Application Of A Multi-Objective Genetic Algorithm To The Modified Diet Problem

Elif KALDIRIM
Department of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
elifkaldirim@gmail.com

Zekeriya KÖSE
Department of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
kosezekeriya@yahoo.com

ABSTRACT

The application developed in this study is an optimization program which provides users a daily menu containing all the required amounts of nutrients with a minimum cost and a maximum rating. The rating is taken from the user via a user interface. A user can rate all individual items available in the database or groups of meals on a scale from 0 to 10. The program also uses the age and gender of the user to determine daily nutritional and energy requirements. A multi-objective genetic algorithm is used to determine an appropriate daily menu based on cost, user preferences and nutritional requirements.

Categories and Subject Descriptors

I.2.8-Problem Solving, Control Methods and Search

General Terms

Design, algorithms, optimization

Keywords

Genetic Algorithms, Bi-objective Diet Problem, Multi-objective Genetic Algorithms, NSGA-II Algorithm

1. INTRODUCTION

The application developed in this study is an optimization program which provides users a daily menu containing all the required amounts of nutrients with a minimum cost and a maximum rating. This project is an improved version of the work of Kahraman and Seven [1]. In [1], the two objectives (minimum cost, maximum rating) were combined using a weighted sum approach. However, in this study, we used the two objectives seperately and implemented a state-of-the-art multi-objective genetic algorithm, namely NSGA-II. In [1] only upper bounds on the allowed amounts of daily nutritional requirements were used, however we used a more realistic approach and worked with not only upper bounds but also used lower bounds as given in tables. Another shortcoming in [1] was the fact that the daily menu could contain only one serving of each food item. However, we also allowed multiple servings of the food items. In this paper, first the modified diet problem will be introduced, then the NSGA-II approach and its implementation to the diet problem will be presented.

2. MODIFIED DIET PROBLEM

In [1], a modified version of the classical diet problem [2] is implemented. In this study, we further modify the problem and use a multi-objective GA approach for its solution.

2.1 The Classical Diet Problem

The classical diet problem is a 0/1 multi-dimensional knapsack problem with the objective to generate a menu with the lowest cost subject to some daily nutritional requirement constraints defined as lower and upper bounds on nutritional element (e.g. energy, protein, calcium, iron, vitamin A, etc) intake amounts.

2.2 The Bi-Objective Diet Problem

The bi-objective diet problem introduced in [1] is similar to the classical diet problem but has an additional objective. The additional objective is to maximize the preference of the user. The user is expected to rate meals according to personal taste. By using the age and gender information input by the user, constraints are determined.

The bi-objective diet problem is also a 0/1 multi-dimensional knapsack problem in which there are n numbers of dishes corresponding to items. If the dish will be included in a menu the corresponding decision variable is set to 1 and to 0 if it will not. There are two objectives: minimizing the total cost of dishes, maximizing the satisfaction of the user. A person has some daily nutritional requirements for a healthy life. There are upper and lower bounds of protein, Calcium, Magnesium, carbonhydrate, Iron, energy and all kind of vitamins that a person should take per day. The determined menu should include all necessary nutritions with respect to the lower and upper bounds.

2.3 The Modified Bi-Objective Diet Problem

In this project, the bi-objective diet problem used in [1] is modified. The main difference lies in the fact that in our project the problem is not a 0/1 multi-dimensional knapsack problem. For each dish, an integer in the interval (0-3) is used to represent how many servings of the corresponding dish will be included in the menu (maximum of three servings). The user rates all available dishes on a scale from 0 to 10 based on his/her personal taste. We

Genetic and Evolutionary Computation Conference (GECCO)'06,
July 8–12, 2006, Seattle, WA, USA.
Copyright 2006 ACM 1-59593-186-4/06/0007…$5.00.

will call these numbers *the rating*. The daily nutritional requirements are determined for each user based on his or her age and gender information. Using this information, the constraints are determined from the database[10].

Formulation of the Modified Bi-Objective Diet Problem:

$$\text{maximize} \sum_{j=1}^{n} x_j * (\text{rating})_j \quad \text{and minimize} \sum_{j=1}^{n} x_j * (\text{cost})_j$$

$$\text{subject to} \quad c_{li} \leq \sum x_j * r_{ij} \leq c_{ui} \qquad i \in \{1,2,...,m\}$$

$$x_j \in \{0,1,2,3\} \qquad j \in \{1,2,...,n\}$$

m: the number of constraints, **n:** the number of dishes

***x_j* :** the decision variable of dish *j* representing how many servings of the *j*th dish should be included in the menu.

$(rating)_j$: the rating of *j*th dish, ***$(cost)_j$* :** the cost of *j*th dish.

