Augmenting Constraint Programming Variable Selection with Domain-Specific Heuristics for a Prize-Collecting Scheduling Problem

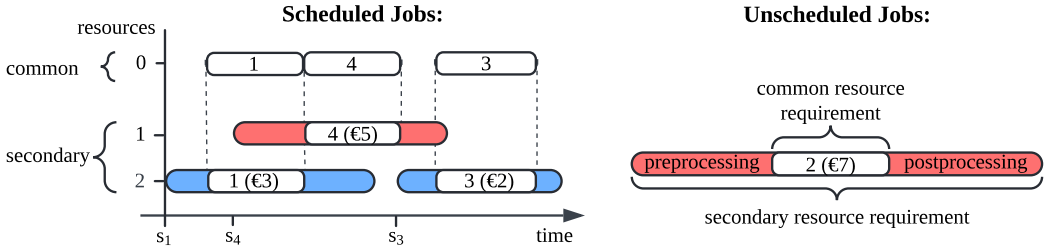
Nikola Petrov
n.p.petrov-1@student.tudelft.nl

Supervisors: Dr. Emir Demirović, Maarten Flippo, Imko Marijnissen

TU Delft

1. Background

Prize-Collecting Job Sequencing with One Common and Multiple Secondary Resources is NP-Hard [1]



Applications:

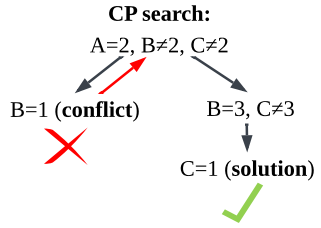
Particle therapy patient scheduling [2]

Scheduling of avionic systems [3]

Constraint Programming (CP):

Variables:
- A∈{2,3}
- B∈{1,3}
- C∈{1,2}

Constraints:
- A≠B
- A≠C
- B≠C



2. Research Question

Can **CP solvers' convergence rate** and **solution quality be improved** using **domain-specific variable selection** based on **job densities** (prize over duration)?

References:

[1] M. Horn, G. R. Raidl, and E. Rönnberg, "A* search for prize-collecting job sequencing with one common and multiple secondary resources," Annals of Operations Research, vol. 302, no. 2, pp. 477–505, 2021.
[2] J. Maschler, M. Riedler, M. Stock, and G. R.Raidl, "Particle therapy patient scheduling: First heuristic approaches," in Proceedings of the 11th Int. Conference on the Practice and Theory of Automated Timetabling. Udine, Italy, 2016, pp. 223–244.
[3] M. Blikstad, E. Karlsson, T. Lööw, and E. Rönnberg, "An optimisation approach for pre-runtime scheduling of tasks and communication in an integrated modular avionic system," Optimization and Engineering, vol. 19,pp. 977–1004, 2018.
[4] Moskewicz, Matthew W., et al. "Chaff: Engineering an efficient SAT solver." *Proceedings of the 38th annual Design Automation Conference*. 2001.

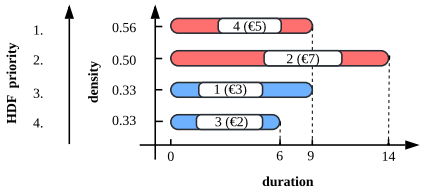
3. Methodology

Variable Selection Heuristics:

Variable State Independent Decaying Sum (VSIDS) [4] (Baseline)

Variables = { J1, J2, J3, J4}
Initial Weights(**VSIDS**) = { 0, 0, 0, 0}
Weights(**VSIDS**) = { **1**, **1**, 0, 0}
Weights(**VSIDS**) = { 1, **2**, **1**, 0}

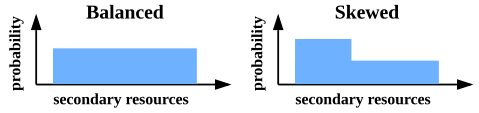
Highest Density First (HDF)



VSIDS + Density

Variables = { J1, J2, J3, J4}
Initial Weights(**VSIDS+Density**) = {0.33, 0.50, 0.33, 0.56}
Weights(**VSIDS+Density**) = {0.33, **1.50**, 0.33, **1.56**}
Weights(**VSIDS+Density**) = {**1.33**, **2.50**, 0.33, 1.56}

Instance sets[1]:

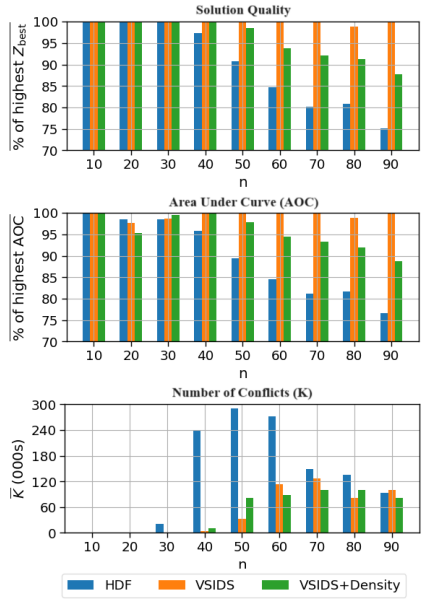


Measurements:

Z_{best} : Final Total Prize(**solution quality**)
AOC : Area Under Curve(**convergence rate**)
K : Conflicts(**to explain the above**)

4. Results

results from **Balanced set, 2 sec. res.**



Solution Quality and Convergence Rate

Density methods deteriorate as instances get larger both in terms of **AOC** & **solution quality**

Conflicts

HDF generates **numerous conflicts** or **slows down** for larger instances. **VSIDS + Density initially mirrors HDF, later recovers** but **slow start effects are noticeable**.

Other Instance Sets

3 secondary resources (easier than 2 sec. res.): **similar trend, lower deterioration of density methods**

skewed (harder than balanced): **similar trend, higher deterioration of density methods**

5. Conclusion & Future Work

Prioritizing high-density jobs is ineffective for PC-JSOCMSR as the common resource is the main bottleneck.

Future Work:

- value selection **preserving common and secondary resources.**
- variable selection based on **upper bounds** computed using **PC-JSOSMS relaxations [1]**