Balancing Meals Using Fuzzy Arithmetic and Heuristic Search Algorithms

Jean-Christophe Buisson and Alexandre Garel

***Abstract*—This paper aims at showing how well-known ideas in the fields of fuzzy arithmetic and heuristic search have been combined in an educational software in nutrition in order to provide not only a better mathematical modeling, but also significant functional improvements for end-users, comparing to other nutrition programs. This software, called Nutri-Expert, helps patients to improve their nutritional habits, by analyzing in detail their food intakes, and by suggesting changes that result in well-balanced meals. Fuzzy arithmetic is used to model the input and database data, and for all computations. A fuzzy pattern matching is performed between total amounts of nutrients and different norm patterns, and the results are displayed using a galvanometer metaphor. A heuristic search algorithm is used to find out minimal sets of pertinent actions to perform on a meal in order to make it well balanced. The search is guided by an evaluation function based on fuzzy pattern matching indexes. The different versions of the algorithm have been benchmarked against a test database of real meals. Finally, the medical efficacy of Nutri-Expert and its acceptance by end-users have been demonstrated in several medical studies, the main results of which are presented.**

***Index Terms*—Fuzzy arithmetic, heuristic speech, Nutir-Expert, nutrition.**

I. Introduction

THE educational software Nutri-Expert has been designed in cooperation with the Diabetes Department of the Toulouse Hospitals, Toulouse, France. For many chronic diseases such as diabetes, a particular attention must be paid to the diet, which should be properly adapted to the patient. A need has been identified by the medical group for an educational software of diet self monitoring, which should perform the daily task of analyzing and correcting the patient's meals at home. This program is available for most computer operating systems, as well as on the Internet.

Medical experiments have shown that an unsupervised six-month use of the software improved very significantly the patients' knowledge on nutrition and their cooking habits, and even improved significantly several physiological indicators such as blood glucose. They have also shown an increase of dialog between the patients and their physicians [11], [12].

Nutri-Expert is composed of several modules closely linked together. The main one is the analysis module, which will be described in detail in this paper, and which allows the patient to analyze a particular meal and get it well balanced. A second module gives well balanced daily diets of different kinds (ordinary, green, exotic, parties, etc.) The quantities of foods are adapted to the patients caloric needs and their medical problems, and a proposed meal can be imported into the analysis module in order to be modified and corrected. There are more than 200 of such meals in the database. A third module acts as an encyclopedia on nutrition, giving information and definitions about the words and concepts used in the software (calories, fat, proteins, blood glucose, criteria of equilibrium, etc.). It also gives complete recipes (more than 60) of dishes used in the other two modules.

Each patient is recorded in the system, and her file contains her main physical parameters such as her age, height, and weight, as well as her level of physical activity and a description of her possible medical problems or diseases. Her weight is asked and updated at most once a week.

The analysis module uses a custom made food composition database of more than 1800 foods, permanently updated by a pool of nutritionists. It gives the values of more than 20 nutrients, for all the common cooked and raw ordinary foods and the most common dishes. The nutritionists have been trained to use the input tools which allow them to express the possible imprecision and fuzziness attached to these values, as it will be shown later.

The meal analysis proceeds as follows. The patient describes her meal food by food. She can choose a food in a hierarchy of categories and subcategories, use a search tool, or point to a picture of it. The main nutritional information about the food is then displayed. The patient must then provide the weight of this food, and she can do it in any of the following several ways:

- by typing the weight in grams;
- by specifiying a number of portions (glasses for a drink, slices for bread, etc.);
- by choosing among a set of pictures showing different quantities of this food (Fig. 1).

As the meal description progresses, the quantities of the main nutrients concerned with the meal balance and their relative position to corresponding norms are displayed as galvanometers, which will be described in detail in Section II-E3)

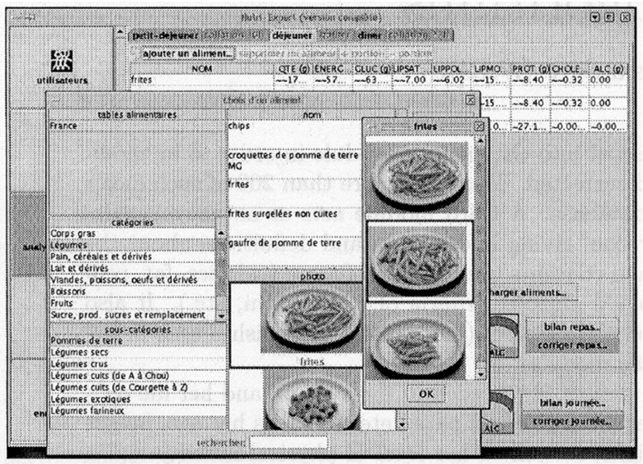
The nutrients involved in the balance are: energy, carbohydrates, fat, protein, complex carbohydrates, simple carbohydrates, saturated fatty acids, monounsaturated fatty acids, polyunsaturated fatty acids, calcium, potassium, dietary fibers, cholesterol, and alcohol. The constraints on these 14 nutrients depend partly on the medical problems of the patient. The evaluation option allows her to have a clear written assessment of his meal, which points out the main problems.

Manuscript received February 28, 2000; revised March 22, 2002 and May 20, 2002.

J.-C. Buisson is with the ENSEEIHT 2, rue Camichel, 31071 Toulouse, France (e-mail: buisson@enseeiht.fr).

A. Garel is at 4, imp. Ferret, 69008 Lyon, France (e-mail: alex.garel@nomad.fr).

Digital Object Identifier 10.1109/TFUZZ.2002.806323



NOM	QTE (g)	ENERC	GLUC (g)	LIPSAT	LIPPOL	LIPMO	PROT (g)	CHOLE	ALC (g)
frites	~~17...	~~57...	~~63...	~~7.00	~~6.02	~~15...	~~8.40	~~0.32	0.00
						-15...	~~8.40	~~0.32	0.00
						0...	-27.1...	-0.00...	-0.00...

Fig. 1. Using pictures to choose foods and food quantities.

She can then try to get the meal well balanced by adding or removing foods or by changing the weights of the foods, but this is a difficult task, which she seldom carries out to the end. The correction option automatically finds the smallest sets of acceptable changes which make the meal well balanced. They are only suggestions, and the patient does not have to follow them. Usually, she modifies one or two elements, and the analysis cycle starts again.

The details of the balancing algorithm, and how the system deals with the inherent imprecision and fuzziness of data is the main purpose of this paper.

II. Fuzzy Arithmetic

A. Sources of Imprecision in Nutri-Expert

There are two different sources of imprecision or fuzziness in Nutri-Expert. The first is in the food composition database, where some values are imprecisely known, sometimes even completely unknown. For instance, there is only one entry for *apple* in the database although there are several species of apple on sale, which can all be at various stages of maturity. In this case, imprecision is a result of the averaging of several values. For other foods, the values of some nutrients have not been measured, and the nutritionists must still express what they know about it, even if it is very imprecise.

The second source of imprecision comes from the patient, when he enters the food weights for his meal into the analysis module. If the weights of some packaged items are precisely known (yogurts, bottles, etc.), all other foods (vegetables, meat, pastas, etc.) must be either weighted with a kitchen scale or described in terms of portions. In practice, patients rarely use a scale, so we have been assuming that the weights of non-packaged items were an approximate assessment expressed by a whole number of portions.

B. Fuzzy Intervals

A fuzzy interval is a fuzzy set [13] of real numbers, denoted M, with a membership function μ_M which is unimodal and upper semicontinuous, that is $\forall\alpha \in]0,1]$, $M_\alpha = \{r|\mu_M(r) \geq \alpha\}$ (the α – *cut* of M) is a closed interval (cf Fig. 2)

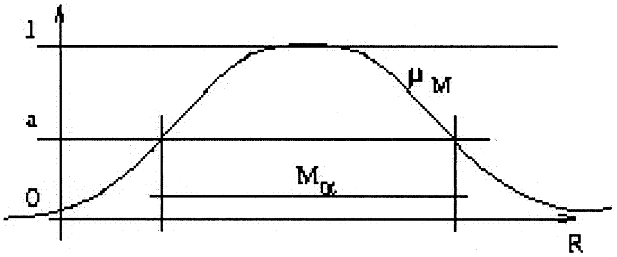


Fig. 2. Fuzzy interval.

