

Constraint-Aware Automated Daily Meal Planning: A Controlled Comparison of Genetic Algorithm and Particle Swarm Optimization

Kohei Arai^{1*}, Diva Kurnianingtyas², Nathan Daud³

Faculty of Science and Engineering, Saga University, Saga, Japan¹

Faculty of Computer Science, Universitas Brawijaya, Malang 65154, Indonesia^{2, 3}

Abstract—This study presents a controlled proof-of-concept comparison of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for automated daily meal-plan generation. The evaluation is deliberately restricted to the data available in the current prototype: a 24-item Japanese-style food database, one nutritional profile, one random seed (42), 100 iterations, and an equal population/swarm size of 60. Both optimizers use the same scalarized fitness function that combines nutrient-target adherence, penalties for constraint violations, and a food-diversity bonus. In the representative run, GA achieved a final fitness of 5.15 compared with 4.92 for PSO and reached its plateau by generation 14, whereas PSO reached its final value near iteration 75. The GA solution had a daily cost of ¥980, while the PSO solution had a lower daily cost of ¥650. Both solutions remained within the specified macronutrient tolerance bounds. These results reveal a configuration-specific trade-off: GA produced a higher scalar fitness and earlier convergence, whereas PSO produced a lower-cost plan and smaller deviations for several macronutrients. Because the experiment uses a single profile and a single random seed, the findings are descriptive and should not be interpreted as statistical evidence of general algorithmic superiority.

Keywords—Automated meal planning; genetic algorithm; particle swarm optimization; constrained combinatorial optimization; personalized nutrition; computational intelligence

I. INTRODUCTION

Personalized meal planning can be formulated as a computational decision problem in which nutritional adequacy, dietary restrictions, user preferences, and affordability must be considered simultaneously. Recent artificial-intelligence-based meal-planning studies have explored reinforcement learning and hybrid optimization to tailor food recommendations to individual needs and preferences [1], [2]. Population-based optimization has also been applied to nutrition- and budget-aware menu generation, including maternal nutrition planning [3].

From an optimization perspective, meal planning is a constrained combinatorial search problem because food choices are discrete and multiple criteria must be satisfied at the same time. Contemporary diet-optimization studies show that mathematical models can balance nutritional adequacy with affordability, acceptability, food waste, and other practical objectives [4], [5], [6], [7]. Population-based metaheuristics such as GA and PSO are therefore attractive when the search space is discrete or mixed, the objective is nonlinear, and

feasibility is controlled through penalties or decoding rules [7], [8].

This study makes four contributions within the scope of the available prototype data. First, it formulates four-meal daily planning as a constrained combinatorial optimization problem over a 24-item food database. Second, it implements GA and PSO under matched computational settings and a common scalarized fitness function so that differences are not caused by different objective definitions. Third, it contrasts an integer-based GA representation with a continuous latent-space PSO representation that is decoded into discrete food selections. Fourth, it provides a descriptive head-to-head comparison using final fitness, convergence behavior, daily cost, and macronutrient deviations. The scope is intentionally limited to one user profile and one random seed; therefore, the study evaluates the behavior of the two implementations under a controlled configuration rather than claiming population-level or statistically generalizable superiority.

The remainder of this study is organized as follows. Section II reviews recent work on GA, PSO, and computational meal planning. Section III defines the decision variables, constraints, and fitness function. Section IV describes the system architecture, Sections V and VI present the algorithm design and implementation, Section VII specifies the experimental configuration, and Section VIII reports the descriptive results. Section IX discusses the observed trade-offs and limitations, followed by the conclusion and future research directions.

II. RESEARCH BACKGROUND AND RELATED WORK

A. Genetic Algorithm (GA)

GA is an evolutionary optimization technique that improves a population of candidate solutions through selection, crossover, mutation, and survival mechanisms. Recent reviews emphasize that representation, fitness design, crossover, mutation, and diversity preservation strongly influence search behavior and convergence [7], [9], and the same body of work shows that GA and PSO performance is sensitive to control-parameter choices such as mutation/inertia settings and population size [11]. For the present meal-planning problem, a direct integer representation is appropriate because candidate solutions are constructed from discrete food identifiers grouped into meal slots.

*Corresponding author.

B. Particle Swarm Optimization (PSO)

PSO is a population-based swarm-intelligence method in which particles update their positions using their own best-known position and the best position discovered by the swarm. Although PSO is naturally defined in continuous space, it has been widely adapted to constrained, discrete, and application-specific optimization through customized encodings and decoding mechanisms [8]. In this study, each continuous particle position is interpreted as a food-selection strength and is decoded into a discrete meal plan.

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)) \quad (1)$$

where, ω is the inertia weight controlling the exploration-exploitation balance, c_1 and c_2 are the cognitive and social acceleration coefficients, and r_1 and r_2 are uniformly distributed random scalars. For the discrete meal-selection task, the continuous position vector is not used directly as the final plan; instead, it is decoded by selecting the highest-strength eligible foods for each meal.

