</sub>* と比べると非常に遅い. しかし, 受給率に関しては, tree C や graph C<sup>+</sup> では *Simple*, *Simple<sub>all</sub>* よりもよい結果が得られている. 他の入力グラフでも概ね *Simple*, *Simple<sub>all</sub>* よりも受給率が高いものの, 一部 *Simple*, *Simple<sub>all</sub>* の方が受給率が高い場合もある. これら全ての入力グラフの最適解の受給率は 100% であるが, 入力グラフの作成の際には, 最適解において供給点  $u$  が供給する需要点は,  $u$  に近い位置にあるように作成した. *Simple*, *Simple<sub>all</sub>* はアルゴリズムからもわかるように, 各供給点からなるべく近い位置にある需要点に供給するため, このような入力グラフで受給率が

高くなったと考えられる. 需要量, 供給量をランダムに与えた tree C や graph C<sup>+</sup> では, *Simple*, *Simple<sub>all</sub>* と比べて *Fuzzym*, *Fuzzy<sub>ℓ</sub>*, *Fuzzym<sub>ℓ</sub>*, *Fuzzy<sub>ℓ</sub><sup>b</sup>* の受給率の方が非常に高いことからわかるように, 一般に *Fuzzy* は高い受給率を与える解法であると言える. 今回の実験では  $GA$  の計算時間は辺数によらないことがわかったが,  $GA_1$  は比較的高速に解を求めることができる反面, 受給率が著しく低い. また,  $GA_2$  は計算時間が長く, 一部のデータに対しては  $KZN$  よりも長い.  $GA_2$  は  $GA_1$  よりも各遺伝操作の回数が多くなるが, それにも関わらず受給率が他手法と比べてかなり低いことから,  $GA$  は受給率の悪い手法であることが明らかになった. どの発見的解法も, 最大供給量  $m_s$  の変化による計算時間の変化はほぼないようであるが,  $KZN$  は  $m_s$  が大きくなると計算時間が非常に長くなっている. 実際  $KZN$  の計算時間は  $m_s$  の 2 乗のオーダーに比例するので,  $KZN$  は最大供給量が大きい場合には不向きである.

以上今回の実験からは, 最も高速に解を求めるのなら *Simple<sub>all</sub>* がよく, より一般的なグラフにおいて受給率のよい解を求めるのなら *Fuzzy* が, 特に計算時間も考慮に入れると *Fuzzym<sub>ℓ</sub>* がよいことがわかった.

## 6 まとめと今後の課題

本稿では, グラフの最大供給分割問題に対する既存の発見的解法, および木の場合に最適解を求める解法に対しいくつかの改良を施し, それらの性能を計算機実験により比較評価した. できるだけ高速に解を求めるのなら *Simple<sub>all</sub>* がよく, 一般的なグラフにおいて受給率のよい解を求めるのなら *Fuzzym<sub>ℓ</sub>* がよいことが今回の結論である.

今後の課題として, 以下のようなものが挙げられる.

- (1) 入力データの多様化. (木への辺付加本数の増加, 需要量や供給量の実数値への拡張等)
- (2) 今回未実装のため比較できなかった解法 *Switch*, *HNN*, *IZN<sub>ε</sub>* の実装と性能比較.
- (3) さらに高速かつ受給率の高い解を求める解法の開発.

## 参考文献

- [1] 蒲倉, 周, 西関: “木の分割問題を解くアルゴリズム”, Technical Report COMP2001-87, IEICE of Japan (2002).
- [2] 今村, 叶井, 川上: “配電融通問題へのファジィ推論の適用”, 電気学会論文誌 B, **113**, 5, pp. 493-498 (1993).
- [3] 青木, 佐藤, 伊藤, 桑原, 金指: “電圧降下制約を考慮した配電系統の事故時負荷切換方法”, 電気学会論文誌 B, **108**, 7, pp. 329-335 (1988).
- [4] 福井, 川上: “電力系統の負荷融通問題におけるホップフィールド形ニューラルネットワークの適用”, 電気学会論文誌 D, **111**, 1, pp. 10-19 (1991).
- [5] 福山, 植木: “負荷融通問題への遺伝的アルゴリズムの適用”, 電気学会論文誌 B, **114**, 4, pp. 361-366 (1994).
- [6] T. Ito, X. Zhou and T. Nishizeki: “Partitioning trees of supply and demand”, Proc. of International Symposium on Algorithms and Computation (ISAAC 2002), pp. 612-623 (2002).

