Combinatorial optimization for JSOCMSR problem by using a SAT solver augmented with a domain-specific heuristic

Author - Artjom Pugatšov
A.Pugatsov@student.tudelft.nl

Responsible professor - Emir Demirović
Supervisors - Konstantin Sidorov, Maarten Flippo

How can domain-specific **heuristics** be applied to **SAT** representation of **JSOCMSR** problem?
What is their impact on the performance compared to general SAT heuristics?

01. Motivation

Job sequencing with one common and multiple secondary resources (JSOCMSR) is an **NP-hard** problem [1].

Existing JSOCMSR research focuses on new problem-specific algorithms.

SAT solvers are powerful **general-purpose** algorithms. Although, if applied directly, problem-specific information is **not utilized**.

By utilizing **problem-specific heuristics**, a general-purpose SAT solver can still be used, but the performance is improved.

02. Problem use cases

Metal forming

[3]

Metal pouring machine is the common resource. Forms are the secondary resources

Cancer treatment scheduling

[4]

Beam therapy for cancer treatment requires a costly beaming device. The beam is the common resource. Rooms are the secondary resources

03. Job sequencing with one common and multiple secondary resources (JSOCMSR)

- All jobs use **one common** resource and **one secondary** resource
- Secondary is used throughout the **entire** duration. Common only during a **segment**
- No two jobs can use the same resource at the same time
- The goal is to schedule all jobs such that the latest job finishes **as early as possible**

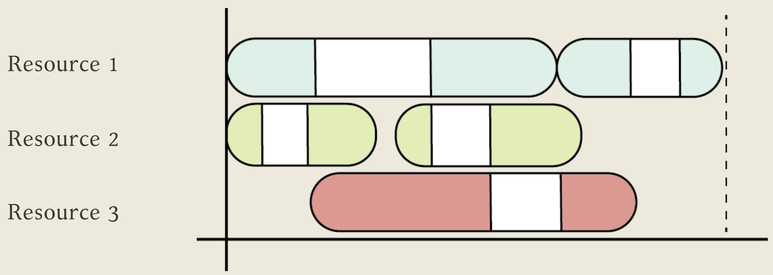

Fig 1.Example of a valid schedule

04. Methodology

- The problem was modeled in terms of **MaxSAT**
- The baseline general heuristic's performance was measured
- The performance of the heuristically augmented version was measured
- They were compared

05. MaxSAT

Maximum satisfiability problem (MaxSAT) consists of:

- Variables that can be **True or False**
- Constraints **AND/OR/NOT** relating variables

The goal is to set **all hard** constraints to True and **maximize** the number of True **soft** constraints

06. SAT encoding

Variables:

$S = \{s_{j,t} \mid j \in Jobs, t \in StartTimes\}$ — potential **start times** of all jobs

$U = \{u_{j,t,r} \mid j \in Jobs, t \in [0, .., maxEndTime], r \in Resources\}$ — resource r is **used** by job j at time t including common and secondary resource

$M = \{u_t \mid t \in [0, .., maxEndTime]\}$ — time t is within the **makespan**

Hard constraints:

- Constraints that directly model the problem: resource used throughout the entire job; no conflicts between jobs; all jobs need to be scheduled
- Auxiliary constraints for determining makespan: the time job ends at is within the makespan; if time is within the makespan, all previous times are within the makespan

Soft constraints:

- Makespan variables are false

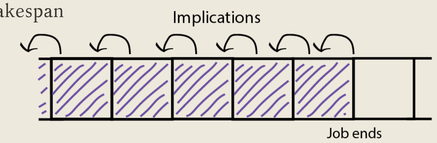

Fig 2.Illustration of the auxiliary "All the previous times are within makespan" constraint

07. Heuristic

Baseline:
Variable State Independent Decaying Sum (**VSIDS**) determines the order of variable assignments based on previous conflicts. It is widely used and has shown effectiveness for a variety of different SAT instances [2].

Augmented:
Variables are first ordered by start time and then by duration. This way the **longest** jobs are scheduled **as early as possible.**

Two heuristic versions have been tested:

1. Plain variable ordering
2. Variable ordering with VSIDS

08. Experiments and results

Three sets of experiments have been run:

1. Baseline VSIDS
2. Only variable ordering
3. Variable ordering + VSIDS

Variable ordering + VSISD has the same performance as the baseline for smaller instances of less than 100 jobs as seen in Figure 3

Heuristic only has a better performance than the baseline for bigger instances with 100 and more jobs

VSIDS + heuristic has the best performance for bigger instances

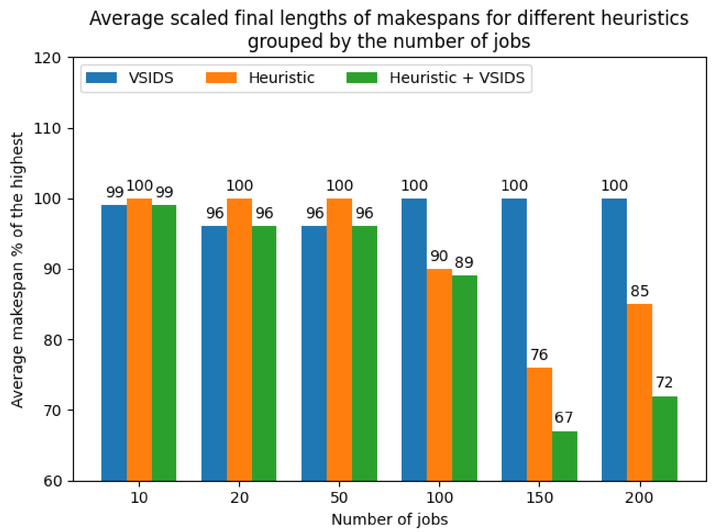

Fig 3. Average scaled final makespan

09. Future work

The encoding of the problem can be modified by taking into account an initial upper bound

Problem-specific heuristics for constraint satisfaction problem (CSP) formulation can be explored.

References:

[1] Matthias Horn, Gunther Raidl, and Christian Blum. Job sequencing with one common and multiple secondary resources: An a*/beam search based anytime algorithm. Artificial Intelligence, 277:103173, 09 2019

[2] Karem A Sakallah et al. Anatomy and empirical evaluation of modern sat solvers. Bulletin of the EATCS, (103):96–121, 2011

[3] Kilroy79, molten metal industry logo liquid iron or steel vector, <https://www.vectorstock.com/royalty-free-vector/molten-metal-industry-logo-liquid-iron-or-steel-vector-29528602>

[4] Radiotherapy free icon. <https://www.flaticon.com/freeicon/radiotherapy_4193423? related_id=4193418>