AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

A Process-Oriented Deadlock Recovery Policy for Flexible Manufacturing Systems

Dr hab. inż. Andrei Karatkevich, prof. AGH
Katedra Informatyki Stosowanej

Seminarium Katedry Informatyki Stosowanej, 30 grudnia 2022

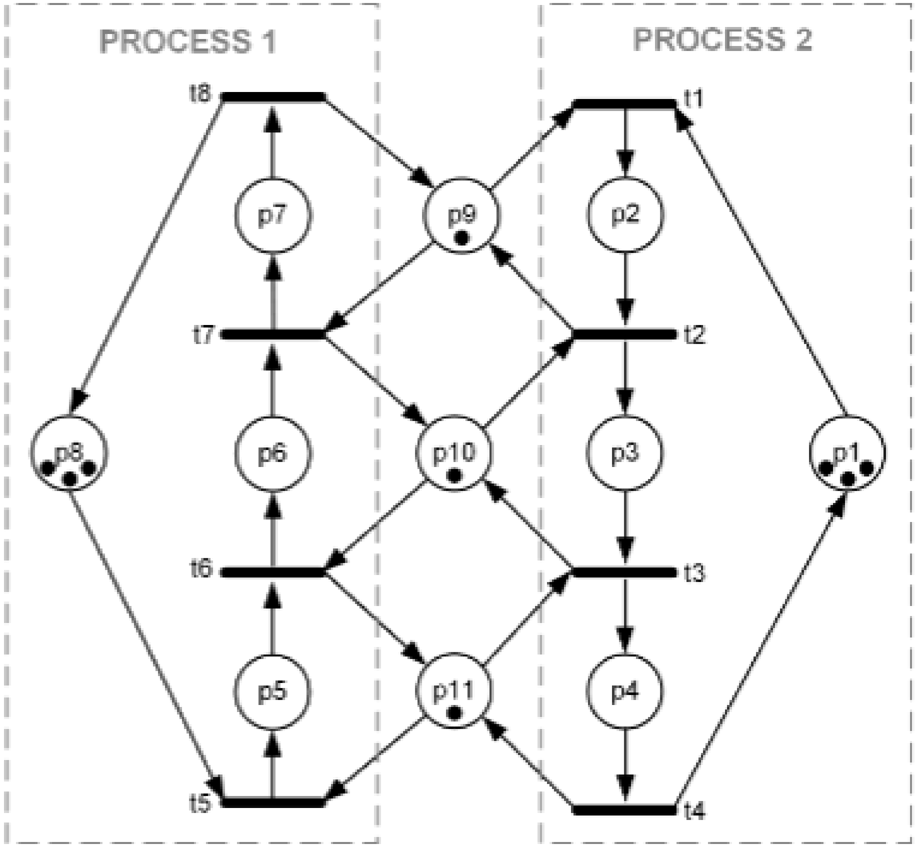
Agenda

- Petri nets for modeling the production systems (S^3PR nets)
- Dealing with the deadlocks
- A new algorithm of deadlock recovery
- Formal results
- Cycles in dependency graphs
- An idea of heuristic method

Petri nets for modeling the manufacturing systems

- A popular model (introduced by J. Ezpeleta, J. M. Colom and J. Martinez): S^3PR
- S^3PR is a composition of sequential processes (SM-nets) with shared resources added
- 3 kinds of places: activity places, idle places and resource places
- One idle place for a process
- Initially only idle and resource places are marked

