

Development of a Genetic Algorithms Optimization Algorithm for a Nutritional Guidance Application

Petri Heinonen¹ Esko K. Juuso²

¹Nutri-Flow Oy, Finland, petri.heinonen@nutri-flow.fi

²Control Engineering, Faculty of Technology, University of Oulu, Finland, esko.juuso@oulu.fi

Abstract

Personalized easy to follow nutritional guidance is getting more important, since lifestyle related health problems are increasing. To gain a healthy balanced diet usually requires knowledge of a licensed nutritionist. There is a Fuzzy Expert System (FES) which applies knowledge of nutritionists, health data of an individual, personalized nutritional recommendation, and a meal diary with food composition data to balance a diet. FES generates a set of foods and beverages which should be altered in the diet with information on the direction and importance of the change. This paper presents a selection and a development of an optimization algorithm to be integrated with FES to provide easy to follow nutritional guidance. The selection process is carried out as a literature review. The development of selected Genetic Algorithms (GA) is carried out as an integrated part of Nutritional Guidance application, Nutri-Flow[®], since FES generates the search space, and is an important part of a Fitness Function of the optimization algorithm. The selection of the design parameters, are described and the test results are presented. Validation of the overall model is carried out with an expert analysis and comparison of the nutrient intake from the initial diet and recommended diet.

Keywords: genetic algorithms, optimization, nutritional guidance

1 Introduction

Personal dietary guidance is an important tool for achieving global and national targets of the battle on non-communicable diseases caused by lifestyle habits on diet and physical exercise (Heinonen, 2009). Micronutrient malnutrition due to eating habits is getting common while non-nutrient dense energy rich foods are getting common in diets (IFPRI, 2016).

The Internet is filled with calorie calculators and other similar applications which calculates energy and nutrient intake levels according to a meal diary. Average consumers cannot balance their diets by knowing which micronutrients should be added to balance the diet. Traditionally, a licensed nutritionist is needed to convert the nutrient level information into foodstuff level information as a meal plan. (Heinonen, 2009)

Nutri-Flow[®] Software gives personalized dietary guidance on the foodstuff level as foods and beverages by applying a national food composition database, national nutritional recommendations with personal health data and eating habits. Fuzzy logic handles the uncertainty and imprecise values, and the mapping from the nutrient level to foodstuff level is carried out with Fuzzy Expert System (FES), which contains licensed nutritionists' knowledge in a rule base. (Heinonen et al., 2009)

Optimization in Nutri-Flow[®] Software is needed to find a level of change for a set of foodstuffs to reach a more balanced diet. A type of an optimization problem usually defines which optimization methods are applicable within the problem domain. (Heinonen, 2009)

Direct search is one approach to solve optimization problems which have non-differentiable or discontinuous objective functions. Most traditional optimization methods require gradient or higher derivatives from the objective function to work properly. (Kolda et al., 2003)

Heuristic search is a set of optimization methods with rules to guide the optimization process towards to the global optimum. Genetic Algorithms (GAs) belongs to the heuristic search group. The theory was invented in 1975 and GA implements the concept from Darwinian principle of natural selection (Darwin, 1859). The terminology in GA is closely adapted from natural genetics (Holland, 1975). A solution is called a chromosome, which has locus bind variables. The solution can be coded in a binary form, but with many real world problems high precision makes chromosomes very long, and it gets inefficient. (Davis, 1991)

A set of solutions or a population is put in a competitive environment where each chromosome has a fitness value evaluated by an objective function (Holland, 1975), (DeJong, 1993). In every iteration round, GA operators are applied to the population. A crossover operator combines genetic materials of selected chromosomes creating a new population. Mutation operation makes random variation in the population to prevent the optimization process from stopping at local minimums. (Holland, 1975) Since the theory was published, several GA operators have been

developed with variations according to the current optimization problem (Goldberg, 1989a)

This paper presents the GA optimization model developed for the Nutri-Flow® Software.

2 Selecting an Optimization Method

The type of optimization problem is the key to selecting an optimization method. In this study, the optimization problem has a large amount of variables and there are several solutions which are almost equally good.

