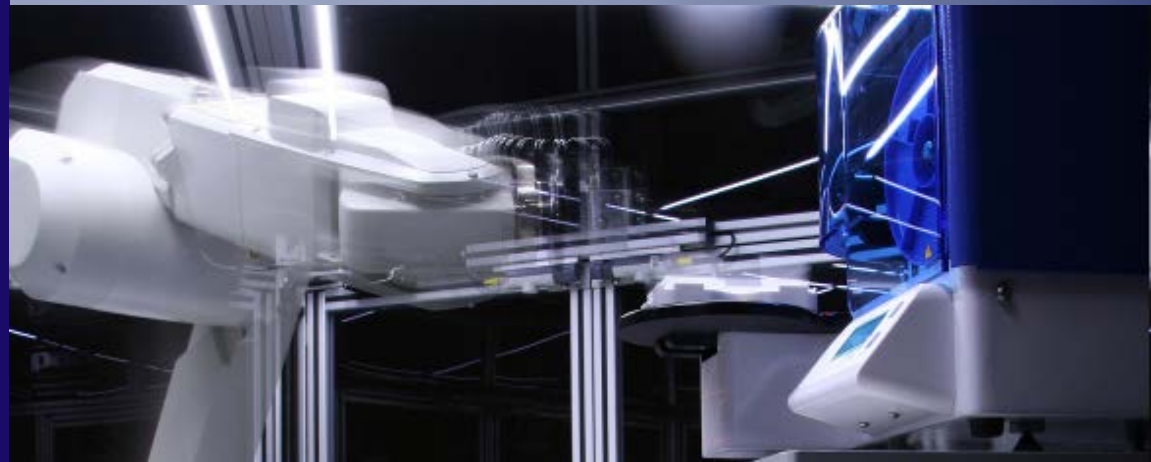


The logo for PAA (Peak Analysis & Automation) features the lowercase letters 'paa' in a bold, sans-serif font, enclosed within a rounded rectangular border.