A fuzzy interval generalizes the concept of a closed interval, including real numbers. It may model the range of some variable x with some more sophistication than a usual interval. Namely, the *support* $S(M) = \{r|\mu_M(r) \geq 0\}$ is the widest range for x (x cannot take a value outside of $S(M)$), while the *core* $\overset{\circ}{M} = \{r|\mu_M(r) = 1\}$ is the set of most plausible values of x, called modal values.

A fuzzy interval is a convenient tool for representing imprecise quantities. In many real situations in which we are to evaluate a given parameter (not known with precision), a closed interval is unsatisfactory. If we make the interval rather large in order to be sure that the real value is inside it, the subsequent calculations based on it may yield too imprecise results to be of any practical interest. On the contrary, if the interval is too narrow, the high precision of the results could be illusory if an error have been made at the beginning. The fuzzy interval allows us to be pessimistic and optimistic at the same time: the support of the interval will be chosen large enough to be sure that no value is unduly excluded, and the core will represent what we think are the most plausible values.

We shall assume that the involved fuzzy numbers are all of the same type, i.e., there are shape function L, R, modal values $\underline{m}$, $\overline{m} \in R$, spreads $\alpha, \beta \leq 0$ such that (see [5])

$$\begin{aligned} u \leq \underline{m} : \mu_M(u) =& L\left(\frac{\underline{m} - u}{\alpha}\right) \\ \underline{m} \leq u \leq \overline{m} : \mu_M(u) =& 1 \\ u \geq \overline{m} : \mu_M(u) =& R\left(\frac{u - \overline{m}}{\beta}\right) \end{aligned}$$

where L (or R) is an upper semicontinuous monotonic function with

$$\begin{aligned} L(0) =& 1, \qquad 0 < L(x) < 1 \, \forall x \in]0,1[\\ L(x) =& 0 \, \forall x > 1. \end{aligned}$$

For a value x which is inside the support of M, but outside of its peak, the membership degree is $\mu(x) \in]0,1[$.

What is of primary importance is to determine the set of values which are completely impossible (for which μ_M is equal to 0) and the set of values which are completely possible (for which μ_M is equal to 1); the remaining subsets of the domain correspond to gradual transitions. The possibility theory is not very sensitive to slight variations of the possibility degrees; what really matters is that if a value is considered more possible than another one for a variable, then a greater possibility degree must be assigned to the former value.[1]

[1] This is why possibility theory is often said to be a "qualitative quantitative model."

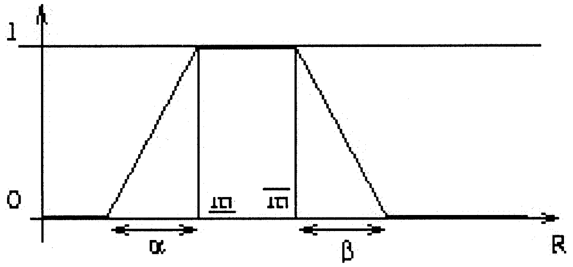


Fig. 3. Fuzzy interval modeled by a 4-tuple.

In other words, the linear by parts $\max(1-x,0)$ function for L and R will provide the desired properties of resilience.

From a computational point of view, such intervals will be modeled by the 4-tuple $(\underline{m}, \overline{m}, \alpha, \beta)$ (cf. Fig. 3).

Precise values, ordinary intervals and fuzzy numbers can be represented by these 4-tuples.

C. Elicitation of Fuzzy Data From Users

It seemed unrealistic to ask a patient anything about the imprecision of his foods' weights. When a user gives a weight value, the program looks up in the food database if it is a packaged item or not. If so, the weight is assumed to be a precise number, represented by a fuzzy interval ($m, m,$ 0, 0). If it is not a packaged item, the program looks if the weight has been given as a number of portion or in grams. The number of portion is assumed to imply a greater imprecision. In both cases, the precise value given by the patient is transformed in a fuzzy interval, using a transforming function associated to the food and the portion/gram choice, the name of which is stored in the food database. Presently there are 15 different such functions. For instance, the f/10/20 function is associated to the bread's weight when expressed as a number of slices, and transforms a weight x into the fuzzy interval $(x-10\%, x+10\%, x*20\%, x*20\%)$.

As for the values stored in the food composition database, they are all fuzzy intervals, and the nutritionists have been instructed how to deal with them. A value is stored as a precise number, along with a transforming function such as the aforementioned f/10/20; the set of transforming function has been elaborated by the nutritionists themselves. When a value field is left blank, it means that absolutely nothing is known on the value. It corresponds to a possibility distribution uniformly equals to 1, which can be implemented by the fuzzy interval $(-\infty, +\infty, +\infty, +\infty)$. For instance, it is often the case for the calcium of commercial foods, the quantity of which is not printed on the package. The total amount of calcium in a meal containing such foods can still be computed, but the result will have an increase of imprecision affecting its upper bound.

D. Fuzzy Arithmetic

Zadeh's extension principle [14] can be used to compute the fuzzy range of $M*N$, where M and N are two fuzzy intervals associated to variables x and y respectively, and $*$ an operation between real numbers. Dubois and Prade [5] have shown that, applying the rules of computation of possibility theory, and supposing that x and y are not linked (noninteractive)

$$\mu_{M*N}(\omega) = \sup\left\{\min(\mu_M(u), \mu_N(v)) \mid u*v=\omega\right\}. \quad (1)$$

In terms of α-*cuts*, we have

$$\mu_{M*N}(u) = \sup\{\alpha \in]0,1] \mid u \in M_\alpha * N_\alpha\} \quad (2)$$

that is to say

$$M_\alpha * N_\alpha \subseteq (M*N)_\alpha. \quad (3)$$

It can be shown that, for arithmetic operations, the supremum in (1) is attained and (3) stands as an equality, so

$$(M*N)_\alpha = M_\alpha * N_\alpha = \{u*v | u \in M_\alpha, v \in N_\alpha\} \quad (4)$$

(4) allows us to calculate $M*N$ for the four arithmetic operations,M and N being the trapezoid-shaped fuzzy intervals $(\underline{m_1}, \overline{m_1}, \alpha_1, \beta_1)$ and $(\underline{m_2}, \overline{m_2}, \alpha_2, \beta_2)$

For addition and substraction, it is easy to see that

$$\begin{aligned} M+N =& (\underline{m_1}+\underline{m_2}, \overline{m_1}+\overline{m_2}, \alpha_1+\alpha_2, \beta_1+\beta_2) \quad (5)\\ M-N =& (\underline{m_1}-\overline{m_2}, \overline{m_1}-\underline{m_2}, \alpha_1+\beta_2, \beta_1+\alpha_2). \quad (6)\end{aligned}$$

For multiplication and division, an approximation must be made, for the result is no longer trapezoidal. The most important parts of the result, namely the core and the support, can be calculated exactly, and we have no choice but to draw a straight line between them on both sides if we want a trapezoid-shaped result again. This approximation will leave unchanged the order of the membership values of [0,1].

