## Basic use test

**1 Create a project**

**2 Upload energy model (ost.zip)**

**3 Set external parameters, for example**

**DHPrice** = 0.00011

**ElectricityPrice** = 0.00012

**SolarThInvCost** = 100

**StorageInvCost** = 150

-Create external parameters for each parameter

-Create external parameter set

-Assign the values for the parameters

**4 Create metrics for the project, for example**

**DHCost**

integrate(CHP\_HEAT\_NOM.MULTIPLYER\_OUTPUT, 0, Infinity) / HOUR\_S \* DHPrice

**ElectricityCost**

*integrate(ELEC\_BAL\_NOM.MULTIPLYER\_OUTPUT, 0, Infinity) / HOUR\_S \* ElectricityPrice*

**SolarThCost**

*SolarThInvCost \* (URBAN\_AREA\_01.SolarTH\_m2 + URBAN\_AREA\_02.SolarTH\_m2 + URBAN\_AREA\_03.SolarTH\_m2 + URBAN\_AREA\_04.SolarTH\_m2 + URBAN\_AREA\_05.SolarTH\_m2 + URBAN\_AREA\_06.SolarTH\_m2 + URBAN\_AREA\_07.SolarTH\_m2 + URBAN\_AREA\_08.SolarTH\_m2 + URBAN\_AREA\_09.SolarTH\_m2 + URBAN\_AREA\_10.SolarTH\_m2 + URBAN\_AREA\_11.SolarTH\_m2 + URBAN\_AREA\_12.SolarTH\_m2 + URBAN\_AREA\_14.SolarTH\_m2)*

**StorageCost**

StorageInvCost \* (URBAN\_AREA\_01.Tank\_Vol + URBAN\_AREA\_03.Tank\_Vol + URBAN\_AREA\_04.Tank\_Vol + URBAN\_AREA\_05.Tank\_Vol + URBAN\_AREA\_06.Tank\_Vol + URBAN\_AREA\_07.Tank\_Vol + URBAN\_AREA\_08.Tank\_Vol + URBAN\_AREA\_09.Tank\_Vol + URBAN\_AREA\_10.Tank\_Vol + URBAN\_AREA\_11.Tank\_Vol + URBAN\_AREA\_12.Tank\_Vol + URBAN\_AREA\_13.Tank\_Vol + URBAN\_AREA\_14.Tank\_Vol)

**5 Create scenarios (>2) for the project with different input parameter settings**

**-Alter** *A#\_Solarth\_m2*

**6 Simulate scenarios**

**7 Create a database optimization set, for example with expression**

(Minimize) DHCost + SolarThCost + StorageCost

**8 Run database search**

**9 Test visualization**

**-Different chart types**

**-Time series charts (using different output variables)**

**-Summary charts (with metrics)**

**10 Test table results**

## Genetic algorithm test

* log in
* create project
* upload ost.zip
* create optimization set, type = genetic algorithm
* create GA objective function: name = max\_heat, sense = maximize, expression = integrate(HEAT\_BALANCE\_NOM.MULTIPLYER\_OUTPUT, 0, Infinity) / 3600
* create GA objective function: name=min\_elec, sense = minimize, expression = integrate(ELEC\_BAL\_NOM.MULTIPLYER\_OUTPUT, 0, Infinity) / 3600
* create decision variable: name = a, type = Double, lower bound = 0, upper bound = 10000
* create decision variable: name = b, type = Double, lower bound = 1, upper bound = 2
* create constraint: name = prodconstraint, lower bound = 1000, expression = a\*b, upper bound = 18000
* edit model input parameters: select component URBAN\_AREA\_02, then edit
  + SolarTH\_m2 = a
  + Tank\_Vol =2000.0\*b
* select component URBAN\_AREA\_03, then edit
  + SolarTH\_m2 = 5000+a
  + Tank\_Vol = 5000.0\*b
* edit algorithm parameters:
  + max runtime = 30
  + population size = 10
  + number of parents per generation = 5
  + number of offspring per generation = 5
* run algorithm
* wait for 30+ minutes (the Opt4J GA backend only checks runtime between GA iterations, so depending on population size, it can take much longer than the maximum runtime)
* select “open scenario” to see the generated scenarios

