Generalised single valued neutrosophic number and its
application to neutrosophic linear programming

Tuhin Bera¹ and Nirmal Kumar Mahapatra²

¹ Department of Mathematics, Boror S. S. High School, Baghn, Howrah-711312, WB, India. E-mail: tuhin78bera@gmail.com
² Department of Mathematics, Panskeura Banamali College, Panskeura RS-721152, WB, India. E-mail: nirmal_hridoy@yahoo.co.in

Abstract. In this paper, the concept of single valued neutrosophic number (SVN-number) is presented in a general-
ized way. Using this notion, a crisp linear programming problem (LP-problem) is extended to a neutrosophic linear
programming problem (NLP-problem). The coefficients of the objective function of a crisp LP-problem are consid-
ered as generalized single valued neutrosophic number (G_SVN-number). This modified form of LP-problem is here
called an NLP-problem. An algorithm is developed to solve NLP-problem by simplex method. Finally, this simplex
algorithm is applied to a real life problem. The problem is illustrated and solved numerically.

Keywords: Single valued neutrosophic number; Neutrosophic linear programming problem; Simplex method.

1 Introduction

Introduction of fuzzy set by Zadeh [10] and then intuitionistic fuzzy set by Atanassov [8] brought a golden
opportunity to handle the uncertainty and vagueness in our daily life activities. The fuzzy sets are evaluated by
the membership grade of an object only, whereas intuitionistic fuzzy set meets the membership and the non-
membership grade of an object simultaneously. To deal with uncertainty more precisely, Smarandache [3,4]
initiated the notion of neutrosophic set (NS), a generalised version of classical set, fuzzy set, intuitionistic fuzzy
set etc. In the neutrosophic logic, each proposition is estimated by a triplet viz., truth grade, indeterminacy grade
and falsity grade. The indeterministic part of uncertain data, introduced in NS theory, plays an important role
to make a proper decision which is not possible by intuitionistic fuzzy set theory. Since indeterminacy always
appears in our routine activities, the NS theory can analyse the various situations smoothly. But it is too difficult
to apply the NS theory in real life scenario for it’s initial character as pointed out by Smarandache. So to apply
in real spectrum, Wang et al. [6] brought the concept of single valued neutrosophic set (SVN-set). Ranking of
fuzzy number and intuitionistic fuzzy number is an interesting subject needed in decision making, optimization,
even in developing of various mathematical structures. From time to time, several ranking methods [2,5,9,13-15]
have been adopted by researchers. Naturally, the ranking of neutrosophic number also was come into
consideration from beginning of NS theory. Deli and Subas [7] considered a ranking way of neutrosophic
numbers and have used it to a decision making problems. Abdel-Baset [11,12] solved group decision making
problems based on TOPSIS technique by use of neutrosophic number. To estimate and solve the NLP-problem
in different direction, some respective attempts [1,16] by researchers are seen.

This paper introduces the structure of SVN-number in a different way to opt the notion of generalized single
valued trapezoidal neutrosophic number (G_SVTN-number), generalized single valued triangular neutrosophic
number (G_SVTN-number) and develops an algorithm to solve NLP-problem by simplex method. The proposed
simplex algorithm is applied to a real life problem. The problem is illustrated and solved numerically.

The organisation of this paper is as follows. Section 2 deals some preliminary definitions. The concept of
G_SVN-number, G_SVTN-number, G_SVTN-number and their respective parametric form are presented in Sec-
tion 3. The concept of NLP-problem and it’s solution procedure are proposed in Section 4 and Section 5.

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to
neutrosophic linear programming.
respectively. In Section 6, the simplex method is illustrated by suitable examples. Finally, the present work is summarised in Section 7.

2 Preliminaries

Some basic definitions are provided to bring the main thought of this paper here.

2.1 Definition [18]

A continuous $t$- norm $\ast$ and $t$- conorm $\diamond$ are two continuous binary operations assigning $[0, 1] \times [0, 1] \rightarrow [0, 1]$ and obey the under stated principles:
(i) $\ast$ and $\diamond$ are both commutative and associative.
(ii) $x \ast 1 = 1 \ast x = x$ and $x \diamond 0 = 0 \diamond x = x, \forall x \in [0, 1]$.
(iii) $x \ast y \leq p \ast q$ and $x \diamond y \leq p \diamond q$ if $x \leq p, y \leq q$ with $x, y, p, q \in [0, 1]$.

$x \ast y = xy, x \ast y = \min\{x, y\}, x \ast y = \max\{x + y - 1, 0\}$ are most useful $t$-norms and $x \diamond y = x + y - xy, x \diamond y = \max\{x + y, 1\}$ are most useful $t$-conorms.

2.2 Definition [3]

An NS $Q$ on an initial universe $X$ is presented by three characterisations namely true value $T_Q$, indeterminant value $I_Q$ and false value $F_Q$ so that $T_Q, I_Q, F_Q : X \rightarrow [0, 1]$. Thus $Q$ can be designed as : $\{< u, (T_Q(u), I_Q(u), F_Q(u)) >: u \in X\}$ with $0 \leq \sup T_Q(u) + \sup I_Q(u) + \sup F_Q(u) \leq 3$. Here $1^+ = 1 + \delta$, where 1 is standard part and $\delta$ is non-standard part. Similarly $-0 = 0 - \delta$. The non-standard set $]-0, 1[$ is basically practiced in philosophical ground and because of the difficulty to adopt it in real field, the standard subset of $]-0, 1[$ i.e., $[0, 1]$ is applicable in real neutrosophic environment.

2.3 Definition [6]

An SVN-set $Q$ over a universe $X$ is a set $Q = \{< x, T_Q(x), I_Q(x), F_Q(x) >: x \in X$ and $T_Q(x), I_Q(x), F_Q(x) \in [0, 1]\}$ with $0 \leq \sup T_Q(x) + \sup I_Q(x) + \sup F_Q(x) \leq 3$

2.4 Definition [7]

Let $a_i, b_i, c_i, d_i \in \mathbb{R}$ (the set of all real numbers) with $a_i \leq b_i \leq c_i \leq d_i$ ($i = 1, 2, 3$) and $w_{\bar{p}}, u_{\bar{p}}, y_{\bar{p}} \in [0, 1] \subset \mathbb{R}$. Then an SVN-number $\bar{p} = (\{[a_1, b_1, c_1, d_1]; w_{\bar{p}}\}, \{[a_2, b_2, c_2, d_2]; u_{\bar{p}}\}, \{[a_3, b_3, c_3, d_3]; y_{\bar{p}}\})$ is a special SVN-set on $\mathbb{R}$ whose true value, indeterminant value, false value are respectively defined by the mappings $T_{\bar{p}} : \mathbb{R} \rightarrow [0, w_{\bar{p}}]$, $I_{\bar{p}} : \mathbb{R} \rightarrow [u_{\bar{p}}, 1]$, $F_{\bar{p}} : \mathbb{R} \rightarrow [y_{\bar{p}}, 1]$ and they are given as :

$$T_{\bar{p}}(x) = \begin{cases} g_T^l(x), & a_1 \leq x \leq b_1, \\ w_{\bar{p}}, & b_1 \leq x \leq c_1, \\ g_T^r(x), & c_1 \leq x \leq d_1, \\ 0, & \text{otherwise.} \end{cases}$$

$$I_{\bar{p}}(x) = \begin{cases} g_I^l(x), & a_2 \leq x \leq b_2, \\ u_{\bar{p}}, & b_2 \leq x \leq c_2, \\ g_I^r(x), & c_2 \leq x \leq d_2, \\ 1, & \text{otherwise.} \end{cases}$$

$$F_{\bar{p}}(x) = \begin{cases} g_F^l(x), & a_3 \leq x \leq b_3, \\ y_{\bar{p}}, & b_3 \leq x \leq c_3, \\ g_F^r(x), & c_3 \leq x \leq d_3, \\ 1, & \text{otherwise.} \end{cases}$$

The functions $g_T^l : [a_1, b_1] \rightarrow [0, w_{\bar{p}}], g_I^r : [c_2, d_2] \rightarrow [u_{\bar{p}}, 1], g_F^r : [c_3, d_3] \rightarrow [y_{\bar{p}}, 1]$ are continuous and non-decreasing functions satisfying : $g_T^l(a_1) = 0, g_T^l(b_1) = w_{\bar{p}}, g_I^r(c_2) = u_{\bar{p}}, g_I^r(d_2) = 1, g_F^l(c_3) = y_{\bar{p}}, g_F^l(d_3) = 1.$

---

*T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.*
The functions \( g^T_{x} : [c_1, d_1] \rightarrow [0, w_\tilde{p}], g^T_{y} : [a_2, b_2] \rightarrow [u_\tilde{p}, 1], g^F_{x} : [a_3, b_3] \rightarrow [y_\tilde{p}, 1] \) are continuous and non-increasing functions satisfying: \( g^T_{x}(c_1) = w_\tilde{p}, g^T_{x}(d_1) = 0, g^T_{y}(a_2) = 1, g^T_{y}(b_2) = u_\tilde{p}, g^F_{x}(a_3) = 1, g^F_{x}(b_3) = y_\tilde{p} \).

### 2.4.1 Definition [7]

If \( [a_1, b_1, c_1, d_1] = [a_2, b_2, c_2, d_2] = [a_3, b_3, c_3, d_3] \), then the SVN-number \( \tilde{p} \) is reduced to a single valued trapezoidal neutrosophic number as: \( \tilde{p} = \langle \langle [a_1, b_1, c_1, d_1]; w_\tilde{p}, u_\tilde{p}, y_\tilde{p} \rangle \rangle \).

### 2.5 Definition [17]

The \((\alpha, \beta, \gamma)\)-cut of an NS \( P \) is denoted by \( P_{(\alpha, \beta, \gamma)} \) and is defined as: \( P_{(\alpha, \beta, \gamma)} = \{ x \in X : T_P(x) \geq \alpha, I_P(x) \leq \beta, F_P(x) \leq \gamma \} \) with \( \alpha, \beta, \gamma \in [0, 1] \) and \( 0 \leq \alpha + \beta + \gamma \leq 3 \). Clearly, it is a crisp subset \( X \).

