Optimizing breeding schemes //

Manual / Index selection

Using a desired gains index for multi-trait selection with BLUPs

- This manual describes how to calculate and implement multi-trait selection for quantitative traits using a desired gains index with BLUPs.
- Published on 01/11/2023
- Excellenceinbreeding.org/toolbox/tools/eib-breeding-scheme-optimization-manuals

CGIAR | INITIATIVE ON Accelerated Breeding

cgiar.org/initiative/accelerated-breeding

Using a desired gains index for multi-trait selection with BLUPs

Author

Christian R. Werner, Dr / c.werner@cgiar.org
Quantitative Geneticist, Accelerated Breeding Initiative

Editor

Dorcus Gemenet, Dr / d.gemenet@cgiar.org
Quantitative Genetics Lead, Accelerated Breeding Initiative

Contributors

Giovanny Eduardo Covarrubias-Pazaran
Quantitative Genetics and Biometrics Lead, IRRI

Keith Gardner, Dr
Quantitative Geneticist, CIMMYT

Marlee Labroo, Dr
Principal Traited Genome Design Scientist, Bayer CropScience

CGIAR

INITIATIVE ON
Accelerated Breeding

Table of Contents

Introduction 1

1. **Define the breeding target and calculate the vector of desired gains (*d*)** 2
    - 1.1 The breeding objective is defined in terms of a Target Product Profile 4
    - 1.2 The breeding objective is defined in terms of standard deviations gain 4
    - 1.3 The breeding objective is defined in terms of percent gain 5
2. **Calculate the weights (*w*) for the desired gains index** 6
3. **Calculate the index values for all individuals under selection** 6
4. **Use the software DESIRE to tweak the desired gains vector *d*** 7
5. **Use the software DESIRE to derive the desired gains vector without an a priori breeding target** 14

Introduction

The desired gains index is an effective method for improving multiple quantitative traits at the same time. It allows breeders to specify the gain they want to make in each trait and achieve it within a minimum number of generations. The desired gains index does not require "relative economic weights", which are usually difficult to get for some or all of the traits of interest. Instead, index weights are calculated based on the specified desired gains (i.e., the breeder does not set the weights).

The amount of gain that a breeder would like to see in each trait (i.e., the desired gains) can be expressed in different ways. Breeders can, for example, define a Target Product Profile to set the minimum performance standards for a future variety. Alternatively, the breeding target can be defined in terms of standard deviations gain or percent gain per trait. We will demonstrate all three approaches in this document.

The desired gains index can be used with raw phenotypes, BLUEs, and BLUPs. Here, we explain how the desired gains index is generated with BLUPs, while we cover the desired gains index with BLUEs and raw phenotypes in a separate manual.

In this manual, we demonstrate how to:

1. Define the breeding target and calculate the vector of desired gains (d).
2. Calculate the weights (w) for the desired gains index.
3. Calculate the index values for all individuals under selection.
4. Use the software DESIRE to tweak the desired gains vector d (optional).
5. Use the software DESIRE to derive the desired gains weights (w) without an a priori breeding target.

1. Define the breeding target and calculate the vector of desired gains (*d*)

All we need to generate a desired gains index is 1) the BLUPs for all quantitative traits of interest, and 2) a clearly defined breeding target. We use a data frame with BLUPs of 500 bean genotypes for yield (YLD), flowering time (FLT), and cooking time (CKT). The BLUPs were calculated using a multi-trait ridge-regression mixed model (rrBLUP) for genomic selection.

```
dim(blups) # Dimensions of the data frame: 500 genotypes, 3 traits.
```

```
## [1] 500   3
```

```
head(blups, 10)
```

```
##             YLD      FLT      CKT
## ID_1  0.7229723 34.20606 39.87525
## ID_2  0.8766339 33.50479 38.85036
## ID_3  0.8309414 34.85511 39.79975
## ID_4  0.6789020 35.16362 39.78710
## ID_5  0.9335854 34.27582 39.15532
## ID_6  0.8813646 34.44118 40.35914
## ID_7  0.7987138 33.72891 41.22677
## ID_8  0.6120280 34.88984 40.14626
## ID_9  1.0282515 34.93828 39.70509
## ID_10 0.8755989 34.10283 39.86577
```

Our breeding objectives are:

- Increase yield (t/ha),
- Stabilise flowering time at 35 days after sowing, and
- Reduce cooking time (min).

First, we calculate the population means and the variance-covariance matrix of the three traits.

```
pop_means <- colMeans(blups)
pop_means
```

```
##        YLD        FLT        CKT
##  0.7780435 34.4938833 40.0361170
```

```
covar_ebv <- cov(blups)
covar_ebv
```

```
##            YLD       FLT        CKT
## YLD 0.01879885 0.0265010 0.03090733
## FLT 0.02650100 0.5247061 0.17696505
## CKT 0.03090733 0.1769651 0.88476853
```

NOTE: The R stats function cov() calculates the sample variances of and covariances between the traits (i.e., the denominator *n - 1* is used). However, since we generate the selection index for a closed set of individuals in our breeding population, the population variances and covariances should be calculated instead (the denominator *n* is used). Therefore, we recommend to use a function such as popVar() in 'AlphaSimR' (although the differences may be small at large population sizes).

```
AlphaSimR::popVar(blups)
```

```
##            [,1]      [,2]       [,3]
## [1,] 0.01876125 0.0264480 0.03084552
## [2,] 0.02644800 0.5236567 0.17661112
## [3,] 0.03084552 0.1766111 0.88299900
```

We can now use the means and the variance-covariance matrix[1] to derive the vector of desired gains (*d*) and calculate the index weights (*w*).

