AI and Optimization for Sustainable Applications

Nysret Musliu
Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling
Institute of Logic and Computation, DBAI
Faculty of Informatics, TU Wien
Outline

- Sustainable applications
  - Employee planning and scheduling
  - Production planning and scheduling
  - Food waste reduction

- AI and optimization techniques
  - Modeling
  - Exact, heuristic and hybrid techniques
  - Machine learning
    - Supervised learning
    - Reinforcement learning

- Conclusions
Investigated Applications in our Lab

- Rotating Workforce Scheduling
- Shift Design
- Break Scheduling
- Nurse Rostering
- Torpedo Scheduling
- Electric Vehicle Charging
- Tourist Trip Planning
- Social Golfer Problem
- High School Timetabling
- Production Leveling Problem
- Parallel Machine Scheduling
- Industrial Oven Scheduling
- Physician Scheduling During a Pandemic
- Unicost Set Covering
- (Hyper)tree Decomposition
- Graph Coloring
- Traveling Salesman Problem
- Vehicle Routing
- Sudoku
- Bus Driver Scheduling
- Test Laboratory Scheduling
- Artificial Teeth Production Scheduling
- Project Scheduling
- Paint Shop Scheduling Problem
- Curriculum-based Course Timetabling
- Food Waste Reduction
Employee Scheduling

- Work schedules influence the lives of employees

- Unsuitable timetable can have a tremendous negative impact on one's health, social life, and motivation at work

- Organizations in the commercial and public sector must meet their workforce requirements and ensure the quality of their services and operations
Employee Scheduling

Real world employee scheduling problems appear in many companies

- Airports
- Call centers
- Air traffic control
- Hospitals
- Public transport
- Production plants
- ...
Employee Scheduling Problems

Phase 1:
Workforce requirements
Employee Scheduling Problems

Phase 1: Workforce requirements

Phase 2: Shift Design/Break Scheduling
**Employee Scheduling Problems**

**Phase 1:**
Workforce requirements

**Phase 2:**
Shift Design/Break Scheduling

**Phase 3:**
Assignment of shifts

Selected papers: [3,4,11,12]
Example: Rotating Workforce Scheduling

Length of schedule: If the schedule is cyclic the total length of a planning period will be: \( \text{NumberOfEmployees} \times 7 \)

<table>
<thead>
<tr>
<th></th>
<th>Mo</th>
<th>Tu</th>
<th>We</th>
<th>Th</th>
<th>Fr</th>
<th>Sa</th>
<th>Su</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>E</td>
<td>D</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>A</td>
<td>A</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>G</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>H</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>I</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>J</td>
<td>D</td>
<td>D</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>K</td>
<td>Z</td>
<td>Z</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>L</td>
<td>Z</td>
<td>Z</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>

Number of employees

Employees working shifts:
D: Day shift;  A: Afternoon shift,  N: Night shift;  Day off
## Constraints

### Temporal requirements:

Required number of employees in shift $i$ during day $j$.

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

### Not allowed sequences of shifts:

Maximum and minimum length of periods of successive shifts.

E.g.: $N: 2-5$, $D: 2-6$

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

Possible soft constraints:
- Optimization of free weekends (weekends off)
- …
In these applications it is important to
- Reduce resource consumption, including energy
- Increase production efficiency
Test Laboratory Scheduling

Selected papers: [1,5]
Industrial Oven Scheduling

**Task:** Jobs need to be scheduled and batched efficiently for processing in ovens

**Challenge:** Many constraints and solution objectives need to be considered

Selected papers: [8]
Other real-world problems...

