Network Flow Modeling for Flexible Manufacturing Systems with Re-entrant Lines

Haitham Hindi & Wheeler Ruml

Palo Alto Research Center (PARC)

California, USA

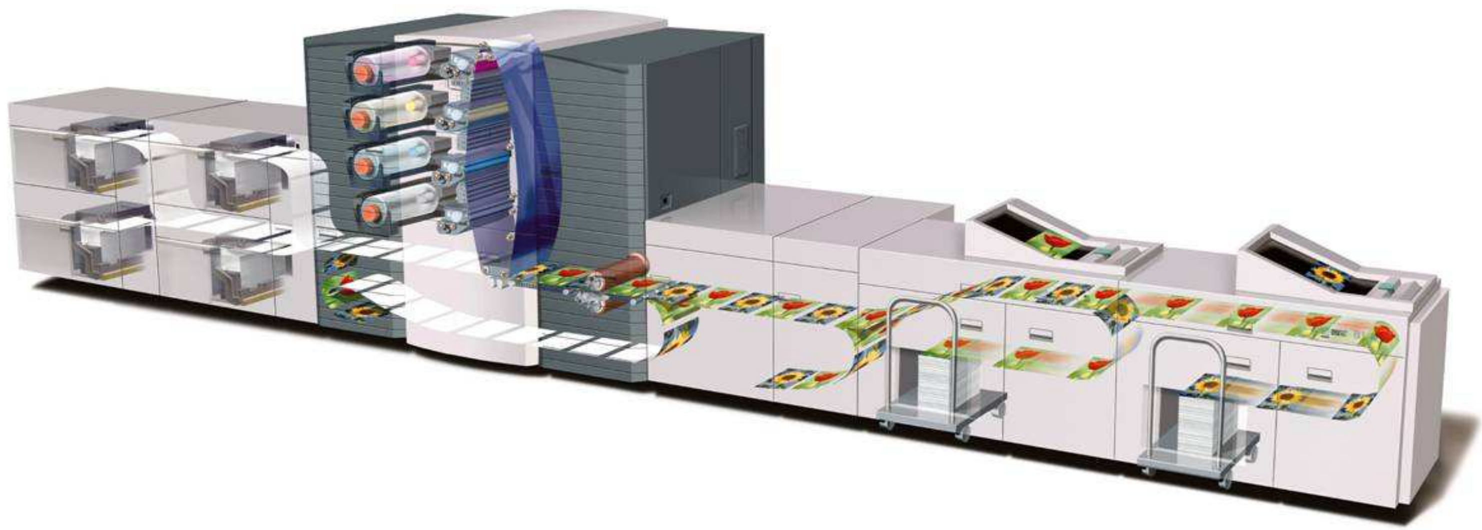
December, 2006

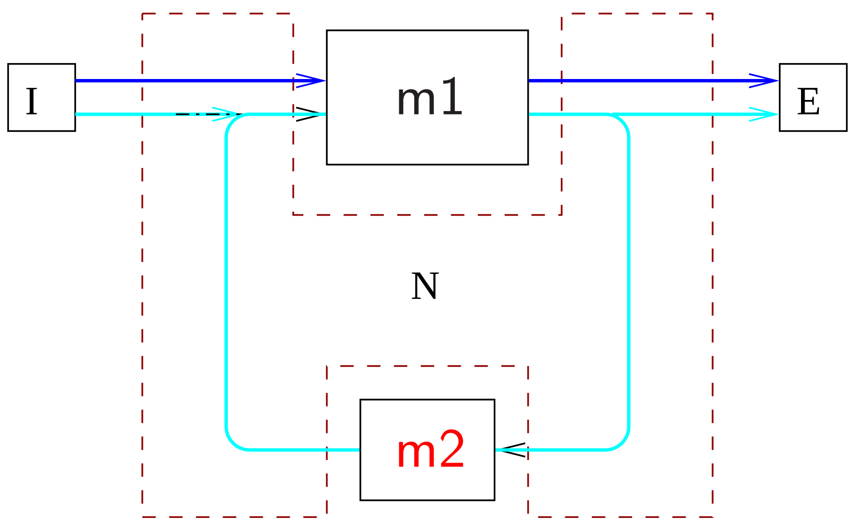
Outline

- motivation FMS networks from printing systems
- extend multicommodity network flow model to FMS networks
- examples of FMS networks
- conclusion

Inspired by Maglaras, Meyn, Kumar, Harrison, Weiss, Dai, Kelly, Xiao, Johansson, Boyd, Bertsimas, Sethuraman, Gamarnik, etc

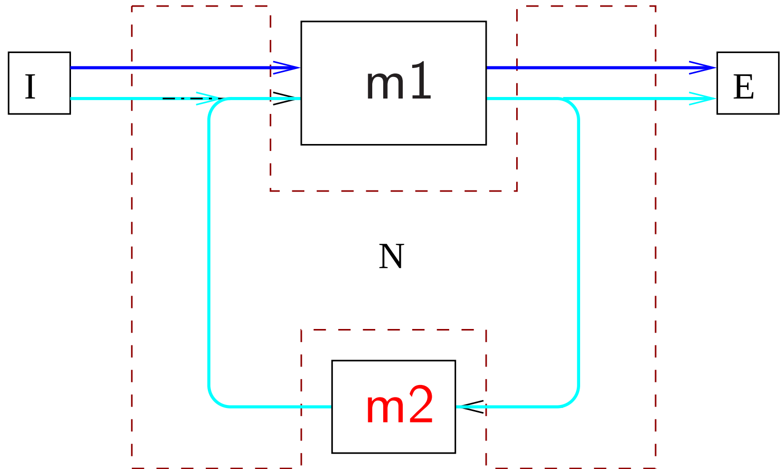
Motivation: printer as re-entrant manufacturing line





- all deterministic
- no buffers
- A*-search based scheduler

A simple re-entrant manufacturing line

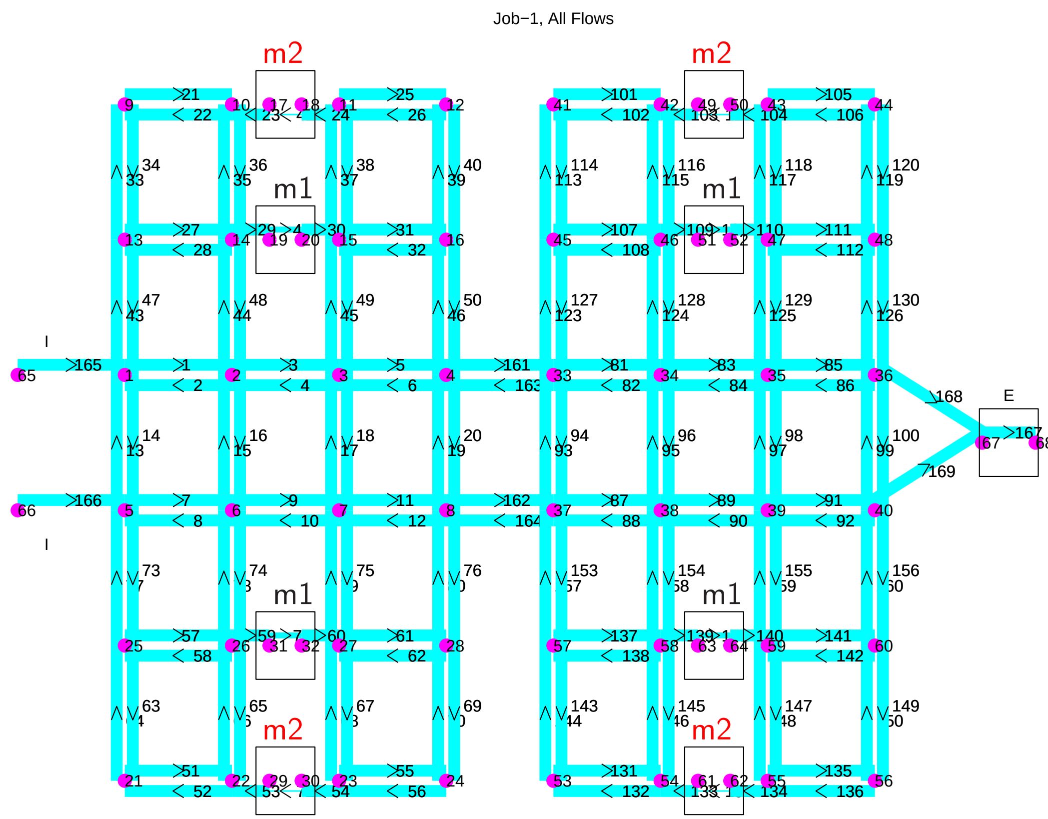


- set of components (I=input, E=exit, mi=machine i)
- interconnected by a network (N),
- jobs requiring different sequences of operations:

  Job1 (duplex) : I $\rightarrow$ m1 $\rightarrow$ m2 $\rightarrow$ m1 $\rightarrow$ E

