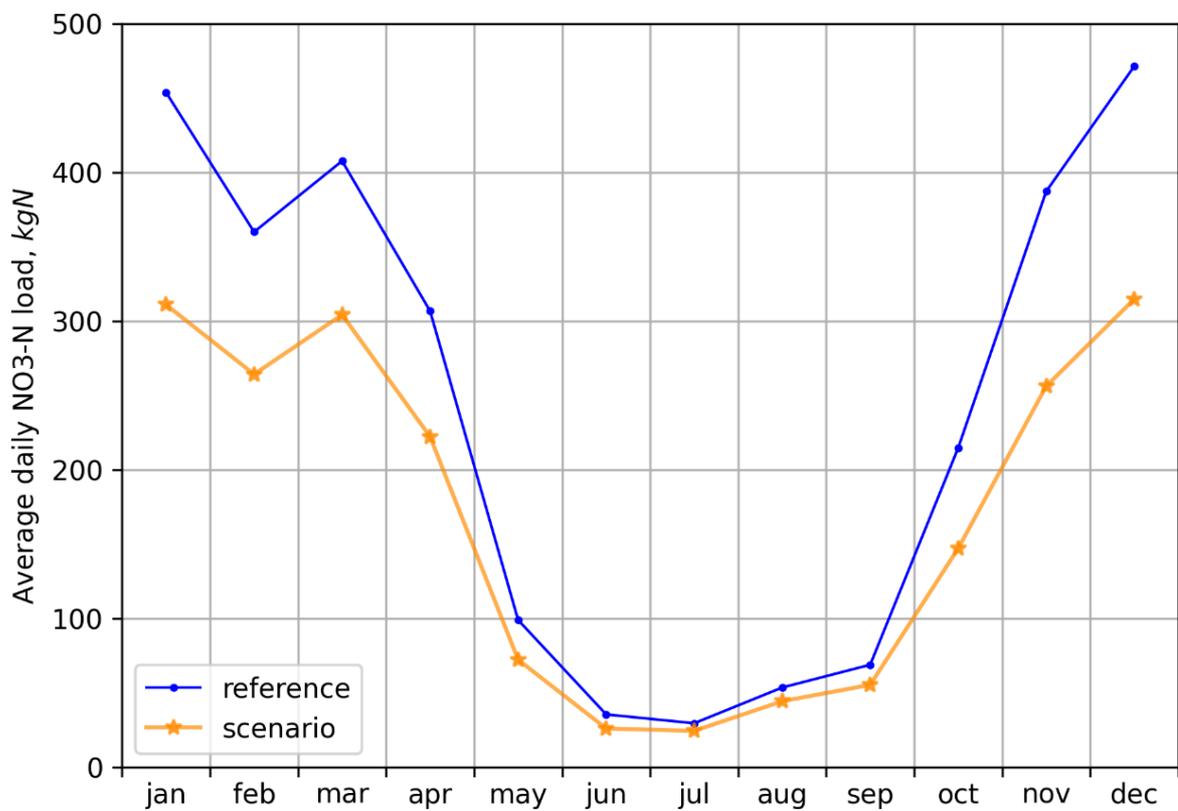


TECHNICAL REPORT ON DEVELOPMENT AND EMPLOYMENT OF MEASURE OPTIMISATION

DELIVERABLE R7



Prepared within the LIFE GoodWater IP Action C1: “Development of the water quality and quantity system for the territory of Latvia”

Rīga, 2023

Technical report on development and employment of measure optimisation

Uldis Bethers, Juris Seņņikovs, Pēteris Bethers, Andrejs Timuhins, SIA “Procesu analīzes un izpētes centrs”

Recommended citation: PAIC, 2023. Technical report on development and employment of measure optimisation. LIFE GoodWater IP Deliverable R7, Rīga, 48p.

Document prepared in the frame of the integrated project “Implementation of River Basin Management Plans of Latvia towards good surface water status” (LIFE GOODWATER IP, LIFE18 IPE/LV/000014), project has received funding from the LIFE Programme of the European Union and the State Regional Development Agency.

The information reflects only the LIFE GoodWater IP project beneficiaries’ view and the European Climate, Infrastructure and Environment Executive Agency is not responsible for any use that may be made of the information contained therein.

© PAIC, 2023

Document versioning sheet	
Document version number	v 1.0
Document planned elaboration date	12.2023
Document elaboration date	12.2023
Document actual version elaboration date	12.2023
Project activity/sub-activity number	C1



Summary

This document is a technical report on the development of the script library for optimisation of the measures aimed at improvement of the water quality. Report contains the description of these scripts and illustration of their employment.

Document is written in English, it contains 48 pages, 18 figures, 10 tables and 5 references.



Table of contents

Summary	3
Introduction	6
1. Library of scenario scripts	7
1.1. General principles	7
1.2. Application of HRU level measures	7
1.3. Point sources	9
1.4. Land use change	11
1.4.1. Land use conversion by land use groups	11
1.4.2. Land use conversion by land use code	12
1.5. Catch crops	13
1.5.1. Catch crops	13
1.5.2. Planting of winter crops	14
1.5.3. Plant cover in winter	16
1.6. Tillage	18
1.6.1. No-plough technology	18
1.6.2. Substituting autumn ploughing with spring ploughing	19
1.6.3. Postponing a sod ploughing to late autumn	21
1.7. Fertilization amount	22
1.8. Fertilization timing	23
1.9. Fertilization technique	24
1.10. Enlarged grassed waterways and buffer zones	25
1.10.1. Grassed waterways	25
1.10.2. Filter strips	26
1.11. Tile drainage parameters	27
1.12. Sedimentation ponds and wetlands	28
1.13. Pesticides	29
1.14. Land use update	30
1.15. Managed drainage	31
2. Employment of scripts	33
2.1. Zaņa catchment	33
2.2. Fertilization timing	36
2.3. Advanced fertilization techniques	38
2.4. Fertilization reduction measure	39
2.5. Filter strips and grassed waterways	40
2.6. Land use change	41
2.7. Managed drainage	43
2.8. No-plough measure	44



2.9.	Substituting autumn ploughing with spring ploughing	45
2.10.	Catch crops	46
2.11.	Plant cover in winter	47
	References	48



Introduction

This document is a Technical report on the development of the Python script the script library for optimisation of the measures aimed at improvement of the water quality. It is Deliverable R7 within LIFE Goodwater IP project, activity C1.

The developed scripts may be used with the water quantity and quality modelling system delivered within Deliverable R5 “Complete modelling system – data, models, results” PAIC (2023b). This modelling system comprises of the

- data prepared within Deliverable R1 – Data base for model implementation, PAIC (2020);
- water quantity and quality modelling tools described in Deliverable R3 – Modelling results for Reference and Baseline scenarios, PAIC (2022b);
- model setups of the calibrated and validated modelling system for the water quantity and quality in the territory of Latvia delivered as Deliverable R2 – Calibrated and validated modelling system in PAIC (2022a);
- modelling results delivered as Deliverable R2 – Calibrated and validated modelling system in PAIC (2022a) and described in the Deliverable R4 – Documentation of development, calibration, validation and results of SWAT modelling system in PAIC (2023a).

The Python scripts which realize individual water quality improvement measures in SWAT+ are documented in Chapter 1.

The employment of the scripts for one of focal catchments of Goodwater project – the Zaņa catchment of Venta RBD is illustrated in Chapter 2.

1. Library of scenario scripts

1.1. General principles

Python scripts were prepared for the automatic setup preparation for the Modeling system with adjusting its settings for the modeling application of pollution abatement measures or introduction of new pollution sources. These scripts are described in the further sections.

Measure scripts are realized by the class *Measures* defined in *measures.py*. All measures could be subdivided in two parts:

1. **HRU level measures** are measures that are applicable to each of the HRUs separately and modify HRU parameters only.
2. **Catchment level measures** require parameter change on a catchment level.

The measures are realized by a unified one-line calls in specially designed main script file *main_measures.py*. There are examples of measure calls provided in this file. It is possible to call measures sequentially, although it is not possible to combine some measures with others – in this case the last called measure will be applied. To apply the measures users should modify *main_measures.py* file according to their needs and run it. It is recommended to change *settings.defaultCaseSubDir* variable to provide a name for a setup with measures applied to avoid overwriting of existing runs.

Some of the measures described further are split into combination of different measures, and several additional measures are introduced to increase a flexibility of work with the modelling system.

The library of scripts is provided in folder *lib/py*.

1.2. Application of HRU level measures

All HRU level measures are implemented by function *implementMeasure* of class *Measures* defined in *measures.py*. The signature of the call with common parameters for all HRU level measures is

```
implementMeasure(  
    measuredefinition,  
    watershednamelist=[],  
    application_fraction=1.0,  
    fractiontype=measureFunctions.HruFractionType.first,  
    landuse_list=[],  
    soillist=[],  
    **x  
)
```

1. *measuredefinition* is an object of the class *HRULevelMeasure*, that defines the measure. Implemented measures are listed in *measures.py* file.



7



2. *watershednamelist* is a list of watersheds to which measure are applied. Watershed names are strings in the form 'Basin/Name'. In case of empty list measure are applied to all watersheds.
3. *application_fraction* is a number between 0.0 and 1.0 that determines to which fraction of the total area of eligible HRUs selected measure will be applied.
4. *fractiontype* is a variable of type *measureFunctions.HruFractionType* that determines method by which HRUs are selected according to given measure application fraction. Valid methods are:
 - a. *measureFunctions.HruFractionType.first* – the measure will be applied to the list of eligible HRUs in the sequential order until desired *application_fraction* of total area of eligible HRUs is reached. One of HRUs can be splitted to ensure exact *application_fraction*. Therefore number of HRUs will be either the same or one more than original.
 - b. *measureFunctions.HruFractionType.shuffle* - similar to *measureFunctions.HruFractionType.first*, but order of eligible HRUs are randomly shuffled.
 - c. *measureFunctions.HruFractionType.split* – each of the eligible HRUs will be splitted into two with areas according to *application_fraction*. In this case multiple new HRUs will be added to the setup.
5. *landuse_list* is a list of landuses to which measure will be applied, in the case of empty list measure will be applied to all eligible land uses.
6. *soillist* is a list of soils to which measure will be applied, in the case of empty list measure will be applied to all soils.

Measures can be called sequentially. In this case subsequent measures will be applied to already modified setup.



1.3. Point sources

Introduction of new point sources or updating values of the data of the existing point sources in the database is a catchment level measure. It allows for adding, deleting or modification of point sources. It can combine initially supplied point source data with additional point source data supplied by user and save resulting data in the database and in the data files.

Call

```
scenarios.implement_additional_point_sources(
    monthly_catchment_table = ("point_sources", "ps_month_catchment"),
    yearly_catchment_table = ("point_sources", "ps_year_catchment"),
    yearly_data_table = ("point_sources", "ps_year_data"),
    ADPS_MONTHLY_FILE = settings.commonDataDir +
    "PointSources/monthly_measures.txt",
    COMB_MONTHLY_FILE = settings.commonDataDir +
    "PointSources/comb_monthly_measures.txt",
    ADPS_YEARLY_FILE = settings.commonDataDir +
    "PointSources/Small_PointSources_measures.txt",
)
```

Input parameters

Table 1: Input variables.

Name	Default value
monthly_catchment_table	("point_sources", "ps_month_catchment")
yearly_catchment_table	("point_sources", "ps_year_catchment"),
yearly_data_table	("point_sources", "ps_year_data"),
ADPS_MONTHLY_FILE	settings.commonDataDir + "PointSources/monthly_measures.txt"
COMB_MONTHLY_FILE	settings.commonDataDir + "PointSources/comb_monthly_measures.txt"
ADPS_YEARLY_FILE	settings.commonDataDir + "PointSources/Small_PointSources_measures.txt"

- *monthly_catchment_table* is a new name of the table in the database. It replaces parameter *settings.PS_catchment_table*.
- *yearly_catchment_table* is a new name of the table in the database. It replaces parameter *settings.PS_YEARLY_CATCHMENT_TABLE*.
- *yearly_data_table* is a new name of the table in the database. It replaces parameter *settings.PS_YEARLY*.
- *ADPS_MONTHLY_FILE* is a path to additional station and data file of the point sources.
- *COMB_MONTHLY_FILE* is a path to combined data file (initial data plus additional data) of the point sources. It replaces parameter *settings.PS_MONTHLY_FILE*.
- *ADPS_YEARLY_FILE* is a path to additional station and data file of the point sources.

ADPS_MONTHLY_FILE points to the additional monthly station and data files. The format of this file is similar to the original monthly point source files except the station file (*.sta) contains additional column *command*. Values of data in this column can be one of the following *add*, *overwrite*, *remove*, *none*. The procedure goes through additional station file and searches

matching station in the original station file row (station) at a distance not more than 10 m. In case there are several such stations, names will be compared and the matching station will be selected (if any).

