Hyperfunctions and Superhyperfunctions in Linear Programming: Foundations and Applications

Takaaki Fujita[1], **Maisam Jdid**[2,3,*], **Florentin Smarandache**[4]
[1]Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan
[2]Faculty of Science, Damascus University, Damascus, Syria
[3]Department of Requirements, International University for Science and Technology, Ghabageb, Syrian Arab Republic
[4]University of New Mexico, Mathematics, Physics, and Natural Sciences Division 705 Gurley Ave., Gallup, NM 87301, USA
Emails: Takaaki.fujita060@gmail.com; maissam.jdid66@damascusuniversity.edu.sy; smarand@unm.edu

Abstract

A hyperfunction maps each input to a subset of outputs, generalizing classical functions to represent multi-valued or uncertain outcomes. A superhyperfunction extends this idea further by mapping sets (or sets of sets) to higher-order powerset values, thereby capturing complex hierarchical or layered uncertainties. In this paper, we explore the use of hyperfunctions and superhyperfunctions in linear programming. Specifically, we examine the Linear Objective (Profit/Cost) n-SuperHyperfunction and the Linear Utility n-SuperHyperfunction. We hope these concepts will advance both hyperfunction theory and the study of linear programming under uncertainty.

Keywords: Hyperfunction; Superhyperfunction; Linear Programming; Linear Utility Function; Linear Objective (Profit/Cost) Function

1 Introduction

1.1 Linear Programming

Operations research is a modern applied science capable of mathematically modeling concepts such as efficiency and scarcity in practical decision-making contexts.[6] It employs scientific methods to solve complex problems associated with the management of large-scale systems in factories, institutions, and corporations. A mathematical model is a simplified abstraction of a real-world system and typically consists of an *objective function* along with a set of *constraints*. When both the objective function and the constraints are linear, the resulting model is known as a *linear mathematical model*.[5]

Linear programming is a foundational topic in operations research due to its broad applicability across many fields. Given its importance and the need for more precise modeling under real-world uncertainty, previous studies have reformulated linear programming models using *neutrosophic logic*.[13] Neutrosophic logic introduces a novel perspective in modeling by effectively handling indeterminacy and vagueness in real-world decision systems (cf.[?]).

1.2 Our Contributions: Hyperfunction and n-SuperHyperfunction

In this paper, we investigate the application of *hyperfunctions* and n-*SuperHyperFunctions* within the context of linear programming. In particular, we examine the construction of the Linear Objective (Profit/Cost) n-SuperHyperFunction and the Linear Utility n-SuperHyperFunction. These constructs aim to extend hyperfunction theory and contribute to the study of linear programming models under conditions of uncertainty.

2 Preliminaries and Definitions

This section presents the key concepts and definitions required for the discussions in this paper. Unless otherwise stated, all sets and structures considered here are assumed to be finite.

2.1 Hyperfunction and n-Superhyperfunction

In the context of Hyperstructure and n-Superhyperstructure in functions, the concepts of Hyperfunction and n-Superhyperfunction were introduced by Smarandache in 2016 .[2] Several related notions—such as the superhypergraph—are also known and have been actively explored.[11] In particular, superhyperfunctions have been extensively studied with various theoretical developments and applications.[8] The relevant definitions and theorems are presented below.

Definition 2.1 (Base Set)**.** A *base set* S is the foundational set from which complex structures such as powersets and hyperstructures are derived. It is formally defined as:

$$S = \{x \mid x \text{ is an element within a specified domain}\}.$$

All elements in constructs like $\mathcal{P}(S)$ or $\mathcal{P}_n(S)$ originate from the elements of S.

Definition 2.2 (Powerset)**.** The *powerset* of a set S, denoted $\mathcal{P}(S)$, is the collection of all possible subsets of S, including both the empty set and S itself. Formally, it is expressed as:

$$\mathcal{P}(S) = \{A \mid A \subseteq S\}.$$

Definition 2.3 (n-th Powerset)**.** (cf.Smarandache,[19])

The n-th powerset of a set H, denoted $P_n(H)$, is defined iteratively, starting with the standard powerset. The recursive construction is given by:

$$P_1(H) = P(H), \quad P_{n+1}(H) = P(P_n(H)), \quad \text{for } n \geq 1.$$

Similarly, the n-th non-empty powerset, denoted $P_n^*(H)$, is defined recursively as:

$$P_1^*(H) = P^*(H), \quad P_{n+1}^*(H) = P^*(P_n^*(H)).$$

Here, $P^*(H)$ represents the powerset of H with the empty set removed.

