The Austrian Society of Operations Research (ÖGOR) organizes a workshop for PhD students:

Date: **1.8.2022 – 5.8.2022**

at the University for Continuing Education („Donau Uni“), Krems, Lower Austria.

The lecturers are Paolo Brandimarte (Politecnico di Torino) und Antonio Frangioni (Università di Pisa).

Participation is free for ÖGOR members, the participants (or their institutions) have to finance their expenditures for travelling, food and overnight stays.

For application please mail to **raimund.kovacevic@donau-uni.ac.at**, latest May 28.

PROGRAM:

1.8. – 2.8. Antonio Frangioni

Decomposition methods for large-scale programming: (old) theory and some (new) tools

Many large-scale practical problems have "block structure" in the general sense that "most decisions are closely related to a few others but loosely related with the rest". This usually corresponds to the fact that the mathematical model represents a system made of several "components" that would be isolated were it not for relatively few elements that link each with others. In fact, models of complex systems typically have more than one of these structures, possibly nested, corresponding to different geographical and/or temporal scales that the model covers or to different scenarios for the uncertain data. Each one of these "block structures" can be (separately) algorithmically exploited with the two well-known, half-a-century-old decomposition approaches: the dual/Lagrangian/Dantzig-Wolfe one, and the primal/resource/Benders' one. The lecture starts by reviewing their basic ideas, showing that, despite some relevant differences, both boil down to solving a large-scale, nondifferentiable optimization problem whose objective function is itself one or more (possibly, costly) optimization problems. This has (again, 50+ years old) solution algorithms, whose by-the-book implementations has the advantage of being relatively simple but that unfortunately are often not efficient enough in practice. The lecture then reviews a number of ideas (some oldish and some newish) to speed-up decomposition approaches, such as stabilization, dual-optimal cuts, (partial) disaggregation, easy components, structured decomposition, and approximate subproblem solution. However, applying them in practice is challenging due to a limited support from standard modelling and solution tools. This is especially the case if one wants to exploit multiple heterogeneous nested forms of structure, such as those present in many huge-scale models, which is perhaps the only practical route for their solution exploiting the inherent parallelisability of decomposition approaches and HPC architectures. The final part of the lecture is therefore dedicated to the presentation of a new modelling framework that has been recently released, with the hands-on descriptions of some applications showcasing the use cases for which it may hopefully prove a useful tool.

3.8. short presentation of participant's work and social event

4.8. – 5.8. Paolo Brandimarte

An introduction to dynamic programming: from shortest paths to reinforcement learning

The principle of dynamic programming (DP) is the foundation of several solution approaches, exact or approximate, for quite challenging sequential decision problems under uncertainty. Essentially, it may be regarded as a way to decompose a problem with respect to time, by trading off short and long-term rewards. As such, DP is also the cornerstone of quite popular reinforcement learning methods. DP is quite general, as it may be applied to a wide array of problem settings: models may be deterministic and stochastic, discrete and continuous-time, finite and infinite-horizon. This generality comes with a price, as DP suffers from multiple curses, and it is a principle requiring adaptation to each specific case, rather than an off-the-shelves algorithm. Hence, it is important to build strong foundations and a firm understanding of what is needed for its successful application. In this tutorial we will cover the following:

• DP principle and the different shapes that the Bellman equation may take, depending on the information structure.
• Modeling in DP: examples from various business settings (inventory control, revenue management, financial engineering).
• Model-based vs model-free approaches. • Classical numerical methods. • Reinforcement learning and approximate DP; the role of Monte Carlo sampling.
• Application to continuous time problems
• A comparison with alternative approaches: multistage stochastic programming with recourse, adjustable robust optimization, decision rules, simulation-based optimization.

A large part of the tutorial, but not all, relies on my book (From Shortest Paths to Reinforcement Learning: A MATLAB-Based Introduction to Dynamic Programming; Springer, 2021). MATLAB will be used for illustration purposes, but programming skills in any language are sufficient.