To form a mathematical function in this problem domain is challenging and time consuming. It is possible to calculate gradients at certain points, however the system is not continuous due to its nature. Based on this, traditional optimization methods are not selected for testing.

Direct search can be used with an optimization problem domain where is no gradient or higher derivative available. It works also with non-continuous problem space. (Kolda et al., 2003) In the optimization problem of this study, it might be possible to use direct search to find a single solution. This needs further testing.

A real-coded GA has solutions available as a set of comparable solutions. After evolving a population - the set of solutions - the set of the best options is available in the solution domain already without a mapping function. (Man et al., 1999)

The stochastic nature of GA with crossover, mutation, and elitism operators the optimization process is not easily stopped at local minimums. It is argued that real-coded GA does not always reach the best result. However, the real-coded GA is widely used with real word optimization problems. One example of this is represented in (Le and Kim, 2011).

There are also other Evolutionary Programming algorithms, e.g. differential evolution, which could be used in the optimization problem domain of this study.

3 Genetic Algorithms

Genetic algorithms are based on processing of a population of coded solution alternatives.

3.1 Chromosome Coding

A chromosome in GA represents a possible solution for the optimization problem. The size of search domain and accuracy level of result are used when evaluating if the coding should be done in the binary or real-value domain. (Goldberg, 1989a)

Real-value coding is used widely with practical real world optimization where the search domain is usually large and required high accuracy, where the binary coding would be inefficient with extremely long chromosomes. The solution domain is applied in real-value coding thus no result mapping is needed. It has

been argued that the real-value coding does not always yield good results. (Man et al., 1999)

3.2 Population

A population is a term for a set of chromosomes. The best solution should be found by evolving the population by applying GA operators. In this process, the size of the population has an effect on convergence speed and reaching the global optimum. The population size can be fixed or it can vary throughout the optimization process. (Goldberg, 1989b)

Too small a population does not have enough diversity to evolve towards the global optimum. Longer the chromosome, bigger the population should be. (Goldberg, 1989b) When the population size is too big, the evolving needs more iteration rounds to reach the best solution. This affects the computing time with a slow convergence rate. (Affenzeller et al., 2007) There are statistical methods for generating the initial population, however it is usually generated randomly (Reeves and Rowe, 2002).

3.3 Crossover

Crossover operators evolve population towards better solutions by distributing good genetic matter between generations. Crossover starts with selecting parents by using a selection method, usually by the roulette wheel selection method (Sorsa et al., 2008) or by the tournament selection method (Goldberg, 1990).

Good genetic material is found into mating pool by selection methods since probability for selecting the fittest parents from the population is higher than for the worse ones. This is the basis how better solutions are found on every iteration round. (Sorsa et al., 2008)

The tournament selection method is based on randomly selected chromosomes from the population, and the chromosome with the best fitness value is selected. The tournament size k defines how many chromosomes will be selected from the population. Typically, value for it is 2. (Goldberg, 1990)

After a mating population is formed by a selection method, a crossover operator is applied. A design parameter for the crossover operator is crossover probability p_c , which determines if the current parent chromosomes are combined with the crossover operator or if they are moved directly to the offspring population. (Man et al., 1999)

With real-value coded chromosomes, uniform and non-uniform crossover operators can be applied. Where uniform operators act in the similar way in every generation, and non-uniform operators work depending on the age of the population. Two offspring from two parents are formed by an arithmetic operator as follows:

$$y_i^1 = \alpha_i x_i^1 + (1 - \alpha_i) x_i^2, \quad (1)$$

$$y_i^1 = \alpha_i x_i^2 + (1 - \alpha_i) x_i^1,$$

where α_i are uniform random numbers. α can vary in non-uniform crossovers, and is constant in uniform crossovers. (Sorsa et al., 2008). New crossover operators are developed actively (Gegúndez et al., 2007).