C. Related Work

Recent computational nutrition research has moved beyond static rule-based recommendation toward optimization and adaptive decision support. Authors in [1] used reinforcement learning to generate personalized meal plans with an emphasis on user acceptance. Authors in [2] combined PSO and simulated annealing with analytic hierarchy process weighting for personalized meal planning under multiple user-specific criteria. Authors in [3] compared several population-based optimizers, including GA and PSO, for pregnancy menu planning using nutritional requirements and food budget as optimization considerations.

Other recent diet-optimization studies demonstrate that food planning can be framed around multiple practical constraints. Authors in [4] incorporated nutrition, cost, package sizes, and food waste into a mathematical meal-planning model, while [5] optimized diet structures to improve nutritional adequacy while accounting for affordability and environmental implications. Authors in [6] further showed that nutritional requirements can materially constrain optimized healthy eating patterns. The present study is narrower in scope than these works: it does not model a population, weekly planning, sustainability, or clinical outcomes. Instead, it isolates a controlled GA-versus-PSO comparison for a four-meal daily plan using the same 24-food database, the same scalar fitness function, the same population size, and the same iteration budget. This controlled design supports implementation-level comparison, while the single-profile and single-seed scope limits statistical generalization.

III. PROBLEM FORMULATION

A. Decision Variables

A daily meal plan is represented as a set of four meals $M = \{breakfast, lunch, dinner, snack\}$. Each meal $m \in M$ contains a subset of food items $Fm \subseteq F$ from the food database

F , where, $|F| = 24$. Each food item $f \in F$ is characterized by its nutritional content per serving (kcal, protein, fat, carbohydrates), unit price (JPY), food category, and allergen list.

B. Constraints

The following constraints must be satisfied:

- Nutrient bounds: target caloric intake within $\pm 10\%$, protein within $+20\%/-15\%$, fat within $\pm 20\%$, carbohydrates within $+15\%/-15\%$.
- Allergen exclusion: no food item containing a user-specified allergen may appear in any meal.
- Dislike exclusion: explicitly disliked food IDs are permanently excluded.
- Budget constraint: the total daily cost $\sum price(f)$ must not exceed the daily budget (default: ¥1,500).
- Meal structure: each meal may contain 1–4 food items drawn from meal-appropriate categories (e.g., grains, proteins, vegetables, dairy).

C. Fitness Function

A single scalar fitness value is used to rank candidate meal plans. The objective combines nutritional adequacy, penalties for constraint violations, and a diversity reward. Accordingly, the present implementation is a scalarized multi-criteria optimization model rather than a Pareto-based multi-objective optimizer.

$$F(plan) = S_{nut} - C_{violation} + \lambda \cdot diversity(plan) \quad (2)$$

The nutrient score $S_{nut} = \frac{\sum_j w_j}{(1 + dev_j)}$ rewards proximity to nutritional targets, where $dev_j = \max(0, actual_j - ub_j, lb_j - actual_j)$ represents the out-of-bound deviation for nutrient j and w_j is its weight; protein is weighted $1.5\times$ in the current implementation. Constraint penalties subtract $Cal_{allergen} = 100$ for an allergen violation, $C_{dislike} = 10$ for each disliked item, and $0.05 \times$ overage for a budget violation. The diversity term is $\lambda \cdot \log(1 + uniquefoods)$, with $\lambda = 0.3$. Because all components are aggregated into one fitness value, the reported fitness should be interpreted according to this specific weighting and penalty structure.

IV. SYSTEM ARCHITECTURE

The prototype uses a backend-centric architecture (Fig. 1). A RESTful POST /plan endpoint receives a user-profile JSON payload containing nutritional targets, dietary constraints, and the selected optimizer (GA or PSO). The optimization engine executes server-side and returns a structured meal plan, per-food nutritional information, total cost, and optimizer-trace metadata. Both optimizers access the same food database and the same fitness engine, which keeps the comparison focused on the search mechanisms rather than on different data or objective functions.

