

Dietary Menu Planning Using an Evolutionary Method

Barbara Koroušič Seljak

Computer Systems Department, Jožef Stefan Institute, Ljubljana
E-pošta: barbara.korousic@ijs.si

Abstract. In this paper we introduce a multi-objective evolutionary algorithm for multi-level constrained dietary menu planning that quickly finds a diverse set of feasible solutions – nutritionally and gastronomically adequate menus – with the lowest objective function values without examining all the possibilities.

Keywords: genetic algorithms, multi-objective and multi-constrained optimization, multi-dimensional knapsack problem, nutrition

Načrtovanje jedilnikov z evolucijsko metodo

Razširjeni povzetek. V članku predstavljamo računalniško metodo za načrtovanje prehrane. Dandanes poznamo vrsto priporočil in smernic za optimalno prehrano, ki pa jih je zelo težko vpeljati v prakso. Razlogov je več, omenimo kompleksnost priporočil in manjšo zdravstveno pismenost, ki lahko povzroči tudi nepravilno razumevanje zdravstvenih nasvetov. Zato je računalniška metoda, ki omogoča prilagajanje navadnih jedilnikov potrjenim znanstvenim dognanjem, dobrodošla.

Uporabili smo znani in v praksi potrjeni genetski algoritem NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm) avtorja Deb [4] in ga uporabili v večnivojski različici [5]. Osnovna zamisel pristopa je v sočasnem oblikovanju jedilnikov za obroke, dnevne jedilnike in večdnevne (npr. tedenske) jedilnike. Zadostiti moramo več merilom, kot so cena, sezonska kakovost in funkcionalnost živil ter raznolikost, izražena z okusom, konsistenco, barvo, temperaturo, obliko in načinom priprave. Ta merila so lahko tudi konfliktna. Obravnavamo jih enakovredno; ustrezno rešitev izberemo iz množice optimalnih rešitev Pareto po optimizaciji, glede na trenutne potrebe. Jedilniki morajo upoštevati omejitve, ki so različne za obroke, dnevne in večdnevne jedilnike.

Problem načrtovanja jedilnikov smo prevedli v klasični optimizacijski problem polnjenja nahrbtnika, ki je v našem primeru večdimenzionalen. Ker v vsakdanjem življenju naletimo na vrsto problemov, ki jih lahko prevedemo v problem nahrbtnika, so rezultati dela uporabni tudi na drugih področjih.

Metodo smo uporabili v spletni aplikaciji za načrtovanje prehrane [1].

Keywords: genetski algoritmi, večkriterijska optimizacija, večdimenzionalen problem nahrbtnika, prehrana

1 Introduction

In this paper we propose a computer-aided method for planning weekly menus considering the diet-planning principles and the aesthetic standards for taste, consistency, color, temperature, shape, and method of preparation. We applied this method within a Web-based application for monitoring and planning person's dietary needs and requirements [1].

We tackle the menu-planning problem by the evolutionary computation. Using the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) [4] in a multi-level way [5], we generate dietary menus taking into account several constraints of personal preferences and nutritional requirements as well as nine objectives of low cost, high seasonal quality and functionality, and low deviations from uniformly distributed aesthetic standards for taste, consistency, color, temperature, shape, and method of preparation. We solve the problem from two points of view: selection of food items to be included in the solution and scheduling their occurrence within and among days, according to the frequency criteria.

2 The Menu Planning Problem

Mathematically, menu planning reduces to a multi-objective and multi-constrained (multi-dimensional) knapsack problem (MDKP) that is easy to formulate, yet its decision problem is NP-complete. It means that only by using a heuristic optimization method a solution can be found quickly (in a polynomial time).

We define the problem as follows: *Given food items of different values and volumes, find the most valuable composition that fits in a knapsack of fixed volumes. Values are defined subjectively with respect to food functionality, seasonal availability, cost, taste, consistency, color, temperature, shape and method of preparation. Knapsack volumes are defined by the weakly correlated diet-planning principles.*

Food items are selected from a nutritional database that integrates nutritional data of more than 7000 (national and world-wide) foods. We consider the D-A-CH diet-planning principles established by the European nutrition societies.