The commands in the additional point source file are:

- *add* - additional station data will be added to the combined data file, station ID will be automatically changed.
- *overwrite* - the additional station data will replace initial matching station data. If there are no corresponding station found then the *add* command will be executed.
- *remove* – the matching station data will be removed from the combined data file.
- *none* - additional station will be ignored.

All other initial station data will be copied to the resulting combined file.

ADPS_YEARLY_FILE points to additional yearly station and data files. Data file (*.txt) has the same format as in the table of the original small point source database *settings.PS_YEARLY*. The format of the station file (*.sta) format is the same format as described above.



1.4. Land use change

Conversion of one type land uses with given ratio setting to the other types in all model territory or only specific regions is implemented via two measures:

1. Landuse conversion by landuse groups allows to change part of any landuse group (Agricultural, Pasture, Forest, Urban, Wetland, Water, Barren) to any other group.
2. Landuse conversion by landuse code allows change of landuse of HRUs substituting one landuse code by another.

1.4.1. Land use conversion by land use groups

This HRU level measure allows conversion of one landuse group to another in the whole model territory or in the selected model watersheds, the application fraction could also be provided, as well as an optional list of applicable land uses or soils to which the measure is applied.

Call

```
scenarios.implementMeasure(  
  measureFunctions.measure_landuse_conversion_group,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  landuse_list=[],  
  soillist=[],  
  fractiontype=measureFunctions.HruFractionType.first,  
  conversion=measureFunctions.LUConversionGroups(  
    settings.groupCodeForest, settings.groupCodeAgriculture  
  )  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- *conversion* is the variable of type *measureFunctions.LUConversionGroups*. The first parameter of it is an source landuse group, the second is a destination landuse group.
- Application fraction is provided in the general parameter *application_fraction*.



1.4.2. Landuse conversion by landuse code

This HRU level measure allows conversion of one landuse to another on the whole country territory or in the selected model watersheds. The application fraction may also be provided, as well as an optional list of applicable soils to which the measure is applied.

Call

```
scenarios.implementMeasure(  
  measures.measure_landuse_conversion,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  fractiontype = measureFunctions.HruFractionType.first,  
  soillist=[],  
  conversion = measureFunctions.LUConversion('AGRC', 'BARL')  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- *conversion* is the variable of type *measureFunctions.LUConversion*. The first parameter of it is an source landuse code, the second is a destination landuse code.
- Application fraction is provided in the general parameter *application_fraction*.



1.5. Catch crops

Introduction of vegetative cover in autumn and winter through catch crops is realized via three measures, namely: (1) “Catch crops”, (2) “Planting of winter crops” and (3) “Plant cover in winter”.

The measures require modification of the management practice (*plant passports*) for involved plants. The tables of management practices for measures are located on the PostgreSQL database of the project under the scheme *measures*. In case the measures that involve modification of plant passports are called sequentially, the plant passport for a subsequent measure for a particular HRU will override any previously set plant passport for that particular HRU.

1.5.1. Catch crops

This measure corresponds to agricultural optimization measure “*Catch crops*”. The “catch crop” measure may be used for the winter crops summarized in the Table 2. Crop with ID CANP (Annual grasses) is used as a catch crop. Catch crop is planted one day after harvest of the main crop and is removed (using SWAT+ operation ‘kill’) one day before the main crop is planted.

The modification of the management practice for involved plants include:

- All tillage operations that would happen while the catch crop is planted are omitted from the management practice.
- Mineral N fertilization amount is reduced by 20% (reduction applied to all applications proportionally).
- Stubble breaking operation is removed.
- Deep ploughing (Tillage ID *deeplow*) is performed before planting instead of Soil loosening (Tillage ID *rowcult*).

This HRU level measure can be implemented on the whole country territory or in selected model watersheds, the application fraction could also be provided, as well as an optional list of applicable landuse codes or soils to which the measure is applied. Filtering by landuse codes are allowed only for the eligible crops (see Table 2).

Call

```
scenarios.implementMeasure(  
  measures.measure_catchcrops,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  fractiontype=measureFunctions.HruFractionType.first,  
  landuse_list=[],  
  soillist=[],  
)
```



Table 2: Crops eligible for combining with catch crops.

RYE	Winter rye
WBAR	Winter barley
WTRC	Winter triticale
WWHT	Winter Wheat

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

The plant passports table for the measure is located in the table *plantmng_catchcrops* of scheme *measures* in the PostgreSQL database of the project.

1.5.2. Planting of winter crops

Agricultural optimization measure “*Planting of winter crops*” in (PAIC, 2015) means that some type of catch-crop (winter rye, winter rape, winter vetch) is left in the field over the winter after harvest of the eligible summer crops (see Table 3). The crop used as a catch crop is Winter rye (SWAT+ ID: RYE). To avoid the dormancy issue, a crop RYE1 was introduced. This crop is identical in all parameters with original crop RYE except for change of the plant type from *cold_annual* to *warm_annual*. The changes in management practice are as follows (generally):

- No tillage operations in autumn.
- All manure is applied in spring (instead of part in spring, part in autumn).
- Spring tillage practice:
 - Planting delayed by ~2 weeks days;
 - Deep ploughing done ~2 weeks before planting;
 - Stubble breaking is done 2-3 weeks before deep ploughing.
- Spring fertilization:
 - Manure – between stubble breaking and deep ploughing;
 - Mineral fertilisation (both P and N) – together with deep ploughing.
- Mineral N amount reduced by 10% (proportionally in all applications).
- RYE Kill operation is performed on 30th of March for all crops.

Call

```
scenarios.implementMeasure(
  measures.measure_planting_of_winter_crops,
  watershednamelist=watershednamelist,
  application_fraction=1.0,
  fractiontype=measureFunctions.HruFractionType.first,
  landuse_list=[],
  soillist=[],
)
```

Table 3: Crops eligible for combining with winter catch crop (rye).



AGRC	Agricultural land
BARL	Summer barley
BWHT	Buckwheat
CANP	Annual grasses
CELR	Caraway
CORN	Corn
CRRT	Vegetables
CSIL	Corn for silage
GRBN	Beans/ annual legumes
LMIX	Mix of legumes with other crops
LUPN	Lupine
OATS	Oats
PEAS	Peas
POTA	Potatoes
SCAN	Summer rape
SGBT	Sugar beet
SOYB	Soy
STBR	Summer barley (straw fields)
STCN	Summer rape (straw fields)
STRC	Summer triticale+summer rye
STWH	Summer wheat (straw fields)
SWHT	Summer wheat
VTCH	Vetch

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- The plant passports table for the measure is located in the table *plantmng_plantingofwintercrops* of scheme *measures* in the PostgreSQL database of the project.
- Definition of additional plant (RYE1) is located in table *additional_plants* of scheme *measures*. Filtering by landuse codes is allowed only for the eligible crops (see Table 3).



1.5.3. Plant cover in winter

This agricultural optimisation measure “*Plant cover in winter*” generally can be implemented for summer crops, see Table 4.

Table 4: Crops eligible for leaving plant cover throughout the winter. List of crops and the names of alternative crops created for this measure.

SWAT ID	Name	Alt_name
AGRC	Agricultural land	AGR1
BARL	Summer barley	BAR1
BWHT	Buckwheat	BWH1
CANP	Annual grasses	CAN1
CELR	Caraway	CEL1
CORN	Corn	COR1
CSIL	Corn for silage	CS11
LUPN	Lupine	LUP1
OATS	Oats	OAT1
SCAN	Summer rape	SCA1
SOYB	Soy	SOY1
STBR	Summer barley (straw fields)	STB1
STCN	Summer rape (straw fields)	STC1
STRC	Summer triticale+summer rye	STR1
STWH	Summer wheat (straw fields)	STW1
SWHT	Summer wheat	SWH1
VTCH	Vetch	VTC1

The “plant cover” measure was implemented as the following (generalized) scheme for changing the management practice:

- Operation Harvest/Kill (SWAT+ ID 5) in autumn was replaced with operation Harvest only (SWAT+ ID 7).
- Any tillage after harvest and in autumn was removed.
- Kill operation (SWAT+ ID 8) was performed on 30th of March, i.e. shortly before planting of the crop.
- If only the Soil loosening (Tillage SWAT+ ID *rowcult*) was done for original management practice in Spring, then it was replaced with the Autumn tillage type: either Deep ploughing (Tillage SWAT+ ID *deepplow*) or Ploughing (Tillage SWAT+ ID *fallplow*) for the corresponding plant.
- Manure application was done in spring along with Tillage instead of splitting it up between autumn and spring.
- 2 week delay in planting of the crops BARL, OATS, SWHT, STBR was applied.



For this measure it was necessary to create an alternative crop for each of the eligible crops in the Table 4.

Call

```
scenarios.implementMeasure(  
  measures.measure_plant_cover_in_winter,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  fractiontype = measureFunctions.HruFractionType.first,  
  landuse_list=[],  
  soillist=[]  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- The plant passports table for the measure is located in the table *plantmng_platcoverinwinter* of scheme *measures* in the PostgreSQL database of the project.
- Definitions of the additional plants are located in the table *additional_plants* of scheme *measures*. Filtering by landuse codes is allowed only for the eligible crops (see Table 4).



1.6. Tillage

Application of reduced or no tillage technology, alteration of tillage timing is realized via three agricultural optimisation measures: (1) “No-plough technology”, (2) “Substituting autumn ploughing with spring ploughing” and (3) “Postponing a sod ploughing to late autumn”.

The measures require modification of the management practice (*plant passports*) for the involved plants. The tables of management practices for measures are located on the PostgreSQL database of project under the scheme *measures*. In case the measures that involve modification of plant passports are called sequentially, the plant passport for a subsequent measure for a particular HRU will override any previously set plant passport for that particular HRU.

1.6.1. No-plough technology

The agricultural optimisation measure “*No-plough technology*” includes the following changes in management practice:

Table 5: List of crops eligible for no-plough technology.

BARL	Summer barley
BWHT	Buckwheat
CANP	Annual grasses
CELR	Caraway
CORN	Corn
CSIL	Corn for silage
LUPN	Lupine
OATS	Oats
SCAN	Summer rape
SOYB	Soy
SWHT	Summer wheat
VTCH	Vetch
CANA	Winter rape
RYE	Rye
WBAR	Winter barley
WTRC	Winter triticale
WWHT	Winter wheat

- Stubble breaking after harvest is performed as previously, deep ploughing after it is removed.



- Manure is applied with stubble breaking.

This measure is applicable for the crops summarized in Table 5.

Call

```
scenarios.implementMeasure(
  measures.measure_no_plough_technology,
  watershednamelist=watershednamelist,
  application_fraction=1.0,
  fractiontype=measureFunctions.HruFractionType.first,
  landuse_list=[],
  soillist=[],
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- The plant passports table for the measure is located in the table *plantmng_noploughtechnology* of scheme *measures* in the PostgreSQL database of the project.
- Filtering by landuse codes is allowed only for the eligible crops, see Table 5.

1.6.2. Substituting autumn ploughing with spring ploughing

This agricultural optimisation measure “*Substituting autumn ploughing with spring ploughing*” assumes that land cultivation in autumn is replaced by the land cultivation in spring – before the crop planting. This measure is carried out for sandy soils only.

Call

```
scenarios.implementMeasure(
  measures.measure_substituting_autumn_ploughing_with_spring_ploughing
,
  watershednamelist=watershednamelist,
  application_fraction=1.0,
  fractiontype=measureFunctions.HruFractionType.first,
  landuse_list=[],
  soillist=[],
)
```

This measure can be implemented for all summer crops except for those that originally had no autumn ploughing as there was nothing to change. See the list of eligible crops in Table 6. The changes in management practice involve:

- Any ploughing (ID 109 or ID 1) is done in spring 2 weeks before planting.



- All manure is applied with ploughing in the spring.

Table 6: Crop list eligible for substitution of autumn ploughing with spring ploughing.