**peak analysis
& automation**

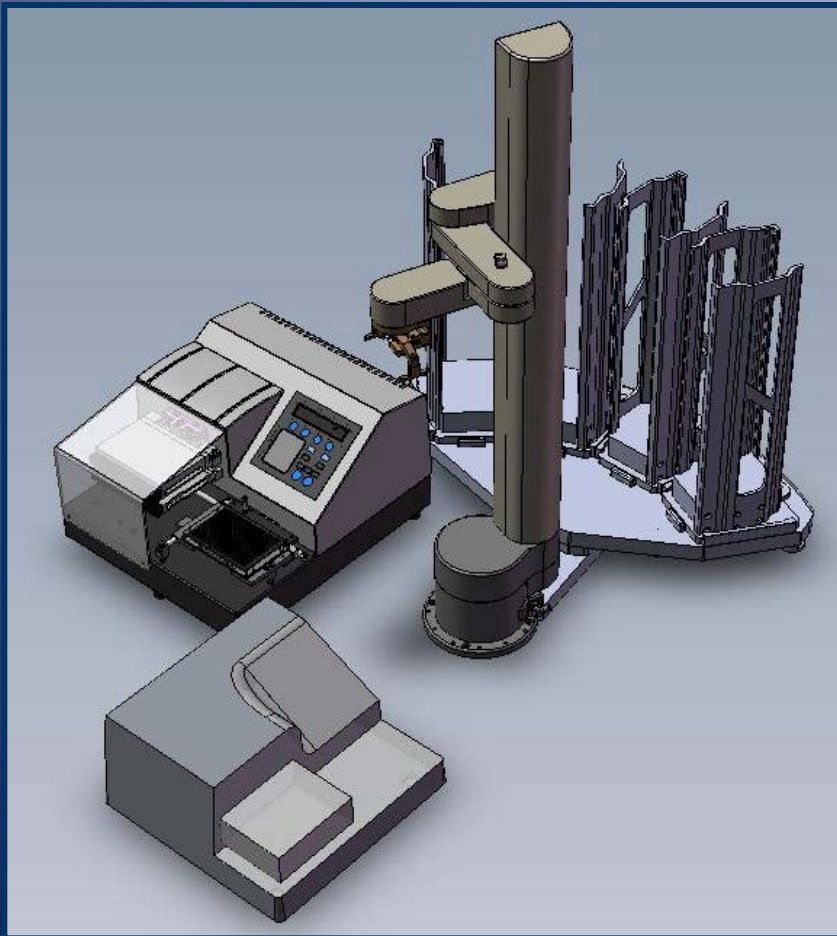
first in laboratory automation

Solutions for Scheduling Assays



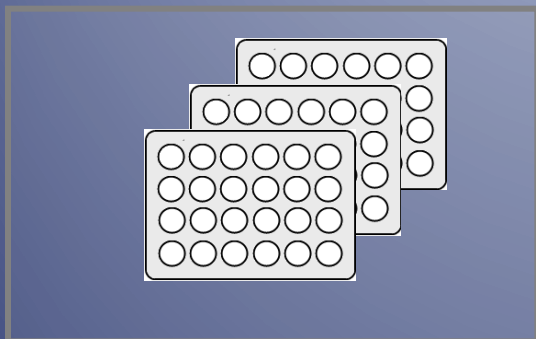
**LRIG
Philadelphia
April 2015**

Scheduling example

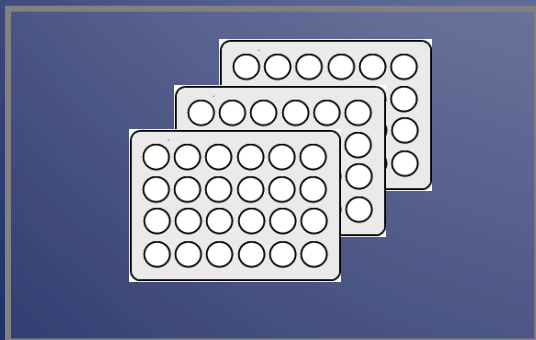


Task	Time(s)
Get plate from Stack and move to Washer	15
Run Washer	60
Move plate from Washer to Dispenser	10
Run Dispenser	40

