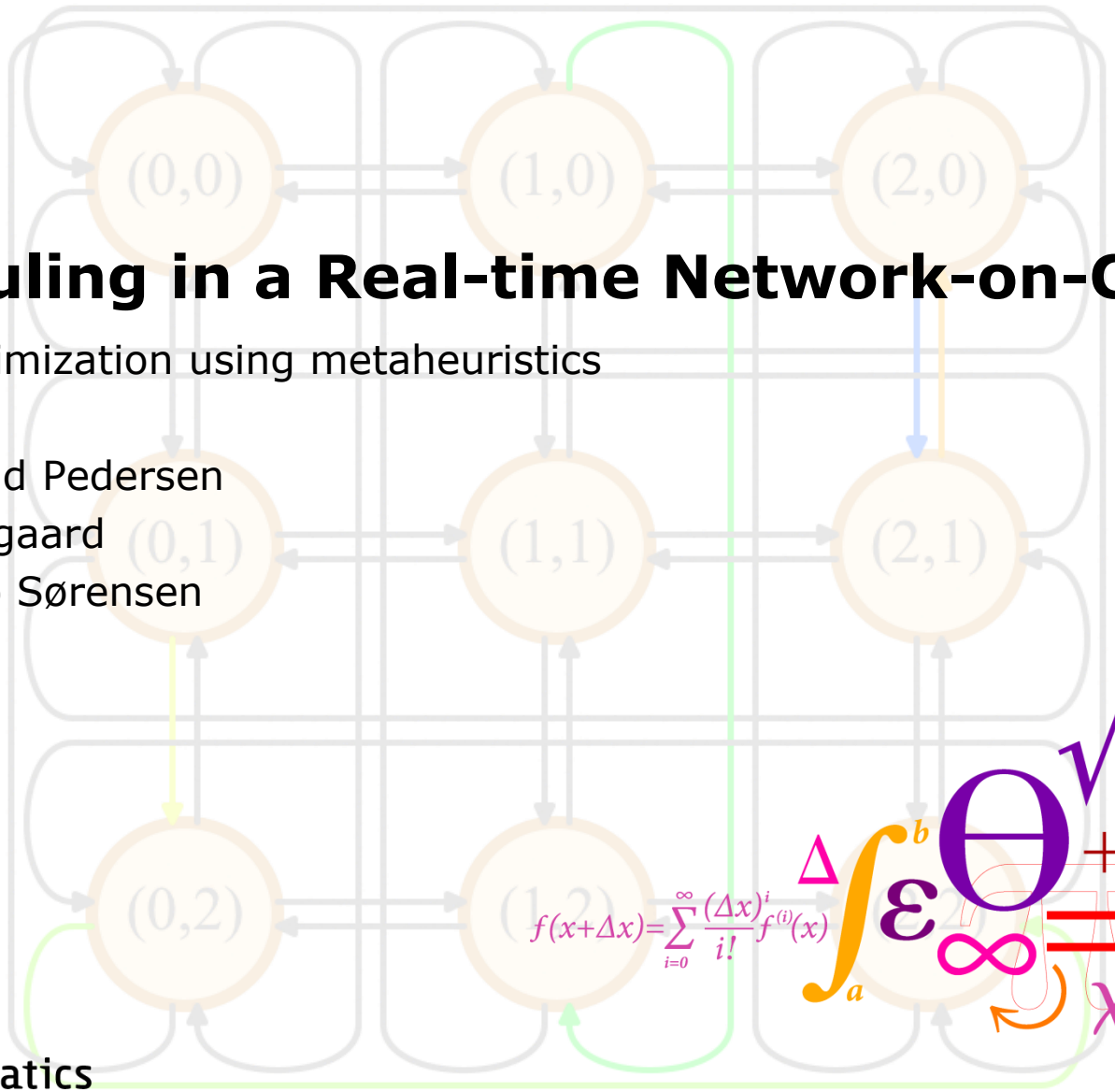
DTU

Scheduling in a Real-time Network-on-Chip

Period minimization using metaheuristics

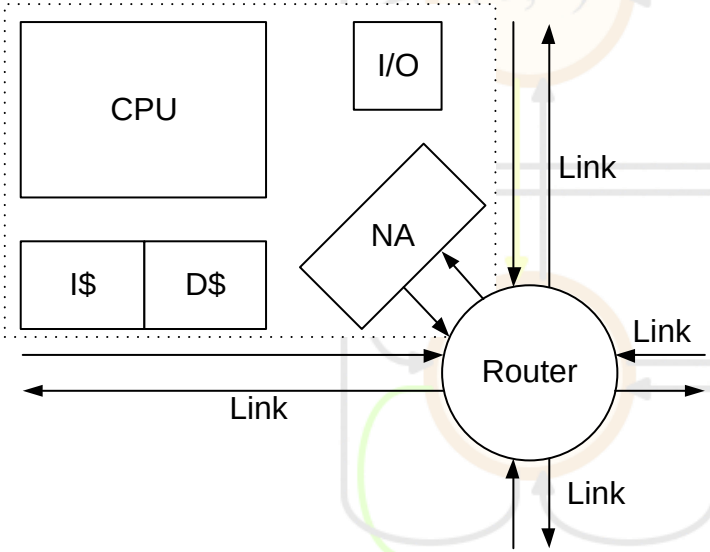
Mark Ruvald Pedersen
Jaspur Højgaard
Rasmus Bo Sørensen