AGRC	Agricultural land	LMIX	Mix of legumes with other crops
ALFA	Perennial legumes	LUPN	Lupine
BARL	Summer barley	OATS	Oats
BWHT	Buckwheat	PEAS	Peas
CANP	Annual grasses	POTA	Potatoes
CELR	Caraway	SCAN	Summer rape
CORN	Corn	SGBT	Sugar beet
CRRT	Vegetables	SOYB	Soy
CSIL	Corn for silage	SWHT	Summer wheat
GRBN	Beans/ annual legumes	VTCH	Vetch

Call

```
scenarios.implementMeasure(
  measures.measure_substituting_autumn_ploughing_with_spring_ploughing
,
  watershednamelist=watershednamelist,
  application_fraction=1.0,
  fractiontype=measureFunctions.HruFractionType.first,
  landuse_list=[],
  soillist=[],
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- The plant passports' table for the measure is located in the table *plantmng_substitutingautumnploughingwithspringploughing* of scheme *measures* in the PostgreSQL database of the project.
- The table of the soil classification is provided in table *lightheavysoils* of scheme *measures*. Filtering by landuse codes is allowed only for the eligible crops, see Table 6.
- Filtering by soil code is allowed only for sandy soils.



1.6.3. Postponing a sod ploughing to late autumn

The agricultural optimisation measure “*Postponing a sod ploughing to late autumn*” assumes that Deep Ploughing (Tillage SWAT+ ID: *deeplow*) and the associated manure application is postponed to the late autumn – end of October or beginning of November. This measure can be carried out for sandy soils only. It is applicable to the crops summarized in Table 7.

Table 7: List of crops eligible for sod-ploughing in late autumn.

BARL	Summer barley	LUPN	Lupine
BWHT	Buckwheat	OATS	Oats
CANP	Annual grasses	SCAN	Summer rape
CELR	Caraway	SOYB	Soy
CORN	Corn	SWHT	Summer wheat
CSIL	Corn for silage	VTCH	Vetch

Call

```
scenarios.implementMeasure(
  measures.measure_postponing_a_sod_ploughing_to_late_autumn,
  watershednamelist=watershednamelist,
  application_fraction=1.0,
  fractiontype = measureFunctions.HruFractionType.first,
  landuse_list=[],
  soillist=[]
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- The plant passports table for the measure is located in table *plantmng_plantmng_postponingasodploughingtolateautumn* of scheme *measures* in the PostgreSQL database of the project.
- The table of soil classification is provided in table *lightheavysoils* of scheme *measures*.
- Filtering by landuse codes are allowed only for the eligible crops, see Table 7.
- Filtering by soil code is allowed only for sandy soils.



1.7. Fertilization amount

Adapting amounts of chemical or/and organic fertilizers on a regional or crop basis by setting percentage of fertilization alteration or maximum amounts allowed for fertilization is implemented by introducing coefficients for relative change of mineral nitrogen, phosphorus and manure. The parameters regulating maximum amount of application of mineral nitrogen, phosphorus and manure (in kg/ha/year) are introduced. The measure is allowed on agricultural land uses only.

Call

```
scenarios.implementMeasure(  
  measures.measure_change_fertilization,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  fractiontype=measureFunctions.HruFractionType.first,  
  landuse_list=[],  
  soillist=[],  
  fertilizersettings=measureFunctions.FertilizerModificationSettings(  
    fertCoef=None,  
    fertCoefN=1.0,  
    fertCoefP=1.0,  
    manureCoef=1.0,  
    fertMaxN=None,  
    fertMaxP=None,  
    fertMaxManure=None,  
  ),  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures. Filtering by land use codes is allowed only for the agricultural land uses.

fertilizersettings is a parameter of type *measureFunctions.FertilizerModificationSettings*. It allows for changes in fertilizer amounts and thresholding of maximum amounts of fertilizers. The fields are:

- *fertCoefN* - coefficient of multiplication for amount of the mineral nitrogen fertilizer;
- *fertCoefP* - coefficient of multiplication for amount of the mineral phosphorus fertilizer;
- *fertCoefManure* - coefficient of multiplication for amount of the manure;
- *fertCoef* – if given (not equal to *None*) then set all multiplication coefficients at once: *fertCoefN=fertCoefP=fertCoefManure=fertCoef*;
- *fertMaxN* – maximal amount in kg/ha/year of the mineral nitrogen fertilizer that could be applied;
- *fertMaxP* – maximal amount in kg/ha/year of the mineral phosphorus fertilizer that could be applied;
- *fertMaxManure* – maximal amount in kg/ha/year of manure that could be applied;

If the script is applied sequentially then only the last set of coefficients applied for a particular HRU will be considered for that particular HRU.



1.8. Fertilization timing

Adapting available timing for the applications of chemical or/and organic fertilizers requires modification of the management practice (*plant passports*) for involved plants. The table of management practice for the measure is located in the PostgreSQL database of project under the scheme *measures*. If the measures that involve modification of the plant passports are called sequentially, then the plant passport for a subsequent measure for a particular HRU will override any previously set plant passport for that HRU.

Call

```
scenarios.implementMeasure(  
  measures.measure_fertilizertiming,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  fractiontype=measureFunctions.HruFractionType.first,  
  landuse_list=[],  
  soillist=[],  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures. The plant passport table for the measure is in the table *plantmng_fertilizertiming* of scheme *measures* in the PostgreSQL database of the project.



1.9. Fertilization technique

Application of advanced fertilization techniques (like combi-drilling, incorporation of fertilizers) to decrease nutrient leakage and to raise nutrient absorption efficiency uses the new features of SWAT+.

This measure allows selecting of fertilizer application type from the list of application operations from the chemical application operations database provided in SWAT+. It could be combined in one call with the changing of amounts of fertilizers and can be applied only to agricultural HRUs.

Call

```
scenarios.implementMeasure(  
  measures.measure_change_fertilization,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  landuse_list=[],  
  soillist=[],  
  fertilizersettings=measureFunctions.FertilizerModificationSettings(  
    fertCoef=1.0,  
    fertCoefN=None, fertCoefP=None, manureCoef=None,  
    fertMaxN=None, fertMaxP=None, fertMaxManure=None,  
    applicationType=measureFunctions.FertilizerApplicationTypes.drill,  
  ),  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

- Filtering by land use codes is allowed only for agricultural land uses.
- *fertilizersettings* is a parameter of the type *measureFunctions.FertilizerModificationSettings*. It allows for changes in fertilizer amounts and thresholding of maximum amounts of fertilizers and selecting type of fertilizer application.
- Fields for changing amounts of fertilizer are described in the previous sections.
- *applicationType* is parameter of the type *measureFunctions.FertilizerApplicationTypes* that represents type of chemical application according to SWAT+ chemical application operations database. The possible values are *drill*, *broadcast*, *band*, *foliar*, *inject*, *aerial_liquid*, *aerial_solid*, *side_dress*, *fertigate*, *basal*, *rope_wick*, *tree_inject*, *default*. Default value for the application type is *inject*.



1.10. Enlarged grassed waterways and buffer zones

Introduction or enlarging grassed waterways and buffer zones along water and erosion sensitive areas is realized via two separate measures.

1.10.1. Grassed waterways

Measure allows introducing of grassed waterways by adding parameters necessary as input for SWAT+. Filtering by land use code and soil type is allowed. Application fraction is controlled by parameter *application_fraction*.

Call

```
scenarios.implementMeasure(  
  measures.measure_grassedwaterways,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  landuse_list=[],  
  soillist=[],  
  fractiontype=measureFunctions.HruFractionType.first,  
  grassedwaterway=measureFunctions.GrassedWaterWayForMeasures,  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

grassedwaterway is the parameter of type *measureFunctions.GrassedWaterWayForMeasures*. It contains parameters necessary for the definition of SWAT+ grassed waterway for HRU:

- *mann* - Mannings's n for grassed waterway (default 0.05).
- *sed_co* - sediment transport coefficient defined by user (default 0.02).
- *dp* - depth of grassed waterway (m) (default 0.75).
- *wd* - width of grass waterway (m) (default 3).
- *len* - length of Grass Waterway (km) (default 0.75).
- *slp* - slope of grass waterway (m/m) (default 0.035).



1.10.2. Filter strips

Measure allows introducing of filter strips by adding the input parameters necessary for SWAT+. Filtering by landuse code and soil type is allowed. Application fraction is controlled by the parameter *application_fraction*.

Call

```
scenarios.implementMeasure(  
    measures.measure_filterstrips,  
    watershednamelist=watershednamelist,  
    application_fraction=1.0,  
    landuse_list=[],  
    soillist=[],  
    fractiontype=measureFunctions.HruFractionType.first,  
    filterstrip=measureFunctions.FilterStripForMeasures,  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

filterstrip is a parameter of the type *measureFunctions.FilterStripForMeasures*. It contains parameters necessary for definition of SWAT+ filter strip for HRU:

- *fld_vfs* - Ratio of field area to filter strip area.
- *con_vfs* - Fraction of flow entering the most concentrated 10% of the filter strip.
- *cha_q* - Fraction of fully channelized flow.

If applied, this measure will override default buffer strips that are automatically generated for the setup.



1.11.Tile drainage parameters

Tile drainage measure allows introducing of tile drains by adding parameters necessary for SWAT+ input. Filtering by land use code and soil type are allowed. Application fraction is controlled by the parameter *application_fraction*.

Call

```
scenarios.implementMeasure(  
  measures.measure_tiledrain,  
  watershednamelist=watershednamelist,  
  application_fraction=1.0,  
  landuse_list=[],  
  soillist=[],  
  fractiontype=measureFunctions.HruFractionType.first,  
  tiledrain=measureFunctions.TileDrainForMeasures,  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

tiledrain is a parameter of type *measureFunctions.TileDrainForMeasures*. It contains parameters necessary for definition of SWAT+ tile drain for HRU:

- *dp* - Depth of drain tube from the soil surface (mm);
- *t_fc* - Time to drain soil to field capacity (hr);
- *lag* - Drain tile lag time (hr);
- *rad* - Effective radius of drains (mm);
- *dist* - Distance between two drain tubes or tiles (mm);
- *drain* - Drainage coefficient (mm/day);
- *pump* - Pump capacity (mm/hour);
- *lat_ksat* - Multiplication factor to determine lateral ksat from SWAT ksat input value.

If applied, this measure will override default tile drains that are automatically generated for the setup.



1.12.Sedimentation ponds and wetlands

Introduction of sedimentation ponds and wetlands within subbasin areas (as opposite of being on the modeled river network) is developed as the catchment level measure. It introduces SWAT+ wetland object to the HRUs of the specified landuse group (Agritultural, Pasture, Forest, Urban). SWAT+ wetland parameters can be passed to the procedure by a dictionary.

Call

```
scenarios.introduction_of_sedimentation_ponds_and_wetlands(  
    watershednamelist=watershednamelist,  
    landuse_to_wetland={  
        settings.groupCodeAgriculture: dict(hru_ps=0.0001, hru_es=0.0001),  
        settings.groupCodePasture: dict(hru_ps=0.0001, hru_es=0.0001),  
    },  
)
```

Input parameters

- *watershednamelist* is a list of watersheds to which the measure is applied.
- *landuse_to_wetland* is a dictionary containing SWAT+ wetland object parameters according to the selected land use group.



1.13.Pesticides

Application of pesticides on selected crops on all country territory or only specific regions and for specific time periods requires modification of the management practice (*plant passports*) for involved plants.

The table of management practices for the measure are in the PostgreSQL database of project under the scheme *measures*. In case if the measures that involve modification of plant passports are called sequentially, the plant passport for a subsequent measure for a particular HRU will override any previously set plant passport for that HRU.

The operation type *Pest* is introduced into plant passports, that allows the application of any pesticide that is present in the SWAT+ pesticides database.

Call

```
scenarios.implementMeasure(  
    measures.measure_pesticides,  
    watershednamelist=watershednamelist,  
    application_fraction=1.0,  
    landuse_list=[],  
    soillist=[],  
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures. The plant passport table for the measure is in the table *plantmng_pesticides* of scheme *measures* in the PostgreSQL database of the project.



1.14.Land use update

The update of the land use during the modelling period is different in SWAT2012 and SWAT+. The former used *.lup files which are no longer supported in SWAT+. Therefore, the land use change should be implemented via decision tables. We provide the description of the script which allows for automated handling of these decision tables.

The implementation uses the functionality of SWAT+ decision tables that allows updating the area fractions of HRUs. Changed fractions are calculated by the script from two CORINE datasets. The script automatically determines the updated HRU fractions according to the following method:

1. The land use classes are aggregated in 7 groups as “Agricultural”, “Pastures”, “Forest”, “Urban”, “Wetlands”, “Water” and “Barren”.
2. For each of the catchments the relative fractions of each of the group of the land use classes are calculated for CORINE 2000 and CORINE 2006.
3. The table containing catchment ID, and 7 numbers – the changes (in %) of the relative fraction of each of the group of land use classes is prepared.

The prepared data is further used by the automated system of the generation of the water quality modeling system for creation of decision table that allows the update. Landuse update (LUP) related variables in *settings.py* are under the class *LanduseUpdate*:

1. Variable *update_fractions* contains a scheme and table name of the Postgre table where land use update proportional fractions per catchment are prepared and stored. Default value of this variable is ("*lup*", "*lupupdate*"). This table contains the following fields:
 - a. *catchmentID* (integer) – catchment ID of subbasin as defined in *CatchmentFile*.
 - b. *Agricultural, Barren, Forest, Pasture, Urban, Water, Wetland* (double) – these fields provide values of update fractions per each land use group.
2. Variable *lup_date* is a start date of landscape update (*yyyy,mm,dd*), if it is empty (*()*) then the landscape update is not applied. Default value of this variable is (*()*).
3. Variables for preparation of the landuse update fractions:
 - a. Variables *classes_1* and *classes_2* define the schema and table names of the two polygon feature classes in Postgre database which define two sequential land uses. Default values of these variables are ("*lup*", "*clc00_lt*") and ("*lup*", "*clc06_lt*").
 - b. Variable *lookup_table* contains the schema and name of Postgre table that defines the relations between the landscape update polygons and the general land use groups. Default value of this variable is ("*lup*", "*luplookup*").
 - c. Variables *lu_1_field*, *lu_2_field*, *lookupfield* define the field name of land use



code in, respectively, first land use table, second land use table and lookup table. Variable *groupfield* define the field name of land use group code in the lookup table.

- d. Variables *catchments_1* and *catchments_2* contain Postgre schemas and table names of the polygon classes which will contain the intersection of the *classes_1* and *classes_2* with the subbasin polygons. Default values of these variables are ("*lup*", "*corine2000*") and ("*lup*", "*corine2006*").
- e. Variables *table_1* and *table_2* are Postgre schema and table names of tables where amounts of land use groups per subbasin will be stored. Default values are ("*lup*", "*lup_table1*") and ("*lup*", "*lup_table2*").

1.15. Managed drainage

Introduction of managed drainage into fields was realized within the new current SWAT+ version capabilities – the decision tables.

This HRU level measure allows changing of tile drainage depth by means of user supplied decision table. Filtering by landuse code and soil type are allowed. Application fraction is controlled by parameter *application_fraction*.

Call