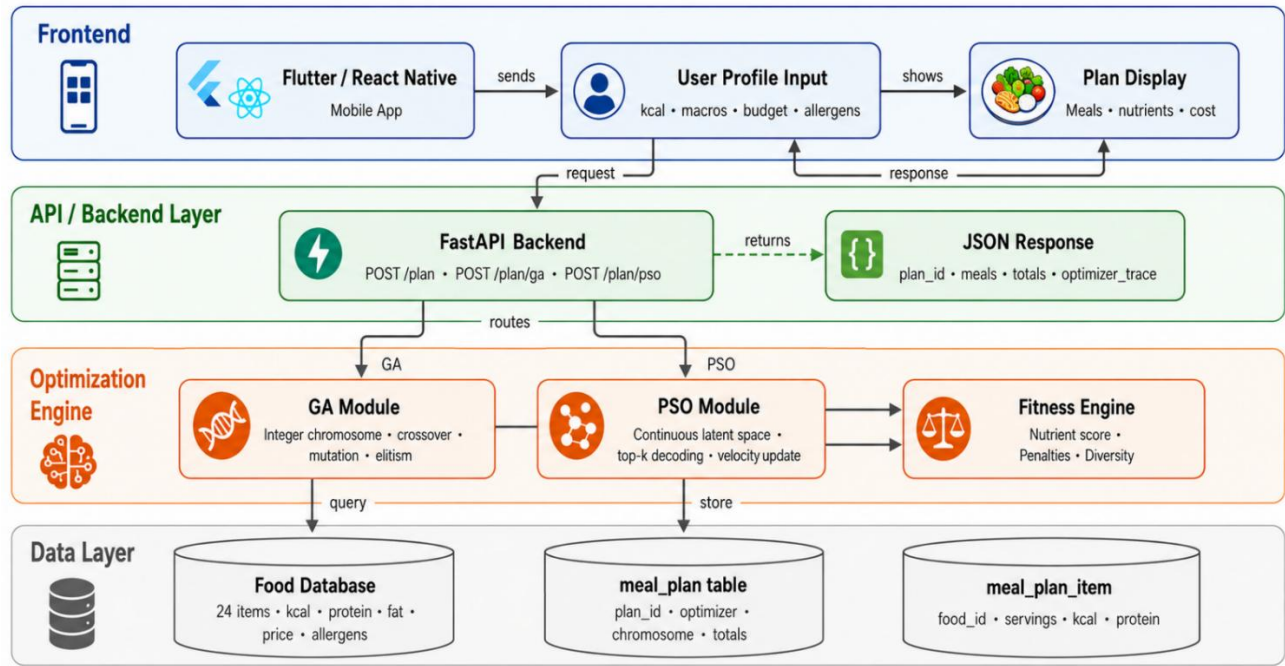


Fig. 1. System architecture of the automated daily meal-planning prototype. The frontend captures the user profile and displays the generated plan; the FastAPI backend routes requests to either the GA or PSO module; and both optimizers use the same fitness engine and data layer.

Table I summarizes the prototype components. Flutter/React Native is used for the frontend, FastAPI handles JSON-based communication, GA and PSO provide alternative optimization engines, and a lightweight SQLite/JSON food store contains the nutritional, price, category, and allergen attributes required by the fitness function.

TABLE I. SYSTEM COMPONENT OVERVIEW

Layer	Component	Responsibility
Frontend	Flutter / React Native	User profile input; plan display
API	FastAPI (Python)	POST /plan endpoint; JSON I/O
Engine	GA / PSO modules	Combinatorial meal search
Data	Food DB (SQLite/JSON)	Nutritional facts; allergen flags

V. ALGORITHM DESIGN

A. GA Encoding and Operators

Each GA chromosome is an integer-keyed meal matrix. The genome is divided into four meal slots, each containing a variable-length list (1–4) of food IDs drawn from the meal-appropriate subset of the food database. Initialization randomly samples valid foods per meal, ensuring allergen compliance. The key operators are:

- 1) *Selection*: tournament selection with $k = 3$ candidates, choosing the individual with the highest fitness.
- 2) *Crossover*: for each meal slot, one parent's foods are inherited with probability 0.5 (uniform crossover at the meal level).
- 3) *Mutation (rate $pm = 0.15$)*: per meal, randomly replace a food with another valid food, insert an additional food (if $|meal| < 4$), or remove a food (if $|meal| > 1$).

4) *Elitism*: the top 10% of individuals are carried forward unchanged to the next generation. This rate follows a commonly used default in the GA literature rather than a value selected through tuning for the present problem, and it is one of several factors that may contribute to GA's faster observed convergence, since preserving the fittest individuals each generation directly protects incremental fitness gains from being lost to crossover or mutation. Its individual contribution relative to selection pressure and mutation rate was not isolated in this study.

B. PSO Encoding and Decoding

Each PSO particle is a real-valued vector $x \in [0, 1]^N$, where $N = 24$ is the number of foods. Component x_i represents the selection strength of food i . At each iteration, particle positions are decoded into discrete meal plans by selecting, for each meal m , the top- k foods ($k \in \{1, 2, 3\}$ sampled uniformly) whose selection strengths are highest among the meal-appropriate, allergen-free food pool. Velocity updates follow the standard inertia-weight PSO equation with $\omega = 0.7$, $c_1 = c_2 = 1.5$.

The two representations determine meal size through different mechanisms. In GA, the number of foods per meal changes only through insertion and deletion mutations acting on an already-instantiated meal. In PSO, meal size is instead re-sampled at every iteration, since k is drawn uniformly from $\{1, 2, 3\}$ independently for each meal each time a particle is decoded. This introduces a source of per-iteration randomness in PSO's meal construction that has no counterpart in GA's representation. Consequently, the two optimizers are matched in population/swarm size, iteration budget, random seed, and fitness function, but not in the decoding-level stochasticity of meal length; the comparison should accordingly be read as

controlled at the level of computational budget and objective function rather than as a fully matched search-space representation.