---

[1] Note that using the variance-covariance matrix of the multivariate BLUPs is only an approximation that works well in most cases. For further details and a more precise calculation of the BLUP variance-covariance matrix, see: Werner CR, Gardner KA, Tolhurst DJ (2024). Reviving the Desired Gains Index: An optimal solution for parent selection in public plant breeding programs.

1.1 The breeding objective is defined in terms of a Target Product Profile

The Target Product Profile describes the performance of an optimal variety for all quantitative traits of interest in absolute values. The vector of desired gains (*d*) is simply the difference between your Target Product Profile and the population trait means.

```
TPP_1 <- c(1.5, 35, 37)      # Target Product Profile in absolute values.
```

```
d_1 <- TPP_1 - pop_means    # Desired gains vector in absolute values.
d_1
```

```
##        YLD        FLT        CKT
##  0.7219565  0.5061167 -3.0361170
```

1.2 The breeding objective is defined in terms of standard deviations gain

If we define our breeding objective in terms of standard deviations gain to be made, we eventually need to turn this objective into absolute values. Hence, this approach is not too different from using a Target Product Profile, but some may find it more intuitive to define the breeding target.

Let us assume we want to improve yield by 2 standard deviations, fix flowering time at 35 days after sowing, and reduce cooking time by 1 standard deviation. We then derive our desired gains vector (*d*) in the following way:

```
target_in_SD <- c(2, 0, -1)       # Desired gains in standard deviations for YLD and CKT.
SD <- sqrt(diag(covar_ebv))       # Extract standard deviations.
d_2 <- target_in_SD * SD          # Desired gains for YLD and CKT in absolute values.
d_2[2] <- 35 - pop_means[2]       # Target value for FLT fixed at 35.
```

Note that we defined gains in terms of standard deviations only for yield and cooking time. Flowering time was fixed at an absolute value of 35 days after sowing.

The Target Product Profile resulting from our breeding objective is then the sum of the population means and the desired gains vector:

```
TPP_2 <- pop_means + d_2     # Target Product Profile.
TPP_2
```

```
##          YLD        FLT        CKT
##    1.052261  35.000000  39.095496
```

1.3 The breeding objective is defined in terms of percent gain

Another intuitive way to define the breeding objective is in terms of percent gain per trait to be made. This approach is very similar to using standard deviations, since the gains expressed in percent also have to be turned into absolute values. Let's assume now that we want to improve yield by 20% and reduce cooking time by 10%. Again, we want to fix flowering time at 35 days after sowing.

```
target_in_percent <- c(0.2, 0, -0.1)        # Improve YLD by 20% and reduce CKT by 10%.
d_3 <- pop_means * target_in_percent        # Calculate desired gains (absolute values).
d_3[2] <- 35 - pop_means[2]                 # Fix FLT at 35 days after sowing.
```

The Target Product Profile resulting from our breeding objective is the sum of the population means and the desired gains vector:

```
TPP_3 <- pop_means + d_3     # Target Product Profile.
TPP_3
```

```
##          YLD          FLT          CKT
##    0.9336522  35.0000000  36.0325053
```

2. Calculate the weights (*w*) for the desired gains index

We now can derive the weights (*w*) for the desired gains index[2]. Here, we use the vector of desired gains *d_1* that we got using the Target Product Profile as defined above. However, the desired gains vectors *d_2* and *d_3* could also be used instead.

```
w <- solve(covar_ebv) %*% d_1     # desired gains index weights w -
                                  # to achieve the Target Product Profile.
```

3. Calculate the index values for all individuals under selection

We calculate the index values for the 500 bean genotypes by multiplying the BLUPs with the desired gains weights (*w*).

```
index_values <- blups %*% w
head(index_values, 6)
```

```
##             [,1]
## ID_1  -158.6477
## ID_2  -146.5280
## ID_3  -153.0310
## ID_4  -159.9012
## ID_5  -145.1799
## ID_6  -153.7048
```

The genotypes with the highest index values are our best parental candidates to reach the Target Product Profile. It is important that we do ***not*** combine the index values with any other selection criteria, such as individual trait measurements.

---

[2] Note that using the variance-covariance matrix of the multivariate BLUPs is only an approximation that works well in most cases. For further details and a more precise calculation of the BLUP variance-covariance matrix, see: Werner CR, Gardner KA, Tolhurst DJ (2024). Reviving the Desired Gains Index: An optimal solution for parent selection in public plant breeding programs.

```
index_values <- data.frame(ID = rownames(index_values),
                           Index = as.vector(index_values))

index_values <- index_values[order(-index_values$Index), ]

parents <- index_values[1:6, ]
parents  # The six genotypes with the highest index values.
```

```
##          ID     Index
## 22    ID_22 -134.4110
## 194  ID_194 -135.6409
## 350  ID_350 -138.0909
## 51    ID_51 -138.1836
## 136  ID_136 -140.3112
## 441  ID_441 -140.5535
```

4. Use the software DESIRE to tweak the desired gains vector *d* (optional)

We can use the stand-alone software DESIRE to investigate alternative Target Product Profiles. This might help with the identification of slightly changed, preferable trait improvement ratios. We will use our present Target Product Profile as a starting point. Then, alternative Target Product Profiles can be easily explored with DESIRE's graphical user interface.

The software can be downloaded from here: https://bkinghor.une.edu.au/desire.htm

DESIRE, however, cannot directly read in our present Target Product Profile as a starting point. Therefore, we use a mathematical relationship between the desired gains index and the Smith-Hazel index, which allows us to express the desired gains weights (w) in terms of economic weights (a), and vice versa. In fact, when using BLUPs, this relationship reduces to $\boldsymbol{w} = \boldsymbol{a}$ (we omit the proof here).

It is important to understand that these "economic weights" we derive using w have no real economic meaning. They simply represent the weights that we need to assign to the index traits in order to get our desired improvement ratio.