### 2.6 Definition [14]

In parametric form, a fuzzy number \( P \) is a pair \((P_L, P_R)\) of functions \( P_L(r), P_R(r), r \in [0, 1] \) satisfying the followings.

(i) Both are bounded functions.

(ii) \( P_L \) is monotone increasing left continuous and \( P_R \) is monotone decreasing right continuous function.

(iii) \( P_L(r) \leq P_R(r), 0 \leq r \leq 1 \).

A trapezoidal fuzzy number is put as \( P = (x_0, y_0, \delta, \zeta) \) where \([x_0, y_0] \) is interval defuzzifier and \( \delta > 0, \zeta > 0 \) are respectively called left fuzziness, right fuzziness. \((x_0 - \delta, y_0 + \zeta)\) is the support of \( P \) and its membership function is:

\[
P(x) = \begin{cases} 
\frac{1}{\delta}(x - x_0 + \delta), & x_0 - \delta \leq x \leq x_0, \\
1, & x \in [x_0, y_0], \\
\frac{1}{\zeta}(y_0 - x + \zeta), & y_0 \leq x \leq y_0 + \zeta, \\
0, & \text{otherwise}. 
\end{cases}
\]

In parametric form \( P_L(r) = x_0 - \delta + \delta r, P_R(r) = y_0 + \zeta - \zeta r \).

For arbitrary trapezoidal fuzzy numbers \( P = (P_L, P_R), Q = (Q_L, Q_R) \) and scalar \( k > 0 \), the addition and scalar multiplication are \( P + Q, kQ \) and they are defined by:

\[
(P + Q)_L(r) = P_L(r) + Q_L(r), \quad (P + Q)_R(r) = P_R(r) + Q_R(r) \quad \text{and} \quad (kQ)_L(r) = kQ_L(r), \quad (kQ)_R(r) = kQ_R(r).
\]

### 3 Generalised single valued neutrosophic number

Here, the structure of \( G_{SVN} \)-number, \( G_{SVTN} \)-number and \( G_{SVTrN} \)-number have been presented.

#### 3.1 Definition

- The support of three components of an SVN-set \( Q \) over \( X \) are given by a triplet \((S_{QT}, S_{QI}, S_{QF})\) where \( S_{QT} = \{ u \in X | T_Q(u) > 0 \} \), \( S_{QI} = \{ u \in X | I_Q(u) < 1 \} \), \( S_{QF} = \{ u \in X | F_Q(u) < 1 \} \).
- The height of the components of \( Q \) are given by a triplet \((H_{QT}, H_{QI}, H_{QF})\) where \( H_{QT} = \max\{ T_Q(u) | u \in X \} \), \( H_{QI} = \max\{ I_Q(u) | u \in X \} \), \( H_{QF} = \max\{ F_Q(u) | u \in X \} \).

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
3.1 Example

Define an SVN-set $Q$ on $\{0, 1, \cdots, 10\} \subset \mathbb{Z}$ (the set of integers) as : $\{< u, (\frac{u}{1+u}, 1-\frac{1}{2}, \frac{1}{1+u}) > | 0 \leq u \leq 10\}$. Then $S_{Q_T} = \{1, \cdots, 10\}$, $S_{Q_l} = \{0, \cdots, 10\}$, $S_{Q_F} = \{1, \cdots, 10\}$ and $H_{Q_T} = 0.909$ at $u = 10$, $H_{Q_l} = 0.999$ at $u = 10$, $H_{Q_F} = 1$ at $u = 0$.

3.2 Definition

A $G_{SVN}$-number $\tilde{p} = (([a_1, b_1, \sigma_1, \eta_1]; w_p), ([a_2, b_2, \sigma_2, \eta_2]; u_p), ([a_3, b_3, \sigma_3, \eta_3]; y_p))$ is a special SVN-set on $\mathbb{R}$ where $\sigma_i (> 0), \eta_i (> 0)$ are respectively called left spreads, right spreads and $[a_i, b_i]$ are the modal intervals of truth, indeterminacy and falsity functions for $i = 1, 2, 3$ respectively in $\tilde{p}$ and $w_p, u_p, y_p \in [0, 1] \subset \mathbb{R}$. The truth, indeterminacy and falsity functions are defined as follows:

$$T_{\tilde{p}}(x) = \begin{cases} \frac{1}{\sigma_1} w_p(x - a_1 + \sigma_1), & a_1 - \sigma_1 \leq x \leq a_1, \\ w_p, & x \in [a_1, b_1], \\ \frac{1}{\eta_1} w_p(b_1 - x + \eta_1), & b_1 \leq x \leq b_1 + \eta_1, \\ 0, & \text{otherwise.} \end{cases}$$

$$I_{\tilde{p}}(x) = \begin{cases} \frac{1}{\sigma_2} (a_2 - x + u_p(x - a_2 + \sigma_2)), & a_2 - \sigma_2 \leq x \leq a_2, \\ u_p, & x \in [a_2, b_2], \\ \frac{1}{\eta_2} (x - b_2 + u_p(b_2 - x + \eta_2)), & b_2 \leq x \leq b_2 + \eta_2, \\ 1, & \text{otherwise.} \end{cases}$$

$$F_{\tilde{p}}(x) = \begin{cases} \frac{1}{\sigma_3} (a_3 - x + y_p(x - a_3 + \sigma_3)), & a_3 - \sigma_3 \leq x \leq a_3, \\ y_p, & x \in [a_3, b_3], \\ \frac{1}{\eta_3} (x - b_3 + y_p(b_3 - x + \eta_3)), & b_3 \leq x \leq b_3 + \eta_3, \\ 1, & \text{otherwise.} \end{cases}$$

In parametric form, a $G_{SVN}$-number $\tilde{p}$ consists of three pairs $(T_p^1, T_p^u), (I_p^1, I_p^u), (F_p^1, F_p^u)$ of functions $T_p^1(r), T_p^u(r), I_p^1(r), I_p^u(r), F_p^1(r), F_p^u(r), r \in [0, 1]$ satisfying the followings.

(i) $T_p^1, T_p^u, F_p^u$ are bounded monotone increasing continuous function.

(ii) $T_p^u, I_p^1, F_p^1$ are bounded monotone decreasing continuous function.

(iii) $T_p^1(r) \leq T_p^u(r), I_p^1(r) \geq I_p^u(r), F_p^1(r) \leq F_p^u(r), r \in [0, 1]$.

3.2.1 Definition

• The support of the components of a $G_{SVN}$-number $\tilde{p}$ are given by a triplet $(S_{P_T}, S_{P_l}, S_{P_F})$ where $S_{P_T} = \{x \in \mathbb{R}|T_p(x) > 0\}, S_{P_l} = \{x \in \mathbb{R}|I_p(x) < 1\}, S_{P_F} = \{x \in \mathbb{R}|F_p(x) < 1\}$.

• The height of the components of $\tilde{p}$ are given by a triplet $(H_{P_T}, H_{P_l}, H_{P_F})$ where $H_{P_T} = w_p, H_{P_l} = 1 - u_p, H_{P_F} = 1 - y_p$.

• The boundaries of the truth function of $\tilde{p}$ are : $LB_{\tilde{p}} = (a_1 - \sigma_1, a_1)$ and $RB_{\tilde{p}} = (b_1, b_1 + \eta_1)$. $LB_{\tilde{p}}$ and $RB_{\tilde{p}}$ are respectively called left boundary and right boundary for truth function of $\tilde{p}$. Similarly, $LB_{\tilde{p}_I} = (a_2 - \sigma_2, a_2)$, $RB_{\tilde{p}_I} = (b_2, b_2 + \eta_2)$ and $LB_{\tilde{p}_F} = (a_3 - \sigma_3, a_3)$. $RB_{\tilde{p}_F} = (b_3, b_3 + \eta_3)$.

• The core for the truth function of $\tilde{p}$ is a set of points at which it’s height is measured. Similarly, the core for other two components are defined.

---

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
3.2.2 Example

Consider a $G_{SVN}$-number $\tilde{p}$ on $\mathbb{R}$ whose three components are as follows:

$$
T_{\tilde{p}}(x) = \begin{cases} 
\frac{0.6(x-11)}{4}, & x \in [11, 15] \\
0.6, & x \in [15, 25] \\
\frac{0.6(36-x)}{11}, & x \in [25, 36] \\
0, & \text{otherwise.}
\end{cases}
$$

Let $\tilde{p}_1 = (0.4, 0.3, 0.2)$ and $\tilde{p}_2 = (0.6, 0.7, 0.5)$. Then $S_{\tilde{p}_1} = (0.3, 0.4, 0.2)$ and $S_{\tilde{p}_2} = (0.6, 0.7, 0.5)$. For that $\tilde{p}$, $H_{\tilde{p}_1} = 0.6, H_{\tilde{p}_2} = 0.1$. Here, $LB_{\tilde{p}} = (11, 15), RB_{\tilde{p}} = (25, 36), LB_{\tilde{p}_1} = (4, 8), RB_{\tilde{p}_1} = (13, 20), LB_{\tilde{p}_2} = (23, 26), RB_{\tilde{p}_2} = (30, 38)$.

The core of truth, indeterminacy and falsity function are $[15, 25], [8, 13], [26, 30]$ respectively.