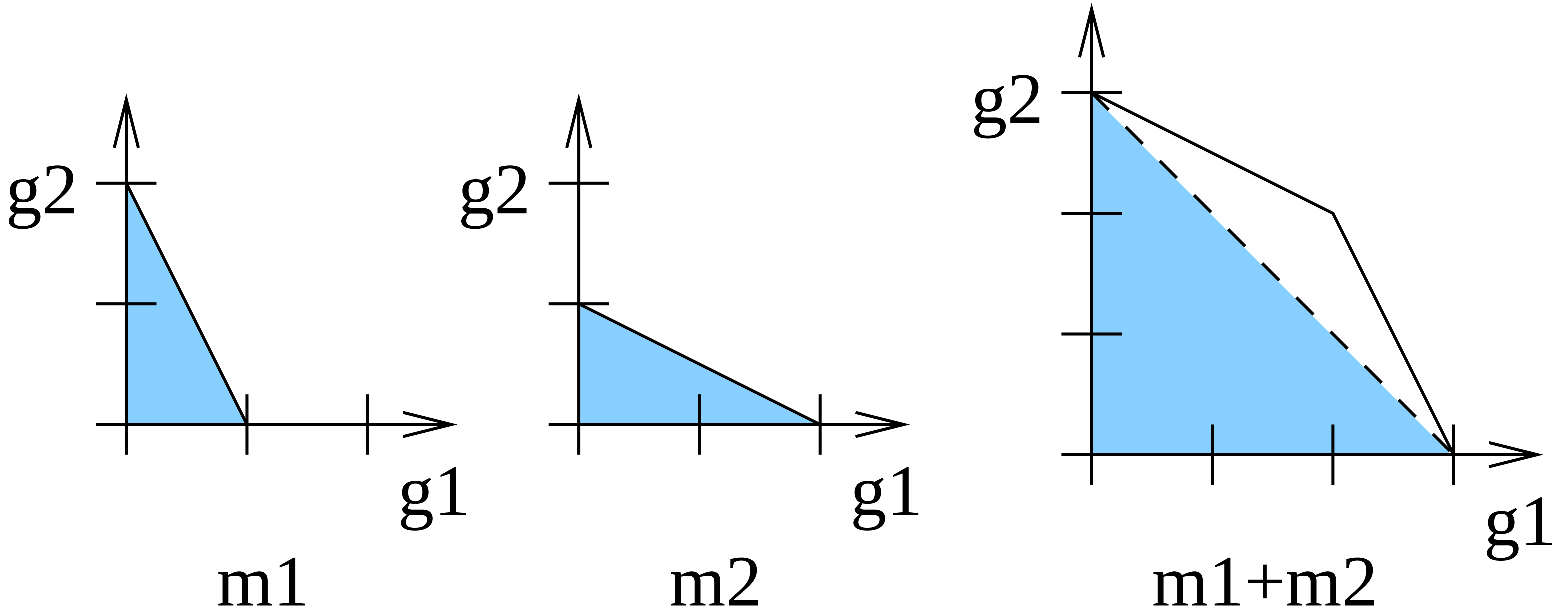
  Job2 (simplex): I $\rightarrow$ m1 $\rightarrow$ E

Flexible Manufacturing Network



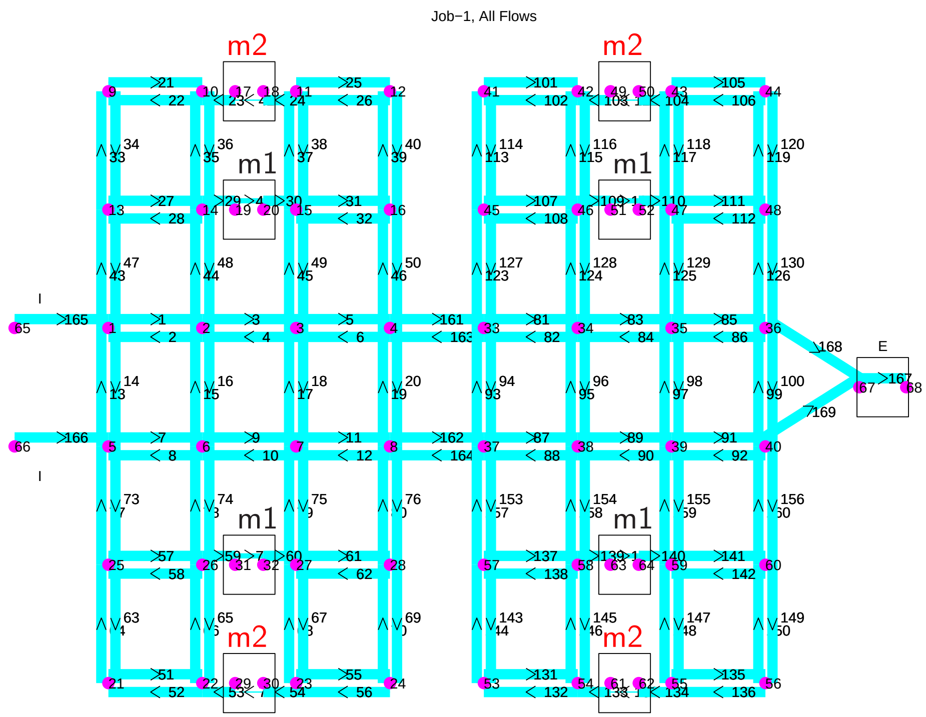
Reconfigurable FMS - can rearrange for different job mixes

Ricardo-Kantorovitch Resource Pooling



- multiple machines processing multiple widgets, with different processing capacities
- by specialization & trade can be more productive as a whole
- assumption: cost of maintenance comparable and transport & commun. cheap

Fault-tolerance & performance costs



- robustness through redundancy
- high-end machine often much costlier than collection of low-end machines

Goals

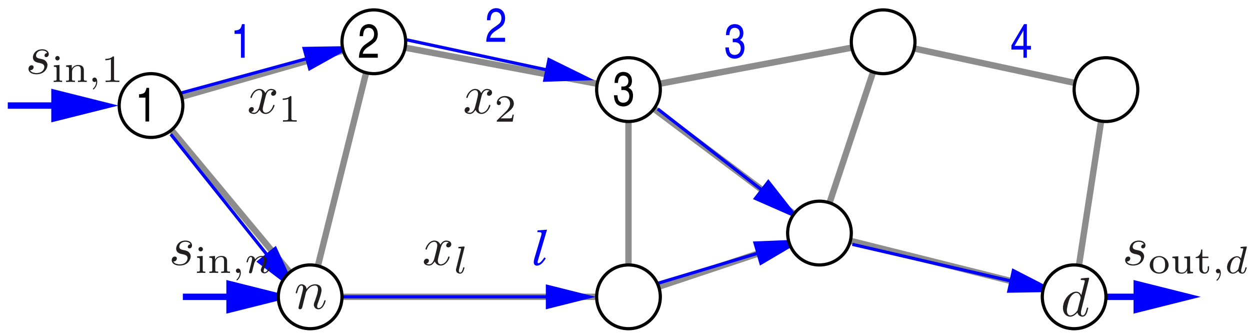
- construct network flow model for FMS
- captures steady state behavior
- design tool for large complex FMS networks
- relaxation ⇒ optimistic bounds on best achievable s.s. performance
- rapid evaluation of failure scenarios
- use as heuristic to enhance planner-scheduler
- based on convex optimization ⇒ handles large problems
- multiobjective design, sensitivity/reliability analysis (duality, Lagrange multipliers, etc)

Limitations

- Ignores discrete nature of widgets
- Doesnt produce real plans, only flow-based bounds on best achievable (since ignoring discreteness = relaxation)
- Still need discrete planner for "widget-level" itineraries
- May not be able to capture non-steady state costs, eg: make-span, rapid switching of routes, excessive start/stop of transports, etc
- Assumes a certain (reasonable) ordering of actions on widget:

  eg: input $\rightarrow$ m1 $\rightarrow$ m2 $\rightarrow$ m1 $\rightarrow$ output

Network flow model



- source flows $s_{\text{in},n} \geq 0$; sink flow $s_{\text{out},n} \geq 0$; links flows $x_l \geq 0$
- link capacity: $x_l \leq \mu_l$ or equivalently $p_l\, x_l \leq 1$; where $p_l = 1/\mu_l$

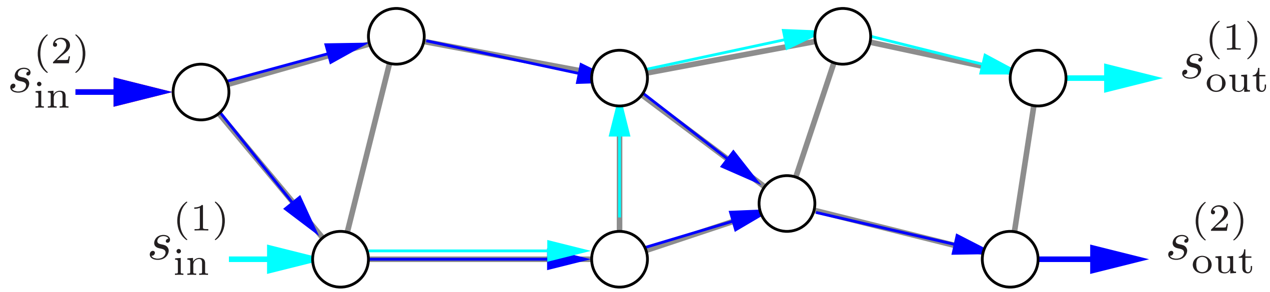
$$A\,x + B_{\text{out}}\,s_{\text{out}} - B_{\text{in}}\,s_{\text{in}} = 0; \quad \textit{(flow conservation)}$$

$$P\,x \preceq \mathbf{1} \quad \textit{(capacity constraints)}$$

$$x \succeq 0; \quad s_{\text{out}} \succeq 0; \quad s_{\text{in}} \succeq 0; \quad \textit{(nonnegativity)}$$

- A is incidence matrix; $P = \text{diag}(p_1, \ldots, p_L)$; $\mathbf{1}$ is vector of all-ones

Multicommodity network flow model



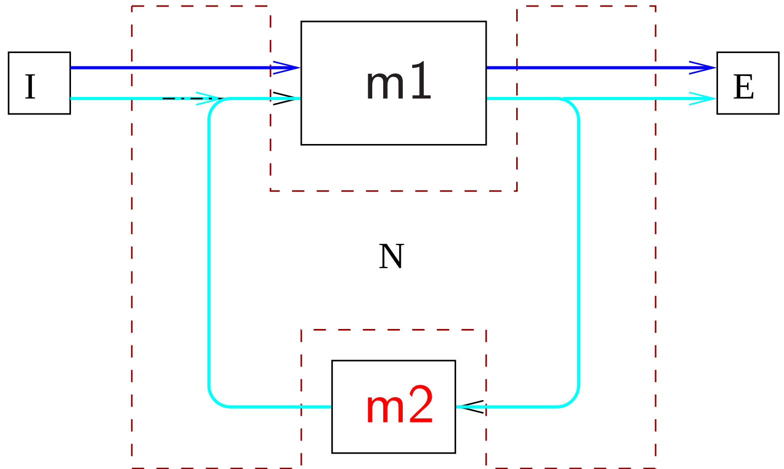
- each commodity i has $s_{\mathrm{in}}^{(i)}$, $s_{\mathrm{out}}^{(i)}$, $x^{(i)}$
- flow conservation holds for every commodity
- commodities must share links: $\sum_i p_l^{(i)} x_l^{(i)} \leq 1$

$$A\,x^{(i)} + B_{\mathrm{out}}^{(i)}\,s_{\mathrm{out}}^{(i)} - B_{\mathrm{in}}^{(i)}\,s_{\mathrm{in}}^{(i)} = 0; \quad \forall i \qquad \textit{(flow conservation)}$$

$$\textstyle\sum_i P^{(i)}\,x^{(i)} \preceq \mathbf{1}; \qquad \textit{(capacity constraints)}$$

$$x^{(i)} \succeq 0; \quad s_{\mathrm{out}}^{(i)} \succeq 0; \quad s_{\mathrm{in}}^{(i)} \succeq 0; \quad \forall i \qquad \textit{(nonnegativity)}$$