Input



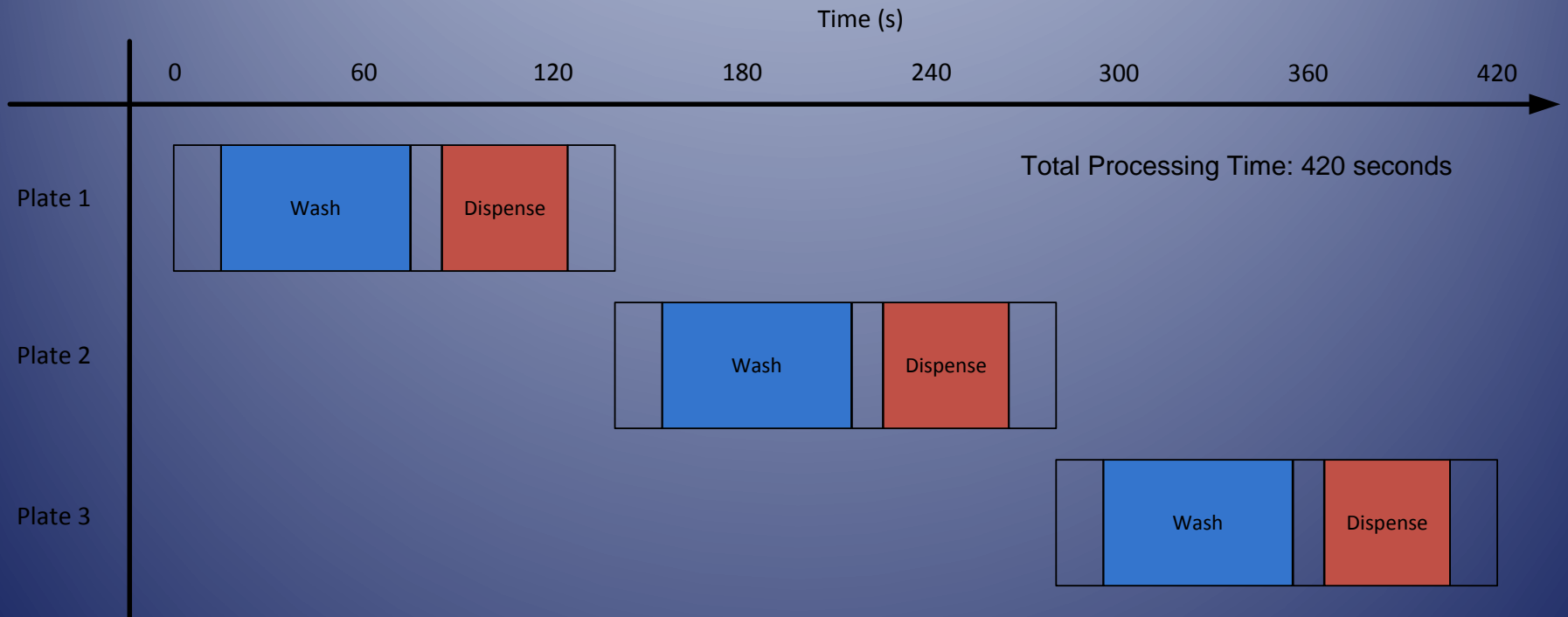
Output



Running - Wash

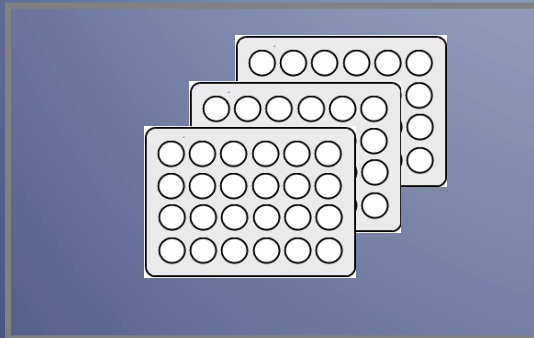


Running - Dispense

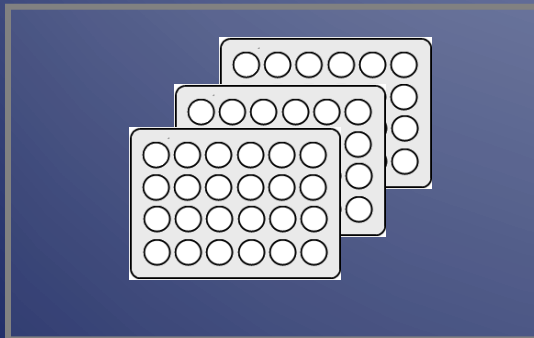


- inefficient
 - time
 - resources

Input



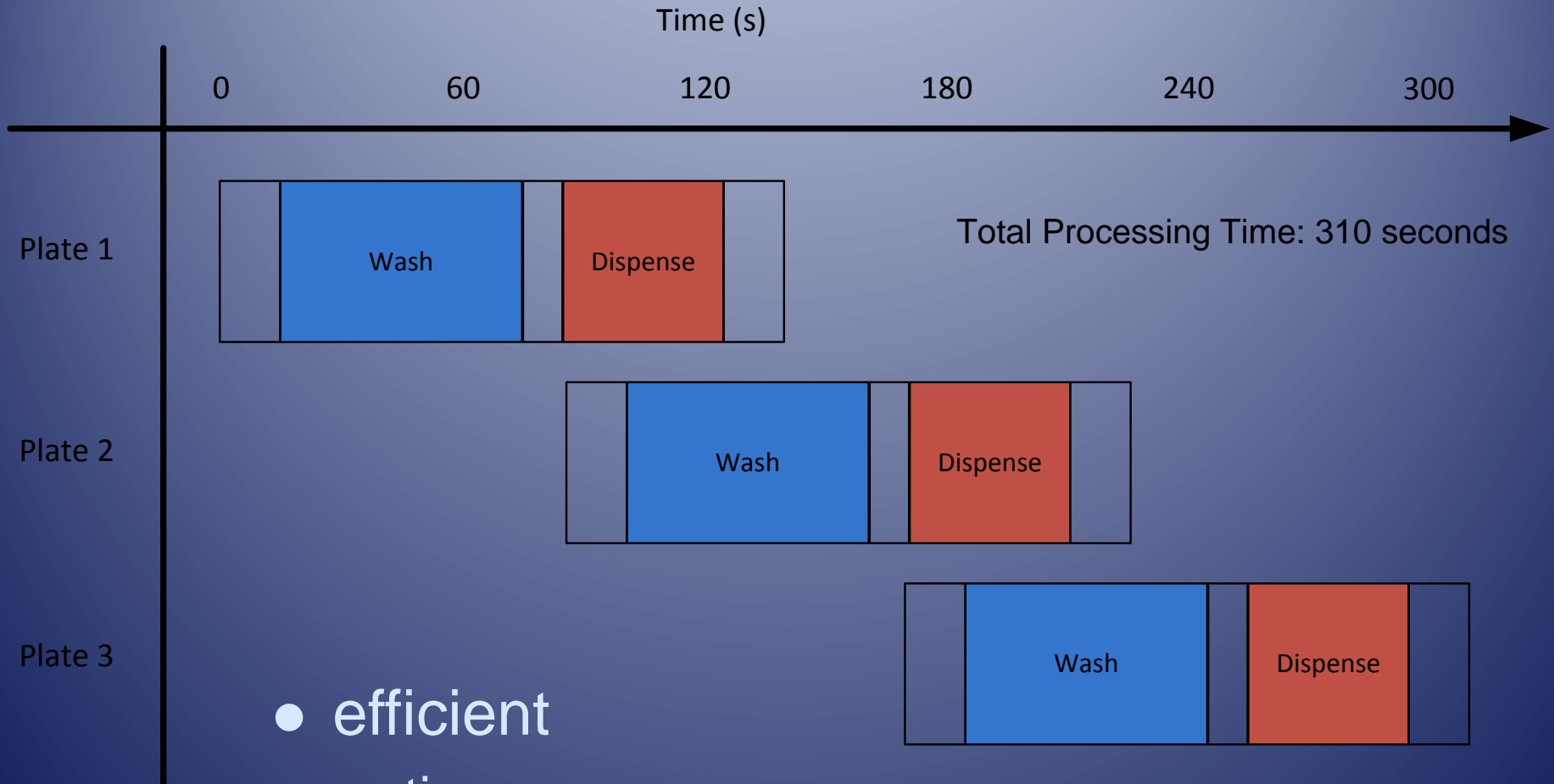
Output



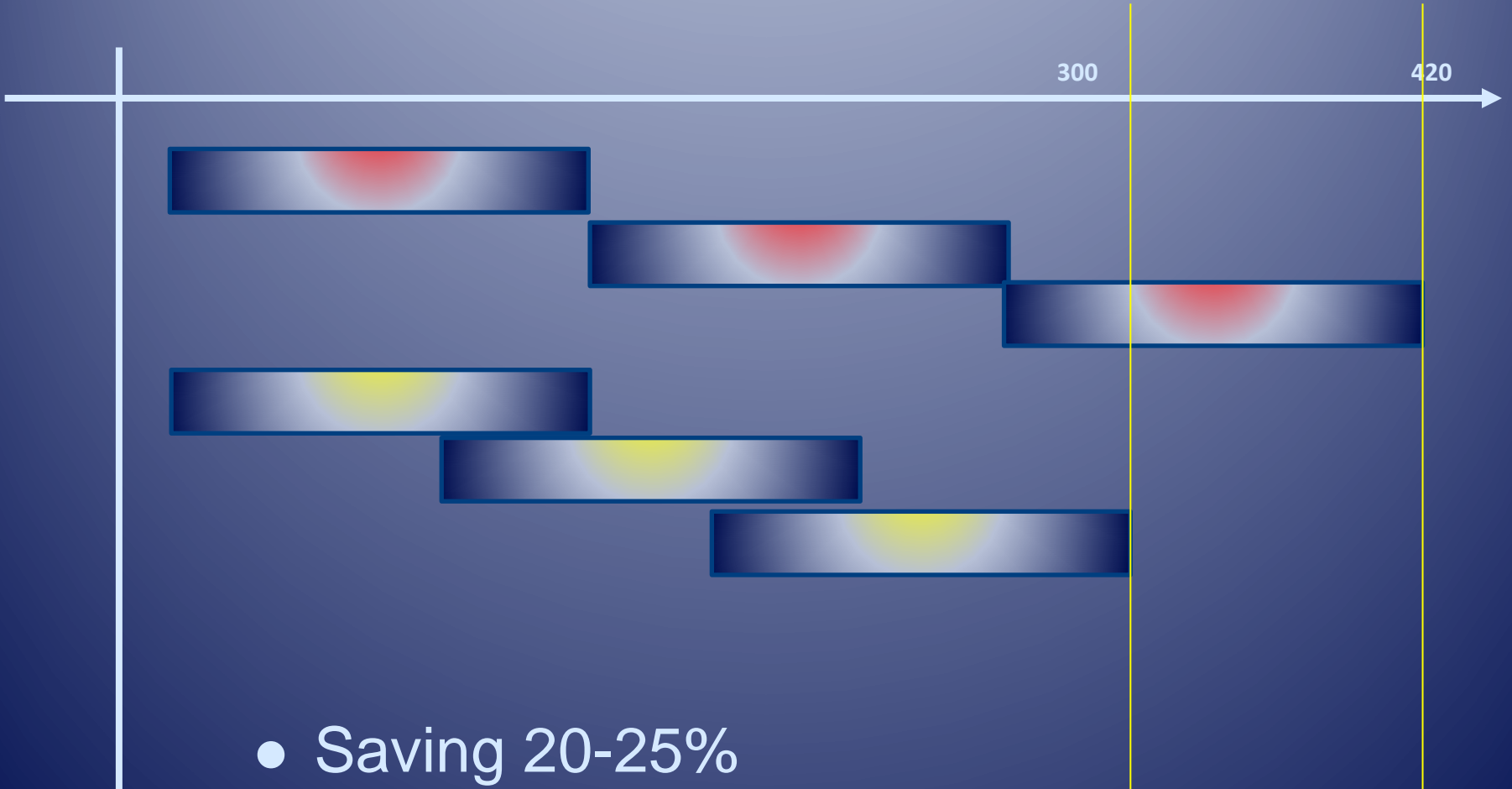
Running - Wash