3.3 Definition

Let us assume two $G_{SVN}$-numbers $\tilde{p}$ and $\tilde{q}$ as follows:

$$
\tilde{p} = \langle [a_1, a'_1, \sigma_1, \eta_1]; w_\tilde{p} \rangle, \langle [a_2, a'_2, \sigma_2, \eta_2]; u_\tilde{p} \rangle, \langle [a_3, a'_3, \sigma_3, \eta_3]; y_\tilde{p} \rangle,
$$

$$
\tilde{q} = \langle [b_1, b'_1, \xi_1, \delta_1]; w_\tilde{q} \rangle, \langle [b_2, b'_2, \xi_2, \delta_2]; u_\tilde{q} \rangle, \langle [b_3, b'_3, \xi_3, \delta_3]; y_\tilde{q} \rangle.
$$

Then for any real number $x$,

(i) Image of $\tilde{p}$:

$$
-\tilde{p} = \langle [-a'_1, -a_1, \sigma_1, \eta_1]; w_{-\tilde{p}} \rangle, \langle [-a'_2, -a_2, \sigma_2, \eta_2]; u_{-\tilde{p}} \rangle, \langle [-a'_3, -a_3, \sigma_3, \eta_3]; y_{-\tilde{p}} \rangle.
$$

(ii) Addition:

$$
\tilde{p} + \tilde{q} = \langle [a_1 + b_1, a'_1 + b'_1, \sigma_1 + \xi_1, \sigma_1 + \eta_1 \alpha]; w_{\tilde{p} + \tilde{q}} \rangle, \langle [a_2 + b_2, a'_2 + b'_2, \sigma_2 + \xi_2, \sigma_2 + \eta_2 \alpha]; u_{\tilde{p} + \tilde{q}} \rangle, \langle [a_3 + b_3, a'_3 + b'_3, \sigma_3 + \xi_3, \sigma_3 + \eta_3 \alpha]; y_{\tilde{p} + \tilde{q}} \rangle.
$$

(iii) Scalar multiplication:

$$
x \tilde{p} = \langle [xa_1, xa'_1, \sigma_1 x, \eta_1 x]; w_{x \tilde{p}} \rangle, \langle [xa_2, xa'_2, \sigma_2 x, \eta_2 x]; u_{x \tilde{p}} \rangle, \langle [xa_3, xa'_3, \sigma_3 x, \eta_3 x]; y_{x \tilde{p}} \rangle
$$

for $x > 0$.

$$
x \tilde{p} = \langle [[xa'_1, xa_1, -x \sigma_1, -x \eta_1]; w_{x \tilde{p}} \rangle, \langle [[xa'_2, xa_2, -x \sigma_2, -x \eta_2]; u_{x \tilde{p}} \rangle, \langle [[xa'_3, xa_3, -x \sigma_3, -x \eta_3]; y_{x \tilde{p}} \rangle
$$

for $x < 0$.

3.4 Corollary

Let $\tilde{p} = \langle [a_1, b_1, \sigma_1, \eta_1]; w_\tilde{p} \rangle, \langle [a_2, b_2, \sigma_2, \eta_2]; u_\tilde{p} \rangle, \langle [a_3, b_3, \sigma_3, \eta_3]; y_\tilde{p} \rangle$ be an $G_{SVN}$-number.

1. Any $\alpha$-cut set of the $G_{SVN}$-number $\tilde{p}$ for truth function is denoted by $\tilde{p}_\alpha$ and is given by a closed interval as:

$$
\tilde{p}_\alpha = [L_{\tilde{p}}(\alpha), R_{\tilde{p}}(\alpha)] = [a_1 - \sigma_1 + \frac{\sigma_1 \alpha}{w_\tilde{p}}, b_1 + \eta_1 - \frac{\eta_1 \alpha}{w_\tilde{p}}], \quad \text{for } \alpha \in [0, w_\tilde{p}].
$$

The value of $\tilde{p}$ corresponding $\alpha$-cut set is denoted by $V_T(\tilde{p})$ and is calculated as:

$$
V_T(\tilde{p}) = \int_{0}^{w_\tilde{p}} [(a_1 - \sigma_1 + \frac{\sigma_1 \alpha}{w_\tilde{p}}) + (b_1 + \eta_1 - \frac{\eta_1 \alpha}{w_\tilde{p}})] \alpha \, d\alpha
$$

$$
= \int_{0}^{w_\tilde{p}} [a_1 + b_1 + \eta_1 - \sigma_1 - \frac{(\eta_1 - \sigma_1) \alpha}{w_\tilde{p}}] \alpha \, d\alpha
$$

$$
= \frac{1}{6} (3a_1 + 3b_1 - \sigma_1 + \eta_1) w_\tilde{p}^2.
$$

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
2. Any $\beta$-cut set of the $G_{SVN}$-number $\tilde{p}$ for indeterminacy membership function is denoted by $\tilde{p}^\beta$ and is given by a closed interval as:

$$\tilde{p}^\beta = [L^\beta_p(\beta), R^\beta_p(\beta)]$$

$$= \left[\frac{(u_\tilde{p} - \beta)\sigma_2 + (1 - u_\tilde{p})a_2}{1 - u_\tilde{p}}, \frac{(\beta - u_\tilde{p})\eta_2 + (1 - u_\tilde{p})b_2}{1 - u_\tilde{p}}\right], \text{ for } \beta \in [u_\tilde{p}, 1].$$

The value of $\tilde{p}$ corresponding $\beta$-cut set is denoted by $V_I(\tilde{p})$ and is calculated as:

$$V_I(\tilde{p}) = \int_{u_\tilde{p}}^{1} \left[\frac{(u_\tilde{p} - \beta)\sigma_2 + (1 - u_\tilde{p})a_2}{1 - u_\tilde{p}} + \frac{(\beta - u_\tilde{p})\eta_2 + (1 - u_\tilde{p})b_2}{1 - u_\tilde{p}}\right](1 - \beta)\,d\beta$$

$$= \int_{u_\tilde{p}}^{1} \left[a_2 + b_2 - \sigma_2 + \eta_2 + \frac{(\sigma_2 - \eta_2)(1 - \beta)}{1 - u_\tilde{p}}\right](1 - \beta)\,d\beta$$

$$= \frac{1}{6}(3a_2 + 3b_2 - \sigma_2 + \eta_2)(1 - u_\tilde{p})^2.$$

3. Any $\gamma$-cut set of the $G_{SVN}$-number $\tilde{p}$ for falsity membership function is denoted by $\gamma\tilde{p}$ and is given by a closed interval as:

$$\gamma\tilde{p} = [L^\gamma_p(\gamma), R^\gamma_p(\gamma)]$$

$$= \left[\frac{(u_\tilde{p} - \gamma)\sigma_3 + (1 - y_\tilde{p})a_3}{1 - y_\tilde{p}}, \frac{(\gamma - y_\tilde{p})\eta_3 + (1 - y_\tilde{p})b_3}{1 - y_\tilde{p}}\right], \text{ for } \gamma \in [y_\tilde{p}, 1].$$

The value of $\tilde{p}$ corresponding $\gamma$-cut set is denoted by $V_F(\tilde{p})$ and is calculated as:

$$V_F(\tilde{p}) = \int_{y_\tilde{p}}^{1} \left[\frac{(u_\tilde{p} - \gamma)\sigma_3 + (1 - y_\tilde{p})a_3}{1 - y_\tilde{p}} + \frac{(\gamma - y_\tilde{p})\eta_3 + (1 - y_\tilde{p})b_3}{1 - y_\tilde{p}}\right](1 - \gamma)\,d\gamma$$

$$= \int_{y_\tilde{p}}^{1} \left[a_3 + b_3 - \sigma_3 + \eta_3 + \frac{(\sigma_3 - \eta_3)(1 - \gamma)}{1 - y_\tilde{p}}\right](1 - \gamma)\,d\gamma$$

$$= \frac{1}{6}(3a_3 + 3b_3 - \sigma_3 + \eta_3)(1 - y_\tilde{p})^2.$$

3.5 Definition

For $\kappa \in [0, 1]$, the $\kappa$-weighted value of an $G_{SVN}$-number $\tilde{b}$ is denoted by $V_\kappa(\tilde{b})$ and is defined as:

$$V_\kappa(\tilde{b}) = \kappa^n V_T(\tilde{b}) + (1 - \kappa^n) V_I(\tilde{b}) + (1 - \kappa^n) V_F(\tilde{b}).$$

$n$ being any natural number.

Thus, the $\kappa$-weighted value for the $G_{SVN}$-number $\tilde{p}$ defined in Corollary 3.4 is:

$$V_\kappa(\tilde{p}) = \frac{1}{6}[(3a_1 + 3b_1 - \sigma_1 + \eta_1)\kappa^n w_\tilde{p}^2 + (3a_2 + 3b_2 - \sigma_2 + \eta_2)(1 - \kappa^n)(1 - u_\tilde{p})^2$$

$$+(3a_3 + 3b_3 - \sigma_3 + \eta_3)(1 - \kappa^n)(1 - y_\tilde{p})^2].$$

3.5.1 Property of $\kappa$-weighted value function

The $\kappa$-weighted value $V_\kappa(\tilde{p})$ and $V_\kappa(\tilde{q})$ of two $G_{SVN}$-numbers $\tilde{p}, \tilde{q}$ respectively obey the followings.

(i) $V_\kappa(\tilde{p} \pm \tilde{q}) \leq V_\kappa(\tilde{p}) + V_\kappa(\tilde{q})$, $V_\kappa(\tilde{p} + \tilde{q}) \geq V_\kappa(\tilde{p}) \sim V_\kappa(\tilde{q})$.

(ii) $V_\kappa(\tilde{p} - \tilde{\mu}) = V_\kappa(0)$, $V_\kappa(\mu \tilde{p}) = \mu V_\kappa(\tilde{p})$ for $\mu$ being any real number.
(iii) $V_\kappa(\bar{p})$ is monotone increasing or decreasing or constant according as $V_T(\bar{p}) > V_I(\bar{p}) + V_F(\bar{p})$ or $V_T(\bar{p}) < V_I(\bar{p}) + V_F(\bar{p})$ or $V_T(\bar{p}) = V_I(\bar{p}) + V_F(\bar{p})$ respectively.

**Proof.** We shall here prove (vi) only. Others can be easily verified by taking any two $G_{SVN}$-numbers. Here,

$$V_\kappa(\bar{p}) = \kappa^n V_T(\bar{p}) + (1 - \kappa^n)(V_I(\bar{p}) + V_F(\bar{p}))$$

$$\frac{dV_\kappa(\bar{p})}{d\kappa} = n\kappa^{n-1}[V_T(\bar{p}) - (V_I(\bar{p}) + V_F(\bar{p})]$$

As $\kappa \in [0, 1]$, so $\frac{dV_\kappa(\bar{p})}{d\kappa} > , < = 0$ for $[V_T(\bar{p}) - (V_I(\bar{p}) + V_F(\bar{p})] > , < = 0$ respectively. This clears the fact.

### 3.6 Definition

Let $G_{SVN}(\mathbb{R})$ be the set of all $G_{SVN}$-numbers defined over $\mathbb{R}$. For $\kappa \in [0, 1]$, a mapping $\mathcal{R}_\kappa : G_{SVN}(\mathbb{R}) \rightarrow \mathbb{R}$ is called a ranking function and it is defined as : $\mathcal{R}_\kappa(\bar{a}) = V_\kappa(\bar{a})$ for $\bar{a} \in G_{SVN}(\mathbb{R})$.

