SUSHEAT

Smart Integration of Waste and Renewable Energy for Sustainable Heat Upgrade in the Industry

D4.3 Bio-inspired TES Rules

Document Summary Information

Grant Agreement No.:	*101103552*	**Acronym:**	SUSHEAT
WP:	*WP 4 – Thermal Energy Storage: Design and Optimization*		
Task:	*Task(s) 4.3 – Bio-inspired Rules for the Design of TES tanks*		
Deliverable ID:	*D4.3 – Bio-inspired TES Rules*		
Submission date:	*2024-05-15*	**Due date:**	*2024-04-30*
Leader:	*Prof. Luisa F. Cabeza (UDL)*		
Participants:	*Dr. Carles Mateu (UDL), Dr. Emiliano Borri (UDL), Dr. Gabriel Zsembinszki (UDL), Nadiya Merhaj (UDL)*		
Dissemination level:	*Public*		
Reviewers:	*Antonio Rovira (UNED), Rubén Barbero (UNED)*		
Approver:	*Luisa F. Cabeza (UDL)*		

☒ Report ☐ Demonstrator ☐ Open Research Data Pilot ☐ Other

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. Grant Agreement No 101103552.

Executive Summary

This deliverable is the main outcome of the work carried out in “Task T4.3. Bio-inspired Rules for the Design of TES tanks”. This document details the rules and constraints that the biologically inspired design must adhere to. The set of constraints is twofold, one subset of those constraints is related to manufacturing constraints (maximum size, thickness, overhang limits, etc.), while the other subset are constraints related to the operation of a thermal energy storage. In this document basis of the genetic algorithm and objective function to achieve optimal design are described.

Revision history (quality control)

Version	Issue Date	% Complete	Changes	Contributor(s)
V1.0	2024-05-15	100%	Deliverable complete	Dr. Luisa F. Cabeza (UDL) Dr. Carles Mateu (UDL) Dr. Emiliano Borri (UDL) Dr. Gabriel Zsembinszki (UDL) Nadiya Merhaj (UDL)

Disclaimer

The content of the publication herein is the sole responsibility of the publishers, and it does not necessarily represent the views expressed by the European Commission or its services.

Any dissemination of results and any communication activity related to the action, reflects only the author's view and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the SUSHEAT consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the SUSHEAT Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the SUSHEAT Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© SUSHEAT Consortium, 2023-2027. All rights reserved. Licensed to the Innovation and Networks.

Table of contents

List of Tables

Glossary of terms and abbreviations

Acronyms

PCM = phase change material
GA = Genetic Algorithms
TES = Thermal Energy Storage
SLS = Selective Laser Sintering

Symbols

A= area
D = diameter
L = length
P = pressure
ΔP = pressure difference
V = volume

1. Introduction

This deliverable details the rules and constraints that the biologically inspired design must adhere to. The set of constraints is twofold, one subset of those constraints is related to manufacturing constraints (maximum size, thickness, overhang limits, etc.), while the other subset are constraints related to the operation of a TES, like temperatures, etc.

The rules are specifications on how the design process should proceed. They specify where and how the branching should occur, how many branches, lengths and widths, etc. As the main chosen algorithm for the optimization process is a genetic algorithm (GA) there are no rules as such, as the evolution with survival of the fittest rule used by the GA is the main driving rule for design evolution. What we have are a set of aspects or features (parameters) that the GA will mutate and that will define the shape, characteristics, and performance of the designs.

1.1. Mapping SUSHEAT Outputs

This deliverable is the main outcome of Task 4.3. Bio-inspired Rules for the Design of TES tanks which define all the design rules for the bio-inspired tank to be designed-manufactured, and tested in “Task T4.4. Artificial intelligence driven new TES designs” and “Task T4.5. Lab-scale experimental evaluation of the newly designed TES tank”.

Table 1. Adherence to SUSHEAT’s GA Deliverable & Tasks Descriptions.

TASKS	
Task 4.3. Bio-inspired Rules for the Design of TES tanks	**Section 2 to Section 3**
This task will detail the mathematical rules and equations that will govern the design process of the TES bioinspired repository. The first stage will involve solving the identified mathematical equations in calculation software such as Mathematica, Sage, Matlab and R. Subsequently, the equations will be translated into Java or Phyton for better use with AI in T4.4. This task will study (but is not limited to) the following structures: • Inspiration on plant leaves and animal/human lungs on the distribution of fluids. • Inspiration on plant leaves and animal/human lungs on fins design. • Inspiration on animal/human fluids distribution on bending of pipes.	The design parameters or genome components are described in Section 2.2 while the objective function is detailed in Section 3.
DELIVERABLE	
D4.3 - Bio-inspired TES Rules	**Section 2 to Section 3**
Equations governing the design and thermal processes for bio-inspired designs. The following structures will be analysed, considered for the design and modelled: plant leaves and animal/human lungs for distribution of fluids, fins design and bending of pipes. (T4.3)	The design parameters or genome components which will be used for the bio-inspired design are described in Section 2.2 while the objective function is detailed in Section 3.