VI. IMPLEMENTATION

Both algorithms are implemented in Python 3 without external optimization libraries so that the encoding, operators, decoding rules, and fitness calculations remain explicit in the prototype code. The implementation consists of approximately 300 lines organized into the following modules:

1) *Food database*: A list of 24 food items (Japanese-style) with nutritional profiles, prices, allergen flags, and meal-category labels.

2) *Fitness engine*: A shared function computing the multi-component fitness score used by both GA and PSO without modification.

3) *GA module*: GeneticAlgorithm class with configurable population size, number of generations, mutation rate, and elite ratio.

4) *PSO module*: PSO class with configurable swarm size, iteration count, inertia weight, and cognitive/social coefficients.

The algorithmic settings used for the representative comparison are shown in Table II.

TABLE II. HYPERPARAMETER SETTINGS USED IN THE REPRESENTATIVE EXPERIMENT

Parameter	GA	PSO
Population/Swarm size	60	60
Iterations/Generations	100	100
Mutation rate (pm)	0.15	-
Elite ratio	10%	-
Inertia weight (ω)	-	0.70
Cognitive Coeff. (c_1)	-	1.50
Social Coeff. (c_2)	-	1.50
Random seed	42	42

Algorithm 1 presents a representative excerpt of the GA chromosome initialization used in the prototype.

Algorithm 1: GA Chromosome Initialization

```
def _init_individual(self) -> MealPlan:
    | return MealPlan(meals={mt: random_meal(mt, self.profile,
    | random.randint(1,3))
    for mt in MEAL_STRUCTURE })
```

VII. EXPERIMENTAL SETUP

The controlled experiment uses one test profile: 2,000 kcal/day, 90 g protein, 60 g fat, 250 g carbohydrates, a ¥1,500 daily budget, and shellfish as an allergen restriction. Both implementations use seed 42, 100 iterations, and an equal population/swarm size of 60. The best-so-far fitness is recorded at each iteration. Because only one seed is available, all numerical results are descriptive single-run observations rather than estimates of expected performance.

The comparison uses final scalar fitness, plateau iteration, daily cost, and deviations from the calorie, protein, fat, and carbohydrate targets. These measures are reported directly from the available seed-42 execution and are interpreted descriptively within the scope of the prototype.

VIII. RESULTS

Under the configured single run (seed 42), GA and PSO generated the following best daily meal plans:

GA Optimal Menu (Fitness: 5.15)

Meal Details:

Breakfast: Salmon, Tofu, Banana, Apple

Lunch: Rice, Bread, Spinach, Broccoli

Dinner: Soba Noodles, Pork Miso Soup, Chicken Breast, Cabbage

Snacks: Orange, Almonds, Rice Ball, Milk

Total: 2,045 kcal / Protein: 106g / Cost: ¥980

PSO Optimal Menu (Fitness: 4.92)

Meal Details:

Breakfast: Bread, Pork Loin, Soba Noodles

Lunch: Miso Soup, Bread

Dinner: Miso Soup, Bread, Pork Loin

Snacks: Rice Ball, Milk, Orange Total: 2,027 kcal / Protein: 94g / Cost: ¥650

Implementation characteristics relevant to interpretation of the results are summarized as follows:

- GA encoding: integer food identifiers organized by meal slot.
- PSO encoding: continuous food-selection strengths in $[0,1]$ decoded to the highest-ranked eligible foods.
- Shared fitness function: scalar combination of nutrient adherence, constraint penalties, and food diversity.
- GA operators: tournament selection, meal-level uniform crossover, replacement/insertion/deletion mutation, and elitism.
- PSO update: inertia, personal-best attraction, and global-best attraction followed by discrete meal decoding.

For seed 42, GA achieved the higher final scalar fitness (5.15 versus 4.92). This comparison describes the configured execution only and is interpreted together with convergence behavior, daily cost, and macronutrient deviations rather than as evidence of universal algorithmic superiority.

A. Convergence Behavior

Fig. 2 shows the best-so-far fitness trace over 100 iterations for the representative run. GA starts at 3.28, reaches 5.09 by generation 10, and plateaus at 5.15 by generation 14. PSO starts at 3.67, reaches 4.88 by iteration 10, and improves to its final value of 4.92 near iteration 75, as shown in the full-resolution

curve in Fig. 2. Table III samples only four checkpoints (iterations 1, 10, 25, and 100) for brevity; it does not include a point at iteration 75, so readers should consult Fig. 2 for the exact location of PSO's late-stage increase. Relative to each algorithm's own final plateau in this run, GA required about 5.4 times fewer iterations than PSO (14 versus 75) to reach its final value. This pattern is consistent with the direct discrete GA

representation and meal-level recombination implemented in the prototype and is interpreted only within the current configuration. Table III reports representative fitness values at selected iterations and complements the full convergence trace in Fig. 2. Macronutrient-target adherence is examined separately in Fig. 3.