Petri nets for modeling the manufacturing systems: S^3PR

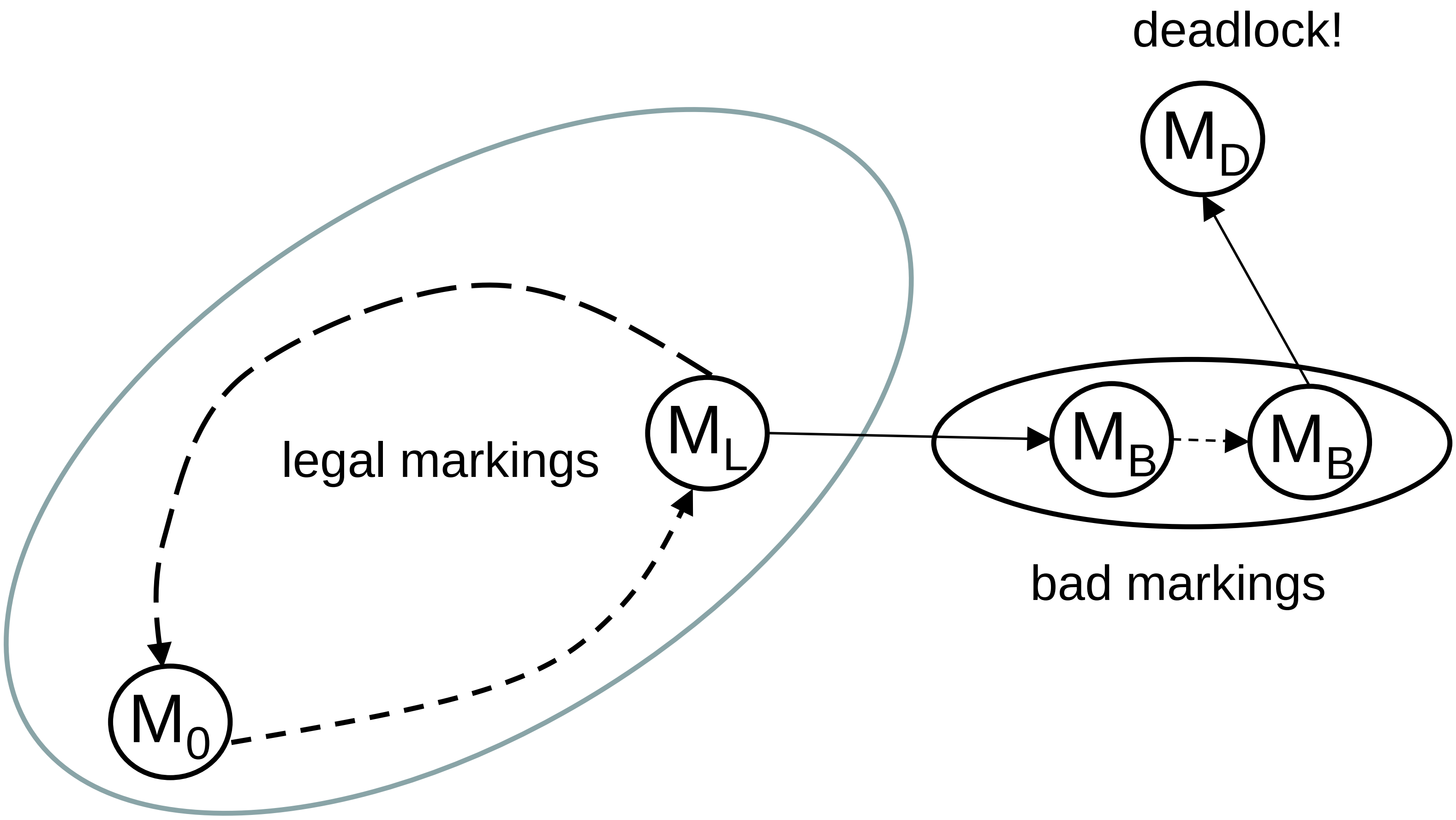


Deadlocks: how to deal with them?