Definition 2.4 (Hyperoperation)**.** (cf.[21]) A *hyperoperation* is a generalization of a binary operation where the result of combining two elements is a set, not a single element. Formally, for a set S, a hyperoperation $\circ$ is defined as:

$$\circ : S \times S \to \mathcal{P}(S),$$

where $\mathcal{P}(S)$ is the powerset of S.

Definition 2.5 (Hyperfunction)**.** [19] A *Hyperfunction* is a function where the domain remains a classical set S, but the codomain is extended to the powerset of S, denoted $\mathcal{P}(S)$. Formally, a Hyperfunction f is defined as:

$$f : S \to \mathcal{P}(S).$$

For any $x \in S$, $f(x) \subseteq S$ is a subset of S. This allows the function to map an element of S to multiple elements, enabling greater flexibility compared to classical functions.

Example 2.6 (University Course Prerequisites as a Hyperfunction)**.** Let

$$S = \{\text{Calculus, LinearAlgebra, DiscreteMath, DataStructures, Algorithms, Databases}\}.$$

Define the hyperfunction

$$f : S \longrightarrow \mathcal{P}(S)$$

by assigning to each course the set of its direct prerequisites:

$$\begin{aligned} f(\text{Calculus}) &= \emptyset, \\ f(\text{LinearAlgebra}) &= \{\text{Calculus}\}, \\ f(\text{DiscreteMath}) &= \{\text{Calculus}\}, \\ f(\text{DataStructures}) &= \{\text{DiscreteMath}\}, \\ f(\text{Algorithms}) &= \{\text{DataStructures, DiscreteMath}\}, \\ f(\text{Databases}) &= \{\text{DataStructures}\}. \end{aligned}$$

Here, for example, Algorithms maps to the set {DataStructures, DiscreteMath}, meaning a student must have completed both Data Structures and Discrete Mathematics before enrolling in Algorithms. This hyperfunction captures the "many-to-many" relationship between courses and their prerequisites, which cannot be represented by an ordinary single-valued function.

Definition 2.7 (SuperHyperOperations)**.** Let H be a non-empty set, and let $P(H)$ be the powerset of H. Define the n-th powerset $P^n(H)$ recursively:

$$P^0(H) = H, \quad P^{k+1}(H) = P(P^k(H)) \text{ for } k \geq 0.$$

A *SuperHyperOperation* of order (m, n) is an m-ary operation:

$$\circ^{(m,n)} : H^m \to P_*^n(H)$$

where $P_*^n(H)$ denotes the n-th powerset of H possibly excluding the empty set (for a classical-type SuperHyperOperation) or including it (for a Neutrosophic-type SuperHyperOperation).

If the codomain is $P_*^n(H)$ without the empty set, we call it a *classical-type (m,n)-SuperHyperOperation*. If the codomain is $P^n(H)$ including the empty set, we call it a *Neutrosophic (m,n)-SuperHyperOperation*.

In either case, these SuperHyperOperations are higher-order generalizations of hyperoperations, capturing multi-level complexity through n-th powerset constructions.

Definition 2.8 (n-Superhyperfunction)**.** (cf.Smarandache,[19]) An *n-Superhyperfunction* generalizes the concept of a Hyperfunction by using the n-th powerset $\mathcal{P}_n(S)$ as the codomain. Formally, for $n \geq 2$, an n-Superhyperfunction f is defined as:

$$f : \mathcal{P}_r(S) \to \mathcal{P}_n(S),$$

where $0 \leq r \leq n$, and $\mathcal{P}_n(S)$ is the n-th powerset of S. This definition allows f to map subsets of S (from $\mathcal{P}_r(S)$) to elements in the n-th powerset $\mathcal{P}_n(S)$.

Example 2.9 (Ingredient–Recipe Mapping as a 2–Superhyperfunction)**.** Let

$$S = \{\text{Eggs, Flour, Sugar, Milk}\}.$$

Then

$$\mathcal{P}_1(S) = \mathcal{P}(S), \quad \mathcal{P}_2(S) = \mathcal{P}\big(\mathcal{P}(S)\big).$$

Define

$$f : \mathcal{P}_1(S) \longrightarrow \mathcal{P}_2(S), \qquad A \mapsto \big\{\, R \subseteq S \mid A \subseteq R \big\}.$$

That is, for each ingredient–set A, $f(A)$ is the set of all "recipes" (subsets of S) that include every ingredient in A. For example,

$$f\big(\{\text{Eggs, Flour}\}\big) = \{\{\text{Eggs, Flour}\}, \{\text{Eggs, Flour, Sugar}\}, \{\text{Eggs, Flour, Milk}\}, \{\text{Eggs, Flour, Sugar, Milk}\}\}.$$

Since each $f(A)$ is a subset of $\mathcal{P}(S)$, we have $f(A) \in \mathcal{P}_2(S)$. Thus f is a 2–Superhyperfunction.