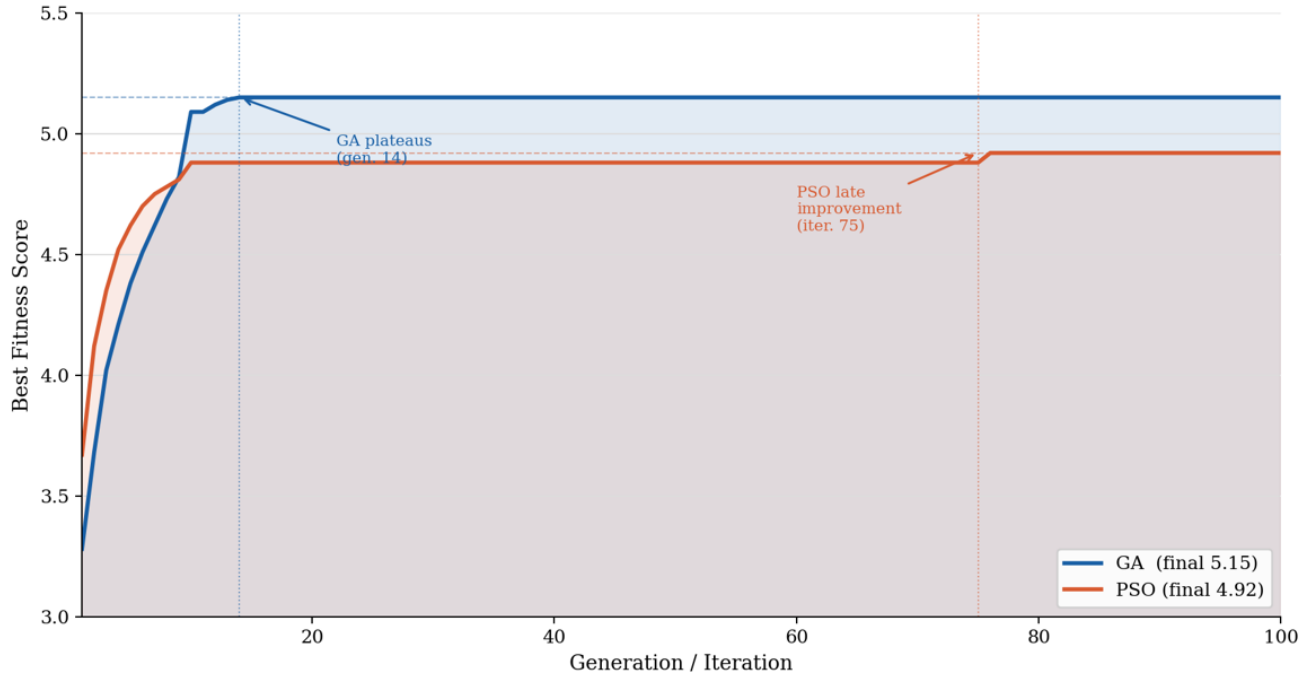


Fig. 2. Representative best-fitness convergence curves for seed 42 over 100 iterations. GA plateaus at 5.15 by generation 14, whereas PSO's fitness plateaus at 4.88 and then increases again near iteration 75, reaching its final value of 4.92. The figure is descriptive of this single run and is not an averaged convergence profile.

