Machine Learning for Optimization

WS 2025: Project/Assignment 1

Nysret Musliu
Database and Artificial Intelligence Group
Institute of Logic and Computation, TU-Wien

Assignment

Select one of problems below:

- **The Social Golfer Problem or**
- **Rotating Workforce Scheduling**

Task1:

Implement two methods to solve the selected problem

- A complete approach (e.g., CP, SAT, MIP, ASP, …)
- A (meta)heuristic algorithm

Task2:

Implement a framework for automated algorithm selection that leverages machine learning techniques.

2 students can work in one group

Task1.1: Complete/Exact method

- You are encouraged to use one of these frameworks/complete solvers:
  - MiniZinc constraint modeling language
    http://www.minizinc.org/
    - See also Coursera online course: https://www.coursera.org/learn/basic-modeling
    - Various solvers can be used from MiniZinc including MIP solver Gurobi
  - OR-Tools (Solver of Google) https://developers.google.com/optimization
  - CP-Optimizer (Solver of IBM, academic license required)
    https://www.ibm.com/analytics/cplex-cp-optimizer
  - Any language supported by OR-Tools and CP-Optimizer can be used
- The use of other systems (SAT, MIP solvers, …) is also possible

Task 1.2 Heuristic technique

You can implement one of the following techniques:

- Local search (e.g., simulated annealing, tabu search…)
- Large neighborhood search
- Customized heuristic techniques.

You can use any programming language, but you should implement the techniques yourself

Task 2: Automated algorithm selection

Implement a framework for automated algorithm selection that leverages machine learning techniques

1. Implement at least five features that effectively characterize the problem instances. You may use different types of features such as direct features, model-based features, probing features, etc. (see the papers below)
2. Run your algorithms on at least 100 instances.
3. Construct a classification dataset, where each entry represents one problem instance. The features are those implemented in step 1, and the class represents the best algorithm for that instance (e.g., Complete or Heuristic)
4. Train a machine learning (ML) algorithm using 65% of the instances from your dataset
5. Present the results of your ML algorithm (classification accuracy, confusion matrix)
6. Compare the methods: Complete, Heuristic, and AlgorithmSelector across all instances

Literature:

- Lecture Slides (Rice's framework…)
- Leyton-Brown, K., Nudelman, E. and Shoham, Y., 2009. Empirical hardness models: Methodology and a case study on combinatorial auctions. Journal of the ACM (JACM), 56(4), pp.1-52. https://dl.acm.org/doi/10.1145/1538902.1538906
- Musliu, N., Schwengerer, M. (2013). Algorithm Selection for the Graph Coloring Problem. In: Nicosia, G., Pardalos, P. (eds) Learning and Intelligent Optimization. LION 2013. https://doi.org/10.1007/978-3-642-44973-4_42
- Kletzander, L., Musliu, N. & Smith-Miles, K. Instance space analysis for a personnel scheduling problem. Ann Math Artif Intell 89, 617–637 (2021). https://doi.org/10.1007/s10472-020-09695-2
- De Coster, A., Musliu, N., Schaerf, A. et al. Algorithm selection and instance space analysis for curriculum-based course timetabling. J Sched 25, 35–58 (2022). https://doi.org/10.1007/s10951-021-00701-x
- M. Abseher, N. Musliu, S. Woltran. Improving the Efficiency of Dynamic Programming on Tree Decompositions via Machine Learning- DOI: https://doi.org/10.1613/jair.5312

Schedule

- Submission deadline:
  - 16.12.2025, discussions: 17.12 - 19.12.

  OR
  - 23.12.2025, discussions: 08.01. – 12.01.
- You should submit in TUWEL a zip file that includes:
  - Your source code
  - Slides or a PDF report describing the work in your project (including a short description of your methods, instances used, features implemented, results, and comparison)

Discussion

Each member of the group should be able to:

- Answer questions about the code and the work done in the project
- Answer questions about the applied methods and be aware of other methods that exist for combinatorial optimization problems (covered in the first and second lectures)
- Answer questions about automated algorithm selection (covered in the third lecture and papers given in slide 5)

Grading:

- Grading is based on the submission and the discussion
- The discussion counts as Exam 1 in this class, and you can earn 30% of the total points in this discussion/exam
- There will be two additional discussions
- Project 2 will be available until 23.12., and the discussion will take place at the end of January
- Project 3 will be available at the end of January, and the discussion will take place at the end of February/beginning of March

Problems

Select one of problems below:

- **The Social Golfer Problem or**
- **Rotating Workforce Scheduling**

The Social Golfer Problem

Problem specification is given in:

http://www.csplib.org/prob/prob010/index.html

> "The coordinator of a local golf club has come to you with the following problem. In her club, there are 32 social golfers, each of whom play golf once a week, and always in groups of 4. She would like you to come up with a schedule of play for these golfers, to last as many weeks as possible, such that no golfer plays in the same group as any other golfer on more than one occasion. "

General Problem:

> Schedule m groups of n golfers over p weeks, such that no two golfers play in the same group more than once