***c_{ui}* :** the upper bound of constraint number *i*

c_{li} : the lower bound of constraint number *i*

r_{ij} : the nutritional value of dish number *j* with respect to to constraint *i*.

3. THE NSGA-II ALGORITHM

In this section, the NSGA-II algorithm and the way it is used in this project will be explained.

3.1 Background Research

Since they can locate multiple Pareto-optimal solutions in one run, multi-objective evolutionary algorithms (MOEAs)[3], [4], [5], have been suggested over the last decade.

The Non-dominated Sorting Genetic Algorithm (NSGA) was first introduced by Srinivas and Deb[3] and criticised for its three characteristics[7]:

1) High computational complexity of non-dominated sorting: Non-dominated sorting algorithms have a complexity of $O(MN^3)$ (M is the number of objectives while N is the population size)

2) Need for specifying the sharing parameter σ: In order to insure diversity, the sharing method is used, which requires the specification of a sharing parameter. A parameter-less diversity is more desirable.

3) Lack of elitism: Rudolph[6] asserted that elitisim can increase the performance of GAs and keep good individuals in the population.

3.2 Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)

3.2.1 A fast non-dominated sorting approach

NSGA[3] has been criticised for three main reasons, and NSGA-II is suggested instead. The NSGA algorithm finds all non-dominated fronts in a complexity of $O(mN^3)$. In the following we describe the NSGA-II algorithm that has $O(mN^2)$ complexity.

In the NSGA-II algorithm, for each solution the following properties are calculated in a complexity of $O(mN^2)$:

1) n_i :the number of solutions which dominate the solution i

2) S_i :the set of solutions which the solution i dominates

All of the solutions whose n_i=0 are put in the F_1 list. F_1 represents the current non-dominated front. For each solution (i) in F_1, all of the members (j) in S_i are visited and their n_j decreased by one. If n_j becomes zero, member j is put in the list H. After checking all members in F_1, these members are declared as the first non-dominated front and the above procedure is repeated for the list H being the new current front. Every iteration has a complexity of O(N), all non-dominated fronts are declared in a complexity of $O(mN^2)$ in the worst case so the overall complexity is $O(mN^2)$+O(mN) or $O(mN^2)$[7]. Here is the procedure for sorting the population according to the non-domination rank:

```
Fast-nondominated-sort(P)
for each p Є P
  for each q Є P
      if(p<q) then          if p dominates q
          $S_p$= $S_p$U {q}     include q in $S_p$
        else if (q<p)       if p is dominated by q
          $n_p$= $n_p$ +1       increment $n_p$
  if $n_p$=0 then      if no solution dominates p
     $F_1$= $F_1$ U {p} p is a member of 1st front
i=1
while $F_1$≠ Ø
  H = Ø
  for each p Є $F_1$ for each member p in $F_1$
    for each q Є $S_p$ modify members in $S_p$
        $n_q$= $n_q$ -1       decrement $n_q$
        if $n_q$ = 0        if $n_q$ is zero then
            H = H U {q} q is member of H
  i = i+1
$F_i$= H current front is formed using H
```

3.2.2 Density Estimation

For each objective function, the solutions with smallest and largest function values are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute difference in the function values of two adjacent solutions. The overall crowding distance is the sum of the individual distance values corresponding to each objective.

3.2.3 Crowded Comparison Operator $\geq_n$

This operator helps the algorithm to determine which individual should be selected for the newly created population. There are two criteria to compare individuals: non- domination rank and the local crowding distance. The crowded comparison operator first chooses the individual with the smallest rank, if individuals have the same non-domination ranks, the individual with the biggest distance is selected.

4. IMPLEMENTATION OF THE NSGA-II ALGORITHM

The NSGA-II algorithm is implemented for our modified bi-objective diet problem as follows:

First a population of size 100 is created randomly (P_0). For each individual in P_0 all objective function values are calculated, and for each constraint k, if there is any constraint violation, the variable *constr_violation* (starting with zero) is decreased by a ratio of *(violation amount /(upper-lower bound))* eg: if the required amount of calories for a 19 years old girl is 1100-1300 kcal and the gene combination gives a menu with 1600 kcal, then

constr_violation= 0 – (1600-1300)/(1300-1100)= -1.5

A feasible individual must have zero *constr_violation*. After all of these calculations, the population is sorted using the fast-nondominated-sort procedure. Sorting means assigning non-domination ranks to individuals. Then the following steps are applied to the population P_0 to create a child population Q_0:

Tournament Selection: We used a binary tournament selection operator but the selection criterion is now based on the Crowded Comparison Operator $\geq_n$
Recombination: Uniform crossover with 0.5 probability is used.
Mutation: bitwise mutation (with probability of 1/N)