For $\bar{a}, \bar{b} \in G_{SVN}(\mathbb{R})$, their ranking is defined as :

$$\bar{a} >_{\mathcal{R}_\kappa} \bar{b} \iff \mathcal{R}_\kappa(\bar{a}) > \mathcal{R}_\kappa(\bar{b}), \bar{a} <_{\mathcal{R}_\kappa} \bar{b} \iff \mathcal{R}_\kappa(\bar{a}) < \mathcal{R}_\kappa(\bar{b}), \bar{a} =_{\mathcal{R}_\kappa} \bar{b} \iff \mathcal{R}_\kappa(\bar{a}) = \mathcal{R}_\kappa(\bar{b}).$$

### 3.7 Definition

An $G_{SVN}$-number $\bar{p}$ is called a $G_{SVTN}$-number if three modal intervals in $\bar{p}$ are equal. Thus $\bar{p} = (\{[a_0, b_0, \sigma_1, \eta_1]; w_{\bar{p}}\}, ([a_0, b_0, \sigma_2, \eta_2]; u_{\bar{p}}), ([a_0, b_0, \sigma_3, \eta_3]; y_{\bar{p}}))$ is an $G_{SVTN}$-number whose truth, indeterminacy and falsity functions are as follows :

$$T_{\bar{p}}(x) = \begin{cases} \frac{1}{\sigma_1} w_{\bar{p}}(x - a_0 + \sigma_1), & a_0 - \sigma_1 \leq x \leq a_0, \\ \frac{1}{\sigma_1} w_{\bar{p}}, & x \in [a_0, b_0], \\ \frac{1}{\eta_1} w_{\bar{p}}(b_0 - x + \sigma_1), & b_0 \leq x \leq b_0 + \eta_1, \\ 0, & \text{otherwise}. \end{cases}$$

$$I_{\bar{p}}(x) = \begin{cases} \frac{1}{\sigma_2} (a_0 - x + u_{\bar{p}}(x - a_0 + \sigma_2)), & a_0 - \sigma_2 \leq x \leq a_0, \\ \frac{1}{\eta_2} (x - b_0 + u_{\bar{p}}(b_0 - x + \eta_2)), & b_0 \leq x \leq b_0 + \eta_2, \\ 1, & \text{otherwise}. \end{cases}$$

$$F_{\bar{p}}(x) = \begin{cases} \frac{1}{\sigma_3} (a_0 - x + y_{\bar{p}}(x - a_0 + \sigma_3)), & a_0 - \sigma_3 \leq x \leq a_0, \\ \frac{1}{\eta_3} (x - b_0 + y_{\bar{p}}(b_0 - x + \eta_3)), & b_0 \leq x \leq b_0 + \eta_3, \\ 1, & \text{otherwise}. \end{cases}$$

In parametric form for $r \in [0, 1] :

$$T_{\bar{p}}^t(r) = a_0 - \sigma_1 + \frac{\sigma_1 r}{w_{\bar{p}}}, \quad T_{\bar{p}}^a(r) = b_0 + \eta_1 - \frac{\eta_1 r}{w_{\bar{p}}};$$

$$I_{\bar{p}}^t(r) = \frac{(1 - u_{\bar{p}}) a_0 + (u_{\bar{p}} - r) \sigma_2}{1 - u_{\bar{p}}}, \quad I_{\bar{p}}^a(r) = \frac{(1 - u_{\bar{p}}) b_0 + (r - u_{\bar{p}}) \eta_2}{1 - u_{\bar{p}}};$$

$$F_{\bar{p}}^t(r) = \frac{(1 - y_{\bar{p}}) a_0 + (y_{\bar{p}} - r) \sigma_3}{1 - y_{\bar{p}}}, \quad F_{\bar{p}}^a(r) = \frac{(1 - y_{\bar{p}}) b_0 + (r - y_{\bar{p}}) \eta_3}{1 - y_{\bar{p}}}. $$

*T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.*
3.8 Definition

A $G_{SVTN}$-number $\tilde{p}$ is called a $G_{SVTN}$-number if the modal interval in $\tilde{p}$ is reduced to a modal point. Thus $\tilde{p} = \langle [a_0, \sigma_1; \eta_1]; w_\tilde{p} \rangle, ([a_0, \sigma_2, \eta_2]; u_\tilde{p}), ([a_0, \sigma_3, \eta_3]; y_\tilde{p})$ is a $G_{SVTN}$-number whose truth, indeterminacy and falsity functions are as follows:

$$T_\tilde{p}(x) = \begin{cases} \frac{1}{\sigma_1} w_\tilde{p}(x - a_0 + \sigma_1), & a_0 - \sigma_1 \leq x \leq a_0, \\ w_\tilde{p}, & x = a_0, \\ \frac{1}{\eta_1} w_\tilde{p}(a_0 - x + \eta_1), & a_0 \leq x \leq a_0 + \eta_1, \\ 0, & \text{otherwise.} \end{cases}$$

$$I_\tilde{p}(x) = \begin{cases} \frac{1}{\sigma_2} (a_0 - x + u_\tilde{p}(x - a_0 + \sigma_2)), & a_0 - \sigma_2 \leq x \leq a_0, \\ u_\tilde{p}, & x = a_0, \\ \frac{1}{\eta_2} (x - a_0 + u_\tilde{p}(a_0 - x + \eta_2)), & a_0 \leq x \leq a_0 + \eta_2, \\ 1, & \text{otherwise.} \end{cases}$$

$$F_\tilde{p}(x) = \begin{cases} \frac{1}{\sigma_3} (a_0 - x + y_\tilde{p}(x - a_0 + \sigma_3)), & a_0 - \sigma_3 \leq x \leq a_0, \\ y_\tilde{p}, & x = a_0, \\ \frac{1}{\eta_3} (x - a_0 + y_\tilde{p}(a_0 - x + \eta_3)), & a_0 \leq x \leq a_0 + \eta_3, \\ 1, & \text{otherwise.} \end{cases}$$

3.8.1 Definition

Let $\tilde{a}$ and $\tilde{b}$ be two $G_{SVTN}$-numbers as follows:

$$\tilde{a} = \langle [a, \sigma_1, \eta_1]; w_\tilde{a} \rangle, ([a, \sigma_2, \eta_2]; u_\tilde{a}), ([a, \sigma_3, \eta_3]; y_\tilde{a}) \rangle,$$

$$\tilde{b} = \langle [b, \xi_1, \delta_1]; w_\tilde{b} \rangle, ([b, \xi_2, \delta_2]; u_\tilde{b}), ([b, \xi_3, \delta_3]; y_\tilde{b}) \rangle.$$ Then for any real number $x$,

(i) Image of $\tilde{a}$:

$$\bar{a} = \langle [-a, \eta_1, \sigma_1]; w_\bar{a} \rangle, \langle [-a, \eta_2, \sigma_2]; u_\bar{a} \rangle, \langle [-a, \eta_3, \sigma_3]; y_\bar{a} \rangle \rangle.$$

(ii) Addition:

$$\bar{a} + \bar{b} = \langle [a + b, \sigma_1 + \xi_1, \eta_1 + \delta_1]; w_\bar{a} * w_\bar{b} \rangle, ([a + b, \sigma_2 + \xi_2, \eta_2 + \delta_2]; u_\bar{a} \circ u_\bar{b}),$$

$$([a + b, \sigma_3 + \xi_3, \eta_3 + \delta_3]; y_\bar{a} \circ y_\bar{b}) \rangle.$$

(iii) Scalar multiplication:

$$x\tilde{a} = \langle [xa, x\sigma_1, x\eta_1]; w_\tilde{a} \rangle, ([xa, x\sigma_2, x\eta_2]; u_\tilde{a}), ([xa, x\sigma_3, x\eta_3]; y_\tilde{a}) \rangle$$ for $x > 0$.

$$x\tilde{a} = \langle [xa, -x\eta_1, -x\sigma_1]; w_\tilde{a} \rangle, ([xa, -x\eta_2, -x\sigma_2]; u_\tilde{a}), ([xa, -x\eta_3, -x\sigma_3]; y_\tilde{a}) \rangle$$ for $x < 0$.

(iv) The $\kappa$-weighted value $V_\kappa(\bar{a})$ of $\tilde{a}$ is given as:

$$V_\kappa(\bar{a}) = \frac{1}{6} \left[ (6a - \sigma_1 + \eta_1) \kappa^nw_\bar{a}^2 + (6a - \sigma_2 + \eta_2)(1 - u_\bar{a})^2 + (6a - \sigma_3 + \eta_3)(1 - y_\bar{a})^2 \right] (1 - \kappa^n) \]$$

3.8.2 Remark

Definition 2.4.1 shows that the supports (i.e., the bases of trapeziums (triangles)) for truth, indeterminacy and falsity function are all same. Then the value of truth, indeterminacy and falsity function (i.e., the area of individual trapezium (triangle)) differs in respect to their corresponding height only. But by Definition 3.7, we consider different supports (i.e., bases of trapeziums (triangles) formed) for truth, indeterminacy and falsity functions. Thus we can allow the supports and heights together to differ the value of truth, indeterminacy and falsity functions.
falsity functions in the present study. Briefly, Definition 2.4.1 is a particular case of Definition 3.7. Hence decision maker has a scope of flexibility to choose and compare different $G_{SVN}$-numbers in their study. The facts are shown by the graphical Figure 1 and 2. Figure 1 and Figure 2 represent Definition 2.4.1 and Definition 3.7 respectively.

### 3.9 Definition

1. The zero $G_{SVTN}$-number is denoted by $\tilde{0}$ and is defined as :

   $\tilde{0} = \langle (0, 0, 0, 0); 1, 0, 0, 0; 0, 0, 0; 0 \rangle$.

2. The zero $G_{SVTN}$-number is denoted by $\tilde{0}$ and is defined as :

   $\tilde{0} = \langle (0, 0, 0, 0); 1, 0, 0, 0; 0, 0, 0; 0 \rangle$.

