9th International Conference "Distributed Computing and Grid Technologies in Science and Education"
(GRID'2021)

Some Aspects of the Workflow Scheduling in the Computing Continuum Systems

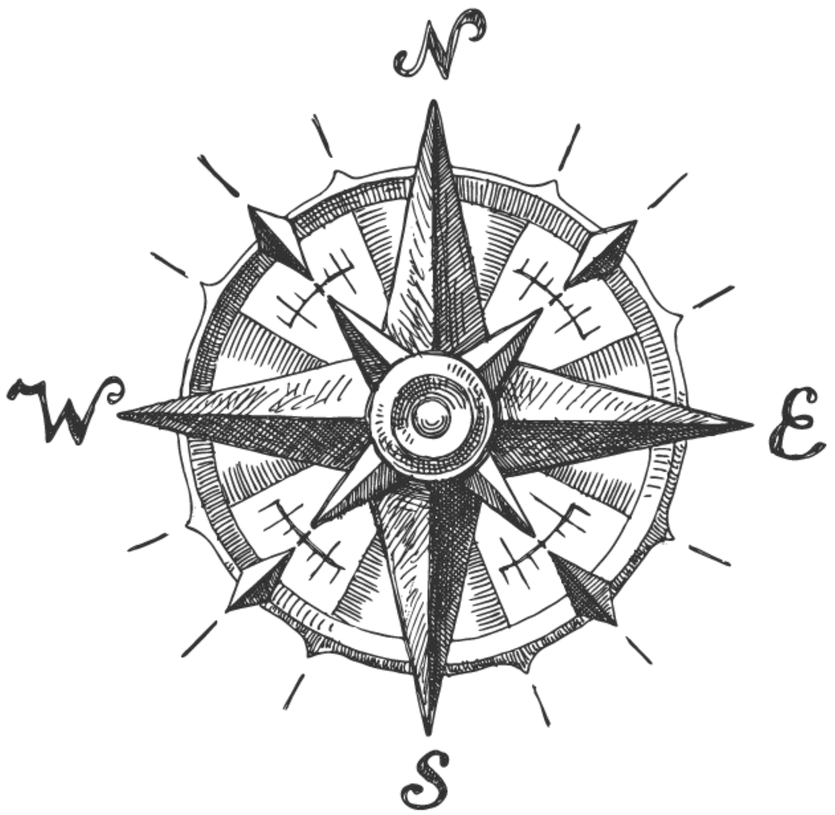
*MSc **Vladislav Kashansky**, Dr. Gleb Radchenko, Dr. Radu Prodan*

vladislav.kashanskii@aau.at

Virtual Event, 07.07.2021

Talk Structure

- Concept of the Computing Continuum
- Workflows Concept
- MRCPSP Problem Formulation
- MILP Event-Based Model
- Difficulties of the Solution, Lower Bounds
- Heuristic Approach
- Future Directions

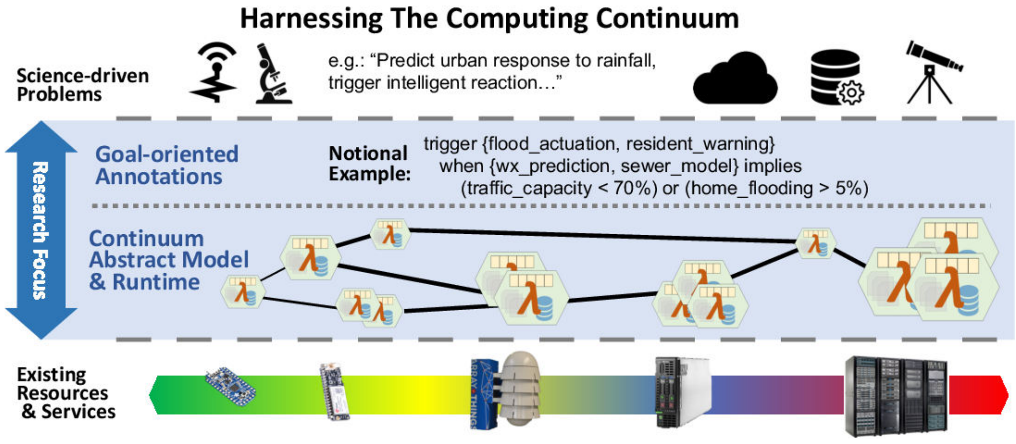


Computing Continuum – Current State

- Recent advancements in the field of parallel and distributed computing led to the definition of the **computing continuum** as the environment comprising highly heterogeneous systems with dynamic spatio-temporal organizational structures, varying in-nature workloads, complex control hierarchies, governing computational clusters with multiple scales of the processing latencies, and diverse sets of the management policies.

- **Examples:**
  - **social platforms** that analyze concurrently various motion patterns and opinion dynamics related to human behavior at various spatio-temporal scales;
  - self-organizing **vehicle fleets** and **drone swarms** that receive information from a large group of spatial sensors and need to make decisions locally.

- Since emergence of the concept in 2020 **[1]** , there is a lack of a **reproducible model** of the computing continuum, especially for better understanding scheduling heuristics, as real systems do not preserve this quality and hinder the comparative performance analysis of the novel scheduling approaches.



