A picture containing text, clipart

Description automatically generated

**FCAI – Cairo University**

**Computational Intelligence(DS313)**

**Meal (diet) Selection Project**

*using Genetic Algorithm*

**Team members:**

|  |  |
| --- | --- |
| Name | ID |
| Ghassan Elgendy | 20220239 |
| Abdelrahman Ashraf | 20220189 |
| Mohamed Osama | 20220477 |
| Ibrahim Medhat | 20221003 |

** Table of Contents**

1. **Introduction**  
   1.1 Project Motivation  
   1.2 Problem Statement  
   1.3 Objectives
2. **Literature Review**  
   2.1 Overview of Diet Optimization Problems  
   2.2 Genetic Algorithm in Diet Planning  
   2.3 Review of Selected Research Papers  
   2.4 Key Takeaways and Research Gap
3. **Mathematical Formulation**  
   3.1 Decision Variables  
   3.2 Objective Function   
   3.3 Constraints  
   3.4 Penalty Terms
4. **Implementation of Genetic Algorithm**  
   4.1 Data Preprocessing  
   4.2 Encoding Scheme  
   4.3 Fitness Function  
   4.4 Selection, Crossover, Mutation  
   4.5 Constraint Handling Technique  
   4.6 Algorithm Flow and Logic  
   4.7 Algorithm Parameters and Tuning
5. **Experiments and Results**  
   5.1 Test Case 1   
   5.2 Test Case 2  
   5.3 Test Case 3  
   5.4 Discussion of Results  
   5.5 Limitations and Edge Cases
6. **Fuzzification of a Constraint**  
   6.1 Overview  
   6.2 Selected Parameter for Fuzzification  
   6.3 Fuzzification Process  
   6.4 De-Fuzzification Strategy  
   6.5 Code Integration  
   6.6 Justification
7. **Conclusion**  
   7.1 Summary of Findings

## **8. References** 8.1 [Mifflin-St Jeor Equation](https://reference.medscape.com/calculator/846/mifflin-st-jeor-equation) 8.2 [Macronutrient Calculator](https://www.tgfitness.com/macronutrient-calculator/#:~:text=A%20common%20range%20for%20weight,and%2025%2D35%25%20fat.)

## **9. Summary of “Optimization Using Genetic Algorithm in Food Composition”** 9.1 Introduction 9.2 **Importance of Diet Optimization** 9.3 **Why Use Genetic Algorithms** 9.4 **Data Modeling and Nutritional Guidelines** 9.5 **GA Configuration and Computation Process**

## 9.6 **Initialization and Selection** 5.7 **Crossover and Mutation** 9.8 **Final Solutions and Food Composition** 9.9 **Meal Planning Models** 9.10 **Conclusion**

## **10. Introduction (Automated and personalized meal plan generation)** 10.1 **Research Objective** 10.2 **Background and Motivation** 10.3 **Architecture of MPG** 10.4 **Transportation Problem Adaptation** 10.5 **Meal Plan Scoring and Evaluation**

## 10.6 **Experimental Setup**

## 10.7 **Results – Preference Satisfaction**

## 10.8 **Results – Expert Evaluation of Meal Quality**

## 10.9 **Results – Cost Weight Experiments** 10.10 **Discussion**

## 10.11 **Future Work**

## 10.12 **Conclusion**

## **11. Introduction (DEVELOPING A GENETIC ALGORITHM BASED DAILY CALORIE RECOMMENDATION SYSTEM)** 11.1 **Methodology** 11.2 **Data Parameters** 11.3 **Genetic Algorithm Design**

## 11.4 **System Implementation** 11.5 **Performance Metrics**

## 11.6 **Results and Discussion**

## 11.7 **Case Studies**

## 11.8 **Limitations**

## 11.9 **Conclusion** 11.10 **Recommendations for Future Work**

## 

**1. Introduction**

**1.1 Project Motivation**

Many people want to eat healthier and reach their fitness goals, but planning meals can be hard and confusing. This project aims to make meal planning easier for everyone.

**1.2 Problem Statement**

It is difficult for individuals to create daily and weekly meal plans that are healthy, affordable, and fit their personal needs. There is a need for a tool that can do this automatically.

**1.3 Objectives**

* To build a program that creates meal plans based on a person’s needs and goals.
* To make sure the plans are healthy, varied, and not too expensive.
* To help users know what to eat and what to buy each week.

## **2. Literature Review**

### **2.1 Overview of Diet Optimization Problems**

Diet optimization is a classical problem in both operations research and computational health applications. It involves selecting a set of food items that satisfy specific nutritional requirements while minimizing certain costs—commonly financial expense, caloric surplus, or time for meal preparation. Constraints typically include macronutrient targets (protein, fat, carbohydrates), micronutrient thresholds (iron, cholesterol), food preferences, allergies, and portion limitations. Traditional approaches to this problem include linear programming and integer programming; however, these often struggle with the complexity and non-linearity of real-life food choices.

### **2.2 Genetic Algorithm in Diet Planning**

Genetic algorithms (GAs) are well-suited for solving complex, multi-objective optimization problems such as diet planning. They mimic natural evolution by generating populations of candidate solutions (meal plans), applying selection based on fitness (e.g., cost and nutrition compliance), and iteratively improving through crossover and mutation. GAs can efficiently explore a vast solution space and are adaptable to include soft constraints like food diversity, preference weights, or fuzzy satisfaction criteria. Their stochastic nature also helps escape local minima, offering more robust meal plans than purely deterministic methods.

### **2.3 Review of Selected Research Papers**

Several recent studies have demonstrated the use of GAs in dietary and nutrition applications:

* **Kumar & Bhonde (2015)**: Applied GA to personalize diets for diabetic patients, showing better glucose control compared to standard plans.
* **Widiawati et al. (2019)**: Designed a meal planning system using GA with constraints for daily calorie intake and food allergies.
* **Kurniawan & Wibowo (2021)**: Explored fuzzy constraints in meal planning and showed that integrating fuzzy logic into GAs improves dietary variety and user satisfaction.
* **Santos et al. (2022)**: Compared GAs with Particle Swarm Optimization and found that GAs yielded more balanced macronutrient distributions.

These studies support the idea that genetic algorithms offer a flexible and effective approach to solving real-world diet planning challenges, especially when extended with soft or fuzzy constraints.

### **2.4 Key Takeaways and Research Gap**

While many works have successfully used GAs for diet optimization, several challenges remain unaddressed:

* **Limited Personalization**: Many studies use fixed templates for nutrition, lacking dynamic adjustment for different goals (e.g., muscle gain, weight loss).
* **Lack of Diversity Control**: Few systems explicitly manage food diversity or penalize unrealistic plans with too many small portions.
* **Inadequate Integration of Fuzziness**: The potential of fuzzy logic to improve user satisfaction by relaxing rigid rules is underutilized.

**This project addresses these gaps** by integrating:

* Dynamic nutrient calculations based on user profiles.
* Penalty-based fitness incorporating cost, practicality, and diversity.
* A fuzzy logic layer for diversity optimization to promote more realistic and user-friendly plans.