3.4 Mutation

Mutation is a GA operator which creates a random variation in the population. It maintains the population diversity by generating new genetic material. With correct design parameters, mutation prevents GA from stopping at local minimums. (Sorsa et al., 2008)

Uniform and non-uniform mutation operators can be used with real-coded chromosomes. The uniform mutation operator can be applied for each gene in a chromosome using the same mutation probability where initial population is $x_i^t = \langle v_1, \dots, v_n \rangle$, after a mutation operation with $1 \leq n$ it becomes $x_m^t = \langle v_1, \dots, v'_k, \dots, v_n \rangle$. The random value for v'_k is in a feasible range for the locus k . (Michalewicz, 1996)

One of the most commonly used non-uniform mutation operator is the Michalewicz's non-uniform mutation (Michalewicz, 1996), where the mutated element v'_k is calculated by

$$v'_k = \begin{cases} v_k + \Delta(t, u_k - v_k), & \text{if a random digit is 0} \\ v_k + \Delta(t, v_k - l_k), & \text{if a random digit is 1} \end{cases}, \quad (2)$$

where

$$\Delta(t, y) = y \left(1 - r \left(1 - \frac{t}{T} \right)^b \right). \quad (3)$$

Mutation probability p_m controls the mutation operations. When using too big values for p_m , good genetic material can be lost and the convergence rate slows down. With very low values, there is little or no effect on the population.

3.5 Elitism

When altering the genetic pool of a population with GA operators, there is a possibility to lose the best solution of an iteration round. The elitism operator is designed to save the best chromosome and transfer it to the next generation. This is done usually by replacing the worst chromosome from the new population with the best chromosome from previous population. (Man et al., 1999)

4 Genetic Algorithms in Domain of Dietary Optimization

In this study, the GA optimization is an interconnected module in Nutri-Flow[®] Software as represented in Figure 1. Nutri-Flow[®] Software has a database where

personal dietary habits and personal health data are stored when filled in. A nutritional state of a diet is assessed and the guidance is mapped to foodstuff level, e.g. as foods and beverages. (Heinonen et al., 2009)

FES has two hierarchical levels representing main and sub-groups in the classification of foodstuffs in Nutri-Flow[®] Software database. The input consists of 30 nutrient variables and the output of FES has 129 variables, including 87 sub-groups of foodstuff classification and 42 foodstuff variables. The rule base in FES has 590 rules on the first hierarchical level, and 400 rules on the second level. (Heinonen, 2009) The expertise is introduced with the rules. Development of FES is presented more detailed in (Heinonen, 2009; Heinonen et al., 2009).

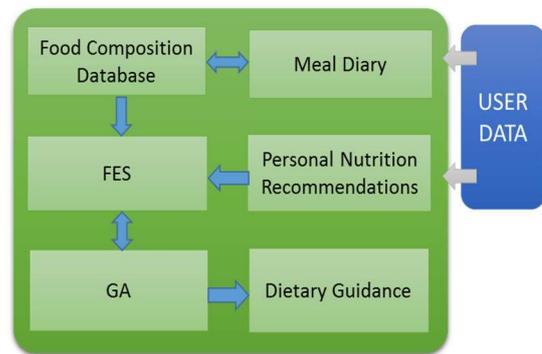


Figure 1. Schematic model of Nutri-Flow[®] software (Heinonen, 2009).

The output of FES is used to form a search space for GA. An initial population is generated randomly using the search space. A gene in a chromosome represents a foodstuff with recommended daily intake level. A chromosome represents a dietary recommendation in foods and beverages with their recommended daily intake levels.

An objective function is formed to analyze the fitness of the chromosomes. The recommendation takes into account also personal taste and allergies with the nutritional status.

A diagram of the GA optimization process is presented in Figure 2. Because the initial population is a random set, it is recommended to run GA optimization several times with a new initial population. GA in the Nutri-Flow[®] software is set to run 10 re-runs with a new initial population each time. All the best solutions of the last population of each GA re-run are stored, and the best of the best is used to form the dietary guidance. The selection of GA re-runs and other GA design parameters are described later on this paper.