Running - Dispense



- efficient
 - time
 - resources



- Saving 20-25%

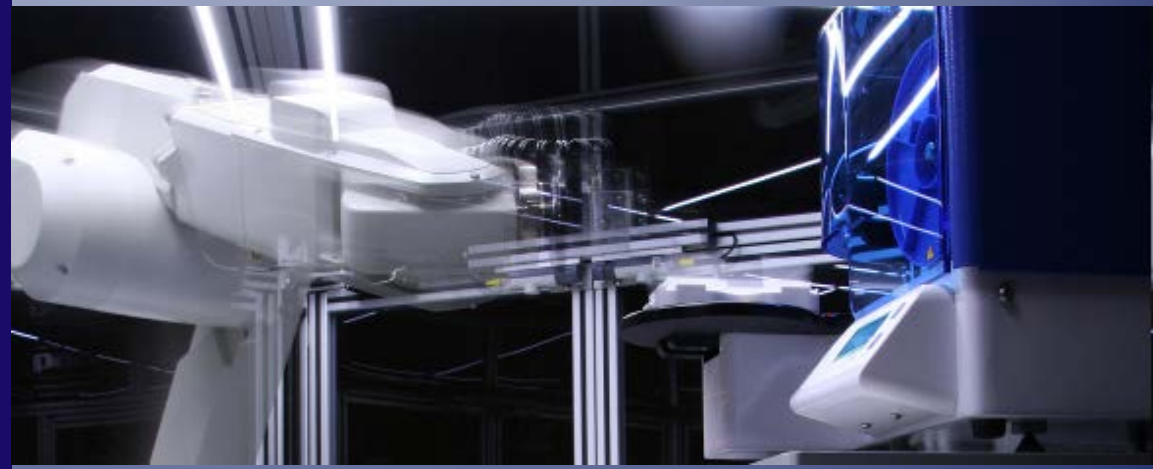
- improved throughput by efficient scheduling
- scheduling is optimally processing tasks
- objectives include
 - minimizing time
 - maximizing use of the rate determining resource
- constraints (time & resources) must be obeyed