Beckman, P., et al. [1]

Networks, networks, networks …

- **Computational network**, which provides ***structural*** knowledge about the possible information flows within the system and possible interactions of agents via adjacency matrix;

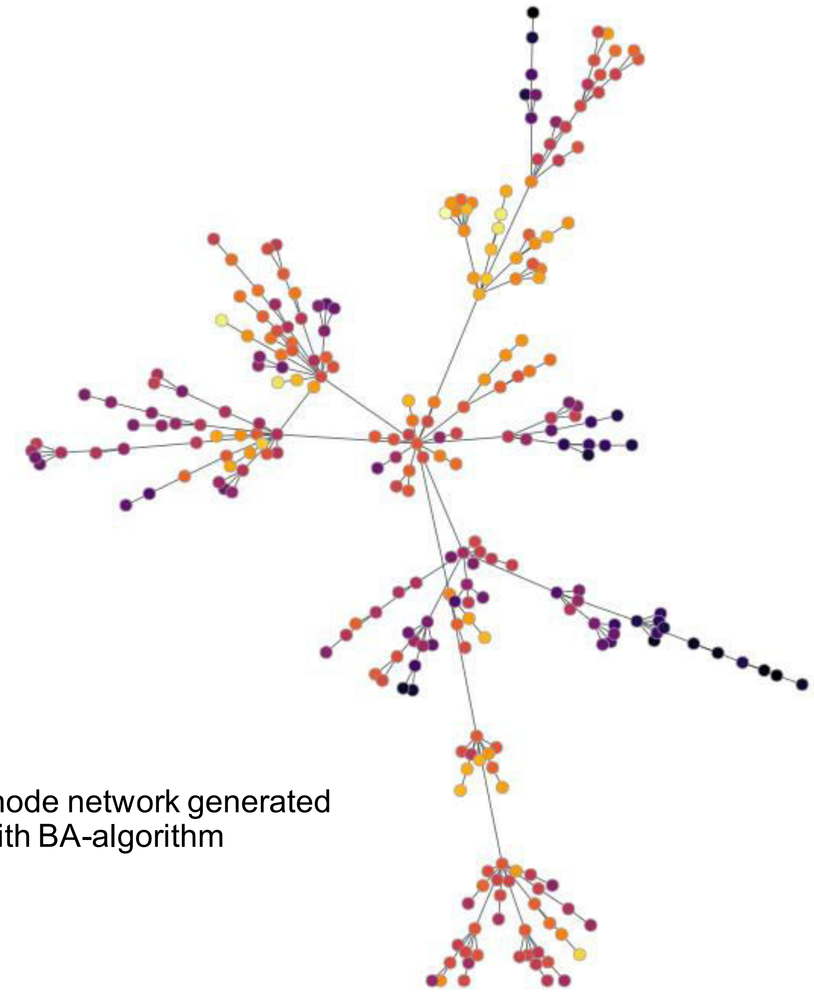
- **Recursive network,** which forms a multi-layer DAG and provides knowledge about non-equilibrium dynamic processes in the computational network. This component is optional and only required if behavior of the network is considered far from equilibrium. Examples include: **Dynamic Bayesian Networks** (DBN) and **Hidden Markov Models** (HMM);

- **Workload network** is the set of tasks, represented as the **DAG** that governs computational process in the computing continuum, prescribing **arrow of the time**.

Structural Model of the Computational Network with Scale-Free Topology

- Class of random networks with scale-free property;
- Describes well some natural and human-made systems, including the Internet, the world wide web, citation networks, and some social networks are thought to be approximately scale-free;
- The network begins with an initial connected network of n nodes;
- New nodes are added and attached with probability:

$$p_i = \frac{k_i}{\sum_j k_j}$$



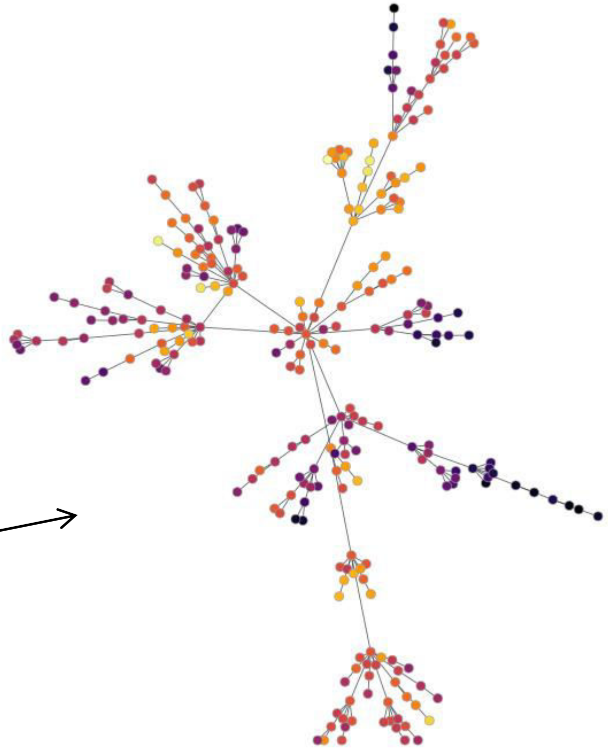
256-node network generated with BA-algorithm

Spin-based model over Computational Network

- For an analytical insight into simulation and phase transitions, we rely on the equilibrium statistical physics framework, which considers that the network N can be in any possible microscopic configuration;