More generally, for any $n \geq 2$, one can define an n–Superhyperfunction

$$f : \mathcal{P}_r(S) \longrightarrow \mathcal{P}_n(S) \quad (0 \leq r \leq n)$$

by

$$f(A) = \big\{ B \subseteq \mathcal{P}_{n-1}(S) \mid A \subseteq B \big\},$$

mapping each r-subset of S to the family of $(n-1)$-subsets that contain it.

2.2 Linear Function

Linear programming optimizes a linear objective function subject to linear equality and inequality constraints, modeling resource allocation and planning problems.[14, 18, 23] Among the functions applicable in linear programming, the most fundamental is the linear function. We present several examples of commonly used linear functions.

Definition 2.10 (Linear Objective (Profit/Cost) Function)**.** (cf.[7]) Let $x = (x_1, \ldots, x_n)^T$ be the decision vector and $c = (c_1, \ldots, c_n)^T$ the coefficient vector. A *linear objective function* is

$$f_{\text{obj}}(x) = c^T x = \sum_{j=1}^{n} c_j \, x_j,$$

which one seeks to maximize (profit) or minimize (cost).

Definition 2.11 (Linear Utility Function)**.** (cf.[12]) In a resource allocation problem, a linear *utility function* might be

$$U(x) = \sum_{j=1}^{n} u_j \, x_j,$$

where each u_j represents the marginal utility per unit of x_j.

3 Result of This Paper

This section presents the main results of this paper.

3.1 Linear Objective (Profit/Cost) Hyperfunction

The definition of the Linear Objective (Profit/Cost) Hyperfunction is provided below.

Definition 3.1 (Linear Objective Hyperfunction)**.** Let $S = \mathrm{R}^n$ be the decision space, and let

$$C \subseteq \mathrm{R}^n$$

be a nonempty compact set of possible coefficient vectors. The *Linear Objective Hyperfunction* is the mapping

$$H_{\text{obj}} : S \longrightarrow \mathcal{P}(\mathrm{R}), \qquad H_{\text{obj}}(x) = \{ c^T x \mid c \in C \},$$

which assigns to each decision vector x the set of all possible objective values $c^T x$ as c varies over C.

Example 3.2 (Furniture Production with Price Uncertainty)**.** Let

$$S = \{(x_1, x_2) \in \mathrm{R}^2 \mid x_1, x_2 \geq 0\}$$

represent weekly production quantities of chairs (x_1) and tables (x_2). Due to market fluctuations, the per-unit profit vector lies in the compact set

$$C = [10,\ 12] \times [8,\ 9] \ \subset \ \mathrm{R}^2 .$$

Then the Linear Objective Hyperfunction

$$H_{\mathrm{obj}}(x) = \{\, c^T x \mid c \in C\} = \big\{\, c_1 x_1 + c_2 x_2 \mid c_1 \in [10, 12],\, c_2 \in [8, 9]\big\}$$

is the interval

$$H_{\mathrm{obj}}(x) = \big[\, 10x_1 + 8x_2,\ 12x_1 + 9x_2\big].$$

The production plan must satisfy resource constraints:

$$\begin{aligned} 2x_1 + 3x_2 &\leq 200, \qquad \text{(labor hours)} \\ 4x_1 + x_2 &\leq 160, \quad \text{(material units)} \\ x_1,\, x_2 &\geq 0. \end{aligned}$$

A robust optimization formulation uses the hyperfunction's extremal values:

$$\underline{Z}(x) = 10x_1 + 8x_2, \qquad \overline{Z}(x) = 12x_1 + 9x_2.$$

Thus the robust–optimistic linear program is

$$\begin{aligned} &\max_{x_1, x_2} \ (\underline{Z}(x),\, \overline{Z}(x)), \\ \text{s.t.} \quad &2x_1 + 3x_2 \leq 200, \\ &4x_1 + x_2 \leq 160, \\ &x_1, x_2 \geq 0. \end{aligned}$$

This example illustrates how H_{obj} models profit uncertainty in linear programming.