1.2. Deliverable Overview and Structure

The deliverable provides the partners a glimpse on the inner working of the evolutionary algorithms used to create bio-inspired designs. It begins with an introduction to how Genetic Algorithms (GA) work. Although is not an extensive introduction, it will be especially useful for those not coming from an IT background, as most of the terms used by GA are quite different from those used in other areas of Artificial Intelligence (most terms come from biology, not from computer science).

Then a list of the genome components, the genes, that is the parameters of the design that the algorithms will use to create optimal designs follows. With the definition of the fitness function, akin to an objective function for an optimization algorithm, closing the document.

2. Biomimetic rules

The chosen design process to create the desired TES tank designs, genetic algorithms, is also inspired by nature's own work method, the evolution by survival of the fittest.

Genetic algorithms impose their own set of design rules to operate, requiring functioning that we provide with a *genome* that expresses the parameters and characteristics of the designed individual.

To properly introduce the rules, constraints, and parameters of the biomimetic designs, first, we will introduce, briefly, the key algorithm for design evolution, GA. That will help contextualize how the parameters and constraints guide the design process, thus, constituting the basic rules of the design. After that introduction to GA, those parameters and constraints will be introduced in the following sections.

2.1. Gentle introduction to genetic algorithms

Genetic Algorithms are a powerful computational approach inspired by the principles of natural evolution by survival of the fittest. In the field of design, genetic algorithms offer a fascinating method for generating, optimizing, and evolving solutions to complex design problems.

At its core, a genetic algorithm mimics the process of natural selection by iteratively evolving a population of potential solutions towards an optimal (or quasi-optimal) solution.

The genetic algorithm concept is simple: it works exactly as in nature, where advantageous traits are selected and passed on through generations through DNA inheritance. In a genetic algorithm, potential design solutions are represented as individuals within a population. These individuals, whose main characteristics are encoded in a *genome*, undergo processes of crossover, mutation, and selection, akin to the natural processes of reproduction, mutation, and survival of the fittest.

Through successive generations, genetic algorithms iteratively refine and improve upon these solutions, converging towards designs that meet specific criteria or objectives. This iterative process allows genetic algorithms to explore vast solution spaces efficiently, often discovering innovative designs that might not be immediately obvious through traditional design methodologies, an ability that makes them so desirable as an option for our project.

Evolutionary design via genetic algorithms has been applied in various domains, from engineering and architecture to art and product design. They are versatile tools for exploring and optimizing designs and uncovering novel solutions to complex problems.

2.1.1. Components of a genetic algorithm

The key components of a genetic algorithm, and the relation of those components to our project, are:

2.1.1.1. Representation

Each individual of our population, that represents a possible TES Tank design, has its defining characteristics expressed as a genome. That genome describes, in our case, things like how many branches will the main conduit divide into, at which point, with what angles, etc. Similar to how human DNA, our genome, expresses eye colour, hair colour, baldness tendency, etc.

This genome has to be encoded into a format suitable for manipulation by the algorithm. Usually, binary strings, byte sequences, real-valued vectors, or other suitable representations are used depending on the nature of the problem, the implementation details, programming language capabilities, etc.

2.1.1.2. Fitness Function

A fitness function is a mathematical function that evaluates the quality of each potential solution/individual within the population. It gives a measure of how well a design satisfies the requirements, objectives, and its performance as a TES tank. For example, a TES tank design should include the energy storage capacity, the charge and discharge speed, etc. And, in the end, return a *goodness* measure, i.e., how good is a tank design when compared to other designs. This is of special importance, as its use will be, mainly, to compare designs between them.

This fitness function then guides the selection process by assigning higher probabilities of reproduction to individuals/designs with higher fitness scores. It acts like the survival of the fittest in nature, those individuals/designs with better characteristics, that will have a better fitness score, will survive for the next generation and will reproduce, and generate new designs.

2.1.1.3. Selection

Just like natural selection, individuals/designs with higher fitness scores are more likely to be selected for reproduction than those with lower scores. Genetic algorithm (GA) programmers have various selection techniques: roulette wheel selection, tournament selection, and/or rank-based selection at their disposal to choose individuals for the next generation.

