A Real-Time Framework for a DEVS-based Migraine Prediction Simulator System

Josué Pagán[1,3], José L. Risco-Martin[1], José M. Moya[2,3], and José L. Ayala[1]

[1] Computer Architecture and Automation Department (DACYA), Complutense University of Madrid, Spain
{jpagan,jlrisco,jayala}@ucm.es
[2] LSI - Electronic Engineering Department
Technical University of Madrid, Spain
josem@die.upm.es
[3] CCS-Center for Computational Simulation, Spain
{josueportiz,josem}@die.upm.es

Abstract. The migraine disease is one of the most disabling neurological diseases that negatively impacts on the quality of life and on the cost of the public health services. The prediction of a migraine symptomatic event through monitorization of hemodynamic variables has been previously demonstrated in our previous works. In this paper, a first approach for the development of a simulator for a real time migraine prediction system is shown. The simulator has been implemented using a formal description language and validated using Grammatical Evolutionary models. The results encourage to develop real time techniques to trigger accurate alarms and real time repairing techniques of disrupted signals. All these problems will be faced in our future work by HW/SW co-simulation and including Hardware In the Loop components, in order to simulate failures in sensors or trigger alarms.

Keywords: Simulation, DEVS, Grammatical Evolution, prediction, migraine

1 Introduction

The migraine is a recurrent and chronic headache and one of the most disabling neurological diseases. The migraine affects around 10% of population worldwide (1; 2), and 15% in Europe (3) leading to costs of €1,222 per European patient per year (4).

A migraine is a sequence of neurological processes before the pain starts. To avoid the pain, the intake of the drugs must be in advance to the beginning of these processes—otherwise the effectiveness of the drugs is low or null. The migraine causes fatigue, anxiety or cardiovascular problems, among others. In general, the migraine disease degrades the quality of life of the migraine sufferers (migraineurs).

Predicting the income of a symptomatic event allows patients to take the pills in advance in order to stop the pain. In previous works we have faced the problem of prediction of migraines by modeling its symptoms; and it was demonstrated by the authors in previous works (5; 6) that prediction of the symptoms is feasible through the analysis of the changes in four hemodynamic variables controlled by the Autonomous Nervous System (ANS): skin temperature, electrodermal activity (EDA), oxygen saturation (SpO2) and heart rate (HR).

Once we have proved that migraines can be modeled and we have effectively calculated prediction models, the next step in this research is to perform the prediction in real time. For gathering data from ambulatory patients, a co-designed monitoring device has been developed in collaboration with the company M2C (7). However, this prototype does not have an alarm interface yet and, in addition, the access to patients for conducting the study is a difficult task. Therefore, we have decided to simulate a real time predictive alarm system that integrates our migraine models. This paper focuses on describing our first approaches on the design of such a simulator of a migraine prediction system. The real time alarm system is under development, and the remaining of the modules of the simulator are presented in this paper and validated using Grammatical Evolutionary (GE) algorithms to predict migraines.

The final aim is to substitute all the software elements of the simulator for hardware modules. This goal is going to be reached gradually using co-simulation hardware-software. This is an incremental design with easy block substitution for which a formal language description simulator-based is preferred. As formalism, the Discrete Event Systems Specification (DEVS) has been chosen to model and simulate our system. DEVS (8) allows to try the simulator by inclusion of Hardware In the Loop (HIL)—something tested in (9).

In this paper we use the xDEVS open source Java library (10) with the aim of making a future implementation on a hardware device. DEVS is a modular and hierarchical modeling formalism, with all of the advantages and uses of simulation systems, such as: completeness, verifiability, extensibility, and maintainability and allows execution of Monte Carlo simulations, parallel simulation using threads or distributed using webs (11), as an example.

There are also other simulators such as the Ptolemy II (12), a project describing discrete-event models through a semantic focused on application to cyber-physical and embedded systems; or Simulink (13) from MATLAB, which is a not formal tool and engineering-oriented. Among all these possibilities, we have decided to use xDEVS because of the previous experience in the team with it and satisfaction with its performance.

In this paper a DEVS-based simulator is shown to validate the proof of concept of migraine prediction modeling using the previously developed GE models in (6). A top-down view of the simulator is drawn. First, a general overview is shown, to specify later each of the atomic and coupled models of the simulator.