Theorem 3.3. *H_{obj} is a Hyperfunction on S.*

Proof. By construction, for each $x \in S$,

$$H_{\mathrm{obj}}(x) = \{\, c^T x \mid c \in C\} \subseteq \mathrm{R}$$

is a nonempty (since $C \neq \emptyset$) and compact subset of R (continuous image of a compact set). Therefore

$$H_{\mathrm{obj}} : S \to \mathcal{P}(\mathrm{R})$$

satisfies the definition of a Hyperfunction, mapping each element of S to a subset of R. □

Theorem 3.4. *The classical Linear Objective Function*

$$f_{\mathrm{obj}}(x) = c_0^T x$$

is recovered as a special case of H_{obj} by choosing $C = \{c_0\}$.

Proof. If $C = \{c_0\}$, then for each $x \in S$,

$$H_{\mathrm{obj}}(x) = \{\, c^T x \mid c \in \{c_0\}\} = \{\, c_0^T x\}.$$

Identifying the singleton set $\{c_0^T x\}$ with its unique element $c_0^T x$ recovers the classical objective value. Hence H_{obj} generalizes the linear objective function. □

3.2 Linear Objective (Profit/Cost) n-SuperHyperfunction

The definition of the Linear Objective (Profit/Cost) n-SuperHyperfunction is provided below.

Definition 3.5 (Linear Objective n-SuperHyperfunction)**.** Let $S = \mathrm{R}^n$ be the decision space and $C \subseteq \mathrm{R}^n$ a nonempty compact set of coefficient vectors. Define for each $x \in S$:

$$H_{\mathrm{obj}}^{(1)}(x) = \{\, c^T x \mid c \in C\} \;\subseteq\; \mathrm{R},$$

and recursively for $k = 2, \ldots, n$:

$$H_{\mathrm{obj}}^{(k)}(x) = \mathcal{P}\big(H_{\mathrm{obj}}^{(k-1)}(x)\big) \setminus \{\emptyset\},$$

the family of all nonempty subsets of $H_{\mathrm{obj}}^{(k-1)}(x)$. Then

$$H_{\mathrm{obj}}^{(n)} : S \;\longrightarrow\; \mathcal{P}_n(\mathrm{R}), \quad x \;\mapsto\; H_{\mathrm{obj}}^{(n)}(x)$$

is called the *Linear Objective n-SuperHyperfunction*.

Example 3.6 (Three-Level Profit Uncertainty in Production Planning)**.** Let

$$S = \{\, x = (x_1, x_2) \in \mathrm{R}^2 \mid x_1, x_2 \geq 0\}$$

be the decision space for producing two products. Suppose profit per unit is uncertain under three scenarios:

$$C = \big\{\, c^{(1)} = (15, 10),\; c^{(2)} = (12, 8),\; c^{(3)} = (10, 6)\big\}.$$

Define the first-level objective values

$$H_{\mathrm{obj}}^{(1)}(x) = \{\, c^{(i)T} x \mid i = 1, 2, 3\} = \{\, 15x_1 + 10x_2,\; 12x_1 + 8x_2,\; 10x_1 + 6x_2\}.$$

Then the second and third levels are

$$H_{\mathrm{obj}}^{(2)}(x) = \mathcal{P}\big(H_{\mathrm{obj}}^{(1)}(x)\big) \setminus \{\emptyset\},$$

$$H_{\mathrm{obj}}^{(3)}(x) = \mathcal{P}\big(H_{\mathrm{obj}}^{(2)}(x)\big) \setminus \{\emptyset\}.$$

Concretely,

$$H_{\mathrm{obj}}^{(3)}(x) = \big\{\{r_i\},\; \{r_i, r_j\},\; \{r_1, r_2, r_3\} \;\big|\; 1 \leq i < j \leq 3\big\},$$

where $r_1 = 15x_1 + 10x_2$, $r_2 = 12x_1 + 8x_2$, $r_3 = 10x_1 + 6x_2$.

Consider the linear programming model

$$\begin{aligned} &\max_{x \in S}\; H_{\mathrm{obj}}^{(3)}(x), \\ \text{s.t.}\quad &x_1 + x_2 \leq 100, \\ &2\,x_1 + 3\,x_2 \leq 120. \end{aligned}$$

Here, maximizing the third-level SuperHyperfunction $H_{\mathrm{obj}}^{(3)}$ captures hierarchical profit uncertainty across all single-scenario, two-scenario, and three-scenario combinations.