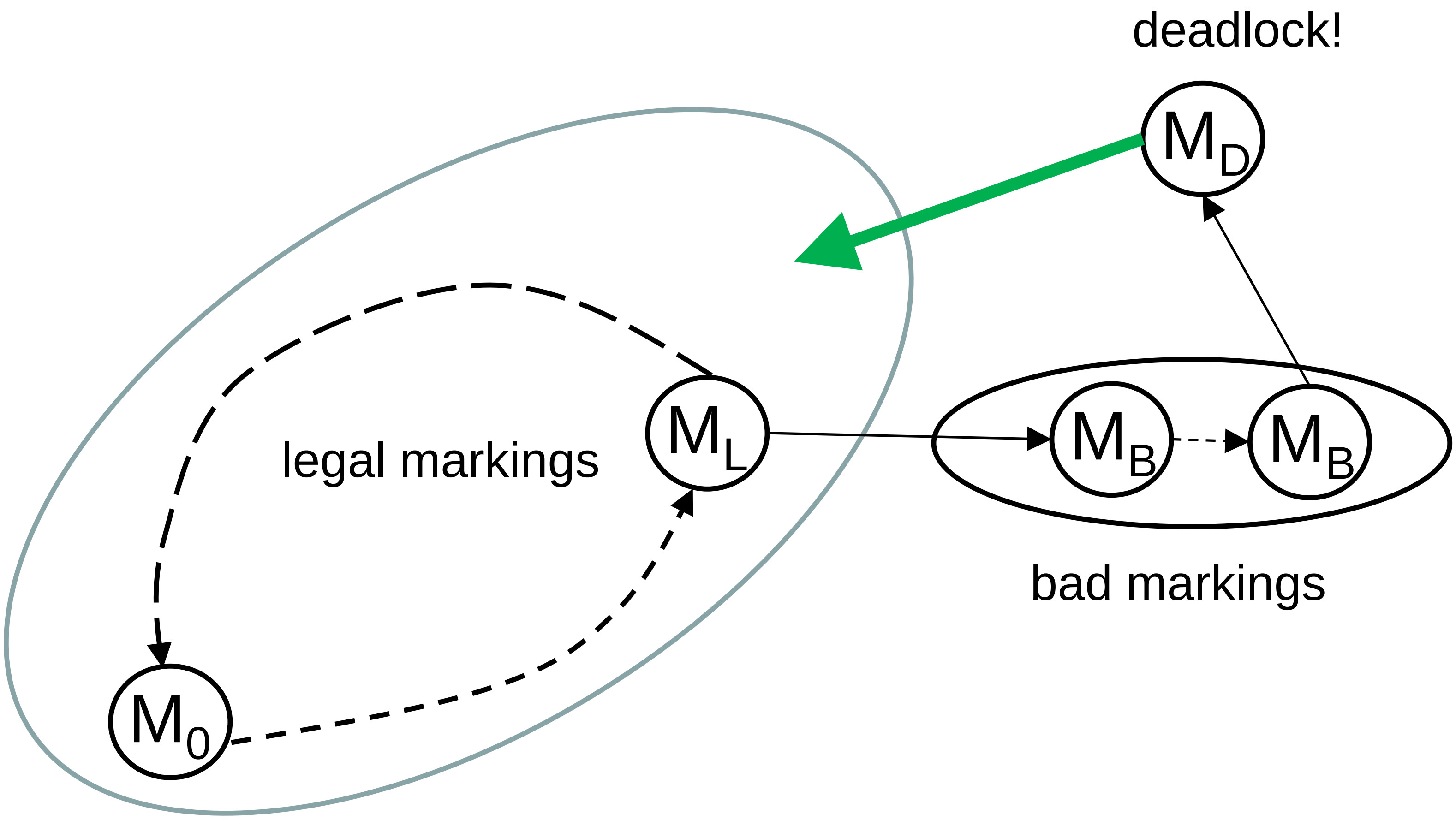
Three main approaches to manage the deadlocks:

- Deadlock prevention:
  - adding to the net the „monitors” in such a way that the deadlocks cannot happen
- Deadlock avoiding:
  - providing a policy which forbids firing of certain transitions in certain states, avoiding entering the „bad” markings
- Deadlock recovery:
  - adding to the net the „recovery transitions” allowing to leave the deadlocks and providing liveness