表 3: 受給率の平均値 (%) (5.1 節で示した  $m_s$  が 200, 2000 である全データ (いずれも 6600 個) についての平均)

| type                                 | A     |       | A <sup>+</sup> |       | B     |       | C     |       | C <sup>+</sup> |       |
|--------------------------------------|-------|-------|----------------|-------|-------|-------|-------|-------|----------------|-------|
|                                      | 200   | 2000  | 200            | 2000  | 200   | 2000  | 200   | 2000  | 200            | 2000  |
| <i>Simple</i>                        | 92.83 | 95.60 | 89.29          | 91.65 | 94.60 | 95.76 | 66.15 | 69.20 | 70.77          | 77.85 |
| <i>Simple<sub>all</sub></i>          | 91.21 | 91.39 | 85.05          | 84.21 | 94.44 | 94.07 | 65.97 | 69.80 | 69.19          | 75.83 |
| <i>Fuzzy<sub>m</sub></i>             | 94.49 | 94.48 | 89.28          | 89.60 | 93.84 | 94.00 | 68.86 | 73.57 | 74.74          | 79.46 |
| <i>Fuzzy<sub>2</sub></i>             | 92.77 | 93.29 | 90.08          | 90.09 | 94.34 | 94.76 | 68.14 | 72.74 | 74.74          | 80.26 |
| <i>Fuzzy<sub>5</sub></i>             | 93.05 | 93.32 | 89.43          | 90.00 | 94.25 | 94.42 | 68.49 | 73.21 | 74.67          | 79.45 |
| <i>Fuzzy<sub>m</sub><sup>p</sup></i> | 94.00 | 94.05 | 89.20          | 89.75 | 93.22 | 93.25 | 67.94 | 72.54 | 73.71          | 80.10 |
| <i>Fuzzy<sub>2</sub><sup>p</sup></i> | 93.47 | 93.96 | 89.02          | 89.89 | 94.39 | 94.64 | 68.11 | 72.56 | 74.47          | 79.76 |
| <i>Fuzzy<sub>5</sub><sup>p</sup></i> | 93.68 | 93.85 | 89.14          | 89.74 | 94.04 | 94.18 | 68.20 | 72.77 | 74.17          | 79.90 |
| <i>GA<sub>1</sub></i>                | 84.12 | 84.04 | 69.63          | 69.65 | 82.25 | 82.42 | 63.22 | 67.26 | 61.03          | 64.21 |
| <i>GA<sub>2</sub></i>                | 88.11 | 87.96 | 78.18          | 77.73 | 86.16 | 85.92 | 63.54 | 67.57 | 64.58          | 69.45 |
| <i>KZN</i>                           | 100   | 100   | -              | -     | 100   | 100   | 71.86 | 76.74 | -              | -     |

表 4:  $|V_d| = 500, |V_s| = 20$  のときの受給率の平均値 (%)

| type                                 | A     |       | A <sup>+</sup> |       | B     |       | C     |       | C <sup>+</sup> |       |
|--------------------------------------|-------|-------|----------------|-------|-------|-------|-------|-------|----------------|-------|
|                                      | 200   | 2000  | 200            | 2000  | 200   | 2000  | 200   | 2000  | 200            | 2000  |
| <i>Simple</i>                        | 87.58 | 93.92 | 90.55          | 92.03 | 88.50 | 93.89 | 72.50 | 74.86 | 81.29          | 87.25 |
| <i>Simple<sub>all</sub></i>          | 87.08 | 87.94 | 81.83          | 79.94 | 91.61 | 91.70 | 72.97 | 76.48 | 76.57          | 83.17 |
| <i>Fuzzy<sub>m</sub></i>             | 94.77 | 95.05 | 97.90          | 97.58 | 94.60 | 94.50 | 79.21 | 83.37 | 86.26          | 92.03 |
| <i>Fuzzy<sub>2</sub></i>             | 91.88 | 92.39 | 97.95          | 97.86 | 95.40 | 95.45 | 77.59 | 81.85 | 86.45          | 92.27 |
| <i>Fuzzy<sub>5</sub></i>             | 92.28 | 92.65 | 97.68          | 97.39 | 94.97 | 94.90 | 78.20 | 82.27 | 85.99          | 91.76 |
| <i>Fuzzy<sub>m</sub><sup>p</sup></i> | 93.84 | 94.40 | 97.46          | 97.38 | 93.70 | 93.57 | 77.84 | 81.80 | 86.11          | 91.87 |
| <i>Fuzzy<sub>2</sub><sup>p</sup></i> | 93.24 | 93.74 | 96.61          | 96.95 | 95.40 | 95.41 | 77.41 | 81.56 | 85.56          | 91.30 |
| <i>Fuzzy<sub>5</sub><sup>p</sup></i> | 93.55 | 93.69 | 97.18          | 97.13 | 94.56 | 94.41 | 78.01 | 82.05 | 85.70          | 91.57 |
| <i>GA<sub>1</sub></i>                | 87.69 | 87.80 | 74.24          | 74.57 | 86.64 | 86.55 | 74.11 | 77.07 | 70.08          | 73.82 |
| <i>GA<sub>2</sub></i>                | 89.64 | 89.52 | 82.48          | 81.63 | 87.21 | 87.71 | 75.32 | 78.79 | 75.88          | 79.61 |
| <i>KZN</i>                           | 100   | 100   | -              | -     | 100   | 100   | 83.24 | 87.82 | -              | -     |