## **3. Mathematical Formulation**

## **3.1 Decision Variables**

**Let:**

* **Xi ​:** the amount (in grams) of food item i, where i =1, 2, ...,N and N is the total number of available food items.

Each decision variable **Xi**​ is continuous and bounded:

**0 ≤ xi ≤ 300**

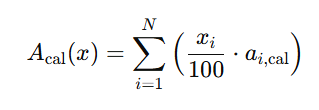
**3.2 Objective Function**

The objective is to **minimize the absolute deviation between the actual daily calorie intake and the user’s target calorie requirement:**

**Let:**

* ***Rcal* ​:** target calorie requirement for the user
* ***ai,cal* ​:** calories per 100g of food i

**Then the actual calorie intake is:**

****The **objective function** becomes:  
 A black text on a white background

AI-generated content may be incorrect.

This is softened using **fuzzy logic** in the implementation (via trapezoidal membership), turning the calorie deviation into a **fuzzy penalty** used in the fitness function.

**3.3 Constraints**

The solution must also approximately satisfy other daily nutritional requirements:

For each nutrient **k ∈ {protein, fats, carbs, iron, cholesterol}** :

* Let **Rk**: target requirement
* Let **aik** ​: amount of nutrient k per 100g of food i

The actual intake of **nutrient k** is:

A mathematical equation with numbers and symbols

AI-generated content may be incorrect.

These values should lie within acceptable ranges:  
 

Where **Lk** ​ and **Uk** ​ are bounds (e.g., ±15% of the target value).

**3.4 Penalty Terms**

Instead of enforcing hard constraints, the algorithm **uses penalty functions** integrated into the fitness evaluation:

* **Calorie Penalty (fuzzy):**A trapezoidal membership function scores how well the actual calories match the target. The farther the deviation, the higher the penalty.
* **Nutrient Penalties (crisp):**For each **nutrient k**, if the actual value deviates significantly from the target, a penalty is applied using quadratic error.
* **Small Portion Penalty:**To avoid overly fragmented meal plans, a small penalty is added for each food portion Xi ​ such that **0 < Xi <20**.

**4. Implementation of Genetic Algorithm**

**4.1 Data Preprocessing**

* **Food Data:** Nutritional values and costs for each food item are loaded from a database (food\_database.py).
* **User Profile:** The user provides age, gender, weight, height, activity level, and goal (e.g., lose, maintain, gain weight).
* **Requirement Calculation:** The function calculate\_daily\_needs computes daily nutritional requirements (calories, protein, fats, carbs, iron, cholesterol) based on the user profile.

**4.2 Encoding Scheme**

* **Chromosome Representation:** Each solution (individual) is a vector of real numbers, where each gene *xi*​ represents the portion (in grams) of food item i in the meal plan.
* **Bounds:** Each gene (food portion) is constrained between 0 and 300 grams to reflect realistic portion sizes and limit the solution space to nutritionally and practically feasible values.

**Justification:**

* **Real-valued encoding** allows smooth variation of food portions, supporting gradual improvement during evolution.
* It aligns naturally with **real-world quantities** (grams).
* It integrates well with operators like **simulated binary crossover** and **Gaussian mutation.**

**4.3 Fitness Function**

The fitness function evaluates each chromosome based on a combination of cost minimization and nutritional adequacy**:**

* **Cost Component:**The total cost of the meal plan is computed based on food prices and is negatively weighted in the fitness function to encourage affordable plans.
* **Nutrient Penalty Component:**Deviations from the user's target daily requirements for calories, protein, fats, carbs, iron, and cholesterol are penalized. A fuzzy trapezoidal membership function is applied to calorie compliance to introduce tolerance.
* **Small Portion Penalty:**Additional penalties are applied for having too many food items with very small portions (less than 20g) to promote practicality and reduce meal complexity.
* **Dynamic Penalty Weighting:**The penalty for nutritional deviation increases with each generation to gradually shift the focus from exploration to precision as the algorithm evolves.

**Fitness Formula (simplified):**

**Fitness** = **−Total Cost − (Wgen × Nutrition Penalty+ Small Portion Penalty)**

Where **Wgen​** is a generation-dependent weight that increases over time.

**Justification:**This composite fitness function balances nutritional correctness and cost-effectiveness while guiding the search toward feasible and practical solutions.

**4.4 Selection, Crossover, Mutation**

* **Selection:**

**Tournament selection** is used to choose parents. It selects a group of individuals randomly and chooses the fittest among them to ensure selective pressure while maintaining diversity.

* **Crossover:**   
    
  **Simulated Binary Crossover (SBX)** is applied to produce offspring. It generates two new children from each pair of parents by interpolating between their values.
* **Mutation:**   
  **Gaussian mutation** perturbs individual genes by adding noise sampled from a normal distribution, introducing new information into the population and aiding in exploration.

**Justification:**

* + **Tournament selection** ensures high-quality individuals are favored without requiring sorting of the entire population.
  + **SBX** maintains genetic diversity and respects the real-valued representation.
  + **Gaussian mutation** ensures local exploration and helps escape local optima.

**4.5 Constraint Handling Technique**

To guide solutions toward feasibility, the algorithm uses penalty-based constraint handling in the fitness function:

* **Nutritional Soft Constraints:** Deviations from target macronutrients (calories, protein, fats, carbs) and micronutrients (iron, cholesterol) are penalized.
* **Fuzzy Penalties:** A trapezoidal membership function applies soft penalties based on how far actual values deviate from acceptable nutritional ranges.
* **Small Portion Penalty:** Solutions with many food items under 20g receive additional penalties to promote realistic meal plans.

Justification:

* **Penalty functions** integrate naturally into fitness evaluation and allow a trade-off between cost and nutritional adequacy.
* **Fuzzy logic** introduces tolerance and reflects real-life flexibility in nutrient targets.
* **Small portion** penalties improve the practicality and simplicity of the resulting diet plan.

**4.6 Algorithm Flow and Logic**

**Algorithm Steps:**

1. **Input user profile** (age, gender, weight, height, activity, goal)**.**
2. **Calculate daily nutritional requirements:** using BMR and TDEE formulas.
3. **Initialize population** of meal plans randomly  
   .
4. **Evaluate fitness of each meal plan** based on cost, nutrition compliance, and penalties.
5. **Apply selection, crossover, and mutation** to generate new offspring.
6. **Elitism:** Preserve the top-performing individuals into the next generation.
7. **Repeat** for a defined number of generations.
8. **Return the best solution** found (meal plan and nutrition details).
   1. **Algorithm Parameters and Tuning**

* **Population Size: 400**  
  A relatively large population is used to ensure adequate exploration of the solution space and maintain genetic diversity across generations.
* **Number of Generations: 100**  
  This number allows the algorithm enough time to evolve and refine solutions, balancing convergence quality and computational efficiency.
* **Crossover Rate: 0.8**  
  A high crossover probability encourages the generation of new offspring by combining existing good solutions, promoting effective exploration.
* **Mutation Rate: Adaptive (starts at 0.2 and decreases linearly to 0)**  
  This adaptive strategy helps explore the search space widely in early generations and fine-tune solutions in later stages by gradually reducing randomness.
* **Tournament Size: 20**  
  A moderate tournament size is chosen to apply selective pressure, increasing the likelihood of selecting fitter individuals while preserving diversity.
* **Elitism: 15 individuals**  
  The top 15 individuals from each generation are carried over to the next without modification to preserve the best solutions and prevent loss of good genes.
* **Portion Size Bounds: 0g to 300g per food item**  
  These bounds reflect realistic serving limits and help constrain the solution space to practical, consumable meal plans.