It can be checked that

$$M \times N = (\underline{m_1}*\underline{m_2}, \overline{m_1}*\overline{m_2}, \alpha_3, \beta_3)$$

where

$$\begin{aligned} \alpha_3 =& \alpha_2*(\underline{m_1}-\alpha_1)+\alpha_1*\underline{m_2}\\ \beta_3 =& \beta_1*\overline{m_2}+(\overline{m_1}+\beta_1)*\beta_2 \quad (7)\end{aligned}$$

and

$$\frac{M}{N} = \left(\frac{\underline{m_1}}{\overline{m_2}}, \frac{\overline{m_1}}{\underline{m_2}}, \alpha_4, \beta_4\right)$$

where

$$\begin{aligned} \alpha_4 =& \frac{\underline{m_1}*\beta_2+\alpha_1*\overline{m_2}}{\overline{m_2}*(\beta_2+\overline{m_2})}\\ \beta_4 =& \frac{\underline{m_2}*\beta_1+\alpha_2*\overline{m_1}}{\underline{m_2}*(\underline{m_2}-\alpha_2)}. \quad (8)\end{aligned}$$

E. Fuzzy Pattern Matching

1) Computation: Using the fuzzy arithmetic described above, the program is able to compute assessments on a set of nutrients, leading to a vector of data $D_1, D_2, \ldots D_n$. Each D_i is a fuzzy interval, which must be compared to a corresponding norm P_i, and the different pattern matching indices must be aggregated in order to assess to normality of the whole meal.

After discussion with the medical group, it has been established that the different norms P_i were also fuzzy intervals. For instance, they told us that, for an ordinary lunch and for all kinds of medical diseases, "the fat percentage of the caloric goal must be approximately between 25% and 35%." They added that 20% and 40% were clearly out of the norm, so their statement could be translated into the following trapezoid-shaped fuzzy interval: (5,25,35,5).

Let P and D be the fuzzy interval representing the pattern and the data, respectively. Two scalar measures are used to estimate the compatibility between P and D: a degree of possibility of matching $\Pi(P;D)$ and a degree of necessity of matching $N(P;D)$ which are, respectively, defined by (see [15])

$$\Pi(P;D) = \sup_{u\in U} \min(\mu_P(u), \mu_D(u)) \tag{9}$$

$$N(P;D) = \inf_{u\in U} \max(\mu_P(u), 1-\mu_D(u)). \tag{10}$$

The measure $\Pi(P;D)$ estimates to what extent it is possible that P and D refer to the same value u; in other words, $\Pi(P;D)$ is a *degree of overlapping* of the fuzzy set of values compatible with P, with the fuzzy set of values compatible with D. The measure $N(P;D)$ estimates to what extent it is necessary (i.e., certain) that the value to which D refers is among the ones compatible with P; in other words $N(P;D)$ is a *degree of inclusion* of the set of values compatible with D into the set of values compatible with P. The duality possibility/necessity, i.e., the necessity of an event corresponds to the impossibility of the opposite event, is expressed here by the relation

$$N(P;D) = 1 - \Pi(\overline{P};D) \tag{11}$$

where $\mu_{\overline{P}} = 1 - \mu_P$ is the membership function of $\overline{P}$, complement of the fuzzy set of values compatible with P. Clearly, we always have $\Pi(P;D) > N(P;D)$. Besides it is worth noticing that $N(F;F) = 1$ if and only if μ_F if the membership function of an ordinary subset of U; otherwise, we have only $N(F;F) \geq 1/2$. Indeed, when two identical constants have a fuzzy meaning, we cannot be completely sure that they refer exactly to the same set of values. In any case we have $N(S(F);F) = 1$, where $S(F)$ is the support of F. The limiting cases where $\Pi(P;D)$ and $N(P;D)$ take values 0 and 1 are also useful to characterize. As previously defined, let $S(F)$ and $\overset{\circ}{F}$ be the support and the peak of F, respectively. Then it can be checked that

1) $\Pi(P;D) = 0$ if and only if $S(P) \cap S(D) = \phi$;
2) $\Pi(P;D) = 1$ if and only if $\overset{\circ}{P} \cap \overset{\circ}{D} \neq \phi$;
3) $N(P;D) = 1$ if and only if $S(D) \subseteq \overset{\circ}{P}$.

Note that Property 3) defines a stronger inclusion between fuzzy sets than the usual one (i.e., $\mu_D \leq \mu_P$) which only implies $N(P;D) \geq 0.5$.

In conclusion, $\Pi(P;D)$ and $N(P;D)$ are not ad hoc similarity measures; they have clear and precise semantics which correspond to the nature of the flexible pattern-matching problem.

The computation of $\Pi(P;D)$ and $N(P;D)$ on trapezoid-shaped intervals is a matter of lines intersection (cf. Figs. 4 and 5). Let $(\underline{M}_P, \overline{M}_P, \alpha_P, \beta_P)$ and $(\underline{M}_P, \overline{M}_P, \alpha_P, \beta_P)$ be the representation of P and D, respectively.

It can be checked that

$$\Pi(P;D) = \min(1, XP, YP) \tag{12}$$

where

$$XP = \max\left(0, 1 - \frac{\underline{m}_D - \overline{m}_P}{\alpha_D + \beta_P}\right)$$

and

$$YP = \max\left(0, 1 - \frac{\underline{m}_P - \overline{m}_D}{\alpha_P + \beta_D}\right)$$

$$N(P;D) = \min(1, XN, YN) \tag{13}$$

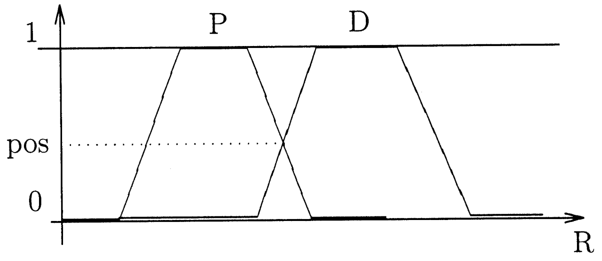


Fig. 4. Geometric computation of the possibility measure.

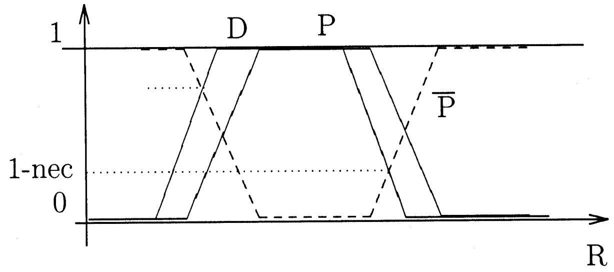


Fig. 5. Geometric computation of the necessity measure.

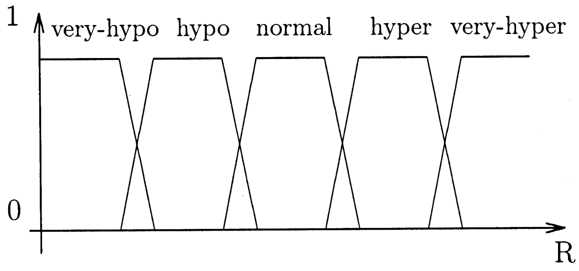


Fig. 6. Overlapping norms of Nutri-Expert.

where

$$XN = \max\left(0, \frac{\overline{m}_P - \overline{m}_D + \beta_P}{\beta_P + \beta_D}\right)$$

and

$$YN = \max\left(0, \frac{\underline{m}_D - \underline{m}_P + \alpha_P}{\alpha_P + \alpha_D}\right).$$

2) Overlapping Norms: As for the norm patterns used in Nutri-Expert, it has been decided with the medical group that, for each nutrient, five partially overlapping fuzzy intervals would be used, called: *very-hypo, hypo, normal, hyper, very-hyper* (cf. Fig. 6).

3) Using Galvanometers to Represent Matching Degrees: For each nutrient, the possibility and necessity compatibility degrees must be displayed to the end-user in a clear and not oversimplified manner.

A metaphor of galvanometer has been used, with colored area representing the accepted parts of the data domain, and where the needle may be "fuzzy" (Fig. 7).

More exactly, the green, orange and red parts represent the core, the support and the complementary of the support respectively, and the needle has the thickness of the support of the data. The nutrient label at the bottom is displayed in a background the color of which summarizes the compatibility: green if the needle

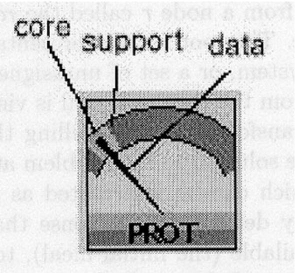


Fig. 7. Galvanometer. The core and support are the lightly colored areas; the "fuzziness" of the needle expresses the imprecision of data.

if completely included in the green area ($N(P;D) = 1$), red if the needle if completely in the red area ($\Pi(P;D) = 0$), and a mix of red and green in other situations.