Torpedo Scheduling, ACP Challenge, 2016

Parallel Machine Scheduling

Paint Shop Scheduling

Selected papers: [6, 7, 9, 10]
The General Obstacle

- NP-hard (intractable) problems
- No efficient algorithms could be found yet
- P problems can be solved efficiently (in polynomial time)
- P \neq NP \,? \, (Millennium Prize Problem)

Tremendous size of the search space of possible solutions

Example: 12 employees, 1 week, 4 shifts

\[ 4^{84} \]
Food Waste Reduction

- APPETITE project aims to reduce food waste in retail environment
- Motivated by current waste of food
- Utilize integration of
  - Artificial Intelligence (AI) based prognosis algorithms
  - Logistical optimizations
- Collaboration of
  - TU Wien, Fraunhofer, WU Wien, IT-PS, Invenium, Kastner, Metro, Spar
Goals

- Reduce food waste through
  - Utilization of advanced forecasting methods (using AI)
  - Integration of forecasts into ordering process

- Advanced forecasting using **Supervised Learning**
- Use provided sales and demand data for forecasting
  - Utilize weather and movement data to improve forecasts
- Empirically evaluate forecasting
  - Depending on algorithms chosen
  - Depending on input dataset(s) provided
Input Data

- Internal data
  - Product data (and “related” products)
  - Product acquisition data
  - Sales data and promotions

- External data
  - Weather data
  - Movement data (from cellular network)

- Contextual data
  - Day of the week, holidays, ...
Current Status of APPETITE Project

- Area analysis for the locations has been finished
- Weather and movement data are being gathered
- Data preprocessing and dataset creation is being finalized
- Initial empirical evaluation of performance

- Next steps: Finalizing platform implementation
AI and Optimization Techniques
Research work in the CD-Lab Artis

- Formal mathematical formulations
- Identification of related problems in the literature
- Complexity analysis
- General variants of problems
- New problem instances provided to the literature

- Modelling techniques
- AI/Optimization solving techniques
- Meta/Hyper-heuristics
- Hybrid algorithms
- Algorithm selection and instance space analysis

Existing problems

New challenging problems provided by the industry
## AI and optimization methods

<table>
<thead>
<tr>
<th>Complete approaches</th>
<th>Heuristic techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematical programming</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>Constraint programming</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>Answer set programming</td>
<td>Evolutionary Strategies</td>
</tr>
<tr>
<td>SAT/SMT</td>
<td>Memetic Algorithms</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Hybrid methods

- Large Neighborhood Search
- Hyper-heuristics
- Machine learning based approaches
- ...
Modeling approaches

- **Mathematical Programming**
  - Solvers: Gurobi, CPLEX, SCIP, ...
- **Constraint Programming**
  - Solvers: OR-Tools, Chuffed, CP Optimizer...
  - The MiniZinc challenge: [https://www.minizinc.org/challenge.html](https://www.minizinc.org/challenge.html)
- **Answer Set Programming**
  - Solvers: Potassco (the Potsdam Answer Set Solving Collection), DLV, ...
- **SAT**
  - Solvers: [http://www.satcompetition.org/](http://www.satcompetition.org/)
- ...
Rotating workforce scheduling: A constraint model

\[ \sum_{k \leq 0}^{u_w} (T_{t(j+k)} = O) > 0, \quad j \in TT \] (1)

\[ \sum_{k \leq 1}^{l_w} (T_{t(j+k)} = O) = 0, \quad j \in TT, T_j = O \land T_{t(j+1)} \neq O \] (2)

\[ \sum_{k \leq 0}^{u_O} (T_{t(j+k)} \neq O) > 0, \quad j \in TT \] (3)

\[ \sum_{k \leq 1}^{l_O} (T_{t(j+k)} \neq O) = 0, \quad j \in TT, T_j \neq O \land T_{t(j+1)} = O \] (4)

\[ \sum_{k \leq 0}^{u_{sh}} (T_{t(j+k)} \neq sh) > 0, \quad j \in TT, sh \in A \] (5)

\[ \sum_{k \leq 1}^{l_{sh}} (T_{t(j+k)} \neq sh) = 0, \quad j \in TT, sh \in A, T_j \neq sh \land T_{t(j+1)} = sh \] (6)

\[ T_j = sh_1 \rightarrow T_{t(j+1)} \neq sh_2, \quad j \in TT, (sh_1, sh_2) \in F_2 \] (7)

\[ T_j = sh_1 \land T_{t(j+1)} = O \rightarrow T_{t(j+2)} \neq sh_2, \quad j \in TT, (sh_1, sh_2) \in F_3 \] (8)

\[ \sum_{i \in 1..n} (S_{i,j} = sh) = R_{sh,j}, \quad j \in 1..w, sh \in A \] (9)

\[ \sum_{i \in 1..n} (S_{i,j} = O) = o_j, \quad j \in 1..w \] (10)