Legal, bad and deadlock markings



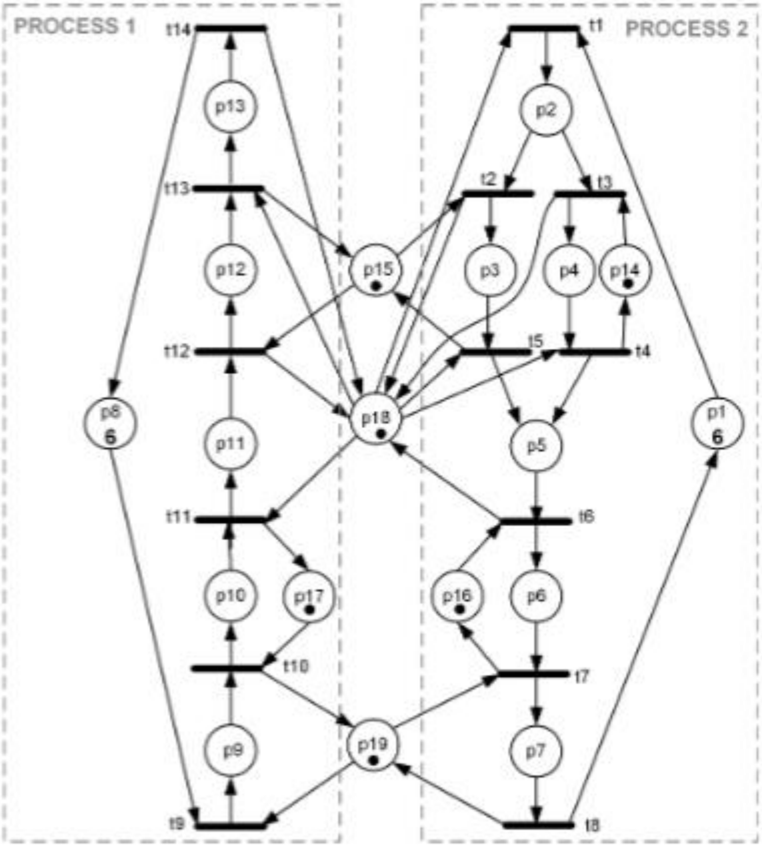
Deadlock recovery



Deadlock recovery

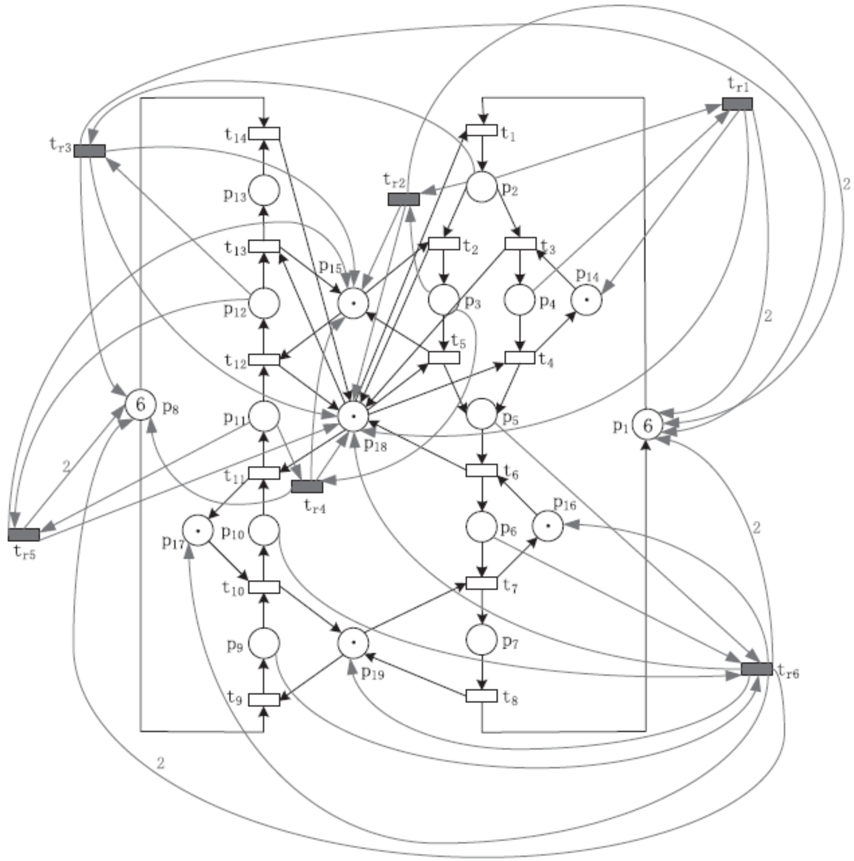
- A lot of methods
- Usually requiring constructing of the reachability graphs
- Sometimes considering every pair (deadlock marking - legal marking)
- Finding a set of transitions leading from a deadlock to a legal marking (usually *any* one)
- Tending to minimize number of recovery transitions and/or number of arcs