- Considering dynamical model defined over the network N we map the two-dimensional unit vector in frames of XY-model [2]:

$$\vec{S}_m = (cos(\theta_m),\ sin(\theta_m))$$



- Direct interpretation consists in considering the existence of a centralized policymaker in the computing continuum. A spin vector S then represents the dynamic scalar degree of agent's belief to the "center", prescribing mechanism of the consensus formation.

$$\mathcal{H}(\theta) = \sum_{m \in A} \sum_{q \neq m} J_{mq} \cdot [1 - \cos(\theta_m - \theta_q)]$$

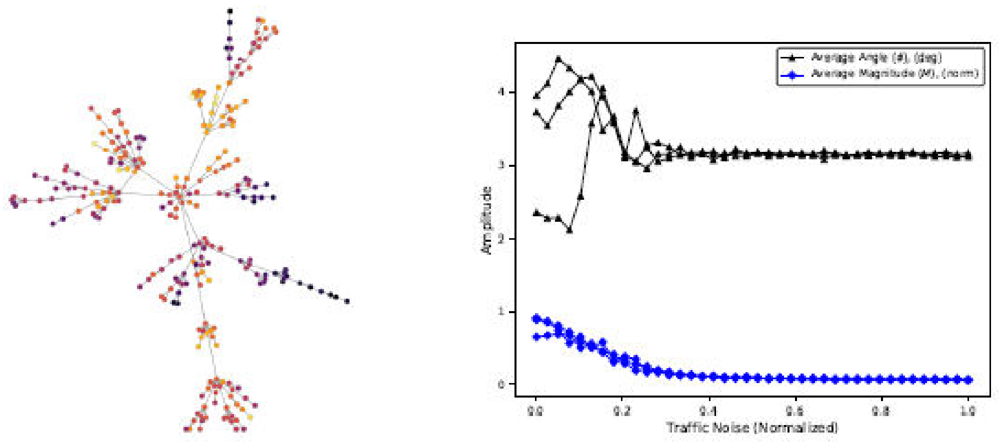
Numerical Simulations

- We obtain **[2]** the distribution P via Monte Carlo simulation using the Metropolis algorithm.

- At each time step, we induce a random walk on the graph N, choose one random spin and rotate its angle by some random increment, keeping it in a range [0; 2pi]. States are accepted with the probability:

$$p(\theta_{n+1} | \theta_n) = \min\{1, e^{-\Delta\mathcal{H}}\}$$

$$\Delta\mathcal{H} = \mathcal{H}_{n+1} - \mathcal{H}_n$$



(a) (b)

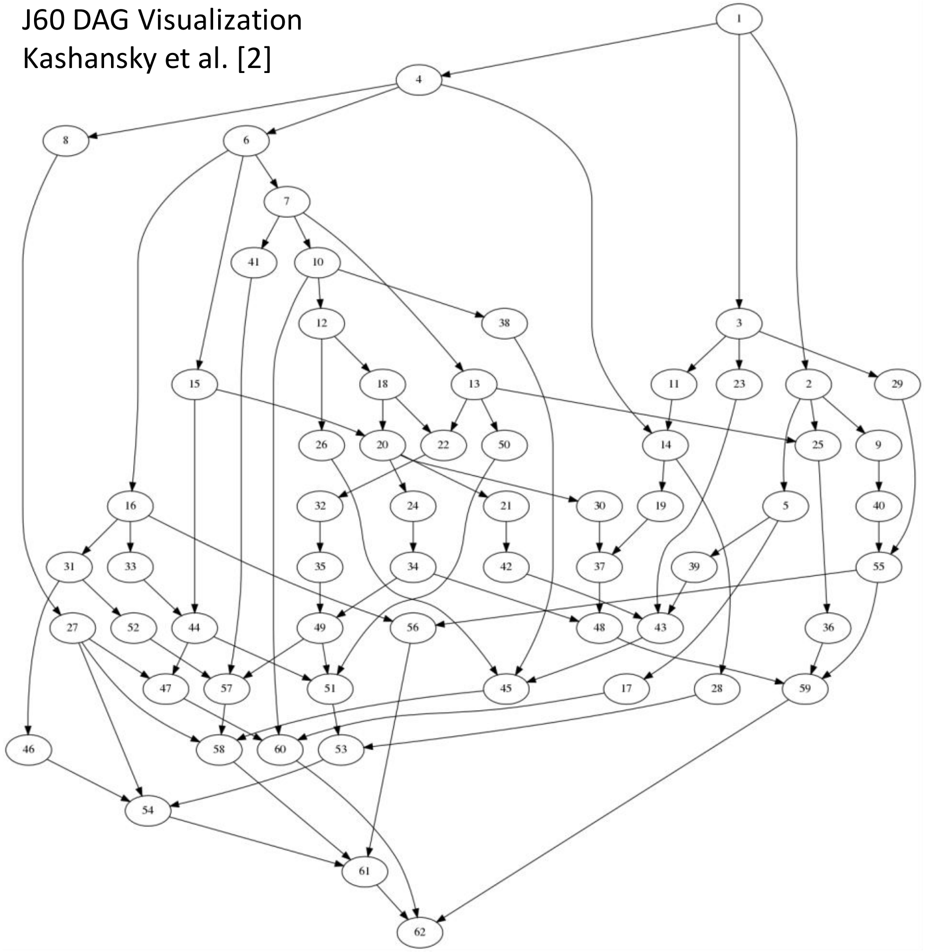
(a) Snapshot after $1.5 \cdot 10^3$ iterations of the Metropolis-Hastings dynamics of the model defined over Barabasi-Albert network ($n = 256$, $\eta = 0.1$) with triangular initial graph; darker colors correspond to values of θ_m close to 0. Visualized with Fruchterman-Reingold layout algorithm and Cairo library. (b) Order parameters ϕ and $\mathcal{M}$ as functions of the noise regime η.