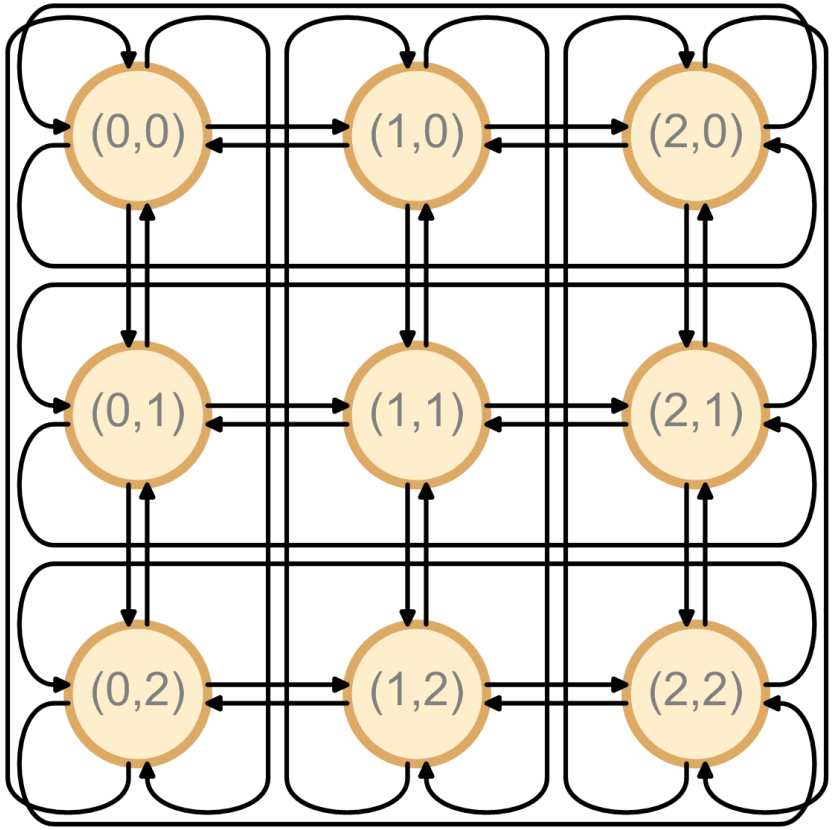


DTU Informatics
Department of Informatics and Mathematical Modeling

Problem domain

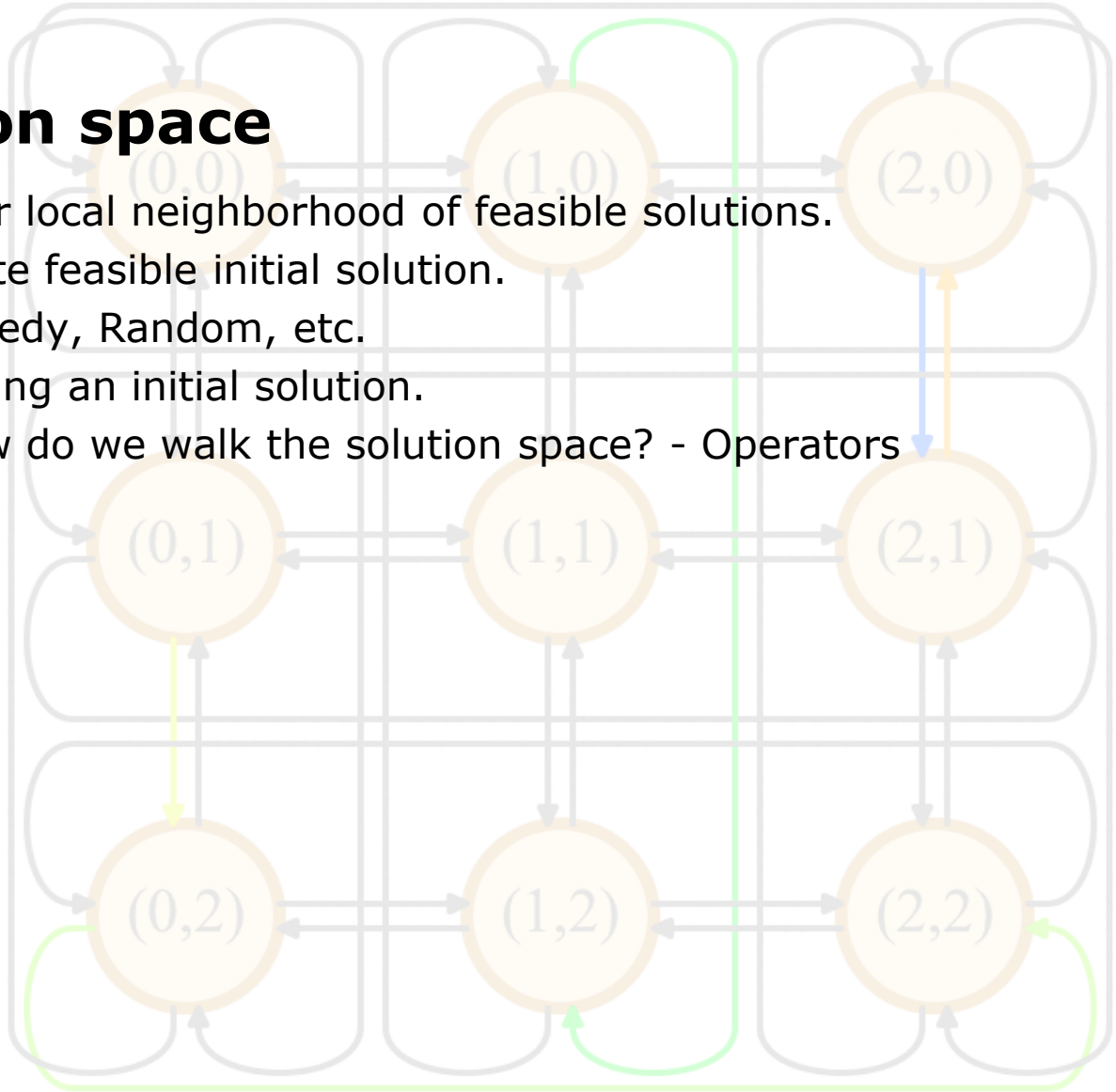
- NP-complete: Integer multi-commodity flow problem
- No buffering
- Real-time system
- Network-on-Chip
- Inter-processor communication
- Scheduling communication





Solution space

- No clear local neighborhood of feasible solutions.
- Generate feasible initial solution.
  - Greedy, Random, etc.
- Improving an initial solution.
  - How do we walk the solution space? - Operators



Operators

Destroy and rebuild paths.

Operators for destroy:

- Dominating paths
- Dominating rectangle
- Late paths
- Random

Operators are chosen adaptively.

Rebuild:

- Greedy randomized first-fit.

Metaheuristics

Greedy randomized search procedure (GRASP)

```
procedure our_grasp(β)
     best = infinite
     while (time left)
        current = initial_solution(β)
        if (current shorter than best) then best = current
        operator = choose_operator()
        chosen = operator()
        destroy(chosen)
        repair(chosen)
        if (current shorter than best) then best = current
        punish_reward(operator)

return best
```

Local search

- Local search of GRASP is choose_operator, destroy and repair.
- Random-operator not used.
- Problems with no clear local neighborhood.
- Very large solution spaces.

Metaheuristics

Adaptive Large Neighborhood Search (ALNS)