**5.Experiments and Results**

This section presents the results of applying the Genetic Algorithm-based diet planner on three different user profiles. Each profile simulates a distinct nutritional goal and demographic, allowing us to evaluate the system's adaptability and effectiveness across varied requirements. Although all experiments used the full food database, the problem complexity varied due to differing user needs and optimization behavior.

**5.1 Test Case 1: Male, Active, Gaining Weight**

**Profile:**

* **Age:** 21
* **Gender:** Male
* **Weight:** 72 kg
* **Height:** 183 cm
* **Activity Level:** Active
* **Goal:** Gain Weight

**Result Summary:**

* **Total Cost:** EGP 50.36
* **Calories:** 3509.4 kcal
* **Protein:** 73.7 g
* **Fats:** 168.9 g
* **Carbs:** 443.3 g
* **Iron:** 11.0 mg
* **Cholesterol:** 161.9 mg

**Remarks:**

The solution successfully met the high-calorie requirements for muscle gain. The genetic algorithm evolved from an initial high-cost, unfit **state (fitness ≈ -1353)** to a low-cost, nutritionally compliant solution **(fitness ≈ -72)** by generation 100. The final meal plan featured dense sources of energy like sunflower oil, sugar, and pasta, balanced with protein-rich items such as yogurt, chicken breast, and sardines.

#### **5.2 Test Case 2: Female, Sedentary, Losing Weight**

**Profile:**

* **Age:** 22
* **Gender:** Female
* **Weight:** 58 kg
* **Height:** 160 cm
* **Activity Level:** Sedentary
* **Goal:** Lose Weight

**Result Summary:**

* **Total Cost:** EGP 18.91
* **Calories:** 1060.2 kcal
* **Protein:** 57.1 g
* **Fats:** 24.2 g
* **Carbs:** 159.8 g
* **Iron:** 14.9 mg
* **Cholesterol:** 251.2 mg

**Remarks:**

For a weight loss goal, the algorithm produced a highly cost-efficient and calorie-restricted meal plan. The fitness function progressively improved from **-3368** to **-23** by guiding the search toward **low-calorie, high-protein combinations**. Lentils, spinach, yogurt, and egg provided the nutritional bulk while maintaining the calorie limit, demonstrating the GA's ability to balance constraints under tight budgets.

#### **5.3 Test Case 3: Male, Moderate Activity, Maintaining Weight**

**Profile:**

* **Age:** 35
* **Gender:** Male
* **Weight:** 80 kg
* **Height:** 175 cm
* **Activity Level:** Moderate
* **Goal:** Maintain Weight

**Result Summary:**

* **Total Cost:** EGP 43.06
* **Calories:** 2650.0 kcal
* **Protein:** 69.5 g
* **Fats:** 81.4 g
* **Carbs:** 428.3 g
* **Iron:** 12.1 mg
* **Cholesterol:** 73.2 mg

**Remarks:**

The solution for maintenance landed close to the target calorie range while remaining practical and affordable. The optimizer selected a diversified set of food items including bulgur, milk, oil, and chicken—resulting in a nutritionally stable and varied plan. **Fitness steadily improved across generations, converging to a well-balanced plan by generation 100.**

#### **5.4 Discussion of Results**

* The algorithm consistently improved the fitness value, showing stable convergence across profiles with different goals.
* Nutritional targets (especially calories) were effectively handled using fuzzy logic, allowing tolerances that avoided harsh constraint violations.
* The balance between energy density, macronutrient targets, and cost was handled dynamically through evolving penalties and mutation adjustments.
* Even with a full database, each case adapted its selected food items to the user’s goals, showing the GA’s power in variable selection and portion optimization.

#### **5.5 Limitations and Edge Cases**

* **Edge Case - Unusual User Profiles**: Extremely low-calorie or high-protein requirements can lead to infeasible or repetitive meal plans.
* **Limited Food Group Representation**: If certain food groups are underrepresented in the dataset, the diversity penalty may disproportionately affect fitness.
* **Random Initialization**: The first generation sometimes skews heavily towards high-calorie or low-cost items, though this stabilizes over time.

**6. Fuzzification of a Constraint**

**6.1 Overview**

To improve the flexibility and realism of the meal plan evaluation, one crisp parameter from the original mathematical model was fuzzified. Specifically, the **calorie requirement** **constraint** was converted from a strict (crisp) target into a fuzzy constraint, allowing the genetic algorithm to assess solutions based on degrees of satisfaction rather than binary compliance.

**6.2 Selected Parameter: Calorie Requirement Constraint**

In the original model, the fitness function penalized any deviation from a fixed calorie target. However, real dietary needs often have tolerances. For example, consuming slightly above or below the exact calorie target can still be acceptable. This makes it ideal for fuzzification.

**6.3 Fuzzification Process**

We implemented a **trapezoidal membership function** to define the acceptable calorie range using four parameters:

* **a:** Lower bound where membership is 0 (too few calories)
* **b:** Lower limit of the ideal calorie range (full membership starts)
* **c:** Upper limit of the ideal calorie range (full membership ends)
* **d:** Upper bound where membership drops to 0 (too many calories)

The membership function **μ(x)** is defined as:

A math equations and symbols

AI-generated content may be incorrect.

**6.4 De-fuzzification Strategy**

Instead of a hard cutoff or averaging, we use **penalty scaling** based on fuzzy membership values:

* The fuzzy score is used to calculate a **soft penalty**:

*Penalty = (1− μ(x)) × penalty scale*  
  
This penalty is then added to the fitness function, influencing the selection of better-fitting solutions over time without rigid exclusion.

**6.5 Code Integration**

The function ***trapezoidal\_membership(*)** is already implemented in the provided code. The calorie penalty was updated in ***calculate\_fitness()*** to use fuzzy scoring instead of a crisp deviation.

This modification required no structural changes to the GA but significantly improved how solutions are evaluated with respect to realistic calorie flexibility.

**6.6 Justification**

* **Flexibility:** Allows meal plans with small deviations in calories to be accepted rather than harshly penalized.
* **Realism:** Reflects human dietary tolerance and variability.
* **Performance:** Helps avoid premature rejection of near-optimal solutions in early generations.