**peak analysis
& automation**

first in laboratory automation

scheduling types

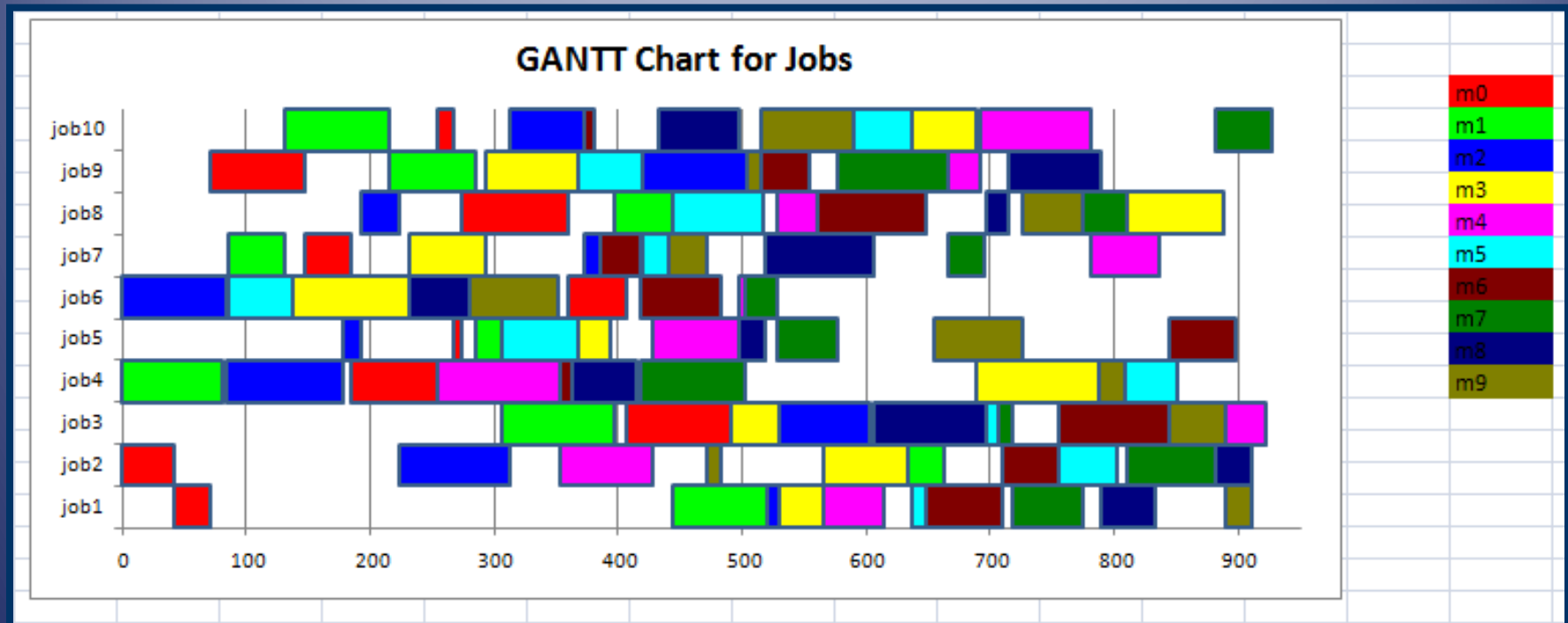


- prepares pre-emptive plan of plates processing
- mathematical approach
 - calculating optimal schedule
 - exact or heuristic
- complex algorithms evaluate all the permutations
- problem in laboratory automation is similar to the

