SMMSO 2017

The 11th Conference on Stochastic Models of Manufacturing and Service Operations

June 4-9, 2017
Acaya, Italy

Decomposition via Column Generation for Control of Tandem Production Lines

ÖZGE BÜYÜKDAĞLI

EMRE NADAR

MURAT FADILOĞLU

yaşar university, izmir; bilkent university, ankara

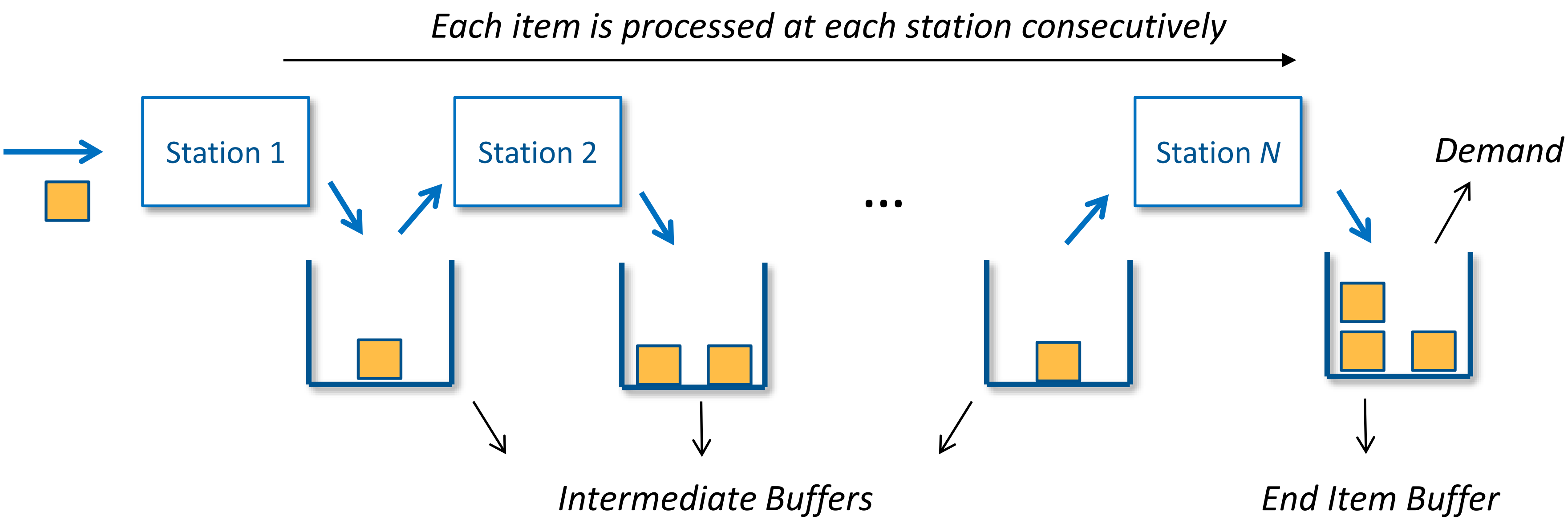
Content

- Tandem Production Lines
- Problem Formulation
- Column Generation Decomposition Approach
- Proposed Column Generation Rule
- Numerical Results

Tandem Production Lines

A production/inventory system with N workstations connected via intermediate buffers, terminating at a finished goods buffer.

- Exponential processing times, Poisson demand.



Modeling Framework

We consider the **production control** of this Markovian make-to-stock tandem production system.

- An MDP

At **decision** epochs:
Production decisions (continue or stop) at each station

Looking for:
The policies that **minimize long-run system cost**

Literature

Veatch and Wein (1994) : first to apply optimal control paradigm in two-station tandem lines.

Karaesmen and Dallery (2000) : studied the performance of available pull mechanisms in two-station tandem lines.