### 4 Neutrosophic Linear Programming Problem

Before to discuss the main result, we shall remember the crisp concept of an $LP$-problem. The standard form of an $LP$-problem is :

$$\text{Max } z = cx \quad \text{such that } Ax = b, \ x \geq 0$$

where $c = (c_1, c_2, \cdots, c_n), b = (b_1, b_2, \cdots, b_n)^t$ and $A = [a_{ij}]_{m \times n}$.

In this problem, all the parameters are crisp. we shall now define $NLP$-problem.

### 4.1 Definition

An $LP$-problem having some parameters as $G_{SVN}$-number is called an $NLP$-problem. Considering the coefficient of the variables in the objective function in an $LP$-problem in term of $G_{SVN}$-numbers, an $NLP$-problem is designed as follows :

$$\text{Max } \tilde{z} = r_{\tilde{c}} \tilde{x} \quad \text{such that } Mx = b; \ x \geq 0$$

\[ (4.1) \]
where \( b \in \mathbb{R}^m, x \in \mathbb{R}^n, M \in \mathbb{R}^{m \times n}, c^j \in (G_{SVN}(\mathbb{R}))^n \) and \( \mathcal{R}_\kappa \) is a ranking function.

### 4.2 Definition

1. \( x \in \mathbb{R}^n \) is a feasible solution to equation (4.1) if \( x \) satisfies the constraints of that.
2. A feasible solution \( x^* \) is an optimal solution if for all solutions \( x \) to (4.1), \( \tilde{c}x^* \geq_{\mathcal{R}_\kappa} \tilde{c}x \).
3. For the NLP-problem (4.1), suppose \( \text{rank}(M, b) = \text{rank}(M) = m \). \( M \) is partitioned as \([B, N]\) where \( B \) is a non-singular \( m \times m \) matrix i.e., \( \text{rank}(B) = m \). A feasible solution \( x = (x_B, x_N)^t \) to (4.1) obtained by setting \( x_B = B^{-1}b, x_N = 0 \) is called a neutrosophic basic feasible solution (\( N_{BFS} \)). Here \( B \) and \( N \) are respectively called basis and non basis matrix. \( x_B \) is called a basic variable and \( x_N \) is called a non-basic variable.
4. In an \( N_{BFS} \) if all components of \( x_B > 0 \), then \( x \) is non-degenerate \( N_{BFS} \) and if at least one component of \( x_B = 0 \), then \( x \) is degenerate \( N_{BFS} \).

### 5 Simplex Method for NLP-problem

The NLP-problem (4.1) can be put as follows:

\[
\begin{align*}
\text{Max } \tilde{z} &=_{\mathcal{R}_\kappa} \tilde{c}_B x_B + \tilde{c}_N x_N \\
\text{such that } &B x_B + N x_N = b; \quad x_B, x_N \geq 0
\end{align*}
\]

where the characters \( B, N, x_B \) and \( x_N \) are already stated. Then we have,

\[
\begin{align*}
x_B + B^{-1}N x_N &= B^{-1}b \\
\Rightarrow \tilde{c}_B x_B + \tilde{c}_B B^{-1}N x_N &=_{\mathcal{R}_\kappa} \tilde{c}_B B^{-1}b \\
\Rightarrow \tilde{z} &=_{\mathcal{R}_\kappa} \tilde{c}_N x_N + \tilde{c}_B B^{-1}N x_N =_{\mathcal{R}_\kappa} \tilde{c}_B B^{-1}b \\
\Rightarrow \tilde{z} + (\tilde{c}_B B^{-1}N - \tilde{c}_N) x_N &=_{\mathcal{R}_\kappa} \tilde{c}_B B^{-1}b.
\end{align*}
\]

(5.1)

(5.2)

For an \( N_{BFS} \), treating \( x_N = 0 \), we have \( x_B = B^{-1}b \) and \( \tilde{z} =_{\mathcal{R}_\kappa} \tilde{c}_B B^{-1}b \) from (5.1) and (5.2), respectively. We can rewrite the NLP-problem as given in Table 1.

### Table 1: Tabular form of an NLP-problem.

<table>
<thead>
<tr>
<th>( \tilde{c}_j )</th>
<th>( \tilde{c}_B )</th>
<th>( \tilde{c}_N )</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{z} )</td>
<td>( \tilde{z}_B )</td>
<td>( \tilde{z}_N )</td>
<td>( B^{-1}b )</td>
</tr>
<tr>
<td>( \tilde{z} )</td>
<td>1</td>
<td>0</td>
<td>( \tilde{c}_B B^{-1}N - \tilde{c}_N )</td>
</tr>
</tbody>
</table>

We can get all required initial information to proceed with the simplex method from Table 1. The neutrosophic cost row in the Table 1 is \( \tilde{\lambda}_j =_{\mathcal{R}_\kappa} (\tilde{c}_B B^{-1}a_j - c_j) a_j \notin B \) giving \( \tilde{\lambda}_j =_{\mathcal{R}_\kappa} (\tilde{z}_j - \tilde{c}_j) \) for non-basic variables. The optimality arises if \( \tilde{\lambda}_j \geq_{\mathcal{R}_\kappa} \tilde{0} \), \( \forall a_j \notin B \). If \( \tilde{\lambda}_l <_{\mathcal{R}_\kappa} \tilde{0} \) for any \( a_l \notin B \), we need to replace \( x_{B_l} \) by \( x_l \). We then compute \( y_l = B^{-1}a_l \). If \( y_l \leq 0 \), then \( x_l \) can be increased indefinitely and so the problem admits unbounded optimal solution. But if \( y_l \) has at least one positive component, then one of the current basic variables blocks that increase, which drops to zero.

### 5.1 Theorem

In every column \( a_j \) of \( M \), if \( \tilde{z}_j - \tilde{c}_j \geq_{\mathcal{R}_\kappa} \tilde{0} \) holds for an \( N_{BFS} x_B \) of the NLP-problem (4.1) then it is an optimal solution to that.
Theorem

Let $M = [a_{ij}]_{m \times n} = [a_1, a_2, \ldots, a_n]$ where each $a_i = (a_{1i}, a_{2i}, \ldots, a_{mi})^T$ is $m$ component column vector. Suppose $B = [\eta_1, \eta_2, \ldots, \eta_m]$ is the basis matrix and $\bar{z}_B = \eta_\kappa \bar{c}_B x_B = \eta_\kappa \sum_{i=1}^m \bar{c}_B x_{Bi}$, where $\bar{c}_B$ is the price corresponding to the basic variable $x_B$. Then any column $a_i$ of $M$ may be put as a linear combination of the vectors $\eta_1, \eta_2, \ldots, \eta_m$ of $B$. Let

$$a_i = y_{i1}\eta_1 + y_{i2}\eta_2 + \cdots + y_{im}\eta_m = \sum_{i=1}^m y_{il}\eta_l = By_i \Rightarrow y_i = B^{-1}a_i.$$  

where $y_i = (y_{i1}, y_{i2}, \ldots, y_{im})^T$ being $m$ component scalars represents $a_i$, the $l$-th vector of $M$. Assume that $\bar{z}_i = \eta_\kappa \bar{c}_B y_i = \eta_\kappa \sum_{i=1}^m \bar{c}_B y_{il}$.

Let $x = [x_1, x_2, \ldots, x_n]^T$ be any other feasible solution of the $NLP$-problem (4.1) and $\tilde{z}$ be the corresponding objective function. Then,

$$Bx_B = b = Mx \Rightarrow x_B = B^{-1}(Mx) = (B^{-1}M)x = yx$$

where $B^{-1}M = y = [y_{ij}]_{m \times n} = [y_1, y_2, \ldots, y_n]$ with $y_i$ defined as above. Thus,

$$\begin{pmatrix} x_{B1} \\ x_{B2} \\ \vdots \\ x_{Bn} \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Equating $i$-th component from both sides, we have $x_{Bi} = \sum_{j=1}^n y_{ij} x_j$. Now,

$$\tilde{z}_j - \bar{c}_j \geq \eta_\kappa \bar{0} \Rightarrow (\tilde{z}_j - \bar{c}_j) x_j \geq \eta_\kappa \bar{0} \text{ [as } x_j > 0 \text{]} \Rightarrow \sum_{j=1}^n (\tilde{z}_j - \bar{c}_j) x_j \geq \eta_\kappa \bar{0}$$

$$\Rightarrow \sum_{j=1}^n \tilde{z}_j x_j - \sum_{j=1}^n \bar{c}_j x_j \geq \eta_\kappa \bar{0} \Rightarrow \sum_{j=1}^n x_j (\bar{c}_B y_j) - \tilde{z} \geq \eta_\kappa \bar{0}$$

$$\Rightarrow \sum_{j=1}^n x_j (\sum_{i=1}^m \bar{c}_B y_{ij}) - \tilde{z} \geq \eta_\kappa \bar{0} \Rightarrow \sum_{i=1}^m \bar{c}_B (\sum_{j=1}^n y_{ij} x_j) - \tilde{z} \geq \eta_\kappa \bar{0}$$

$$\Rightarrow \sum_{i=1}^m \bar{c}_B \bar{x}_{Bi} - \tilde{z} \geq \eta_\kappa \bar{0} \Rightarrow \tilde{z}_B - \tilde{z} \geq \eta_\kappa \bar{0}.$$  

Thus $\tilde{z}_B$ is the maximum value of the objective function. This optimality criterion holds for all non-basic vectors of $M$. If $a_l$ be in the basis matrix $B$, say $a_l = \eta_l$, then

$$a_l = \eta_l = 0.\eta_1 + 0.\eta_2 + \cdots + 0.\eta_{l-1} + 1.\eta_l + 0.\eta_{l+1} + \cdots + 0.\eta_m$$

i.e., $y_l$ is a unit vector $e_l$ with $l$-th component unity.

Since $a_l = \eta_l$, we have $\bar{c}_l = \bar{c}_{Bl}$ and so

$$\tilde{z}_l - \bar{c}_l = \eta_\kappa (\bar{c}_B y_l - \bar{c}_l) = \eta_\kappa (\bar{c}_B e_l - \bar{c}_l) = \eta_\kappa (\bar{c}_l - \bar{c}_B) = \eta_\kappa \bar{0}.$$  

Thus as a whole $\tilde{z}_j - \bar{c}_j \geq \eta_\kappa \bar{0}$ is the necessary condition for optimality.