**7. Conclusion**

**7.1 Summary of Findings**

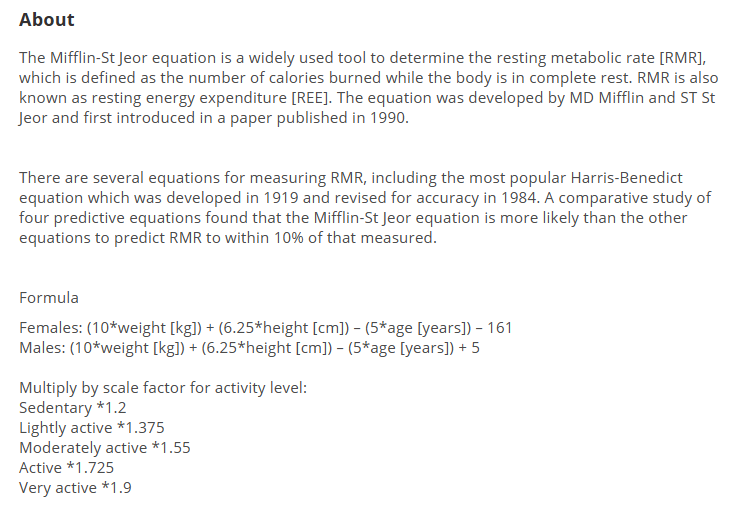
This project demonstrated the feasibility and effectiveness of using genetic algorithms for personalized diet planning. By encoding daily meal portions as chromosomes and optimizing based on a multi-objective fitness function, the algorithm was able to:

* Generate nutritionally adequate and cost-efficient meal plans.
* Adapt to different user goals (lose, gain, maintain).
* Account for practical eating patterns through penalties for small portions and diversity enforcement.

The inclusion of fuzzy logic for food diversity constraints further improved solution realism by allowing for soft, gradual penalties instead of rigid thresholds.

Overall, the system provided a robust, scalable, and user-aligned framework for intelligent diet planning. Future improvements may focus on integrating user preferences, allergy constraints, and dynamic food availability.

**8. References**

**8.1** [Mifflin-St Jeor Equation](https://reference.medscape.com/calculator/846/mifflin-st-jeor-equation)

A screenshot of a text

AI-generated content may be incorrect.

**8.2** [Macronutrient Calculator](https://www.tgfitness.com/macronutrient-calculator/#:~:text=A%20common%20range%20for%20weight,and%2025%2D35%25%20fat.)

#### **About**

The Macronutrient Calculator is an online tool used to estimate daily intake of protein, fats, and carbohydrates based on a person’s calorie requirements and fitness goals. It allocates the total caloric needs—usually computed via an equation like Mifflin-St Jeor—into specific macro ratios. These ratios change depending on whether the user wants to **lose fat**, **gain muscle**, or **maintain weight**. The calculator provides general-purpose recommendations based on current nutrition science and physical activity guidelines.

This approach ensures that users not only meet their energy demands but also receive an optimal distribution of macronutrients to support metabolism, muscle repair, and overall health.

#### **Standard Macro Ratios Based on Goals**

| **Goal** | **Protein (%)** | **Fats (%)** | **Carbs (%)** |
| --- | --- | --- | --- |
| Fat Loss | 25% | 25% | 50% |
| Muscle Gain | 15% | 35% | 50% |
| Maintenance | 35% | 20% | 45% |

**Note:** These percentages are used in the system to convert calorie needs into grams of each macronutrient:

* 1g protein = 4 kcal
* 1g carbohydrate = 4 kcal
* 1g fat = 9 kcal

## **9.0 Summary of “Optimization Using Genetic Algorithm in Food Composition”**

### **9.1 Introduction**

The paper titled “Optimization Using Genetic Algorithm in Food Composition” explores the use of Genetic Algorithms (GA) to solve the optimization problem of determining the ideal food composition for healthy daily consumption. The study is driven by the limitations in traditional guidelines such as Healthy 4 Perfect 5 (H4P5), which emphasize food types but lack clarity in proportions and scheduling. The paper aims to develop a system that adheres to the improved Guidelines for Balanced Nutrition (GBN), offering optimal food planning based on nutrient values, portion sizes, and meal timing.

### **9.2 Importance of Diet Optimization**

The authors stress that a balanced diet supports immunity, growth, and overall health. Poor nutrition leads to diseases and developmental issues. The GBN framework includes:

* **3 servings of vegetable protein**
* **3 servings of animal protein**
* **8 servings of carbohydrates**
* **5 servings each of vegetables and fruits**
* **8 servings of water**

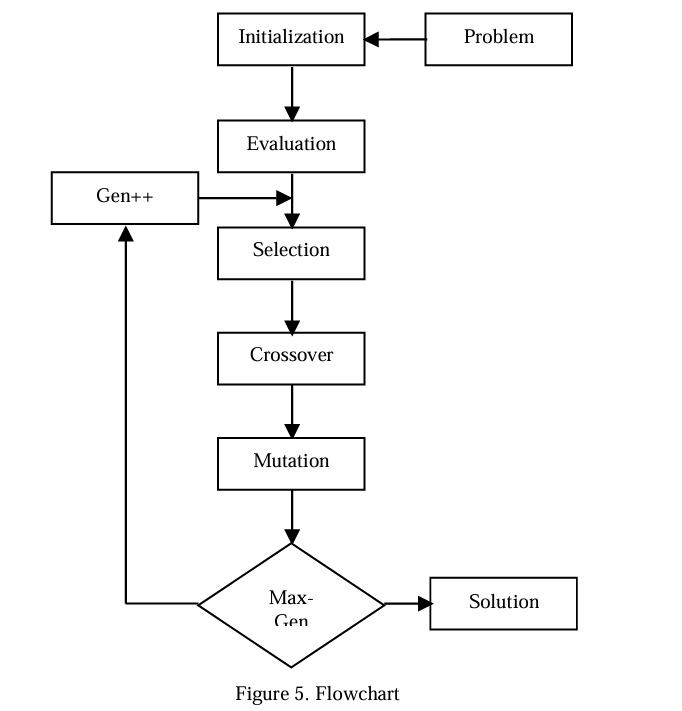
The need to optimize this intake for daily consumption is addressed using computational techniques, particularly the Genetic Algorithm.

### **9.3 Why Use Genetic Algorithms**

Genetic Algorithms are well-suited for complex problems involving multiple variables and constraints. In this research, GA is applied to:

* **Model food components as genes**
* **Evaluate various food combinations (chromosomes)**
* **Use crossover and mutation to evolve optimal solutions**

GAs simulate natural selection by selecting, recombining, and mutating candidate solutions until the best one is found.