In this study, we propose a methodology based on the optimal control paradigm to characterize the best production policies in a way that

longer (more than two stations) lines, multi product, breakdowns and repairs can be analyzed

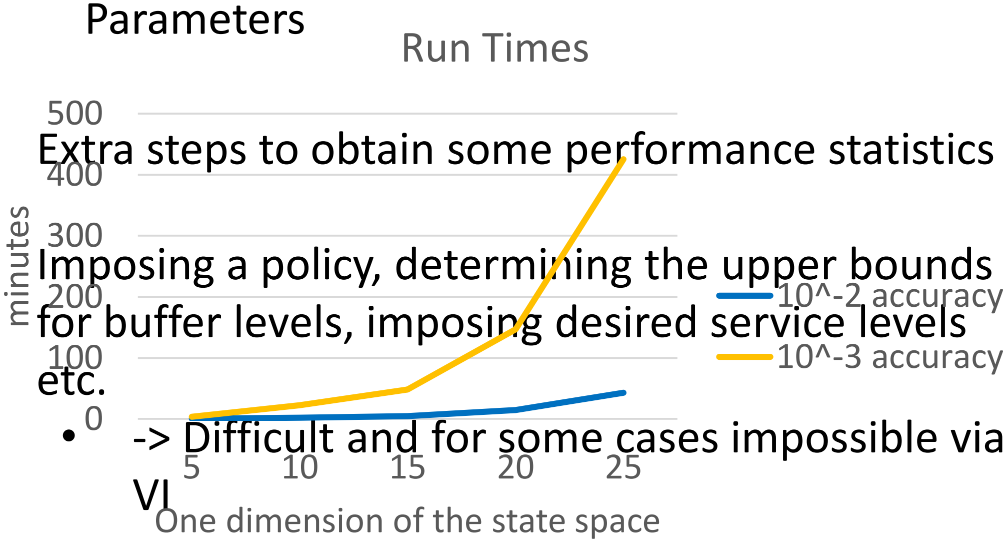
- comprehensive numerical experiments
- observe the system response to the changes in parameters, and induce managerial insights for design and control of the system.

Value Iteration (VI) Approach

Value Iteration Steps → Opt. Production Policies, Min. System Cost → One Step Transition Matrix → Steady State Analysis → Avg. Finished Goods inv., Avg. WIP, Lost Sales Rate, Min. System Cost

Parameters → Value Iteration Steps

- Extra steps to obtain some performance statistics
- Imposing a policy, determining the upper bounds for buffer levels, imposing desired service levels etc.
  - -> Difficult and for some cases impossible via VI



Linear Programming Approach

In order to obtain optimal policies faster with a more transparent representation, we prescribe

a **Linear Programming** approach.

Via LP,

- We reduce the run times,
- Directly obtain steady state probabilities.
  - Average inventory levels
  - Service levels etc.

In addition, we gain the flexibility to impose a variety of systemwide specifications and policy structures…

LP Formulation

Indices and Parameters:

i: Stations, $i \in \{1,2,\ldots,N\}$

M_i: Inventory truncation level for the buffer after station i

x: Intermediate and finished items inventory level (state vector)

- $x \in \mathbb{S} = \{(x_1, x_2, \ldots, x_N) | x_i \in \{0,1,\ldots,M_i\}\}$

u: Production decision for each station

- $u \in \mathbb{U}(x) = \{(u_1, u_2, \ldots, u_N) | u_i \in \{0,1\},\ u_i = 0 \text{ if } x_{i-1} = 0 \ or\ x_i = M_i\}$

μ_i: Processing rate for station i

λ: Demand rate

h_i: Inventory holding cost for the buffer after station i

c: Unit lost sales cost

LP Formulation

Minimize $\sum_{x\in\mathbb{S}}\sum_{u\in\mathbb{U}(x)} C_{xu}\pi_{xu}$

subject to

$\sum_{u\in\mathbb{U}(x)} \pi_{xu}\sum_{y\in\mathbb{S}} \nu_{y|x,u} - \sum_{y\in\mathbb{S}}\sum_{u\in\mathbb{U}(y)} \nu_{x|y,u}\pi_{yu} = 0,\ \forall x \in \mathbb{S},$

$\sum_{x\in\mathbb{S}}\sum_{u\in\mathbb{U}(x)} \pi_{xu} = 1$

$\pi_{xu} \geq 0\ , \quad \forall x \in \mathbb{S}, \forall u \in \mathbb{U}(x)$

Decision variables, π_{xu}: $P\{state = x, action = u\}$

LP Formulation

Minimize $\sum_{x \in \mathbb{S}} \sum_{u \in \mathbb{U}(x)} \boldsymbol{C_{xu}} \pi_{xu}$

subject to

$\sum_{u \in \mathbb{U}(x)} \pi_{xu} \sum_{y \in \mathbb{S}} \nu_{y|x,u} - \sum_{y \in \mathbb{S}} \sum_{u \in \mathbb{U}(y)} \nu_{x|y,u} \pi_{yu} = 0, \ \forall x \in \mathbb{S},$