```
scenarios.implementMeasure(
  measures.measure_managed_drainage,
  watershednamelist=watershednamelist,
  application_fraction=1.0,
  landuse_list=[],
  soillist=[],
  fractiontype=measureFunctions.HruFractionType.first,
  tiledraindecisiontable=drain_control_dtl,
)
```

Input parameters

General parameters are described in the signature of the call with common parameters for all HRU level measures.

Tiledraindecisiontable is a string containing decision table in SWAT+ format. For a description of format of decision table see SWAT+ input output manual¹.

The sample decision table that allows introducing of managed drainage is given in the variable *measureFunctions.ManagedDrainageDecisionTable.drain_control_dtl*, see also Table 8. It allows for opening of tile drains by setting the depth of drains to 1100 mm in spring (day of year 65) and closing them in autumn (day of year 295) by settings depth of drains to 0 mm.

- The first line of decision table is a comment line.

¹ <https://swatplus.gitbook.io/io-docs/>



- The second line contains, sequentially, the name of decision table, the number of conditions, the number of alternatives and the number of actions.
- Next, the lines describing the conditions follow (comment line and 2 condition lines in the example case). Conditions in the example file are for the day of the year (variable name *jday*). For the list of possible conditions consult the SWAT+ input output manual.
- Further, the lines describing the actions follow (comment line and 2 action lines in the example case). Actions in the example file are of type *drain_control*. This action type allows setting of depth of drain tiles that need to be provided in column named *const*.

Table 8: Sample decision table.

name	conds	alts	acts	!open drains in spring and close in autumn						
control_drainage	2		2	2						
var	obj	obj_n	lim_var	lim_op	lim_cnst	alt1	alt2			
jday	hru	0	null	-	65.00	=	-	!open drains on day 65		
jday	hru	0	null	-	295.00	-	=	!close drains on day 295		
act_typ		obj	obj_n	name	option	const	const2	fp	outcome	
drain_control		hru	0	open_spring	null	1100.0	1.000	null	y	n
drain_control		hru	0	close_fall	null	0.0000	1.000	null	n	y



2. Employment of scripts

2.1. Zaņa catchment

The employment of the scripts which simulate the measures of the water quality improvement is illustrated on the example of Zaņa catchment in Venta River basin district, see Figure 1.

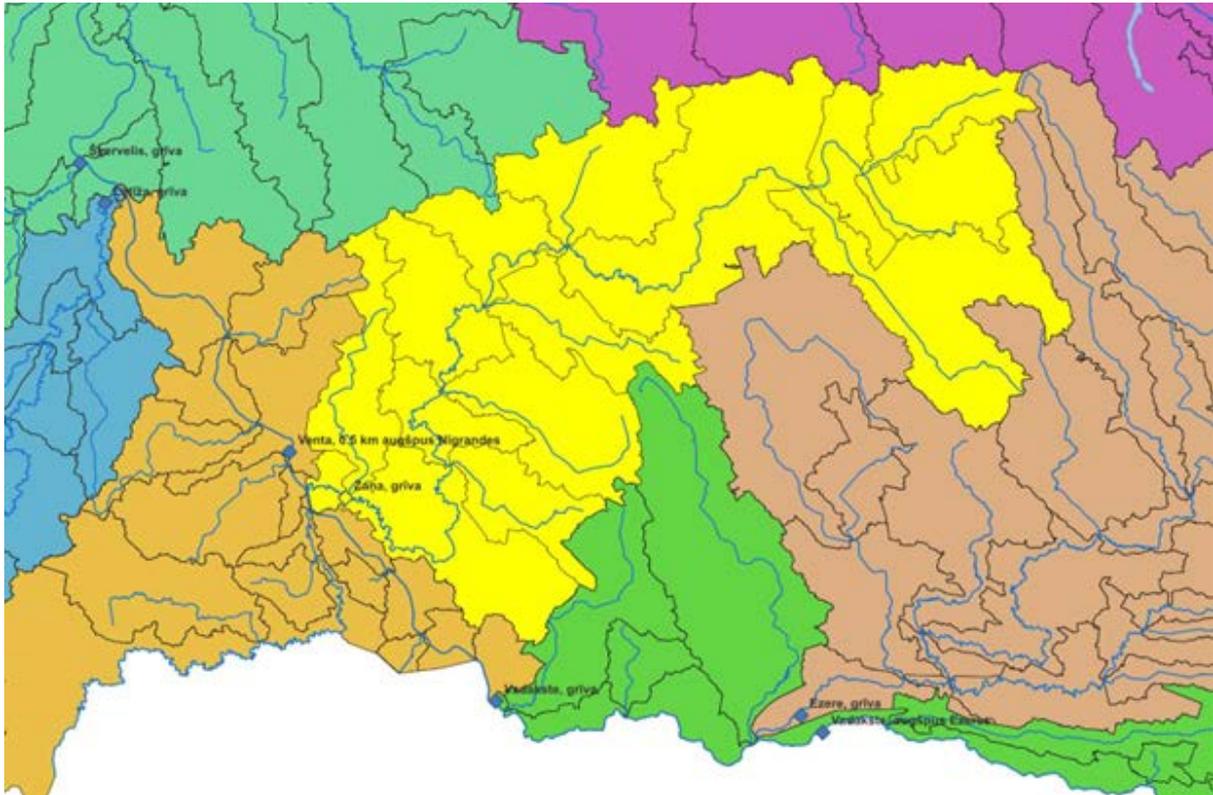


Figure 1: Zaņa catchment and Zaņa mouth water quality observation station.

The modelling and observation results for Years 2006-2019 are provided in Figures 2-5 as time graphs (Figures 2,4) and seasonal cycles (Figures 3,5) of concentrations of the nitrate nitrogen (Figures 2,3) and the phosphate phosphorus (Figures 4,5).