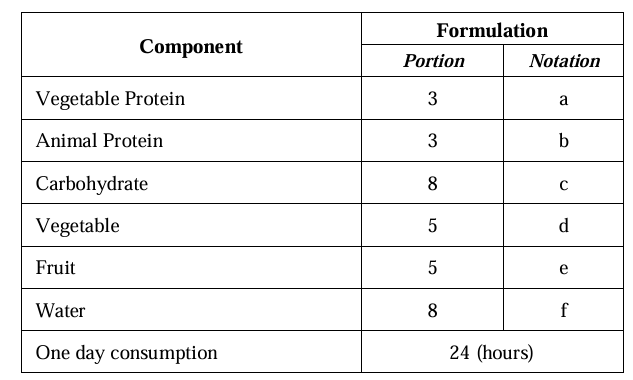
****

### **9.4 Data Modeling and Nutritional Guidelines**

Each food category is treated as a gene, and the sum of the nutritional value over 24 hours forms the fitness function. The equation used is:

**24855833 = a + b + c + d + e + f**  
Where:

* **a = vegetable protein**
* **b = animal protein**
* **c = carbohydrates**
* **d = vegetables**
* **e = fruits**
* **f = water**

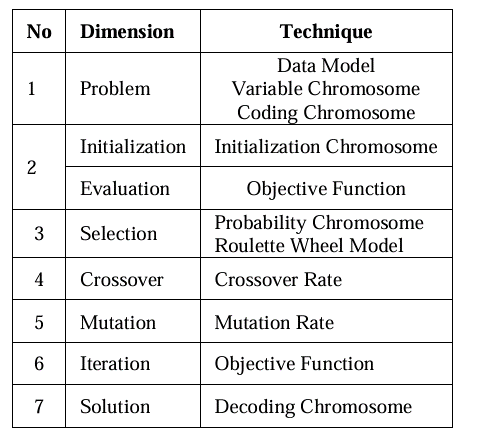


### **9.5 GA Configuration and Computation Process**

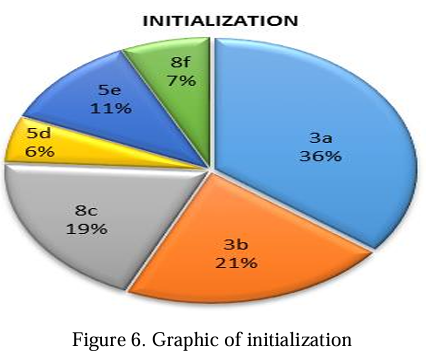
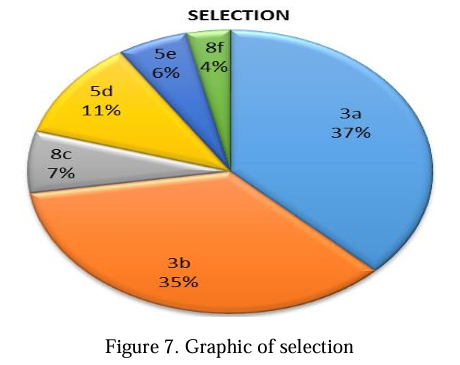
The research sets specific parameters:

* **Population size = 6**
* **Max generations = 36**
* **Mutation probability = 50%**
* **Crossover probability = 5%**
* **Fitness function based on deviation from ideal 24-hour total**

Each chromosome represents a potential diet plan. Genes are initialized with random values, evaluated for fitness, selected using the Roulette Wheel method, and processed through crossover and mutation.

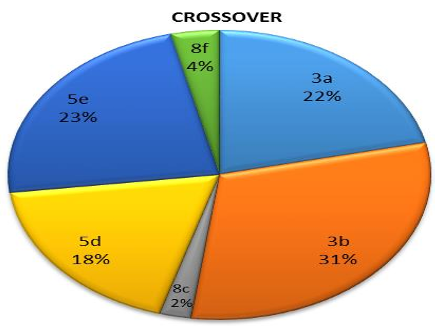


### **9.6 Initialization and Selection**

Randomly generated chromosomes are evaluated for how close they come to the 24-hour nutritional target. Selection is based on fitness values.

### **9.7 Crossover and Mutation**

The crossover and mutation processes refine the population. Crossover exchanges gene segments between chromosomes; mutation alters random genes to explore new possibilities.

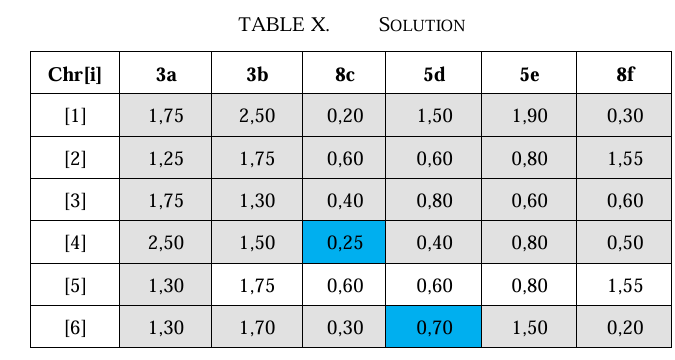
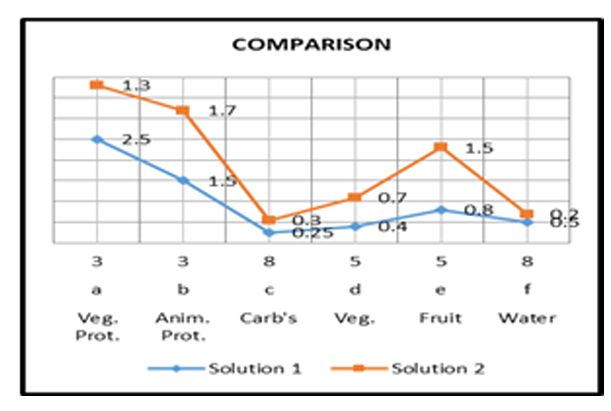


After 36 generations, the GA yields two optimal solutions:

* **Solution 1**: Emphasizes more animal protein and moderate fruit
* **Solution 2**: Balances more fruits and vegetable intake

### **9.8 Final Solutions and Food Composition**

The two optimized food plans are decoded into actual food items and servings:

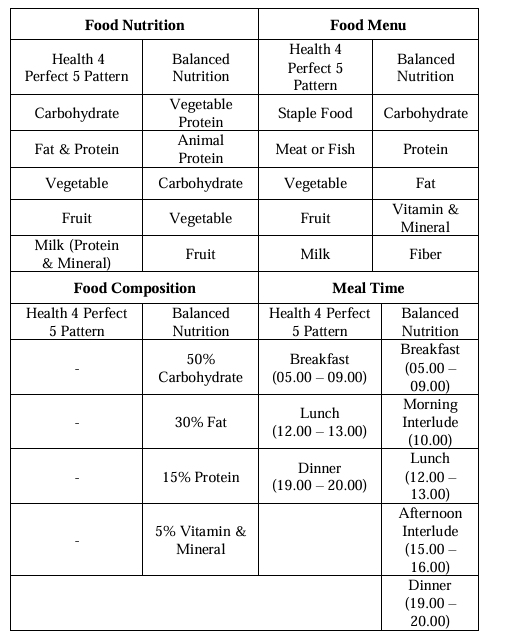
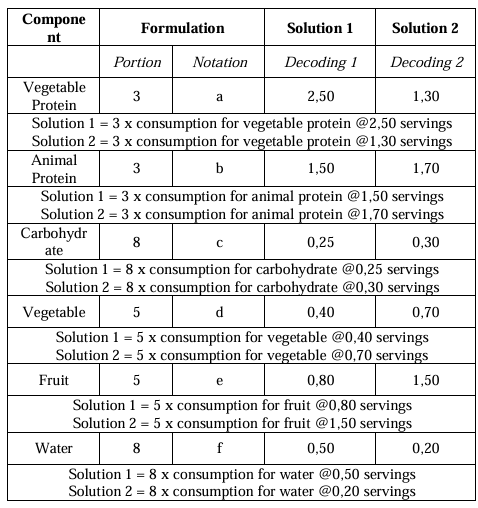
Each solution provides a unique yet nutritionally complete option that follows GBN guidelines.