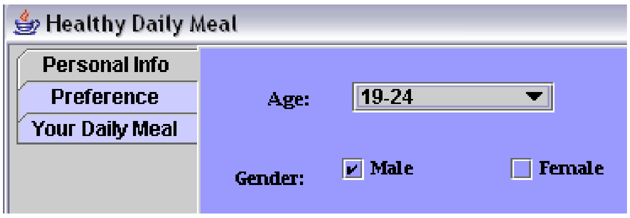
The child population has size N again. The total population, R_t=P_0UQ_0 will have a size of 2N. R_t is sorted based on the non-domination ranks. Then P_{t+1} is created using the first front one by one until the size exceeds N. The solutions of the last accepted front are sorted according to the crowded comparison operator $\geq_n$ and the first solutions that make the size of P_{t+1} exactly N are selected. The same procedure is applied to P_{t+1} to create Q_{t+1} and so on.
One of the most important properties of the NSGA II algorithm is its elitism property. Zitzler and Thiele [8][9] show the importance of elitism in evolutionary multi-criterion optimization. Elitism can fasten the algorithm and prevent the loss of good solutions once they have been found.

5. THE PROBLEM DATA

In our project we use different kinds of data. These are not only personal nutritional requirements, but also nutritional contents and prices of foods. We obtain most of the information from the National Nutrient Database for Standard Reference, Release 17 [10]. We have many tables in our database. These are personal information tables, food information tables and a cost table. The first group includes DRI (Dietary Reference Intake) and RDA (Recommended Dietary Allowance) tables. These tables are taken from the web site of the U.S. Food and Nutrition Board of the National Academy of Sciences [11]. By using the DRI and RDA tables we take the information of maximum and minimum cardinality of the vitamins, elements and energy. The table adequate intake (AI) shows the minimum value of the daily intake of nutrients and tolerable upper intake levels, and the table (UL) shows the maximum value of the daily intake of nutrients [12]. The importance of these values is that; if a person takes an insufficient amount of nutrients or takes more than the upper limit, there might be health risks [12]. The values of (AI) and (UL) may change according to the age and gender information. We use the abbreviated table for taking the amount of nutrients in each food. All kinds of elements, vitamins, electrolytes, makronutrients, energy and water which a food contains are shown [10]. The last important point of our database is cost. In the webpage of the database, there is no cost information, so we assigned estimated prices of the food items in the database. We used the New Turkish Lira (YTL) unit for giving the cost value to the foods. Currently we have only assigned cost values to the fast food group of foods, so our program works only for this food group.

6. THE GRAPHICAL INTERFACE

We created a three-step interface in this project. In the first step the user provides gender and age information.

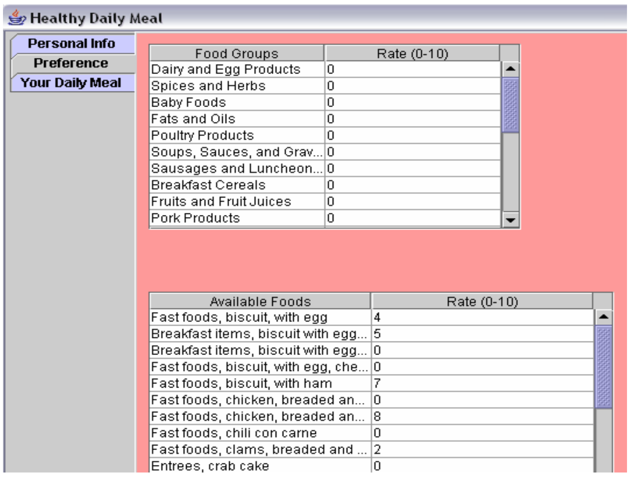


Age: 19-24

Gender: ☑ Male ☐ Female

Figure 1. The first step of user interface.

In the second step the user enters ratings for foods. He or she can enter the value (1 to 10) not only for the main groups of foods, but also for specific food items in each group.

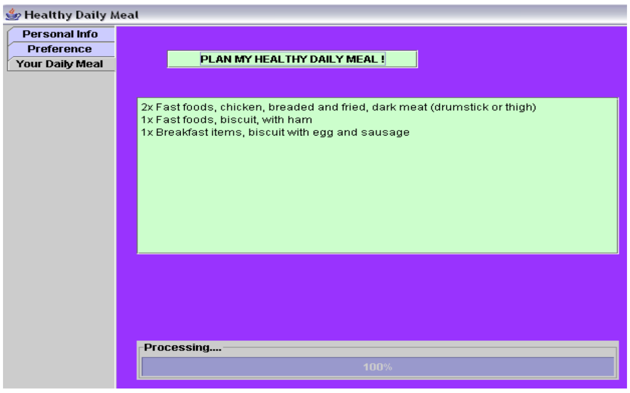


Food Groups	Rate (0-10)
Dairy and Egg Products	0
Spices and Herbs	0
Baby Foods	0
Fats and Oils	0
Poultry Products	0
Soups, Sauces, and Grav...	0
Sausages and Luncheon...	0
Breakfast Cereals	0
Fruits and Fruit Juices	0
Pork Products	0