A simple re-entrant manufacturing line



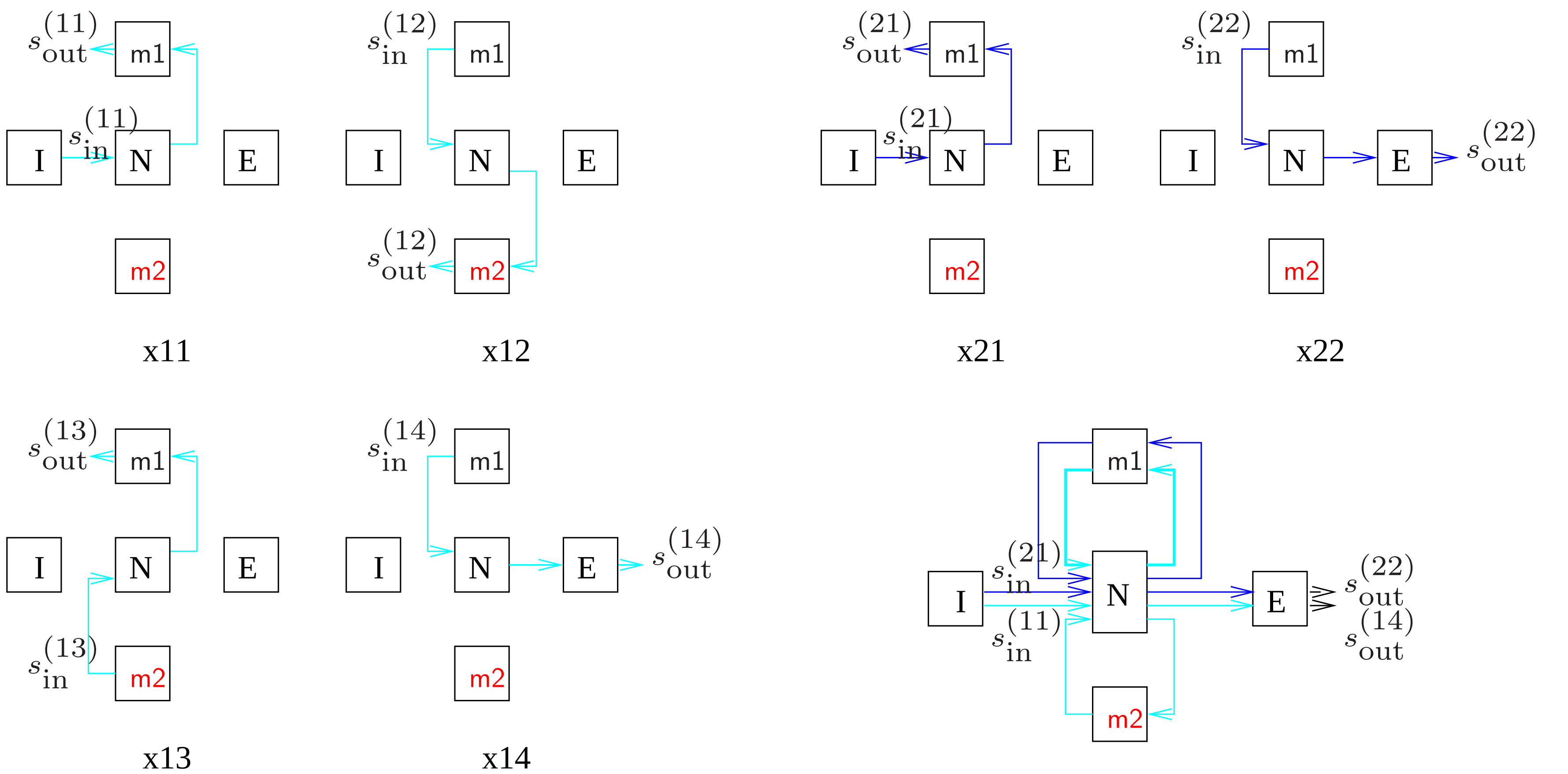
- set of components (I=input, E=exit, mi=machine i)
- interconnected by a network (N),
- jobs requiring different sequences of operations:

  Job1 (duplex) : I → m1 → m2 → m1 → E

  Job2 (simplex): I → m1 → E

Re-entrant flow as cascaded standard flows

***Job1* (duplex) : I → m1 → m2 → m1 → E;** ***Job2* (simplex): I → m1 → E**



Multi-stage multi-commodity network flow model

$x^{(ij)} = \left(x_1^{(ij)}, \ldots, x_L^{(ij)}\right)$, link flows for job-$i$, stage-$j$

$$
\begin{array}{ll}
A\,x^{(ij)} + B_{\mathrm{out}}^{(ij)} s_{\mathrm{out}}^{(ij)} - B_{\mathrm{in}}^{(ij)} s_{\mathrm{in}}^{(ij)} = 0; \quad \forall i, j & \textit{(flow conservation)} \\
\Phi_{ij}\,x^{(ij)} = 0; \quad \forall i, j & \textit{(turn off unused machines)} \\
s_{\mathrm{in}}^{(i,j+1)} = s_{\mathrm{out}}^{(i,j)}; \quad \forall i, \forall j \geq 2 & \textit{(cascade multiple stages)} \\
\sum_{i=1}^{J} \sum_{j=1}^{F_i} P^{(ij)} x^{(ij)} \preceq \mathbf{1} & \textit{(capacity constraints)} \\
x^{(ij)} \succeq 0; \quad s_{\mathrm{out}}^{(ij)} \succeq 0; \quad s_{\mathrm{in}}^{(i1)} \succeq 0; \quad \forall i, j & \textit{(nonnegativity)}
\end{array}
\tag{1}
$$

All constraints affine $\Rightarrow$ convex!

Objective function

- Steady state throughput: one of our main motivations

$$\begin{array}{ll} \max & \sum_{i=1}^{J} s_{\text{in}}^{(i1)} \\ \text{s.t.} & (1). \end{array}$$

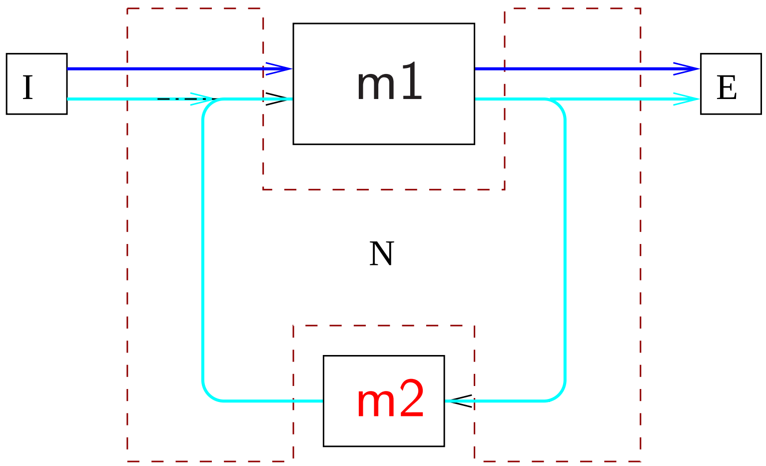
- Link utilization: find sparse flows or short paths with l_1-norm objective

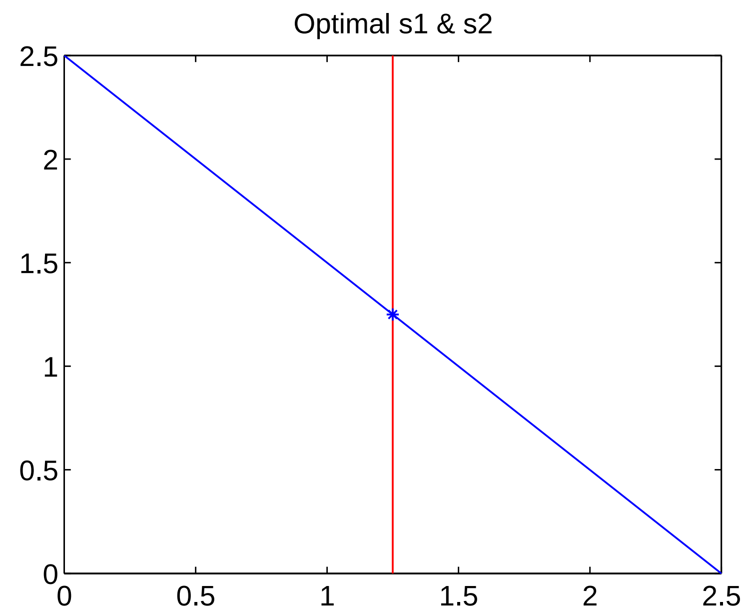
$$\begin{array}{ll} \min & \sum_{i=1}^{J} \sum_{j=1}^{F_i} \|x^{(ij)}\|_1 \\ \text{s.t.} & s_{\text{in}}^{(i1)} \succeq \alpha_{\text{min}}^{(i)}, \ i = 1, \ldots, J; \quad (1). \end{array}$$

- Load Balancing: use objective that penalizes deviation of the flows of the machines from each other, or via constraints