Many other real-world problems can be formulated as a MDKP, for example, the capital budgeting problem, allocating processors in a distributed computer system, project selection, and cutting stock problem.

Multi-dimensional Knapsack Problem

We are given a knapsack of m volumes

$C_k, k = 1, 2, \dots, m$, and n food items. Each item i has nine values $v_{ik} \in I^+, v_{ik} > 0, k = 1, 2, \dots, 9$, and m volumes

$\omega_{ik} \in R^+, \omega_{ik} > 0, k = 1, 2, \dots, m$, one for each capacity.

We are looking for a composition of t items, $t \leq n$, such

that $\sum_{i=1}^t \omega_{ik} x_i \Phi C_k$ (Φ can be \leq or $\geq, k = 1, 2, \dots, m$, $t \leq n$), and for which the total values

$\sum_{i=1}^n v_{ik} x_i, k = 1, 2$ are maximized, while

$\sum_{i=1}^n v_{ik} x_i, k = 3$ and

$$\left(\sum_{j=1}^{n_{a_l}} \sum_{i=1}^n h_{lj}(x_i) - \frac{\sum_{i=1}^n h(x_i)}{n_{a_l}} \right) - \sum_{i=1}^n h(x_i), l = 4, 5, \dots, 9,$$

are minimized, where n_{a_l} is the number of possible states of an aesthetic standard l . The functions used in the above objective function are defined as follows:

$$h_{lj}(x_i) = \begin{cases} 0, & \text{if } x_i = 0 \\ 1, & \text{if } x_i > 0 \wedge v_{il} = j \end{cases}, \quad h(x_i) = \begin{cases} 0, & \text{if } x_i = 0 \\ 1, & \text{otherwise} \end{cases},$$

$i = 1, 2, \dots, n, l = 4, 5, \dots, 9$. The parameter $x_i \in [0, 25P_i, 4P_i]$ denotes the quantity of the selected item i expressed in a unit (gram, milligram, μ gram, millilitre etc.). Its value is limited by the fractions of the item portion size P_i .

Methods for Solving Multi-dimensional Knapsack Problems

The only two exact algorithms that deliver optimum solutions to multi-dimensional knapsack problems in pseudo-polynomial time are based on the branch-and-bound and the dynamic programming approaches. On the other hand, heuristic methods with time complexity bounded by a polynomial in the size parameters of the problem have been known for many decades [9]. A comprehensive review of the multi-constrained 0-1

knapsack problem and the associated heuristic algorithms is given by Chu and Beasley [6]. Some of the ideas are also applicable to non-0-1 MDKPs.

3 The Menu Planning Method

In our case, a knapsack denotes a weekly menu that is composed of seven consecutive daily menus. By default, each daily menu includes five different meals, i.e., a breakfast, morning snack, lunch, afternoon snack, and dinner. However, this composition does not bias the method and can be modified to suit a specific menu-planning problem.

As the menu planning is a multi-dimensional problem with many infeasible solutions (that violate at least one constraint), we decided to solve it by using evolutionary algorithms that have been proved to be well suited for solving problems characterized by local minima. Although evolutionary algorithms search through an arbitrary search space by random decisions, they are far from random search routines. Modeling the random decisions using the Markov's chain analysis, it was shown that evolutionary algorithms can converge to globally optimum solutions [7].

An Evolutionary Algorithm for MDKP of Menu-planning

We applied an evolutionary algorithm NSGA-II in a multi-level way. Namely, the problem of weekly-menu planning is logically composed of several smaller sub-problems, one for each daily menu, whose constraints differ from those of the weekly menu. Then, optimization of daily menus is coordinated in order to obtain the overall weekly menu. Further, each daily-menu planning sub-problem is decomposed into several sub-problems of composing courses into meals. The overall problem and each of the sub-problems are solved by a NSGA-II.