The DESIRE input text file requires 1) the "economic values" of the traits, 2) the standard deviations, and 3) the correlation matrix. We can extract all this information from our data frame.

```
round(w, 3)              # Desired gains weights (w) => starting "economic weights"
```

```
##          [,1]
## YLD  46.327
## FLT   0.352
## CKT  -5.120
```

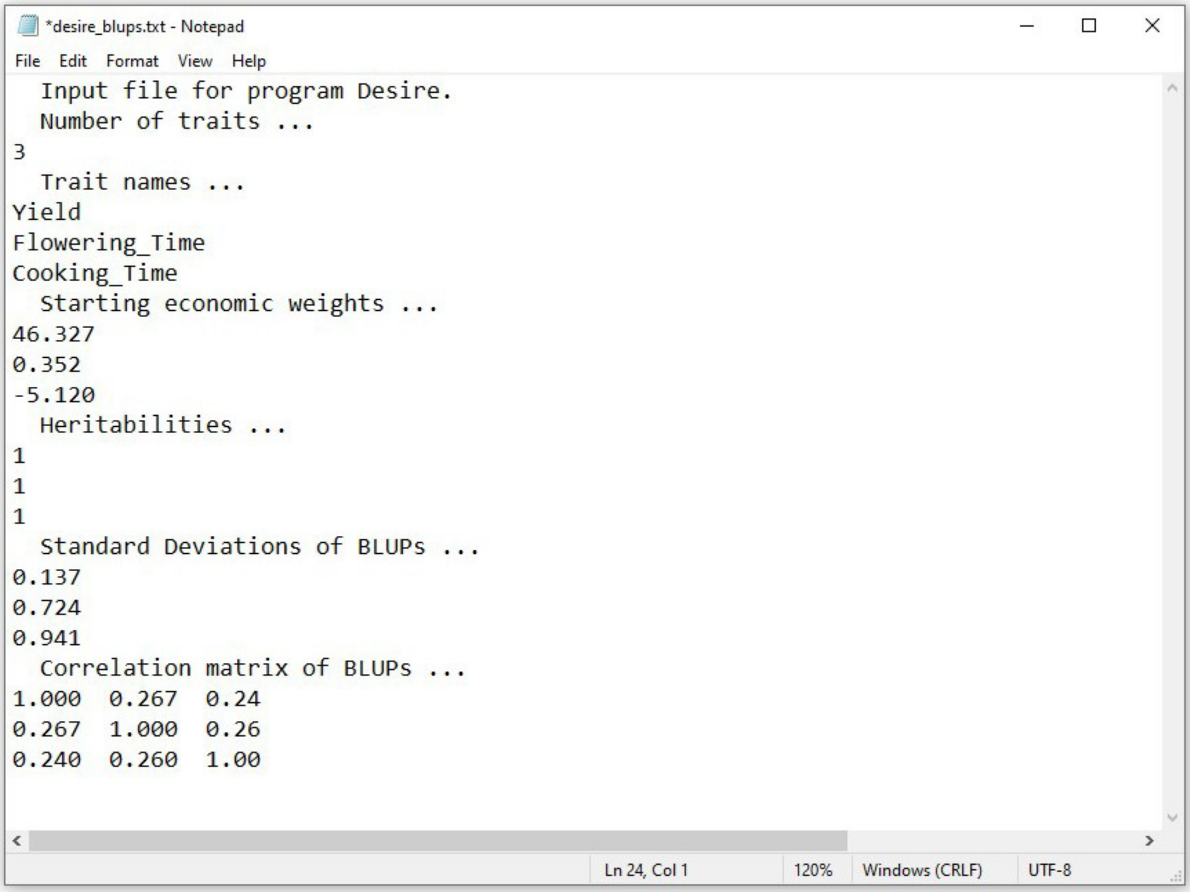
```
round(sqrt(diag(covar_ebv)), 3)       # Standard deviations of the BLUPs.
```

```
##    YLD   FLT   CKT
## 0.137 0.724 0.941
```

```
round(cor(blups), 3)                  # BLUP correlation matrix of the three traits.
```

```
##       YLD   FLT  CKT
## YLD 1.000 0.267 0.24
## FLT 0.267 1.000 0.26
## CKT 0.240 0.260 1.00
```

The input file for DESIRE has to be prepared manually by inserting the values we obtained above. Note that we set all trait heritabilities to 1 since we are using BLUPs (BLUPs have already been shrunk towards the population mean based on the heritability. We don't want DESIRE to shrink them again).