We can find in the literature other simulators for diseases and epidemics. Barhak *et al.*present in (14) an open source simulator software tool for chronic diseases. Their simulator uses Markov transition models, what constraints this tool to the knowledge of the probabilities of transitions. Other example with limitations is Archimedes (15), a commercial simulator for diabetes using an object-oriented approach. Both of them implement a Graphical User Interface (GUI). Our simulation environment is published as Open Source under General Public License (GPL) in (10) but does not implement a GUI yet. In our team, we are working on implementation of a Unified Modeling Language (UML) executable interface as done in (16).

In the remainder of this paper: the specifications on DEVS formalism is shown in Section 2. The experimental set-up is shown in Section 3; where simulator and its parameters are shown. Section 4 shows the evaluation of the simulator and the final discussions are drawn in Section 5.

2 DEVS specification

DEVS is a general formalism for discrete event system modeling based on a mathematical Set Theory (8) to easily implement a formally described system using an existing software/hardware library.

DEVS formally represents a system by three sets: input (X), output (Y) and state (S), and five functions: time advance (ta), external transition (δ_{ext}), internal transition (δ_{int}), confluent (δ_{con}) and output (λ).

There are two types of models in DEVS: atomic and coupled. An atomic model is irreducible and it specifies the behavior for any modeled entity: processes an input event based on its state and condition, and generates an output event and changes its state. An atomic model is defined by the following equation:

$$A = \langle I, O, X, S, Y, \lambda, \delta_{\text{int}}, \delta_{\text{ext}}, \delta_{\text{con}}, ta\rangle \tag{1}$$

where:

- I is the set of input ports.
- O is the set of output ports.
- X is the set of inputs described in terms of pairs port-value: $\{p, v\}$.
- S is the state space. It includes the current state of the atomic model and also two special parameters called σ and *phase*. σ is the time until the next event generation, and the *phase* is a description of the current state (usually in natural language).
- Y is the set of outputs, also described in terms of pairs port-value: $\{p, v\}$.
- $\lambda : S \rightarrow Y$ is the output function. When the time elapsed since the last output function is equal to σ, then λ is automatically executed.
- $\delta_{int} : S \rightarrow S$ is the internal transition function. It is used to change the state S, *phase* and σ, and it is executed right after the output function (λ).
- $\delta_{ext} : Q \cdot X^b \rightarrow S$ is the external transition function. It is automatically executed when an external event arrives to one of the input ports, changing the current state if needed.
  - $Q = (s, e), s \in S, 0 \leq e \leq ta(s)$ is the total state set, where e is the time elapsed since the last transition.
  - X^b is a bag of elements of X.
- $\delta_{con} : Q \cdot X^b \rightarrow S$ is the confluent function, subject to $\delta_{con}(s, \emptyset) = \delta_{int}(s)$. This transition is selected if δ_{ext} and δ_{int} must be executed at the same instant.
- $ta(s) : S \rightarrow \Re_0^+ \cup \infty$ is the time advance function.

A coupled model aggregates and interconnects two or more atomic or coupled models. And it is formally described as:

$$M = \langle I, O, X, Y, C_i, EIC, EOC, IC \rangle \quad (2)$$

where:

- I, O are the set of external (not coupled) input and output ports.
- X is the set of external input events.
- Y is the set of output events.
- C_i is a set of DEVS component models (atomic or coupled). Note that C_i makes this definition recursive.
- EIC is the external input coupling relation.
- EOC is the external output coupling relation.
- IC is the internal coupling relation.

Due to the definition in Eq. 2, a coupled model can itself be a part of a component in a larger coupled model system giving rise to a hierarchical DEVS model construction.

3 Experimental Set-Up

In this section the whole experimental environment is shown briefly emphasizing the simulator. In Section 3.1, the data and the monitoring system are explained. In Section 3.2, the migraine prediction modeling approach using GE is sketched out. Finally, the migraine prediction simulator system is drawn in detail in Section 3.3.

3.1 Data

Data are acquired in a real scenario from migraine patients in an ambulatory study using a Wireless Body Sensor Network (WBSN). Four biometric variables (skin temperature, EDA, electrocardiogram to calculate the HR, and SpO2) are measured using non intrusive sensors and data are sent to an Android smartphone via Bluetooth. The smartphone transmits the data to a cloud storage system. An application in the smartphone allows patients indicate the relative changes of an on-going migraine. These subjective pain marks draw a curve which is normalized (0 to 100%) and modeled as two semi-Gaussian curves from the maximum (5).

