SILESIAN UNIVERSITY OF TECHNOLOGY

FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

Engineer thesis

Design and implementation of web application used for solving vehicle routing problems with time windows.

author: Łukasz Kwiecień

supervisor: Tomasz Jastrząb, PhD

consultant: Name Surname, PhD

Gliwice, October 2021

załącznik nr 2 do zarz. nr 97/08/09

Oświadczenie

Wyrażam zgodę / Nie wyrażam zgody* na udostępnienie mojej pracy dyplomowej / rozprawy doktorskiej*.

Gliwice, dnia 31 października 2021

.....
(podpis)

.....
(poświadczenie wiarygodności
podpisu przez Dziekanat)

* podkreślić właściwe

Oświadczenie promotora

Oświadczam, że praca „Design and implementation of web application used for solving vehicle routing problems with time windows.” spełnia wymagania formalne pracy dyplomowej inżynierskiej.

Gliwice, dnia 31 października 2021

.....

(podpis promotora)

Contents

Chapter 1

Introduction

- introduction into the problem domain
- settling of the problem in the domain
- objective of the thesis
- scope of the thesis
- short description of chapters
- clear description of contribution of the thesis's author – in case of more authors table with enumeration of contribution of authors

Chapter 2

[Problem analysis]

- problem analysis
- state of the art, problem statement
- literature research (all sources in the thesis have to be referenced [**?**, **?**, **?**, **?**])
- description of existing solutions (also scientific ones, if the problem is scientifically researched), algorithms, location of the thesis in the scientific domain

Chapter 3

Requirements and tools

- functional and nonfunctional requirements
- use cases (UML diagrams)
- description of tools
- methodology of design and implementation

Chapter 4

External specification

- hardware and software requirements
- installation procedure
- activation procedure
- types of users
- user manual
- system administration
- security issues
- example of usage
- working scenarios (with screenshots or output files)

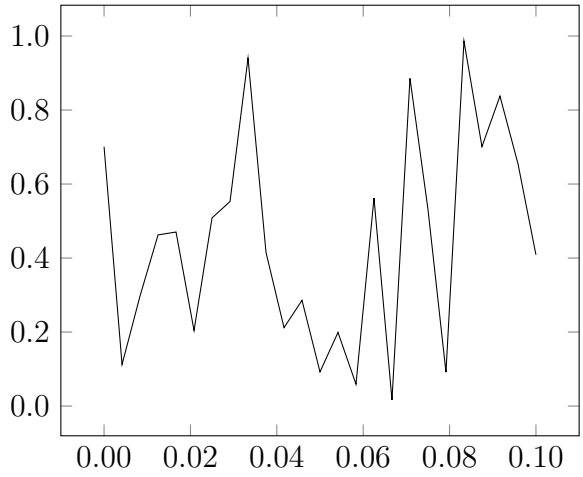

Figure 4.1: A caption of a figure is **below** it.

Chapter 5

Internal specification

- concept of the system
- system architecture
- description of data structures (and data bases)
- components, modules, libraries, resume of important classes (if used)
- resume of important algorithms (if used)
- details of implementation of selected parts
- applied design patterns
- UML diagrams

Use special environment for inline code, eg **descriptor** or **descriptor_gaussian**. Longer parts of code put in the figure environment, eg. code in Fig. 5.1. Very long listings–move to an appendix.

```
1  class descriptor_gaussian : virtual public descriptor
2  {
3      protected:
4          /** core of the gaussian fuzzy set */
5          double _mean;
6          /** fuzzyfication of the gaussian fuzzy set */
7          double _stddev;
8
9      public:
10         /** @param mean core of the set
11             @param stddev standard deviation */
12         descriptor_gaussian (double mean, double stddev);
13         descriptor_gaussian (const descriptor_gaussian & w
               );
14         virtual ~descriptor_gaussian();
15         virtual descriptor * clone () const;
16
17         /** The method elaborates membership to the
               gaussian fuzzy set. */
18         virtual double getMembership (double x) const;
19
20 };
```

Figure 5.1: The **descriptor_gaussian** class.