Kashansky et al. [2]

- Paper [2] accepted to ICCS 2021 Conference.

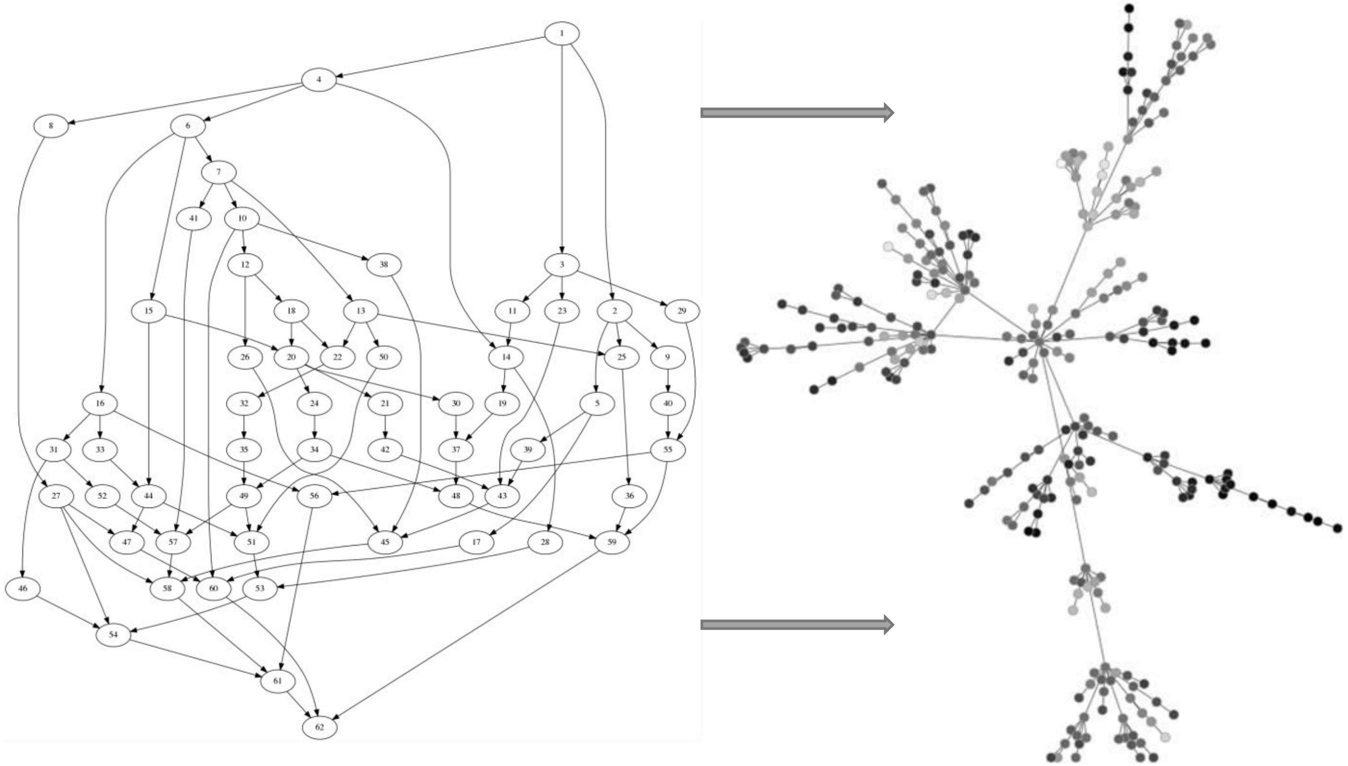
Workflow Network and DAGs

J60 DAG Visualization
Kashansky et al. [2]



- Workflow represented via Directed Acyclic Graph (DAG);
- It encodes strict precedence-relation;
- Represents scenarios where the order of task execution is not negligible;

Workflow Network and Mapping



- We further define a data matrix D that indicates the amount of data transmitted from a task i to a task j. Consequently, we obtain the delay tensor D* for transferring data from task i to task j assigned to the agents m and q:

$$d^*_{ijmq} = \overbrace{\mathcal{T}_{mq}}^{\text{Connection Delay}} + \overbrace{\mathcal{D}_{ij} \cdot \mathcal{B}^{-1}_{mq}}^{\text{Data Transfer Latency}}$$

- The first term T in equation above represents a connection estimation delay, assumed as a small constant. This approach models realistic scenarios of synchronized routing information, leading to the fast connection estimation with low delay.