- Utility, Fairness, Delays: also possible

A Simple Two Job System



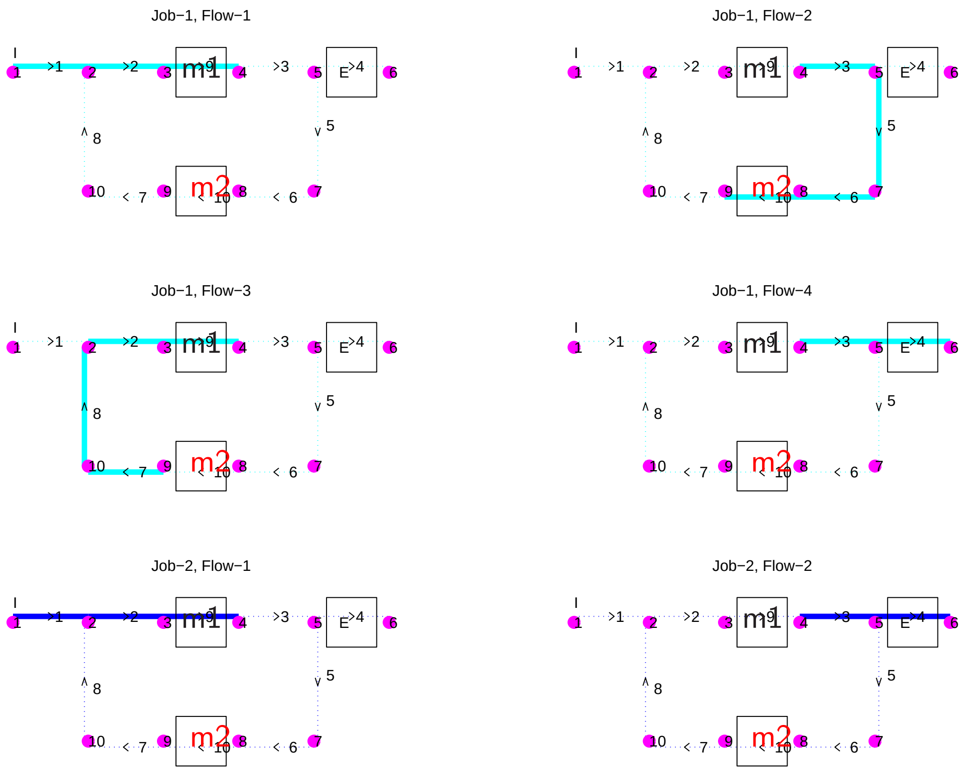


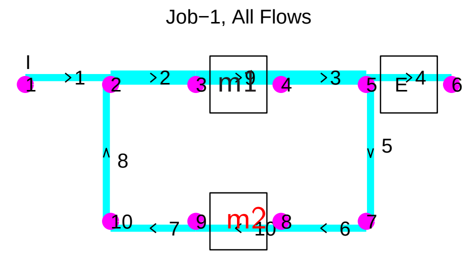
Job1: I $\longrightarrow$ m1 $\longrightarrow$ m2 $\longrightarrow$ m1 $\longrightarrow$ E

Job2: I $\longrightarrow$ m1 $\longrightarrow$ E

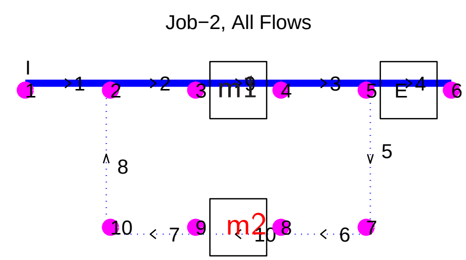
objective: throughput $+ \, \epsilon \, l_1$

A Simple Two Job System





Aggregate flows of Job1.

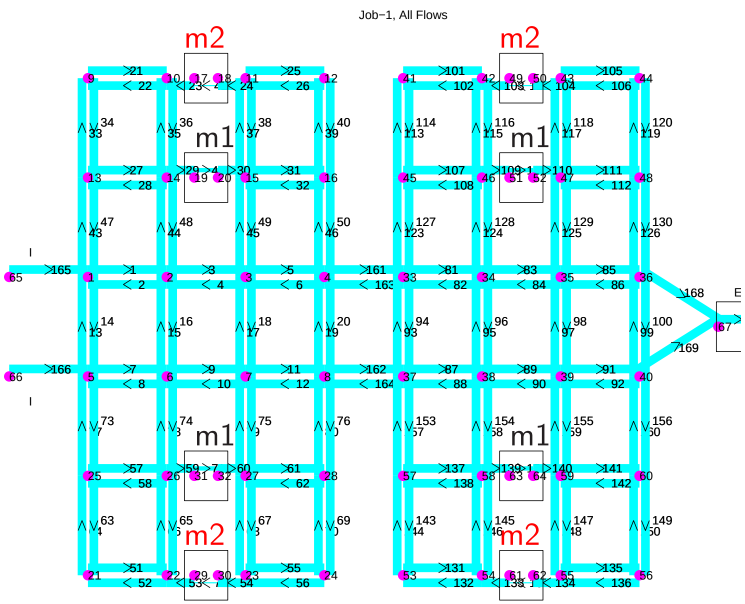


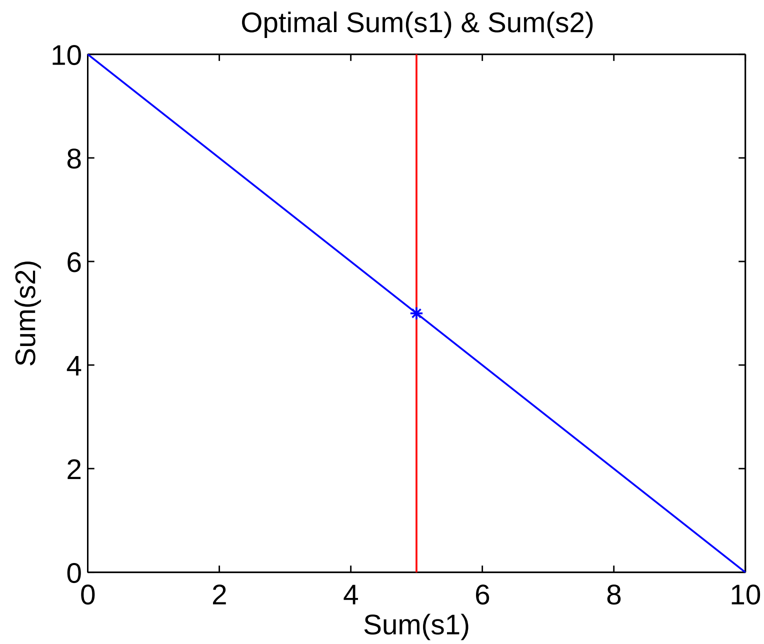
Aggregate flows of Job2.

Job1: I $\longrightarrow$ m1 $\longrightarrow$ m2 $\longrightarrow$ m1 $\longrightarrow$ E