5.2 Theorem

A non-degenerate $N_{BFS} x_B = B^{-1}b, x_N = 0$ is optimal to $NLP$-problem (4.1) iff $\tilde{z}_j - \bar{c}_j \geq \eta_\kappa \bar{0}, \forall 1 \leq j \leq n$.  

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
Proof. Suppose \( x^* = (x_B^*, x_N^*) \) be an \( N_{BFS} \) to (4.1) where \( x_B = B^{-1}b, x_N = 0 \). If \( \tilde{z}^* \) be the objective function corresponding to \( x^* \), then \( \tilde{z}^* =_{\mathfrak{R}_n} \tilde{c}_Bx_B + \tilde{c}_Nx_N =_{\mathfrak{R}_n} \tilde{c}_BB^{-1}b \). Let \( x = [x_1, x_2, \cdots, x_n]^t \) be another feasible solution of \( NLP \)-problem (4.1) and \( \tilde{z} \) be the corresponding objective function. Then,

\[
\tilde{z} =_{\mathfrak{R}_n} \tilde{c}_Bx_B + \tilde{c}_Nx_N =_{\mathfrak{R}_n} \tilde{c}_BB^{-1}b - \sum_{a_j \notin B} (\tilde{c}_B B^{-1} a_j - \tilde{c}_j) x_j \]

This shows that the solution is optimal iff \( \tilde{z}_j - \tilde{c}_j \geq_{\mathfrak{R}_n} 0 \) for all \( 1 \leq j \leq n \).

5.3 Theorem

For any \( N_{BFS} \) to \( NLP \)-problem (4.1), if there is some column not in basis such that \( \tilde{z}_l - \tilde{c}_l <_{\mathfrak{R}_n} 0 \) and \( y_{il} \leq 0 \), \( i = 1, 2, \cdots, m \), then (4.1) admits an unbounded solution.

Proof. Let \( x_B \) be a basic solution to the \( NLP \)-problem (4.1). Re-writing the constraints,

\[
Bx_B + Nx_N = b
\]

\[
\Rightarrow x_B + B^{-1}Nx_N = B^{-1}b
\]

\[
\Rightarrow x_B + B^{-1} \sum_j (a_j x_j) = B^{-1}b, \text{ a}_j s \text{ are the columns of } N
\]

\[
\Rightarrow x_B + \sum_j (B^{-1}a_j x_j) = B^{-1}b
\]

\[
\Rightarrow x_B + \sum_j (y_j x_j) = y_0, \text{ where } a_j = By_j, a_j \notin B
\]

\[
\Rightarrow x_B + \sum_j (y_{ij} x_j) = y_{i0}, \text{ } 1 \leq i \leq m, 1 \leq j \leq n
\]

\[
\Rightarrow x_B = y_{i0} - (\sum_j y_{ij} x_j), \text{ } 1 \leq i \leq m, 1 \leq j \leq n.
\]

If \( x_l \) enters into the basis, then \( x_l > 0 \) and \( x_j = 0 \) for \( j \neq B \cup l \). Since \( y_{il} \leq 0 \), \( 1 \leq i \leq m \) hence \( y_{i0} - y_{il} x_l \geq 0 \). So, the basic solution remains feasible and for that, the objective function is:

\[
\tilde{z}^* =_{\mathfrak{R}_n} \tilde{c}_B x_B + \tilde{c}_N x_N =_{\mathfrak{R}_n} \sum_{i=1}^m \tilde{c}_B i (y_{i0} - y_{il} x_l) + \tilde{c}_l x_l =_{\mathfrak{R}_n} \sum_{i=1}^m \tilde{c}_B i y_{i0} - (\sum_{i=1}^m \tilde{c}_B i y_{il} - \tilde{c}_l) x_l
\]

\[
=_{\mathfrak{R}_n} \tilde{c}_B y_{i0} - (\tilde{c}_B y_{il} - \tilde{c}_l) x_l =_{\mathfrak{R}_n} \tilde{z} - (\tilde{z}_l - \tilde{c}_l) x_l.
\]

It shows that \( \tilde{z}^* >_{\mathfrak{R}_n} \tilde{z} \), as \( \tilde{z}_l - \tilde{c}_l <_{\mathfrak{R}_n} 0 \) and this completes the fact.

5.4 Simplex algorithm for solving \( NLP \)-problem

To solve any \( NLP \)-problem by simplex method, the existence of an initial basic feasible solution is always assumed. This solution will be optimised through some iterations. The required steps are as follows:

Step 1. Check whether the objective function of the given \( NLP \)-problem is to be maximized or minimized. If it is to be minimized, then it is converted into a maximization problem by using the result \( Min(\tilde{z}) = -Max(-\tilde{z}) \).

Step 2. Convert all the inequations of the constraints (\( \leq \) type) into equations by introducing slack variables. Put the costs of the respective variables equal to 0.

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
Step 3. Obtain an $N_{BFS}$ to the problem in the form $x_B = B^{-1}b = y_0$ and $x_N = 0$. The corresponding objective function is $z = c_B B^{-1} b = c_B y_0$.

Step 4. For each basic variable, put $\lambda_B = =_{\Re} z_B - c_B = =_{\Re} 0$. For each non-basic variable, calculate $\lambda_j = =_{\Re} z_j - c_j = =_{\Re} c_B B^{-1} a_j - c_j$ in the current iteration. If all $z_j - c_j \geq =_{\Re} 0$, then the present solution is optimal.

Step 5. If for some non-basic variables, $\lambda_j = =_{\Re} z_j - c_j < =_{\Re} 0$ then find out $\lambda_1 = =_{\Re} \min \{ \lambda_j \}$. If $y_{il} < 0$ for all $i = 1, \ldots, m$, then the given problem will have unbounded solution and stop the iteration. Otherwise to determine the index of the variable $x_B$, that is to be removed from the current basis, compute

$$\frac{u_{il}}{y_{rl}} = \min \{ \frac{u_{il}}{y_{rl}} : y_{il} > 0, 1 \leq i \leq m \}.$$ 

Step 6. Update $y_{i0}$ by replacing $y_{i0} = =_{\Re} u_{il}/y_{rl}$ for $i \neq r$ and $y_{r0}$ by $=_{\Re} u_{il}/y_{rl}$.

Step 7. Construct new basis and repeat the Step 4, Step 5 until the optimality is reached.

Step 8. Find the optimal solution and hence the optimal value of objective function.

6 Numerical Example

The $NLP$-problems with both $G_{SVT\,N}$-number and $G_{SVT\,N}$-number are solved by the use of proposed algorithm. For simplicity, we define the $\kappa$-weighted value function for $n = 1$ in rest of the paper.

6.1 Example

Two friends $F_1$ and $F_2$ wish to invest in a raising share market. They choose two particular shares $S_1$ and $S_2$ of two multinational companies. They also decide to purchase equal unit of two shares individually. The maximum investment of $F_1$ is Rs. 4000 and that of $F_2$ is Rs. 7000. The price per unit of $S_1$ and $S_2$ are Re. 1 and Rs. 3, respectively when $F_1$ purchases. These are Rs. 2 and Rs. 5 at the time of purchasing of share by $F_2$. The current value of share $S_1$ and $S_2$ per unit is Rs. $\tilde{c}_1$ and Rs. $\tilde{c}_2$ (given in $G_{SV\,N}$-numbers), respectively. Now if they sell their shares, formulate an $NLP$-problem to maximize their returns.

The problem can be summarised as follows:

<table>
<thead>
<tr>
<th>Friends $\downarrow$</th>
<th>Shares : $S_1$</th>
<th>$S_2$</th>
<th>Purchasing capacity $\downarrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>Re. 1</td>
<td>Rs. 3</td>
<td>Rs. 4000</td>
</tr>
<tr>
<td>$F_2$</td>
<td>Rs. 2</td>
<td>Rs. 5</td>
<td>Rs. 7000</td>
</tr>
<tr>
<td>Price per unit $\Rightarrow$</td>
<td>$\tilde{c}_1$</td>
<td>$\tilde{c}_2$</td>
<td></td>
</tr>
</tbody>
</table>

Let they individually purchase $x_1$ units of share $S_1$ and $x_2$ units of share $S_2$. The problem is formulated as:

$$\text{Max } \tilde{z} = =_{\Re} \tilde{c}_1 x_1 + \tilde{c}_2 x_2$$

such that

$$x_1 + 3x_2 \leq 4000$$

$$2x_1 + 5x_2 \leq 7000; \; x_1, x_2 \geq 0$$

It is an $NLP$-problem where $\tilde{c}_1 = =_{\Re} (\{5, 8, 1, 3\}; 0.2), (\{5, 8, 3, 4\}; 0.3), (\{5, 8, 2, 1\}; 0.4)$ and $\tilde{c}_2 = =_{\Re} (\{3, 7, 2, 4\}; 0.3), (\{3, 7, 1, 3\}; 0.5), (\{3, 7, 2, 5\}; 0.6)$ are two $G_{SVT\,N}$-numbers with a pre-assigned $\kappa = 0.45$. 

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
Rewriting the given constraints by introducing slack variables:

\[
\begin{align*}
    x_1 + 3x_2 + x_3 &= 4000 \\
    2x_1 + 5x_2 + x_4 &= 7000 \\
    x_1, x_2, x_3, x_4 &\geq 0
\end{align*}
\]

We take the \( t \)-norm and \( s \)-norm as \( p \ast q = \min\{p, q\} \) and \( p \circ q = \max\{p, q\} \), respectively. The first feasible simplex table is as follows:

<table>
<thead>
<tr>
<th>( \tilde{c}_j \Rightarrow )</th>
<th>( \tilde{c}_1 )</th>
<th>( \tilde{c}_2 )</th>
<th>( 0 )</th>
<th>( 0 )</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_B \downarrow )</td>
<td>( x_1 )</td>
<td>( x_2 )</td>
<td>( x_3 )</td>
<td>( x_4 )</td>
<td></td>
</tr>
<tr>
<td>( x_3 )</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4000</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>7000→</td>
</tr>
</tbody>
</table>

Here \( \tilde{c}_1^{(1)} = -\tilde{c}_1 = \langle [-8, -5, 3, 1]; 0.2 \rangle, \langle [-8, -5, 4, 3]; 0.3 \rangle, \langle [-8, -5, 1, 2]; 0.4 \rangle \rangle \),

\( \tilde{c}_2^{(1)} = -\tilde{c}_2 = \langle [-7, -3, 4, 2]; 0.3 \rangle, \langle [-7, -3, 3, 1]; 0.5 \rangle, \langle [-7, -3, 5, 2]; 0.6 \rangle \rangle \)

and \( V_\kappa(\tilde{c}_3^{(1)}) = V_\kappa(\tilde{c}_4^{(1)}) = V_\kappa(0) \).