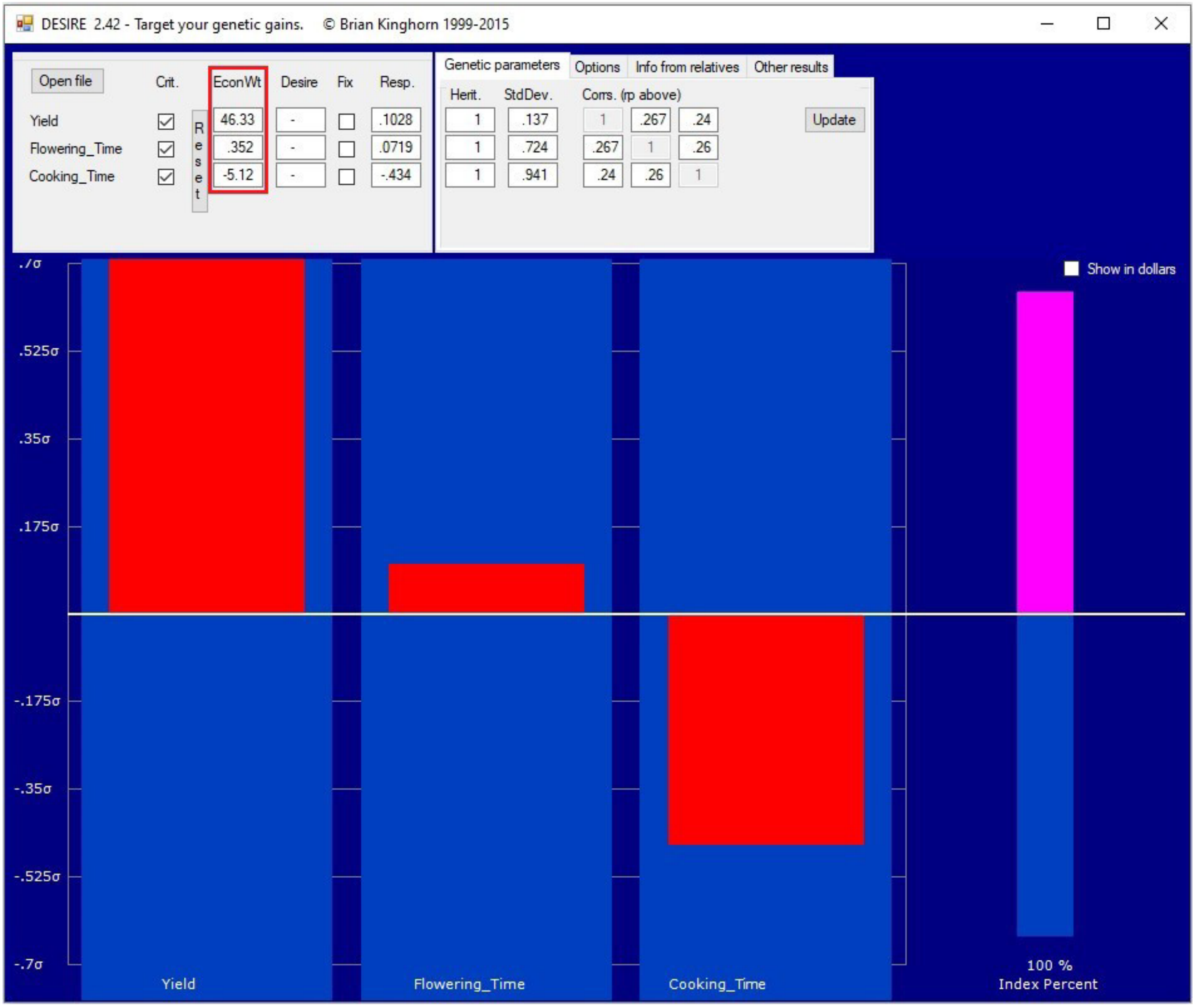


```
  Input file for program Desire.
  Number of traits ...
3
  Trait names ...
Yield
Flowering_Time
Cooking_Time
  Starting economic weights ...
46.327
0.352
-5.120
  Heritabilities ...
1
1
1
  Standard Deviations of BLUPs ...
0.137
0.724
0.941
  Correlation matrix of BLUPs ...
1.000  0.267  0.24
0.267  1.000  0.26
0.240  0.260  1.00
```

We now run the *Desire.exe* executable and load the input file. The program provides various options of which we will only need a subset to tweak our Target Product Profile.

The column “EconWt” contains the starting “economic values”, which are equal to the desired gains weights (w). The red bar plots represent the improvement ratio coming from our Target Product Profile in standard deviations.

The column “Resp” shows the expected response to selection in absolute values. These values will be a fraction or multiple of the desired gains (d) we want to make. Their scaling depends on the selection intensity and the generation interval. The two parameters can be set under “Options” to get a prediction of the gain expected in the next generation based on the breeder’s equation. The desired gains vector, on the other hand, has no specific time horizon associated with it. For now, we leave the selection intensity and the generation interval at the default values of 1.