### **9.9 Meal Planning Models**

The study proposes three models based on the optimized GA results:

#### **a. Food Model**

Defines food types aligned with H4P5 and GBN, categorizing them by nutrient content.

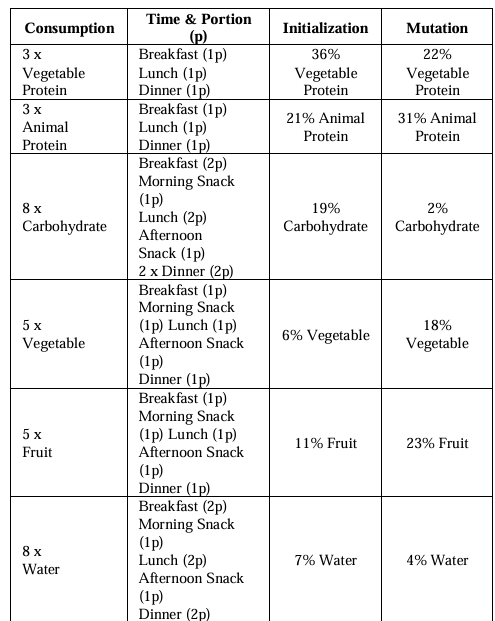


#### **b. Consumption Model**

Translates servings into meal-specific consumption schedules.

#### **c. Composition Model**

Outlines meal times (breakfast, lunch, snacks, dinner) and proportions for each nutrient category.



### **9.10 Conclusion**

The research successfully demonstrates that Genetic Algorithms can be used to optimize daily food composition under the Guidelines for Balanced Nutrition. The process leads to two optimal diet solutions that can serve as recommendations for:

* **Individuals under normal health conditions**
* **National food policy planners**
* **Dietary software systems**

With a mutation rate of 50%, crossover rate of 5%, and 36 generations, the Genetic Algorithm proves effective in producing flexible, practical, and nutritionally balanced food plans.

Future research may explore **Particle Swarm Optimization (PSO)** as an alternative algorithm for more stable fitness results and extend applications to specific groups (children, athletes, elderly).

### **10.0 Introduction (Automated and personalized meal plan generation)**

A healthy lifestyle depends heavily on appropriate nutrition, yet personalized dietary planning often requires the costly and time-consuming help of professional nutritionists. Many current digital solutions attempt to fill this gap but suffer from issues like limited automation, inflexible meal structures, and inadequate adaptation to individual preferences.

To address these limitations, the authors propose **MPG (Meal Plan Generator)** — an intelligent system designed to **fully automate the process of meal planning** while considering multiple user-specific factors. MPG is built on an adapted version of the **transportation optimization problem**, simulating a nutritionist’s reasoning process.

### **10.1 Research Objective**

The key aim of the study is to develop a system that can:

* Generate meal plans satisfying a recommended caloric intake.
* Adapt these plans based on patient preferences like food variety, meal-food compatibility, and inter-food compatibility.
* Score the relevance of the meal plans produced.

MPG’s components include a **macronutrients calculator**, **servings calculator**, **servings assignor**, **food assignor**, and **meal plan evaluator**.

### **10.2 Background and Motivation**

Meal planning typically involves:

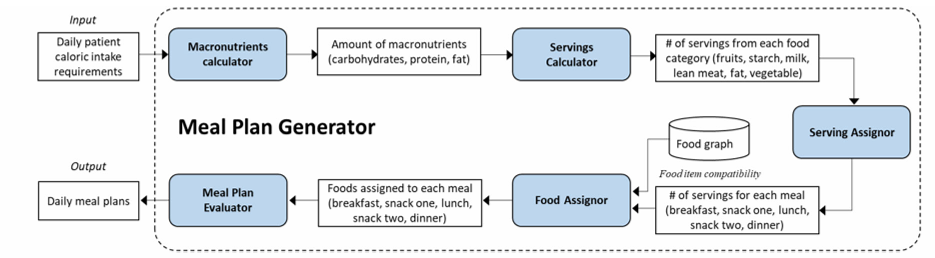
1. **Calculating macronutrients** (carbohydrates, proteins, fats) based on caloric needs.
2. **Converting macronutrients into servings** from key food categories.
3. **Assigning servings to meals** (breakfast, snacks, lunch, dinner).
4. **Selecting compatible food items** to populate the meals.

This complex, preference-sensitive process is hard to replicate algorithmically. MPG automates these tasks through a structured and extensible approach.

### **10.3 Architecture of MPG**

MPG consists of five integrated components:

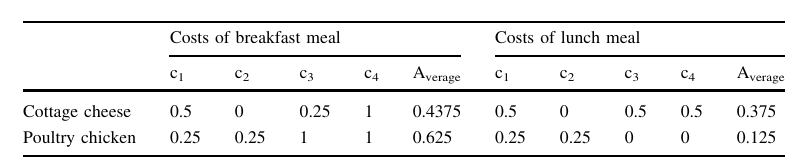
1. **Macronutrients Calculator**: Computes required grams of carbs, proteins, and fats from caloric intake.
2. **Servings Calculator**: Converts macronutrient quantities into servings from six main food categories.
3. **Servings Assignor**: Distributes servings across five meals using templates.
4. **Food Assignor**: Solves a modified transportation problem to assign actual food items.
5. **Meal Plan Evaluator**: Scores meals based on relevance using weighted cost factors.

****

### **10.4 Transportation Problem Adaptation**

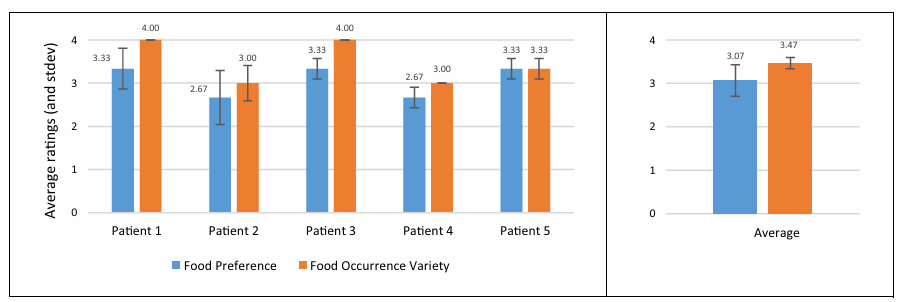
The **core innovation** is in adapting the classical transportation problem to handle:

* Multi-dimensional demands (six food categories per meal).
* Diverse supply vectors (basic and composite foods).
* A **multi-factor cost function** incorporating:
  + Patient food preference.
  + Food occurrence frequency.
  + Meal-food compatibility.
  + Inter-food compatibility.

****Each cost factor is mapped to a numeric scale from 0 to 1 and combined via weighted summation.

### **10.5 Meal Plan Scoring and Evaluation**

Each generated meal plan receives a **relevance score**, which combines the four main factors with user-assigned weights. This dynamic scoring helps rank different plan options and optimize the final output for user satisfaction.