F. Global Matching Evaluation

We can now compute a set of indices $\Pi(P_i;N_i)$ and $N(P_i;N_i)$ when matching separately each data of the set $D_1, D_2, \ldots, D_n$ with the corresponding pattern of the set $P_1, P_2, \ldots, P_n$. Each D_i and P_i were defined above as an assessment and its corresponding norm for each nutrient considered for the balance of the meal.

Since the involved variables are independent (i.e., the fuzzy set of values compatible with a variable does not depend on the value given to another variable), and if we suppose that all the parts of the pattern have an equal importance, an aggregation of the different measures is (see [6])

$$\Pi(P_1 \times \cdots \times P_n; D_1 \times \cdots \times D_n) = \min_{i=1,n} \Pi(P_i;D_i) \tag{14}$$

$$N(P_1 \times \cdots \times P_n; D_1 \times \cdots \times D_n) = \min_{i=1,n} N(P_i;D_i) \tag{15}$$

where × denotes the Cartesian product defined for fuzzy sets by

$$\forall u_i \in U_i, \forall u_j \in U_j,\ \mu_{P_i \times P_j}(u_i, u_j) = \min(\mu_{P_i}(u_i), \mu_{P_j}(u_j)). \tag{16}$$

This aggregation using the "min" operation preserves the respective semantics of possibility and necessity.

Yet, in the particular case of Nutri-Expert, the different variables have not an equal importance. According to the medical team, the fat, carbohydrate and protein percentage and the caloric intake are by far the most important variables. As for the others, it depends very much on the kind of meal and of the particular medical problems of the patient: some variable may be as important as the first four, and others may be completely unimportant.

Let $\omega_1, \omega_2, \ldots, \omega_n$ be the grade of importance of patterns $P_1, P_2, \ldots, P_n$, respectively. It is supposed that $\forall i, \omega_i \in [0,1]$, the greater ω_i the greater the importance of P_i; we also assume that $\max_{i=1,n} \omega_i = 1$ (normalization), i.e., the most important patterns are graded by 1. Then, if s_i denotes a degree of matching of a datum (possibility or necessity) with respect to the atomic pattern P_i, the corresponding degree of matching s of this datum with respect to the whole pattern $(P_1, P_2, \ldots, P_n)$, taking into account the importance assessement, will be given by (see [7] and [8])

$$s = \min_{i=1,n} \max(1 - \omega_i, s_i). \tag{17}$$

Note that if all the ω_i's are equal to 1 (equal importance), we get $s = \min_{i=1,n} s_i$; when $\omega_i = 0$, the matching with the pattern P_i is not taken into account.

The introduction of weights as proposed in (19) amounts to modifying the patterns P_i into P_i^* such that

$$\mu_{P_i^*}(u) = \max(\mu_{P_i}(u), 1 - \omega_i).$$

The implementation in Nutri-Expert of this weighted fuzzy pattern matching technique has been quite simple, and takes place in two stages.

A two-dimensional array indicates for each kind of meal and for each nutrient the norm pattern to use, and its weight in the global matching.

Then the program takes into account the possible medical problems of the patient in the following way. For each disease of the patient (he may have several) and for each nutrient, another two dimensional array specifies a new norm pattern and a new weight, which overrides a possible previous setting of the first stage.

Finally, a meal is said well balanced if the global pattern matching with the "well balanced" norm patterns leads to $\Pi = 1$ and $N \geq 0.7$. Otherwise, for each nutrient responsible for the unbalance, the program makes a verbal comment using a simple algorithm.

III. Heuristic Search Algorithms to Balance Meals

So, Nutri-Expert takes into account the natural imprecision and fuzziness of the user's food quantities as well as the imprecision and fuzziness of a food composition, in order to compute the fuzzy quantity of each of the 14 nutrients considered in a meal. A global pattern matching is then performed to assess to which extend the meal is well balanced. Now, the program aims at telling the patient how he may modify this meal to make it well balanced and adapted to his possible medical problems.

A. Analysis of the Problem

As a example, Romain, a child of eight, wanted to eat for breakfast:

- four slim slices of "baguette" (french bread) (60 g);
- one glass of skimmed milk (250 g);
- four teaspoonfuls of sweet cocoa powder (20 g);
- a teaspoonful of fruit marmelade (15 g).

According to the physical and medical record of Romain, this meal is unbalanced, due in particular to an excess of carbohydrates and a deficiency of fat. An acceptable transformation would be, for instance

- decrease bread quantity to 3 slices (45 g);
- decrease cocoa to one teaspoonful and a half (8 g);
- add 10 g of butter.

Even with the help of an evaluation module which points out the nutrients responsible for an unbalance, patients generally happen to be unable to perform this balancing task by themselves. It is indeed difficult because modifying the weight of a food for instance often leads to have the balance of several nutrients modified at the same time.

This problem could be seen as linear. Foods could be considered as vectors in the space of nutrients while quantities of each food would represent a solution meal.

For the previous example of Romain's meal, the linear point of view would lead to solving

$$\begin{pmatrix} \text{car}_{\text{bread}} & \cdots & \text{car}_{\text{butter}} \\ \text{pro}_{\text{bread}} & \cdots & \text{pro}_{\text{butter}} \\ \vdots & \vdots & \vdots \\ \text{cho}_{\text{bread}} & \cdots & \text{cho}_{\text{butter}} \end{pmatrix} \begin{pmatrix} q_{\text{bread}} \\ \vdots \\ q_{\text{butter}} \end{pmatrix} = \begin{pmatrix} \text{car}_{\text{goal}} \\ \text{pro}_{\text{goal}} \\ \vdots \\ \text{cho}_{\text{goal}} \end{pmatrix}.$$

Where the unknown quantities are $q_{bread}, \ldots, q_{milk}$.

However, the following two aspects of the problem make this model useless:

- for many foods, the quantity should be a whole number of portions (ie. fruits, yoghurt, packaged foods, etc.);
- such equations cannot be solved easily in the framework of fuzzy arithmetic.

After discussing these issues with the medical group, the operational notion of *possible actions* on a meal emerged as central, and it has been characterized to an operational level. That is, we can consider a set of operations that can be applied to a meal in order to transform it. For each food, the database contains all the necessary information to compute the possible actions on the meal. Some elements such as the patients' habits regarding the use of fats when cooking meat or fish, for instance, are asked once and then recorded in their personal file.

So, the meal provided by the patient generates a state–space where the solution is to be found. This state–space is not explicit and we have to build it gradually with a local search algorithm. In addition, each operation has to be weighted with a cost, in order to rank the different found solutions. Putting it all together, everything is set for an heuristic search.

B. Search Algorithms in a State–Space

1) General Form of Search Algorithms: Solving a combinatorial problem such as getting a meal well balanced can be viewed as finding a path in a graph, called the *search graph*, from a node r called the *root* to a node t called a *goal node*. The root is the representation of the initial state of the system, or a set of unassigned decision variables. A path from the root to a goal is viewed as the set of elementary transformations modeling the process of building a feasible solution to the problem at hand.

The search graph, which can be interpreted as a state–space, is only potentially defined, in the sense that only the root is explicitly available (the initial meal), together with a set of rules which specify how to build the successors of a current node.

Using applicable rules on some node ("the father") creates new nodes ("the sons"), together with an arc from the father to each son. This is called *node expansion*. Each arc is valuated by a cost which is supposedly a positive number. The cost of a path is the sum of the costs of its arcs. An optimal solution corresponds to a minimal cost path from the root to a goal node. The graph search methodology consists in applying rules, where possible, to nodes until a goal node is reached. As long as a goal node is not reached, the main problem is to select the proper node to be expanded.

