Journal of Food Composition and Analysis 22 (2009) 414–420

Contents lists available at ScienceDirect

Journal of Food Composition and Analysis

journal homepage: www.elsevier.com/locate/jfca

Original Article

Computer-based dietary menu planning

Barbara Koroušić Seljak *

Computer Systems Department, Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 18 October 2007
Received in revised form 26 January 2009
Accepted 19 February 2009

Keywords:
Optimal nutrition
Dietary recommendations and guidelines
Computer-based menu planning
Nutrient balancing
Evolutionary computation
Multi-objective and multi-constraint optimization
Linear programming
Pareto-optimal solutions
Food composition

ABSTRACT

In this paper, we introduce a computer-based method for menu planning, which applies evolutionary computation. First, we formalize the *n*-day menu-planning problem, decomposing it into several sub-problems at the daily-menu and meal-planning level. We reduce the problem to a multi-dimensional knapsack problem. Then, we define an evolutionary algorithm that quickly finds a diverse set of feasible solutions (i.e. optimal menus) with the optimum objective functions' values, without examining all the possibilities. As the problem is constrained, infeasible solutions need to be repaired in order to direct the "evolution" towards the feasible regions. We present greedy repairing methods that slightly differ at the global level and the sub-problems' levels. At the meal-planning level, we couple repairing with linear programming to balance infeasible meals. We conclude the paper with the presentation of empirical results, which showed that the evolutionary method may outperform a human. A computer was able to find the Pareto-optimal front of 21-day menus with respect to a dietary advice in equal or less time than a human professional, who designed a daily menu. However, the human factor is still important in the last stage, when a solution has to be selected from the Pareto front.

1. Introduction

Increasing incidence of chronic diseases (CDs) in modern societies requires a shift from adequate towards *optimal nutrition*, not only providing us with the energy and nutrients but also contributing to the well-being and health. Optimal nutrition, as part of a healthy lifestyle, is an important factor in the prevention of CDs and may also have a therapeutic potential.

Today, many recommendations and guidelines on optimal nutrition, based on results of advanced research methods and tailored to the needs of a society, are available. They consider the current knowledge of the relationship between our immune system, aetiology of CDs and health status. However, their implementation in practice is difficult for several reasons, including the complexity of dietary recommendations and guidelines and limited health literacy that may lead to misunderstanding. Dollahite et al. (1995) found that professionally designed menus published in diet manuals may fail to meet all recommendations and guidelines.

In this paper, we propose a computer-based method for planning optimal menus with respect to agreed evidence-based dietary recommendations and guidelines. This method assists a human in fitting regular menus to new health paradigms.

The paper is organized as follows: Section 2 presents a brief survey of computer-based methods for menu planning and an introduction to evolutionary computation and multi-objective optimization; Section 3 provides a formulation of the menu-planning problem; Section 4 describes an evolutionary algorithm for menu planning; and Section 5 presents empirical results and concluding remarks.

2. Computer-based methods for menu planning

Menu planning is an art that one often learns by a costly heuristic or trial-and-error process. A well-trained professional can cope with the complexity of regular menu planning, but as soon as one attempts to meet the needs of diverse groups, control costs and quality, and schedule production tightly for maximum utilization of labour and equipment time, the probability of success diminishes. Therefore, computer-based methods, which facilitate the routine decisions in menu planning, are useful.

2.1. History

Four decades ago, the need for computer-based methods for menu planning was recognized. As a result, the quality of information and data necessary for computerization of the menu-planning process became available to many institutions and its apparent feasibility thus increased. In 1964, Balintfy (Balintify, 1964; Eckstein, 1983) applied *linear programming*

* Tel.: +386 1 477 33 63; fax: +386 1 477 38 82.
E-mail address: barbara.korousic@ijs.si.

0889-1575/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jfca.2009.02.006

techniques to build the first computer-based menu planner, which optimized menus for nutritional adequacy and budgeted food cost. Brown (1966) developed primitive techniques for controlling the palatability of individual non-selective menus using *random selection techniques*. A year later, these techniques were adopted by Eckstein (1967) to satisfy menus with several constraints, comprising: cost, color, texture, shape, calories, variety and acceptability by the target population.

In the 1970s, interest in computer applications in menu planning appeared to wane; lack of funding and inadequate software components were contributing causes.

In the 1990s, when the field of artificial intelligence was revived, the menu-planning problem became popular again. Two types of menu planners, *case-based* and *rule-based*, were implemented. One of them was JULIA (Hinrichs, 1992), an interactive menu planner that was used to plan meals to satisfy a group of guests, despite conflicting food preferences and evolving constraints. Ganeshan and Farmer (1995) implemented a Prolog catering system. Marling et al. (1999) designed a menu-planning tool for individuals, taking dietary requirements and personal preferences into account. That tool integrated case-based and rule-based reasoning to meet multiple constraints.