Job Shop Scheduling problem

- fixed set of n jobs
- each job consists of a chain of operations
- fixed set of m machines that each operation uses
- each machine can handle at one operation at a time
- each operation needs to be processed during an uninterrupted time period of a given length on a given machine
- aim is to find a schedule that has minimal length

- schedules required for this problem
 - are the most difficult to solve
- problem is non-deterministic polynomial time
 - hard mathematically
- 10 operations require
 - 3,628,800 scheduling permutations to be analyzed



10 Job, 10 machine problem
 An MIP solver takes 5 minutes to solve this

- problem can be simplified
- sequence of operations have to be in a set order
- operations for each job are the same
- cyclic scheduling methods can be adopted
- allowable time delays are introduced
 - plate cycle time increases
 - throughput increases as well

- Mathematical Solvers
- Exact methods
- Local search methods

- Linear
- Non-linear

- Branch and Bound
- Dynamic Programming
- Cutting plane

- Simulated annealing
- Tabu Search
- Adaptive Search

Many Solver tools available for solving scheduling problems

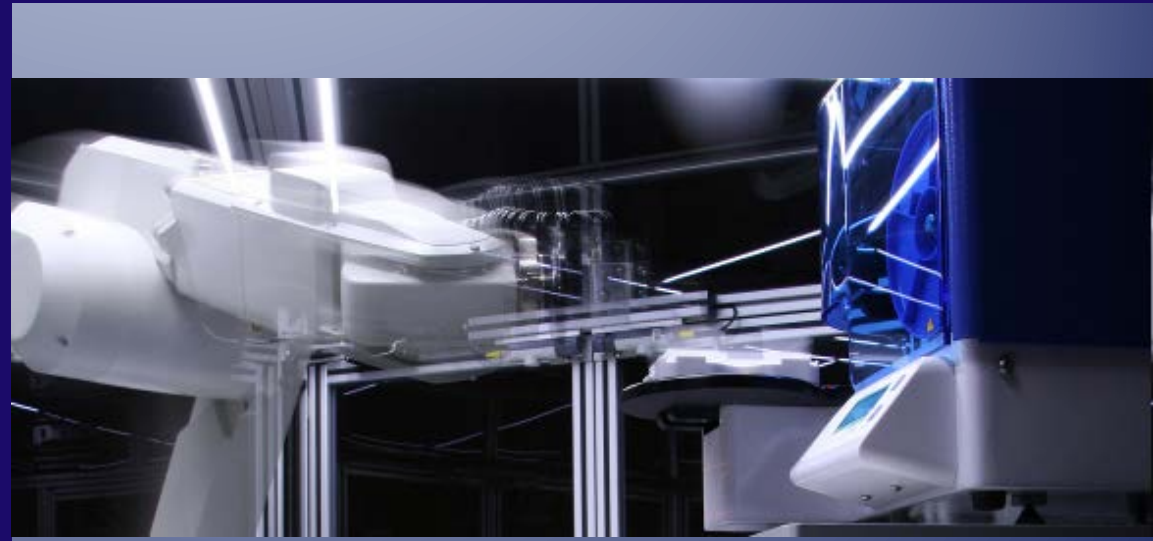
- The General Algebraic Modeling System (GAMS) – www.gams.com
- Microsoft Solver Foundation – www.solverfoundation.com
- IBM Optimization Subroutine Library – www.ibm.com/us



**peak analysis
& automation**

first in laboratory automation

**scheduler
types**



- each plate processed in exactly the same way
- incubation times run for the exact specified time
- allow the user to see how the assay will run
- see the effect of adding more instruments
 - pooling
- cannot react to events
- error states cannot be handled automatically

- schedule is determined before the run starts
- schedule is recalculated at runtime based on events
- rescheduling occurs when task durations differ
- true dynamic scheduling responds to all events
 - with deviation from current schedule

- method of running a process when events occur
 - device state change
- system is told how to process plates
 - based upon a set of rules