Example 3.7 (Two-Product Production Planning with Hierarchical Profit Uncertainty)**.** A factory manufactures two products: chairs (P_1) and tables (P_2). Let

$$x_1, x_2 \;\geq 0$$

be the quantities of chairs and tables produced. There are two market scenarios with per-unit profits

$$c^{(1)} = (10, 8), \quad c^{(2)} = (12, 9).$$

Resource constraints (per week) are:

$$2x_1 + 3x_2 \;\leq\; 240 \quad \text{(labor hours)},$$

$$4x_1 + 2x_2 \;\leq\; 160 \quad \text{(material units)}.$$

First-Level Hyperfunction Define the Linear Objective Hyperfunction

$$H_{\text{obj}}^{(1)}(x) = \big\{\, 10x_1 + 8x_2,\ 12x_1 + 9x_2 \big\},$$

which gives the profit in each scenario.

Second-Level SuperHyperfunction The 2-SuperHyperfunction

$$H_{\text{obj}}^{(2)}(x) = \mathcal{P}\big(H_{\text{obj}}^{(1)}(x)\big) \setminus \{\emptyset\}$$

yields the family of nonempty subsets:

$$\big\{\{10x_1 + 8x_2\},\ \{12x_1 + 9x_2\},\ \{10x_1 + 8x_2,\ 12x_1 + 9x_2\}\big\},$$

capturing single-scenario and combined-scenario profits.

Production Planning Model The set-valued objective can be interpreted via its extremal elements:

$$\underline{Z}(x) = \min H_{\text{obj}}^{(1)}(x) = \min\{10x_1 + 8x_2,\ 12x_1 + 9x_2\},$$

$$\overline{Z}(x) = \max H_{\text{obj}}^{(1)}(x) = \max\{10x_1 + 8x_2,\ 12x_1 + 9x_2\}.$$

A robust–optimistic linear program is

$$\begin{aligned} &\max_{x_1,x_2}\ (\underline{Z}(x),\ \overline{Z}(x)), \\ \text{s.t.}\quad &2x_1 + 3x_2 \ \le\ 240, \\ &4x_1 + 2x_2 \ \le\ 160, \\ &x_1, x_2 \ \ge 0. \end{aligned}$$

Here the interval $[\underline{Z}, \overline{Z}]$ summarizes hierarchical profit uncertainty across both scenarios.

Theorem 3.8. *For each $n \ge 1$, $H_{\text{obj}}^{(n)}$ is an n-SuperHyperfunction.*

Proof. By induction on k. For $k = 1$, $H_{\text{obj}}^{(1)}(x) \subseteq \mathrm{R}$ is nonempty (since $C \neq \emptyset$) and compact (continuous image of compact C), so $H_{\text{obj}}^{(1)} \colon S \to \mathcal{P}(\mathrm{R})$ is a Hyperfunction. Assume $H_{\text{obj}}^{(k-1)}(x)$ is nonempty and compact for all x. Then $\mathcal{P}\big(H_{\text{obj}}^{(k-1)}(x)\big) \setminus \{\emptyset\}$ is a nonempty family of nonempty compact subsets of R. Hence $H_{\text{obj}}^{(k)} : S \to \mathcal{P}_k(\mathrm{R})$ satisfies the definition of an k-SuperHyperfunction. By induction, the result holds for $k = n$. □

Theorem 3.9. *$H_{\text{obj}}^{(n)}$ generalizes the Linear Objective Hyperfunction $H_{\text{obj}}^{(1)}$.*

Proof. By definition, the first-level mapping $H_{\text{obj}}^{(1)}$ is exactly the Linear Objective Hyperfunction. For $n > 1$, $H_{\text{obj}}^{(n)}$ builds on $H_{\text{obj}}^{(1)}$ by taking successive powersets. Thus $H_{\text{obj}}^{(n)}$ reduces to $H_{\text{obj}}^{(1)}$ when restricted to its first level, and extends it through higher-order subset embeddings, showing that $H_{\text{obj}}^{(n)}$ is indeed a proper generalization. □

3.3 Linear Utility HyperFunction

The definition of the Linear Utility HyperFunction is provided below.

Definition 3.10 (Linear Utility HyperFunction)**.** Let $S = \mathrm{R}^n$ be the allocation space, and let

$$U_C \subseteq \mathrm{R}^n$$

be a nonempty compact set of possible marginal-utility vectors. The *Linear Utility HyperFunction* is the mapping

$$H_{\mathrm{util}} : S \longrightarrow \mathcal{P}(\mathrm{R}), \qquad H_{\mathrm{util}}(x) = \{\, u^T x \mid u \in U_C \},$$

which assigns to each allocation x the set of all possible utility values $u^T x$ as u varies over U_C.