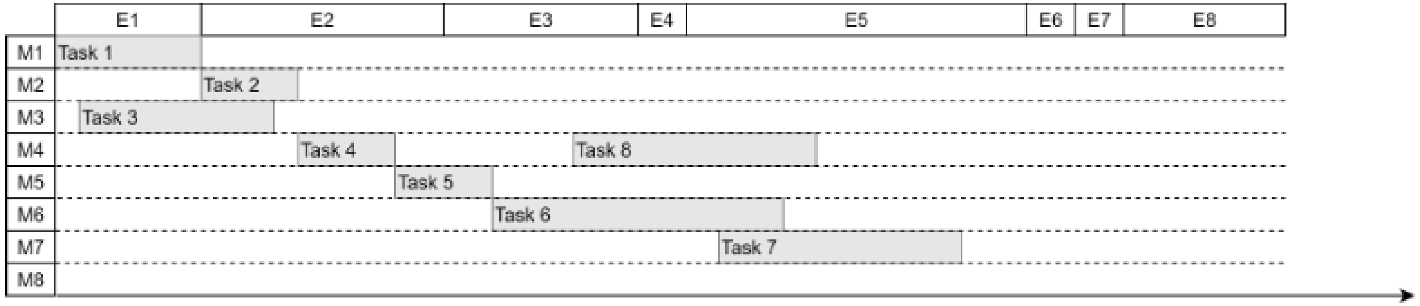
- We compute the execution time of the given task in V with the following formula:

$$\tau^1_{im} = \max_j \{d^*_{ijmq};\ \forall j \in \mathcal{P}_i\} + \tau^0_{im} \cdot \mathcal{B}^{-1}_{mm}$$

MRCPSP Problem

- Multi-Mode Resource-Conserained Project Scheduling Problem

  - Considers set of the **heterogeneous machines** with different processing speeds and other properties like reliability and cost
  - Considers **set of tasks** with precedence constraints in the form of DAG
  - Considers sets of **resource constraints** specified for each task-machine pair
  - Various objectives are possible: Makespan, Weighted number of late jobs, Total Costs etc.



MRCPSP Problem – MILP Formulation

- Minimum makespan MILP Formulation, extension of the work [3]
- **NP-hard** problem in the *strong sense*
- *Incorporates:*
  - Assignment Constraints;
  - Precedence Constraints;
  - Resource Constraints
- Solving to optimality – **HPC problem** domain
- Run-times are given by:

$$\tau_{im}^1 = \max_j \{d_{ijmq}^*;\ \ \forall j \in \mathcal{P}_i\} + \tau_{im}^0 \cdot \mathcal{B}_{mm}^{-1}$$

$$\begin{aligned}
\min \quad & t_n \\
\text{S.t:} \quad & t_0 = 0 \\
& t_{e+1} - t_e \geq 0 && \forall e \in \mathcal{E} \setminus \{n\} \\
& t_f - t_e - \tau_{im}^1 \cdot x_{ime} + \tau_{im}^1 \cdot (1 - y_{imf}) \geq 0 && \forall m \in M, \forall i \in V, \forall (e,f) \in E^2 \wedge e \leq f \\
& \sum_{m \in M} \sum_{f \in E} f \cdot y_{imf} - \sum_{m \in M} \sum_{e \in E} e \cdot x_{ime} \geq 1 && \forall i \in V \\
& \sum_{m \in M} \sum_{e \in E} x_{ime} = 1 && \forall i \in V \\
& \sum_{m \in M} \sum_{e \in E} y_{ime} = 1 && \forall i \in V \\
& \sum_{e \in \mathcal{E}} x_{ime} - y_{ime} = 0 && \forall i \in V, \forall m \in M \\
& \sum_{m \in M} \sum_{a=e}^{n} y_{ima} + \sum_{m \in M} \sum_{b=0}^{e-1} x_{jmb} \leq 1 && \forall (i,j) \in A, \forall e \in \mathcal{E} \\
& r_{0k}^* - \sum_{i \in V} r_{ik} \cdot x_{ik0} = 0 && k = 1, ..., m \\
& r_{ek}^* - r_{e-1,k}^* + \sum_{i \in V} r_{ik} \cdot (y_{ike} - x_{ike}) = 0 && k = 1, ..., m, \forall e \in \mathcal{E} \setminus \{0\} \\
& 0 \leq r_{ek}^* \leq 1 && k = 1, ..., m, \forall e \in \mathcal{E} \\
& x_{ime} \in \{0,1\} && \forall i \in V \\
& y_{ime} \in \{0,1\} && \forall i \in V \\
& t_e \geq 0 && \forall e \in E
\end{aligned}$$

Two-Phase Heuristic

- 1st phase – Generate machine to task matching taking into account locality principles and processing speed-related weighting;
  - *Possible non-markovian extensions to implement with Simulated Annealing and Genetic Programming;*
  - *Highly depends on the objective space structure;*
  - *Most fast heuristics like HEFT finish only 1st phase;*

