

# 1 Priority TaskList

## 1.1 short-term goals + High priority

- Re-read OCUS paper review-style
- Document what has been done + what not + choices
- Clean code !
- Re-read SMUS paper and see what we missed and what choices we made in the OCUS (pocus-pas) paper.
- Re-read postponing optimisation paper and see what we missed and what choices we made in the OCUS (pocus-pas) paper.
- Keeping track of the subsets in the same way as we do for OCUS incr
- Ability to zoom-in on explanations generated i.e *using nested explanations* ?
- Generate the explanation sequence with OCUS with the same cost function as in ECAI

## 1.2 Long-term goals (low priority)

- Translate the weighted constraints from OUS into an SMUS cnf specification
- setup experiment for quantifying difference in computation of this approach
- Look for a better cost-function defined on the constraints.
- Look for meta-constraints added to the specification
- Define how the nested explanations are constructed: which constraints are activable in the nested explanation reasoning and which can't or are prohibited.
- Present the user with alternative explanations (of the same cost or similar costs)

# 2 IJCAI

## 2.1 SMUS

From early experiments we see that SMUS outperforms OUS on CNF instances when they both solve the same problem, i.e. OUS with unit weights on constraints.

### 1. OUS als SMUS

**Experiments** Test on a large scale how both systems perform on CNF instances

**Theory** Re-read SMUS paper and see what we missed and what choices we made in the OCUS (pocus-pas) paper.

### 2. SMUS als OUS

**Implementation** Translate the weighted constraints from OUS into an SMUS cnf specification

- Ex: Constraints weighted 10, is implemented as 10 literals which together have to be satisfied in order to use the constraint.

**Experiment** setup experiment for quantifying difference in computation of this approach

## 2.2 Postponing optimisation

Experiments from *OCUS* show that postponing optimisation doesn't improve the computation time for building the whole explanation sequence.

**Theory** Re-read postponing optimisation paper and see what we missed and what choices we made in the OCUS (pocus-pas) paper.

**Implementation** Keeping track of the subsets in the same way as we do for OCUS incr

## 2.3 SUDOKU

Sudoku encoding in python notebook in "experiments/03\_OMUS/02\_OUS/explain\_sudoku.ipynb".

**Efficiency** Explanation generation is slow!

**Explanation Quality** The cost function defined for explaining generates explanations that can be better. For example, for a given square, the explanation algo, needs to derive a lot of negative literals (i.e. 8) which is cheaper than deriving the positive knowing together with the constraints of the 8 negatives.

**Theory** [Look for meta-constraints added to the specification](#)

**Theory** [Look for a better cost-function defined on the constraints.](#)

**Future work** [Defining a way to learn cost function structures based on the feedback of users?](#)

**Implementation** [Ability to zoom-in on explanations generated i.e \*using nested explanations\* ?](#)

## 2.4 Visualisation

Code for generating the visualisation in python is ready.

**Implementation** [Generate the explanation sequence with OCUS with the same cost function as in ECAI](#)

## 2.5 Explanations

Ideas for improving the current OCUS explanations.

### 2.5.1 Nested explanations

When deriving a fact, try to generate explanation with counterfactual reasoning.

**Implementation** [Generate nested explanations with OCUS](#)

**Theory** [Define how the nested explanations are constructed. Which constraints are activable in the nested explanation reasoning and which can't or are prohibited.](#)

- Example: For a given square in the grid. Certain explanations in the puzzles are using 2 bijectives in order to be able to use the information for a third bijectivity. The problem is that the third bijectivity could have been used in the first place, but **wasn't** because the explanation algorithm prohibited from using these constraints.

### 2.5.2 Alternative explanations - User experiments

See future work of 2.3.

**Theory** [Present the user with alternative explanations \(of the same cost or similar costs\)](#)

## 3 FWO

Content to be discussed at the kick-off meeting.