Creating a markup language for linear programs

Your bachelor’s thesis

**Setting** Many combinatorial optimization problems can be expressed as compact integer linear programs (ILPs). Consider, for example, the ILP formulation of the vertex cover problem. Given a graph $G = (V, E)$, the problem consists of finding a minimal amount of vertices that cover all the edges.

\[
\begin{align*}
\text{min} & \quad \sum_{v \in V} x_v \\
\text{subject to} & \quad x_u + x_v \geq 1 \quad \forall \{u, v\} \in E \\
& \quad x_v \in \{0, 1\} \quad \forall v \in V
\end{align*}
\]  

This formulation is so compact since it basically consists of only one constraint. This constraint is repeated $|E|$ times with different variables, which we can easily express with the $\forall$ quantification. However, the popular .lp format, which is interpreted by many state-of-the-art ILP solvers such as Gurobi or Mosek, requires to explicitly write down every single constraint. This can complicate the generation or debugging process of the ILP model.

In this bachelor thesis, you will define a markup language for ILPs that allows for a compact and human-readable expression of commonly used combinatorial optimization problems. This framework should offer an interface to read the compact formulation and autogenerate the code to solve this problem with an ILP solver of choice. This work will use Gurobi as the underlying ILP solver, but the framework should be modular enough to allow an easy extension to other ILP solvers.

There are certain approaches to generate complex ILPs in a compact way, such as ZIMPL [5], used by the ILP solver SCIP [3], or GAMS [2]. Another goal of this work is to compare our new framework to those existing approaches.
**Research goals**  You are supposed to work on the following topics

- Define syntax and grammar of a markup language for combinatorial optimization problems
- Implement a parser for this language
- Implement a compiler for the markup language and autogenerate code to solve the optimization problem using the Gurobi C++ interface.

The first milestone will be the ability to generate and solve the vertex cover ILP (1) with only 5 lines of code.

**Preliminaries**  You took/are taking the core course Optimization (or an equivalent course at another university), and you have basic programming skills (C/C++ preferred). Moreover, you like to code and do experimental research with your self-written framework.

**Our offer**  We offer a bachelor’s thesis for computer science. We will guide you in theoretical and practical questions and offer regular meetings to discuss the state of your thesis.

- Advisor: Maximilian John (Phd student)
- Supervisor: Dr. Andreas Karrenbauer (Senior researcher)

**Literature**  The following literature provides a full overview of the topic and can be seen as a work of reference instead of a mandatory literature.

- On (Integer) Linear Programming [6]
- On Combinatorial Optimization [1]
- About the Gurobi Interface [4]

**References**