Chapter 6

Verification and validation

- testing paradigm (eg V model)
- test cases, testing scope (full / partial)
- detected and fixed bugs
- results of experiments (optional)

Chapter 7

Conclusions

- achieved results with regard to objectives of the thesis and requirements
- path of further development (eg functional extension ...)
- encountered difficulties and problems

Table 7.1: A caption of a table is **above** it.

	method						
			alg. 3			alg. 4, $\gamma = 2$	
ζ	alg. 1	alg. 2	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Appendices

List of abbreviations and symbols

DNA	deoxyribonucleic acid
MVC	model–view–controller
N	cardinality of data set
μ	membership function of a fuzzy set
$\mathbb{E}$	set of edges of a graph
$\mathcal{L}$	Laplace transformation

Listings

(Put long listings in the appendix.)

```
1  partition fcm_possibilistic::doPartition
2                                  (const dataset & ds)
3  {
4     try
5     {
6        if (_nClusters < 1)
7           throw std::string ("unknown number of clusters"
               );
8        if (_nIterations < 1 and _epsilon < 0)
9           throw std::string ("You should set a maximal
               number of iteration or minimal difference --
               epsilon.");
10       if (_nIterations > 0 and _epsilon > 0)
11          throw std::string ("Both number of iterations
               and minimal epsilon set -- you should set
               either number of iterations or minimal
               epsilon.");
12
13       auto mX = ds.getMatrix();
14       std::size_t nAttr = ds.getNumberOfAttributes();
15       std::size_t nX    = ds.getNumberOfData();
16       std::vector<std::vector<double>> mV;
17       mU = std::vector<std::vector<double>> (_nClusters)
```

```
        ;
18      for (auto & u : mU)
19          u = std::vector<double> (nX);
20      randomise(mU);
21      normaliseByColumns(mU);
22      calculateEtas(_nClusters, nX, ds);
23      if (_nIterations > 0)
24      {
25          for (int iter = 0; iter < _nIterations; iter++)
26          {
27              mV = calculateClusterCentres(mU, mX);
28              mU = modifyPartitionMatrix (mV, mX);
29          }
30      }
31      else if (_epsilon > 0)
32      {
33          double frob;
34          do
35          {
36              mV = calculateClusterCentres(mU, mX);
37              auto mUnew = modifyPartitionMatrix (mV, mX);
38
39              frob = Frobenius_norm_of_difference (mU,
                    mUnew);
40              mU = mUnew;
41          } while (frob > _epsilon);
42      }
43      mV = calculateClusterCentres(mU, mX);
44      std::vector<std::vector<double>> mS =
            calculateClusterFuzzification(mU, mV, mX);
45
46      partition part;
47      for (int c = 0; c < _nClusters; c++)
```

```
48        {
49            cluster cl;
50            for (std::size_t a = 0; a < nAttr; a++)
51            {
52                descriptor_gaussian d (mV[c][a], mS[c][a]);
53                cl.addDescriptor(d);
54            }
55            part.addCluster(cl);
56        }
57        return part;
58    }
59    catch (my_exception & ex)
60    {
61        throw my_exception (__FILE__, __FUNCTION__,
              __LINE__, ex.what());
62    }
63    catch (std::exception & ex)
64    {
65        throw my_exceptionn (__FILE__, __FUNCTION__,
              __LINE__, ex.what());
66    }
67    catch (std::string & ex)
68    {
69        throw my_exception (__FILE__, __FUNCTION__,
              __LINE__, ex);
70    }
71    catch (...)
72    {
73        throw my_exception (__FILE__, __FUNCTION__,
              __LINE__, "unknown␣expection");
74    }
75 }
```

Contents of attached CD

The thesis is accompanied by a CD containing:

- thesis (LaTeX source files and final `pdf` file),
- source code of the application,
- test data.

List of Figures

List of Tables