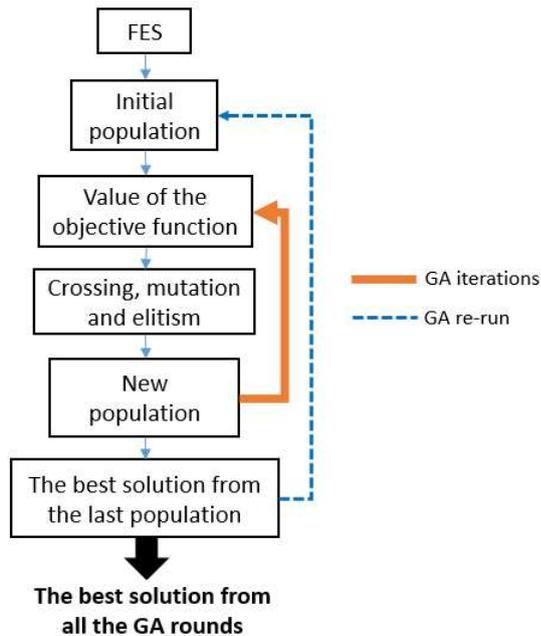


Figure 2. The GA Optimization process.

5 Developed Model

5.1 Search Space

The search space is formed according to the output of FES. The output stores the information about the foods to be added or reduced with the importance of the change. There are three different sets of foodstuffs in the output: “reduce”, “no change” and “add”.

There is a need to assess feasible ranges of daily intake levels for the foods, because the initial population is generated randomly within the given intake range for each food. The intake range for foods to be reduced is formed as $[0, i]$, where i is the initial intake level. The foods, which have no need to be changed, will be kept at the original intake level, i . Foods to be added have a range with $[i, m]$, where m represents a recommended maximum intake level, defined by licensed nutritionists. If no preset for maximum intake is present, the maximum value is evaluated from the initial intake level.

The search space has an effect on performance and calculation time of Nutri-Flow® software. If the initial population is biased due to infeasible range of intake levels of foods, the convergence of the fitness value is slow, and the global optimum might never be reached.

5.2 Objective Function

In this study, the objective is to minimize the fitness value. The objective function uses FES to determine the nutritional status of a solution. FES fuzzifies the nutrient and gives membership grades for each fuzzy membership function. The goal is to minimize “too little” and “too much” membership grades μ . The FES membership functions are represented in Figure 3.

Personal nutritional recommendations for 30 nutrients are used to tune the membership functions, data points A, B and C presented in Figure 3. The data points A, B and C represent Lower Intake level (LI), Recommended Intake level (RI), and Upper Intake level (UL), respectively (Heinonen, 2009).

There is no need to use normalizing function for each nutrient input levels since membership grades are already normalized to $[0,1]$.

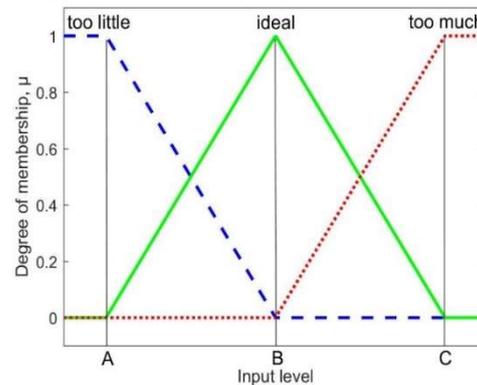


Figure 3. FES Input Membership Functions (Heinonen, 2009).

There are also other objectives to be minimized which are represented in the objective function as follows:

$$\text{MIN}(a \sum_1^n (l_n \mu_{l,n} + u_n \mu_{u,n}) + b \sum_1^m |d_m| + c \sum_1^p k_p), \quad (4)$$

where a is weighting coefficient for nutrition status
 l_n is weighting coefficient for importance of nutrient n at “too little” condition,
 $\mu_{l,n}$ is membership grade for “too little” membership function for nutrient n ,
 u_n is weighting coefficient for importance of nutrient n at “too much” condition,
 $\mu_{u,n}$ is membership grade for “too much” membership function for nutrient n ,
 b is weighting coefficient for level of change
 d_m is distance from current diet for foodstuff m
 c is weighting coefficient for importance level of other variables
 k_p is other diet related variables.

The first term represents the difference between the recommended nutrient intake level and the current intake level in the fuzzy domain. According to licensed nutritionists, some nutrients are more important in keeping close to the recommendation level than others. The weighting factors l_n and u_n are used to emphasize the importance of the nutrient according to intake level. The weighting factors are tuned by licensed nutritionists. In this study, 30 nutrients are taken into account when assessing the individual nutritional state.