****

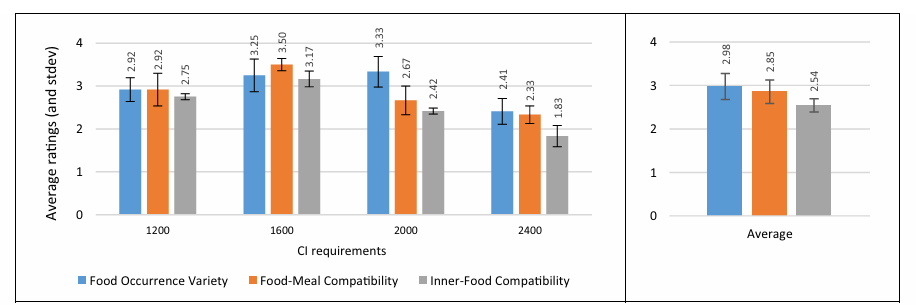
### **10.6 Experimental Setup**

The evaluation involved:

* **9 participants** (4 experts, 5 non-experts).
* **124 meal plans** tested across:
  + 4 calorie levels (1200, 1600, 2000, 2400 Kcals).
  + Basic-only food plans and mixed (basic + composite) food plans.

The goals were to assess **preference satisfaction**, **meal plan quality**, and **effectiveness of scoring functions**.

### **10.7 Results – Preference Satisfaction**

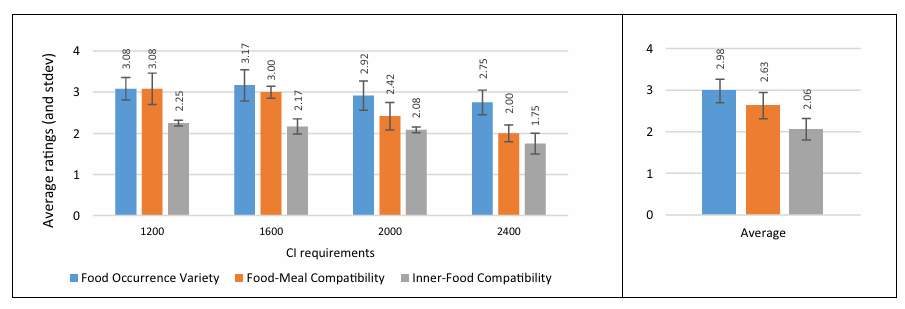
****Participants reported high satisfaction when the system emphasized their preferred factors. For example, increasing the weight of food preference resulted in meal plans with foods they liked mos

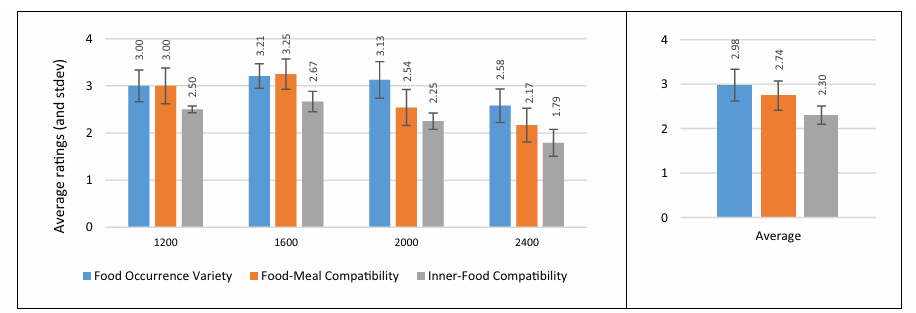
**10.8 Results – Expert Evaluation of Meal Quality**

Nutritionists rated plans on criteria like:

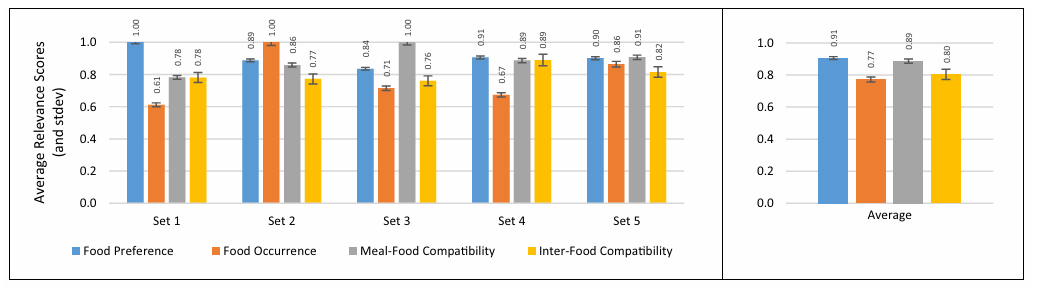
* Variety.
* Inter-food and meal-food compatibility.

Overall score: **2.67 / 4**, with **basic-only plans** scoring slightly higher than mixed ones.

****  
Expert scores for mixed (composite) food plans.



### **10.9 Results – Cost Weight Experiments**

To test the robustness of the scoring model, five different weight settings were applied. Results showed that the relevance scores appropriately adjusted to emphasize the chosen dominant factor in each configuration.

Scores for each cost factor across different weighting settings

### **10.10 Discussion**

MPG’s strength lies in its **flexibility and automation**. Users can control which factors matter more, and the system dynamically adjusts meal selection accordingly. It bridges the gap between rigid pre-defined meal apps and expensive expert consultations.

Challenges noted:

* Computational time for large food databases.
* Dependency on predefined food compatibility graphs.

### **10.11 Future Work**

The authors plan to:

* Integrate MPG into a broader system called **PIN (Personal Intelligent Nutritionist)**.
* Add **real-time health monitoring**, **exercise tracking**, and **adaptive goal updates**.
* Incorporate **food image recognition** and **machine learning-based preference learning**.
* Enable support for **special diets** (e.g., vegan, gluten-free).

### **10.12 Conclusion**

MPG represents a significant advancement in personalized, automated meal planning. It addresses gaps in current systems by integrating nutrition science, optimization theory, and user-centric personalization. With further enhancements, MPG has the potential to become a comprehensive digital nutrition assistant.

**11.0 Introduction (DEVELOPING A GENETIC ALGORITHM BASED DAILY CALORIE RECOMMENDATION SYSTEM)**

The research focuses on the development of a \*Genetic Algorithm (GA)-based daily calorie recommendation system\* tailored to individual human needs. The paper addresses the problem of generic calorie estimations that fail to consider individual differences such as body composition, metabolism, and activity level. To improve precision in dietary recommendations, the authors integrate computational intelligence, specifically evolutionary computing, into a nutrition-focused application.

The necessity of personalized dietary recommendations is emphasized due to rising global health issues such as obesity, diabetes, and other lifestyle diseases. Existing manual or traditional digital calorie calculators are often rigid and lack adaptability, prompting the need for more dynamic, intelligent systems like the one proposed in this study.

**11.1 Methodology**

The methodology includes the design and implementation of a \*Genetic Algorithm\* to optimize daily calorie recommendations. GA is chosen because of its adaptability and efficiency in solving nonlinear optimization problems, particularly where multiple parameters must be balanced.