## Grid search test

* Create project
* Import energy model ost.zip
* Create optimization set.  Name = ”test grid search”, Type = Genetic algorithm
* On the main genetic algorithm page, switch the type from “genetic algorithm” to “grid search” and press “Save”
* Press “Edit” button below the model input parameters table
* For creating 2 groups, press “New Group” button twice
* Select component OSTERSUNDOM\_ALUE
* Set A11\_pipe\_len = “5000.0, 8000.0” and select group G1
* Set A12\_pipe\_len = “5000.0, 10000.0” and select group G1
* Set A12\_Storage\_Vol = “15.0, 30.0” and select group G1
* Set Central\_storage\_Vol = “120.0, 240.0, 360.0” and select group G2
* Select “Ok”
* Select “Run algorithm” on the main genetic algorithm page
* Wait a few minutes for results to appear in the database.  They can be seen e.g. on the “Open scenario” page.

## Import data tests

### Import project

1. Create project  
2. Import testmodel.zip  
3. Import project file test-project.csv

### Import scenarios

First make test file for scenarios:

1. Create project  
2. Import ost.zip  
3. Create scenario  
4. Simulate scenario  
5. Export scenario file

1. Create project  
2. Import ost.zip  
3. Import the exported scenario file  
4. Check that there is a new scenario in the project

### Import optimization set

1. Create project  
2. Import ost.zip  
3. Import optimization set file ost-problem.csv

### Import optimization problem (GA)

1. Create project  
2. Import ost.zip  
3. Import optimization problem file ost-problem.csv

## Export data tests

### Export project

1. Create project  
2. Import testmodel.zip  
3. Import test-project.csv  
4. Export project files

### Export scenarios

1. Create project  
2. Import testmodel.zip  
3. Import test-project.csv  
4. Create scenario  
5. Export scenario file

### Export simulation results

1. Create project  
2. Import ost.zip  
3. Create scenario   
4. Simulate scenario  
5. Export simulation results   
6. Check that the file includes the values for all the output variables

### Export external parameter sets

1. Create project  
2. Import ost.zip  
3. Create external parameter  
4. Create external paremeter set  
5. Set value for the external parameter  
6. Export external parameter sets file

### Export metrics

1. Create project  
2. Import testmodel.zip  
3. Import test-project.csv  
4. Create external paremeter set  
5. Set values for the ext params  
6. Create scenario  
7. Simulate scenario  
8. Export metrics file

### Export optimization set

1. Create project  
2. Import ost.zip  
3. Create external parameter  
4. Create external parameter set  
5. Assign values for external parameters  
6. Create DB optimization set  
7. Create objective function  
8. Create constraint  
9. Export optimization set  
10. Check that the file includes the objective function, constraint and external parameter value

### Export optimization problem

1. Create project  
2. Import ost.zip  
3. Create external parameter  
4. Create external parameter set  
5. Assign values for external parameters  
6. Create GA optimization set  
7. Create objective function  
8. Create constraint  
8. Create decision variable  
9. Export optimization problem  
10. Check that the file includes the objective function, constraint, decision variable and external parameter value

## User management test

* Log in as an admin user
* Create users for each global user role: admin, expert, standard, guest
* Go to edit user page for the expert user
* Check that it shows the expert role
* Add standard role to one of the existing projects
* Check that the role is then listed in the page
* Edit guest user and add guest role to a project
* Check that the role is then listed in the page
* Log in as expert user
* Open the project that the user has access to
* Check that it succeeds
* Try to open any other project, should not be possible
* Try to create a new project, check that it succeeds
* Try to create a new scenario, check that it succeeds
* Log in as standard user
* Try to create a new project, should not be possible
* Log in as guest user
* Try to create a new project, should not be possible
* Open the project that the user has access to
* Try to create a scenario or optimization set, should not be possible
* Log in as an admin user
* Remove project roles of expert and guest users
* Check that they don’t appear in the role lists
* Remove user guest
* Check that it doesn’t appear in the list