Data are processed offline to repair disruption using a Gaussian Process Machine Learning approach (GPML) (17; 18)—for the purpose of this paper, the preprocessing modules (data conversion and GPML) are not used in real time.

Data from two female patients (Patient A and B) have been chosen from two ideally 24 hours studies. Patient A is a young patient suffering from migraines with aura and no medical treatment, and Patient B is a middle aged patient suffering from migraines without aura and using preventive treatment. The training dataset, M, is composed of 15 randomly selected migraines from Patient A and 8 randomly selected migraines from Patient B.

3.2 Migraine prediction modeling using GE

The explanation of the prediction models obtained using GE is out of the range of this paper. All the details are explained in (6). Thus, in this section we will only give insights of the GE migraine prediction modeling process.

GE (19) is a grammar-based form of Genetic Programming (GP) (20), used to generate programs in any language, where a Genetic Algorithm (GA) selects a group of production rules expressed in a Backus Naur Form (BNF) grammar. The GE leads to automatically select optimal mathematical expressions dependent on some or all the input variables u_x.

In this work we have used the HERO Java library published in (21) under GPL license. This work performs a mono-objective study, with the Normalized Root Mean Square Error (NRMSE), or fit, as objective function to minimize (Eq. 3). 150000 generations of 250 individuals each have been used to train the GE models. The length of the chromosomes is 100, and the probability of crossover and mutation are 0.9 and 0.083 respectively. For further details see (6).

$$fit = 100 \times \left(1 - \frac{||y - y_p||}{||y - mean(y)||}\right) \tag{3}$$

The four aforementioned hemodynamic inputs: i) skin temperature ($u_1[k]$), ii) electrodermal activity ($u_2[k]$), iii) heart rate ($u_3[k]$) and iv) oxygen saturation ($u_4[k]$) are synchronized with a period of time of 1 minute; these data fed the GE algorithms. The output of the GE $y_p[k]$ tries to fit the Gaussian model of the real subjective pain. Each migraine in the training set was used to train models for 10 and 20 minutes of prediction horizon. The three models with a fit (Eq. 3) higher than 70% were selected to compute an average prediction (6). The results for both of the patients and 20 minutes of prediction horizon are shown in Equations 4 through 9.

$$y_p[k+1] = y_r[k-20] + e^{\left(log\left(49*10^{-5}+u_2[k-108]\right)-y_r[k-113]+sin(u_2[k-114])\right)} \tag{4}$$

$$y_p[k+1] = y_r[k-20] + \frac{e^{\frac{u_3[k-81]}{u_2[k-70]}}}{u_3[k-20]} \tag{5}$$

$$y_p[k+1] = y_r[k-20] + \left(65*10^{-4} - u_3[k-20]\right) * \Delta\left(u_2[k+\tau]\Big|_{-100}^{-20}\right) \tag{6}$$

$$y_p[k+1] = \frac{y_r[k-20] * u_4[k-134]}{\frac{1}{3} * u_4[k-134] + y_r[k-46] + \Delta\left(u_4[k+\tau]\Big|_{-37}^{-20}\right) + cos\left(\Delta\left(u_3[k+\tau]\Big|_{-140}^{-20}\right)\right)} \tag{7}$$

$$y_p[k+1] = y_r[k-20] + y_r[k-29] - y_r[k-45] + log(u_3[k-53]) \tag{8}$$

$$y_p[k+1] = y_r[k-20] * log\left(log\left(max\left(u_2[k+\tau]\Big|_{-136}^{-20}\right)\right)\right) \tag{9}$$

With the expressions Equations 4 through 6, only EDA (u_2) and HR (u_3) are needed to predict her migraines for Patient A and 20 minutes of prediction horizon. For Patient B, EDA, HR, and SpO2 (u_4) are needed to predict migraines with 20 minutes in advance (Equations 7 through 9).

3.3 The migraine prediction simulator system

In this section, a top-down view of the migraine prediction simulator system is drawn. Figure 1a represents the top view of the migraine predictor simulator system. The blocks are described in detail in our previous works (5; 6), but they are not required for a comprehensive understanding of our current research.