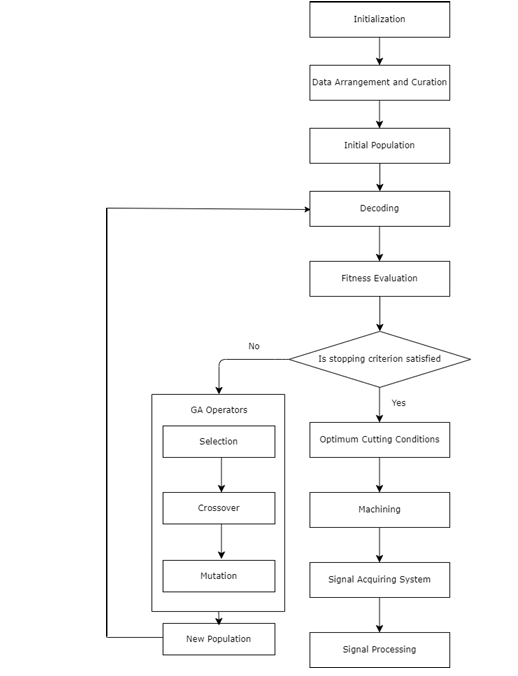
**11.2 Data Parameters**

The system considers critical physiological and lifestyle variables:

* Age
* Gender
* Height
* Weight
* Activity level (sedentary, moderately active, very active)
* Body Mass Index (BMI)

These inputs help to estimate the Basal Metabolic Rate (BMR), which is then modified based on activity level to arrive at the Total Daily Energy Expenditure (TDEE).

**Block diagram of system architecture:**

****

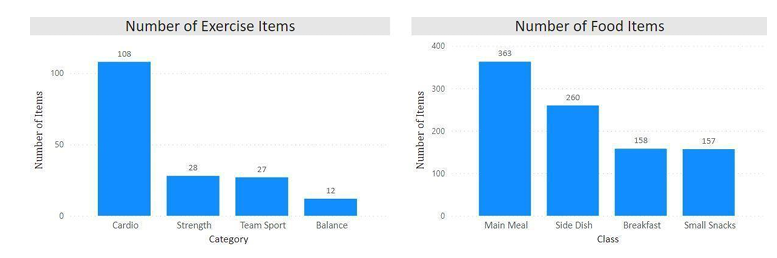
**11.3 Genetic Algorithm Design**

The GA is constructed with the following components:

* Population Initialization\*: Each individual in the population represents a calorie recommendation.
* Fitness Function\*: Designed to minimize the error between the calculated energy requirement and the desired objective (e.g., weight maintenance, loss, or gain).
* Selection\*: Roulette wheel selection method.
* Crossover and Mutation\*: Applied to introduce variation and help the algorithm explore the solution space.
* Termination Condition\*: Either a fixed number of generations or a convergence criterion.

The fitness function plays a crucial role and is based on the Harris-Benedict equation, modified according to the individual’s lifestyle.

**GA operational flow:**

****

**11.4 System Implementation**

The implementation is carried out in a MATLAB environment, and the interface is developed for user-friendliness. Users input their details, and the GA processes the data to output the recommended calorie intake.

GUI interface showing input fields and output:

**11.5 Performance Metrics**

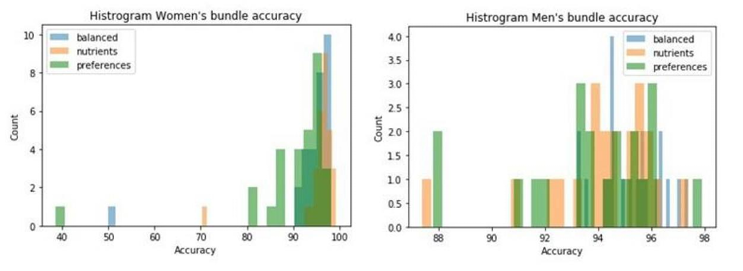
Several test cases are examined to validate the effectiveness of the GA-based system:

* Accuracy: Compared to standard manual calculators, the system shows better accuracy and adaptability.
* Time Efficiency: The algorithm converges in a reasonable number of generations, making it computationally practical.
* Personalization: Unlike static calculators, the model adapts to a user’s physical and activity profile dynamically.

**11.6 Results and Discussion**

The results show a consistent improvement in calorie prediction accuracy. GA finds optimal values that more closely align with medically and nutritionally accepted standards. The system demonstrates a superior ability to adjust to different user profiles, including overweight or underweight individuals, and those with high or low activity levels.

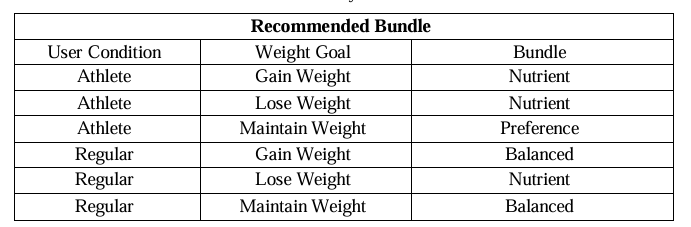
**Graph comparing GA results vs conventional calculators:**



**11.7 Case Studies**

The paper includes several user profiles to illustrate the adaptability of the system. Each profile varies in age, gender, and lifestyle, and the calorie recommendation varies accordingly, validating the flexibility of the GA approach.

**Comparison of calorie recommendations for different users:**

****

**11.8 Limitations**

While effective, the system does not yet account for more nuanced health factors such as existing medical conditions, allergies, or psychological eating behaviors. These limitations highlight the potential for future enhancements using multi-objective optimization or hybrid AI models.

**11.9 Conclusion**

The proposed Genetic Algorithm-based daily calorie recommendation system offers a **more intelligent and personalized approach** to dietary planning. It addresses the shortcomings of static calculators by incorporating evolutionary computation principles, offering better adaptability, accuracy, and user-centric recommendations.

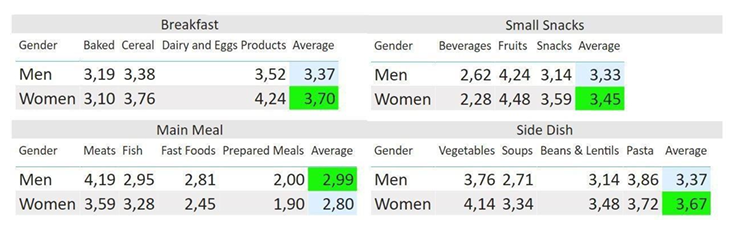
The system is scalable and can be further developed into a mobile or web application. It serves as a proof of concept for the integration of AI techniques in public health tools, especially in dietary management and lifestyle monitoring.

**11.10 Recommendations for Future Work**

Future developments could include:

* Integration of additional health parameters such as chronic conditions or blood sugar levels.
* Development of mobile or cloud-based applications for broader accessibility.
* Use of hybrid models (e.g., combining GA with neural networks) for enhanced learning and prediction.
* Inclusion of feedback mechanisms for continual user-specific calibration.

**Proposed future model architecture integrating feedback loop:**

****