Model provides satisfactory levels of nutrients' concentrations in the circumstances of the absence of water quantity measurements.

The study was performed to assess the reliability of the modelling system for the modelling of water quality improvement measures. The use of the scripts implementing various measures is tested in the further sections of this Chapter. If not stated otherwise the study is concentrated on a sample catchment Venta/Zaņa catchment, years 2008-2019.

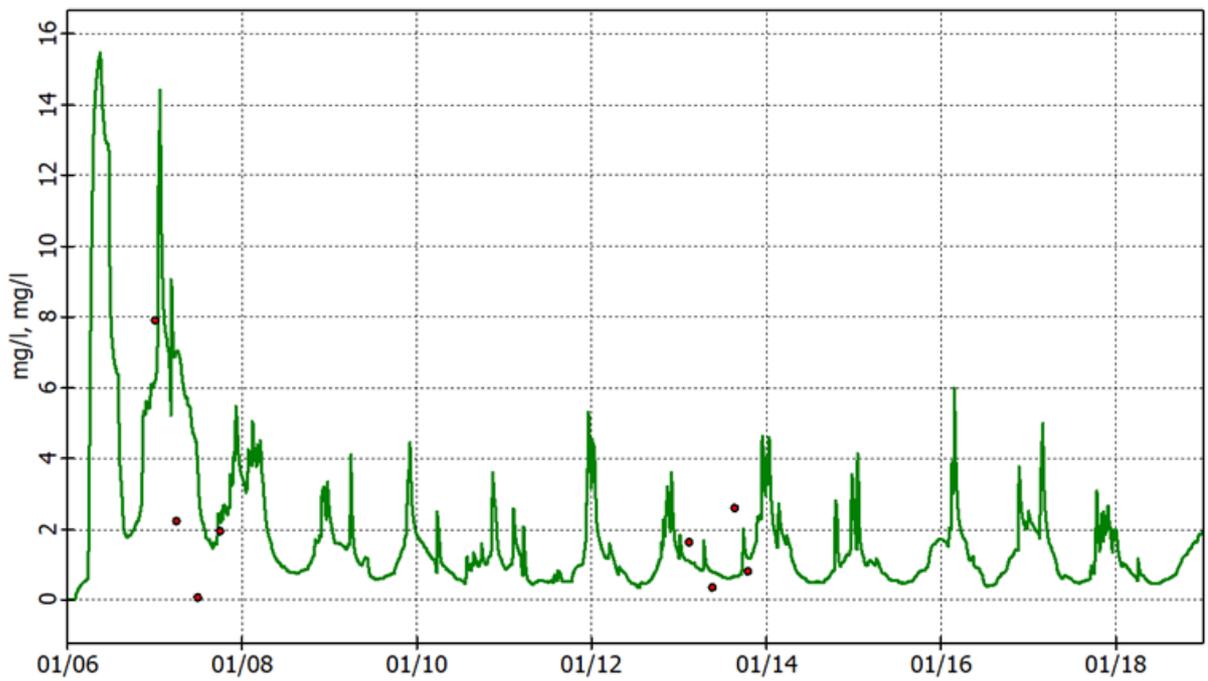


Figure 2: Time graph of observed and measured N-NO3 concentration at Zaņa mouth,

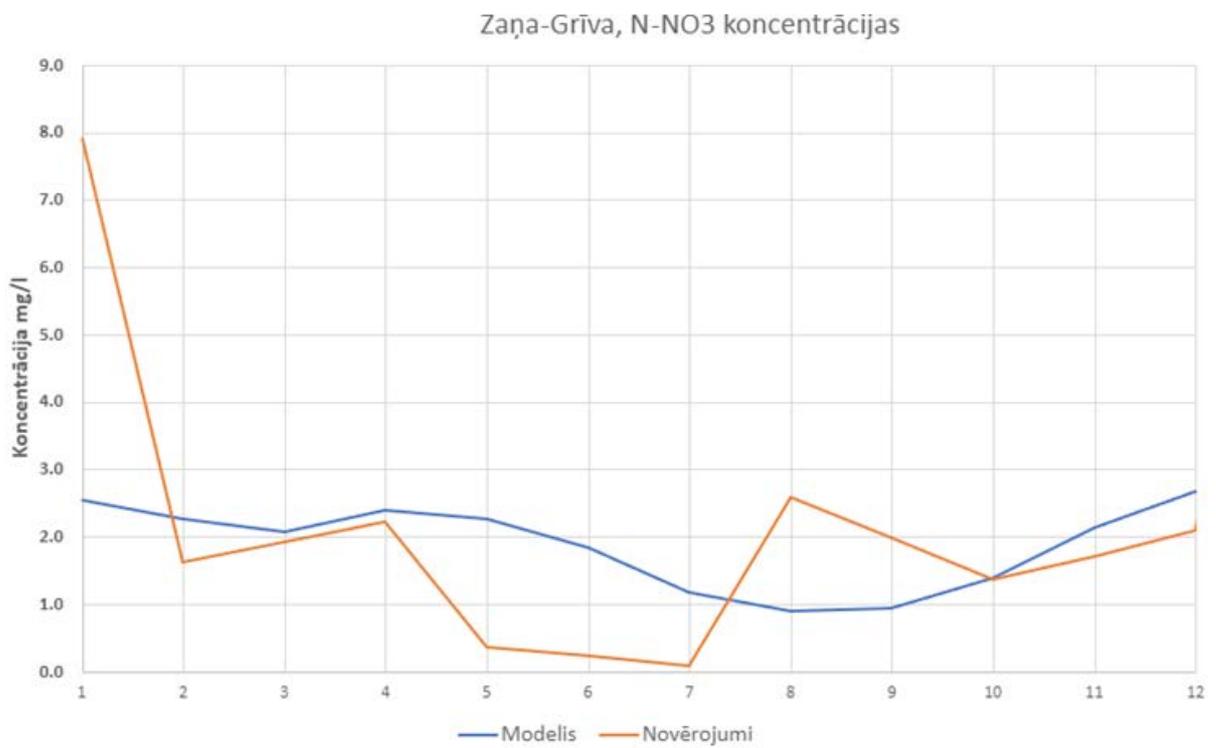


Figure 3: Observed and measured seasonal cycle of N-NO3 concentration at Zaņa mouth,

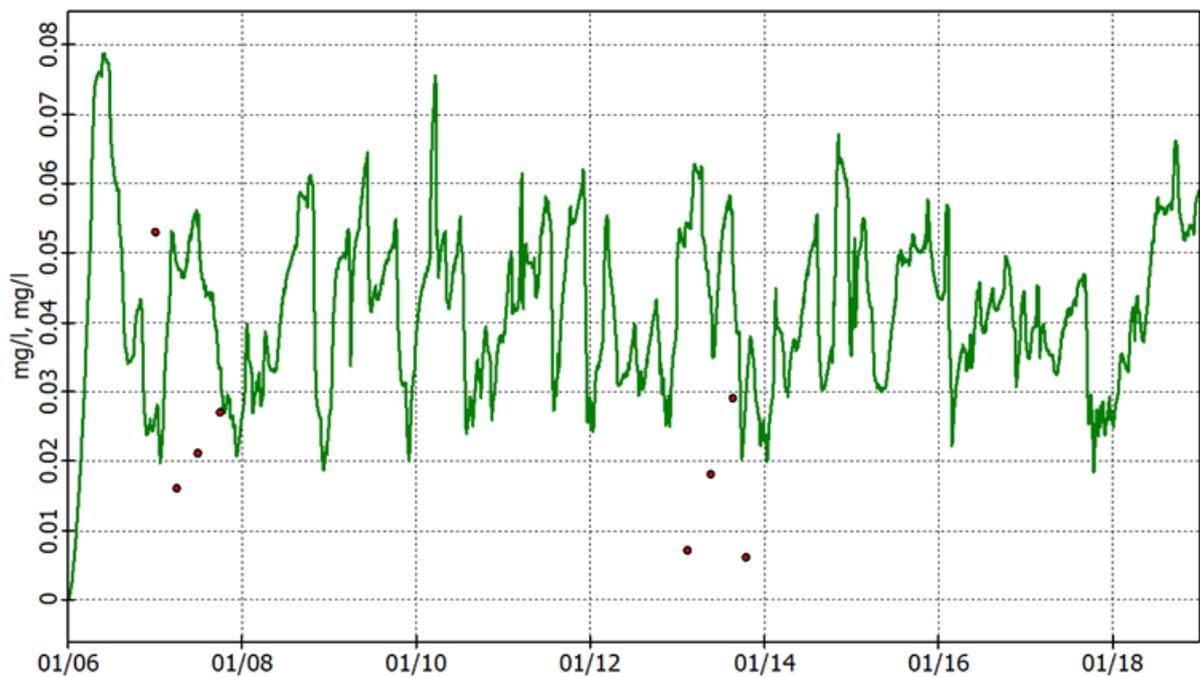


Figure 4: Time graph of observed and measured P-PO4 concentration at Zaņa mouth,

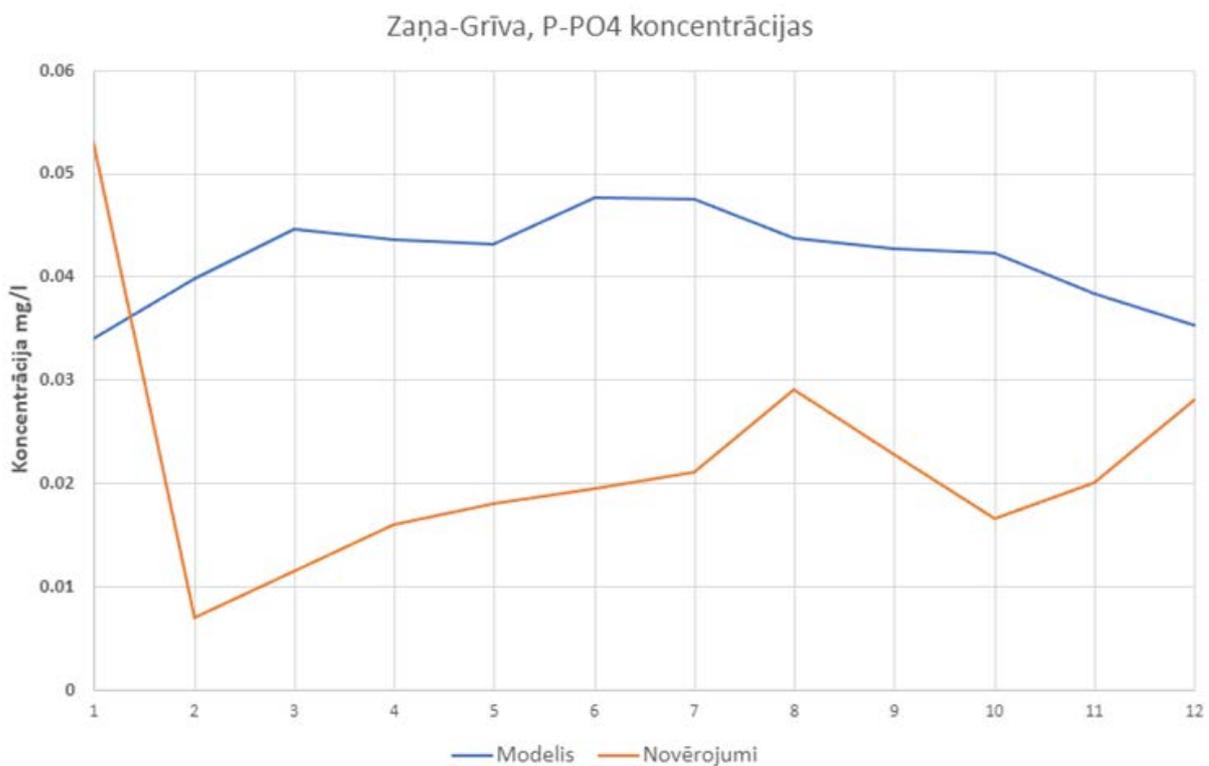


Figure 5: Observed and measured seasonal cycle of P-PO4 concentration at Zaņa mouth,

2.2. Fertilization timing

The fertilization timing measure requires modification of the management practice defined in the plant passports for involved plants. The management practice table is located in table *plantmng_fertilizertiming* under schema *measures*.

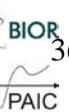
Following two tables show the default (Table 9) and the modified (Table 10) plant management praxis for winter wheat. Modified praxis distributes the application of the mineral nitrogen fertilizer more evenly in time (changes marked in green). There is some effect of this small change as could be seen in Figure 6.

Table 9: Default table of management praxis for winter wheat.

id	swat_id	opnpk	optype	hu	value	value2	value_s	day	mon
283	WWHT	1.5	Nitr	0.0493675	0.5				
284	WWHT	2.5	Nitr	0.2388165	0.25				
285	WWHT	3.5	Nitr	0.3505621	0.25				
286	WWHT	4.5	H	1.1					
287	WWHT	5.5	T	0.0057526	1				
288	WWHT	6.5	T	0.0791582	2				
289	WWHT	7.5	T	0.1476506	4				
290	WWHT	8.5	Phos	0.1476506	1				
291	WWHT	9.5	Manure	0.1476506	1				
292	WWHT	10.5	P	0.1541307	1115.2105				

Table 10: Modified table of management praxis for winter wheat.

id	swat_id	opnpk	optype	hu	value	value2	value_s	day	mon
283	WWHT	1.5	Nitr	0.0493675	0.4				
284	WWHT	2.5	Nitr	0.2388165	0.35				
285	WWHT	3.5	Nitr	0.3505621	0.25				
286	WWHT	4.5	H	1.1					
287	WWHT	5.5	T	0.0057526	1				
288	WWHT	6.5	T	0.0791582	2				
289	WWHT	7.5	T	0.1476506	4				
290	WWHT	8.5	Phos	0.1476506	1				
291	WWHT	9.5	Manure	0.1476506	1				
292	WWHT	10.5	P	0.1541307	1115.2105				



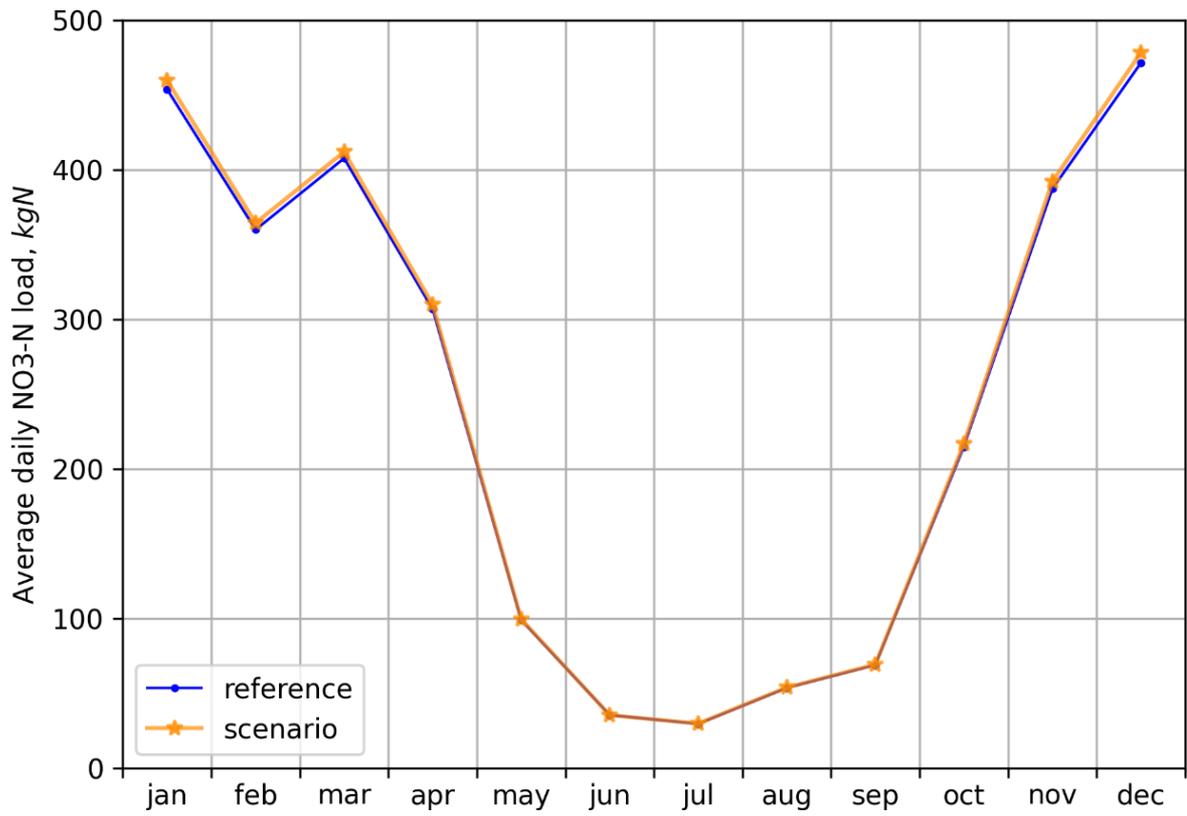


Figure 6: Effect of fertilization timing measure in Venta/Zaņa catchment, years 2008-2019.