Example 3.11 (Resource Allocation with Utility Uncertainty)**.** Consider an organization allocating resources to two activities, x_1 and x_2. The allocation vector is

$$x = (x_1, x_2)^T, \quad x_1, x_2 \geq 0.$$

Due to stakeholder disagreement, the marginal-utility vector lies in

$$U_C = [2,\, 3] \times [1,\, 2] = \{\, u = (u_1, u_2)^T \mid u_1 \in [2,3],\, u_2 \in [1,2] \}.$$

By definition, the Linear Utility HyperFunction is

$$H_{\mathrm{util}}(x) = \{\, u^T x \mid u \in U_C \} = \big\{\, u_1 x_1 + u_2 x_2 \mid u_1 \in [2,3],\, u_2 \in [1,2] \big\} = [\, 2x_1 + 1x_2,\; 3x_1 + 2x_2 \,].$$

Resource constraints are:

$$\begin{aligned} x_1 + 2x_2 &\leq 80, && \text{(labor capacity)} \\ 3x_1 + x_2 &\leq 90, && \text{(material capacity)} \\ x_1,\, x_2 &\geq 0. \end{aligned}$$

A robust-optimistic linear program uses the extremal values of the hyperfunction:

$$\underline{U}(x) = 2x_1 + 1x_2, \qquad \overline{U}(x) = 3x_1 + 2x_2.$$

We then solve

$$\begin{aligned} & \max_{x_1, x_2} \; (\, \underline{U}(x),\, \overline{U}(x) \,), \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 80, \\ & 3x_1 + x_2 \leq 90, \\ & x_1,\, x_2 \geq 0. \end{aligned}$$

This formulation captures both the worst-case and best-case utility outcomes, reflecting hierarchical stakeholder preferences.

Theorem 3.12. *H_{util} is a Hyperfunction on S.*

Proof. Since U_C is nonempty and compact, its continuous image under the map $u \mapsto u^T x$ is nonempty and compact for each fixed $x \in S$. Hence

$$H_{\mathrm{util}}(x) = \{\, u^T x \mid u \in U_C \}$$

is a nonempty compact subset of R. Therefore H_{util} satisfies the definition of a Hyperfunction, mapping each $x \in S$ into $\mathcal{P}(\mathrm{R})$. □

Theorem 3.13. *The classical Linear Utility Function*

$$U(x) = u_0^T x$$

is recovered as a special case of H_{util} by choosing $U_C = \{u_0\}$.

Proof. If $U_C = \{u_0\}$, then for every $x \in S$,

$$H_{\mathrm{util}}(x) = \{\, u^T x \mid u \in \{u_0\} \} = \{\, u_0^T x \}.$$

Identifying the singleton $\{u_0^T x\}$ with its unique element $u_0^T x$ recovers the classical utility value. Thus H_{util} generalizes the linear utility function. □

3.4 Linear Utility n-SuperHyperFunction

The definition of the Linear Utility n-SuperHyperFunction is provided below.

Definition 3.14 (Linear Utility n-SuperHyperfunction)**.** Let $S = \mathrm{R}^m$ be the allocation space and $U_C \subseteq \mathrm{R}^m$ a nonempty compact set of possible marginal-utility vectors. Define for each $x \in S$:

$$H^{(1)}_{\text{util}}(x) = \{\, u^T x \mid u \in U_C \} \;\subseteq\; \mathrm{R},$$

and recursively for $k = 2, \ldots, n$:

$$H^{(k)}_{\text{util}}(x) = \mathcal{P}\big(H^{(k-1)}_{\text{util}}(x)\big) \setminus \{\emptyset\} \;\subseteq\; \mathcal{P}_k(\mathrm{R}),$$

where $\mathcal{P}_k(\mathrm{R})$ denotes the kth powerset of R. Then the mapping

$$H^{(n)}_{\text{util}} : S \;\longrightarrow\; \mathcal{P}_n(\mathrm{R}), \qquad x \;\mapsto\; H^{(n)}_{\text{util}}(x),$$

is called the *Linear Utility n-SuperHyperfunction*.