表 5:  $|V_d| = 500, |V_s| = 20$  のときの平均計算時間 (sec)

| type                                 | A      |        | A <sup>+</sup> |        | B      |        | C      |        | C <sup>+</sup> |        |
|--------------------------------------|--------|--------|----------------|--------|--------|--------|--------|--------|----------------|--------|
|                                      | 200    | 2000   | 200            | 2000   | 200    | 2000   | 200    | 2000   | 200            | 2000   |
| <i>Simple</i>                        | 0.0003 | 0.0003 | 0.0005         | 0.0009 | 0.0003 | 0.0000 | 0.0003 | 0.0002 | 0.0003         | 0.0004 |
| <i>Simple<sub>all</sub></i>          | 0.0004 | 0.0004 | 0.0008         | 0.0005 | 0.0002 | 0.0002 | 0.0004 | 0.0005 | 0.0002         | 0.0007 |
| <i>Fuzzy<sub>m</sub></i>             | 0.0074 | 0.0082 | 4.2673         | 5.7495 | 0.0401 | 0.0812 | 0.0077 | 0.0073 | 3.8245         | 5.7383 |
| <i>Fuzzy<sub>2</sub></i>             | 0.0061 | 0.0073 | 1.6781         | 2.4902 | 0.0445 | 0.1173 | 0.0079 | 0.0096 | 1.6248         | 2.0396 |
| <i>Fuzzy<sub>5</sub></i>             | 0.0061 | 0.0068 | 1.8063         | 2.7778 | 0.0339 | 0.0973 | 0.0084 | 0.0087 | 1.7312         | 2.1323 |
| <i>Fuzzy<sub>m</sub><sup>p</sup></i> | 0.0073 | 0.0088 | 3.2877         | 2.7683 | 0.0365 | 0.0787 | 0.0073 | 0.0073 | 2.3156         | 2.6578 |
| <i>Fuzzy<sub>2</sub><sup>p</sup></i> | 0.0073 | 0.0077 | 0.4212         | 0.4469 | 0.0555 | 0.1450 | 0.0077 | 0.0064 | 0.4185         | 0.4396 |
| <i>Fuzzy<sub>5</sub><sup>p</sup></i> | 0.0074 | 0.0088 | 0.6798         | 0.7705 | 0.0387 | 0.1108 | 0.0073 | 0.0060 | 0.8228         | 0.7848 |
| <i>GA<sub>1</sub></i>                | 0.0684 | 0.0708 | 0.1121         | 0.1013 | 0.0720 | 0.0732 | 0.0726 | 0.0744 | 0.1138         | 0.1103 |
| <i>GA<sub>2</sub></i>                | 5.8012 | 5.6674 | 11.663         | 11.131 | 5.5886 | 5.8119 | 6.0522 | 5.9691 | 11.072         | 11.177 |
| <i>KZN</i>                           | 3.1971 | 54.919 | -              | -      | 2.408  | 50.496 | 2.558  | 52.301 | -              | -      |