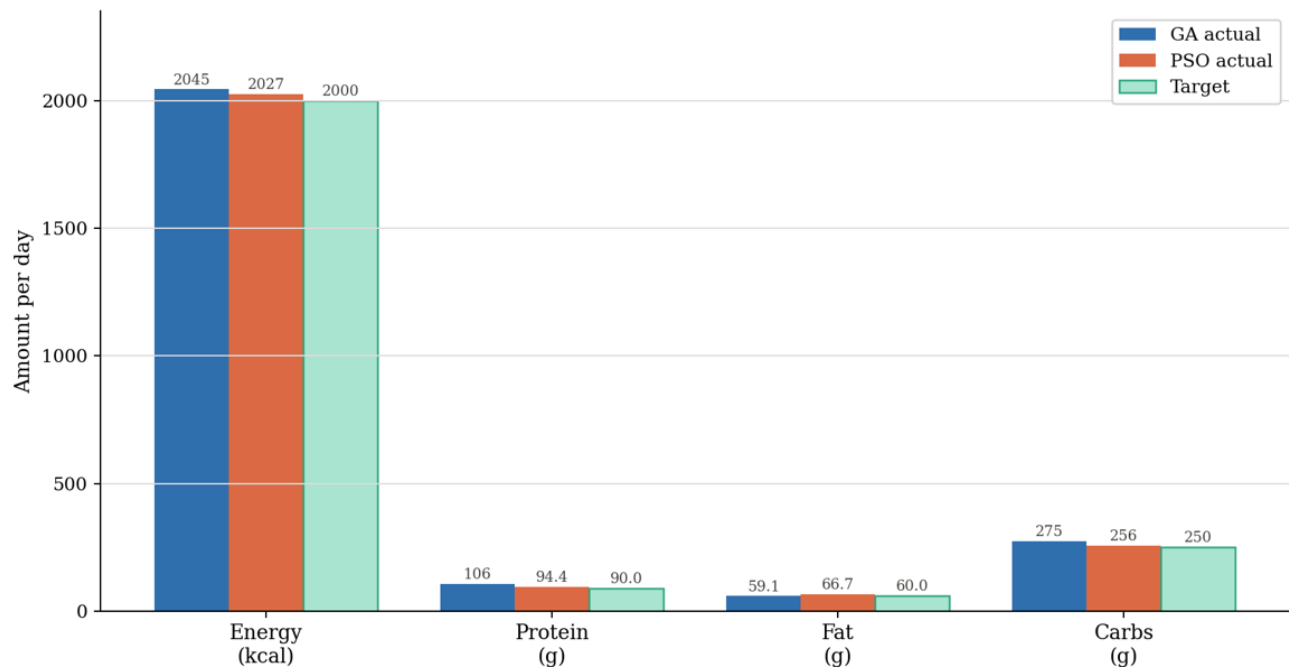


Fig. 3. Macronutrient totals for the representative GA and PSO best plans compared with the daily targets. The plotted values correspond to the single seed-42 experiment.

TABLE III. REPRESENTATIVE CONVERGENCE TRACE FOR SEED 42

Iteration	GA Fitness	PSO Fitness
1	3.28	3.67
10	5.09	4.88
25	5.15	4.88
100	5.15	4.92

These four checkpoints were chosen for brevity and do not include iteration 75, near which PSO's fitness rises from 4.88 to its final value of 4.92; this transition is visible in the full-resolution trace in Fig. 2.

B. Nutrient Accuracy

Fig. 3 compares the macronutrient totals of the best GA and PSO plans with the specified targets. The GA plan provides 2,045 kcal, 106.1 g protein, 59.1 g fat, and 275.0 g carbohydrates, corresponding to deviations of +2.3%, +17.9%, -1.5%, and +10.0%, respectively. The PSO plan provides 2,027 kcal and 94.4 g protein and, according to Table IV, shows smaller deviations than GA for energy, protein, and carbohydrates, while GA is closer to the fat target. Both plans remain within the tolerance bounds encoded in the current fitness formulation.

C. Summary

Representative final fitness and daily cost are GA = 5.15 / ¥980 and PSO = 4.92 / ¥650. Table IV consolidates the raw head-to-head metrics used in the subsequent discussion.

TABLE IV. DESCRIPTIVE HEAD-TO-HEAD COMPARISON FOR ONE PROFILE AND SEED 42

Metric	GA	PSO
Final fitness	5.15	4.92
Convergence generation	~14	~75*
Daily cost (¥)	980	650
Caloric accuracy ($\Delta kcal$)	+45 (+2.3%)	+27 (+1.4%)
Protein accuracy (Δg)	+16.1 (+17.9%)	+4.4 (+4.9%)
Fat accuracy (Δg)	-0.9 (-1.5%)	+6.7 (+11.2%)
Carbohydrate accuracy (Δg)	+25 (+10.0%)	+5.8 (+2.3%)

*PSO's convergence generation is read from the full-resolution trace in Fig. 2. Table III samples only iterations 1, 10, 25, and 100 for brevity and does not itself include a point at iteration 75.

IX. DISCUSSION

A. Advantages of GA for Meal Planning

The earlier GA plateau observed in Fig. 2 is consistent with the representation used in this prototype. Integer chromosomes directly encode discrete food selections, and meal-level crossover can preserve and recombine partial meal structures. Genetic-algorithm performance is known to depend strongly on representation, selection, crossover, mutation, and parameter settings rather than on the algorithm label alone [7], [9]. A more directly comparable meal-planning study by [3] evaluated GA together with PSO, duck swarm algorithm (DSA), and whale optimization (WO) on a 55-food maternal menu-planning problem. That study reported statistically different fitness behavior among the evaluated optimizers. The present experiment likewise shows different GA and PSO behavior in

the representative run, with GA reaching a higher final fitness and an earlier plateau. However, the fitness formulations, objective directions, datasets, and experimental protocols differ across the two studies. Their numerical fitness values should therefore not be compared directly.