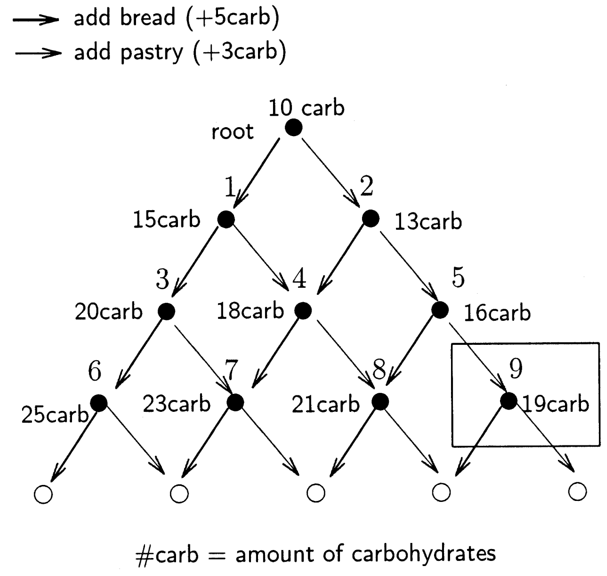


Fig. 8. Example of breadth-first graph search.

Let us consider a list of "generated meals" and another of "visited meals." At the beginning, "generated meals" contains the meal given by the patient (= the *initial meal*) and "visited meals" is empty. The graph search procedure consists in the repetition of the following.

```
repeat
 a choose a meal $m$ from "generated
meals"–"visited meals"[2]
 b check the balance of meal $m$
 c generate meal $m$'s neighbors by ap-
plying operations and put them in "gener-
ated meals"
 d put meal $m$ in "visited meals"
until check in b is positive
```

It should be noted that the choice of the current meal in **a** is not precisely defined. This choice, along with the way nodes are put in "generated meals" in **c**, leads to different classes of search algorithms, with different properties.

For instance, on Fig. 8 is represented a simplified search graph, where "generated meals" is used in **c** as a first-in–first-out (FIFO)[3] and where the first meal in **a** is chosen. Such exploration is called a breadth-first search algorithm, because it examines all states that are n operators applications from the initial state before any that are $n + 1$ away. It has the important property that it will eventually explore every state of the state–space.

Only the amount of carbohydrates is considered here, and the *initial meal* contains 10 g, whereas the goal is 19 g. To simplify, the quantities are represented as precise values. The only two possible operations are: add Bread and add Pastry.

2) Heuristic Search Algorithms: Heuristic search is a particular case of the previous algorithm where the program tries

[2] "-" is the minus operation for sets.

[3] The nodes are selected in the order in which they where put.

to cleverly visit the state–space in order to quickly find the solution. In the previous search canvas, each stage (**a**, **b**, **c**, and **d**) can be now refine to be ruled by an heuristic. Let us assume that, for each meal m in "generated meals," the following is known:

- the minimum cost of transformation from the *initial meal* to m, noted g;
- an estimation of the cost to transform m to its closest solution, noted h.

Then, the algorithm is called an "*A* algorithm" and it goes as follows.

```
repeat
 a choose the meal $m$ from "generated
meals" – "visited meals" with the best
evaluation
 b check the balance of meal $m$
 c generate meal $m$'s neighbors by ap-
plying operations and put them in "gen-
erated meals". For each generated meal,
compute an evaluation term $f = g + h$
 d put meal $m$ in "visited meals"
until check in b is positive
```

f is an estimation of total cost of closest solution in the area.

From the previous example, it is possible to replace the breadth-first search by an A search. The heuristic term h is the goal amount of carbohydrates minus this actual amount and infinity if we are over (Fig. 9).

If h is always lower than the real cost to the closest solution (i.e., h is "optimistic") then the search is called A^* and it has been proved that the first found solution always has the lowest possible cost ([9]).

Note: An heuristic h_1 is better informed than another h_2 if

$$\forall m \in \text{“meals,”} h_1^{(m)} > h_2^{(m)}. \tag{18}$$

C. First Balancing Algorithms

1) A Simple Set of Operations: In the first generation of the algorithms, we used the following set of operations on foods:

- modify its quantity to a minimum;
- modify its quantity to an average;
- modify its quantity to a maximum;
- remove it from the meal;
- replace it by the same amount of a better equivalent;

where minimum,average and maximum quantities are found in the foods database. The computation of g, minimum cost from the initial meal to the current meal, was performed on the basis of an equal cost of 1 for all of these operations.

2) Breadth-First Search Algorithm: 2392/3479: Considering that the primary goal is to change as less food as possible, each operation is weighted with the same cost (= uniform cost). The very first algorithm which has been implemented as a reference was the breadth-first search, as described in Section III-B.I.

To test it, we used a database of 3479 meals which had been really entered into the program by users. They were the users' meals of a given period of time, and we did not make any change on them. Some are clearly aberrant, with the same food repeated four times for instance. A meal's correction is said to exist for the algorithm if it is found before 20 000 states are developed.

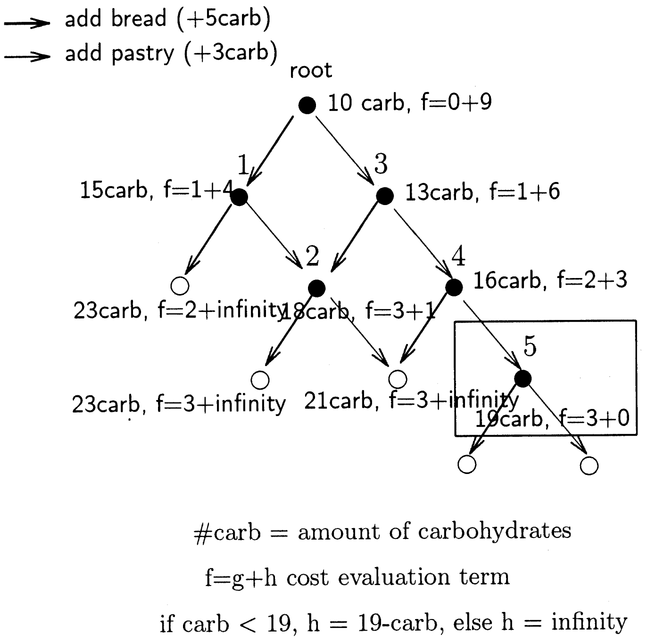


Fig. 9. Example of an A search.

For the breadth-first algorithm, 1087 meals were not corrected, so the success rate was 2392/3479.

3) First Heuristic Algorithm: 2569/3479: Thereafter, the first heuristic search algorithm developed was of type *A* (cf. Section III-B.II). Its h term (in $f = g + h$) was

$$h = \sum_{n \in \text{“unsatisfied nutrients”}} \omega_n \tag{19}$$

where ω_n weights the importance of the nutrient n, in practice, 2 for important nutrients and 1 for others, down to 0 for those which are too strongly linked with others (e.g., saturated lipids with lipids).

This term characterizes the extent of nutritional imbalance rather than the cost of the closest solution.

From the test database of 3479 meals, 2569 were corrected.

D. More Sophisticated Balancing Algorithms

It had been found on the previous experiments that the main problem was the set of possible operations on foods, which generated a gap too large between minimum/average/maximum quantities for some foods.

1) New Set of Operations: It appears that, for each food f, there exists a quantity a^f which is the smallest portion of f we can reasonably consider in practice. This quantity a^f has been called *increment amount of* f; it is, for instance, a spoonful for every food which can be served with a spoon, etc.

We then adopted the following new set of operations:

- add a quantity a^f to the quantity of food f;
- substract a quantity a^f to the quantity of food f;
- replace food f by the same amount of a better equivalent;

where the quantity cannot be outside the minimum/maximum range.

This new set raises new objections. Increasing then decreasing a food has a nonzero cost while it brings back to the same meal. So, it is important to bear in mind that in $f = g + h$, g is the minimum cost from the *initial meal*.