The main idea behind the multi-level method is to optimize each sub-problem independently using a NSGA-II with the aim to find the overall Pareto-optimal solutions to the problem (i.e., solutions that cannot be improved upon without hurting at least one of the objectives) using the global NSGA-II.

Encoding

We encode candidate solutions of the weekly menu-planning problem and its sub-problems by real-valued coding. In our presentation, a chromosome at the highest level contains seven data carrying the information about the daily menus. At the next level, a chromosome contains five data, one for each meal. They carry the information about the meals. At the deepest level, a chromosome is formed of a number of pairs $(code_i, x_i)$, where $code_i$ denotes the code of the food item i and x_i its quantity expressed in grams. The number of pairs depends on the meal type, and by

default varies between 1 and 10 depending on the number of courses (dishes) of the meal. Meal courses may include hors d'oeuvre (appetizer) or soup, main dish, side dish, vegetables, salad, bread, dessert, cheese, fruit and drink.

Considering approximately 7000 food items from ten groups (one for each course type) and three courses per meal, the number of possible solutions is huge.

Populations

In our implementation, the global evolutionary algorithm (NSGA-II) starts the evolution from a global population of either random candidate solutions or solutions known from experience. The global population's size is N and remains constant over all generations. Each sub-problem at the next two levels is solved by a local NSGA-II and operates on its own population of the same size N . Initially, the second-level and the third-level local populations are filled with the candidate solutions from the global population and the second-level local populations, respectively.

Beside the global population, we use an additional global pool of candidate solutions that has a function of an archive of the union of solutions generated by the sub-problems (Fig. 1). The maximum size of the global pool is seven times the size of the population size N . It includes all feasible Pareto-optimal sub-solutions. At the second level, we use seven local pools of the maximum size $5N$. Their function is equal to the global pool function. Initially, the global and the local pools are empty.

Fitness evaluation

In each generation, fitness of the (global or local) population is evaluated using the following objective functions:

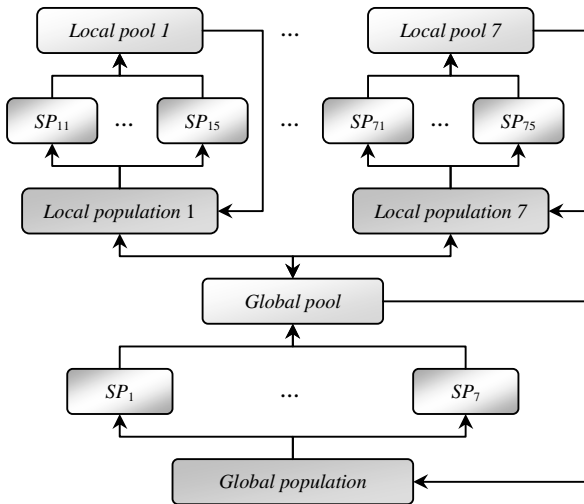


Figure 1. Scheme of the evolutionary algorithm (SP_{*i*} denotes a sub-problem *i* that is solved by a local NSGA-II)

$$\begin{aligned}
 f_k(\bar{x}) &= \frac{1}{\sum_{i=1}^n v_{ik} x_i}, \quad k = 1, 2, f_3(\bar{x}) = \sum_{i=1}^n v_{i3} x_i, \\
 f_l(\bar{x}) &= \left(\sum_{j=1}^{n_{a_l}} \left| \sum_{i=1}^n h_{lj}(x_i) - \frac{\sum_{i=1}^n h(x_i)}{n_{a_l}} \right| \right) - \sum_{i=1}^n h(x_i), \\
 h_{lj}(x_i) &= \begin{cases} 0, & \text{if } x_i = 0 \\ 1, & \text{if } x_i > 0 \wedge v_{il} = j \end{cases}, \\
 h(x_i) &= \begin{cases} 0, & \text{if } x_i = 0 \\ 1, & \text{otherwise} \end{cases}, \quad i = 1, 2, \dots, n, \quad l = 4, 5, \dots, 9,
 \end{aligned} \tag{1}$$