2.3. Advanced fertilization techniques

Inject is the default setting in calibrated model – so no changes are expected using the default parameters. Selecting any other application type will produce notable changes. Figure 7 demonstrates results for the measure with *drill* application type.

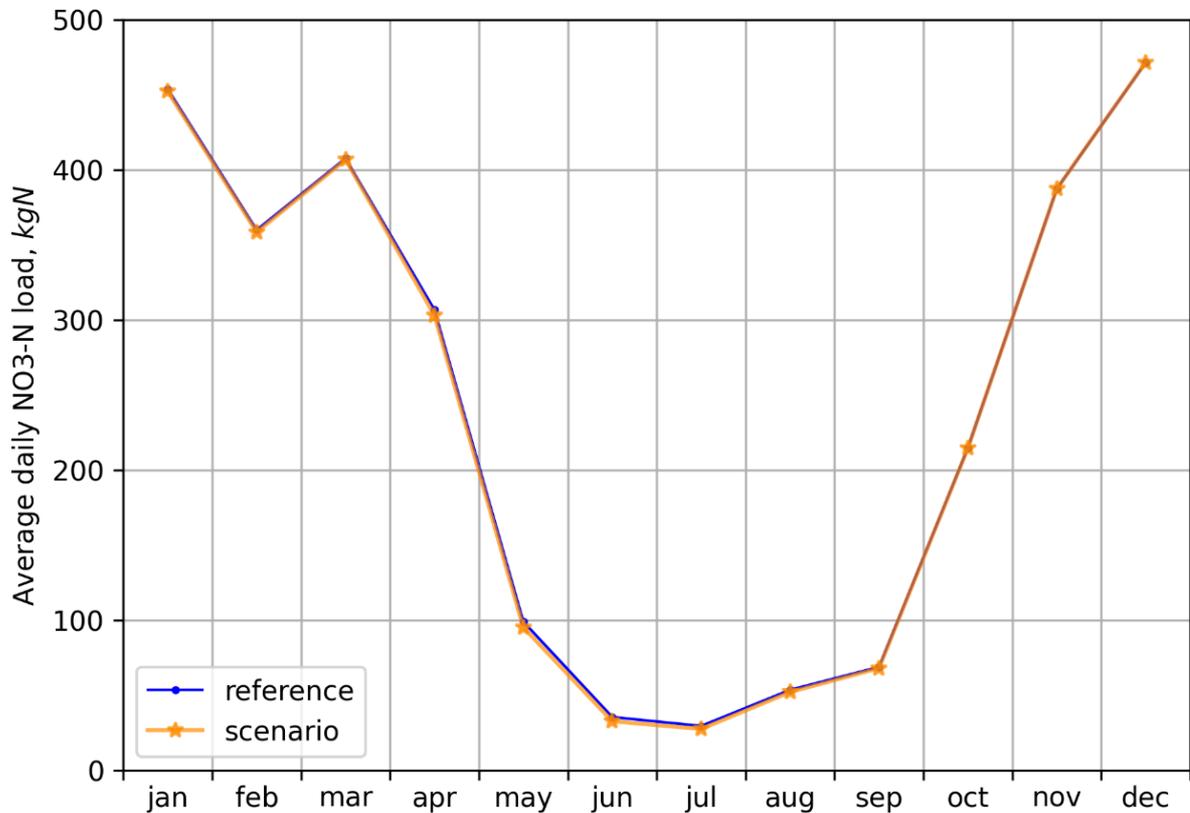


Figure 7: Effect of advanced fertilization (*drill*) measure in Venta/Zaņa catchment, years 2008-2019.

2.4. Fertilization reduction measure

Effect of 50% and 100% fertilization reduction is shown in Figures 8-9.

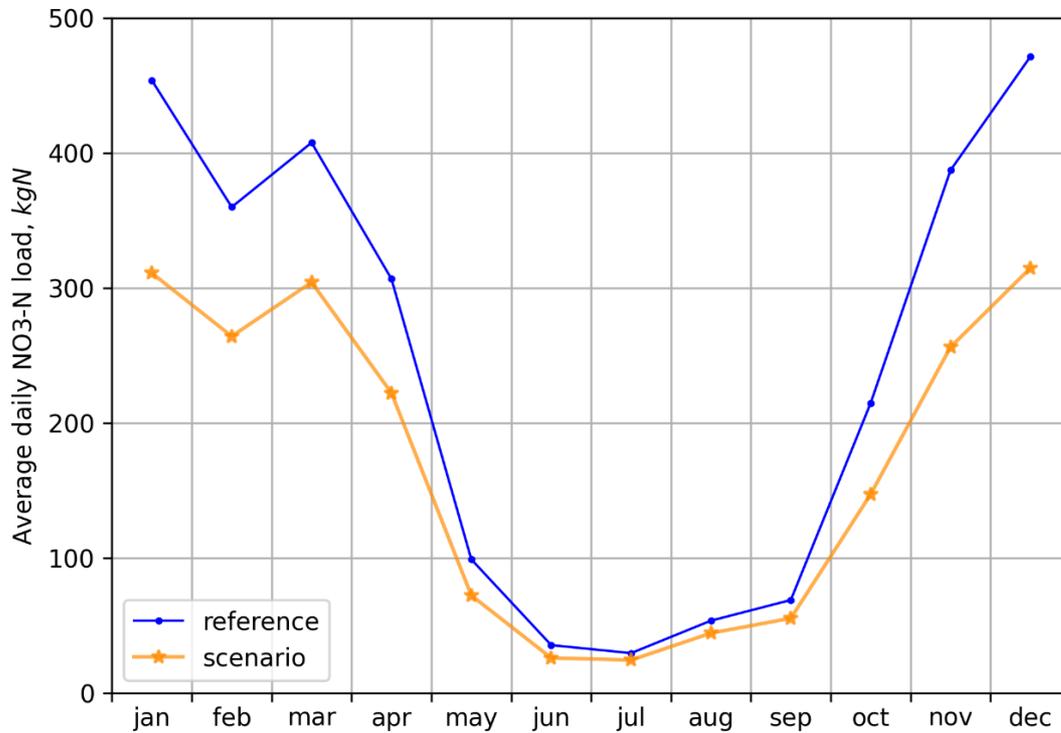


Figure 8: Effect of 50% fertilization reduction in Venta/Zaņa catchment, years 2008-2019.

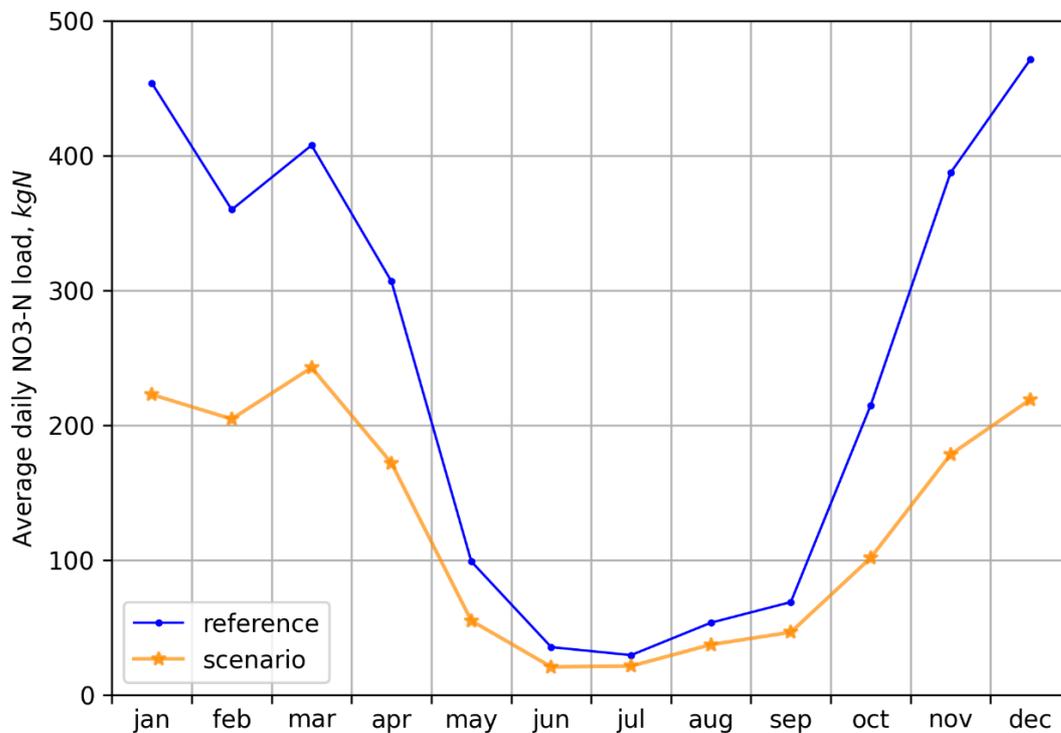


Figure 9: Effect of 100% fertilization reduction in Venta/Zaņa catchment, years 2008-2019.

2.5. Filter strips and grassed waterways

The filter strip measure is a HRU level measure. The filter strip parameters are applied to all HRUs that fall under selection criteria. User supplies these criteria in a call to *scenarios.implementMeasure* function. In the measure implementation we use the same parameters that SWAT+ uses to describe filter strips. See the effect of application of the filter strip measure in Figure 10.

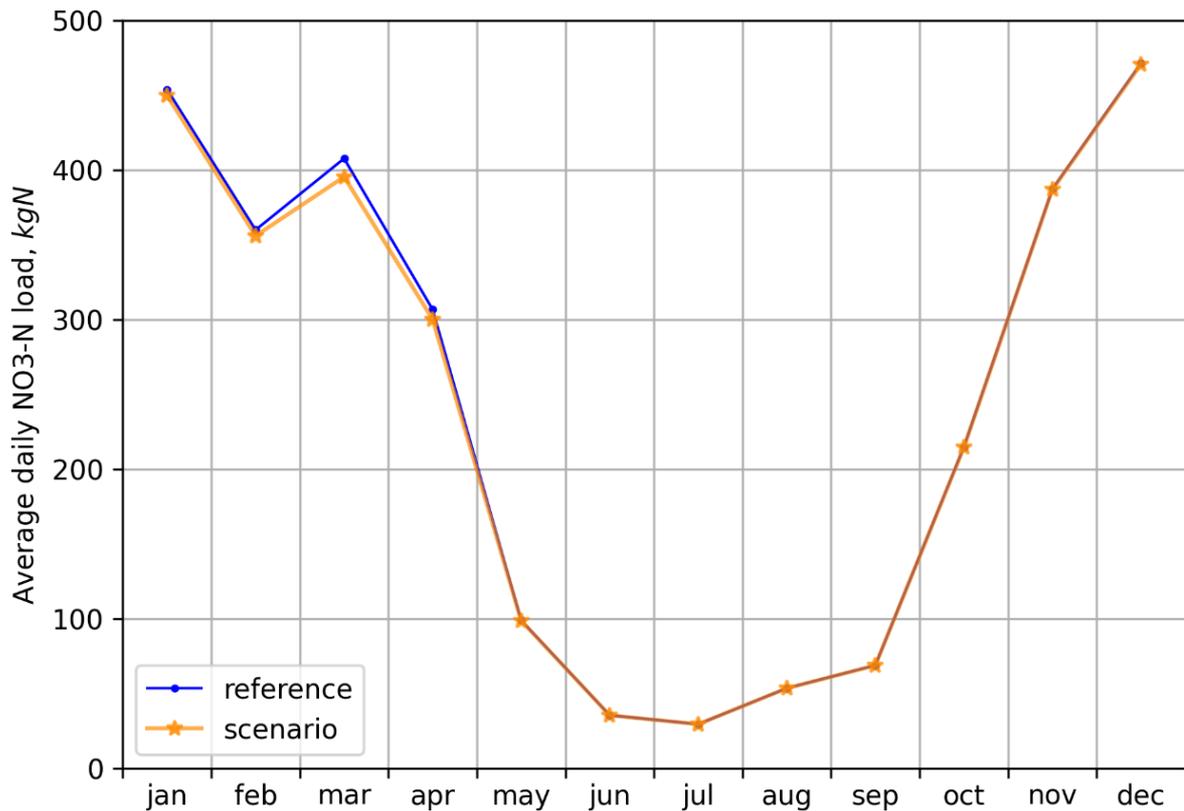


Figure 10: Effect of measure *filterstrips* on annual cycle of nitrate load in Venta/Zaņa catchment, years 2008-2019.

The same explanation applies also to grassed waterways measure. For the description of meanings of parameters, we refer to SWAT+ I/O manual.

When testing the grassed waterways measure, we also checked the SWAT+ source code for the implementation of grassed waterways and found multiple bugs there. The main error ensures that the grassed waterways are effectively switched off even if user requests them. We modified the SWAT+ source code to allow grassed waterways to be enabled. From the SWAT+ source code it is also obvious that grassed waterways do not have direct influence on the nitrate loads, but only to phosphates and organic nitrogen (see Figure 11) and phosphorus.

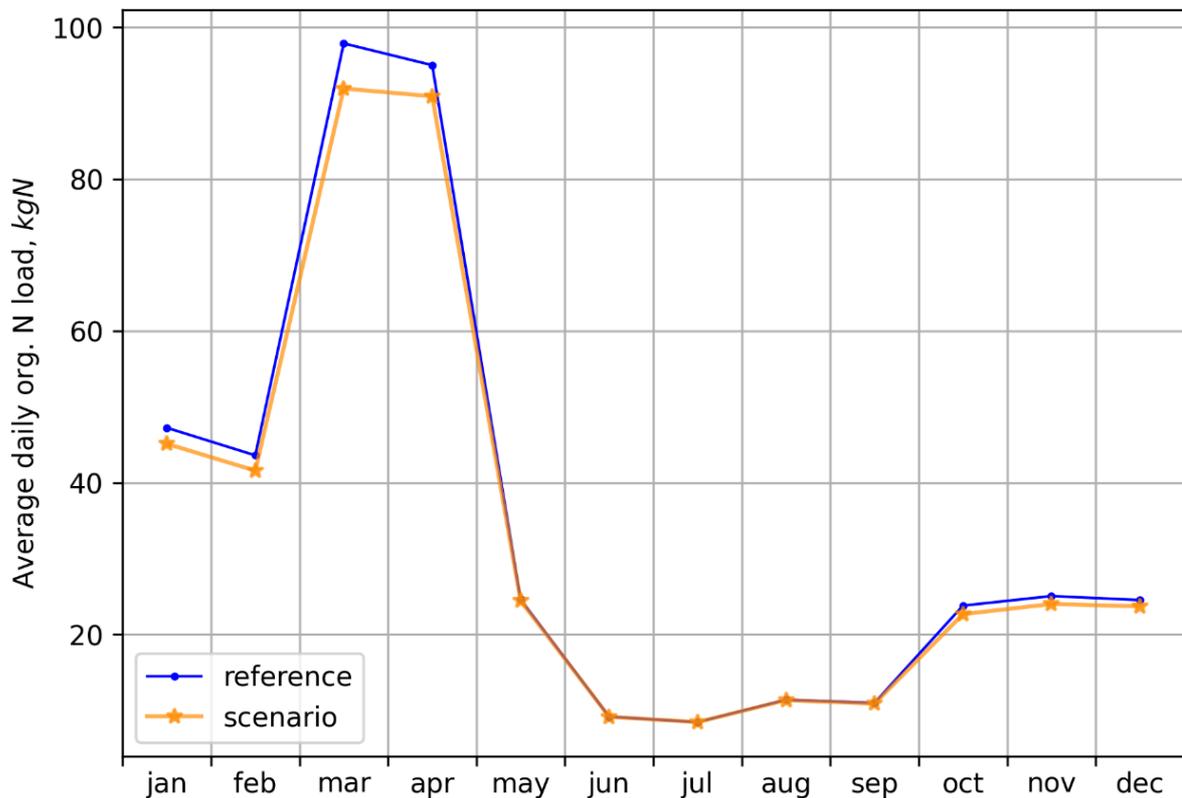


Figure 11: Effect of measure *grassedwaterways* on organic nitrogen load in Venta/Zaņa catchment, years 2008-2019.

2.6. Land use change

One should be careful when converting arable land to forests. The lower nitrate loads from the forest land in the model comes mainly from the absence of tile drains in the forest. In the measure script library it is possible to combine land use change measure with the measure that closes the tile drains (*measure_tiledrain*), by setting the depths of drains (parameter *dp*) to 0. To invoke such a scenario user should at first call the land use change measure and sequentially request the closure of tile drains on land use code *FRST*. The effect of such combined measure in reducing nitrate load is shown in Figure 12.

