Discipline: Artificial Intelligence, Data Science, Optimization / Operations, Finance, Management, Marketing, and Information Systems

1. Language

English

2. Title

Systematic Design and Application of Metaheuristics

3. Lecturer

Franz Rothlauf (Johannes Gutenberg-Universität Mainz, Information Systems and Business Administration)
http://wi.bwl.uni-mainz.de/rothlauf.html.en

4. Date and Location

10. – 13. October 2022
Johannes Gutenberg-University Mainz, Jakob-Welder-Weg 9, 55128 Mainz

5. Course Description

5.1 Abstract and Learning Objectives

Besides learning how to apply metaheuristics, the interactive 4-day course deals with the following questions:

1. How to systematically choose among different metaheuristics based on the properties of the problem that should be solved?
2. How to systematically design efficient and high-performing metaheuristics?
3. How to consider problem-specific knowledge for the design of metaheuristics?

5.2 Content

Metaheuristics like evolutionary algorithms, genetic programming, variable neighborhood search, tabu search, simulated annealing, and others are applied to large, difficult, or realistic optimization problems, for which efficient classical optimization methods are not available or applicable. This includes the optimization of other AI systems like neural networks. Unfortunately, many text books teach such methods by providing detailed descriptions of the functionality of single examples of metaheuristics neglecting the underlying and common concepts. As a result, the systematic design and application of metaheuristics is often not a systematic engineering task but a result of repeated trial and error. Applicants apply textbook approaches and are surprised that they do not perform well when used for problems of realistic size or complexity.

This course at hand takes a different approach. It teaches the basic, method-independent principles and design guidelines of metaheuristics and how they can be used to systematically develop superior heuristic optimization methods for problems of choice. Consequently, the course focuses on the application side and answers three fundamental questions:

1. It tells the participants on which problems metaheuristics are expected to perform well, and what are problems where other optimization paradigms are a better choice.
2. Participants learn to systematically design an appropriate metaheuristic for a particular problem using a coherent view on design elements and working principles of metaheuristics.
3. Participants learn how to make use of problem-specific knowledge for the design of efficient and effective metaheuristics that solve not only small toy problems but also perform well on large and real-world problems.

The course consists of two parts: the first part interactively teaches basic underlying concepts; the second discusses application examples or application problems provided by the participants. Thus, participants are encouraged to bring in their own application problems. It is not expected that all participants bring in own examples. If the participants do not have any problems they want to solve, representative example problems are provided by the lecturer.

The course is aimed at PhD-students from various disciplines, including operations, production and logistics, supply chain management, finance, marketing and management science. Of course, the course fits well for PhD-students working in the field of artificial intelligence and data science. The elements of the course range from lectures on theory, to the discussion of best practices and case studies, and finally to the presentation of work done by the students.

5.3 Schedule (including start and end time)

<table>
<tr><th></th><th>Day 1</th><th>Day 2</th><th>Day 3</th><th>Day 4</th></tr>
<tr><td>morning (9am-12am)</td><td>Optimization Problems
• Search spaces
• Problem difficulty
• Locality
• Decomposability</td><td>Recombination Strategies
• Genetic Algorithms
• Estimation of Distribution Algorithms</td><td colspan="2" rowspan="2">Workshop which includes
• Presentation of Best Practices
• Presentations of Students
• In-depth Discussion of Cases</td></tr>
<tr><td>afternoon (1pm-6pm)</td><td>Design Elements
• Representation & Operators
• Fitness function
Local Search Strategies
• Variable Neighborhood Search
• Evolution Strategies</td><td>Design Principles
• Locality
• Bias</td></tr>
</table>

5.4 Course format

Lecture/Seminar

Day 1 presents introductory lectures on heuristic optimization, problem structure, design elements for heuristic approaches, and studies the basic principles behind standard local search strategies like variable neighborhood search, evolution strategies, and others.

On day 2, we cover different methods (Genetic Algorithms, Estimation of Distribution Algorithms, Genetic Programming, and others) and discuss general design principles of modern heuristics like locality and biasing the search.

On the last two days (3 and 4) we put the learned theory into real live and either study representative application examples or optimization problems and solution approaches (including non-heuristic ones) provided by the participants. The goal is to get a deeper understanding on why particular approach succeed (or fail) for particular problems, to develop a better understanding on the design of the discussed methods, and to improve the presented case examples.

6. Preparation and Literature

6.1 Prerequisites

The course requires basic skills in mathematics, statistics, and classical optimization methods.