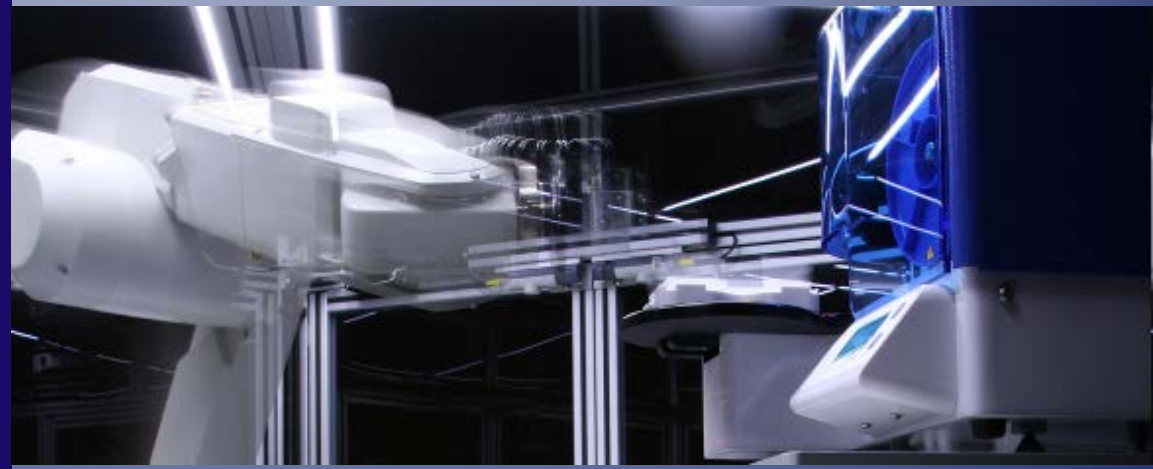
Stage	Activity	Rules
1	Load Plate onto Washer and start running	
2		
3		



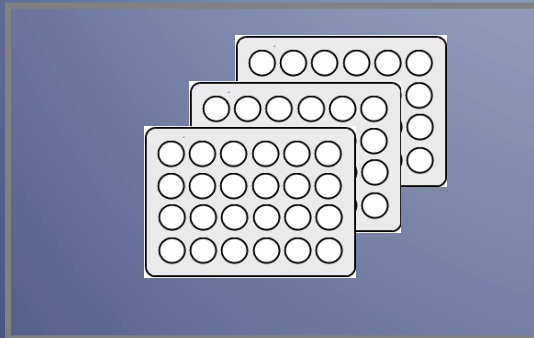
**peak analysis
& automation**

first in laboratory automation

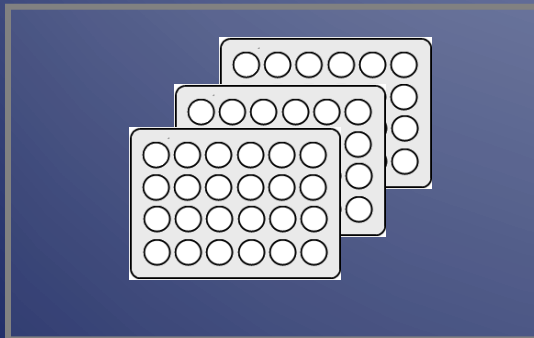
deadlock



Input



Output



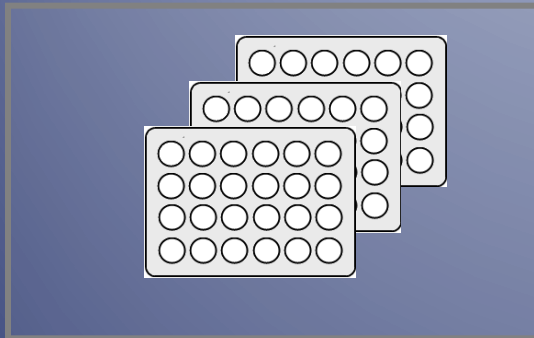
Running - Wash



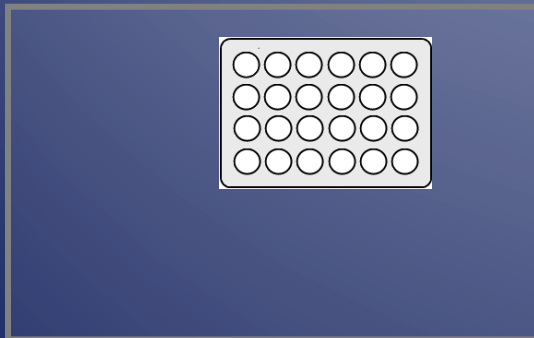
Running - Dispense

- Problem with all schedulers
- add additional step
 1. plate washer
 2. reagent addition
 3. plate washer