Alternative model: global constraints for (9) and (10)

\[ gcc\_low\_up([S_{i,j} | i \in 1..n], A, [R_{sh,j} | sh \in A], [R_{sh,j} | sh \in A]) \] (11)

\[ gcc\_low\_up([S_{i,j} | i \in 1..n], A^+, [R_{sh,j} | sh \in A^+], [R_{sh,j} | sh \in A^+]) \] (12)

Selected papers:

[11, 13]
Example MIP: Parallel Machine Scheduling

\[ \text{minimise } Lex(\Sigma_{j \in J}(T_j), C_{\text{max}}), \text{ subject to} \]

\[ \Sigma_{m \in M}(Y_{j,m}) = 1, \forall j \in J \]

\[ \Sigma_{i \in J_0, i \neq j}(X_{i,j,m}) = Y_{j,m}, \forall j \in J, m \in M \]

\[ \Sigma_{j \in J_0, i \neq j}(X_{i,j,m}) = Y_{i,m}, \forall i \in J, m \in M \]

\[ C_j \geq C_i + s_{i,j,m} + p_{j,m} + V \cdot (X_{i,j,m} - 1), \forall i \in J_0, j \in J, m \in M \]

\[ \Sigma_{j \in J}(X_{0,j,m}) \leq 1, \forall m \in M \]

\[ \Sigma_{i \in J_0, j \in J, i \neq j}(s_{i,j,m} \cdot X_{i,j,m}) + \Sigma_{i \in J}(p_{i,m} \cdot Y_{i,m} + s_{i,0,m} \cdot X_{i,0,m}) \leq C_{\text{max}}, \forall m \in M \]

\[ T_j \geq C_j - d_j, \forall j \in J \]

\[ T_j \geq 0, \forall j \in J \]

Selected papers: [10]
Local Search Techniques

- Are based on the neighbourhood of the current solution

- The solution is changed iteratively with so called neighbourhood relations (moves) until an acceptable or optimal solution is reached
Local Search Techniques

1. Construct the initial solution $s$
2. Generate neighbourhood $N(s)$ of solution $s$
3. Select from the neighbourhood the descendant of the current solution
4. Go to step 2

Advanced metaheuristic techniques
- Simulated Annealing
- Tabu Search
- Large Neighborhood Search
- ...

Metaheuristics include a mechanism to escape local optima
Neighborhoods

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D</td>
<td>D</td>
<td>A</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>A</td>
<td>N</td>
<td>D</td>
<td>N</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>N</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>N</td>
<td>D</td>
<td>N</td>
<td>D</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>E</td>
<td>D</td>
<td>N</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>D</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>G</td>
<td>D</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>D</td>
<td>N</td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>D</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>I</td>
<td>A</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>J</td>
<td>A</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>A</td>
</tr>
<tr>
<td>K</td>
<td>N</td>
<td>A</td>
<td>N</td>
<td>D</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>L</td>
<td>A</td>
<td>A</td>
<td>D</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

Machine 1: 
(a) Before Move Application

Machine 2: 
(b) After Move Application

Selected papers: [10,14,3]
Memetic Algorithms: Crossover

Selected papers: [16]
Hybrid techniques

Methods of Artificial Intelligence
(Machine Learning, Heuristics...)

Methods of Logic

Mathematical Optimization

\[ S_{i,d,t} \iff \bigwedge_{x=1}^{s_l_t} U_{i,d,x} \bigwedge_{y=s_l_t}^{s_l_{max}} \neg U_{i,d,y} \]

\[ \min f = 30 \sum_{\substack{s \in S, k \in K, d \in \{1...7\}}} C_{skd}^{S1} \]
\[ + 15 \sum_{\substack{n \in N, s \in S, d \in \{1...7\}}} (C_{nsd}^{S2a} + C_{nsd}^{S2b}) \]
\[ + 30 \sum_{\substack{n \in N, d \in \{1...7\}}} (C_{nd}^{S2c} + C_{nd}^{S2d}) \]
Algorithm Selection

- Usually several search algorithms are available for solving a particular problem

- **No free lunch theorem**
  “...for any algorithm, any elevated performance over one class of problems is offset by performance over another class” [1]

  “...any two algorithms are equivalent when their performance is averaged across all possible problems” [2]

How to select the best algorithm for a specific problem instance?