Results of changing of arable lands to grasslands are presented in Figure 13. We also note that changing the arable land to grasslands is achieved by converting agricultural land use codes to winter pasture *WPAS*. The management praxis for this type of land use contains also fertilization. If the measure is aimed at converting to the unmanaged perennial grasslands, it is advised to combine this measure with reduction of fertilization by 100%.

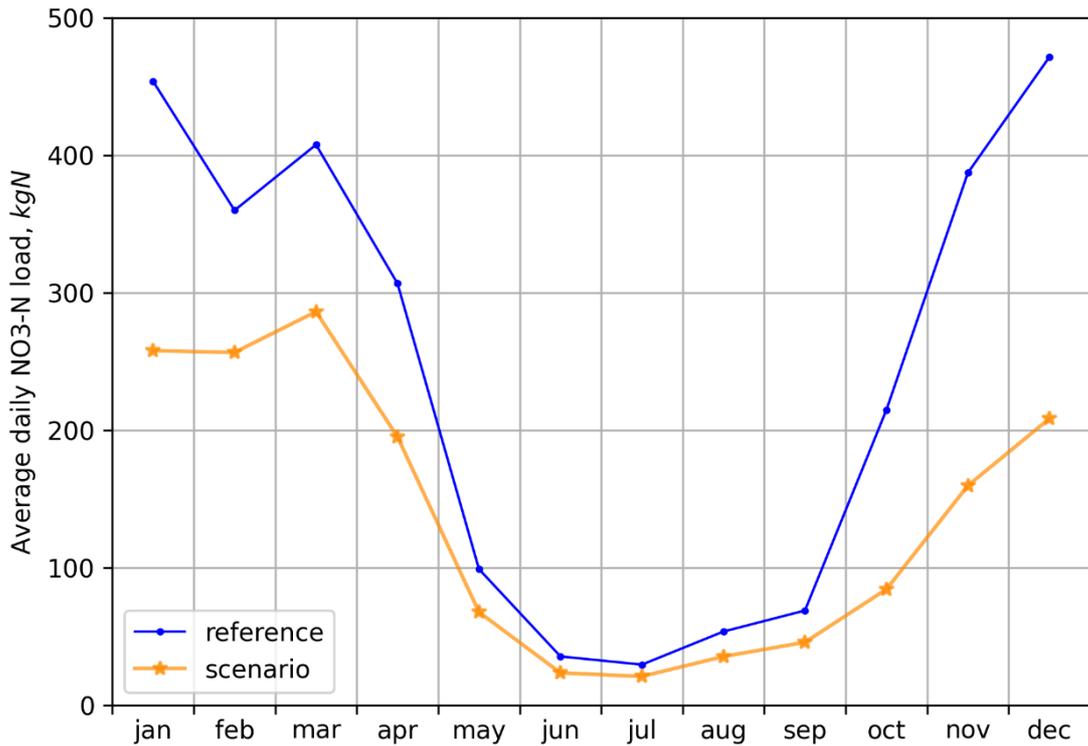


Figure 12: Effect of converting arable lands to forest combined with closure of tile drainage in Venta/Zaņa catchment, years 2008-2019.

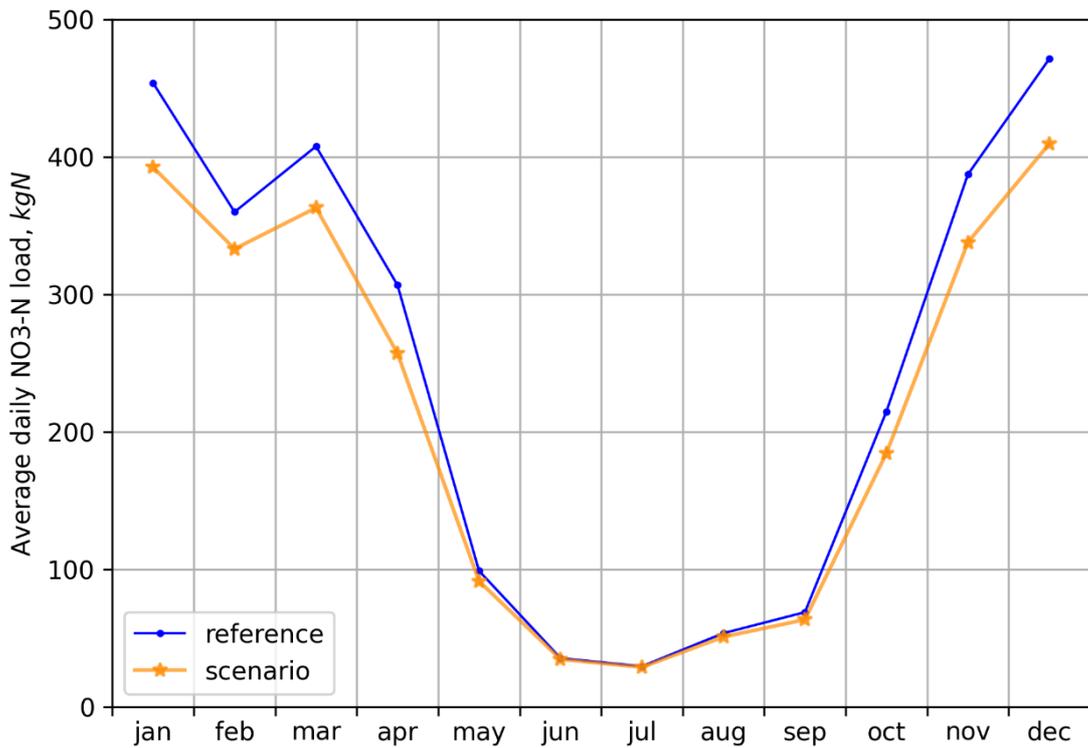


Figure 13: Effect of transforming arable lands to grasslands in SWAT+ for Venta/Zaņa catchment, years 2008-2019.

2.7. Managed drainage

The managed drainage measure allows changing of tile drainage depth by means of user supplied decision table. It is up to the user to provide this table. The example supplied with the modelling system allows for opening of tile drains by setting the depth of drains to 1100 mm in spring (day of year 65) and closing them in autumn by setting the depth of drains to 0 mm (day of year 295). Figure 14 shows the effect of the measure.

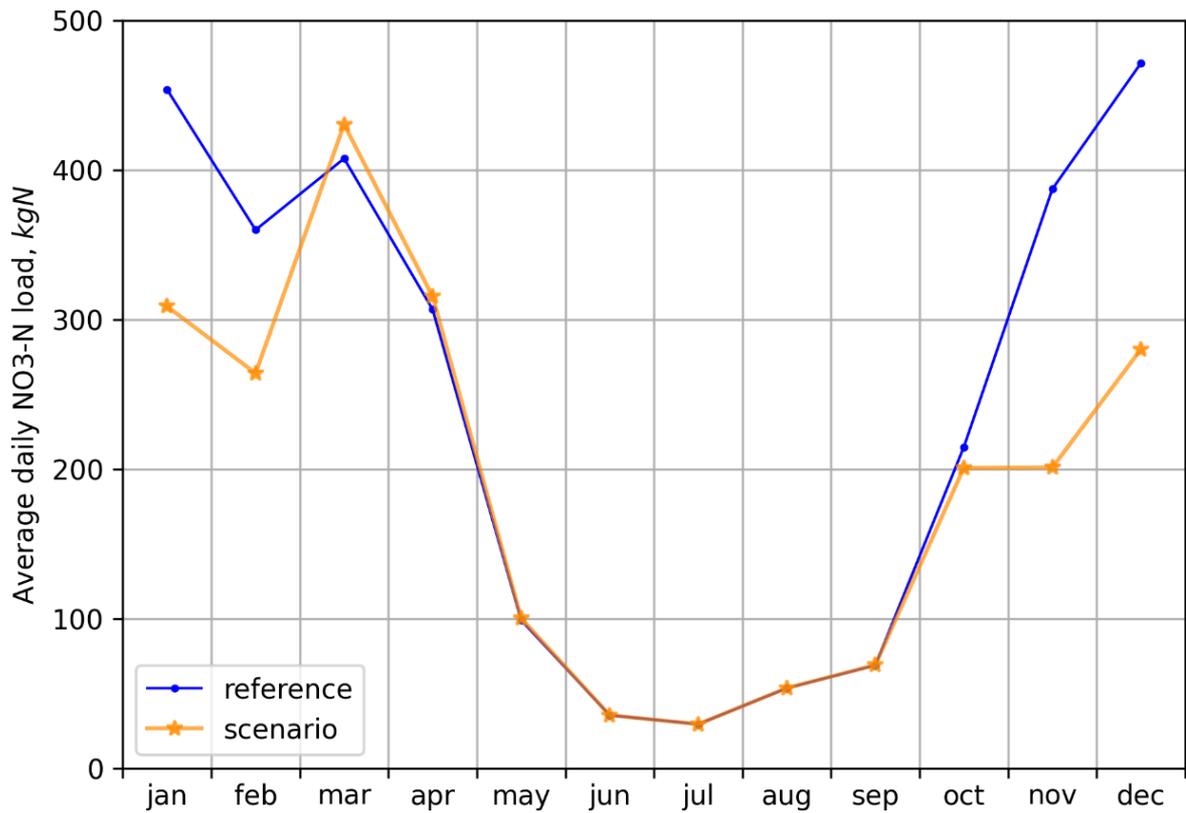


Figure 14: Effect of managed drainage measure in Venta/Zaņa catchment, years 2008-2019.

2.8. No-plough measure

The agricultural optimization measure “*No-plough technology*” relies on the management practice and parameters (*plant passports*) table for the measure. The plant passports for the measure are located in table *plantmng_noploughtechnology* of scheme *measures* in the PostgreSQL database of the project.

We checked the results of the measure, see Figure 15. In case the users need to modify the plant passports according to their needs, they can provide their own table of plant passports and place it under table *plantmng_noploughtechnology* of scheme *measures*.

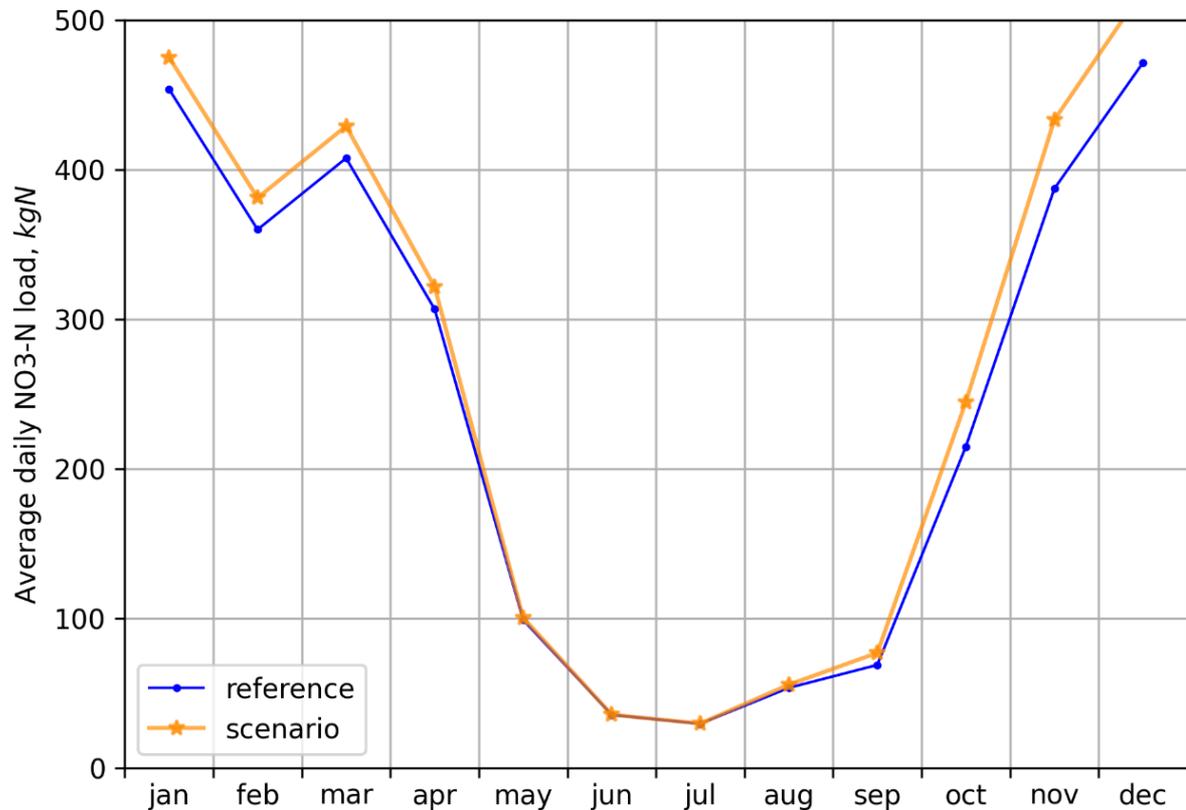


Figure 15: Effect of no-plough measure in Venta/Zaņa catchment, years 2008-2019.

2.9. Substituting autumn ploughing with spring ploughing

The agricultural optimization measure “*Substituting autumn ploughing with spring ploughing*” relies on the management practice and parameters (*plant passports*) table for the measure. The plant passports for the measure are in table *plantmng_substitutingautumnploughingwithspringploughing* of scheme *measures* in the PostgreSQL database of the project.

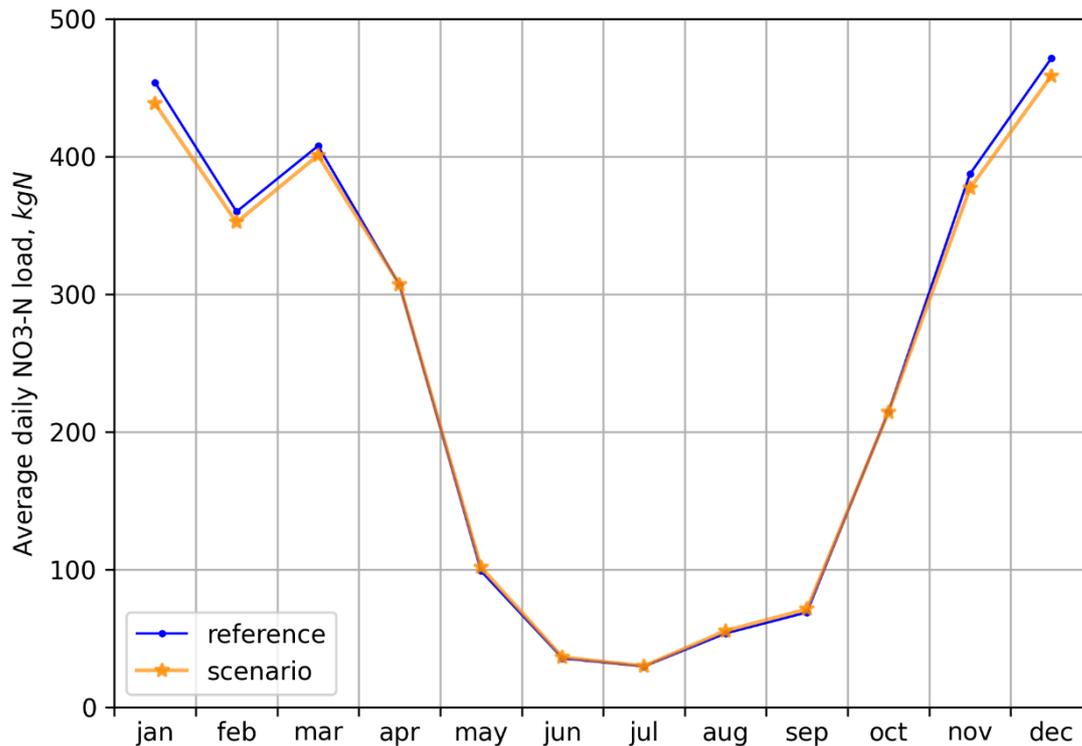