$\sum_{x \in \mathbb{S}} \sum_{u \in \mathbb{U}(x)} \pi_{xu} = 1$

$\pi_{xu} \geq 0 , \quad \forall x \in \mathbb{S}, \forall u \in \mathbb{U}(x)$

Decision variables, π_{xu}: $P\{state = x, action = u\}$

$\boldsymbol{C_{xu}}$: Cost incurred at state x when action $u \in \mathbb{U}(x)$ is chosen

$$C_{xu} = \begin{cases} \sum_{i=1}^{m} x_i h_i & if\ x_N > 0 \\ \sum_{i=1}^{m} x_i h_i + \lambda c & if\ x_N = 0 \end{cases}$$

LP Formulation

Minimize $\sum_{x \in \mathbb{S}} \sum_{u \in \mathbb{U}(x)} C_{xu} \pi_{xu}$

subject to

$$\sum_{u \in \mathbb{U}(x)} \pi_{xu} \sum_{y \in \mathbb{S}} \boldsymbol{\nu_{y|x,u}} - \sum_{y \in \mathbb{S}} \sum_{u \in \mathbb{U}(y)} \nu_{x|y,u} \pi_{yu} = 0, \ \forall x \in \mathbb{S},$$

$$\sum_{x \in \mathbb{S}} \sum_{u \in \mathbb{U}(x)} \pi_{xu} = 1$$

$$\pi_{xu} \geq 0 , \quad \forall x \in \mathbb{S}, \forall u \in \mathbb{U}(x)$$

Decision variables, π_{xu}: $P\{state = x, action = u\}$

$\boldsymbol{\nu_{y|x,u}}$: Rate out from state x to state y under decision $u \in \mathbb{U}(x)$

$$\nu_{y|x,u} = \begin{cases} u_1 \mu_1 & if \ y = x + e_1 \\ u_i \mu_i & if \ y = x - e_{i-1} + e_i, \quad i \in \{2, \dots, N\} \\ \lambda & if \ y = x - e_N , x_N > 0 \\ 0 & else \end{cases}$$

LP Formulation

Minimize $\sum_{x\in\mathbb{S}}\sum_{u\in\mathbb{U}(x)} C_{xu}\pi_{xu}$

subject to

$\sum_{u\in\mathbb{U}(x)}\pi_{xu}\sum_{y\in\mathbb{S}}\nu_{y|x,u} - \sum_{y\in\mathbb{S}}\sum_{u\in\mathbb{U}(y)} \boldsymbol{\nu_{x|y,u}}\pi_{yu} = 0,\ \forall x\in\mathbb{S},$