Job2: I $\longrightarrow$ m1 $\longrightarrow$ E

Multimachine Example with Complex Routing

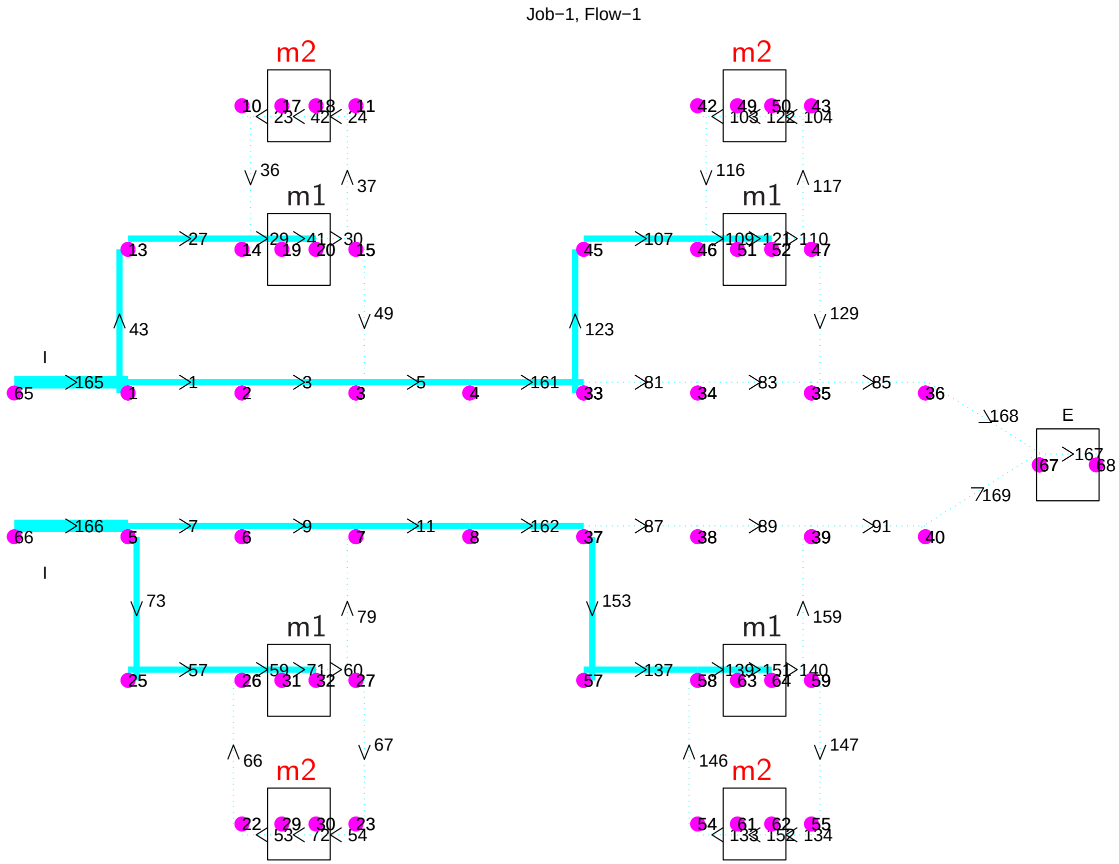




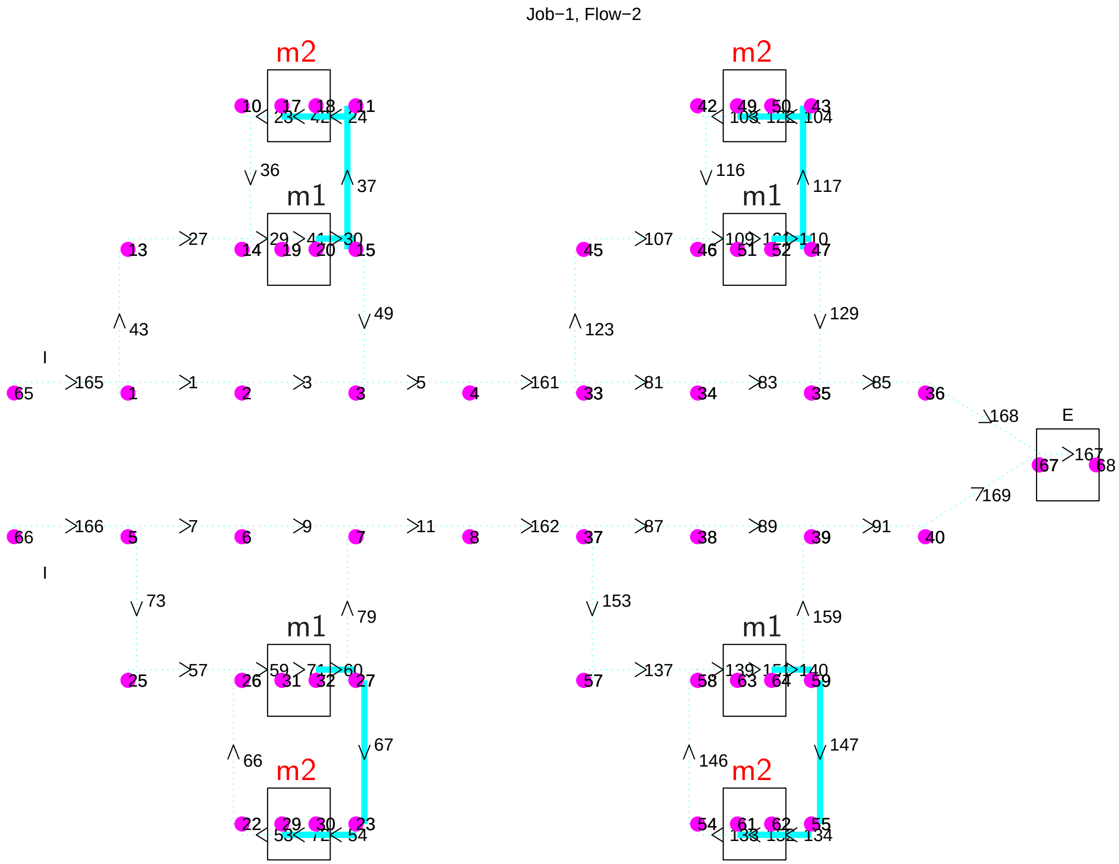
Job1: I $\longrightarrow$ m1 $\longrightarrow$ m2 $\longrightarrow$ m1 $\longrightarrow$ E
Job2: I $\longrightarrow$ m1 $\longrightarrow$ E