Essential Reading Material

Participants are entitled to read some literature as part of their preparation for the course. Some of the optional papers may be used in the class as representative examples of high-quality work in heuristic optimization. The exact choice of the papers depends on the research interests of the participating students. It is the goal to choose such work that fits well to the students' interests.

Belloni, A.; Freund, R.; Selove, M.; Simester, D. (2008): Optimizing Product Line Designs: Efficient Methods and Comparisons. In: Management Science 54 (9), S. 1544–1552.

Blum, C.; Roli, A. (2003): Metaheuristics in combinatorial optimization. In: ACM Computing Surveys 35 (3), S. 268–308.

Bruderer, E.; Singh, J. V. (1996): Organizational evolution, learning, and selection: a genetic-algorithm-based model. In: Academy of Management Journal 39 (5), S. 1322–1349.

Michalewicz Z, Fogel DB (2000) How to Solve It: Modern Heuristics. Springer, Berlin. Chapters 1,2,7,9.

Rothlauf, F. (2011): Design of modern heuristics. Principles and application. Springer, Berlin. Chapters 4,5,6.

Salimans T., Ho J., Chen X., Sidor S., Sutskever I. (2017): Evolution Strategies as a Scalable Alternative to Reinforcement Learning. https://arxiv.org/abs/1703.03864

Whitley, D. (2001): An overview of evolutionary algorithms: practical issues and common pitfalls. In: Information and Software Technology 43 (14), S. 817–831.

6.2 Additional Reading Material / zusätzliche Lektüre

Allen, F.; Karjalainen, R. (1999): Using genetic algorithms to find technical trading rules. In: *Journal of Financial Economics* 51 (2), S. 245–271.

Balakrishnan, P. V.; Jacob, V. S. (1996): Genetic Algorithms for Product Design. In: *Management Science* 42 (8), S. 1105–1117.

Bartz-Beielstein T. (2006): Experimental Research in Evolutionary Computation. The New Experimentalism. Springer, Heidelberg. Chapters 3,7,8.

Debels, D.; Vanhoucke, M. (2007): A Decomposition-Based Genetic Algorithm for the Resource-Constrained Project-Scheduling Problem. In: *Operations Research* 55 (3), S. 457–469.

Midgley, D. F.; Marks, R. E.; Cooper, L. C. (1997): Breeding Competitive Strategies. In: *Management Science* 43 (3), S. 257–275.

Noe, T. H.; Rebello, M. J.; Wang, J. U. N. (2006): The Evolution of Security Designs. In: *Journal of Finance* 61 (5), S. 2103–2135.

Reeves CR, Rowe JE (2003) Genetic Algorithms: Principles and Perspectives. Kluwer, Dordrecht. Chapters 2,4,5,6,8.

Ropke, S.; Pisinger, D. (2006): An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. In: *Transportation Science* 40 (4), S. 455–472.

Schittekat, P.; Sorensen, K. (2009): OR Practice--Supporting 3PL Decisions in the Automotive Industry by Generating Diverse Solutions to a Large-Scale Location-Routing Problem. In: *Operations Research* 57 (5), S. 1058–1067.

Schmidt, M.; Lipson, H. (2009): Distilling Free-Form Natural Laws from Experimental Data. In: *Science* 324 (5923), S. 81–85.

Sung-Soon Choi; Byung-Ro Moon (2008): Normalization for Genetic Algorithms With Non-synonymously Redundant Encodings. In: *IEEE Transactions on Evolutionary Computation* 12 (5), S. 604–616.

6.3 To prepare

Students are entitled to read relevant work of the mandatory literature.

If students want to obtain a graded certificate (in German: benoteter Schein), they are required to prepare a brief presentations on either 1) a particular optimization problem they are dealing with, 2) a description of a possible (heuristic or non-heuristic) solution approach, 3) or both.

If students do only need a certificate of attendance (in German: Teilnahmebescheinigung) the preparation of the presentation is optional.

7. Administration

7.1 Max. number of participants

Twenty

7.2 Assignments

Preparation of one presentation.

7.3 Exam

The final grade will be based on class participation during the lectures and case studies (50%) and the quality of the proposed heuristic approach and the students' performance in the workshop (50%).

7.4 Credits

ECTS

8. Arbeitszeitaufwand / Working Hours

Aufteilung der Arbeitsstunden / Working Hours	**Stunden**
Mandatory Reading	40h
Preparation of Presentation	100h
Course work	40h
SUMME	**180 h**
ECTS: 6	