2) Computing a Better g, Minimal Cost From the Initial Meal: We are going to take into account the fact that removing or replacing a food is a transformation of the initial meal which is somehow more radical than simply changing a food's weight. In the previous computation of g, all operations had the same cost of 1. From now on, the costs of the operations on a meal are

- 1 for increasing a food's quantity by a^f;
- when decreasing a food:
$$\begin{cases} \text{DeleteCost}, & \text{if it causes deletion} \\ 1, & \text{otherwise;} \end{cases}$$
- a constant ReplaceCost for replacement

A large value for ReplaceCost and DeleteCost induces a search which tends to avoid replacements and deletions. Typically, ReplaceCost $= 4$ and DeleteCost $= 3$.

So, the value g for a meal m, denoted g^m, is computed as

$$g^m = \sum_{f \in \text{“foods”}} \frac{|q_f^m - q_f^i|}{a^f} + \sum_{\substack{f \in \text{“initial foods”} \\ q_f^m = 0}} \text{DeleteCost} + \sum_{f \in \text{“foods” } f \text{ replaced}} \text{ReplaceCost} \tag{20}$$

where q_f^m and q_f^i are the quantities of food f in the meal m and the *initial meal*, respectively. a^f is the *increment amount* of food f (see Section III-D.I).

3) First Algorithm With the Improved Value of g: 2757/3479: Another version of the algorithm was set up, using the improved value for g, and the following heuristic term h:

$$h = \frac{\sum_{n \in \text{“nutrients”}} \omega_n \cdot (1 - N_n)}{2} \tag{21}$$

where N_n is the necessity of matching of the nutrient n with the NORMO norm (see Section II-E) and ω_n a value weighting the medical relative importance of the various nutrients (carbohydrates, lipids, etc.)

The problem here is that the term $1 - N_n$ is almost always 1, and decreases only near the perfect matching.

From the test database of 3479 meals, 2757 were corrected.

E. Heuristic Search Algorithms Close to A^*

The set of operations which is described in Section III-D.I increases the size of the state–space, so we need a more efficient heuristic search to reach solutions.

1) Distance Before Matching: Let us consider the *distance before matching* for nutrient n

$$\Delta_n(m, \text{goal}) = \frac{\phi_n^{\text{goal}} - \phi_n^m}{\alpha_n^{\max}} \tag{22}$$

where ϕ_n^{goal} is the goal quantity of nutrient n, ϕ_n^m the quantity of nutrient n in meal m and $\alpha_n^{\max}$ is the maximum increment for nutrient n in one operation

$$\alpha_n^{\max} = \max_{f \in \text{“foods”}} (a^f \varphi_n^f) \tag{23}$$

with φ_n^f quantity of nutrient n in (1 g of) food f and a^f *increment amount* of food f (in grams).[4]

$\Delta_n(m, goal)$ **measures the minimum cost, in number of operations, it demands to level the nutrient** n; in other words, it is the number of operations it takes to have this nutrient normalized using the better food for it. $\Delta_n(m, goal)$ is negative if the nutrient is over its norm, positive if it is under.

To balance a meal m, each nutrient has to be normalized, particularly the nutrient with the maximum *distance before matching*. So, the closest solution is farther than

$$\max_{n \in \text{“nutrients”}} |\Delta_n(m, \text{goal})|. \tag{24}$$

In fact, more information can be worked out from this heuristic term. If the nutrient unbalance responsible for the max is negative, a lower estimation of the number of “increase food” operations to the nearest solution ($\Delta_n > 0$) is

$$h_{\text{increase}}^m = \max\left(0, \max_{n \in \text{“nutrients”}} (\Delta_n(m, \text{goal}))\right) \tag{25}$$

and the minimum of “decrease food” operations ($\Delta_n < 0$) is

$$h_{\text{decrease}}^m = -\min\left(0, \min_{n \in \text{“nutrients”}} (\Delta_n(m, \text{goal}))\right). \tag{26}$$

Finally, the closest solution is farther than

$$h^m = h_{\text{increase}}^m + h_{\text{decrease}}^m. \tag{27}$$

2) Exact Computation of the Distance Before Matching: Considering that all computations are made using fuzzy arithmetic, the *distance before matching* for the nutrient n, $\Delta_n(m, \text{goal})$ should measure the number of times you can add $\alpha_n^{\max}$ to nutrient n before it matches the goal norm. It is not as simple as a minus operation, and may be expensive in time.

Assuming $\alpha_n^{\max}$ and $\phi_n^{\text{goal}} - \phi_n^m = \Delta\phi$ are fuzzy computed, the distance is found by dichotomy, using the matching function $\text{qualify}(\text{data}, \text{pattern})$, which returns HYPO, NORMO, or HYPER.

The goal of the dichotomy is to find Δ_n, minimum number by which to multiply $\alpha_n^{\max}$ so that $\text{qualify}(\Delta_n \dot{\alpha}_n^{\max}, \Delta\phi)$ is NORMO.

Let us consider the case where the nutrient is under its norm.[5]

The dichotomy begins with a lower bound ($= L$) and an upper bound ($= U$) such that

- $\text{qualify}(L \times \alpha_n^{\max}, \Delta\phi)$ is HYPO;
- $\text{qualify}(U \times \alpha_n^{\max}, \Delta\phi)$ is NORMO.

The dichotomy algorithm is as follows.

```
repeat
 $M \longleftarrow U + L/2$ {compute the middle}
 if $(\text{qualify}(M \times \alpha_n^{\max}, \delta) = NORMO)$ then
  $U \longleftarrow M$
```

[4] Consequently, $a^f \varphi_n^f$ is the quantity of nutrient n that an operation on food f add or remove.

[5] The case of a nutrient over its norm is symetric.

```
  else
    $L \longleftarrow M$
  end if
until $(U - L) < 0.1$.
```

Then, L contains the minimum value of $\Delta_n(m, \text{goal})$.[6]

Initial values U and L have to be determined, as close as possible of the solution to accelerate the dichotomy. As qualify is based on the value of possibility of matching plus necessity of matching, and considering that $0 \leq$ necessity $\leq$ possibility ≤ 1

- L is the value where Possibility for NORMO starts to be greater than 0[7];
- U is the value where Necessity for NORMO reaches 1[8] considering shapes of involved fuzzy sets, this is

$$L = \max\left(0, \underline{\underline{\left(\frac{\Delta\phi}{\alpha\text{max}_n}\right)}}\right) \tag{28}$$

$$U = \max\left(0, \frac{\underline{\Delta}}{\underline{\underline{\alpha_n^{\max}}}}\right). \tag{29}$$

There after h^m is computed from $\Delta_n(m, \text{goal})$ with (27).

3) Algorithm Using This Exact Computation: 2882/3479: The heuristic term using the exact computation of the distance before matching is "optimistic," since it is a worst case measure for each nutrient. This gives a truly A^* algorithm which is a good standard to evaluate other algorithm's performances, and which has been implemented.

From the test database of 3479 meals, 2882 were corrected.

4) Simplifying the Evaluation of the Distance Before Matching: 2910/3479: Keeping fuzziness apart to simplify, $\alpha_n^{\max}$ can be computed as a real number. This is done in (23) by replacing φ_n^f with an average of its fuzzy value (a^f is already real). Yet, ϕ_n^{goal} and ϕ_n^m are fuzzy numbers. The difference of nutrient n between meal m and goal $\Delta\phi = \phi_n^{\text{goal}} - \phi_n^m$ is computed as

- distance between lower bounds, if $\underline{\underline{\phi_n^m}}$ beneath $\underline{\underline{\phi_n^{\text{goal}}}}$ (cf. Fig. 10);
- distance between upper bounds, if $\overline{\overline{\phi_n^m}}$ over $\overline{\overline{\phi_n^{\text{goal}}}}$;
- 0 if bounds of ϕ_n^m are included in the bounds of ϕ_n^{goal}.

