Honey Bee Mating Optimization for generating personalized food menu recommendations

# Definition of the fitness function

The fitness function is an evaluation function that rates each state or solution. Usually, it returns a higher value for better states, and of course, lower for worse states.

For the diet of a person, the fitness function models the quality of the menu, in terms of following factors:

1. Kilocalories;
2. Macro-nutrients:
   * Carbohydrates;
   * Proteins;
   * Fats;
3. Micro-nutrients:
   * Vitamins A, B, C, D;
   * Calcium;
   * Iron;
   * Sodium;
4. Deviation of the quantities of the previously mentioned items from the desired ones;
5. Doctor's prescription;
6. Patient's preferences;
7. Cost and delivery time of the food package.

As the doctor or the patient wishes to control the meals on different levels, a fitness function for every level is computed. The evaluation of a solution, which in our case is a meal for a day is done on three different levels and then combined to form the final mark.

The first is the nutrients' and kilocalories' level and deals with evaluating the meals from these points of view. If the number of calories and the quantities of nutrients that they contain are close to the ideal (desired) ones, then this function will have a great value. The ideal values are given are computed based on the user’s profile (gender, age, physical activity etc.) and can be overwritten by the doctor.

The second level deals with food items. It takes into consideration inclusion or exclusion of items to meet the requirements or preferences of the doctor or patient. If some undesired food item exist in a food package, then it is marked lower.

Finally, the third level evaluates a solution based on the cost and delivery time of the food package by the food provider.

Equation

|  |  |
| --- | --- |
| Level | Evaluation by |
| 1 | Deviation of kilocalories, macro and micro-nutrients from the desired values |
| 2 | Existence/absence of required/restricted or (un)desired food items |
| 3 | Cost and delivery time |

Table : Subjects of each level of the fitness function

# Representation of a solution

# The “ideal” functions for components

The evaluation of a component of a solution (e.g. a nutrient or a food item) will require two pieces of information:

1. The real quantity of the component; this is what one solution proposes. For example, if the meal contains 1700 kilocalories, it is said that the “quantity” of the component “kilocalories” is 1700.
2. The desired quantity of the component; this what the quantity of one component is compared to and should be as closest to. This is called the “ideal” value (or quantity) of one component. For example, for a female adult, the desired quantity of kilocalories is 2000, thus, it is said that the “ideal function” of “kilocalories” is 2000.

The solutions that contain components in quantities closer to the “ideal” values are marked higher than the ones that contain components in quantities further from the “ideal” values. The function *error margin* evaluates each component from this point of view (deviation on the real value from the ideal value) and marks it by a number between 0 (smallest) and 1 (greatest). In the following paragraphs, the procedures for computing the ideal values are described.

There are separate procedures to compute the ideal values for nutrients (level 1), food items (level2) and for cost and delivery time (level 3).

## Computation of the ideal values for nutrients

The user has a profile that contains the following pieces of information about him or her that are used for computing the ideal nutritional values:

* Gender;
* Weight;
* Height;
* Age;
* PAF (physical activity factor);
* Optional desire for the user to gain or lose weight.

Based on the provided information, the ideal quantity of kilocalories intake is computed by [Harris-Benedict equation](http://en.wikipedia.org/wiki/Harris%E2%80%93Benedict_equation).

For example, for men, Harrison-Benedict equations states that the number of kilocalories should be:

, where BMR is the [Basal Metabolic Rate](http://en.wikipedia.org/wiki/Basal_metabolic_rate) and 66 + 13.75 \* weight + 5 \* height – 6.76 \* age.

The other macro- and micronutrients are computed by the formulas provided by [Reference Daily Intake (RDI)](http://en.wikipedia.org/wiki/Reference_Daily_Intake) and Tolerable Upper Intake Levels (UL). For each nutrient, either a fixed value or an interval is stored, or both. The latter permits the real value to belong to an interval in order to be scored maximum.

The following image is an extract from the Java class that models the previous information: a mapping from each nutrient to either a fixed value, an interval or both.