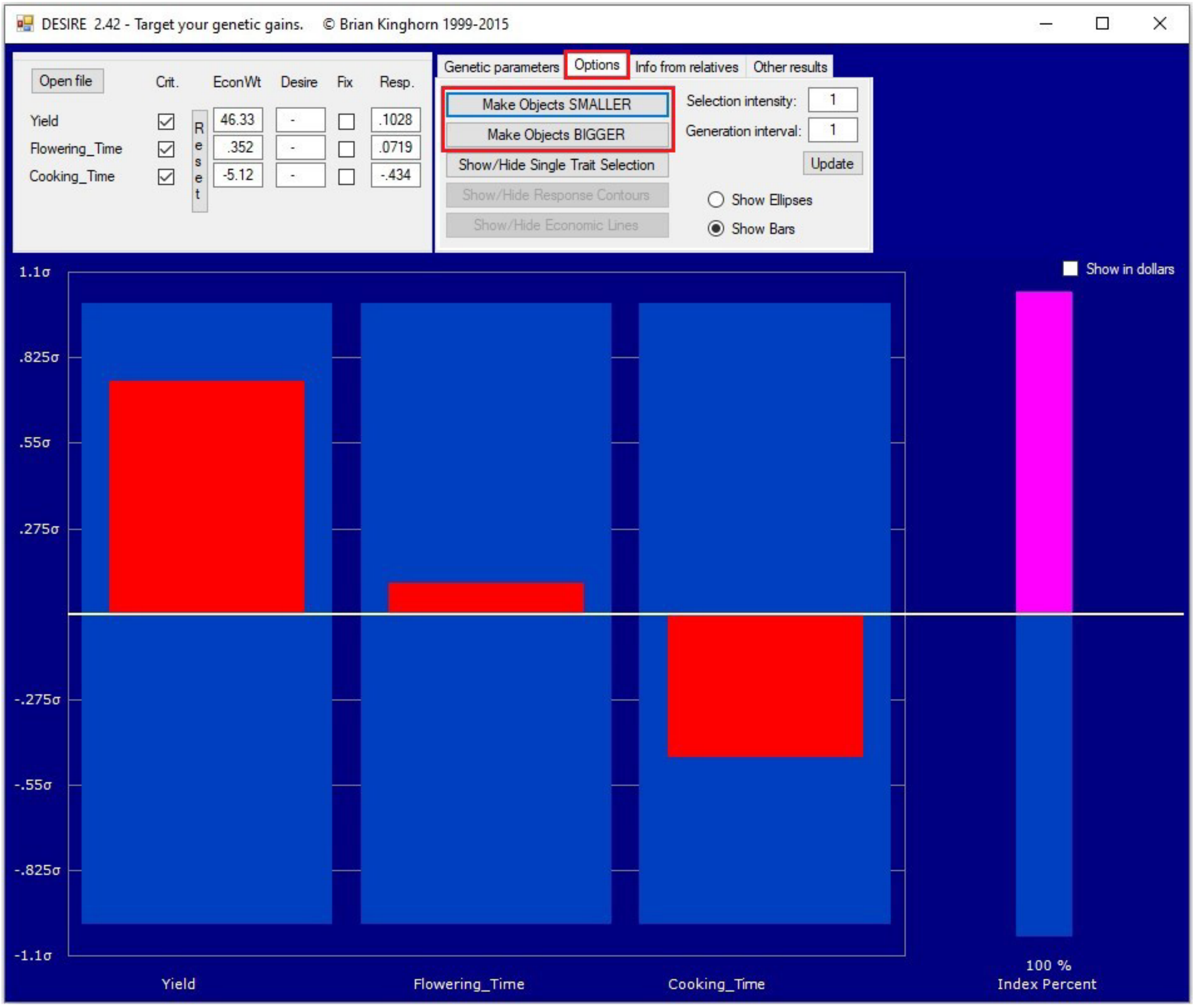


DESIRE 2.42 - Target your genetic gains. © Brian Kinghorn 1999-2015

	Crit.	EconWt	Desire	Fix	Resp.
Yield	☑	46.33	-	☐	.1028
Flowering_Time	☑	.352	-	☐	.0719
Cooking_Time	☑	-5.12	-	☐	-.434

Herit.	StdDev.	Corrs. (rp above)		
1	.137	1	.267	.24
1	.724	.267	1	.26
1	.941	.24	.26	1

First of all, we click on the “Options” rider and adjust the the size of the red bars using “Make Objects SMALLER” or “Make Objects BIGGER”. If desired, the selection intensity and the generation interval can also be adjusted here (confirm with “update”).



DESIRE 2.42 - Target your genetic gains. © Brian Kinghorn 1999-2015

	Crit.	EconWt	Desire	Fix	Resp.
Yield	☑	46.33	-	☐	.1028
Flowering_Time	☑	.352	-	☐	.0719
Cooking_Time	☑	-5.12	-	☐	-.434