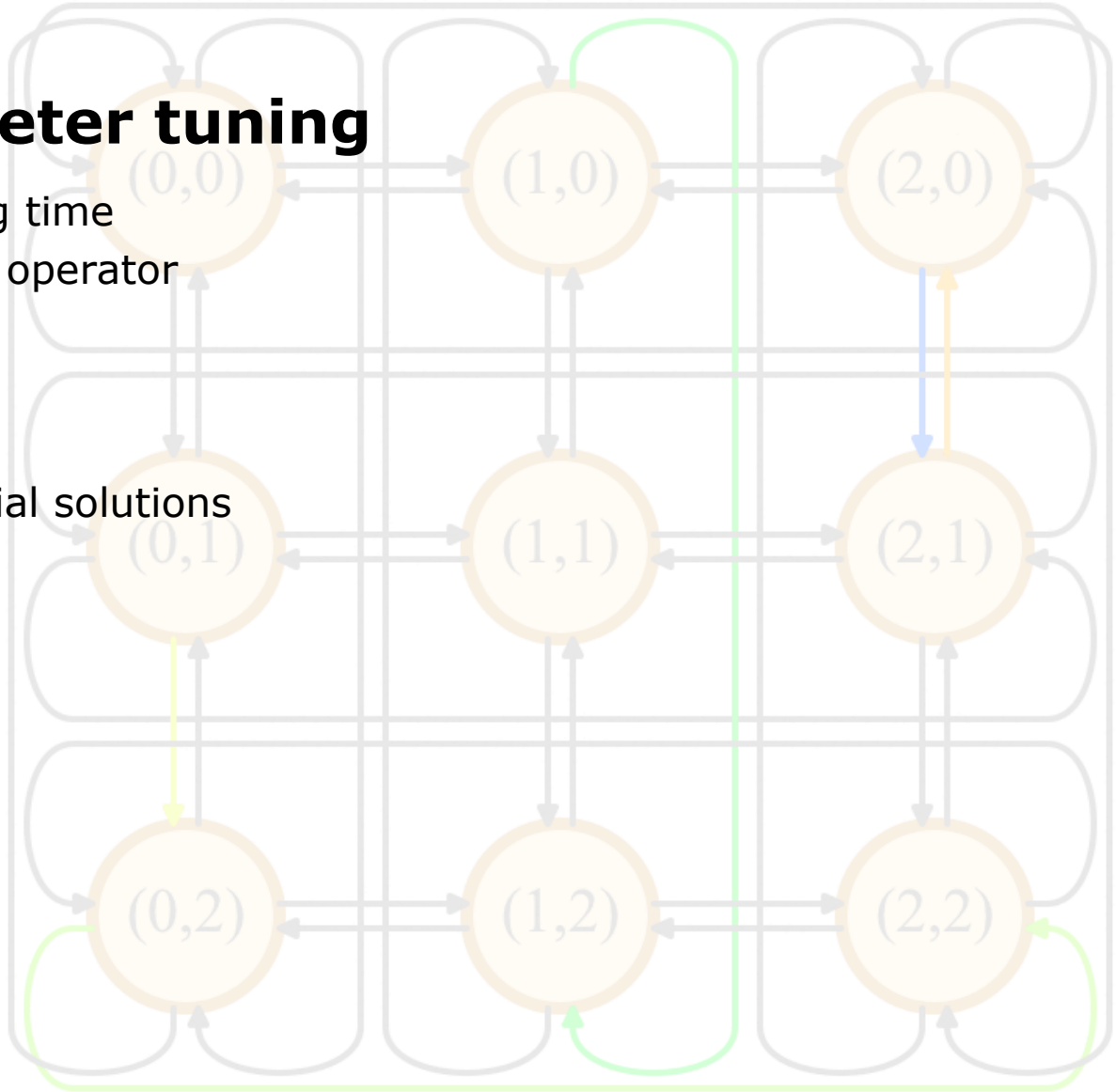
```
procedure our_alns()
     best = current = initial_solution()
     while (time left)
        operator = choose_operator()
        chosen = operator()
        destroy(chosen)
        repair(chosen)
        if (current shorter than best) then best = current
        punish_reward(operator)

return best
```

- Finds feasible solution easily.
- Destroy and repair gives very large neighborhood.
- Iteratively tries shortening the schedule.
- Escaping faster from local minima: All new solutions are accepted.

Parameter tuning

- Running time
- Choose operator
- GRASP
  - β
- ALNS
  - Initial solutions



Results

Overall best metaheuristic GRASP.