where v_{i1} denotes functionality of the food item i , v_{i2} its quality in the season, v_{i3} the cost, v_{i4} the taste, v_{i5} the consistency, v_{i6} the color, v_{i7} the temperature, v_{i8} the shape, v_{i9} the method of preparation, and n_{a_l} the number of possibilities for the l^{th} aesthetic standard. The aim of the global and the local evolutionary algorithms is to minimize the objective functions of (1).

Infeasible Solutions

A candidate solution may be highly fit but infeasible if it violates at least one problem constraint. At the deepest level, the constraints for meals are the least restrictive:

- Each food item can be selected in a quantity that is limited by its original portion size:

$$g_1(\bar{x}) = x_i \geq 0.25P_i, \quad g_2(\bar{x}) = x_i \leq 2P_i, \tag{2}$$

- The energy provided by the meal has to be within the lower limit and the upper limit:

$$g_3(\bar{x}) = \sum_{i=1}^{N_C} \omega_{iE} x_i \geq 0.9E, \quad g_4(\bar{x}) = \sum_{i=1}^{N_C} \omega_{iE} x_i \leq 1.1E, \tag{3}$$

where ω_{iE} denotes the energy of 100 grams of the food item i , x_i the quantity of the item i expressed in grams, and E the meal requirement for energy.

- The basic nutrients (i.e., proteins, lipids and carbohydrates) need to be balanced:

$$\begin{aligned}
 g_5(\bar{x}) &= \sum_{i=1}^{N_C} \omega_{iP} 4x_i \geq 0.1E, \quad g_6(\bar{x}) = \sum_{i=1}^{N_C} \omega_{iP} 4x_i \leq 0.15E, \\
 g_7(\bar{x}) &= \sum_{i=1}^{N_C} \omega_{iL} 9x_i \geq 0.15E, \quad g_8(\bar{x}) = \sum_{i=1}^{N_C} \omega_{iL} 9x_i \leq 0.3E, \\
 g_9(\bar{x}) &= \sum_{i=1}^{N_C} \omega_{iC} 4x_i \geq 0.55E, \quad g_{10}(\bar{x}) = \sum_{i=1}^{N_C} \omega_{iC} 4x_i \leq 0.75E,
 \end{aligned} \tag{4}$$

where ω_{iP} , ω_{iL} , ω_{iC} denote the quantity of proteins, lipids and carbohydrates, respectively, in 100 grams of the food item i , and N_C is the number of courses in the meal. Because the quantities are expressed in grams, conversion factors (4 for

proteins and carbohydrates, and 9 for lipids) are required to attain to calories. We applied the usual balancing factors for adults (0.1 and 0.15 for proteins, 0.15 and 0.3 for lipids, and 0.55 and 0.75 for carbohydrates) but may be changed.

At the upper level, there are additional constraints that need to be satisfied by a feasible chromosome presenting a daily menu:

- Simple sugars should account for only 10 percent or less of the daily total energy intake:

$$g_{11}(\vec{x}) = \sum_{i=1}^{N_c} \omega_{iS} 4x_i \leq 0.1E_d, \quad (5)$$

where E_d denotes the daily requirement of energy.

- The daily intake of saturated fatty acids should be limited to 10 percent of the daily total energy intake:

$$g_{12}(\vec{x}) = \sum_{i=1}^{N_c} \omega_{iF} 9x_i \leq 0.1E_d. \quad (6)$$

- The recommended daily intake of the dietary fiber is 10 grams per 1000-kalorie energy intake and should not exceed 40 grams:

$$g_{13}(\vec{x}) = \sum_{i=1}^{N_c} \omega_{iV} x_i \geq 10 \frac{E_d}{1000}, g_{14}(\vec{x}) = \sum_{i=1}^{N_c} \omega_{iV} x_i \leq 40 \quad (7)$$