Selection intensity: 1

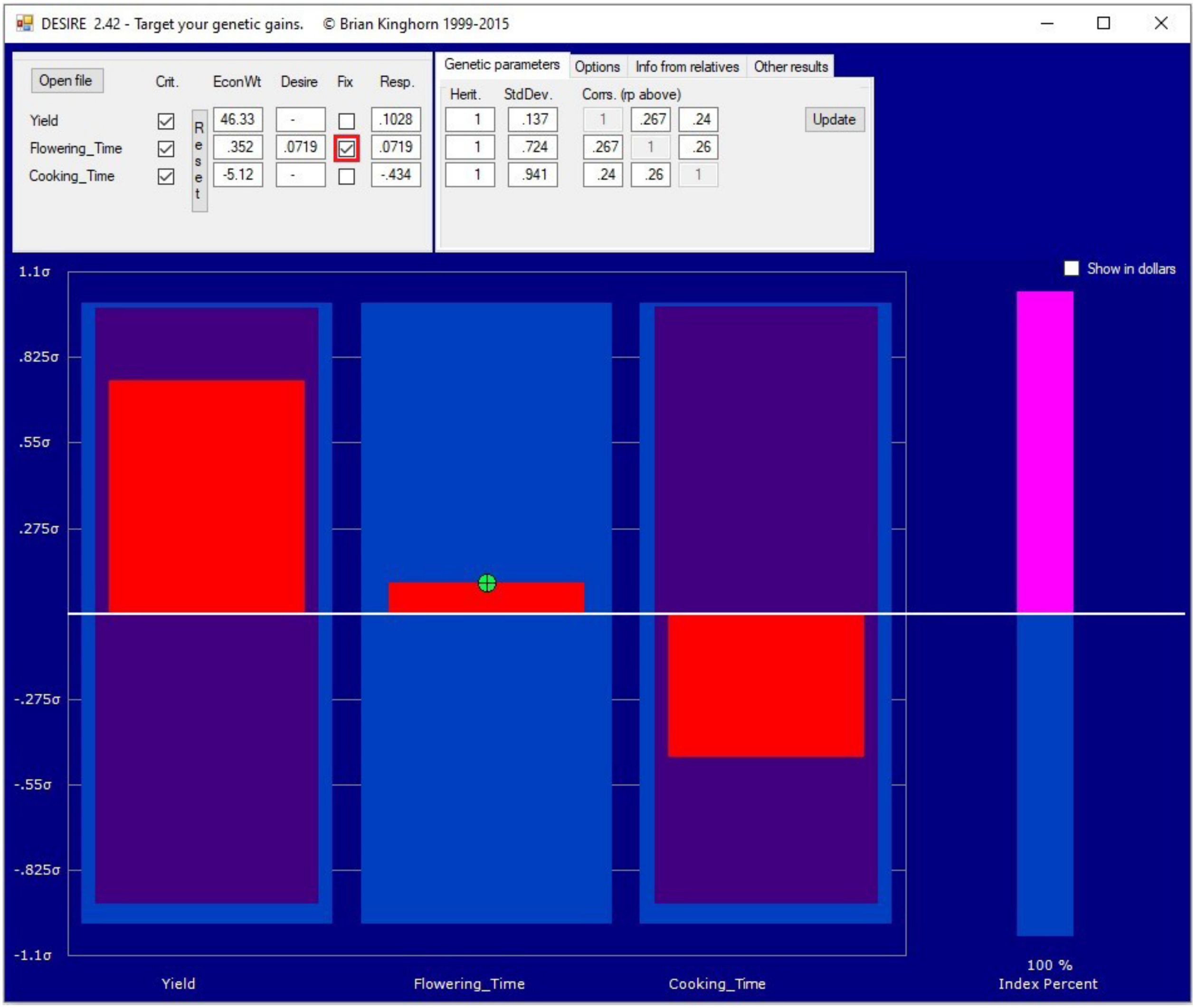
Generation interval: 1

○ Show Ellipses

● Show Bars

☐ Show in dollars

Next, before we change the height of any bar, we need to fix Flowering Time. This is necessary to achieve the desired target of 35 days after sowing as defined above.



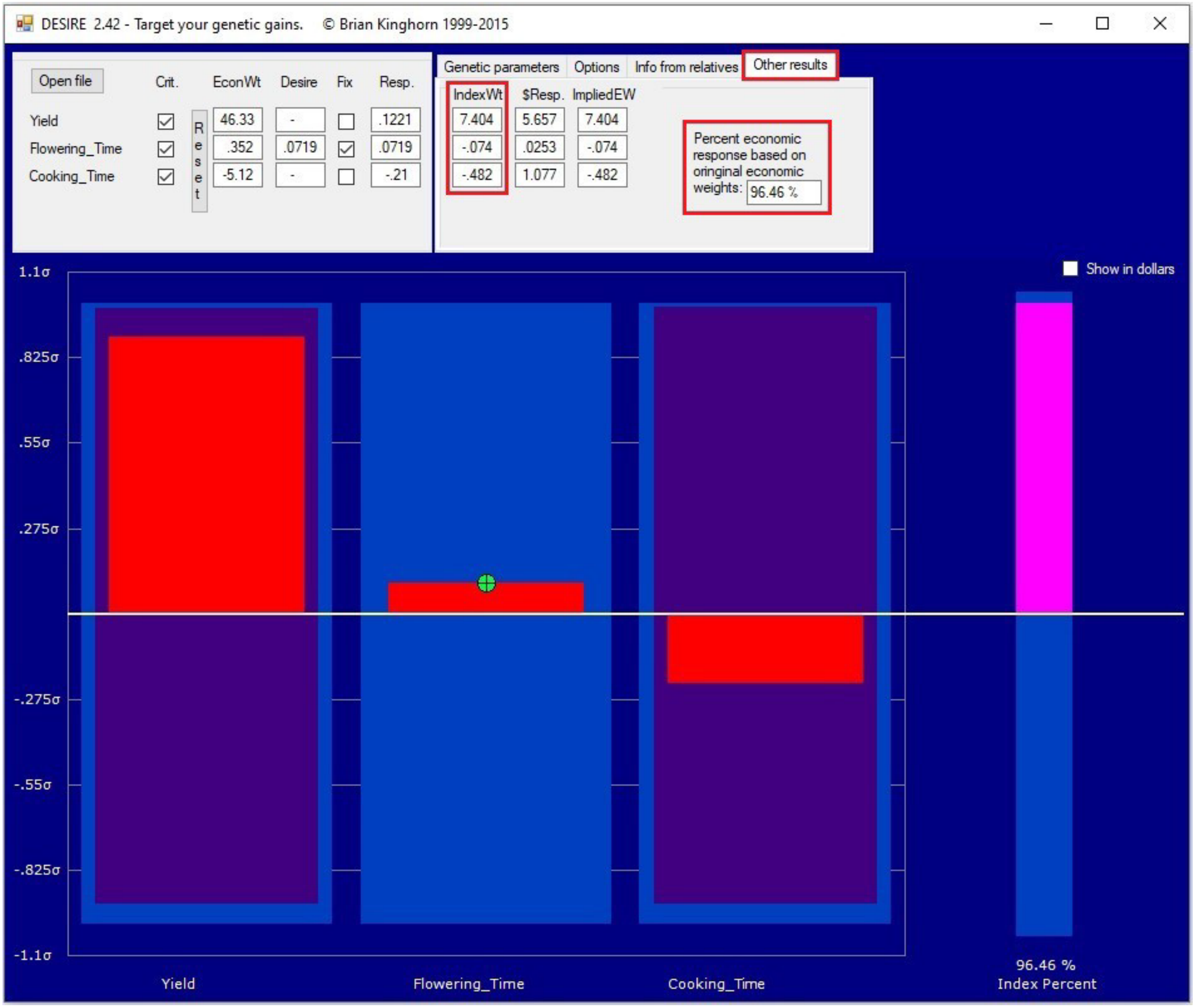
	Crit.	EconWt	Desire	Fix	Resp.
Yield	☑	46.33	-	☐	.1028
Flowering_Time	☑	.352	.0719	☑	.0719
Cooking_Time	☑	-5.12	-	☐	-.434

Herit.	StdDev.	Corrs. (rp above)		
1	.137	1	.267	.24
1	.724	.267	1	.26
1	.941	.24	.26	1

Now, we can explore alternative improvement ratios for yield and cooking time by clicking on the red bars. When we change the Target Product Profile, DESIRE automatically takes the genetic correlations between the three traits into account. Therefore, we will observe a change in cooking time when we change yield, and vice versa.