An S³PR net (a known example)



282 reachable markings, 16 deadlock markings

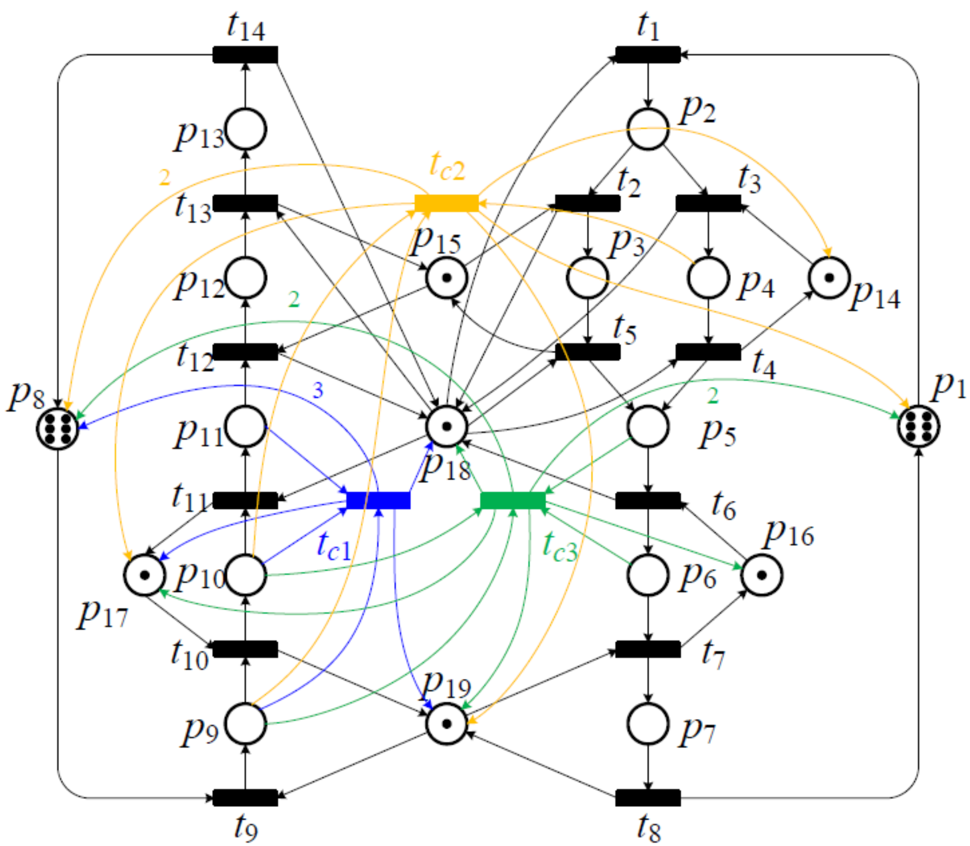
Deadlock recovery (examples)



Recovery:
6 transitions,
36 arcs

Y. Lu, Y. Chen, Z. Li and N. Wu, ”An Efficient Method of Deadlock Detection and Recovery for Flexible Manufacturing Systems by Resource Flow Graphs”, IEEE Trans. Autom. Sci. Eng., 2021

Deadlock recovery (examples)



Recovery:
3 transitions,
25 arcs

Y. L. Pan, ”One Computational Innovation Transition-Based Recovery Policy for Flexible Manufacturing Systems Using Petri nets”, Appl. Sci., vol. 10, no. 7, 2332, 2020.