- The minimum and the maximum sodium requirements for adults in Slovenia are set at 550 and 2400 milligrams per day, respectively:

$$g_{15}(\vec{x}) = \sum_{i=1}^{N_c} \omega_{iNa} x_i \geq 500, g_{16}(\vec{x}) = \sum_{i=1}^{N_c} \omega_{iNa} x_i \leq 2400 \quad (8)$$

At the highest level, beside the meal and the daily-menu constraints, a chromosome presenting a weekly menu has to satisfy all the remaining constraints for nutrients, such as cholesterol, monounsaturated fatty acids, omega-3 and omega-6 polyunsaturated fatty acids, trans-fatty acids, water-soluble and fat-soluble vitamins, water, major minerals, and trace minerals, to become a feasible solution. Formal definitions of these constraints are similar to those of equations (3) or (7), but are beyond the scope of this paper.

We decided to repair a certain part of infeasible solutions in each generation to speed up the procedure of finding an optimal solution. It was shown that repair schemes, such as the Lamarckian and the Baldwinian greedy repair, outperform a penalty function approach [8]:

- At the deepest level of the evolutionary algorithm, we apply a local optimization procedure of linear programming in order to convert infeasible solutions into feasible ones. This procedure based on the simplex method modifies the quantities of selected infeasible chromosome food items to satisfy the problem constraints.
- At the upper levels, we also try to repair infeasible

solutions by ‘replacing’ certain critical meals with more appropriate ones. We apply the Baldwinian repair, where replacement is used only to evaluate the fitness values of each solution [8]. Critical meals are those that do not satisfy the constraints on major food groups (i.e., breads, cereal, rice, and pasta / vegetables / fruits / milk, yogurt, and cheese / meat, poultry, fish, beans, eggs, and nuts / fats, oils, and sweets). Namely, a daily menu has to be composed of a certain number of foods from each major food group. A weekly menu has to include a diverse set of foods from the major food groups. There may be limitations on frequency of red meat, fish, potatoe etc.

Selection

In order to form a new population at whichever level of the evolutionary algorithm, a binary tournament approach is applied. Solutions from both – the parent and the previous offspring – populations can take part in the tournament if they are sorted by two attributes, i.e., a non-domination rank and a crowding distance. Initially, the offspring population is an empty set.

First, solutions are sorted by the fast non-dominated sorting approach of the NSGA-II. In this approach, the best non-dominated solutions become elites of identical importance, forming Pareto-optimal fronts. Solutions are non-dominated if none solution is better than the others with respect to all equally important objectives. Because every solution from the population is checked with a partially filled population for domination, the maximum time complexity of the non-dominated sorting approach is $O(4mN^2)$, where $2N$ is the population size and m the number of objectives.

Then, solutions are sorted according to their crowding distances. A crowding distance is a measure of the search space around a chosen solution, which is not occupied by any other solution in the population. Its computation requires sorting of the populations according to each objective function value in their ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with the smallest and the largest function values) are assigned an infinite distance value. All other solutions are assigned a distance value equal to the absolute difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding distance value is calculated as the sum of individual distance values corresponding to each objective. The maximum time complexity of this sorting approach is $O(2mN \log 2N)$.

A solution i wins a tournament with another solution j if both solutions are feasible or infeasible and any of the following conditions is true:

- It has a better non-domination rank than solution j .
- Having the same non-domination rank, it has better

crowding distance than solution j .

The first condition makes sure that solution i lies on a better Pareto front than solution j . The second condition resolves the tie of both solutions being on the same non-dominated front by deciding on their crowded distance. The one residing in less crowded area wins. If one solution is feasible and the other is not, the feasible one wins the tournament.

Performing N tournaments, we obtain a new parent population of size N . Other N solutions from the least important Pareto fronts having a smaller crowding distance are discarded.