objective: throughput $+\ \epsilon\, l_1$

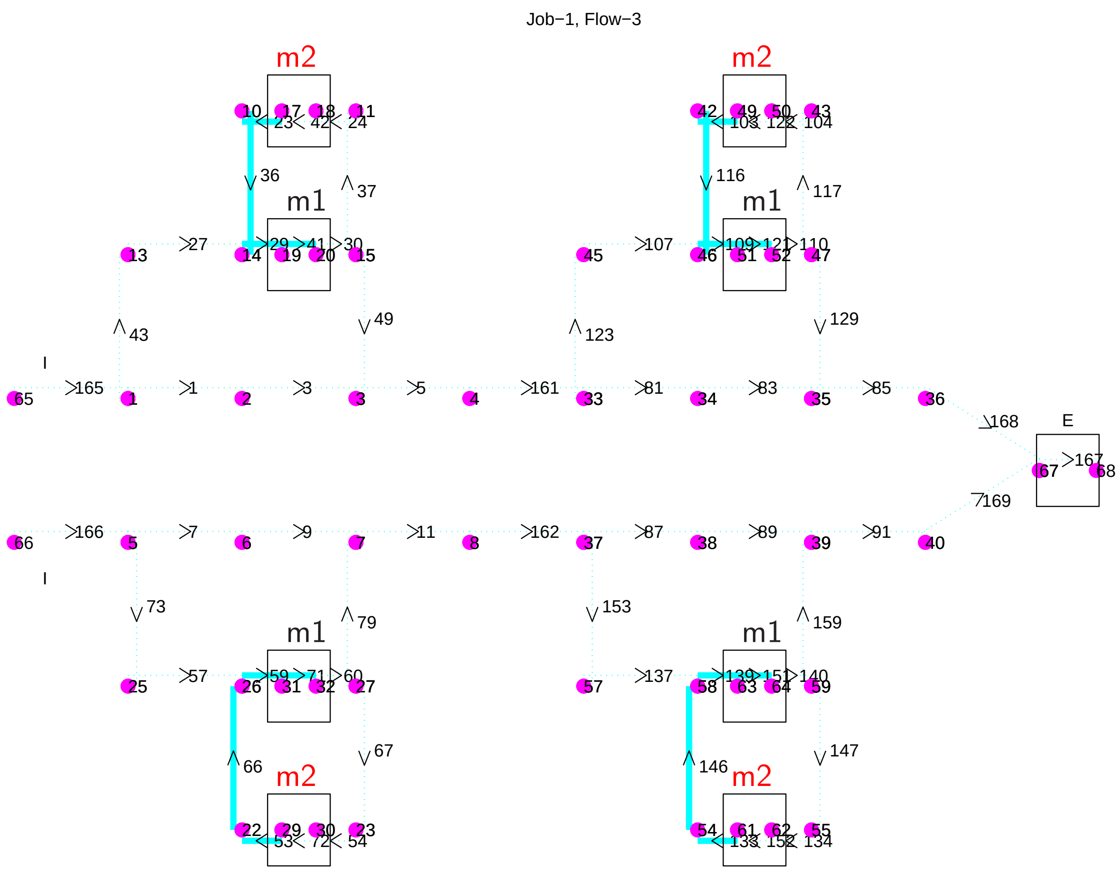
***Job1*: I → m1 → m2 → m1 → E**



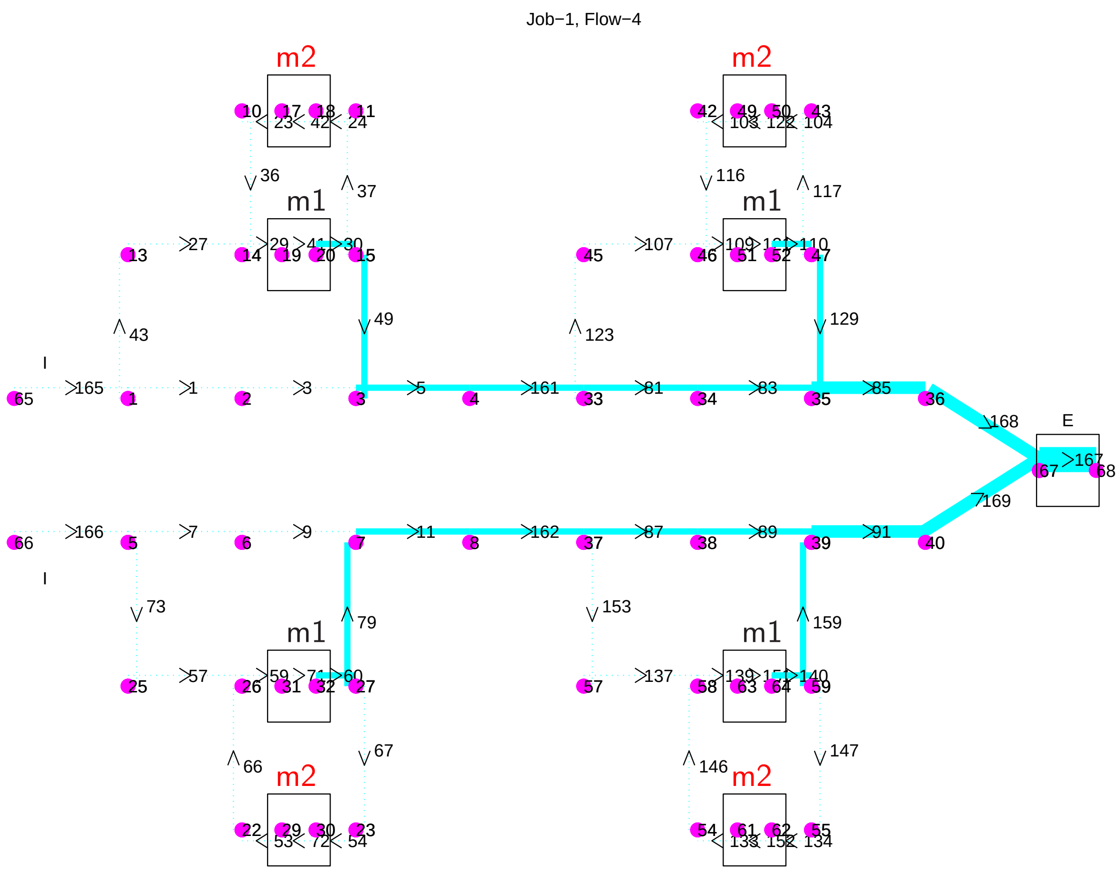
Job1: I → **m1** → **m2** → m1 → E



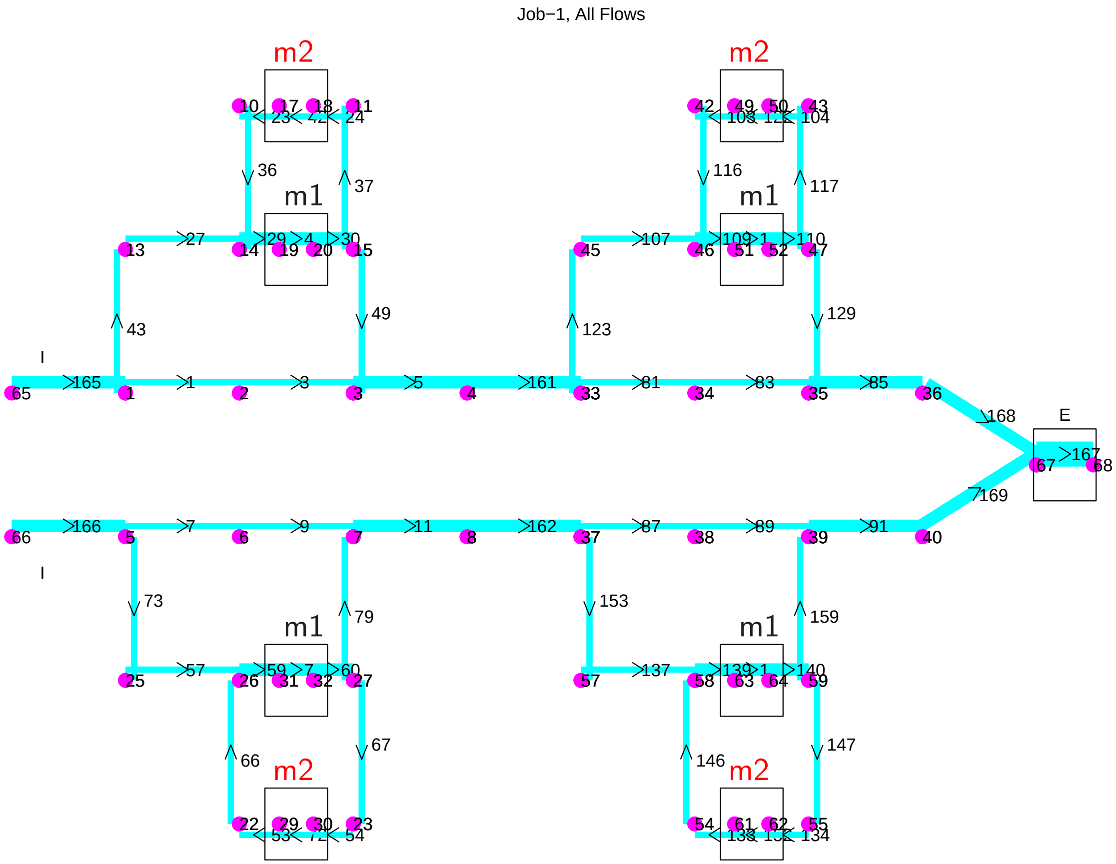
Job1: I → m1 → m2 → m1 → E



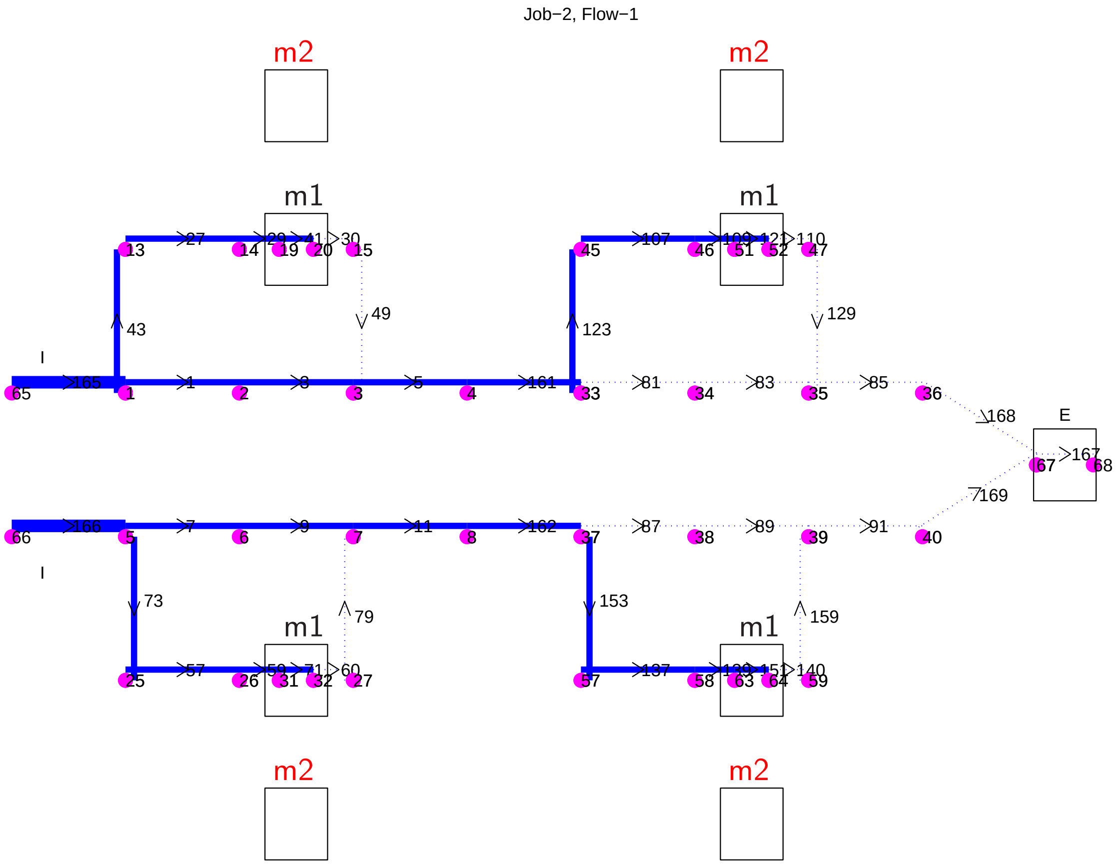
Job1: I → m1 → m2 → **m1** → **E**



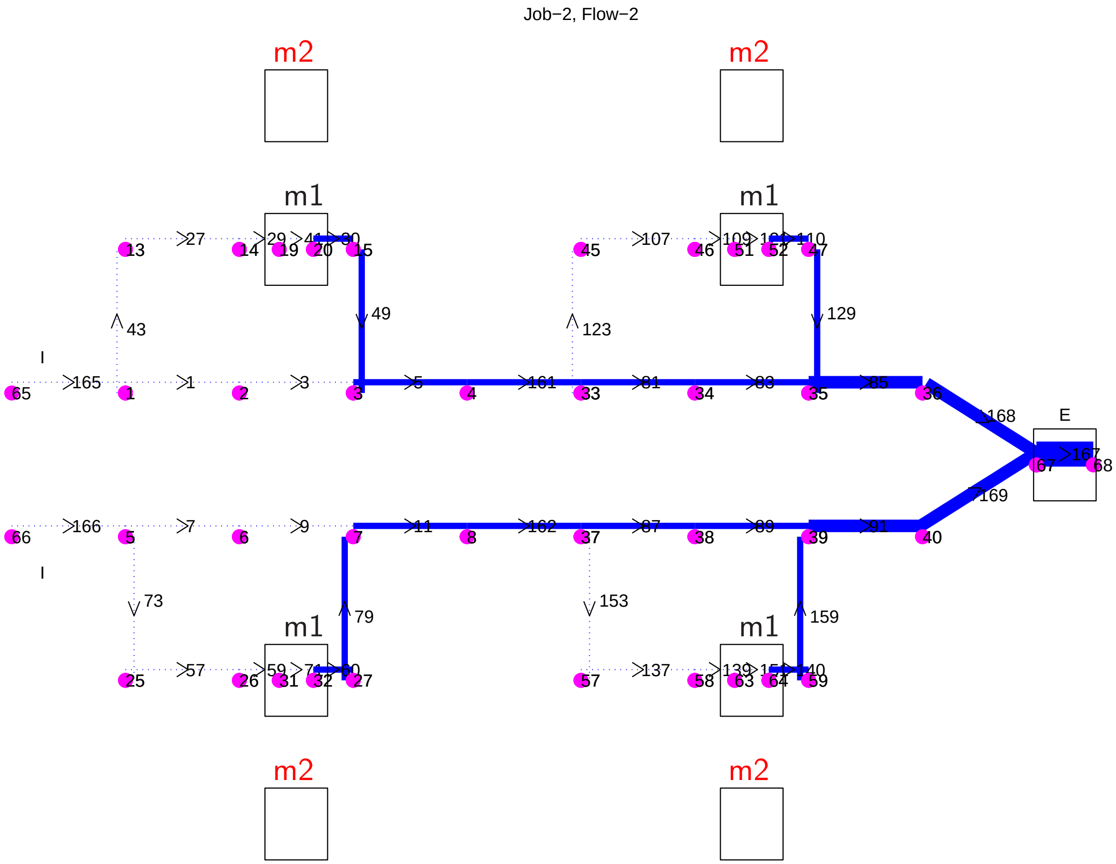