Social Golfer Problem

	Week 1				Week 2				Week 3				Week 4				Week 5			
Group 1	25	2	27	4	26	30	2	6	3	5	2	8	23	31	2	10	29	11	2	24
Group 2	29	6	31	8	28	32	4	8	1	7	4	6	21	29	4	12	31	9	4	22
Group 3	17	10	19	12	18	22	10	14	11	13	10	16	19	27	6	14	25	15	6	20
Group 4	21	14	23	16	20	24	12	16	9	15	12	14	17	25	8	16	27	13	8	18
Group 5	9	18	11	20	17	21	9	13	19	21	18	24	7	15	18	26	17	7	14	28
Group 6	13	22	15	24	19	23	11	15	17	23	20	22	5	13	20	28	19	5	16	26
Group 7	1	26	3	28	25	29	1	5	27	29	26	32	3	11	22	30	21	3	10	32
Group 8	5	30	7	32	27	31	3	7	25	31	28	30	1	9	24	32	23	1	12	30

Group 1	7	13	2	12	32	20	2	14	15	31	2	17	9	19	2	28	1	21	2	22
Group 2	5	15	4	10	30	18	4	16	3	19	4	20	11	17	4	26	3	23	4	24
Group 3	3	9	6	16	5	11	6	12	5	21	6	22	13	23	6	32	5	17	6	18
Group 4	1	11	8	14	7	9	8	10	7	23	8	24	15	21	8	30	7	19	8	20
Group 5	23	29	18	28	19	31	13	1	9	25	10	26	1	27	10	20	9	29	10	30
Group 6	21	31	20	26	15	29	17	3	11	27	12	28	3	25	12	18	11	31	12	32
Group 7	19	25	22	32	21	27	22	28	13	29	14	30	5	31	14	24	13	25	14	26
Group 8	17	27	24	30	23	25	24	26	16	1	32	18	7	29	16	22	15	27	16	28
	Week 6				Week 7				Week 8				Week 9				Week 10			

Instances

- Use the original problem 8-4-x (x=6…10) and other instances that you generate (e.g. 5-3-5, 5-3-6…)
- You should generate at least 100 instances

References

- Markus Triska. Solution Methods for the Social Golfer Problem, Master Thesis, Vienna University of Technology, 2008. http://www.logic.at/prolog/mst.pdf OR
  - Markus Triska, Nysret Musliu: An effective greedy heuristic for the Social Golfer Problem. Ann. Oper. Res. 194(1): 413-425 (2012) https://link.springer.com/article/10.1007/s10479-011-0866-7
  - Markus Triska, Nysret Musliu: An improved SAT formulation for the social golfer problem. Ann. Oper. Res. 194(1): 427-438 (2012) https://link.springer.com/article/10.1007/s10479-010-0702-5
- http://www.cs.brown.edu/~sello/golf.html
- C. Cotta, I. Dotú, A. J. Fernández, and P. V. Hentenryck (2006). Scheduling social golfers with memetic
- evolutionary programming. In Hybrid Metaheuristics, volume 4030 of LNCS, pages 150–161.http://www.springerlink.com/content/u4x98578w5669716/
- I. Dotú and P. V. Hentenryck (2005). Scheduling social golfers locally. In CPAIOR, volume 3524 of
- LNCS, pages 155–167. http://www.springerlink.com/content/nxc82hrhrtpajfl9/
- Harvey, W. and Winterer, T. (2005). Solving the MOLR and social golfers problems. In CP’05, volume 3709 Of LNCS, pages 286–300. http://www.springerlink.com/content/u613864820134778/

Rotating Workforce Scheduling

The schedule is cyclic

	Mo	Tu	We	Th	Fr	Sa	Su
A	D	D	D			D	D
B	D	D	D	D			
C	A	A	A	A			A
D	A	A	A	A	A		
E	D	D	A	A	A		
F	A	A	N	N	N		
G	N	N	N	N	N		
H		N	N	N	N	N	
I			D	D	D	A	A
J				D	D	N	N
K	N				A	A	N
L	N	N			D	D	D

Number of employees

Employees working shifts:

D: Day shift ; A: Afternoon shift ,

N: Night shift; Day off

Constraints

	Mo	Tu	We	Th	Fr	Sa	Su
A	D	D	D			D	D
B	D	D	D	D			
C	A	A	A	A			A
D	A	A	A	A	A		
E	D	D	A	A	A		
F	A	A	N	N	N		
G	N	N	N	N	N		
H		N	N	N	N	N	
I			D	D	D	A	A
J				D	D	N	N
K	N				A	A	N
L	N	N			D	D	D

Not allowed sequences of shifts:

N – D
N D
A D
N A
N – A
A – D

Maximum and minimum length of periods of successive shifts.

e.g.: N: 2-5, D: 2-6

Temporal requirements: required number of employees in shift *i* during day *j*

Monday (Mo): D: 3, N: 3, A: 3

Maximum and minimum length of work days and days-off blocks

e.g.: days-off block: 2-4

work block: 2-6

Objective

Find a cyclic schedule (assignment of shifts to employees) that satisfies the temporal requirement, and all other constraints

Instances/References

Instances:

- https://www.dbai.tuwien.ac.at/staff/musliu/benchmarks/

References:

- Musliu, N., Schutt, A., Stuckey, P.J. (2018). Solver Independent Rotating Workforce Scheduling. CPAIOR 2018. https://doi.org/10.1007/978-3-319-93031-2_31
- N. Musliu. Heuristic Methods for Automatic Rotating Workforce Scheduling. https://www.dbai.tuwien.ac.at/staff/musliu/rotHeuristic.pdf
- Kletzander, L., Musliu, N. & Smith-Miles, K. Instance space analysis for a personnel scheduling problem. Ann Math Artif Intell 89, 617–637 (2021). https://doi.org/10.1007/s10472-020-09695-2