Smaller steps are usually easier to follow when changing eating habits. The second term is a measure how much a chromosome would change the initial diet.

It measures an amount of foodstuffs to be changed and the magnitude of the change. The weighting factor b is used to tune the level of change, and can be altered by a user through the user interface. If the value is set to 0, the magnitude of change is not taken into account.

The third term contains other measurable variables needed in dietary guidance, such as vegetable intake level and energy intake level. Other non-nutritional variables can be used too when these are used to fine-tune the recommendation. These should not have an advert effect on the nutritional status of the diet.

5.3 Coding

It was found out already in (Heinonen, 2009) that binary coding is not a good option to perform GA in this problem and solution domain, therefore the real-valued coding was selected. With real-valued coding, a gene stores a name of a foodstuff with a daily intake level: “a slice of rye bread”, “20 g”. The length of a chromosome depends on the variation of food beverage items in the meal diary during the period to be evaluated, therefore the length of the chromosomes is not fixed.

According to licensed nutritionists, with a recommended three to seven day period of meal tracking, there are average 20 different food items. Similar foods are combined when assessing daily intake levels. To confirm this, there is a need to do statistical evaluation on the meal diary database.

5.4 Population Size

While the chromosome size is not constant, the population size can vary throughout the different optimization run. In this study, the population size is kept constant at 100.

Population size has a strong effect on computing time when the size is too big as shown in (Heinonen, 2009). The effect on varying population size can be monitored through computing time, convergence rate, and the final result.

5.5 Crossover

Arithmetic crossover and tournament selection are used in this study. The design parameters for tournament selection are crossover probability p_c and tournament population size p_s . The selected arithmetic crossover is uniform.

Different values for p_c and p_s are tested while other GA design parameters kept fixed. The performance of the overall system is monitored via test parameters such as the convergence rate of the best and average solution, and fitness of the best solution.

5.6 Mutation and Elitism

The mutation and elitism operators are applied in this study to prevent the optimization process from stopping any local minimum. Elitism is used to prevent of loss the best solution during evolving the population.

Mutation probability m_p is evaluated with different value ranges while other parameters kept fixed. The overall system is monitored with the same parameters as crossover, the convergence rate of the best and average solution, and the fitness of the best solution.

Elitism is set to keep the best solution of the current population and replace the worst solution from the offspring population. In this study, the elitism operator replaces only one chromosome in the offspring population.

5.7 Additional Parameters

Each diet and goal are different when assessing personal dietary guidance. There is no way to set a global fitness value when the optimization process should be exited. The guidance should be able to follow without too drastic changes in the diet. Therefore, there are additional parameters in the objective function to make the guidance easier to follow.

There should be a certain exit point for the optimization process, which is executed using a counter for iterations. Too small a value for the counter would stop the optimization process too early with a poor result, and a too large number would just waste computing time. The optimum value for iterations can be found when the main design parameters are tuned first. The optimization process can be stopped when the best possible solution is not changing with a certain amount of iterations, but the average fitness value still evolves. In this study, the exit point for iterations is a fixed number, but in the future, the algorithm could analyze the convergence speed of the best solution vs. average solution and stop it when evolving for the best solution is not detected.

The initial population of GA is random. There is a possibility that the solution set has very bad fitness values or has too low a variation between the solutions. This could prevent GA to find the global optimum and cause a stop at a local minimum. This can be prevented to re-run GA with new initial population as presented in Figure 2. The rate of GA re-run should be set high enough. This multiplies the amount of iterations, thus has a direct effect on the computing time. A fixed value for the re-run rate is used in Nutri-Flow® software. The testing of the effect of rate on the re-runs requires a big set of meal diaries, hence it will be done later.

5.8 Java Genetic Algorithm Package

Nutri-Flow® software is currently written in Java and it applies JGAP – Java Genetic Algorithm Package library which provides the GA operators with all necessary design parameters.

Validating Nutri-Flow® software is done with data sets run in Nutri-Flow®, and in Matlab® model. The Nutri-Flow® software is working correctly if the same input generates the same output. GA needs a special focus, since optimization results with one input could

create several different outputs. Generally, there are several different ways to alter initial diet to achieve a certain balanced diet. Validation of GA module requires a large set of test runs with the statistical and nutritional analysis of the result set.