We decide to increase yield, which results in a penalty on cooking time reduction.

After we have found our new improvement ratio, we extract the new desired gains weights from the column "IndexWt" in "Other results" and multiply them with the BLUPs in our data frame. The genotypes with the highest index values are the best parental candidates to reach our new Target Product Profile.


DESIRE 2.42 - Target your genetic gains. © Brian Kinghorn 1999-2015

	Crit.	EconWt	Desire	Fix	Resp.
Yield	☑	46.33	-	☐	.1221
Flowering_Time	☑	.352	.0719	☑	.0719
Cooking_Time	☑	-5.12	-	☐	-.21

IndexWt	$Resp.	ImpliedEW
7.404	5.657	7.404
-.074	.0253	-.074
-.482	1.077	-.482

Percent economic response based on oringinal economic weights: 96.46 %

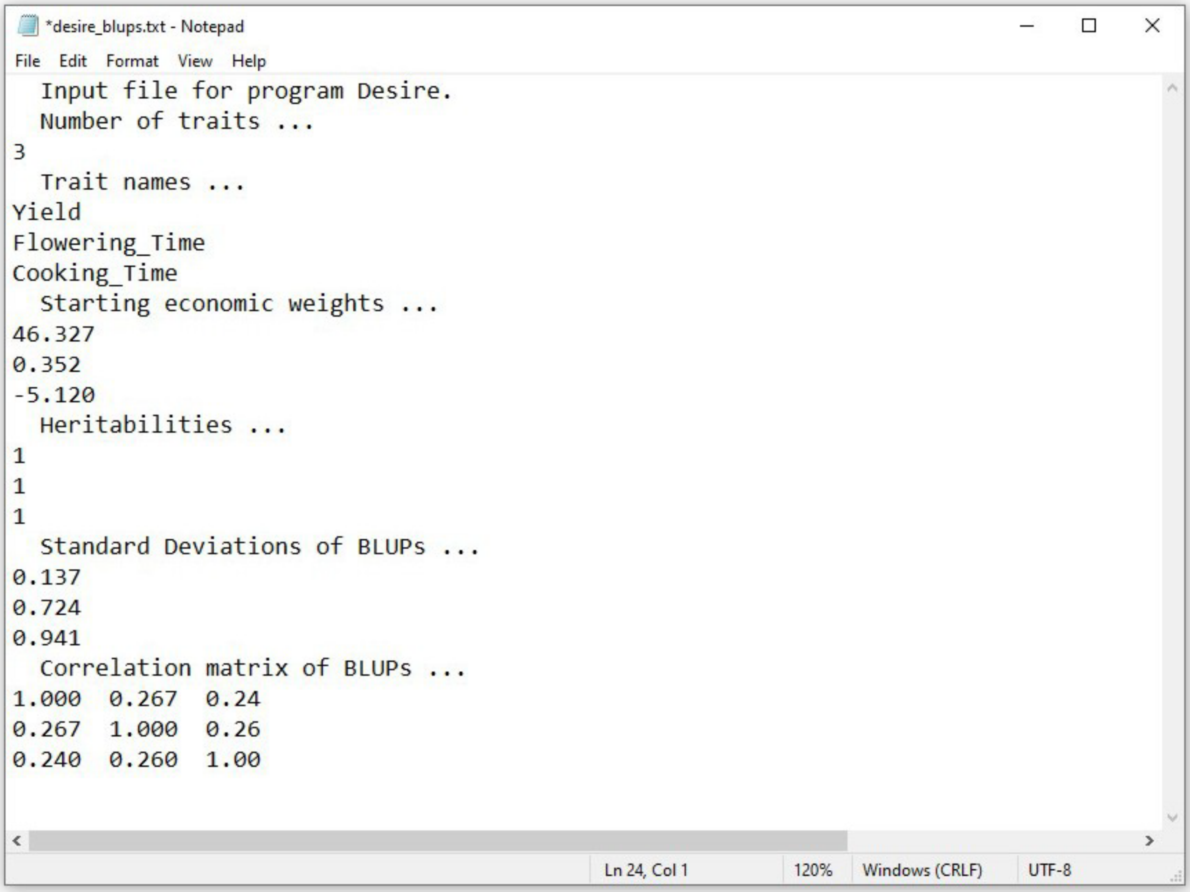
☐ Show in dollars

96.46 %
Index Percent

NOTE: the total economic value of our new target product profile, indicated by the pink bar ("Index Percent"), has slightly decreased. This shall not cause us any concern and can be ignored. DESIRE calculates the total economic value of the Target Product Profile based on the economic values of the traits we provided with the input file. It assumes that these are the true economic values, and every change in the improvement ratio will reduce the total economic value of the Target Product Profile. However, the economic values we provided for the three traits have no real economic meaning, and the changes in the pink bar are irrelevant for our purpose.