Example 3.15 (Resource Allocation with Two-Level Utility Uncertainty)**.** Consider allocating resources $x = (x_1, x_2) \geq 0$ to two projects. Due to stakeholder disagreement, their marginal-utility vector lies in the compact set

$$U_C = \{\, u^{(1)} = (2, 1),\; u^{(2)} = (3, 2)\} \subset \mathrm{R}^2\,.$$

First-Level Utility Hyperfunction For each allocation x, the first-level mapping is

$$H^{(1)}_{\text{util}}(x) = \{\, u^T x \mid u \in U_C\} = \{\, 2x_1 + 1x_2,\; 3x_1 + 2x_2\}.$$

Second-Level SuperHyperfunction The 2-SuperHyperfunction then is

$$H^{(2)}_{\text{util}}(x) = \mathcal{P}\big(H^{(1)}_{\text{util}}(x)\big) \setminus \{\emptyset\} = \big\{\{2x_1 + 1x_2\},\; \{3x_1 + 2x_2\},\; \{2x_1 + 1x_2,\, 3x_1 + 2x_2\}\big\}.$$

Linear Programming Model Suppose resources satisfy:

$$\begin{aligned} x_1 + 2x_2 &\leq 100, && \text{(labor capacity)} \\ 4x_1 + x_2 &\leq 80, && \text{(material capacity)} \\ x_1,\, x_2 &\geq 0. \end{aligned}$$

We interpret the 2-level utility set via its extremal values:

$$\underline{U}(x) = 2x_1 + 1x_2, \quad \overline{U}(x) = 3x_1 + 2x_2.$$

A robust–optimistic formulation is

$$\begin{aligned} & \max_{x_1, x_2} \; (\, \underline{U}(x),\, \overline{U}(x)\,), \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 100, \\ & 4x_1 + x_2 \leq 80, \\ & x_1,\, x_2 \geq 0. \end{aligned}$$

This example demonstrates how $H^{(2)}_{\text{util}}$ captures both single-scenario and combined-scenario utility outcomes in a linear program.

Theorem 3.16. *For each integer $n \geq 1$, the mapping $H^{(n)}_{\text{util}}$ is an n-SuperHyperfunction.*

Proof. We proceed by induction on k. For $k = 1$, $H^{(1)}_{\text{util}}(x)$ is the image of the compact set U_C under the continuous map $u \mapsto u^T x$, hence is a nonempty compact subset of R. Thus $H^{(1)}_{\text{util}} : S \to \mathcal{P}(\mathrm{R})$ is a hyperfunction.

Assume $H^{(k-1)}_{\text{util}}(x)$ is a nonempty compact subset of R for all x. Then its powerset minus the empty set,

$$H^{(k)}_{\text{util}}(x) = \mathcal{P}\big(H^{(k-1)}_{\text{util}}(x)\big) \setminus \{\emptyset\},$$

is a nonempty family of nonempty compact subsets of R. Therefore $H^{(k)}_{\text{util}} : S \to \mathcal{P}_k(\mathrm{R})$ satisfies the definition of a k-SuperHyperfunction. By induction, $H^{(n)}_{\text{util}}$ is an n-SuperHyperfunction. □

Theorem 3.17. *The classical Linear Utility Hyperfunction $H^{(1)}_{\text{util}}$ is recovered as the first-level case of $H^{(n)}_{\text{util}}$.*

Proof. By definition, the first-level mapping of $H^{(n)}_{\text{util}}$ is

$$H^{(1)}_{\text{util}}(x) = \{\, u^T x \mid u \in U_C \},$$

which exactly coincides with the Linear Utility Hyperfunction. Hence $H^{(n)}_{\text{util}}$ extends and generalizes $H^{(1)}_{\text{util}}$ through higher-order subset constructions. □

4 Conclusion and Future Tasks

In this paper, we explored the application of *hyperfunctions* and *SuperHyperFunctions* in the context of linear programming. Specifically, we examined the construction and application of the Linear Objective (Profit/Cost) n-SuperHyperFunction and the Linear Utility n-SuperHyperFunction.

As a direction for future research, we plan to extend the function models introduced in this paper by incorporating the frameworks of *Fuzzy Sets*,[17,22] *Hyperfuzzy Sets*,[10,15] Soft Sets,[16] *Intuitionistic Fuzzy Sets*,[1,3] Vague Sets,[9] Hesitant Fuzzy Sets,[20] and *Plithogenic Sets*.[4] These advanced uncertainty-based structures are expected to enrich the expressive and computational capacity of SuperHyperFunction-based linear programming models.

Funding

This study did not receive any financial or external support from organizations or individuals.

Acknowledgments

We extend our sincere gratitude to everyone who provided insights, inspiration, and assistance throughout this research. We particularly thank our readers for their interest and acknowledge the authors of the cited works for laying the foundation that made our study possible. We also appreciate the support from individuals and institutions that provided the resources and infrastructure needed to produce and share this paper. Finally, we are grateful to all those who supported us in various ways during this project.

Data Availability

This research is purely theoretical, involving no data collection or analysis. We encourage future researchers to pursue empirical investigations to further develop and validate the concepts introduced here.