5.9 Data Acquisition

To keep the results comparable throughout the development work of Nutri-Flow[®] software, there is a set of individuals and data of their dietary habits. The test data represents different average dietary habits from fast food diet to vegan diet. The set of individuals contains male and female persons with different health profiles from athlete to pregnant women, and slim to fat inactive persons.

The used test data is suitable for testing GA optimization performance on the current problem domain. Data acquisition is done using Nutri-Flow[®] software. All the test variables of GA optimization process are saved to a text file and all the numerical data is analyzed with Excel or Matlab[®] which also provides a good platform for testing new calculation ideas.

6 Results and Discussion

6.1 Overall Performance of Nutri-Flow[®] software

Nutri-Flow[®] software provides the output as dietary guidance. Also the performance of GA is recorded separately to provide numerical data for analysis. The overall performance of the system is analyzed by licensed nutritionists.

When a person applies the recommended actions to the initial diet, the result must lead to a better state of nutrition. This can be analyzed numerically by comparing personal recommendations with the nutritional state of the recommended diet. The comparison should be done also in the fuzzy domain, since the values include uncertainty and imprecision. The formed dietary recommendation should be also feasible with recommended foods and their portion sizes and with level of change. For example, nobody would like to eat three tablespoons of cinnamon.

Results of the overall performance are promising. All the requirements are met numerically evaluated. However some of the recommended changes in diet were controversial, since there are expectations that some foods are healthier than others. Some healthy considered foods were recommended to reduce while some other foods were recommended to increase. After a review of licensed nutritionists, also the controversial guidance has been accepted, since the healthiness of a single foodstuff depends on the individual overall diet, not on the health claims on single food.

The tests revealed that, there are cases when GA is not working. Optimization with very limited diets with strong dietary limitations did not lead to any result. The

reason is that, there was not enough variation in the population to find a feasible result. Other case was with very balanced diets. The initial diet was already so close to the recommendations that GA did not always find any better solution.

A statistical approach with a large set of individuals and meal diaries is needed to get more comprehensive data to analyze the performance of the overall system generally.

6.2 Population Size

According to previous contextual testing, the population size was set a fixed value, 100 individuals. There is a need to carry out more tests to show that selected population size is adequate for varying chromosome size. This could be done with the planned statistical testing for the overall system.

However, a population size more than 1000 will lead to longer computational time, which might affect the usability of the service. With current test data, global optimum is reached with current design parameters with all test cases.

6.3 Crossover

In this study, only uniform arithmetic crossover with the tournament selection method was analyzed. Parameters crossover probability p_c and tournament population size p_s are tested with population size 100. The best working values for the parameters are 0.8 and 2 for p_c and p_s , respectively. The test was run with the same data set as with the population test.

6.4 Mutation and Elitism

The same initial test data was applied also with testing mutation and elitism. The population size with this test was kept fixed at 100, and p_c and p_s to 0.8 and 2, respectively. Too big a value for mutation probability m_p slows the convergence rate as it affects the better genetic material more probably. The value 0.02 was selected. According to the data, the best value for m_p was 0.02.

Test runs show that running GA operators without elitism will lead us to lose some of the best solutions during the evolving of the population. This can be seen in Figure 4 where the trend line has peaks for both, the average fitness value and for the best fitness value.

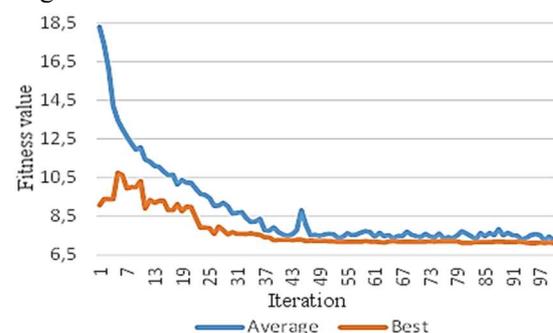


Figure 4. GA optimization without elitism operator.

6.5 Additional Parameters

According to the overall assessment with the numerical data of iterations, the best solution was reached with most of the test data within 100 iteration rounds. It is possible to create a self-monitoring feature to stop iterations when no better solutions are not found after a certain amount of iterations.