Figure 16: Effect of substituting autumn ploughing with spring ploughing in Venta/Zaņa catchment, years 2008-2019.

The results of the measure are shown for nitrate loads in Figure 16. In case when users need to modify the plant passports according to their needs, they can provide their own table of plant passports and place it under table *plantmng_substitutingautumnploughingwithspringploughing* of scheme *measures*.

2.10. Catch crops

The catch crop measure requires modification of the management practice which is provided in plant passports for involved plants. The management practice is defined in table *plantmng_catchcrops* under schema *measures*.

We checked that SWAT+ handling of management praxis involving two plants (in this case main crop and catch crop) is different from SWAT2012. There was also a bug in SWAT+ source code that incorrectly determined the plant code of a scheduled operation if there is more than one plant involved.

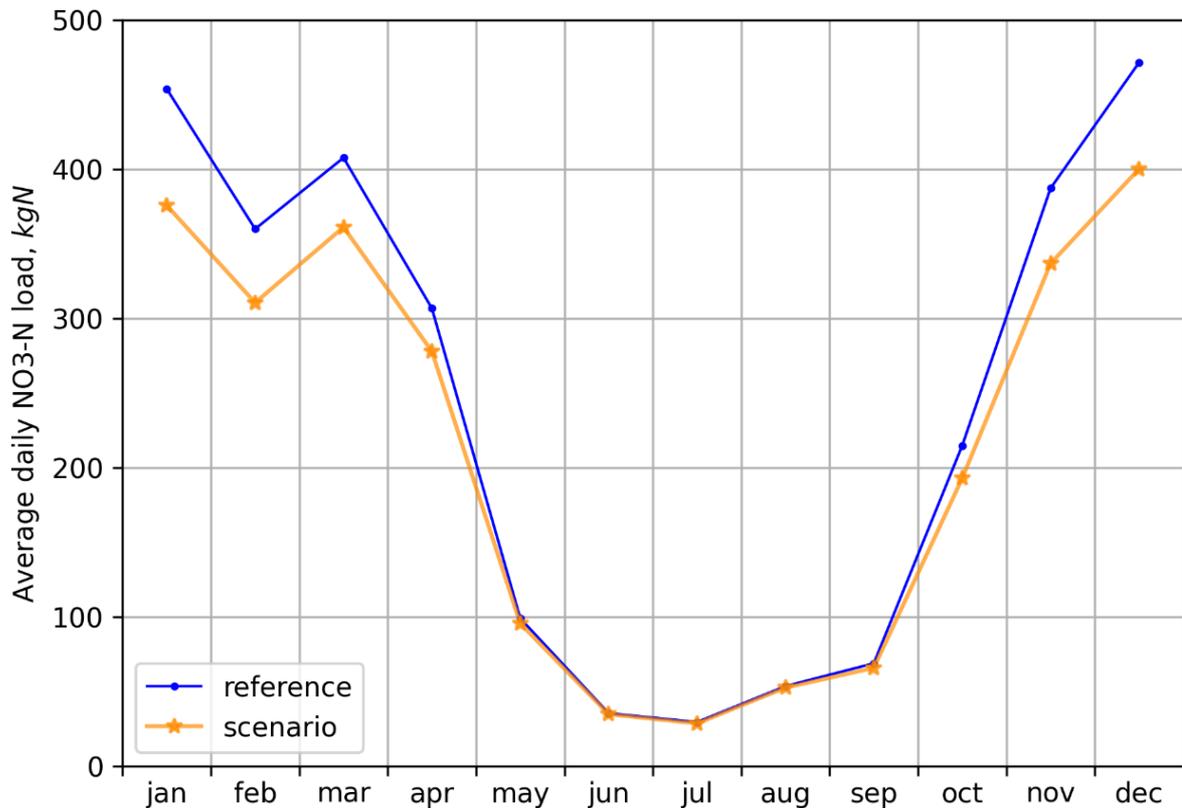


Figure 17: Effect of catch crop measure in Venta/Zaņa catchment, years 2008-2019.

We corrected the SWAT+ code. We also modified the script to allow for multiple plants in management praxis. The results of the measure effect for nitrate loads are shown in Figure 17.

2.11. Plant cover in winter

The measure of plant cover in winter requires a modification of the management practice in plant passports for involved plants. The management practice table is in table *plantmng_plantcoverinwinter* under schema *measures*.

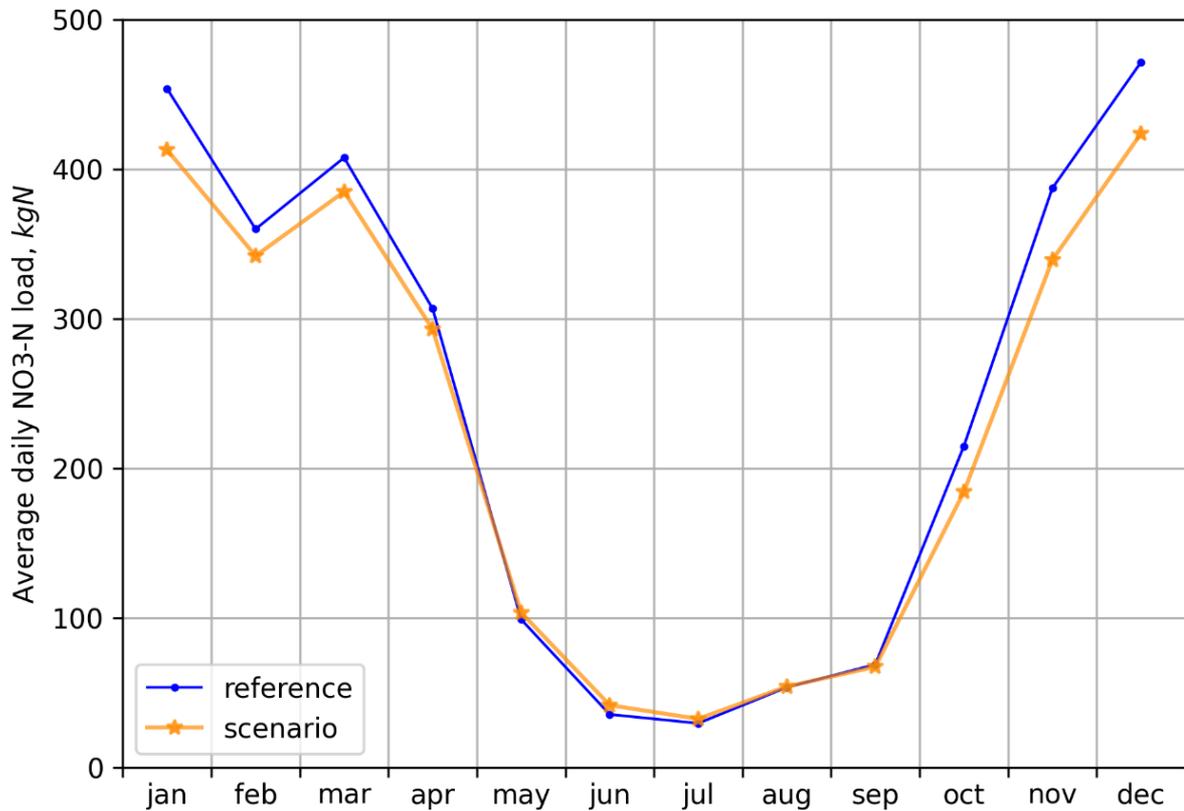


Figure 18: Effect of plant cover in winter in Venta/Zaņa catchment, years 2008-2019.

We checked that SWAT+ handling of management praxis involving two plants (in this case the main crop and catch crop) is different from SWAT2012. There was also a bug in SWAT+ source code that incorrectly determined the plant code of scheduled operation if there is more than one plant involved. We corrected the SWAT+ code. We also modified our script to allow for multiple plants in management practice. The results of the measure for the nitrate loads are shown in Figure 18.

References

PAIC, 2020. DATA BASE FOR MODEL IMPLEMENTATION. LIFE GoodWater IP deliverable R1, Rīga, 31p.

PAIC, 2022a. CALIBRATED AND VALIDATED MODELLING SYSTEM. LIFE GoodWater IP deliverable R2, Rīga, 47p.

PAIC, 2022b. MODELLING RESULTS FOR REFERENCE AND BASELINE SCENARIOS. LIFE GoodWater IP deliverable R3, Rīga, 30p.

PAIC, 2023a. Documentation of development, calibration, validation and results of SWAT modelling system. LIFE GoodWater IP deliverable R4, Rīga.

PAIC, 2023b. Complete modelling system – data, models, results. LIFE GoodWater IP deliverable R5, Rīga.