B. Advantages of PSO for Budget Efficiency

PSO produced a best plan costing ¥650 compared with ¥980 for GA, a reduction of approximately 33.7%, while remaining within the encoded nutritional tolerance bounds. It also achieved smaller deviations for energy, protein, and carbohydrates in the reported solution. This trade-off is relevant to recent personalized nutrition research, where cost and nutritional criteria are often optimized jointly rather than treated as a single performance dimension [2], [5]. For example, [2] showed that a hybrid PSO-Simulated Annealing approach could improve the accuracy of standard PSO when optimizing consistency in an AHP-based personalized meal-planning framework. This illustrates that PSO performance can change substantially with hybridization and problem representation. More broadly, the systematic review by [8] emphasizes that PSO behavior is sensitive to parameter settings, variants, and the balance between exploration and exploitation. Thus, the lower-cost solution obtained here should be interpreted as a configuration-specific result of the continuous latent representation, top-k decoding rule, and current fitness weights, rather than as evidence that PSO generally produces cheaper meal plans.

C. Comparison with Recent Meal-Planning Studies

The present study occupies a narrower but complementary position relative to recent AI-based meal-planning research. Amiri et al. [1] combined reinforcement learning and collaborative filtering and evaluated their approach on a large recipe-and-user-interaction setting, explicitly targeting user acceptance, preference, and nutritional adequacy. By comparison, the current prototype uses a 24-item food database and a single static nutritional profile, with no feedback loop or learned preference model. Its contribution is therefore not adaptive personalization, but a controlled implementation-level comparison of two population-based optimizers under the same food data, constraints, population size, iteration budget, and scalar fitness function.

The observed tension between GA's higher scalar fitness and earlier convergence and PSO's lower cost also reflects the broader multi-criteria nature of diet optimization. Liu et al. [5] demonstrated that diet optimization at a much larger scale requires simultaneous consideration of nutrition, affordability, acceptability, and environmental consequences. Dussiot et al. [6] separately showed that realistic nutrient requirements, particularly micronutrient constraints, can materially alter feasible optimized diets. These studies reinforce two implications for the present results. First, no single metric should be treated as sufficient evidence of overall meal-plan quality. Second, the apparent preference between GA and PSO may change when additional criteria such as micronutrients, portion sizes, acceptability, or longer planning horizons are introduced. Direct numerical benchmarking against these studies would be inappropriate because their decision variables, objectives, datasets, and evaluation protocols differ substantially from those used here.

D. Limitations

This study has several important limitations. First, the food database contains only 24 Japanese-style items, which limits the search-space diversity and does not represent a comprehensive real-world food catalog. Second, only one nutritional profile and one random seed (42) are evaluated. The study therefore reports descriptive single-run results rather than mean performance, standard deviation, or significance testing across repeated runs, and the findings should not be read as statistically generalizable. Established nonparametric methodologies for comparing stochastic optimizers across repeated runs, such as the Wilcoxon signed-rank test, are available for this purpose and are recommended for future multi-seed replication [12]. Third, portion sizes are fixed at one serving per selected item, preventing continuous portion adjustment. Fourth, the fitness function models calories and three macronutrients but not micronutrients, meal timing, clinical conditions, or longitudinal variety. Fifth, neither GA's operator settings (mutation rate, elite ratio, tournament size) nor PSO's coefficients (inertia weight, cognitive and social coefficients) were tuned or subjected to a parameter-sensitivity analysis. Both GA and PSO performance are known to depend strongly on such settings [7]–[9], [11], so the reported comparison reflects one arbitrarily chosen configuration per algorithm rather than each algorithm's best achievable performance. Sixth, the manuscript does not report the wall-clock or computational runtime of the two optimizers, even though runtime is a practically relevant dimension for comparing population-based metaheuristics and has been highlighted in prior GA–PSO comparisons [10]. These limitations define the present work as a focused proof-of-concept comparison and should be considered when interpreting the reported GA–PSO trade-offs.

X. CONCLUSION

This study presents a controlled proof-of-concept comparison of GA and PSO for automated four-meal daily planning using a common 24-food database, identical population/swarm size, an equal iteration budget, and the same constraint-aware scalar fitness function. Both algorithms produced meal plans that satisfied the encoded macronutrient tolerance bounds within the representative execution.

For the single profile and seed 42, GA achieved the higher final fitness (5.15 versus 4.92) and reached its final plateau earlier (generation 14 versus iteration 75, per the full-resolution trace in Fig. 2). PSO, however, produced the lower-cost plan (¥650 versus ¥980) and smaller deviations from the energy, protein, and carbohydrate targets.

Accordingly, the present evidence supports a configuration-specific trade-off rather than a universal recommendation of one optimizer. Under the current scalar fitness, GA is preferable when the emphasis is on the reported fitness value and convergence behavior, whereas the PSO solution is attractive when lower cost and closer matching of several macronutrient targets are prioritized. The conclusions are intentionally confined to the available prototype data and the representative seed-42 execution.

XI. FUTURE RESEARCH DIRECTIONS

Future research may extend the present proof-of-concept by enlarging the food database to represent a broader set of realistic food choices and by allowing serving sizes to vary rather than remaining fixed at one portion per selected item. Methodological extensions may also combine discrete food selection with continuous portion-size optimization or adopt Pareto-based multi-objective optimization when nutrient deviation, cost, and diversity are treated as separate competing objectives. The nutritional scope may be broadened from calories and macronutrients to relevant micronutrients, while longer planning horizons could support weekly or monthly menus with cross-day variety and meal-timing constraints. Further application-oriented work may additionally consider multiple nutritional profiles and domain-expert or user assessment to examine the practical usefulness of the generated meal plans.