Job1: I → m1 → m2 → m1 → E



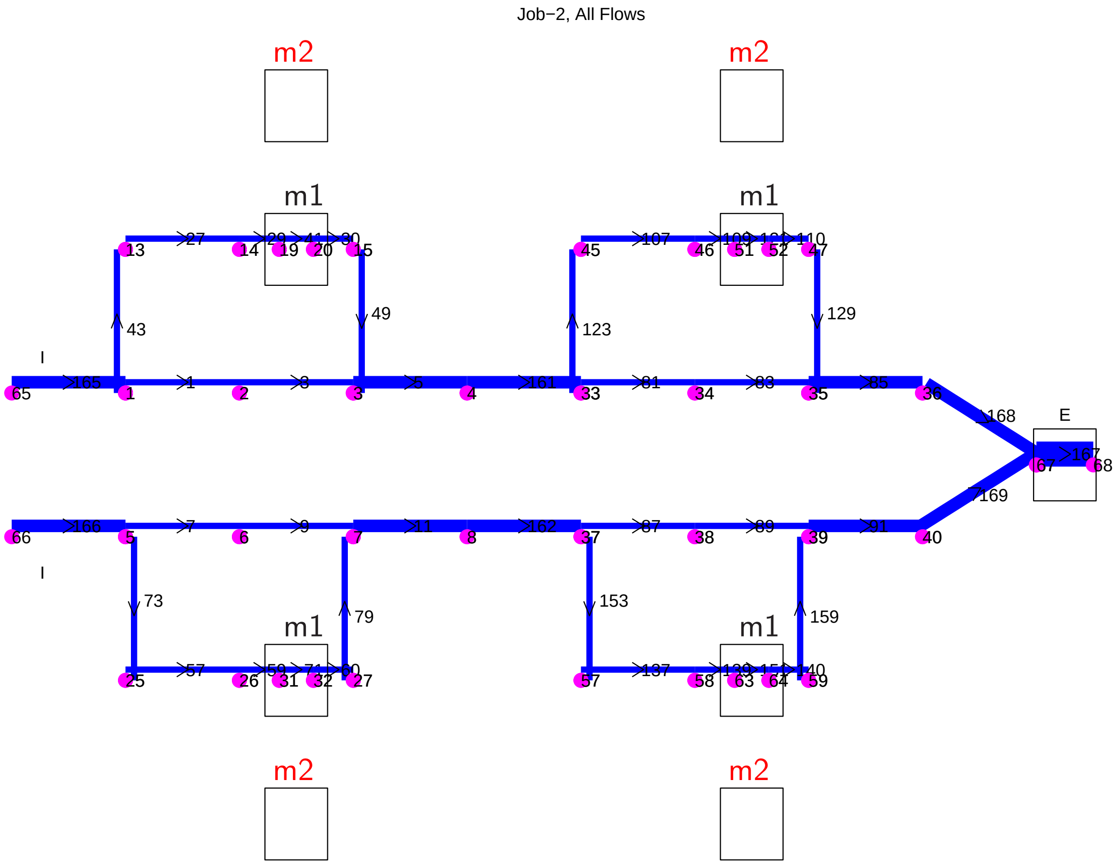
Job2: I → m1 → E



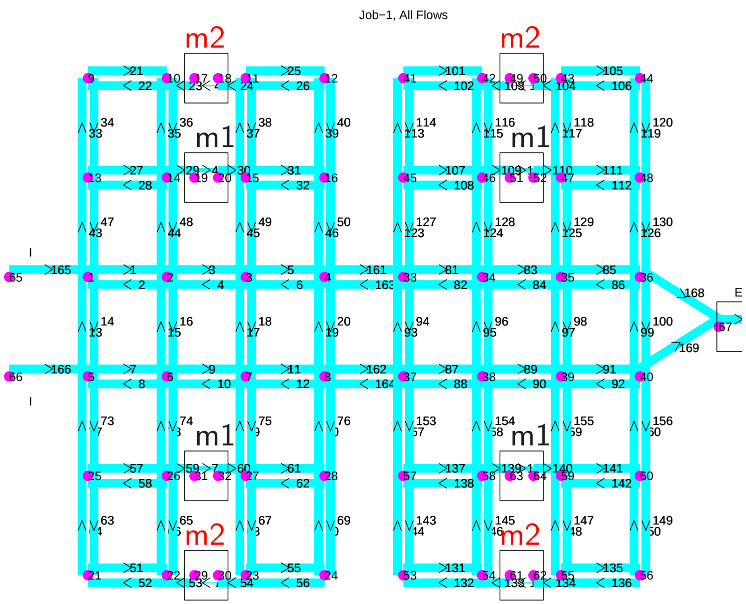
Job2: I → m1 → E

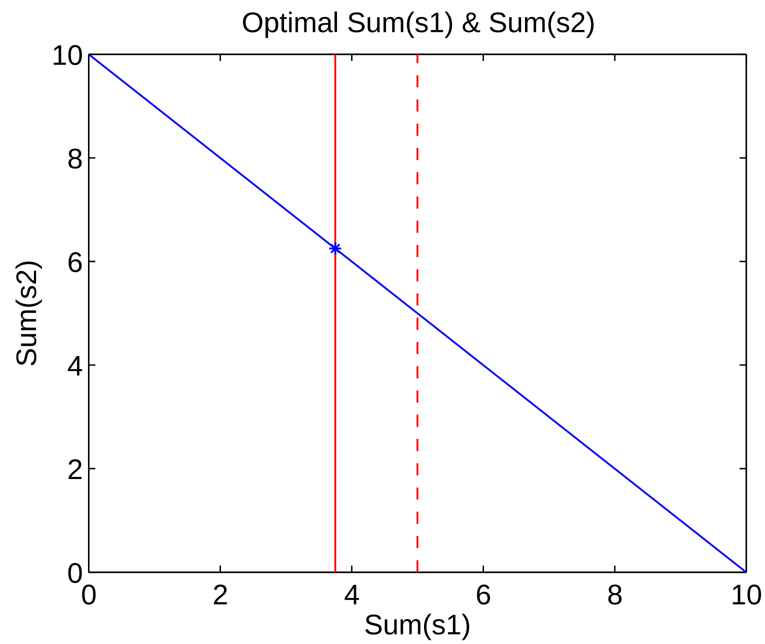


Job2: I → m1 → E



Multimachine Example: broken upper-right m2

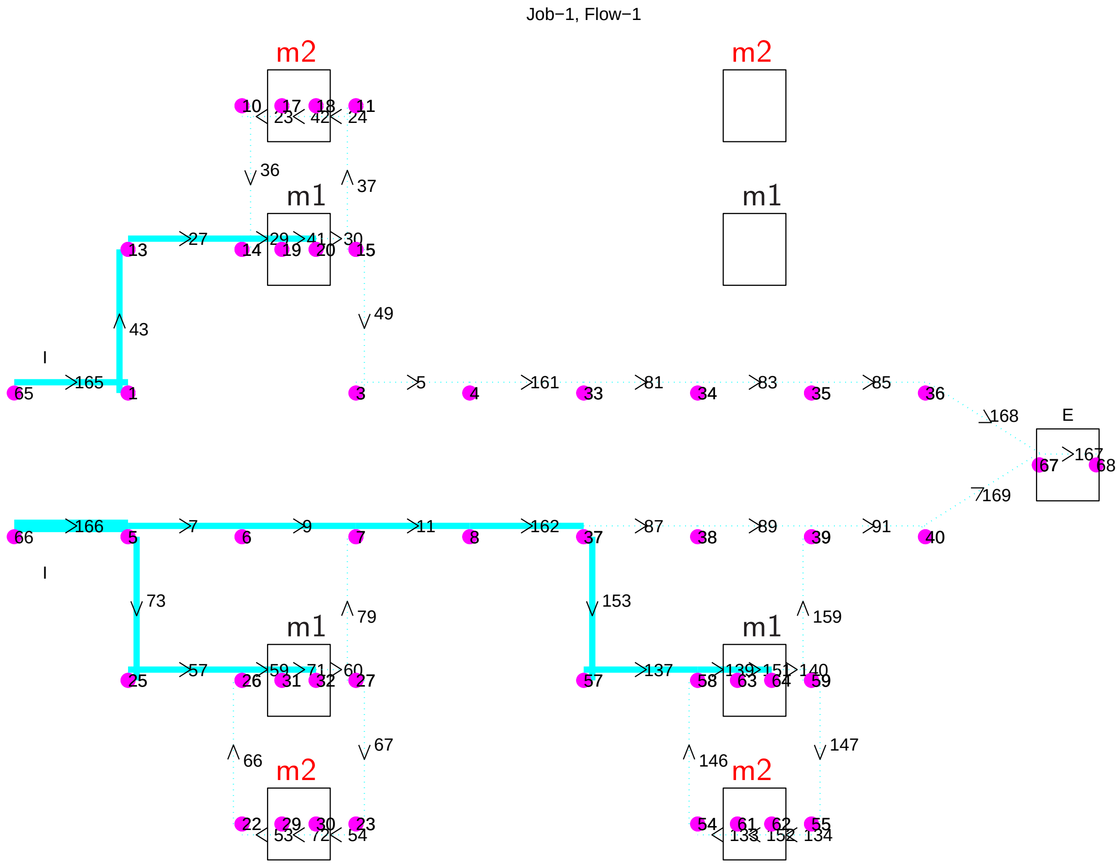




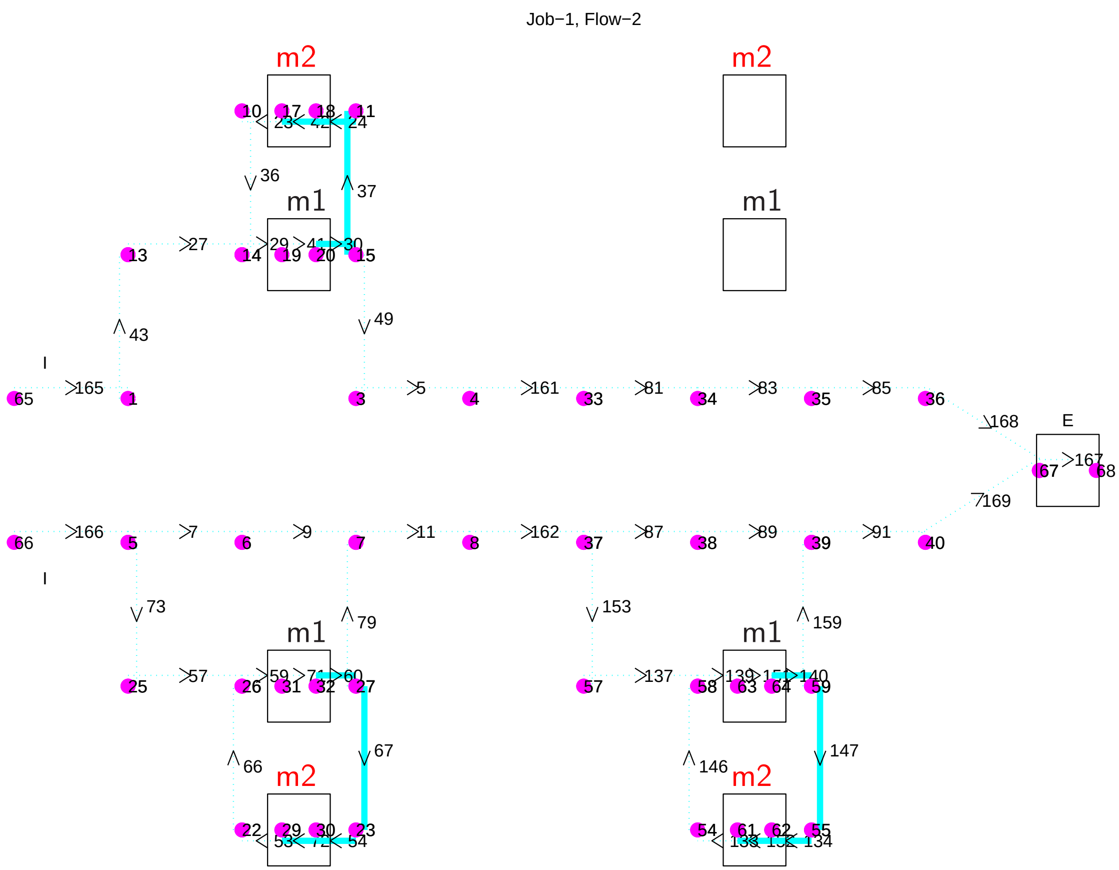
Job1: I → m1 → m2 → m1 → E
Job2: I → m1 → E