Selection ensures that those individuals that exhibit desirable traits and characteristics, better charge and discharge speeds, and higher storage capacity, continue *alive* onto the next generation, and are interbred with other survivals, generating new individuals/designs. And that those individuals with lower scores, i.e. worse performance, less storage capacity, etc. are eliminated from the population, thus removing those traits from the genetic pool.

2.1.1.4. Crossover

Crossover is the key process of the operation of genetic algorithms. During crossover, genetic material from selected individuals (parents) is mixed to create new individuals/designs (offspring). This process introduces diversity into the population by combining traits from different solutions. This process imitates what happens in nature when species and individuals reproduce. For instance, in humans, reproduction implies that the new offering genome will be composed of a mixture of both parents genomes, thus creating new individuals with a mix of characteristics of the reproducing parents.

2.1.1.5. Mutation

To avoid individuals/designs converging to suboptimal solutions, the local-maxima problem, mutations are required. Mutations introduce random changes into the genetic material of offspring solutions. This helps explore new regions of the solution space and prevents premature convergence to suboptimal solutions.

This phenomenon also occurs in nature, where spontaneous and random mutations occur in all living organisms, that, sometimes are passed down on to offspring, and, if those mutations offer an evolutionary advantage (better survivability, i.e., higher fitness) they become part of that species genetic traits (or, via speciation processes, result in newer species).

2.1.1.6. Exploration/Exploitation balance

As the population evolves over multiple generations, new offspring replacing less fitness individuals, it is possible that solutions converge to a non-optimal design, but to a suboptimal one. Besides mutation, other approaches as elitism strategies are used to balance exploration and exploitation balance of the solution space. A balance that is also managed to change during the different generations, going from an exploration-focused initial generations, to an exploitation-based approach on the latter ones. Exploration allows for more areas of the search space to be checked and probed fast, while the exploitation phase focuses on, once a basic almost optimal design is found, refine progressively that design.

2.2. Design parameters or genome components

Once it was established that we would use genetic algorithms for the evolution of the TES tank design, the key things one must define for genetic algorithms operation are how the genome is built, that is, which parts (design parameters) will be encoded into the genome. Those parameters will be what will give meaning to each part of the genome, mapping each gene to one physical characteristic of the tank design.

We will now provide a list of those genome components (parameters) that we have identified so far. This list is not closed, these are the minimum parameters that will be needed for any design process. During the project, as we continue to evolve more and more designs, and test them, some additional parameters may arise that can be considered and added to the genome.

2.2.1. Shell parameters

These parameters refer to the shell enclosing the tank. There are only two parameters, height and diameter (that will be constrained). We have not considered the thickness of the shell as a possible parameter as we believe that sticking to existing industry standard thickness will make much more sense, as, in the end, the shell has to be manufactured somehow.

The designed tanks are all cylindrical, as that is shape is the common practice in the industry, with the maximum diameter and height restricted by manufacturing limits.

2.2.1.1. Shell diameter

This parameter specifies the outer diameter of the shell. The diameter has minimum and maximum constraints, determined mainly by manufacturing capabilities.

Together with the shell height determine the total volume of the tank. They are both constrained together.

2.2.1.2. Shell height

This parameter specifies the outer height of the shell. Limited by fabrication capabilities.

Used to determine the total volume of the tank with the shell diameter. Both height and diameter are constrained together.

2.2.2. Parent pipe parameters

The parent pipe, the one entering the tank with the heat transfer fluid (HTF) will have a different treatment than the inner pipes on the tank, that way designs can deal with the fact that pipe comes from the outside in a fixed position.

2.2.2.1. Pipe length

The length of the entry pipe. Will be constrained.

2.2.2.2. Pipe diameter

The diameter of the entry pipe will be constrained. As with the thickness, it is possible that in the end designs and implementations, the diameter to be determined not by evolution by the GA, but by manufacturing constraints and directives, for example, limiting the diameter to a common or standard available diameter in the industry.

2.2.2.3. Pipe thickness

We have provided for the possibility that the thickness of the entry pipe could be determined by the GA. However, the most possible outcome from the following work packages is that the thickness would be better if fixed to a standard or common manufacturing pipe thickness.

2.2.3. Pipe parameters

These are the most important, core, parameters. They define how the pipes inside the TES tank are shaped. These parameters will be, in the initial iterations of the design algorithm the same (scaled if needed) for all layers of inner pipes.

That is, if we decide to have 3 layers (3 divisions of pipes), all layers will have the same parameters (adequately adapted), for instance, if the first layer has a branching angle α,the second and successive layers will also have an α branching angle. In successive iterations, these parameters will be unique for each layer, so, the first layer can have an angle α, the second layer can have a β branching angle, the third one a γ, and so on.