Then \( V_\kappa(\tilde{c}_1^{(1)}) = \frac{1}{6}(31.64\kappa - 33.28) \) and \( V_\kappa(\tilde{c}_2^{(1)}) = \frac{1}{6}(10.4\kappa - 13.28) \) by Definition 3.5.

Clearly \( V_\kappa(\tilde{c}_1^{(1)}) < 0, V_\kappa(\tilde{c}_2^{(1)}) < 0 \) and \( V_\kappa(\tilde{c}_1^{(1)}) = V_\kappa(\tilde{c}_2^{(1)}) = 0 \) for \( \kappa = 0.45 \).

Then \( \tilde{c}_1^{(1)} \neq \tilde{c}_2^{(1)} \). So \( x_1 \) enters in the basis and as \( \min\{4000/1, 7000/2 \} = 3500 \), the leaving variable is \( x_4 \). The revised table is:

<table>
<thead>
<tr>
<th>( \tilde{c}_j \Rightarrow )</th>
<th>( \tilde{c}_1 )</th>
<th>( \tilde{c}_2 )</th>
<th>( 0 )</th>
<th>( 0 )</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_B \downarrow )</td>
<td>( x_1 )</td>
<td>( x_2 )</td>
<td>( x_3 )</td>
<td>( x_4 )</td>
<td></td>
</tr>
<tr>
<td>( x_3 )</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>-1/2</td>
<td>500</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>1</td>
<td>5/2</td>
<td>0</td>
<td>1/2</td>
<td>3500</td>
</tr>
</tbody>
</table>

where \( V_\kappa(\tilde{c}_3^{(2)}) = V_\kappa(\tilde{c}_4^{(2)}) = V_\kappa(0) \) and

\[
\begin{align*}
    \tilde{c}_2^{(2)} &= \frac{5}{2}\tilde{c}_1 - \tilde{c}_2 \\
    &= 2.5\langle [5, 8, 1, 3]; 0.2 \rangle, \langle [5, 8, 3, 4]; 0.3 \rangle, \langle [5, 8, 2, 1]; 0.4 \rangle \rangle \\
    &\quad - \langle [3, 7, 2, 4]; 0.3 \rangle, \langle [3, 7, 1, 3]; 0.5 \rangle, \langle [3, 7, 2, 5]; 0.6 \rangle \rangle \\
    &= \langle [5.5, 17, 6.5, 9.5]; 0.2 \rangle, \langle [5.5, 17, 10.5, 11]; 0.5 \rangle, \langle [5.5, 17, 10, 4.5]; 0.6 \rangle \rangle.
\end{align*}
\]

\[
\begin{align*}
    \tilde{c}_4^{(2)} &= \frac{1}{2}\tilde{c}_1 = \langle [2.5, 4, 0.5, 1.5]; 0.2 \rangle, \langle [2.5, 4, 1.5, 2]; 0.3 \rangle, \langle [2.5, 4, 1, 0.5]; 0.4 \rangle \rangle.
\end{align*}
\]

Then \( V_\kappa(\tilde{c}_2^{(2)}) = \frac{1}{6}(26.92 - 24.1\kappa) \) and \( V_\kappa(\tilde{c}_4^{(2)}) = \frac{1}{6}(16.64 - 15.82\kappa) \) by Definition 3.5.

Clearly \( V_\kappa(\tilde{c}_2^{(2)}) > 0 \) and \( V_\kappa(\tilde{c}_4^{(2)}) > 0 \) for \( \kappa = 0.45 \).
Hence the optimality arises and Max $\tilde{z} = 3500\tilde{c}_1$, which, using $\kappa$ - weighted function, becomes Rs. 11107 approximately. Then corresponding return of $F_1$ and $F_2$ becomes Rs. 7607 and of Rs. 4107 respectively.

6.1.1 Example

Consider the NLP-problem defined in Example 6.1 with a pre-assigned $\kappa = 0.96$.

The initial simplex table (Table 5) is same as Table 3.

Table 5: First iteration

<table>
<thead>
<tr>
<th>$\bar{c}_j \Rightarrow$</th>
<th>$\bar{c}_1$</th>
<th>$\bar{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4000 →</td>
</tr>
<tr>
<td>$x_4$</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>7000</td>
</tr>
<tr>
<td>$\tilde{z} \Rightarrow$</td>
<td>$\bar{c}_1^{(1)}$</td>
<td>$\bar{c}_2^{(1)}$</td>
<td>$\bar{c}_3^{(1)}$</td>
<td>$\bar{c}_4^{(1)}$</td>
<td></td>
</tr>
</tbody>
</table>

Here $V_\kappa(\bar{c}_1^{(1)}) = V_\kappa(\bar{c}_4^{(1)}) = V_\kappa(\bar{0})$ and $V_\kappa(\bar{c}_1^{(1)}) < 0$, $V_\kappa(\bar{c}_2^{(1)}) < 0$ with $V_\kappa(\bar{c}_1^{(1)}) - V_\kappa(\bar{c}_2^{(1)}) > 0$ for $\kappa = 0.96$. Then $\bar{c}_1^{(1)} > \bar{\kappa}_\kappa(\bar{c}_2^{(1)})$. So $x_2$ enters the basis and as $\min\{\frac{4000}{3}, \frac{7000}{3}\} = \frac{4000}{3}$, the leaving variable is $x_3$. The revised table is:

Table 6: Second iteration

<table>
<thead>
<tr>
<th>$\bar{c}_j \Rightarrow$</th>
<th>$\bar{c}_1$</th>
<th>$\bar{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>0</td>
<td>4000/3</td>
</tr>
<tr>
<td>$x_4$</td>
<td>1/3</td>
<td>0</td>
<td>-5/3</td>
<td>1</td>
<td>1000/3 →</td>
</tr>
<tr>
<td>$\tilde{z} \Rightarrow$</td>
<td>$\bar{c}_1^{(2)}$</td>
<td>$\bar{c}_2^{(2)}$</td>
<td>$\bar{c}_3^{(2)}$</td>
<td>$\bar{c}_4^{(2)}$</td>
<td>$\frac{4000}{3} - \bar{c}_2$</td>
</tr>
</tbody>
</table>

where $V_\kappa(\bar{c}_2^{(2)}) = V_\kappa(\bar{c}_4^{(2)}) = V_\kappa(\bar{0})$ and

$\bar{c}_1^{(2)} = \frac{1}{3} \bar{c}_2 - \bar{c}_1 = \langle([-7, -8/3, 11/3, 7/3]; 0.2),([-7, -8/3, 13/3, 4]; 0.5),([-7, -8/3, 5/3, 11/3]; 0.6)\rangle,$

$\bar{c}_3^{(2)} = \frac{1}{3} \bar{c}_2 = \langle([1, 7/3, 2/3, 4/3]; 0.3), ([1, 7/3, 1/3, 1]; 0.5), ([1, 7/3, 2/3, 5/3]; 0.6)\rangle.$

Then $V_\kappa(\bar{c}_1^{(2)}) = \frac{1}{18}(31.32\kappa - 34.96)$ and $V_\kappa(\bar{c}_3^{(2)}) = \frac{1}{18}(13.28 - 10.4\kappa)$.

Clearly $V_\kappa(\bar{c}_1^{(2)}) < 0$ and $V_\kappa(\bar{c}_3^{(2)}) > 0$ for $\kappa = 0.96$. So $x_1$ enters the basis and as $\min\{\frac{4000}{3}, \frac{1000}{3}\} = 1000$, the leaving variable is $x_4$. The revised table is:

Table 7: Third iteration

<table>
<thead>
<tr>
<th>$\bar{c}_j \Rightarrow$</th>
<th>$\bar{c}_1$</th>
<th>$\bar{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>1000 →</td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>0</td>
<td>-5</td>
<td>3</td>
<td>1000</td>
</tr>
<tr>
<td>$\tilde{z} \Rightarrow$</td>
<td>$\bar{c}_1^{(3)}$</td>
<td>$\bar{c}_2^{(3)}$</td>
<td>$\bar{c}_3^{(3)}$</td>
<td>$\bar{c}_4^{(3)}$</td>
<td>$1000(\bar{c}_1 + \bar{c}_2)$</td>
</tr>
</tbody>
</table>

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
where $V_\kappa(\tilde{c}_1^{(3)}) = V_\kappa(\tilde{c}_2^{(3)}) = V_\kappa(\tilde{0})$ and

$$
\tilde{c}_3^{(3)} = -5\tilde{c}_1 + 2\tilde{c}_2 = \langle([34, -11, 19, 13]; 0.2), ([34, -11, 22, 21]; 0.5), ([34, -11, 9, 20]; 0.6)\rangle,
$$

$$
\tilde{c}_4^{(3)} = 3\tilde{c}_1 - \tilde{c}_2 = \langle([8, 21, 7, 11]; 0.2), ([8, 21, 12, 13]; 0.5), ([8, 21, 11, 5]; 0.6)\rangle.
$$

Then $V_\kappa(\tilde{c}_3^{(3)}) = \frac{1}{6}(48.2\kappa - 53.84) < 0$ and $V_\kappa(\tilde{c}_4^{(3)}) = \frac{1}{6}(34.96 - 31.32\kappa) > 0$ for $\kappa = 0.96$. So $x_3$ enters in the basis and the leaving variable is $x_2$. The revised table is:

<table>
<thead>
<tr>
<th>$\tilde{c}_j$</th>
<th>$\tilde{c}_1$</th>
<th>$\tilde{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B \downarrow$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$500$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>-1/2</td>
<td>$3500$</td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>5/2</td>
<td>0</td>
<td>1/2</td>
<td>$3500$</td>
</tr>
</tbody>
</table>

where $V_\kappa(\tilde{c}_1^{(4)}) = V_\kappa(\tilde{c}_3^{(4)}) = V_\kappa(\tilde{0})$ and $\tilde{c}_2^{(4)} = \frac{5}{2}\tilde{c}_1 - \tilde{c}_2$ and $\tilde{c}_4^{(4)} = \frac{1}{2}\tilde{c}_1$. Then $V_\kappa(\tilde{c}_2^{(4)}) = \frac{1}{6}(26.92 - 24.1\kappa) > 0$ and $V_\kappa(\tilde{c}_4^{(4)}) = \frac{1}{6}(16.64 - 15.82\kappa) > 0$ for $\kappa = 0.96$.