表 6:  $|V_d| = 500, |V_s| = 100$  のときの平均計算時間 (sec)

| type                                 | A      |        | A <sup>+</sup> |        | B      |        | C      |        | C <sup>+</sup> |        |
|--------------------------------------|--------|--------|----------------|--------|--------|--------|--------|--------|----------------|--------|
|                                      | 200    | 2000   | 200            | 2000   | 200    | 2000   | 200    | 2000   | 200            | 2000   |
| <i>Simple</i>                        | 0.0022 | 0.0021 | 0.0027         | 0.0031 | 0.0023 | 0.0012 | 0.0020 | 0.0023 | 0.0023         | 0.0032 |
| <i>Simple<sub>all</sub></i>          | 0.0005 | 0.0004 | 0.0013         | 0.0013 | 0.0008 | 0.0008 | 0.0011 | 0.0004 | 0.0016         | 0.0017 |
| <i>Fuzzy<sub>m</sub></i>             | 0.0823 | 0.0817 | 0.4223         | 0.5531 | 0.1253 | 0.1462 | 0.0767 | 0.0779 | 0.3584         | 0.3728 |
| <i>Fuzzy<sub>2</sub></i>             | 0.0605 | 0.0862 | 0.3166         | 0.4305 | 0.0990 | 0.1451 | 0.0902 | 0.0990 | 0.2605         | 0.2690 |
| <i>Fuzzy<sub>5</sub></i>             | 0.0640 | 0.0740 | 0.3381         | 0.4364 | 0.1163 | 0.1565 | 0.0810 | 0.0776 | 0.2997         | 0.3340 |
| <i>Fuzzy<sub>m</sub><sup>b</sup></i> | 0.0715 | 0.0827 | 0.1898         | 0.2042 | 0.1070 | 0.1375 | 0.0824 | 0.0751 | 0.1916         | 0.1953 |
| <i>Fuzzy<sub>2</sub><sup>b</sup></i> | 0.0670 | 0.0831 | 0.2311         | 0.2802 | 0.0892 | 0.1780 | 0.0848 | 0.0845 | 0.6884         | 0.8697 |
| <i>Fuzzy<sub>5</sub><sup>b</sup></i> | 0.0739 | 0.0966 | 0.1284         | 0.1402 | 0.1458 | 0.1409 | 0.0755 | 0.0694 | 0.1791         | 0.1873 |
| <i>GA<sub>1</sub></i>                | 0.4545 | 0.4450 | 0.8501         | 0.8404 | 0.4688 | 0.4813 | 0.4692 | 0.4690 | 0.8398         | 0.8611 |
| <i>GA<sub>2</sub></i>                | 105.36 | 104.20 | 203.26         | 202.80 | 109.97 | 113.92 | 106.83 | 107.03 | 196.28         | 199.90 |
| <i>KZN</i>                           | 13.339 | 189.08 | -              | -      | 12.334 | 167.11 | 11.818 | 168.13 | -              | -      |

表 7:  $|V_d| = 1000, |V_s| = 20$  のときの平均計算時間 (sec)

| type                                 | A      |        | A <sup>+</sup> |        | B      |        | C      |        | C <sup>+</sup> |        |
|--------------------------------------|--------|--------|----------------|--------|--------|--------|--------|--------|----------------|--------|
|                                      | 200    | 2000   | 200            | 2000   | 200    | 2000   | 200    | 2000   | 200            | 2000   |
| <i>Simple</i>                        | 0.0006 | 0.0005 | 0.0016         | 0.0018 | 0.0005 | 0.0006 | 0.0006 | 0.0006 | 0.0011         | 0.0014 |
| <i>Simple<sub>all</sub></i>          | 0.0009 | 0.0015 | 0.0021         | 0.0012 | 0.0009 | 0.0013 | 0.0009 | 0.0016 | 0.0023         | 0.0023 |
| <i>Fuzzy<sub>m</sub></i>             | 0.0175 | 0.0197 | 48.415         | 66.876 | 1.1182 | 1.3235 | 0.0190 | 0.0181 | 38.335         | 57.741 |
| <i>Fuzzy<sub>2</sub></i>             | 0.0138 | 0.0187 | 22.296         | 33.635 | 1.1173 | 1.5432 | 0.0183 | 0.0191 | 17.569         | 27.446 |
| <i>Fuzzy<sub>5</sub></i>             | 0.0148 | 0.0182 | 23.046         | 36.549 | 1.2545 | 1.4479 | 0.0168 | 0.0201 | 18.580         | 28.625 |
| <i>Fuzzy<sub>m</sub><sup>b</sup></i> | 0.0181 | 0.0195 | 33.252         | 28.164 | 0.9558 | 1.1495 | 0.0186 | 0.0180 | 21.206         | 30.010 |
| <i>Fuzzy<sub>2</sub><sup>b</sup></i> | 0.0146 | 0.0165 | 2.1483         | 2.2884 | 1.1342 | 1.6041 | 0.0173 | 0.0168 | 1.9172         | 2.3505 |
| <i>Fuzzy<sub>5</sub><sup>b</sup></i> | 0.0148 | 0.0180 | 3.8595         | 4.5645 | 1.2043 | 1.3391 | 0.0155 | 0.0142 | 3.7821         | 4.5284 |
| <i>GA<sub>1</sub></i>                | 0.4366 | 0.4463 | 0.8306         | 0.8248 | 0.4320 | 0.4547 | 0.4917 | 0.4846 | 0.8236         | 0.8506 |
| <i>GA<sub>2</sub></i>                | 105.82 | 104.09 | 197.63         | 201.78 | 103.79 | 105.84 | 104.00 | 104.27 | 197.99         | 201.11 |
| <i>KZN</i>                           | 10.759 | 152.89 | -              | -      | 6.3485 | 121.48 | 8.0484 | 133.59 | -              | -      |