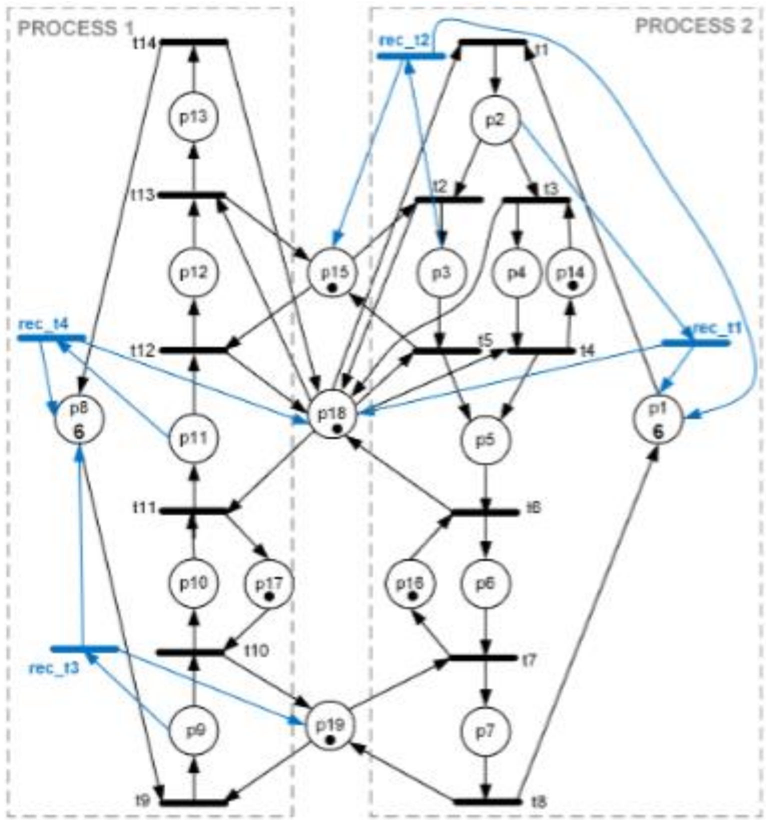
Idea of the proposed approach

- Shifting from a deadlock to *any* legal marking may have no sense for the modelled system
- A state of an S^3PR Petri net can be understood as a combination of proces instances
- Resetting of a process instance seems to have a sense for a manufacturing system
- We are looking for resetting of minimal numer of process instances for every deadlock
- A transition resetting a process instance always has exactly 3 arcs

Steps of the algorithm (without details)

- Constructing the reachabiliry graph (alas…)
- Identifying legal, bad and deadlock markings
- For every deadlock finding the minimal sets of proces instances resetting of which leads to the legal markings
- Selecting such sets that the whole number of recovery transitions is minimized

Deadlock recovery by the proposed algorithm



Recovery:
4 transitions,
12 arcs

How is it possible?

- How the result can be simpler than provided by the „exact methods”?
- Because the known methods recover with single transition, and in our method recovery may be by a firing sequence

Correctness of the algorithm
(formal results)

Proposition 1. Every reachable deadlock of an S^3PR net can be changed into a legal marking by means of resetting a subset of active process instances.

Proposition 2. Any reachable marking of an S^3PR net with only one active process instance is not a deadlock.

Corollary 1. For any deadlock marking of an S^3PR net there exists a proper subset of the active process instances such that resetting them changes the marking into a legal one.

Formal results (cd…)

Proposition 3. Let *N* be an S^3PR net, let *M* be a marking reachable in it, let *M′* be a marking obtained from *M* by resetting a process instance, removing a token from an activity place *p*; let *t* be a transition enabled at *M* such that $p \notin {}^{\bullet}t$, Then *t* is enabled at *M′*.

Proposition 4. Let *N* be an S^3PR net, let *M* be a marking reachable in it, let *M′* be a marking obtained from *M* by resetting a process instance. Then *M′* is reachable in *N*.

Corollary 2. Resetting of a process instance does not change the set of reachable markings.