5. Use the software DESIRE to derive the desired gains vector without an a priori breeding target

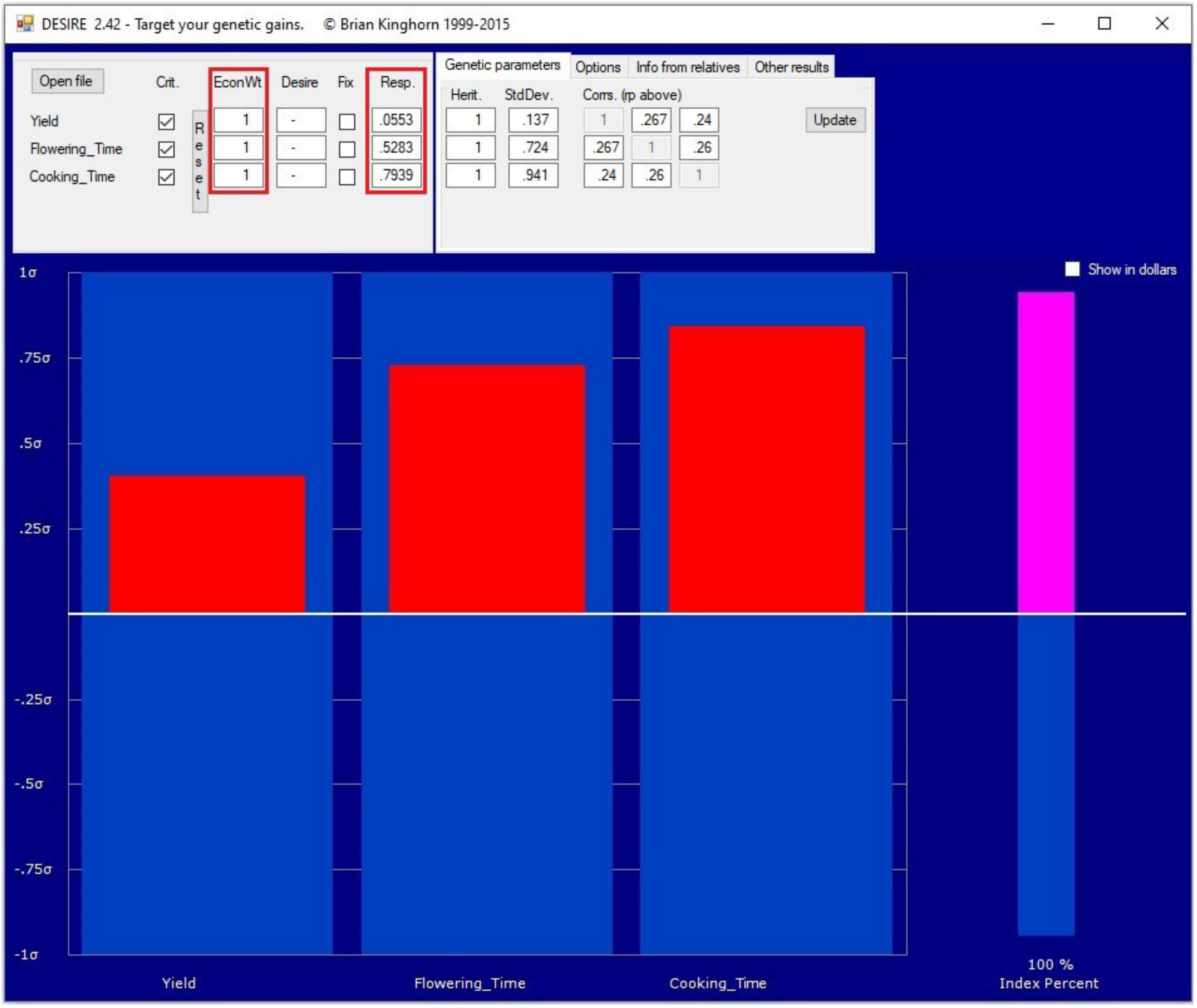
If we have no Target Product Profile to start with, we can try to identify our breeding target by exploring our data. Therefore, we simply set all the "economic values" to 1 in the DESIRE input file.



```
  Input file for program Desire.
  Number of traits ...
3
  Trait names ...
Yield
Flowering_Time
Cooking_Time
  Starting economic weights ...
46.327
0.352
-5.120
  Heritabilities ...
1
1
1
  Standard Deviations of BLUPs ...
0.137
0.724
0.941
  Correlation matrix of BLUPs ...
1.000  0.267  0.24
0.267  1.000  0.26
0.240  0.260  1.00
```

We can now explore alternative Target Product Profile by clicking on the red bars. The column "Resp" shows the expected response to selection in the next generation, assuming a selection intensity of 1. We adjust the traits until we achieve a satisfactory improvement ratio, which defines our Target Product Profile. This expected response to selection then is our desired gains vector (*d*).

Some users may find it more intuitive to set a Target Product Profile for a generation interval longer than one year. In this case, the generation interval can be increased under "Options", and the expected response to selection will be scaled by the number of generations.



	Crit.	EconWt	Desire	Fix	Resp.
Yield	☑	1	-	☐	.0553
Flowering_Time	☑	1	-	☐	.5283
Cooking_Time	☑	1	-	☐	.7939

Herit.	StdDev.	Corrs. (rp above)		
1	.137	1	.267	.24
1	.724	.267	1	.26
1	.941	.24	.26	1

After we have found our desired response to selection, we extract the desired gains weights from the column "IndexWt" in "Other results" and multiply them with the BLUPs in our data frame. The genotypes with the highest index values are the best parental candidates to reach our new Target Product Profile.

	Crit.	EconWt	Desire	Fix	Resp.
Yield	☑	1	-	☐	.1185
Flowering_Time	☑	1	-	☐	.1164
Cooking_Time	☑	1	-	☐	-.262

IndexWt	$Resp.	ImpliedEW
7.162	.1185	7.162
.0482	.1164	.0482
-.556	-.262	-.556

Percent economic response based on oringinal economic weights: -1.98 %

NOTE: the total economic value of our new target product profile, indicated by the pink bar ("Index Percent"), has decreased drastically. This shall not cause us any concern and can be ignored. DESIRE calculates the total economic value of the Target Product Profile based on the economic values of the traits we provided with the input file. It assumes that these are the true economic values, and every change in the improvement ratio will reduce the total economic value of the Target Product Profile. However, the economic values of 1 that we provided for the three traits have no real economic meaning, and the changes in the pink bar are irrelevant for our purpose.