Ethical Approval

As this research is entirely theoretical in nature and does not involve human participants or animal subjects, no ethical approval is required.

Conflicts of Interest

The authors confirm that there are no conflicts of interest related to the research or its publication.

Disclaimer

This work presents theoretical concepts that have not yet undergone practical testing or validation. Future researchers are encouraged to apply and assess these ideas in empirical contexts. While every effort has been made to ensure accuracy and appropriate referencing, unintentional errors or omissions may still exist. Readers are advised to verify referenced materials on their own. The views and conclusions expressed here are the authors' own and do not necessarily reflect those of their affiliated organizations.

References

[1] Muhammad Akram, Bijan Davvaz, and Feng Feng. Intuitionistic fuzzy soft k-algebras. *Mathematics in Computer Science*, 7:353–365, 2013.

[2] Mumtaz Ali, Irfan Deli, and Florentin Smarandache. The theory of neutrosophic cubic sets and their applications in pattern recognition. *Journal of intelligent & fuzzy systems*, 30(4):1957–1963, 2016.

[3] Krassimir T Atanassov and Krassimir T Atanassov. *Intuitionistic fuzzy sets*. Springer, 1999.

[4] Muhammad Azeem, Humera Rashid, Muhammad Kamran Jamil, Selma Gütmen, and Erfan Babaee Tirkolaee. Plithogenic fuzzy graph: A study of fundamental properties and potential applications. *Journal of Dynamics and Games*, 2024.

[5] Temur Chilachava and Nugzar Kereselidze. Continuous linear mathematical model of preventive information warfare. *Sokhumi State University Proceedings, Mathematics and Computer Sciences*, 7(7):113–141, 2009.

[6] Joseph G Ecker, Michael Kupferschmid, et al. *Introduction to operations research*. Wiley New York, 1988.

[7] Shu-Cherng Fang and Guangzhi Li. Solving fuzzy relation equations with a linear objective function. *Fuzzy Sets and systems*, 103(1):107–113, 1999.

[8] Takaaki Fujita. Exploration of graph classes and concepts for superhypergraphs and n-th power mathematical structures. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, 3(4):512.

[9] W-L Gau and Daniel J Buehrer. Vague sets. *IEEE transactions on systems, man, and cybernetics*, 23(2):610–614, 1993.

[10] Jayanta Ghosh and Tapas Kumar Samanta. Hyperfuzzy sets and hyperfuzzy group. *Int. J. Adv. Sci. Technol*, 41:27–37, 2012.

[11] Mohammad Hamidi, Florentin Smarandache, and Mohadeseh Taghinezhad. *Decision Making Based on Valued Fuzzy Superhypergraphs*. Infinite Study, 2023.

[12] Eric Jacquet-Lagrèze, Rachid Meziani, and Roman Slowinski. Molp with an interactive assessment of a piecewise linear utility function. *European Journal of Operational Research*, 31(3):350–357, 1987.

[13] Maissam Jdid and Florentin Smarandache. *Converting Some Zero-One Neutrosophic Nonlinear Programming Problems into Zero-One Neutrosophic Linear Programming Problems*. Infinite Study, 2024.

[14] Mariano Jiménez, Mar Arenas Parra, Amelia Bilbao-Terol, and Maria Victoria Rodríguez. Linear programming with fuzzy parameters: An interactive method resolution. *Eur. J. Oper. Res.*, 177:1599–1609, 2007.

[15] Young Bae Jun, Kul Hur, and Kyoung Ja Lee. Hyperfuzzy subalgebras of bck/bci-algebras. *Annals of Fuzzy Mathematics and Informatics*, 2017.

[16] Pradip Kumar Maji, Ranjit Biswas, and A Ranjan Roy. Soft set theory. *Computers & mathematics with applications*, 45(4-5):555–562, 2003.

[17] TM Nishad, Talal Ali Al-Hawary, and B Mohamed Harif. General fuzzy graphs. *Ratio Mathematica*, 47, 2023.

[18] Tong Shaocheng. Interval number and fuzzy number linear programmings. *Fuzzy sets and systems*, 66(3):301–306, 1994.

[19] Florentin Smarandache. *SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions*. Infinite Study, 2023.

[20] Vicenç Torra and Yasuo Narukawa. On hesitant fuzzy sets and decision. In *2009 IEEE international conference on fuzzy systems*, pages 1378–1382. IEEE, 2009.

[21] Souzana Vougioukli. Helix hyperoperation in teaching research. *Science & Philosophy*, 8(2):157–163, 2020.

[22] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[23] Hans-Jürgen Zimmermann. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1:45–55, 1978.