DECLARATION ON GENERATIVE AI

Generative AI tools were used only for figure-design assistance. The authors independently verified all scientific content, data, results, interpretations, and references and take full responsibility for the manuscript.

REFERENCES

- [1] M. Amiri, F. Sarani Rad, and J. Li, "Delighting palates with AI: Reinforcement learning's triumph in crafting personalized meal plans with high user acceptance," *Nutrients*, vol. 16, no. 3, p. 346, 2024, doi: 10.3390/nu16030346.
- [2] F. Sarani Rad, M. Amiri, and J. Li, "Optimizing nutritional decisions: A particle swarm optimization–simulated annealing-enhanced analytic hierarchy process approach for personalized meal planning," *Nutrients*, vol. 16, no. 18, p. 3117, 2024, doi: 10.3390/nu16183117.
- [3] D. Kurnianingtyas, N. Daud, and K. Arai, "Automated menu planning for pregnancy based on nutrition and budget using population-based optimization method," *IAES International Journal of Artificial Intelligence*, vol. 14, no. 5, pp. 3483–3492, 2025, doi: 10.11591/ijai.v14.i5.pp3483-3492.
- [4] M. A. van Rooijen, J. C. Gerdessen, G. D. H. Claassen, and S. De Leeuw, "Optimizing household food waste: The impact of meal planning, package sizes, and performance indicators," *Resour. Conserv. Recycl.*, vol. 205, p. 107559, 2024, doi: 10.1016/j.resconrec.2024.107559.
- [5] X. Liu, L. Xin, and X. Li, "The global nutrition can be greatly improved with diet optimization," *Resour. Conserv. Recycl.*, vol. 202, p. 107343, 2024, doi: 10.1016/j.resconrec.2023.107343.
- [6] A. Dussiot et al., "Modeled healthy eating patterns are largely constrained by currently estimated requirements for bioavailable iron and zinc—a diet optimization study in French adults," *Am. J. Clin. Nutr.*, vol. 115, no. 3, pp. 958–969, 2022, doi: 10.1093/ajcn/nqab373.
- [7] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.
- [8] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022, doi: 10.1007/s11831-021-09694-4.
- [9] R. Chaudhary, P. Verma, R. Salgotra, and A. H. Gandomi, "Five decades of genetic algorithms: A systematic and bibliometric review (1975–2025)," *Archives of Computational Methods in Engineering*, pp. 1–75, 2026, doi: 10.1007/s11831-025-10487-2.

- [10] Ali Fozooni, Osman Kamari, Mostafa Pourtalebiyan, Masoud Gorgich, Mohammad Khalilzadeh, Amin Valizadeh, An Analysis of the Operation Factors of Three PSO-GA-ED Meta-Heuristic Search Methods for Solving a Single-Objective Optimization Problem, *Comput Intell Neurosci.* 2022 Oct 14;2022:2748215. doi: 10.1155/2022/2748215.
- [11] Georgios Papazoglou, Pandelis Biskas, Review and Comparison of Genetic Algorithm and Particle Swarm Optimization in the Optimal Power Flow Problem, *Energies* 2023, 16(3), 1152; <https://doi.org/10.3390/en16031152>.
- [12] J. Carrasco, S. García, M.M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, *Swarm and Evolutionary Computation*, 2020, 100665, 54, <https://doi.org/10.1016/j.swevo.2020.100665>.

AUTHOR'S PROFILE

Kohei Arai received BS, MS, and PhD degrees in 1972, 1974, and 1982, respectively. He was with the Institute for Industrial Science and Technology of

the University of Tokyo from April 1974 to December 1978 and was with the National Space Development Agency of Japan from January 1979 to March 1990. From 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He was a professor at Saga University from 1990 to 2017. Also, he was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology from 1998 to 2000 and was a councilor of Saga University in 2002 and 2003. He was also an executive councilor for the Remote Sensing Society of Japan from 2003 to 2005. He was the Vice Chairman of the Scientific Committee and an Award Committee member of ICSU/COSPAR. He has been a Science Council of Japan Special Member (COSPAR Committee) since 2012. He is an Adjunct Professor at Nishi-Kyushu University and Kurume Institute of Technology, as well as Prishtina International University. He has written 134 books and published 779 journal papers as well as 612 conference papers. He received 66 awards, including ICSU/COSPAR Vikram Sarabhai Medal in 2016, and Science award of Minister of Ministry of Education of Japan in 2015. He is now Editor-in-Chief of IJACSA. <https://orcid.org/0009-0001-6433-1592>, <https://scholargps.com/scholars/84340950120240/kohei-arai>, <http://teagis.ip.is.saga-u.ac.jp/index.html>