technische universität dortmund

computer science 12

# Master Thesis

## Dirichlet-Rescale Algorithm in Numerical Simulation on Real-Time Systems

Mario Günzel
Dr.-Ing. Kuan-Hsun Chen
Prof. Dr. Jian-Jia Chen

Otto-Hahn Str. 16
Technische Universität Dortmund
Email: mario.guenzel@tu-dortmund.de
March 19, 2021

In real-time system (RTS) researches, using numerical simulations with synthetic task sets to empirically evaluate the effectiveness of schedulability tests is a well-accepted benchmarking. In the literature, several methods of generating task sets are widely used, e.g., *UUnifast* proposed in 2005, *UUnifast-Discard* proposed in 2009, *RandFixedSum* proposed in 2006, etc. [1] with different pros and cons. Last year in one of the best RTS conferences, a new method called *Dirichlet-Rescale* (DRS) algorithm is proposed by Griffin et al. in [2], which seems to be more efficient and powerful than the aforementioned methods.

Figure 1 illustrates the standard simplex and the targeted area (defined by $x \leq 0.5$, $y \leq 0.45$ and $z \leq 0.7$). The main idea of DRS is to take the multiple constraints abstracting from the targeted task model and efficiently rescale the simplex $S$ to move a value from that simplex to the targeted area, which is more efficient than restarting the procedure as long as there is an unfeasible generated set like UUnifast-Discard.

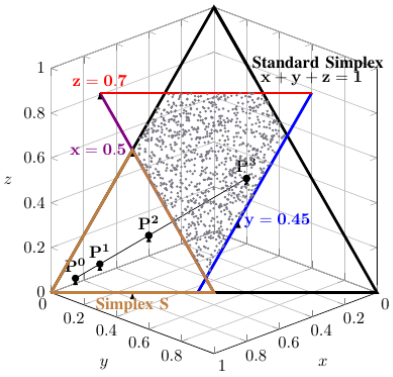


Figure 1: The output of Dirichlet-Rescale over 1000 runs in [2].

The DRS algorithm seems to be the next unicorn for empirical studies on various real-time systems, e.g., mixed criticality systems, multi-core systems, typical and worst-case execution times, and resource locking protocols, if necessary enhancement or integration can be provided. For example, self-suspending task models, at the moment have no "de facto" synthetic benchmarks to evaluate scheduling algorithms and schedulbility tests, which is of key interest to DAES group.

In this master thesis, we would like to figure out the real power of this DRS algorithm. For deciding the workload parameters for self-suspending task models, we foresee that the DRS algorithm might provide additional insights to guide the synthesis of tasksets and create some corner task sets systematically but efficiently. The expected outcome could be very effective for the upcoming researches of real-time systems, and the experience of adapting the DRS algorithm can also be generalized on various real-time system researches as a series of researches.

**In this thesis**,[1] students first should get familiar with the Dirichlet-Rescale (DRS) algorithm and explore its potential to be used as benchmark for self-suspending task sets. Afterwards, it should be implemented and released as part of the evaluation framework [3] of our group with a clear coding convention. Using this framework, the student should compare the new benchmark and previous benchmarks by evaluating already implemented schedulability tests and scheduling algorithms.

### Required Skills:

- Good Knowledge of real-time systems
- Basic knowledge of numerical simulations
- Comfortable in Python programming

### Acquired Skills after the thesis:

- Knowledge about self-suspending domains
- Deep understanding of workload generators
- Experience of research campaigns

### References:

[1] Paul Emberson, Roger Stafford, and Robert I. Davis, Techniques for the synthesis of multiprocessor tasksets, 2010, Preprint version.

[2] David Griffin, Iain Bate, Robert I. Davis. Generating Utilization Vectors for the Systematic Evaluation of Schedulability Tests, Preprint version.

[3] LS12 DEAS Group. Evaluation Framework for Self-Suspending Task Systems. Github Repository.

[1] Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.