Input



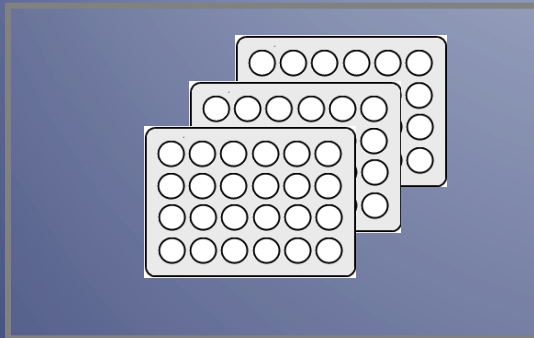
Output



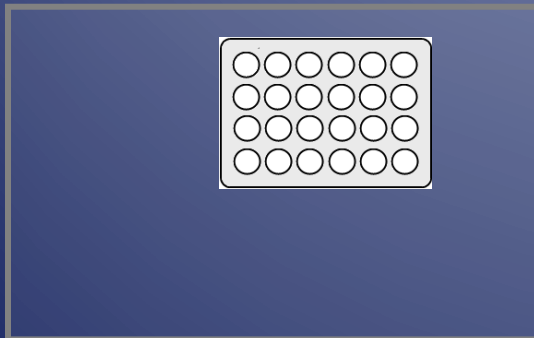
DEADLOCK!!

plates – multiple plates

Input

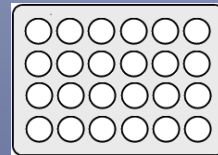
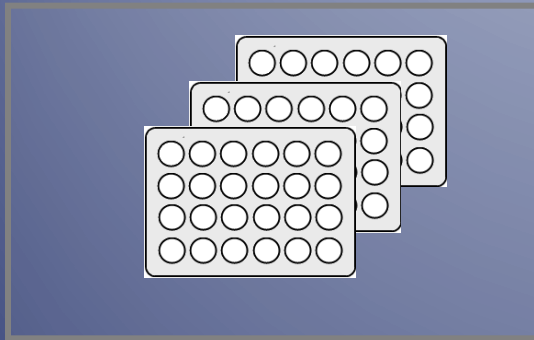


Output



- detect deadlocks by using Petri Nets
- control addition of new plates with pacing timers
 - slows the throughput
 - the degree of interleaving is reduced
- use buffers/relief nests

Input



Output

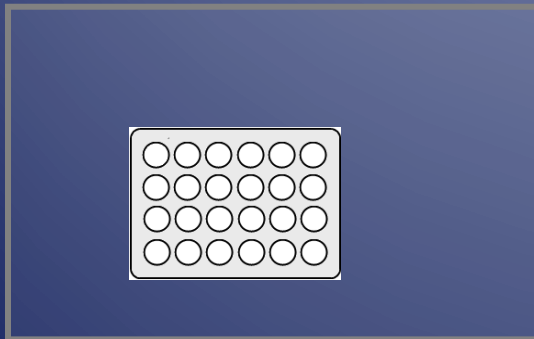


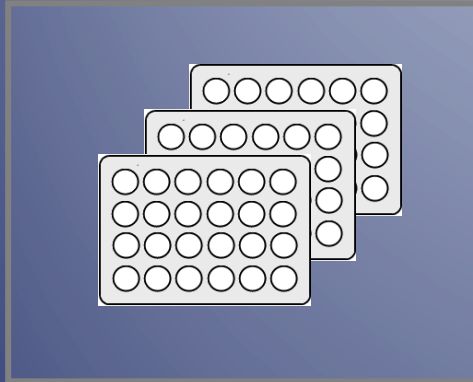
Plate nest



- no need for CPU intensive scheduling
- system can react to events as the run proceeds
- good for assays where process is unknown before the run starts
- deadlock can be determined and eliminated

Maximizing rate limiting resource utilization

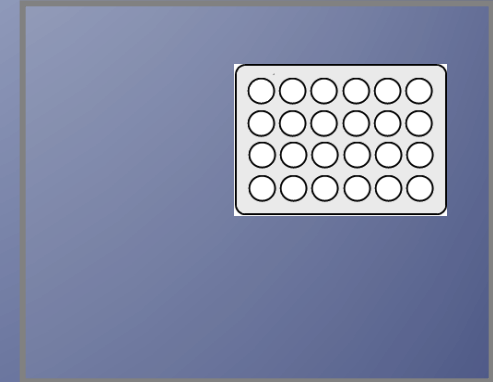
Input



Transferring ...



Output



delid

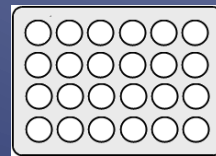


Plate
nest 1

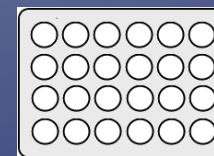


Plate
nest 2

relid

- different ways to schedule and process plates through an automated system
- choose the right scheduler for your application
- determine the rate limiting step in the process and maximize its utilization
- avoid deadlocks