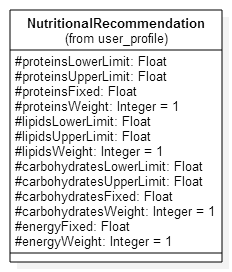


Figure : Java class that models ideal nutritional values

In other words, every nutrient is an arithmetic function that depends on the user’s profile

Equation

## Computation of the ideal values for food items

The user has the possibility to specify a list of food items that he or she wishes to be present in his or her recommended menu. Therefore, the user’s profile contains:

* A “like” list – the list of ingredients that the user likes; the ideal value for these is 1;
* A “dislike” list – the list of ingredients that the user does not like; the ideal value for these is 0.

Equation ideal values for food items

An important property of the two sets (like and dislike list) is that they have a void intersection, because the user is able to specify only one from the values (like or dislike):

Equation

## Ideal values for cost and delivery time

The cost and delivery time is specified by the user and they default to 0 in order to decrease the cost and time.

# The weights for components

Each component in one solution can be more “important” than other by assigning it a weight greater than the other. The motivation for this is giving priority to the user’s predilections for some component and also to doctor’s prescription.

The weights are level-dependent, i.e. they are meant to classify components within same level. For example, it does not make sense to say that the weight for calcium is greater that the weight of the cheese cake. However, it does make sense that the weight for calcium is greater that the weight of the kilocalories.

## Weights for nutrients

The weights for nutrients are taken into consideration in a hierarchical manner. This is achieved by amplifying the nutrient's requirements coming from a doctor more than those coming from the patient. Nevertheless, there are also default general recommendations for every nutrient, which are the least level on importance.

|  |  |
| --- | --- |
| Source | Weight |
| Doctor’s prescription | 100 |
| Personal preference | 50 |
| Default | 1 |

Table : Example of weights

For example, if *n* is a nutrient for which the doctor specified its value, then *w(n)* = 100. The values in this table are subject to be changed, in order to improve the results.

## Weights for food items

The user has the possibility to specify that he or she wishes a specific food item in his or her meal but also *not* to specify anything about certain items. In the former case, to model the fact that the user specified his or her preference, the item is put into the *specified list* and a weight value of 1 is assigned to the corresponding food item whereas in the latter case, when nothing is specified for a certain item, the weight value 0 is assigned.

Equation

An important aspect is that the *specified list* is actual the union of the *like list* and the *dislike list*, because the user can only specify one from the 2 values (like or dislike). In other words, the *like list* and *dislike list* form a partition of the *specified list* (see also Equation 4):

Equation The specified list

## Weights for cost and deliver time

The weights for cost and delivery time do not have any significant impact on the fitness function, so they are set to value 1.

# The Error Function

So far, the quantities of a particular nutrient have been deduced, but achieving these exact numbers is practically impossible. There is the need to analyze and penalize the deviation of the real value from the ideal value. Because of the different criteria that the solution is evaluated on, the functions differ in their implementation, but the output is standardized.

Irrespective of the criterion (level) that the error function is applied to, the co-domain is the interval [0, 1], where 1 indicates that the evaluated quantity is the ideal one whereas 0 indicates an unacceptable quantity. Therefore,and if grows, the error function should decrease.

## Level 1 Error Function

The level 1 Error Function analyses one single nutrient, more specifically, a mark based on the deviation of the real value from the ideal one. As mentioned in the previous chapters, it must be present in the meal in quantities close to the ideal ones (see 3.1 Computation of the ideal values for nutrients). There are two possibilities to store the ideal value, either as a single fixed value or as an interval. For each of these, different but similar error functions will be used.

### Fixed value

The Gauss function

Equation The Gauss function

meets the previously mentioned requirements and could be used as an excellent error function for this particular purpose. The b parameter is our ideal value for a specific nutrient and the final function is the following:

Equation the error function for fixed value

C is an additional parameter with the default value 1.