Algorithm Selection

Many success stories
Planning and Scheduling, Routing, Combinatorial Auctions, SAT, TSP, Graph Coloring, Tree Decomposition, Timetabling, ...
Selected papers: [15,17,18]
Algorithm selection

Input (see [8] and [9]):

- Problem space $P$ that represents the set of instances of a problem class
- A feature space $F$ that contains measurable characteristics of the instances generated by a computational feature extraction process applied to $P$
- Set $A$ of all considered algorithms for tackling the problem
- The performance space $Y$ represents the mapping of each algorithm to a set of performance metrics

Problem:
For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space, such that the selected algorithm $a \in A$ maximizes the performance mapping $y(a(x)) \in Y$

Supervised machine learning: Classification

<table>
<thead>
<tr>
<th>NrClauses</th>
<th>NrVariables</th>
<th>...</th>
<th>Best Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>80</td>
<td></td>
<td>Alg1</td>
</tr>
<tr>
<td>4000</td>
<td>400</td>
<td></td>
<td>Alg1</td>
</tr>
<tr>
<td>30000</td>
<td>8500</td>
<td></td>
<td>Alg3</td>
</tr>
<tr>
<td>300</td>
<td>78</td>
<td></td>
<td>Alg2</td>
</tr>
<tr>
<td>2000</td>
<td>540</td>
<td></td>
<td>Alg3</td>
</tr>
<tr>
<td><strong>10000</strong></td>
<td><strong>450</strong></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: SAT problem

\[ F(x) = (x_{1} \lor \overline{x}_{37} \lor x_{73}) \land (\overline{x}_{11} \lor \overline{x}_{12}) \land \ldots \land (\overline{x}_{2} \lor x_{43} \lor x_{22}) \]
## Supervised machine learning: Regression

<table>
<thead>
<tr>
<th>NrClauses</th>
<th>NrVariables</th>
<th>...</th>
<th>Alg1: Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>80</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>4000</td>
<td>400</td>
<td></td>
<td>450</td>
</tr>
<tr>
<td>30000</td>
<td>8500</td>
<td></td>
<td>2350</td>
</tr>
<tr>
<td>300</td>
<td>78</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>2000</td>
<td>540</td>
<td></td>
<td>170</td>
</tr>
<tr>
<td>10000</td>
<td>450</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Supervised machine learning techniques

- Decision/Regression Trees
- Random Forest
- Bayesian Networks
- Neural Networks
- ...

Automated Machine Learning:
- Process of automating of machine learning when applied to a data set
- Automated optimization of hyperparameters
- Automated algorithm selection
- Automated feature selection, preprocessing...
Instance Space Analysis and Algorithm Selection

Rotating Workforce Scheduling

Selected papers: [19]
Course timetabling

(b) Best performance (with time)
Reinforcement learning

Sutton and Barto. Reinforcement Learning, 2018
Hyper-heuristics

Reinforcement learning for Hyper-heuristics

Based on objective value difference and time

Hyper-heuristic

Agent

Environment

Low-level heuristics

Problem domain

Selected papers: [20,21,22]
Conclusions

- Many problems within sustainable applications are still solved manually
- AI and optimization offer tremendous potential for further improving solutions in these domains
- Combination of AI and optimization techniques is crucial

Success stories:
- Test lab scheduling
- Workforce scheduling
- Machine scheduling
- Oven scheduling
- Educational timetabling
- ...
Conclusions

- Many problems within sustainable applications are still solved manually
- AI and optimization offer tremendous potential for further improving solutions in these domains
- Combination of AI and optimization techniques is crucial

Challenges

- Domain specific solutions
- New challenging/large-scale problems
- The availability of data is critical for machine learning
  - Usually, most of the time is invested in the collection and preparation of the data
- ...


7) Felix Winter, Nysret Musliu, Emir Demirovic, Christoph Mrkvicka: Solution Approaches for an Automotive Paint Shop Scheduling Problem. ICAPS 2019: 573-581