$\Delta\phi$ does not perform a truly "optimistic" evaluation of the distance before matching for several reasons

- $d(\text{meal}, \text{goal})$ may be pessimistic (Fig. 11);
- the real value of φ_n^f (and, consequently, $\alpha_n^{\max}$) is an average of the fuzzy term, not an optimistic evaluation.

However, this heuristic gives better results than with the exact computation. In fact, in most cases, the estimation of the distance before matching is just above its real value. This term seldom exagerates the cost to the closest solution[9] and the search is very close to A^*.

From the test database of 3479 meals, 2910 were corrected.

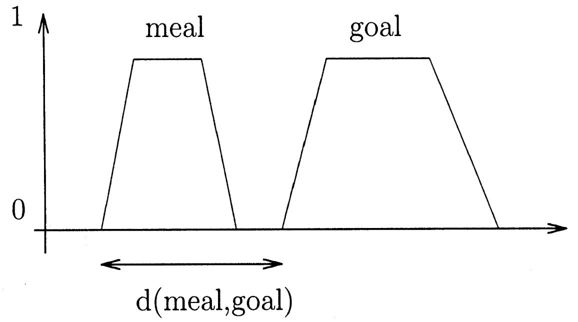


Fig. 10. Example for $d(meal, goal)$.

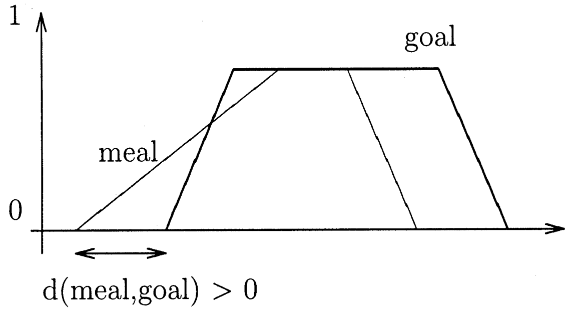


Fig. 11. Example where $d(meal, goal)$ is pessimistic.

5) Fuzzy Heuristic: The idea now is to compute the distance before matching as a fuzzy number. But of course this choice leads to an A search with a fuzzy heuristic.

So, $\alpha_n^{\max}$ is computed with "fuzzy max" of $a^f \varphi_n^f$. $\Delta\phi = \phi_n^{\text{goal}} - \phi_n^m$ is also computed as a fuzzy number. Now, the nutrient quantity begins to match its goal when it overlaps the closer part P of the goal ϕ_n^{goal}. That is

- the part between $\underline{\underline{\phi_n^{\text{goal}}}}$ and $\underline{\underline{\phi_n^{\text{goal}}}}$ if $\phi_n^m < \phi_n^{\text{goal}} (= P_{\text{low}})$;
- the part between $\overline{\overline{\phi_n^{\text{goal}}}}$ and $\overline{\overline{\phi_n^{\text{goal}}}}$ if $\phi_n^m < \phi_n^{\text{goal}} (= P_{\text{up}})$.

Both the lower P_{low} and upper part P_{up} can be seen as fuzzy sets,[10] (see Fig. 12).

Once P is chosen, the distance before matching can be computed as the fuzzy number

$$\Delta_n(m, \text{goal}) = \frac{P - \phi_n^m}{\alpha_n^{\max}}. \tag{30}$$

Finally, h_{increase}^m and h_{decrease}^m takes the "fuzzy max" and the "fuzzy min" of the $\Delta_n(m, \text{goal})$. Consequently, we have to compare fuzzy numbers to choose the meal to be developed (in Section III-B.I **a**). Such an operation can be very hard to work out completely. However, a gross but simple means to sort fuzzy sets is to compare their area-average values (see Fig. 13).

There, since h^m is a true optimistic evaluation of the closest solution, the search is A^* but in the sense of "area-average" order. Unfortunately, this order let us lose most part of the profit of fuzzy computation and results are not those expected. Moreover, during the distance before matching computation, fuzzy division spreads fuzziness which brings confusion to the search.

[6] With a precision of 0.1.

[7] If this value is lower than 0, we choose 0 since we know that $\Delta_n \geq 0$.

[8] N may never reaches 1 but a value "has if" is taken.

[9] It slightly exagerates the distance before matching, which is often far beneath the cost to the closest solution.

[10] P_{low} and P_{up} are viewed as an optimistic restriction of the goal.

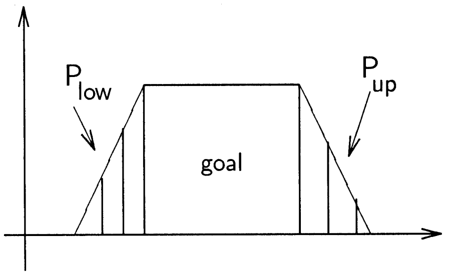


Fig. 12. Both parts can be seen as fuzzy sets.

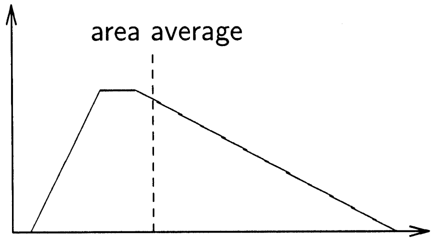


Fig. 13. Area-average value of a fuzzy number.

F. Modifying the A Search

1) Developing the Best Way: 2969/2479: In a concern for performance, our goal is now to limit the size of "generated meals" (in Section III-B.II). In our search, the order in which operations are applied is of little importance. Moreover, h^m_{increase} and h^m_{decrease} (in (27)) indicates the minimum number of increase/decrease operations to be done independently. Then when generating neighbors, it can be decided to only develop meals which will decrease the biggest term. That is, during computation of h^m a value direction^m is set to indicate whether

- $h^m_{\text{increase}} > h^m_{\text{decrease}}$ then $\text{direction}^m = \text{UP}$;
- $h^m_{\text{increase}} < h^m_{\text{decrease}}$ then $\text{direction}^m = \text{DOWN}$;
- both are equal then $\text{direction}^m = \text{ALL}$.

Afterwards, when the meal is selected, only decrease or increase operations are performed according to the value of direction^m. This adaptation of the standard A search algorithm is an improvement here, since there are less meals that are uselessly visited.

From the test database of 3479 meals, 2969 were corrected. Moreover, the algorithm is significantly faster than all previous.

2) Introducing a Second Term in the Heuristic: 3059/3479: The h^m_{increase} and h^m_{decrease} terms as seen in Section III-E.I are a kind of approximation to the first degree. Indeed, they consider each nutrient separately and the final result reflects the distance to the closest solution for only one nutrient. Yet, we could had a second degree, if we could take into account other nutrients as well.

Let us consider we have already computed the *distance before matching* for each nutrient and the nutrient responsible for the max, $n_{\max}$ (and min) in h^m_{increase} (and h^m_{decrease}). Now, for each food

- we compute N_f, the number of increase operations corresponding to h^m_{increase} with this food;
- then, for other nutrients (in excess), we compute the excess which would eventually remain after these operations;
- thereafter, we compute distance before matching for these remaining excesses, r^f_n;
- finally, we take the max of remaining excesses and add it to N^f.

So, we obtain $N^f + \max_{\substack{n\in\text{nutrients}\\ n\neq n_{\max}}}(r^f_n)$, which is the minimum number of operations required to find a solution using food f to level nutrient $n_{\max}$.

To compute this, we need, for each nutrient, the percentage that one *increment amount of food* f brings compared with the maximum $\alpha^{\max}_n$: P^f_n Then, it can be checked that

$$N_f = P^f_{n_{\max}} * h^m_{\text{increase}}. \tag{31}$$

Consequently, r^f_n is computed as the distance before matching minus quantities brought by the N_f first increases

$$r^f_n = \Delta_n - N_f * P^f_n. \tag{32}$$

Example: In the meal m: eggs, butter, cereal, milk, the distance before matching for nutrients is

- 15 for lipids;
- 10 for carbohydrates;
- 7 for proteins.