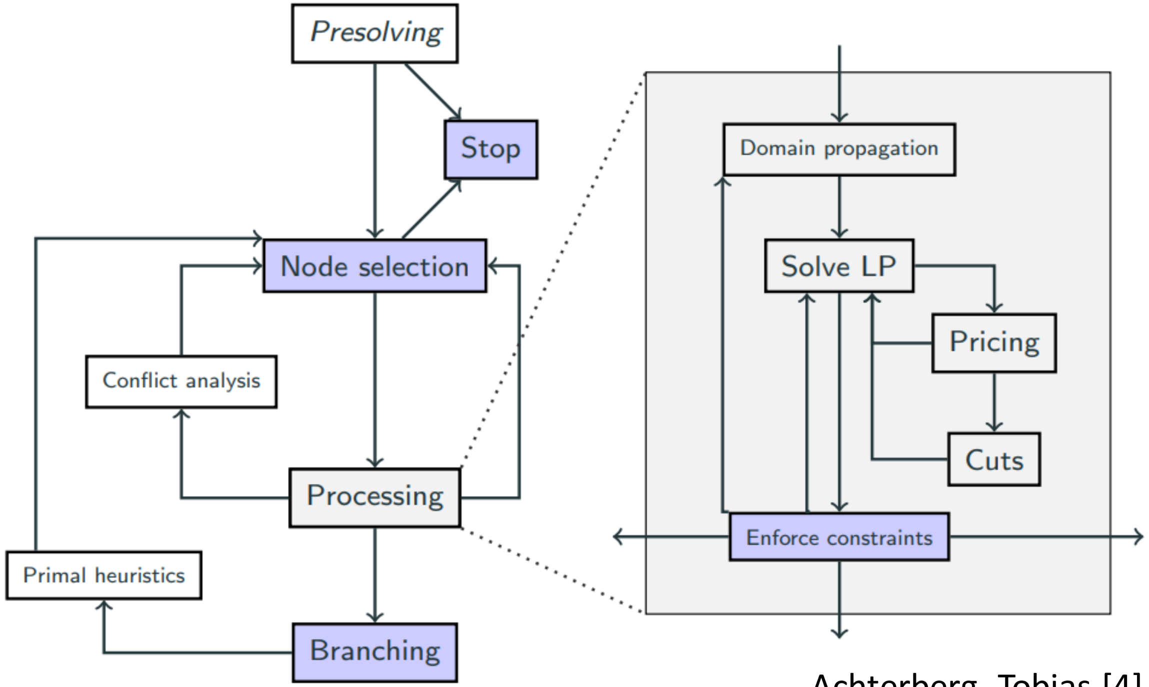
- 2nd phase – Generate minimum makespan schedule taking into account resource constraints;
  - *Problem reduction to the RCPSP case;*
  - *Wide variety of heuristics available;*
  - *Discussion of the MILP RCPSP scenario [3] with O(2n^2 + 2n) binary variables, O(n + 1) continuous variables;*
  - *Solutions quality is much higher, however at increased computation costs;*

SCIP Optimization Suite

- Provides a fast open-source IP, MIP and MINLP solver;

- Incorporates
  - MIP features (cutting planes, LP relaxation);
  - MINLP features;
  - CP features (domain propagation);
  - SAT-solving features (conict analysis, restarts);
  - branch-cut-and-price framework,
  - Has a modular structure via plugins;
  - Free for academic purposes.

- Possible to parallelize branch-and-bound based methods in a distributed or shared memory computing environment.



Achterberg, Tobias [4]

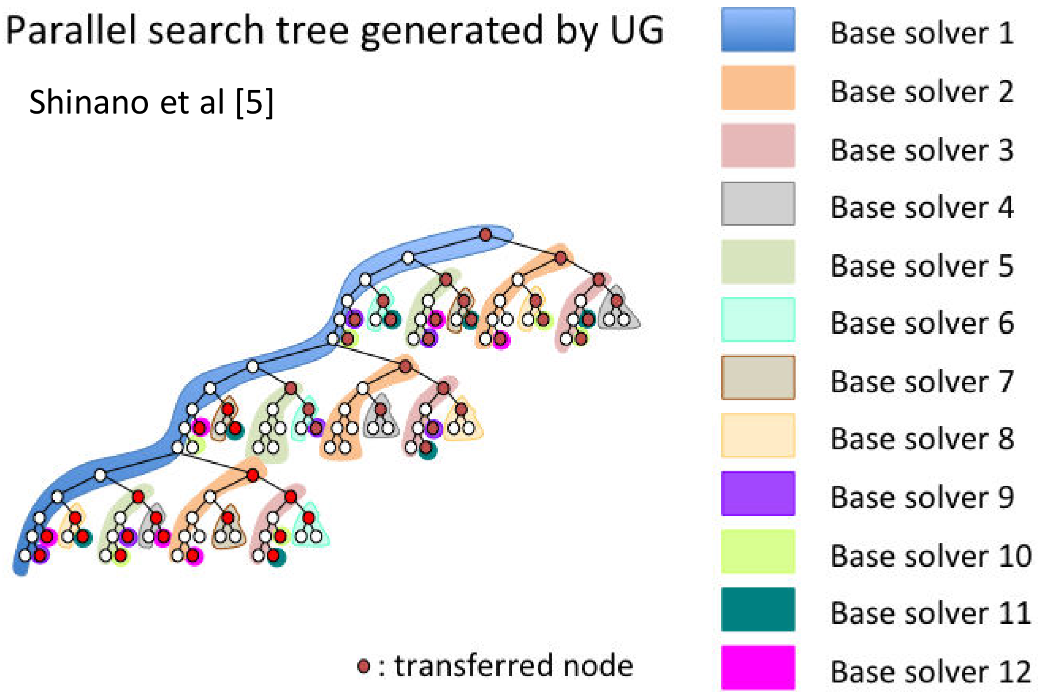
Preliminary Results - SCIP Direct Run in Single Thread Regime