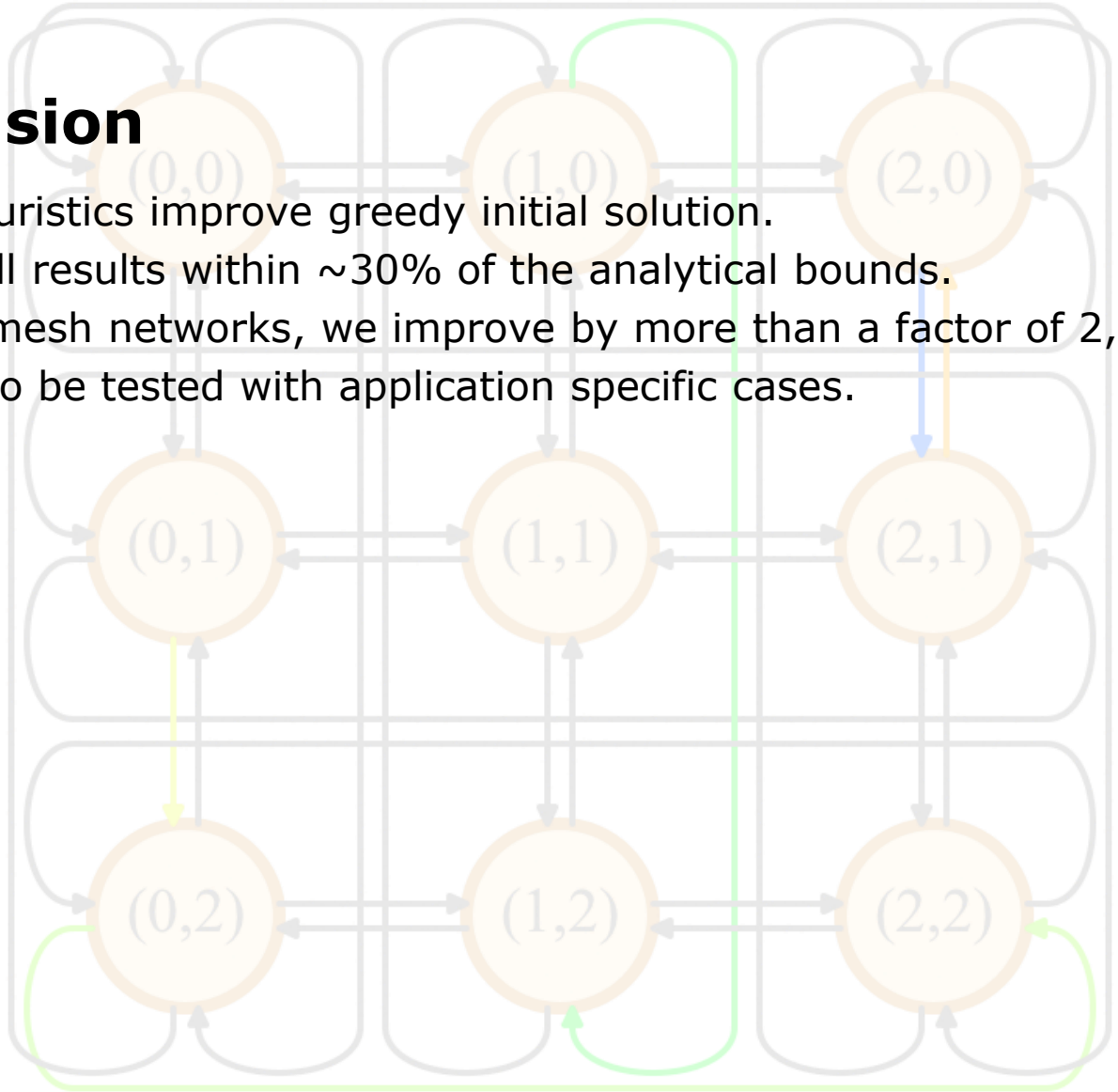
Low relatedness between neighboring solutions. -> GRASP

Size	Mesh					Bi-torus				
	Bounds[1]	[2]	GREEDY	ALNS	GRASP	Bounds[1]	[2]	GREEDY	ALNS	GRASP
3×3	8 (10)	28	13	**11**	**11**	8 (10)	11	12	**10**	**10**
4×4	16 (18)	59	24	**21**	**21**	15 (18)	20	21	**19**	**19**
5×5	25 (34)	112	41	39	**37**	24 (28)	**28**	32	30	30
6×6	54	–	66	65	**61**	35	–	45	45	**43**
7×7	66	–	98	97	**94**	48	–	64	63	**61**
8×8	128	481	144	144	**138**	64	88	87	86	**85**
9×9	135	–	201	201	**195**	90	–	**113**	**113**	**113**
10×10	250	974	271	271	**267**	125	158	154	153	**151**
15×15	600	3467	**886**	**886**	899	420	481	**471**	**471**	474

Table 4: Results compared to the heuristic results of [2]. Numbers in parenthesis are optimal schedule periods

Conclusion

- Metaheuristics improve greedy initial solution.
- All-to-all results within ~30% of the analytical bounds.
- For all mesh networks, we improve by more than a factor of 2,0.
- Needs to be tested with application specific cases.



Scheduler output