2.2.3.1. Branching factor (number of branches)

This parameter encodes the number of branches that will split from the main branch at each successive layer. The higher this parameter, the denser the coverage of pipes, but conversely, less space will be available for the storage material per se.

2.2.3.2. Branching point

The point in the parent pipe of this child in which the branching will take place. This is one of the parameters that define the basic geometry of the pipe network, this branching point parameter can be further refined (divided into different "*subparameters*") as the geometry of the network grows in complexity (due to intermediate experimental results and different manufacturing capabilities).

2.2.3.3. Branching angle

When a child branch splinters from its parent pipe it does it at a given angle. This angle is encoded in this gene/parameter. This parameter encodes a simple angle in the first iteration of the algorithm but will become a more complex parameter in the following iterations. Allowing for complex angles, even branching backwards if it is deemed interesting from a design point of view. In initial iterations all pipes in all the layers will branch at a given angle, in following iterations, different layers will have different branching angles. Even, the algorithm could allow for different angles in branches in the same layer, even if, for manufacturing constraints, this does not end in final designs.

2.2.3.4. Pipe length

It is the length of the child pipe before another branching. Besides other constraints, one obvious constraint is that the sum of the lengths of all the pipes projected on the central axis of the shell should equal the size (height) of the TES tank, i.e., pipes should cover all the distance between the HTF inlet and the outlet of the tank, obviously. As with all other parameters, initial iterations will keep this parameter equal for all layers, increasing complexity in successive iterations.

2.2.3.5. Pipe diameter

The diameter of the child pipes, as with other parameters, will be variable per layer in successive iterations. Similarly to the number of branches, high diameters will imply less space for thermal storage, and, of course, will have effects on pressure drop (one of the targets of the optimization process).

2.2.3.6. Pipe thickness

The thickness of the child pipes. This parameter will probably not be used in real designs, as per easing manufacturing of the tanks, sticking with common or standard industry thicknesses will be better.

2.2.3.7. Other geometric parameters

This list of parameters is an open one. As experimentation progresses and more and more design iterations are created, more complex geometries will be tested, and more and more geometric parameters to define such geometries will be added to the genome. For example, backwards branching, asymmetric branches, etc. will surely be tested, thus requiring the driving parameters to be added to the genome.

2.3. Design constraints

We will introduce now the constraints that can limit the possible designs: it is desirable to limit the possible designs. Although, theoretically, genetic algorithms can explore the whole search space (i.e., all possible design configurations), it is often desirable to limit this exploration, for several motives. First, reducing the search space reduces, also, the time needed to find an acceptable solution: genetic algorithms' search space often grows exponentially with respect to the complexity of the design, and limiting that growth can lead to significant time gains. Second, as the desired designs have to be manufactured, adding constraints that ensure that manufacturability is key.

2.3.1. Fabrication constraints

The prototypes for the tanks designed in this project will be fabricated via 3D printing due to the complex shapes of the biomimetic designs. That introduces a series of constraints stemming from the capabilities of the available metal 3D printers and from the technological limits imposed by metal 3D printing.

2.3.1.1. Shell maximum size

Whichever company prints the designs, will have a maximum printable volume on their equipment. The two companies we have, either worked with or are about to, have the following volumes:

- 30 cm x 30 cm x 30 cm
- 100 cm x 100 cm x 100 cm

That is a limit (a constraint) that will have to be imposed on the genetic algorithm so the design can be printed on those 3D printers.

2.3.1.2. Angles and overhangs

Depending on the technology used to 3D print the designs, either by fused deposition or by laser sintering (or any other technique), there are limits on the geometry and shape to be printed. For deposition techniques, overhangs (parts hanging in the air without support) are limited, with the most common limit around 60º. As for powder-based methods, like Selective Laser Sintering (SLS), the design has to allow for the non-sintered powder to have a way out of the design.

2.3.1.3. Minimum thickness

The same that happens for angles and overhangs, happens on minimum thickness of walls. Depending on the technology used to 3D print the design outer and inner walls will have a minimum width (called resolution in 3D printing technology). Although modern 3D printing has resolutions on the order of tenths and hundredths of millimeters, that, for our application should be enough, as walls will have to be wider than that to hold the pressure of the heat transfer fluid.

2.3.2. Geometric constraints

There are some additional constraints related to the global shape of the tank. In this case, we have decided on a cylindrical tank, as that shape is common in the industry. That implies other related constraints.