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dualbound	primalbound	gap
Q 0.3s	1	0	2	-	9118k	0	-	355	609	355	609	0	0	0	0.000000e+00	1.321000e+03	Inf
k 0.3s	1	0	8	-	9487k	0	-	355	611	355	609	0	2	0	0.000000e+00	1.318000e+03	Inf
V 0.3s	1	0	9	-	9563k	0	-	355	628	355	609	0	20	0	0.000000e+00	1.302000e+03	Inf
V 0.3s	1	0	9	-	9619k	0	-	355	633	355	609	0	25	0	0.000000e+00	1.193000e+03	Inf
0.3s	1	0	83	-	9631k	0	25	355	671	355	609	0	63	0	0.000000e+00	1.193000e+03	Inf
0.6s	1	0	544	-	16M	0	52	355	684	355	701	92	76	0	0.000000e+00	1.193000e+03	Inf
0.7s	1	0	613	-	17M	0	44	355	686	355	712	103	78	0	0.000000e+00	1.193000e+03	Inf
E 0.7s	1	0	1021	-	17M	0	44	355	687	355	712	103	79	0	0.000000e+00	7.570000e+02	Inf
L 0.7s	1	0	1021	-	17M	0	44	355	687	355	712	103	79	0	0.000000e+00	6.820000e+02	Inf
* 2.1s	24	2	2798	57.4	18M	7	-	332	1002	332	609	54	406	484	0.000000e+00	5.620000e+02	Inf
* 2.3s	31	5	3093	54.0	19M	9	-	332	950	332	609	54	438	586	0.000000e+00	4.920000e+02	Inf
* 2.3s	31	5	3093	54.0	19M	9	-	332	950	332	609	54	439	588	0.000000e+00	4.920000e+02	Inf
* 2.3s	32	4	3100	52.5	19M	9	-	332	832	332	609	54	445	593	0.000000e+00	4.460000e+02	Inf
* 2.3s	32	4	3100	52.5	19M	9	-	332	832	332	609	54	445	594	0.000000e+00	3.930000e+02	Inf
* 2.6s	46	8	3490	45.0	19M	15	-	332	919	332	617	86	553	742	0.000000e+00	3.790000e+02	Inf
* 2.6s	46	8	3490	45.0	19M	15	-	332	919	332	617	86	553	743	0.000000e+00	3.490000e+02	Inf
* 2.6s	47	7	3494	44.1	19M	15	-	332	887	332	617	86	556	744	0.000000e+00	3.400000e+02	Inf
* 2.6s	47	7	3496	44.2	19M	15	-	332	887	332	617	86	556	745	0.000000e+00	3.260000e+02	Inf
* 2.6s	47	7	3496	44.2	19M	15	-	332	887	332	617	86	556	746	0.000000e+00	2.960000e+02	Inf
3.2s	100	16	4577	31.6	21M	17	7	332	896	332	617	192	692	975	1.020000e+02	2.960000e+02	190.20%
* 3.4s	182	23	5006	19.7	21M	20	-	332	934	332	617	199	772	1045	1.470000e+02	2.650000e+02	80.27%
s 3.4s	183	22	5007	19.6	21M	20	4	332	934	332	617	199	772	1045	1.470000e+02	2.570000e+02	74.83%
* 3.4s	186	20	5009	19.3	21M	22	-	332	923	332	617	199	773	1045	1.470000e+02	2.440000e+02	65.99%
3.4s	200	22	5031	18.1	21M	22	-	332	913	0	0	199	785	1045	1.470000e+02	2.440000e+02	65.99%
3.5s	300	26	5436	13.4	21M	27	-	332	1041	0	0	205	916	1090	1.470000e+02	2.440000e+02	65.99%

- Direct run with SCIP in Default Configuration
- Major problem – ***Weak LP Lower Bounds !***

- 10 machines and 10 tasks
- Long running times with large duality gap, even for low dimensional problems