Hence the optimality arises and the optimal solution is $x_1 = 3500, x_2 = 0$.

**6.1.2 Remark**

From Example 6.1 and Example 6.1.1, it is seen that the final simplex tables in both cases are same. So, if the optimality exists for an NLP-problem, the optimal solutions are always unique whatever the value of $\kappa$ assigned. Depending upon the chosen $\kappa$, the number of iteration to reach at optimality stage may vary but it does not affect the optimal solutions. However, the character $\kappa$ plays an important role to assign the optimal value of the objective function in a problem. The fact is shown in Table 9. So, the value of $\kappa$ is an important factor in any such NLP-problem. Since the share market depends on so many factors, we claim $\kappa$ as the degree of political turmoil of the country in the present problem.

**6.1.3 Sensitivity analysis in post optimality stage**

We shall analyse the results of the problem in Example 6.1 for different values of $\kappa$ in post optimality stage, shown by the Table 9.

<table>
<thead>
<tr>
<th>$\kappa$</th>
<th>0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$V_\kappa(\tilde{z})$</td>
<td>19413.33</td>
<td>17567.67</td>
<td>15722</td>
<td>13876.33</td>
<td>12030.67</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>$x_1$</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$V_\kappa(\tilde{z})$</td>
<td>10185</td>
<td>8339.33</td>
<td>6493.67</td>
<td>6448</td>
<td>2802.33</td>
</tr>
</tbody>
</table>

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
6.2 Example

Max $\tilde{z} = \kappa \bar{c}_1 x_1 + \bar{c}_2 x_2$

s.t. $2x_1 + 3x_2 \leq 4$
$5x_1 + 4x_2 \leq 15$
$x_1, x_2 \geq 0$

is an NLP-problem where $\bar{c}_1 = \langle ([8, 1, 3]; 0.6), ([8, 3, 4]; 0.2), ([8, 2, 1]; 0.5) \rangle$ and $\bar{c}_2 = \langle ([6, 2, 6]; 0.7), ([6, 4, 3]; 0.4), ([6, 3, 5]; 0.3) \rangle$ are two $G_{SVTrN}$-numbers with a pre-assigned $\kappa = 0.9$.

Rewriting the given constraints by introducing slack variables:

$2x_1 + 3x_2 + x_3 = 4$
$5x_1 + 4x_2 + x_4 = 15$
$x_1, x_2, x_3, x_4 \geq 0$

The $t$-norm and $s$-norm are $p \ast q = \max \{p + q - 1, 0\}$ and $p \diamond q = \min \{p + q, 1\}$, respectively. The first feasible simplex table is as follows:

<table>
<thead>
<tr>
<th>$\bar{c}_j \Rightarrow$</th>
<th>$\bar{c}_1$</th>
<th>$\bar{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B \downarrow$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4 →</td>
</tr>
<tr>
<td>$x_4$</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>$\tilde{z} \Rightarrow$</td>
<td>$\bar{c}_1^{(1)}$</td>
<td>$\bar{c}_2^{(1)}$</td>
<td>$\bar{c}_3^{(1)}$</td>
<td>$\bar{c}_4^{(1)}$</td>
<td></td>
</tr>
</tbody>
</table>

Here $\bar{c}_1^{(1)} = -\bar{c}_1 = \langle ([8, 3, 1]; 0.6), ([8, 4, 3]; 0.2), ([8, 1, 2]; 0.5) \rangle$,
$\bar{c}_2^{(1)} = -\bar{c}_2 = \langle ([6, 6, 2]; 0.7), ([6, 3, 4]; 0.4), ([6, 5, 3]; 0.3) \rangle$,
and $V_\kappa(\bar{c}_1^{(1)}) = V_\kappa(\bar{c}_4^{(1)}) = V_\kappa(0)$.

Then $V_\kappa(\bar{c}_1^{(1)}) = \frac{1}{6}(25.11 \kappa - 43.11)$ and $V_\kappa(\bar{c}_2^{(1)}) = \frac{1}{6}(11.62 \kappa - 31.22)$ by Definition 3.8.1.

Clearly $V_\kappa(-\bar{c}_1) < 0$, $V_\kappa(-\bar{c}_2) < 0$ and $V_\kappa(-\bar{c}_1) - V_\kappa(-\bar{c}_2) > 0$ for $\kappa = 0.9$.

So $x_2$ enters in the basis and as $\min \{4/3, 15/4\} = 4/3$, the leaving variable is $x_3$. The revised table is as:

<table>
<thead>
<tr>
<th>$\bar{c}_j \Rightarrow$</th>
<th>$\bar{c}_1$</th>
<th>$\bar{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B \downarrow$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>2/3</td>
<td>1</td>
<td>1/3</td>
<td>0</td>
<td>4/3 →</td>
</tr>
<tr>
<td>$x_4$</td>
<td>7/3</td>
<td>0</td>
<td>-4/3</td>
<td>1</td>
<td>29/3</td>
</tr>
<tr>
<td>$\tilde{z} \Rightarrow$</td>
<td>$\bar{c}_1^{(2)}$</td>
<td>$\bar{c}_2^{(2)}$</td>
<td>$\bar{c}_3^{(2)}$</td>
<td>$\bar{c}_4^{(2)}$</td>
<td></td>
</tr>
</tbody>
</table>

$T. Bera and N. K. Mahapatra$, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.
where $V_\kappa(\tilde{c}_2^{(2)}) = V_\kappa(\tilde{c}_4^{(2)}) = V_\kappa(\tilde{0})$ and

$$
\tilde{c}_1^{(2)} = \frac{2}{3} \tilde{c}_2 - \tilde{c}_1 = \langle([-4, 13/3], 0.3), ([4, 20/3], 0.6), ([4, 16/3], 0.8)\rangle,
$$

$$
\tilde{c}_3^{(2)} = \frac{1}{3} \tilde{c}_2 = \langle([2, 2/3], 0.7), ([2, 4/3], 0.4), ([2, 1, 5/3], 0.3)\rangle.
$$

Then $V_\kappa(\tilde{c}_1^{(2)}) = \frac{1}{18}(8.62\kappa - 14.92)$ and $V_\kappa(\tilde{c}_3^{(2)}) = \frac{1}{18}(31.22 - 11.62\kappa)$ by Definition 3.8.1.

Clearly, $V_\kappa(\tilde{c}_1^{(2)}) < 0$ but $V_\kappa(\tilde{c}_3^{(2)}) > 0$ for $\kappa = 0.9$. So $x_1$ enters in the basis and as $\min\left\{\frac{4}{3}, \frac{29}{7}\right\} = 2$, the leaving variable is $x_2$. The revised table is:

<table>
<thead>
<tr>
<th>$\tilde{c}_j \Rightarrow$</th>
<th>$\tilde{c}_1$</th>
<th>$\tilde{c}_2$</th>
<th>$\tilde{0}$</th>
<th>$\tilde{0}$</th>
<th>R.H.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_B \downarrow$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>3/2</td>
<td>1/2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>-7/2</td>
<td>-5/2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>$\tilde{z} \Rightarrow$</td>
<td>$\tilde{c}_1^{(3)}$</td>
<td>$\tilde{c}_2^{(3)}$</td>
<td>$\tilde{c}_3^{(3)}$</td>
<td>$\tilde{c}_4^{(3)}$</td>
<td>2$\tilde{c}_1$</td>
</tr>
</tbody>
</table>

where $V_\kappa(\tilde{c}_1^{(3)}) = V_\kappa(\tilde{c}_4^{(3)}) = V_\kappa(\tilde{0})$ and

$$
\tilde{c}_2^{(3)} = \frac{3}{2} \tilde{c}_2 - \tilde{c}_1 = \langle([6, 7.5, 6.5], 0.3), ([6, 7.5, 10], 0.6), ([6, 8, 4.5], 0.8)\rangle,
$$

$$
\tilde{c}_3^{(3)} = \frac{1}{2} \tilde{c}_2 = \langle([4, 0.5, 1.5], 0.6), ([4, 1.5, 2], 0.2), ([4, 1, 0.5], 0.5)\rangle.
$$

Then $V_\kappa(\tilde{c}_2^{(3)}) = \frac{1}{5}(7.46 - 4.31\kappa)$ and $V_\kappa(\tilde{c}_3^{(3)}) = \frac{1}{6}(21.555 - 12.555\kappa)$ by Definition 3.8.1.

Obviously, $V_\kappa(\tilde{c}_2^{(3)}) > 0$ and $V_\kappa(\tilde{c}_3^{(3)}) > 0$ for $\kappa = 0.9$. Hence the optimality arises. The optimal solution is $x_1 = 2, x_2 = 0$ and so Max $\tilde{z} = _R 2\tilde{c}_1$.

### 7 Conclusion

In this paper, the crisp LP-problem has been generalised by considering the coefficients of the objective function as $G_{SVN}$-numbers. This generalised form of crisp LP-problem is called NLP-problem. Then a simplex algorithm has been proposed to solve such NLP-problems. Finally, the newly developed simplex algorithm has been applied to a real life problem. The concept has been illustrated by suitable examples using both $G_{SVTN}$-numbers and $G_{SVTN}$-numbers. In future, the concept of a linear programming problem may be extended in a more generalised way by considering some or all of the parameters as $G_{SVN}$-numbers.

### References


T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.


Received: December 27, 2018. Accepted: March 25, 2019.

T. Bera and N. K. Mahapatra, Generalised single valued neutrosophic number and its application to neutrosophic linear programming.