There are four coupled models (shadowed boxes in Figure 1a): `EFsys`, `EFgt`, `Graphs` and `Model`, and seven top-level atomic models: `TEMP`, `EDA`, `HR`, `SpO2`, `Sync`, `Decider` and `Actuator`.

- The coupled model `RootCoupled` is the simulator frame in Java code, and interconnects the models that simulate the hardware modules, with the prediction system model in `EFsys`. The `RootCoupled` is the Figure 1a itself.
- `EFsys` performs the processing from data of the single-output atomic models (`TEMP`, `EDA`, `HR` and `SpO2`) and gives an output to the single-input atomic model (`Actuator`).
- The `EFgt` model contains two atomic models to control the data flow through model `G`, and to show simulation statistics through model `T`. If the simulator runs in simulated time, `T` activates the *stop* signal to finish the simulation after the simulation's observation time has elapsed.
- The `Graphs` coupled model has been included to improve the user experience. This model is only suitable for software simulation and plots the input data, the migraine prediction, and the alarm event if it occurs.
- The atomic models: `TEMP`, `EDA`, `HR` and `SpO2` are, for this paper, the previously preprocessed biometric variables mentioned in Section 3.1. For the next versions of the simulator, these models will provide the raw data from sensors, and real time processing models will be included in the simulator after them. Atomic models for variables remain outside of the `EFsys` coupled model so they can be substituted easily for hardware devices that allow the execution of HIL experiments.
- The `Sync` atomic model synchronizes and buffers the data to simultaneously supply the values of the four biometric variables to the coupled model `EFgt`.
- The `Decider` is, for the aim of this paper, an atomic model that distinguish from the prediction if a migraine event occurs or not. In this case, the `Decider` is a trigger activated by a threshold of one single level. The threshold level is 32 in the normalized objective symptomatic pain curve and this represents 50% probability of the maximum pain level (5).
- The `Actuator` is an atomic model ready to be substituted by a hardware device. In a software simulation, this is a dummy model and it does not perform any action.
- The `Model` coupled model executes most of the computation of this first version of the simulator. The atomic models that the `Model` aggregates are shown in Figure 1d. This model implements the idea of an average prediction model