Equation default value for c is 1

For example, if for some nutrient the ideal value is 5 mg per day, the error function is the following

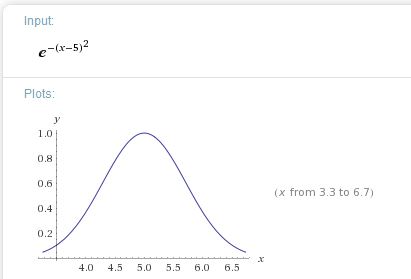


Figure Error function for ideal value 5

In any case,, and, as seen in Figure 2 Error function for ideal value 5, the more distant x from the ideal value is, the lower value the function E takes.

### Interval

The ideal values can belong to an interval, in this case the previous function is adapted to have the value 1 in that interval. For example, the carbohydrates can range from ~200 to ~400 grams, so any value in that interval must be evaluated to 1 (highest).

Suppose that the ideal interval is [a, b], then:

Figure Error margin if the ideal value belongs to an interval

For example, if a = 5 and b = 10, then the plot of the function is in the following figure.

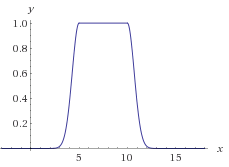


Figure Example plot of the previous function for a = 5 and b = 10

### Scaling the values

The Gauss function takes into consideration the absolute difference between the real and the ideal value whereas the correct criterion is the ratio between the two.

For example, let us take a difference of 2:

It is obvious, that a difference of 2 is much worse in the case of vitamin A as in the case of calories. To avoid this issue, a small adjustment has to be made; the values are scaled to a standard ideal value.

Equation Mapping from unscaled to scaled values

Thus, in principle, the final mark is given by:

For the previous example, the values are scaled in the following manner:

* Let ;

Now, the evaluations seem to be much more valid than in the previous case, the calories have a good mark whereas the vitamin A has a very weak one.

## Level 2 Error Function

The level 2 Error Function evaluates a food item for which the preference of being present or absent in the meal has been specified. In principle, it is the same as the level 1 fitness function, but instead of analyzing nutrients, it analyzes real food items and assesses a fitness value based on the presence (or absence) of ingredients that are on the preference list of the patient in menu.

In this case, the set *C* of components is the set *F* of food items present the solution meal union the *specified list* (see Equation 6). There four possible cases:

1. The user does specifically not want a food item (ideal = 0) and the item is not present in the meal (real = 0). This is marked with 1, because it is as desired.
2. The user wants a food item (ideal = 1) and the item is not present (real = 0). This is marked with a number, because it is not desired to be marked with 1, but also not very critical to be marked with 0. This particular use case is of a lower importance. For example, is a good approximation of “how important” it is.
3. The user does specifically not want a food item (ideal = 0) and the item is present in the meal (real = 1). This is marked with 0, because the meal should avoid that item.
4. The user wants a food item (ideal = 1) and the item is present in the meal (real = 1). This is marked with 1 because it is as desired.

|  |  |  |
| --- | --- | --- |
| Real | Ideal | E(real, ideal) |
| 0 | 0 | 1 |
| 0 | 1 |  |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Table Equation The level 2 Error function

## Level 3 Error Function

The level 3 Error Function evaluates a food package from the cost’s and delivery time’s points of view. In this case, a real quantity less than the ideal one is a positive aspect (e.g. a food package that costs 10$ less than expected), and the error function takes the value 1. If the real quantity is greater than the ideal one, the Gauss function from Equation 8 is applied.

However, at this point, the additional constant *c* (see Equation 9) for the Gauss function is used to decrease the value more dramatically if the real value exceeds the ideal one. In other words, the constant *c* controls the slope of the curve: a smaller value means a more dramatic change.

The following two illustrations show the influence of the constant *c*.