A comprehensive review of the use of optimization techniques based on linear and nonlinear programming was given by Darmon et al. (2002).

2.2. Evolutionary computation

The computer-based method for menu planning we have recently proposed (Koroušić Seljak, 2004) is based on *evolutionary computation*.

In computer science, evolutionary computation is a subfield of artificial intelligence that involves *numerical* and *combinatorical optimization* (CO) problems. Evolutionary computation uses iterative progress, such as growth or development in a population of potential problem solutions. The field comprises many techniques, mostly involving metaheuristic[1] optimization algorithms, such as evolutionary algorithms (EAs) and swarm intelligence (Korošec and Šilc, 2008). These techniques rely on analogies to natural processes; some of them have been inspired by biological mechanisms of evolution. The first ideas were developed in the 1960s by Holland (1961) and Fogel (Fogel and Owens, 1966), and have already reached a stage of some maturity (Michalewicz, 1996).

In recent years, numerous algorithms taking inspiration from nature have also been proposed to handle *continuous* optimization problems: real-coded genetic algorithms using some specific operators, evolution strategies using Gaussian mutations with adaptive or self-adaptive update strategies, and differential evolution, to name a few.

As real-world optimization problems may involve objectives, constraints and parameters, which constantly change with time, *dynamic* consideration using evolutionary computation methods have also raised a lot of interest within the last few years. For these dynamic and uncertain optimization problems the objective of the evolutionary algorithm is no longer to simply locate the global optimum solution, but to continuously track the optimum in dynamic environments, or to find a robust solution that operates optimally in the presence of uncertainties.

[1] A metaheuristic is a heuristic method for solving a very general class of computational problems by combining user-given black-box procedures — usually heuristics themselves — in, it is hoped, an efficient way. The name combines the Greek prefix "meta" ("beyond", here in the sense of "higher level") and "heuristic" (from ευρισκειν, heuriskein, "to find").

Optimization is a procedure of finding and comparing feasible solutions until no better solution can be found. Solutions, which in our case are healthy menus, are termed as follows:

- good or bad in terms of multiple conflicting *objectives*, such as: cost, quality of ingredients, aesthetic standards or other factors; and
- feasible if they satisfy all the problem *constraints* that are defined by dietary recommendations and guidelines.

While classical deterministic optimization methods can at best find one solution in one simulation run, evolutionary techniques are more efficient in finding multiple trade-off optimal solutions in a single simulation run. These solutions have a wide range of values for each objective representing the multi-dimensional *Pareto-optimal front*,[2] requiring an additional decision-making activity for choosing a single solution from the front.

3. Menu-planning model

As today's computers have few limitations, satisfiable menus can be automatically or semi-automatically generated by efficient software (SW) techniques, but only if the menu-planning process is well defined.

Mathematically, menu planning can be reduced to a *multi-dimensional* (i.e. *multi-constrained and multi-objective*) *knapsack problem* (*MDKP*), which is a widely studied CO problem that has many direct counterparts (Garey and Johnson, 1979).

3.1. Formulation of the MDKP

Given foods of different values and volumes, the MDKP is to find the most valuable combination of foods that fits in a knapsack of fixed volumes. *Values* are defined subjectively with respect to food quality, cost and aesthetic parameters (comprising taste, consistency, color, temperature, shape and method of preparation). Knapsack *volumes* are defined by dietary recommendations and guidelines.

3.2. Complexity of the MDKP

MDKP is easy to formulate, yet its decision problem is *NP-complete*[3] (Garey and Johnson, 1979). In complexity theory, the NP-complete problems are the most difficult problems, which cannot be solved by exact SW techniques in deterministic

[2] For a given system, the *pareto-optimal front* is a set of non-dominated solutions, i.e. solutions that cannot be improved upon without hurting at least one of the objectives. The term is named after Vilfredo Pareto, an Italian economist.

[3] In computational complexity theory, the complexity class *NP-complete* (standing for nondeterministic polynomial time) is a class of problems having two properties:

- Any given solution to the problem can be *verified* quickly (in polynomial time); the set of problems with this property is called NP.
- If the problem can be *solved* quickly (in polynomial time), then so can every problem in NP.

Although any given solution to such a problem can be verified quickly, there is no known efficient way to locate a solution in the first place. Indeed, the most notable characteristic of NP-complete problems is that no fast solution to them is known. That is, the time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows. As a result, the time required to solve even moderately large versions of many of these problems easily reaches into billions or trillions of years, using any amount of computing power available today. As a consequence, determining whether or not it is possible to solve these problems quickly is one of the principal unsolved problems in computer science today.