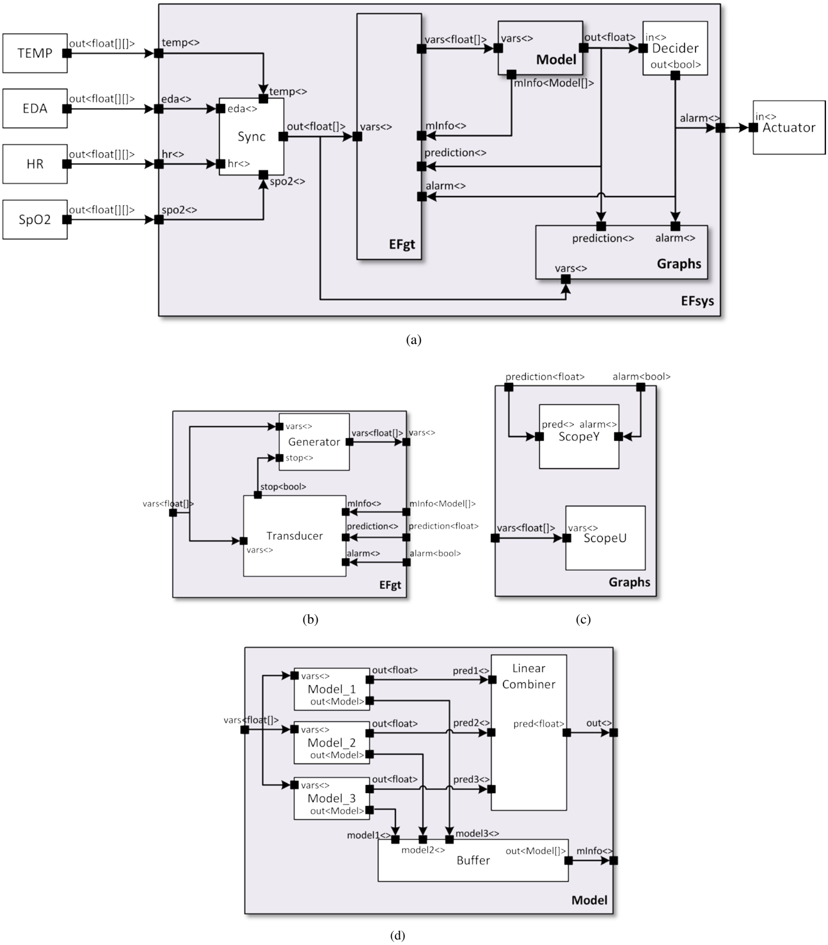

Fig. 1. `RootCoupled` model and detail of the coupled models. (**a**) The `RootCoupled` is the simulator framework containing all the coupled and atomic models in the system; (**b**) `EFgt` coupled model to control the data flow and draw simulation statistics; (**c**) `Graphs` coupled model to improve the user experience; (**d**) `Model` coupled model that aggregates the prediction methodology.

shown in (5) and (6). Three models (`Model_1`, `Model_2` and `Model_3`) are fed with the four synchronized biometric variables. Each one of the models computes a prediction and a linear combination of the three results is performed in the `Linear Combiner` atomic model. These three atomic models can implement any prediction model; in our research we have developed state-space models and GE models for the migraine prediction. The simulator has been validated using the GE prediction Equations 4 through 9 for prediction horizons of 20 minutes. Other equations are used to validate GE models for prediction horizons of 10 minutes (see Section 4). For the sake of simplicity, the `Linear Combiner` computes a non-weighted average.

4 Evaluation

In this section, our initial approach for a real time migraine prediction alarm system is evaluated. Previously developed GE prediction models have been used to validate the simulator's scheme. These models are used in real time where the off-line reparation of the signal cannot take place. This is the first time in the literature that prediction models are used to predict migraine events in real time.

Data from two migraine sufferers have been used. The GE prediction models were trained to predict the incoming migraine events at 10 and 20 minutes in advance. The prediction models (hosted in the atomic models `Model_1`, `Model_2` and `Model_3` of Figure 1d) are fed with the four hemodynamic variables: skin temperature, EDA, HR and SpO2.

Figures 2a and 2b are two examples of real time migraine predictions for Patient A at 10 and 20 minutes, respectively. Figures 2c and 2d are the corresponding ones for Patient B. In Figures 2b and 2d, Equations 4 through 6 and Equations 7 through 9 have been used for Patient A and Patient B respectively. Currently, using the simple threshold implemented in this version, 7 false positives occur in the examples of Figures 2a and 2b. In the future, we will work on reduce the number of false positive alarms in Figures 2b and 2d using a more intelligent `Decider` model.

5 Discussion

This paper describes an initial approach for the development of a migraine prediction simulator system. The simulator has been developed using the DEVS formalism and its functionality has been validated using Grammatical Evolutionary algorithms. The modular and hierarchical formalism of DEVS allows to easily exchange the GE prediction algorithms for others previously developed in our research, such as state-space algorithms.

The migraine disease is one of the most disabling neurological diseases; the prediction of an incoming symptomatic event allows the patient to take the drug in advance an thus be able to stop the pain. In our research, a non intrusive WBSN is used to monitor hemodynamic variables from migraine patients. Prediction models of the migraine have been developed in our previous work using these data. The experiments needed for developing real time prediction models require a complex and

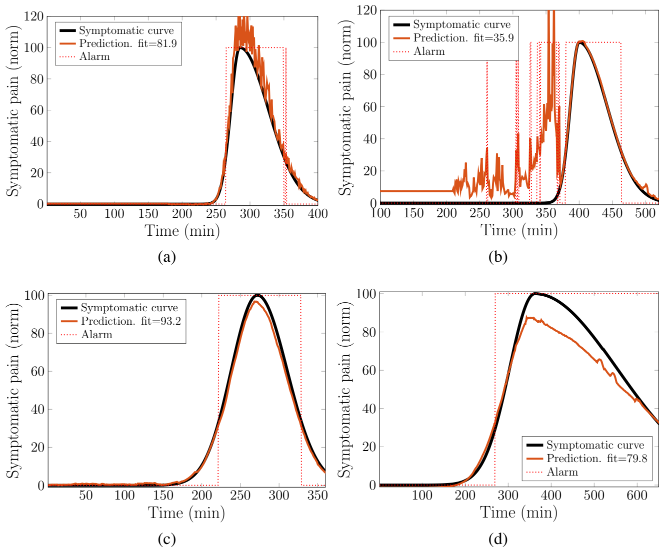

Fig. 2. Results of the validation of the basic simulator using GE algorithms. (**a**) Patient A's migraine, 10 minutes forward; (**b**) Patient A's migraine, 20 minutes forward; (**c**) Patient B's migraine 10 minutes forward; (**d**) Patient B's migraine, 20 minutes forward.