Crossover and Mutation

Solutions from the new parent population are mated pair-wise (using a two-point crossover operator) and mutated to create a new offspring population of size N . This completes one NSGA-II iteration.

Mutation is performed on randomly selected elements of the chromosome. The mutation rate is set to be a small value that linearly decreases with iterations. The selected elements are mutated in one of the following ways chosen with respect to the type of the chromosome:

- by replacing the food item or the dish with a food from the same major food group or a dish from the same course food group, or
- by replacing the meal with a meal of the same type.

Termination Criteria

Once a sub-problem (meal planning or daily-menu planning) is solved by a local NSGA-II (using a wanted-solution approach or a time-out approach), its solutions are unified with the solutions generated by other sub-problems at the same level and saved in their local pool. To obtain chromosomes at the upper level, these solutions need to be completed using the rest of the chromosome sequence from the population at the upper level. Then, solutions from each local pool are sorted by the non-dominated and the crowding-distance sorting approaches to obtain locally optimal solutions forming new local populations. Finally, completed solutions from these local populations are unified and saved into a global pool. A selection of optimal solutions (non-dominated solutions with a large crowding distance) from the global pool is transferred to the global population terminating the overall procedure.

4 Example

As a demonstration, we applied the multi-level NSGA-II to a problem of planning weekly menus for people without specific dietary requirements in a local hospital.

In Tab. 1, we list the parameters used to generate meals, daily menus and weekly menus by the multi-level NSGA-II. We ran the algorithm for 25 times to obtain the experimental results presented in Tab. 2. In

Fig. 2, a part of the feasible search space, whose shape is depicted for three objectives but actually modified by nine objectives, is presented. Tab. 3 gives a subset of the analysis results of a weekly menu generated by the multi-level NSGA-II. This weekly menu was generated with respect to the following requirements for the major food group of meat and its substitutes: white meat, legumes, fish and eggs once per week, and red meat three times per week.

Table 1. NSGA-II parameters for the menu-planning problem

Parameter	Weekly-menu level	Daily-menu level	Meal level
Chromosomes length	7	5	10
Population size	100		
Pool size	700	500	-
Crossover probability	0.7		
Mutation probability	0.14-0.01	0.2-0.01	0.1-0.01
Selection type	Binary tournament selection		
Crossover type	Two-point crossover		
Mutation type	Linear descending mutation		
No. of iterations	95	70	135

5 Summary

We developed a dietary menu planning tool [1] that incorporates the strength of meta-heuristic evolutionary algorithms. Its function is to select a set of foods and schedule them so that diet-planning principles and aesthetic standards are satisfied. We applied the NSGA-II in a multi-level way to solve the weekly-menu problem, which is logically decomposed in several sub-problems, namely, daily-menu and meal planning. The algorithm finds the Pareto-optimal set of diverse optimal solutions that are trade-offs between high seasonal quality and functionality, and low cost and deviations from the aesthetic standards in a reasonable amount of time. We maintain the feasibility of solutions by repairing infeasible solutions in two ways, namely, by the LP simplex method (for meals) and the Baldwinian greedy repair method (for daily and weekly menus). The experimental results showed that the approach distinguishes itself by efficiency and effectiveness.

As the problem of dietary menu-planning belongs to the multi-dimensional knapsack problems, the method could be useful for other intractable problems from this group.

Table 2. Experimental results of the evolutionary algorithm

Percentage of infeasible solutions in each new generation	89		
Percentage of successfully repaired infeasible solutions	65		
	Cost (€)	Quality in season	Functionality
Best result	3.08	18	0
Median	9.7	28	6
Worst result	22.8	48	12
Mean value	9.7	28.3	5.8
Standard deviation	3.1	4.7	3.4