表 8:  $|V_d| = 1000, |V_s| = 100$  のときの平均計算時間 (sec)

| type                                 | A      |        | A <sup>+</sup> |        | B      |        | C      |        | C <sup>+</sup> |        |
|--------------------------------------|--------|--------|----------------|--------|--------|--------|--------|--------|----------------|--------|
|                                      | 200    | 2000   | 200            | 2000   | 200    | 2000   | 200    | 2000   | 200            | 2000   |
| <i>Simple</i>                        | 0.0022 | 0.0021 | 0.0027         | 0.0031 | 0.0023 | 0.0012 | 0.0020 | 0.0023 | 0.0023         | 0.0032 |
| <i>Simple<sub>all</sub></i>          | 0.0005 | 0.0004 | 0.0013         | 0.0013 | 0.0008 | 0.0008 | 0.0011 | 0.0004 | 0.0016         | 0.0017 |
| <i>Fuzzy<sub>m</sub></i>             | 0.0823 | 0.0817 | 0.4223         | 0.5531 | 0.1253 | 0.1462 | 0.0767 | 0.0779 | 0.3584         | 0.3728 |
| <i>Fuzzy<sub>2</sub></i>             | 0.0605 | 0.0862 | 0.3166         | 0.4305 | 0.0990 | 0.1451 | 0.0902 | 0.0990 | 0.2605         | 0.2690 |
| <i>Fuzzy<sub>5</sub></i>             | 0.0640 | 0.0740 | 0.3381         | 0.4364 | 0.1163 | 0.1565 | 0.0810 | 0.0776 | 0.2997         | 0.3340 |
| <i>Fuzzy<sub>m</sub><sup>b</sup></i> | 0.0715 | 0.0827 | 0.1898         | 0.2042 | 0.1070 | 0.1375 | 0.0824 | 0.0751 | 0.1916         | 0.1953 |
| <i>Fuzzy<sub>2</sub><sup>b</sup></i> | 0.0670 | 0.0831 | 0.2311         | 0.2802 | 0.0892 | 0.1780 | 0.0848 | 0.0845 | 0.6884         | 0.8697 |
| <i>Fuzzy<sub>5</sub><sup>b</sup></i> | 0.0739 | 0.0966 | 0.1284         | 0.1402 | 0.1458 | 0.1409 | 0.0755 | 0.0694 | 0.1791         | 0.1873 |
| <i>GA<sub>1</sub></i>                | 0.4545 | 0.4450 | 0.8501         | 0.8404 | 0.4688 | 0.4813 | 0.4692 | 0.4690 | 0.8398         | 0.8611 |
| <i>GA<sub>2</sub></i>                | 105.36 | 104.20 | 203.26         | 202.80 | 109.97 | 113.92 | 106.83 | 107.03 | 196.28         | 199.90 |
| <i>KZN</i>                           | 13.339 | 189.08 | -              | -      | 12.335 | 167.11 | 11.818 | 168.13 | -              | -      |