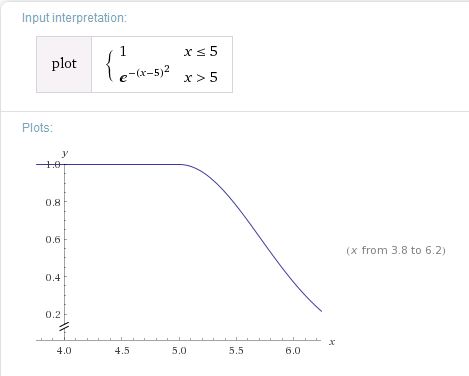


Figure Level 3 Error function with c = 1

It can be seen that the slope is not dramatic.

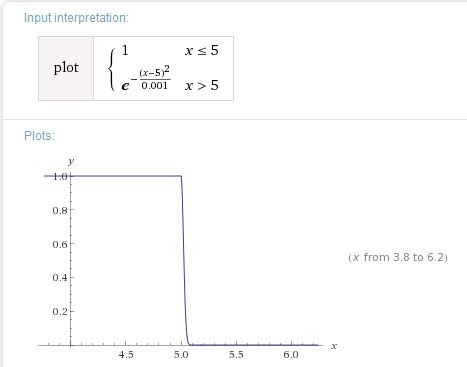


Figure Level 3 Error function with c = 0.001

Hence, the level 3 Error function is:

Equation Level 3 Error function

# The fitness function template

Because the fitness value of a solution depends on more than one criterion, i.e. on different levels (see Table 1: Subjects of each level of the fitness function), the fitness function template will be applied for each criterion in particular; there are 3 criteria (levels) hence applied 3 times and then combined. The template function consists of a weighted average between all the components of a level. The ideal functions and weights for each level have been described in the previous chapters (3. The “ideal” functions for components and 4. The weights for components) and are applied in this chapter.

## Definition

Let *c* be a component from the set *C* of components, *w(c)* the weight of component *c*, real*(c)* the quantity of *c*, *ideal(c)* the ideal quantity of *c*, *E(q(c), ideal(c))* the error function which penalizes any deviation of *q(c)* from the ideal value of *c* and *X* the set of solutions, then:

Equation The fitness function template

Since, because is basically just a weighted average.

## Level 1 fitness function: nutrients

The level 1 fitness function takes a solution as input and provides a value between 0 and 1 to evaluate it from the nutrients’ point of view. It assures that each nutrient comes in the appropriate quantity. The previously mentioned set *C* of components is in this case the set of nutrients, therefore if *N* is the set of nutrients, then:

Equation Fitness function level 1

Where the function *w* is described in chapter 3.1, the function *ideal* in chapter 4.1 and in chapter 5.1.

The generated meals have to contain quantities of nutrients as close as possible to the desired ones, otherwise the evaluation of the bad nutrients will be marked lower, and as consequence of the weighted average, the whole fitness to be marked lower.

Two examples are explained below:

1. A meal containing twice the recommended fat will be scored lower because:

* function E(x, 2x) will have a low value

2. A meal containing sugar for a diabetic patient will also be scored very low because:

* the weight for sugar will be very high for diabetic patients –500;
* the ideal value for sugar is very low (0.001 g) which causes a high level of sugar in the meal the evaluation (the E function) of the sugar level to be very low;
* the weight of the sugar level will highly affect the weighted average in a negative manner;

## Level 2 fitness function: ingredients

The level 2 fitness function takes a solution as input and provides a value between 0 and 1 to evaluate it from the ingredients’ point of view. It assures that each ingredient in the *like* *list* is present in the meal and each ingredient in the *dislike list* is not present in the meal (see Equation 6 The specified list). In this case, the set *C* of components is the set *F* of food items present the solution meal union the *specified list* (see Equation 6).

Equation Fitness function level 2

## Level 3 Fitness Function: Cost and Delivery time

The Level 3 Fitness Function takes into consideration only the two aspects: cost and delivery time. Thus, the set of components.

Equation Fitness function level 3

## The top level fitness function

The top level fitness function is the one that gives the final mark to a solution by combining the previously presented smaller-level fitness functions. The fitness function template is here also applied, therefore, it is just a weighted average between level 1, 2 and 3 fitness functions.