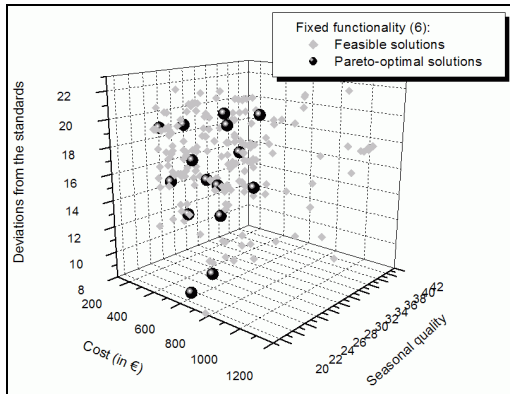


Figure 2. Part of the problem search space

6 Acknowledgement

This work presents results of a joint research made by the Computer Systems Department of the Jožef Stefan Institute and the Department of Food Science and Technology of the Biotechnical Faculty, University of Ljubljana. The author gratefully acknowledges the contributions of Prof. Dr. Dražigost Pokorn and the support provided by the Slovenian Ministry of Health during the progress of this work.

Table 3. Analysis results of a low-cost and high-quality appetizing weekly menu generated by computer

	Mean daily values	DACH Recommended Dietary Allowances	Goal achieved
Energy (kcal)	2036	2000	102 %
Proteins	16 %	10-15 %	✓
Lipid	28 %	15-30 %	✓
Carbohydrates	56 %	55-75 %	✓
Simple sugars	4.5 %	< 10 %	✓
Saturated fats	6.6 %	< 10 %	✓
omega-6:omega-3	4.9:1	5:1	✓
Dietary fibre (g)	33.6	30-40	✓
Cholesterol (mg)	160	300	✓
Sodium (mg)	2500	550-2400	104 %
Starch foods	9.2	11	84 %
Vegetables	4.7	5	94 %
Fruits	3	3	100 %
Milk	2	2	100 %
Meat&substitutes	1.9	2	95 %

7 References

[1] Web application (in Slovene): <http://optijed.ijs.si>

[2] E. F. Eckstein, *Menu Planning*, Third Edition, AVI Publishing Company, Inc., Westport, Connecticut: 1983.

[3] C. R. Marling, G. J. Petot and L. S. Sterling, "Integrating Case-Based and Rule-Based

Reasoning to Meet Multiple Design Constraints", *Computational Intelligence*, Vol. 15, No. 3, 1999.

[4] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd.: 2001, ISBN 0-471-87339-X.

[5] S. Gunawan, A. Farhang-Mehr, and S. Azarm, "Multi-level Multi-objective Genetic Algorithm Using Entropy to Preserve Diversity", In *C. M. Fonseca et al (Eds.): EMO 2003, Lecture Notes in Computer Science 2632*, Springer-Verlag: 2003, pp. 148-161.

[6] P. C. Chu and J. E. Beasley, "A Genetic Algorithm for the Multidimensional Knapsack Problem", *Journal of heuristics*, 4, Kluwer Academic Publishers: 1988, pp. 63-86.

[7] C. L. Karr, I. Yakushin and K. Nicolosi, "Solving inverse initial-value, boundary-value problems via genetic algorithm", *Engineering Applications of Artificial Intelligence*, 13(6), 2000, pp. 625-633.

[8] H. Ishibuchi, S. Kaige and K. Narukawa, "Comparison between Lamarckian and Baldwinian Repair on Multiobjective 0/1 Knapsack Problems", In *C. A. Coello Coello et al (Eds.): EMO 2005, Lecture Notes in Computer Science 3410*, Springer-Verlag Berlin Heidelberg: 2005, pp. 370-385.

[9] P. Korošec, J. Šilc, B. Robič, "Population-based methods as a form of metaheuristic combinatorial optimization", *Electrotechnical Review*, 72(4):214-219, 2005.

Barbara Koroušić Seljak received her Ph. D. degree in Computer Science and Informatics from the University of Ljubljana, Slovenia, in 1997. Currently, she works as a researcher at the Computer Systems Department of the "Jožef Stefan" Institute, Ljubljana. Her research focuses on design of real-time and embedded systems. She is also interested in modern heuristic methods used in solving practical optimization problems.