The nutrient responsible for the max is lipids. One *increment amount* of milk adds

- 66% of $\alpha^{\max}_{\text{lipids}}$, so $N^{\text{milk}} = 15 * 100/66 = 22.5$;
- 10% of $\alpha^{\max}_{\text{carbohydrates}}$, so $r^{\text{milk}}_{\text{carbohydrates}} = 10 - 22.5 * 0.2 = 6.5$;
- 30% of $\alpha^{\max}_{\text{proteins}}$, so $r^{\text{milk}}_{\text{proteins}} = 7 - 22.5 * 0.3 = 0.25$.

So, the evaluation for milk is 29 $(22.5 + 6.5)$.

Now, the new heuristic term for increasing is the minimum of all these values

$$H^m_{\text{increase}} = \min_{f\in\text{foods}}\left(N^f + \max_{\substack{n\in\text{nutrients}\\ n\neq n_{\max}}}(r^n_f)\right). \tag{33}$$

The computation of H^m_{decrease} is similar.

It is interesting to note that, since $\Delta_{n_m\max}$ is greater than each Δ_n, N^f is always higher than $\max_{n\in\text{nutrients}_{\neq n_{\max}}}(r^n_f)$. Therefore, a food f is interesting only if $P^f_{n_{\max}} > 0.5$.

In our previous example, only milk and butter verified $P^f_{n_{\max}} > 0.5$. For milk, evaluation is 29, and for butter evaluation is $15 + 10 = 25$ (butter brings 100% lipids and 0% carbohydrates). So, $H^m_{\text{increase}} = 25$.

This is not a truly optimistic term since, in certain cases, the best way is a mixing of two foods. However, it gives a good evaluation in most cases, and the added computing time is compensated by a faster search and a gain in memory consumption.

From the test database of 3479 meals, 3059 were corrected. It is so far our best algorithm, and it has been implemented in the current version of Nutri-Expert.

G. Results of Medical Evaluations

We will briefly discuss the results of a recent extensive medical evaluation ([12]). Two groups of obese patients were followed up over one year in a randomized study: the first group received close traditional management (seven nutritional visits over the year, with physicians and dietitians conjointly) and the second one also used at home the Nutri-Expert system. 557 patients were enrolled in the study by 16 French centers of diabetology and nutrition. Body mass index (BMI), tests of dietetic knowledge, dietary records and centralized biological measurements were assessed at inclusion, six and 12 months. 341 patients were evaluated at the end of the year.

The group using Nutri-Expert scored significantly better in the tests of dietetic knowledge than the control group. For all patients, nutritional education led to a significant improvement in BMI, dietary records and biological measurements, without significant difference between the two groups. Five years after the end of the study, the weight of 148 patients was recorded; mean BMI was significantly lower than the initial value but there was no significant difference between the two groups.

H. Concluding Discussion

Nutri-Expert uses well known ideas in the fields of fuzzy set theory and heuristic search algorithms, to successfully provide a better mathematical modeling of the nutritionnal problems, as well as better end-user features.

Fuzzy arithmetic is used for all computations on data. Imprecision and fuzziness of user's food quantities as well as food composition values are represented by fuzzy numbers, and computations lead to fuzzy values for each of the 14 considered nutrients in a meal. The user usually chooses the food quantities using pictures, issuing corresponding fuzzy numbers. The norms given by international medical organizations have been used; it is difficult to satisfy them all, and it justifies the need for a balancing algorithm.

During the stage where fuzzy results are computed, then matched against norms, the time spent into fuzzy computations is not perceptible by the user. If food weights are progressively modified, galvanometers and assessment sentences modify continuously in a robust manner, without any important discontinuity in results when nutrient values cross norm boundaries.

When exploring the search space in order to improve the meal, the burden of fuzzy computations grows heavier. A gross performance profiling has shown that the program spends as much time in fuzzy computations as in managing the nodes of the state graph. This price is repaid by the quality of the solutions.

Balancing meals is an activity we practice informally every day, and which happens to be a difficult problem, of an operational and numeric nature. We were lead to adopt a set of operations on a meal which does not enlarge too much the state–space, and yet enables each problem to have a solution in it. Then, we studied different kinds of heuristic evaluation, in order to explore the search graph as fast as possible, and the notion of distance before matching appeared as central.

ACKNOWLEDGMENT

The authors would like to thank the nutritionists and physicians of the medical team at the Toulouse University Hospital, Rangueil, France.

REFERENCES

[1] J.-C. Buisson, H. Farreny, and H. Prade, "The development of a medical expert system and the treatment of imprecision in the framework of possibility theory," *Inform. Sci.*, vol. 37, pp. 211–226, 1985.

[2] J.-C. Buisson, H. Farreny, H. Prade, M. C. Turnin, J. P. Tauber, and F. Bayard, *Toulmed, An Inference Engine Which Deals With Imprecise and Uncertain Aspects of Medical Knowledge*, J. Fox, M. Fieschi, and R. Engelbrecht, Eds. New York: Springer-Verlag, 1987, vol. 33, Lecture Notes in Medical Informatics, pp. 123–140.

[3] J.-C. Buisson, "Nutri-expert, an educational software in nutrition," *Int. J. Intell. Syst.*, vol. 12, pp. 915–933, 1997.

[4] ——, "Approximate reasoning in computer-aided medical decision systems," in *The Handbooks of Fuzzy Sets*. Norwell, MA: Kluwer, 1999, vol. 7, ch. 11, pp. 337–361.

[5] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.

[6] ——, "Evidence measures based on fuzzy information," *Automatica*, vol. 21, pp. 547–562, 1985.

[7] ——, *Possibility Theory – An Approach to the Computerized Processing of Information*. NewYork: Plenum, 1988.

[8] D. Dubois, H. Prade, and C. Testemale, "Weighted fuzzy pattern matching," *Fuzzy Sets Syst.*, vol. 28, pp. 313–331, 1988.

[9] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost path," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, pp. 100–107, Feb. 1968.

[10] N. Nilsson, *Principles of Artificial Intelligence*. Palo Alto, CA: Tioga, 1980.

[11] M. C. Turnin, R. Beddok, J. Clottes, R. Abadie, P. Martini, J. C. Buisson, C. Soule-Dupuy, F. Bayard, and J. P. Tauber, "Telematic expert system diabeto : A new tool for diet self monitoring for diabetic patients," *Diabetes Care*, vol. 15, no. 2, 1992.

[12] M. C. Turnin, O. Bourgeois, G. Cathelineau, A. M. Leguerrier, S. Halimi, D. Sandre-Banon, V. Coliche, M. Breux, E. Verlet, F. Labrousse, D. Bensoussan, J. L. Grenier, M. F. Poncet, F. Tordjman, J. M. Brun, J. F. Blickle, C. Mattei, C. Bolzonella, J. C. Buisson, D. Fabre, J. P. Tauber, and H. Hanaire-Broutin, "Multicenter randomized evaluation of a nutritional educational software in obese patients," *Diabetes Metab.*, vol. 27, pp. 139–147, 2001.

[13] L. A. Zadeh, "Fuzzy sets," *Inform. Control*, vol. 8, pp. 338–353, 1965.

[14] ——, "The concept of a linguistic variable and its application to approximate reasoning," *Inform. Sci.*, pt. Part 1 : 8, pp. 199–249, 1975.

[15] ——, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets Syst.*, vol. 1, no. 1, pp. 3–28, 1978.

Jean-Christophe Buisson received the Engineering degree and the Ph.D. degree in computer science from ENSEEIHT, Toulouse, France.

He is currently an Associate Professor at the School of Engineering in Computer Science, ENSEEIHT. He has been developing computer tools in the diabetes department of Rangueil Hospital, Toulouse, France for more than 12 years.

Alexandre Garel received the Engineering degree in computer science from ENSEEIHT, Toulouse, France. He is currently working toward the Ph.D. degree at the Institut de Recherche en Informatique (IRIT), Toulouse, France.

He is interested in the incorporation of fuzzy computation into heuristic search algorithms.