time-consuming monitoring of patients, that cannot be conducted in many cases. Thus, the development of a simulator becomes necessary to test and emulate a real time predictive alarm system for migraine sufferers. The validation of this first approach of such simulator has opened the alternative to apply HIL techniques for achieving HW/SW co-simulation platforms that play an important role on the robustness against sensor failures or disconnections.

Acknowledgements

Research by Josué Pagán has been funded by by the EU (FEDER) and the Spanish Ministry of Economy and Competitiveness under Research Grants TIN 2015-65277-R and TEC2012-33892.

Bibliography

[1] Lipton, R.B., Scher, A.I.: Epidemiology and economic impact of migraine. Current Medical Research and Opinion® **17**(S1) (2001) s4–12

[2] WHO: Headache disorders, fact sheet n°277 (2012)

[3] Stovner, L.J., Andree, C.: Prevalence of headache in europe: a review for the eurolight project. The journal of headache and pain **11**(4) (2010) 289–299

[4] Linde, M., *et al.*: The cost of headache disorders in europe: the eurolight project

[5] Pagán, J., De Orbe, M.I., Gago, A., Sobrado, M., Risco-Martín, J.L., Mora, J.V., Moya, J.M., Ayala, J.L.: Robust and accurate modeling approaches for migraine per-patient prediction from ambulatory data. Sensors **15**(7) (2015) 15419

[6] Pagán, J., , Risco-Martín, J.L., Moya, J.M., Ayala, J.L.: Grammatical evolutionary techniques for prompt migraine prediction. In: Genetic and Evolutionary Computation Conference (GECCO), 2016

[7] M2C: M2C-Machine to Cloud Solutions website. `http://m2c-solutions.es/`

[8] Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. Academic press (2000)

[9] José L. Risco-Martín, Saurabh Mittal, J.C.F.P.M.J.L.A.: Real-time hardware/software co-design using devs-based transparent ms framework. In: SummerSim-SCSC, 2016. (2016)

[10] xDEVS: Java library for devs. `https://github.com/jlrisco/xdevs` (2016)

[11] Wainer, G.A., Madhoun, R., Al-Zoubi, K.: Distributed simulation of devs and cell-devs models in cd++ using web-services. Simulation Modelling Practice and Theory **16**(9) (2008) 1266–1292

[12] Buck, J.T., Ha, S., Lee, E.A., Messerschmitt, D.G.: Ptolemy: A framework for simulating and prototyping heterogeneous systems. (1994)

[13] MATLAB: version 8.5.0.197613 (R2015a). The MathWorks Inc., Natick, Massachusetts (2015)

[14] Barhak, J., Isaman, D.J., Ye, W., Lee, D.: Chronic disease modeling and simulation software. Journal of biomedical informatics **43**(5) (2010) 791–799

[15] Eddy, D.M., Schlessinger, L.: Archimedes a trial-validated model of diabetes. Diabetes care **26**(11) (2003) 3093–3101

[16] Risco-Martín, J.L., Jesús, M., Mittal, S., Zeigler, B.P.: eudevs: Executable uml with devs theory of modeling and simulation. Simulation **85**(11-12) (2009) 750–777

[17] Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)

[18] Rasmussen, C.E., Nickisch, H.: The gaussian processes web site.

[19] Ryan, C., O'Neill, M.: Grammatical evolution: A steady state approach. Late Breaking Papers, Genetic Programming **1998** (1998) 180–185

[20] Poli, R., Langdon, W.B., McPhee, N.F., Koza, J.R.: A field guide to genetic programming. Lulu. com (2008)

[21] HERO: Heuristic optimization library. `https://github.com/jlrisco/hero` (2016)