The initial population is created with random values within the given range. Sometimes it leads to a bad population which does not lead to the global optimum. GA should be run several times to find the global optimum. In Nutri-Flow[®] Software GA is run 10 times at the moment. This needs also further testing with a larger test set.

7 Conclusions

The nutritional guidance tool combines expertise with extensive food composition data through optimization. According to the overall assessment, the results are promising. All the requirements were met except two cases where no result was found at all. The real-coded chromosomes with GA operators such as crossover, mutation, and elitism can be used in the domain of dietary guidance when the search space is formed by FES. The objective function is crucial in comparing the results. When nutrient intake levels are handled in the Fuzzy domain, the imprecision and uncertainty can be taken into account, too.

The feasible result and a short computation time are essentials to make the Nutri-Flow[®] software usable. The validation of the system was carried out with expert knowledge, comparisons of nutritional status, and monitoring the key features of GA performance. There is a need to carry out more intensive testing with a large test data set. Also other optimization methods suitable for this problem domain will be tested.

References

- M. Affenzeller, S. Wagner and S. Winkler. Self-Adaptive Population Size Adjustment for Genetic Algorithms, *EUROCAST 2007, LNCS 4739*, pages 820–828, 2007. doi: 10.1007/978-3-540-75867-9_103.
- C. Darwin. *On the origins of species by means of natural selection*. Murray, London. 1859.
- L. Davis. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York. 1991. ISBN 0-442-00173-8.
- K. DeJong. *Genetic algorithms are NOT function optimizers*, Whitley LD (ed) FOGA – 2, Morgan Kaufmann, Los Altos, CA, pp 5–17, 1993.
- M. E. Gegúndez, P. Palacios and J. Álvarez. A New Self-adaptative Crossover Operator for Real-Coded Evolutionary Algorithms, *ICANNGA 2007, Part I, LNCS 4431*, pages 39–48, 2007. doi: 10.1007/978-3-540-71618-1_5.
- D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wiley Publishing Company, Massachusetts. 1989a.
- D. E. Goldberg. Sizing Populations for Serial and Parallel Genetic Algorithms, In: *Proc. 3rd International Conference on Genetic Algorithms*, pages 70–79, 1989b.
- D. Goldberg. *A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing*, Tech. Rep. Nb. 90003, Department of Engineering Mechanics, University of Alabama, Tuscaloosa, A. 1990.
- P. Heinonen. *Fuzzy Expert System and GA optimization in Dietary Guidance Application*, Master's Thesis, University of Oulu, March 2009.
- P. Heinonen, M. Mannelin, H. Iskala, A. Sorsa and E. Juuso. Development of a Fuzzy Expert System for a Nutritional Guidance Application, *Proc. Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference, Lisbon, Portugal, July 20-24, 2009*, pages 1685-1690, 2009. ISBN 978-989-95079-6-8.
- J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press. 1975.
- IFPRI. *Global Nutrition Report 2016: From Promise to Impact: Ending Malnutrition by 2030*, International Food Policy Research Institute, Washington, DC. 2016.
- T. G. Kolda, R. M. Lewis and V. Torczon. Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods, *SIAM Review*, 45(3): 385-482, 2003. doi:10.1137/S0036144502428893.
- T.-H. Le and D.-J. Kim. Application of a real-coded genetic algorithm for the fitting of a ship hull surface through a single non-uniform B-spline surface, *Journal of Marine Science & Technology*, 16(2): 226-239, 2011. doi: 10.1007/s00773-011-0118-1.
- K. F. Man, K. S. Tang and S. Kwong. *Genetic algorithms: concepts and designs*, Springer, Springer. 1999. doi: 10.1007/978-1-4471-0577-0.
- Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer-Verlag, Berlin Heidelberg New York, 1996. ISBN 3-540-58090-5.
- C. R. Reeves and J. E. Rowe. *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*, Kluwer Academic Publishers, Boston. 2002. doi: 10.1007/b101880.
- A. Sorsa, R. Peltokangas and K. Leiviskä. Real-coded Genetic Algorithms and Nonlinear Parameter Identification, *IEEE International Conference on Intelligent Systems*, Varna, Bulgaria, September 6-8, 2008.