$\sum_{x\in\mathbb{S}}\sum_{u\in\mathbb{U}(x)}\pi_{xu} = 1$

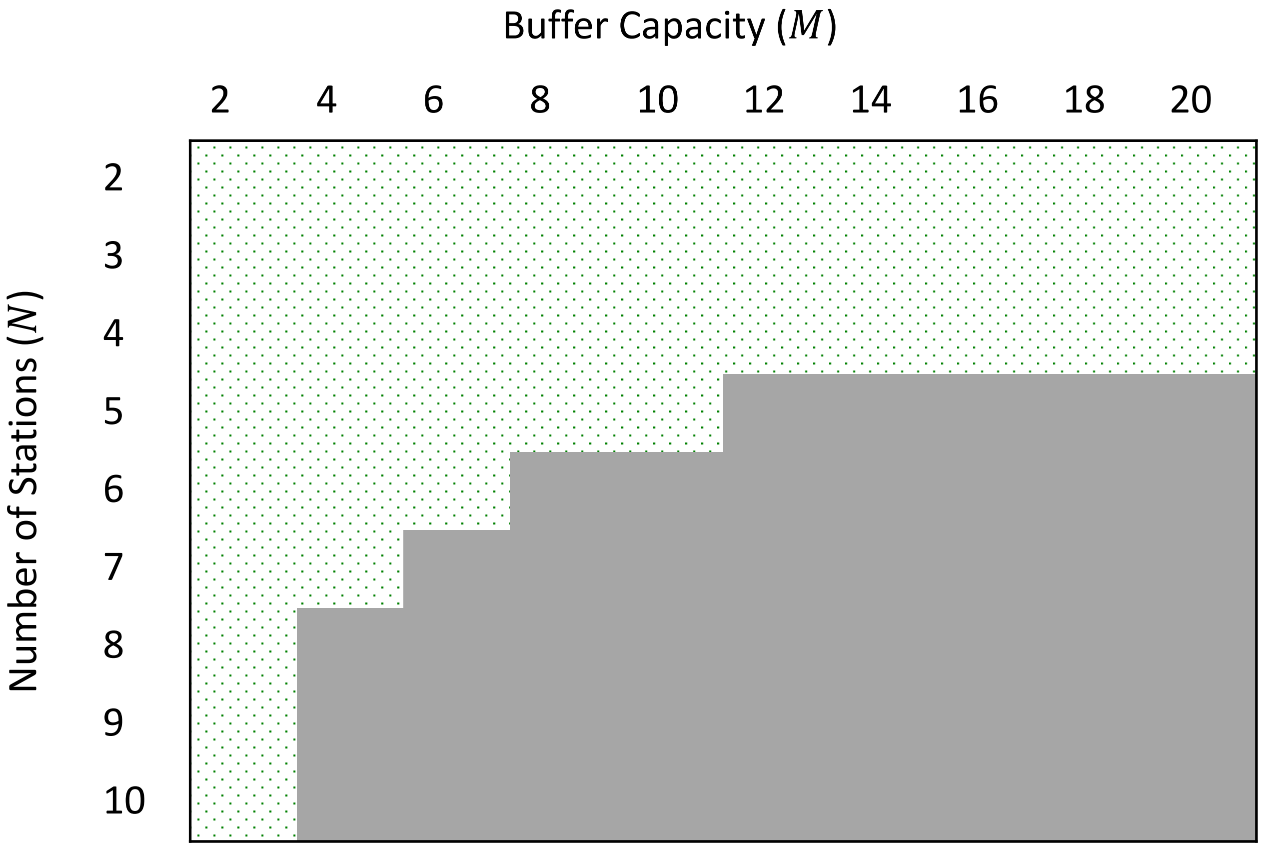
$\pi_{xu}\geq 0\,, \quad \forall x\in\mathbb{S}, \forall u\in\mathbb{U}(x)$

Decision variables, π_{xu}: $P\{state = x, action = u\}$

$\boldsymbol{\nu_{y|x,u}}$: Rate in to state x from state y under decision $u\in\mathbb{U}(y)$

$$\nu_{x|y,u} = \begin{cases} u_1\mu_1 & if\ y = x - e_1 \\ u_i\mu_i & if\ y = x - e_{i-1} + e_i, \quad i\in\{2,\dots,N\} \\ \lambda & if\ y = x + e_N \\ 0 & else \end{cases}$$

Instances Solved via LP



3-hour time limit , 128GB of RAM, 2.67 GHz processor
Using CPLEX solver

Size of the LP

$M^N + 1$ constraints

$M^N.2^N$ decision variables

↓

More decisions, more variables

N=5, M=10
100K constraints
More than 3M variables

Column generation refers to linear programming (LP) algorithms designed to solve problems in which there are a huge number of variables compared to the number of constraints

Column Generation (CG)

Initialization: Construct a feasible Master LP (original LP with restricted DS) with its columns.

↓

Coefficient vector for state action pair

Repeat:

1. Solve Master LP. Obtain dual variable values.
2. Among the columns with negative reduced costs, identify the one with the least cost. (called Subproblem)
3. Add the column to Master LP

Until no column with negative reduced cost left

Master Problem

$Minimize\ \sum_{x \in \mathbb{S}} \sum_{u \in \mathbb{U}'(x)} C_{xu} \pi_{xu}$

subject to

$\sum_{u \in \mathbb{U}'(x)} \pi_{xu} \sum_{y \in \mathbb{S}} \nu_{y|x,u} - \sum_{y \in \mathbb{S}} \sum_{u \in \mathbb{U}'(y)} \nu_{x|y,u} \pi_{yu} = 0\ ,\ \forall x \in \mathbb{S}$

$\sum_{x \in \mathbb{S}} \sum_{u \in \mathbb{U}'(x)} \pi_{xu} \ = 1$

$\pi_{xu} \geq 0\ ,\ \forall u \in \mathbb{U}'(x), \forall x \in \mathbb{S}$

Initially : $|\mathbb{U}'(x)| = 1, \forall x \in \mathbb{S}$, a feasible deterministic policy.