Available Foods	Rate (0-10)
Fast foods, biscuit, with egg	4
Breakfast items, biscuit with egg...	5
Breakfast items, biscuit with egg...	0
Fast foods, biscuit, with egg, che...	0
Fast foods, biscuit, with ham	7
Fast foods, chicken, breaded an...	0
Fast foods, chicken, breaded an...	8
Fast foods, chili con carne	0
Fast foods, clams, breaded and ...	2
Entrees, crab cake	0

Figure 2. The second step of user interface.

In the final window, there is a button for starting the GA run to generate the recommended menu to the user.



```
2x Fast foods, chicken, breaded and fried, dark meat (drumstick or thigh)
1x Fast foods, biscuit, with ham
1x Breakfast items, biscuit with egg and sausage
```

Figure 3. The third step of user interface.

7. CURRENT STATUS

The following stages of the project are completed at the time this paper is written.

- NSGA-II algorithm implementation
- Testing of NSGA-II algorithm
- Preparing the graphical user interface
- Preparing the database which includes not only personal requirements, but also nutritional constraints
- Connecting the database with the user interface

Based on these, currently we can ask the user the required information, obtain all constraints from the database and run the NSGA-II algorithm and finally propose a healthy daily menu to the user.

8. FUTURE WORK

Most of the project is completed at the time this paper is written. There are a few more steps to complete. One of these is testing the performance of the NSGA-II algorithm and comparing the results with those obtained by using the approach given in [1]. The other is extending the cost table and making the database suitable for working with all types of food groups.

In real life a healthy daily meal should ideally consist of at least three different kinds of dishes. However this is not taken into account in the implementation of this project. As a further improvement, dishes may be categorised as breakfast dishes, lunch dishes, dinner dishes etc. As another enhancement, other stae-of-the-art multi-objective GA approaches can be implemented and tested.

9. ACKNOWLEDGEMENTS

This study is the graduation project of the 8th term students Elif Kaldırım and Zekeriya Köse under the supervision of Asst. Prof. Dr. A. Şima Etaner-Uyar from the Computer Engineering Department of Istanbul Technical University, Turkey.

10. REFERENCES

[1] A. Kahraman, and A. Seven: Healthy Daily Meal Planner, *Genetic and Evolutionary Computation Conference – Undergraduate Student Workshop* (GECCO' 05 UGWS), June 25-29, 2005, Washington, D.C. USA.

[2] Argonne National Laboratory, Mathematics and Computer Science Division, The Optimization Technology Center, NEOS Guide, Case Studies, Diet Problem. http://www.fp.mcs.anl.gov/otc/Guide/CaseStudies/diet/

[3] N. Srinivas, and K. Deb, "Multi-Objective function optimization using non-dominated sorting genetic algorithms," Evolutionary computation, Vol. 2, pp. 221-248, 1995.

[4] J. Horn, N. Nafploitis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multi-objective optimizaton", in Michalewicz, Z., editor, Prıceedings of the first IEEE Conference on Evolutionary Computation, Z. Mihalewicz, Ed. Piscataway, New Jersey: IEEE Service Center, 1994, pp 82-87.

[5] J. Gottlieb: Evolutionary Algorithms for Constrained Optimization Problems Dissertation, Technical University of Clausthal, Department of Computer Science, 2000.

[6] G. Rudolph, "Evolutionary search under partially ordered sets," *Technical Report No. CI-67 /99*, Dortmund: Department of Computer Science/ LS11, University of Dortmund, Germany.

[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II ," KanGAL *Report No. 200001*, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India.

[8] Zitzler, E. (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications. Doctoral thesis ETH NO.13398, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Germany: Shaker Verlag.

[9] Zitzler, E. And Thiele, L.(in press) Comparison of MOEAs: Empirical results. *Evolutionary Computation,* 8.

[10] U.S. Department of Agriculture, Agricultural Research Service.2004.USDA Nutrient Database for Standard Reference, Release 17.Nutrient Data Laboratory Home Page, http://www.nal.usda.gov/fnic/foodcomp.

[11] Food and Nutrition Information Center, Dietary Reference Intakes (DRI) and Recommended Dietary Allowances (RDA), http://www.nal.usda.gov/fnic/etext/000105.html

[12] International Food Information Council Foundation, Dietary Reference Intakes: An Update, http://ific.org/publications/other/driupdateom.cfm, August 2002.