Preliminary Results - Parallel SCIP Direct Run over OpenMPI



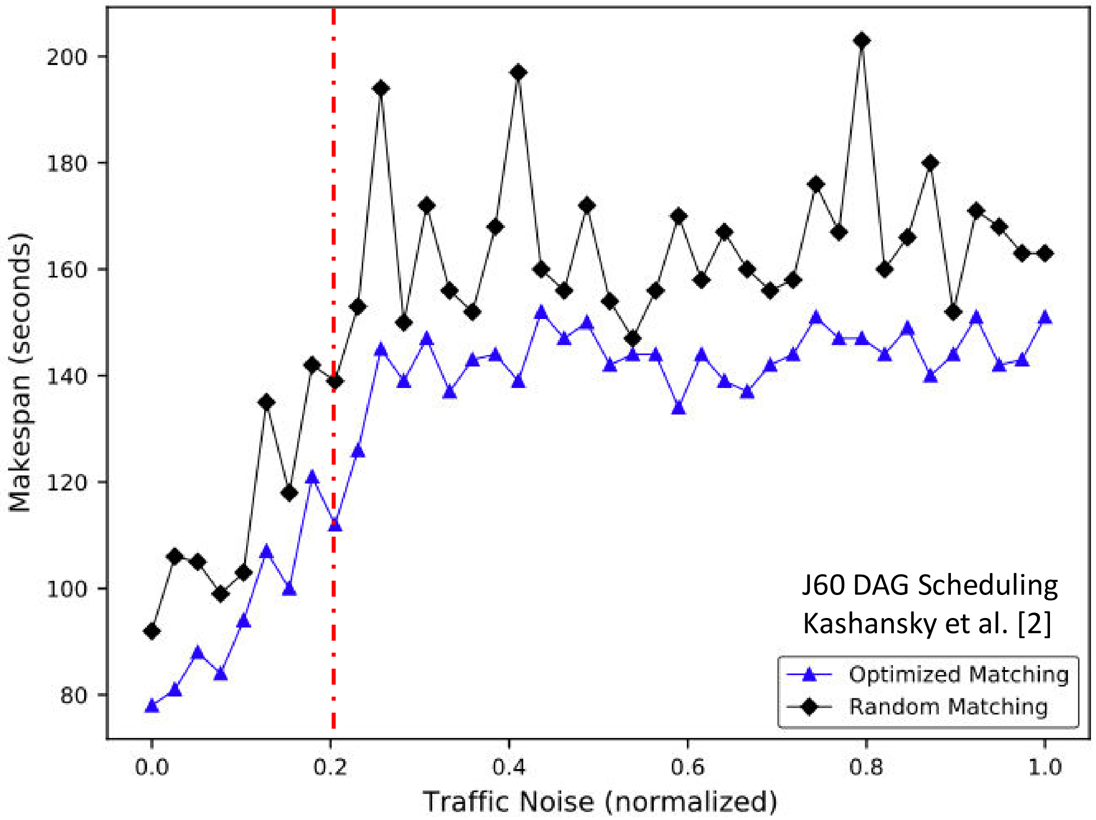
Time	Nodes	Left	Solvers	Best Integer	Best Node	Gap
1	0	1	40	1591.0000	-	-
2	2	1	40	1493.0000	1394.0000	7.10%
5	2	1	40	1493.0000	1398.0000	6.80%
10	13	12	40	1493.0000	1398.0000	6.80%
12	22	21	40	1488.0000	1402.0000	6.13%
14	27	26	40	1450.0000	1402.0000	3.42%
14	27	26	40	1448.0000	1402.0000	3.28%
15	29	24	40	1448.0000	1402.0000	3.28%
16	34	27	40	1439.0000	1402.0000	2.64%
17	34	27	40	1432.0000	1402.0000	2.14%
17	34	27	40	1431.0000	1402.0000	2.07%
17	34	27	40	1425.0000	1402.0000	1.64%
18	52	17	40	1420.0000	1406.0000	1.00%
19	53	18	40	1419.0000	1406.0000	0.92%
19	53	18	40	1418.0000	1406.0000	0.85%
20	60	4	40	1418.0000	1406.0000	0.85%
23	88	0	0	1418.0000	1418.0000	0.00%

- Parallel SCIP Direct Run over MPI
- 40 Parallel Solvers
- 10 machines and 10 tasks
- Solved moderately fast for low dimensional instance

Further Simplification – SGS Heuristic / SCIP Scheduler

- Variation of the list scheduling heuristic **[6]**
- Jobs are considered in a topological order (e.g., sorted by their earliest start)
- Scheduled according to that order as early as possible respecting the precedence and resource constraints;
- Runs in ***O***(J^2 * K) where J is number of tasks and K number of resource constraints;
- Polynomial algorithm: scalability no longer an issue, can be researched on the larger scales for practical purposes
- Precision reduced to integer domain, provides sub-optimal solutions, not applicable to the general cases of arbitrary objective spaces

Preliminary Numerical Results for the SGS Heuristic



- Makespan of the DAG obtained with a variant of the classic local descent with monotonic improvement in the objective function.
- Single and several (*20*) iterations agent selection vs noise regime variation.
- Both cases use Forward-Backward SGS (RCPSP/max) for minimum makespan derivation.
- Task times obtained by scaling normalized execution and transfer times to seconds.

Conclusion and Future Work

- We have discussed the definition of the computing continuum and provided a theoretical model of how high-order computational properties emerge within MRCPSP framework;

- We have initially studied the behavior of the MRCPSP/makespan problem over fully observable computational network.

- We have discussed weak and strong aspects of the problem in terms of two-phase heuristic approach

- Future work:
  - **Specific algorithms for OpenMP and OpenMPI techniques to speed-up computations;**
  - **Large-scale simulations requre non-trivial GPU acceleration techniques;**
  - **We expect to carry out more detailed comparative study of the several heuristics;**

Thank you
Q&A

vladislav.kashanskii@aau.at

References

- [1] Beckman, P., Dongarra, J., Ferrier, N., Fox, G., Moore, T., Reed, D., & Beck, M. (2020). Harnessing the Computing Continuum for Programming Our World. Fog Computing: Theory and Practice, 215-230.
- [2] Vladislav Kashansky, Gleb Radchenko and Radu Prodan, Monte-Carlo Approach to the Computational Capacities Analysis of the Computing Continuum, Proceedings of ICCS’21
- [3] Koné, O., Artigues, C., Lopez, P. and Mongeau, M., 2011. Event-based MILP models for resource-constrained project scheduling problems. *Computers & Operations Research*, *38*(1), pp.3-13.
- [4] Achterberg, Tobias. "SCIP: solving constraint integer programs." Mathematical Programming Computation 1.1 (2009): 1-41.
- [5] Yuji Shinano, Tobias Achterberg, Timo Berthold, Stefan Heinz, and Thorsten Koch, ParaSCIP – a parallel extension of SCIP, Competence in High Performance Computing 2010, editors: C. Bischof, H.-G. Hegering, W. E. Nagel, and G. Wittum, pages 135–148, Springer, 2012.
- [6] Rainer Kolisch and Sönke Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research, 174(1):23–37, 2006.