Formal results (cd…)

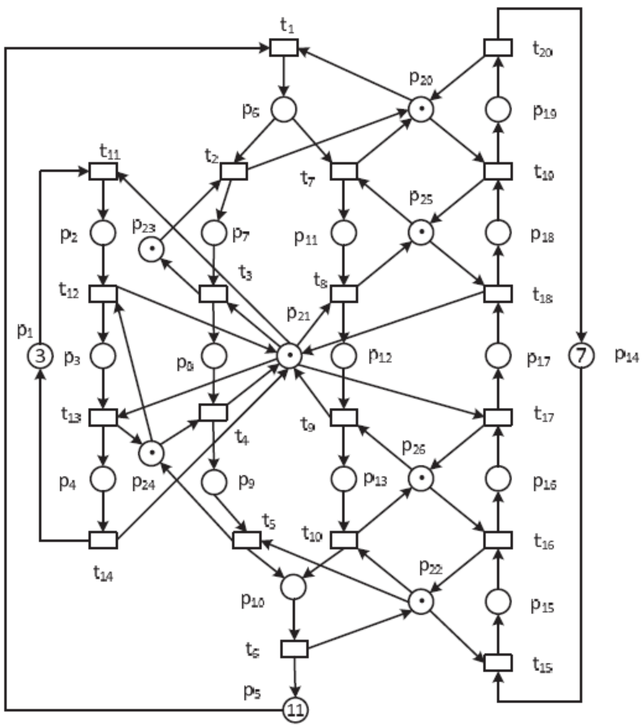
Proposition 5. Let *N* be an S^{3}PR net, let *M* be a legal marking reachable in it, let *M′* be a marking obtained from *M* by resetting a process instance. Then *M′* is a legal marking.

Corollary 3. Resetting of a proces instance cannot change a legal marking into an illegal one.

Formal results (cd…)

Proposition 6. Let N be an S^3PR net, let M_d and M'_d be two deadlock markings reachable in it, such that $\forall p \in P_A$: $(M'_d(p) = 1) \Rightarrow (M_d(p) = 1)$, $\exists p \in P_A$: $M_d(p) = 1$, $M'_d(p) = 0$. Let A and A' be the sets of activity places with tokens at M_d and M'_d, correspondingly. Let $Q \subset A$ be the set of activity places such that resetting of the corresponding process instances changes M_d into a legal marking M. Then resetting of the process instances corresponding to the activity places belonging to the set $Q \cap A'$ changes M'_d into a legal marking M'.

A bigger example



1650 reachable markings, 998 legal markings
and 24 deadlock markings

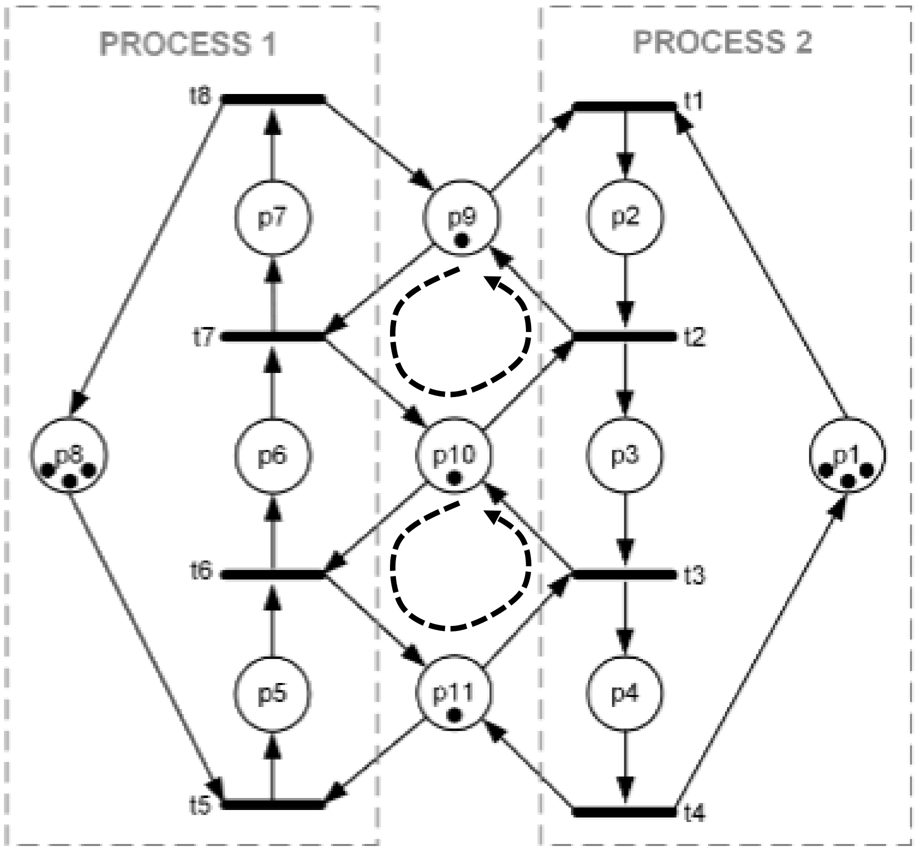
Recovery:
Source paper -
4 transitions, 48 arcs.
Our method –
6 transitions, 18 arcs