As algorithm proceeds, this set enlarges. (max. size of original LP)

Column Generation (CG)

CG Algorithm in general:

Start with initial columns A of Master LP (restricted)

Repeat:

1. Solve Master LP. Obtain dual variable values.
2. Among the columns with negative reduced costs, identify the one with the least cost. (Subproblem)
3. Add the column to Master LP

Until no column with negative reduced cost left

Subproblem

$$\min_{x,u}(Reduced\ Cost(x,u))$$

where

$Reduced\ Cost(x,u) = C_{xu} - \left(g + \sum_{y \in \mathbb{S}} v_{y|x,u}\, h_y - \sum_{x \in \mathbb{S}} v_{x|y,u} h_x\right)$

<u>Dual Problem:</u>

$Maximize\ g$

subject to

$g + \sum_{y \in \mathbb{S}} v_{y|x,u}\, h_y - \sum_{x \in \mathbb{S}} v_{x|y,u} h_x \geq C_{xu}, \forall u \in \mathbb{U}(x), \forall x \in \mathbb{S}$

g and h_i are unrestricted in sign.

Different CG Rules

Generally in CG, at each step:

Master Problem -> Subproblem is solved to obtain the column to be inserted

And at each step only one variable (with the most negative reduced cost) is inserted.

These properties may result as high number of steps (insertion of only one variable at a time) and large computation times (**our** subproblem is an IP).

How to improve?

Observation:

Objective function of the Master Problem is not changed even the column with most negative reduced cost inserted. (*degenerate* steps)

Mostly occur when inserted variable is away from current recurrent region.

Example: M/M/1 : Optimal threshold level 5.

Initial policy: with threshold level 3

- *Inserted state-action belongs to level 6 or higher*

Therefore, this step does not affect Master's objective but included to the basis with value zero. Results as redundant steps.