2.3.2.1. Shell height-to-width ratio

One constraint derived from the selection of a cylindrical shape for the tank is the width-to-to-height ratio. Most existing tanks have ratios around 1:3 (either tall tanks 1 width to 3 height, or long tanks, 3 width to 1 height). We have established this constraint as a working constraint for our algorithms. This will ease also the usage of the designed tanks in spots where previously existing tanks were placed, as shapes will be similar.

3. Fitness (objective) function

Genetic algorithms explore the possible design space in a guided way. Selection, mutation, crossover, and all functions of genetic algorithms are guided by a fitness function. The main goal of that function is to compare two individual designs, to select the best of them for survival and reproduction.

In broad terms, the fitness function measures the "*goodness*" of an existing individual/design. It is, then, the driving force of the genetic algorithm, and it is key to have a good fitness function. What makes a fitness function good is manyfold:

- Easy to compute: as this fitness has to be computed for each individual, on every generation, long computation times or heavier hardware requirements may render the problem at hand unassailable.
- As accurate as possible: the fitness function should be a good measure of how well an individual satisfies the requirements, and how optimal it is. The fitness score will be used to compare individuals against each other, removing those with lower scores: an inaccurate function could lead to removing promising individuals, thus stopping the exploration of areas of the search space that may have optimal solutions.

The fitness function can measure several aspects or parameters of the individual, in fact, its operation is similar to that of a multi-objective optimization function. The following sections is a non-comprehending list of the aspects that the fitness function measures, as more aspects can be added as experimentation progresses.

3.1. Effectiveness

The effectiveness can be described as the ratio of the actual heat discharged (Q_{act}) over the theoretical maximum heat that can be discharged (Q_{th}) [1] :

$$\varepsilon = \frac{Q_{act}}{Q_{th}} = \frac{\bar{T}_{HTF,out} - \bar{T}_{HTF,in}}{\bar{T}_{PCM} - \bar{T}_{HTF,in}} = 0.830477 - 17.2411 \cdot \frac{\dot{m}}{A} + 184.522 \cdot \left(\frac{\dot{m}}{A}\right)^2 - 1038.48 \cdot \left(\frac{\dot{m}}{A}\right)^3 + 3022.2 \cdot \left(\frac{\dot{m}}{A}\right)^4 - 4065.01 \cdot \left(\frac{\dot{m}}{A}\right)^5 + 1717.23 \cdot \left(\frac{\dot{m}}{A}\right)^6 \qquad (1)$$

3.2. Energy density

As we are designing a storage tank, the amount of thermal energy that the tank can store given a volume is a key aspect of its "*goodness*". Thus, a measure of that amount over the volume of the tank will be part of the fitness function:

$$ED_{PCM} = \frac{Heatstored}{Volumeoftheshell} \qquad (2)$$

3.3. Uniformity

One goal of the design will be to place pipes so not to leave gaps between different pipes. Big gaps or distances between sibling pipes will create chunks of PCM that are far enough from any pipe not to be affected by those pipes heat (not melting, so, not storing energy). As of now, the measure we are using to evenly spread those pipes is the standard deviation between the centres of the pipes (the inverse of the standard deviation, as less deviation is better).

During experimentation and on successive iterations of the genome specification, especially if asymmetric pipe configurations are implemented, other measures can, and probably will be implemented.

3.4. Pressure drop

An obvious measure for an individual, especially given the complex pipe network, will be that of the pressure drop experimented by the heat transfer fluid. Our optimal designs should have no or negligible pressure drop. It will be measured with:

$$\Delta P = f \cdot \frac{L}{D} \cdot \frac{\rho \cdot V^2}{2} \quad (3)$$

3.5. Parameter weights

The final fitness function will be a function combining (weighted sum) all the afore mentioned parameters. As of now, before experimentation, all parameters have equal weight, treating them as equally important. From experimentation, we will infer the relative importance of those parameters.

4. Conclusions

As we have seen, the use of genetic algorithms provides us with a flexible optimization framework, where we can, easily add new goals to consider when optimizing (via the fitness function), or more aspects of the tank that can be varied from design proposal to design proposal (via the parameters/genes).

By its very nature, genetic algorithms, also are highly parallelizable algorithms, which can come in handy as designs cover more parameters and the fitness function becomes much more complex and involved to compute. Thanks to this implicit parallelism we can use existing mass computation resources to perform evolution experiments in shorter times.

This document provides the list of required parameters (genes) that implementation must cover as a minimum. The list is not exhaustive, as there is room for improvement in the form of new genes codifying more convoluted shapes.

References

[1] A. Castell, M. Belusko, F. Bruno, L.F. Cabeza, Maximisation of heat transfer in a coil in tank PCM cold storage system, Appl Energy 88 (2011) 4120–4127. https://doi.org/10.1016/j.apenergy.2011.03.046.