Y. Dong, Y. Chen, S. Li, M. A. El-Meligy and M. Sharaf, ”An efficient deadlock recovery policy for flexible manufacturing systems modeled with Petri nets”, IEEE Access, vol. 7, pp. 11785-11795, 2019.

Problem with the existing methods: computational complexity

- Most of the methods (including ours) require constructing of the reachability graph
- Some of the methods of deadlock prevention or avoiding require detecting syphons or cycles in the dependency graphs – their number may exponentially depend on the net size
- A quick heuristic method would be useful

Dependency graphs in S^3PR nets



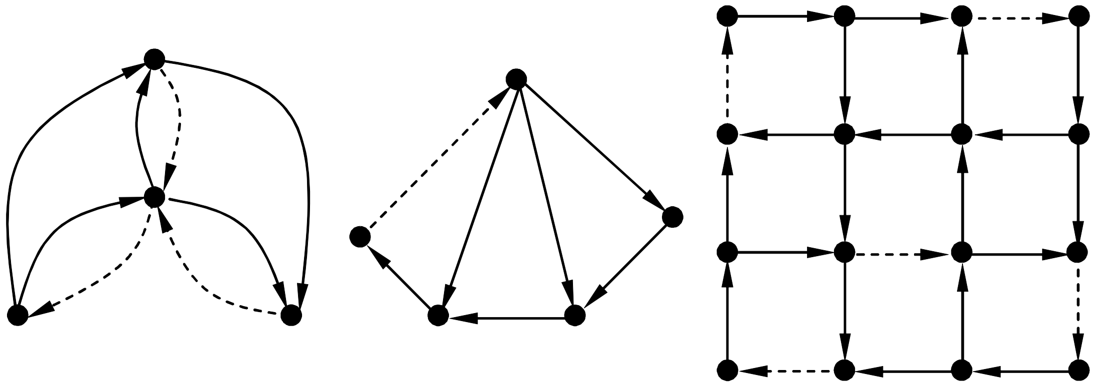
- Can be directly obtained from the net graphs – consist of the resource places and the transitions
- At least one cycle in the dependency graph corresponds to every deadlock
- The recovery transitions should break such cycles

Dependency graphs in S^3PR nets

- Can be used for deadlock prevention. How to use for deadlock recovery without creating the livelocks?
- Still exponential computational complexity

Idea of the heuristic approach

- Breaking the cycles by resetting the proces instances
- Applying a heuristic for „decyclization” (not requiring constructing of all the elementary cycles)



Summary

- A new algorithm of deadlock recovery in the S^3PR nets is proposed (proces-oriented)
- A quick heuristic algorithm of deadlock recovery is proposed
- Besides, there is an idea on deadlock detection in the S^3PR nets…

Publikacje

- Grobelna I., Karatkevich A. *A deadlock recovery policy for flexible manufacturing systems with minimized traversing within Reachability Graph*// Proceedings of IEEE International Symposium INFOTEH, Jahorina, 2022, pp. 1-6 (electronic edition)
- Karatkevich A., Grobelna I. *A Process-Oriented Deadlock Recovery Policy for Flexible Manufacturing Systems*// ICARCV 2022, Singapore – accepted (140 pkt)

- Grobelna I., Karatkevich A. *A Quick Heuristic Algorithm for Enforcing Liveness of S^3PR Petri Nets* - under construction…
- Karatkevich A., Grobelna I. *Deadlock Recovery Policies for Manufacturing Systems using Reachability Graph Traversal* - under construction…

Dziękuję za uwagę!