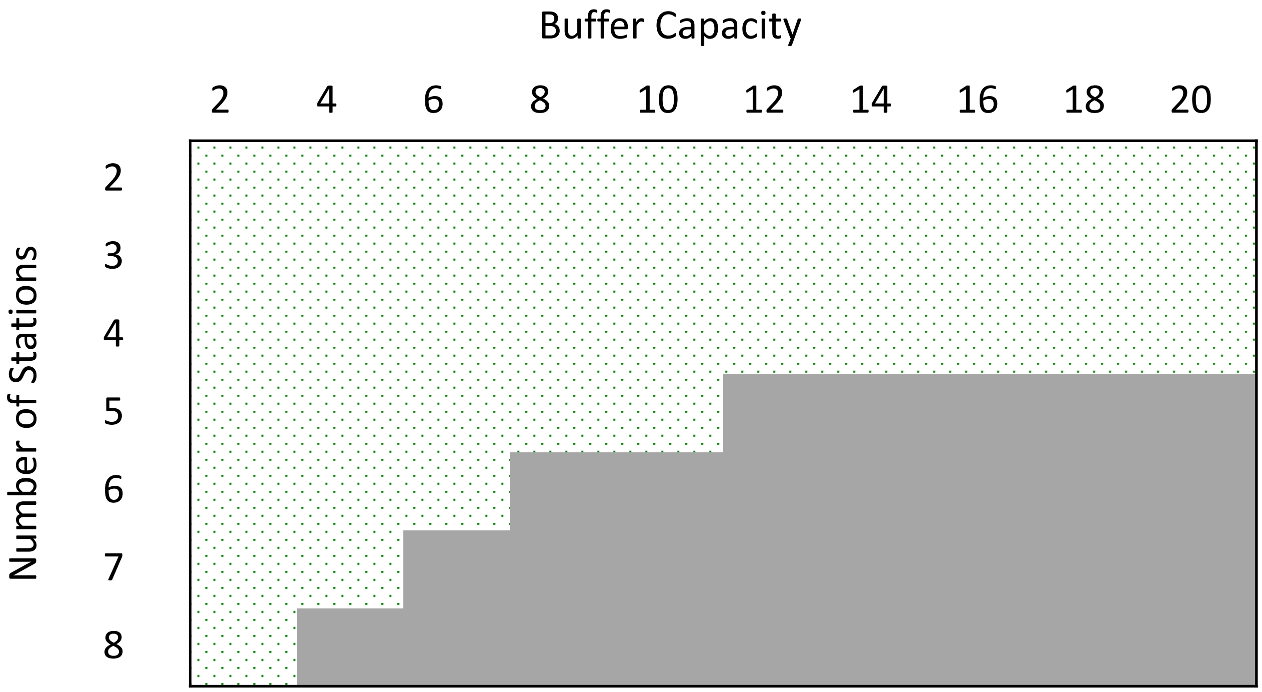
All Neighbors Rule

insert best columns that correspond to

neighbor states

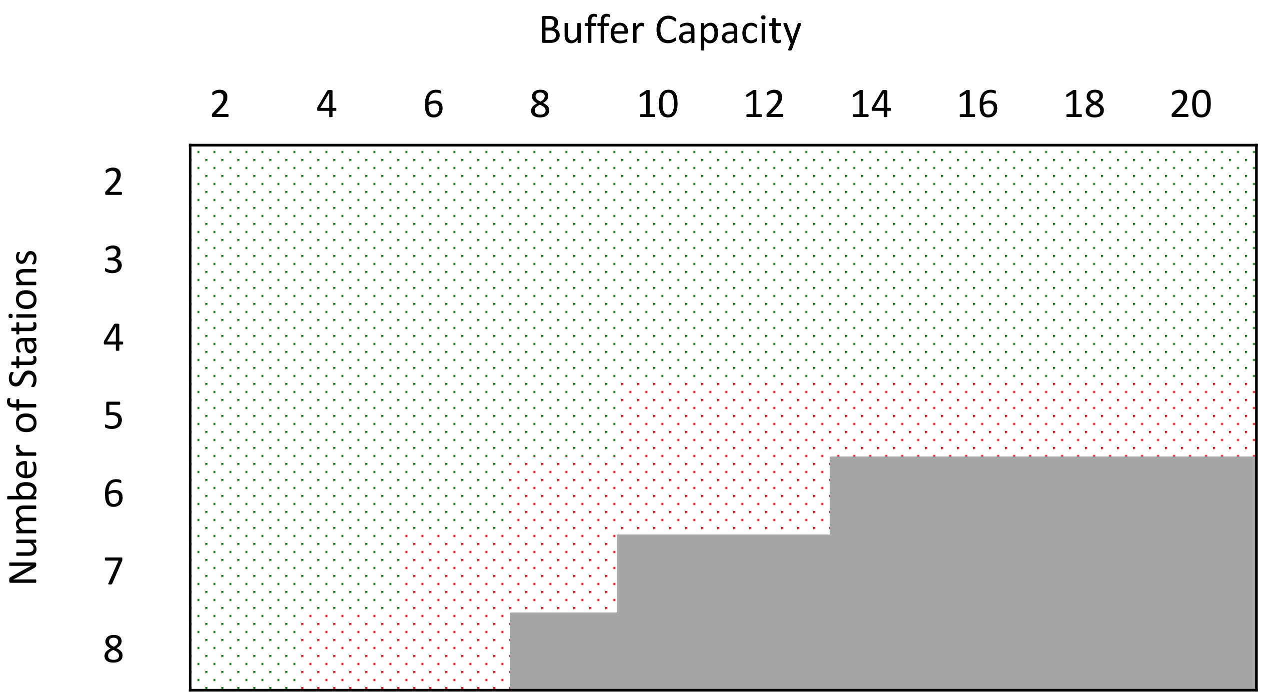
to the states with positive steady state probabilities in the current iteration.

Instances Solved
via LP



3-hour time limit , 128GB of RAM, 2.67 GHz processor

Instances Solved
via All Neighbors Rule



3-hour time limit , 128GB of RAM, 2.67 GHz processor
Initial policy: Never Produce

Run Time Comparison

# of stations	buffer size	LP	CG with All Neighbor Rule			
			optimal	%5	%10	%20
4	10	24	79	7	4	2
	12	35	4	2	1	1
	14	270	58	8	4	2
	16	530	70	12	7	3
	18	4344	206	38	22	14
	20	47	28	7	5	5
5	10	140	28	11	6	4
	12	24370	433	93	46	19
	14	-	324	77	39	28
	16	-	1241	337	168	168
	18	-	830	185	123	95

Conclusion

- LP has several advantages comparing to iterative methods, BUT as the problem gets bigger it cannot handle.

- Column Generation brings extra advantages on top of LP (run time reduction, extended solvable region)

- Column Generation is a promising algorithm for MDPs

Future Potential

-There are still many things to do

- Better search algortihms/efficient subproblem formulation

-Proposed rule/algorithm can be applied to any MDP. Problem specific properties and structures can also be used.

-It is a flexible approach: can be modified and improved by using the characteristics of the problem