objective: throughput $+\ \epsilon\, l_1$

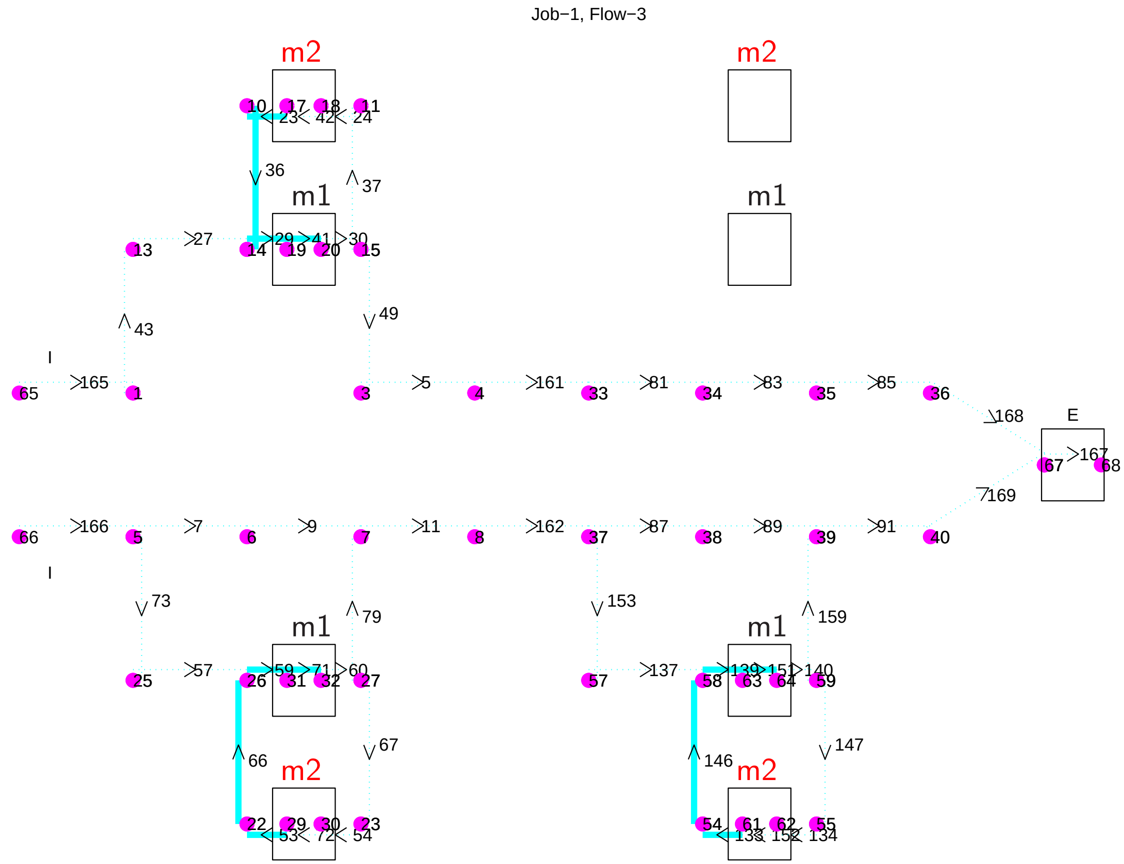
Job1: I → m1 → m2 → m1 → E



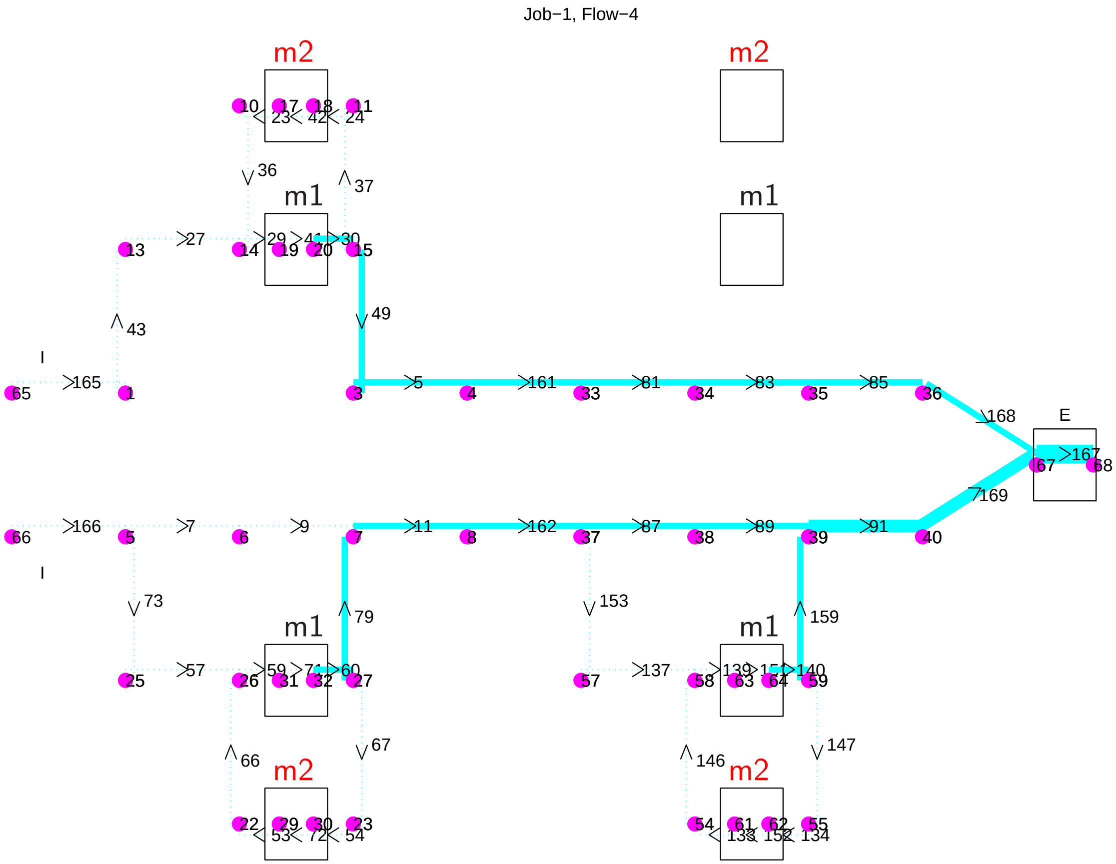
Job1: I → m1 → m2 → m1 → E



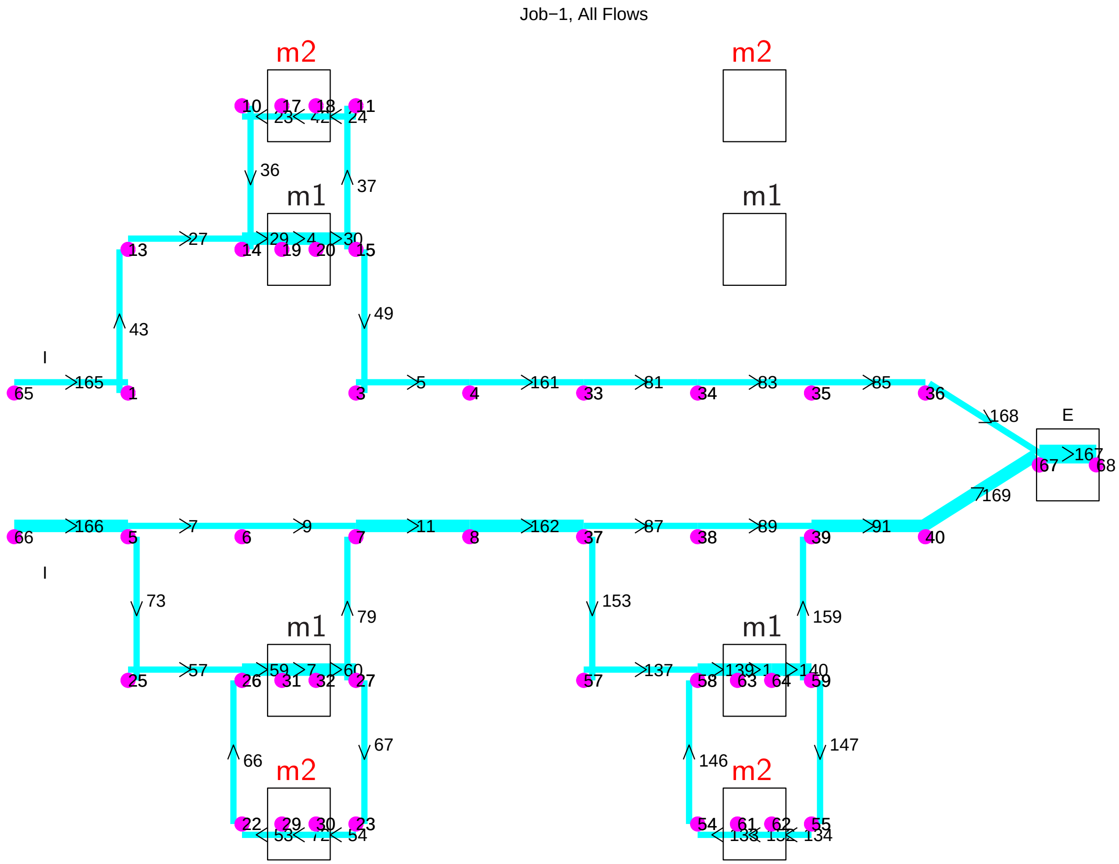
Job1: I → m1 → m2 → m1 → E



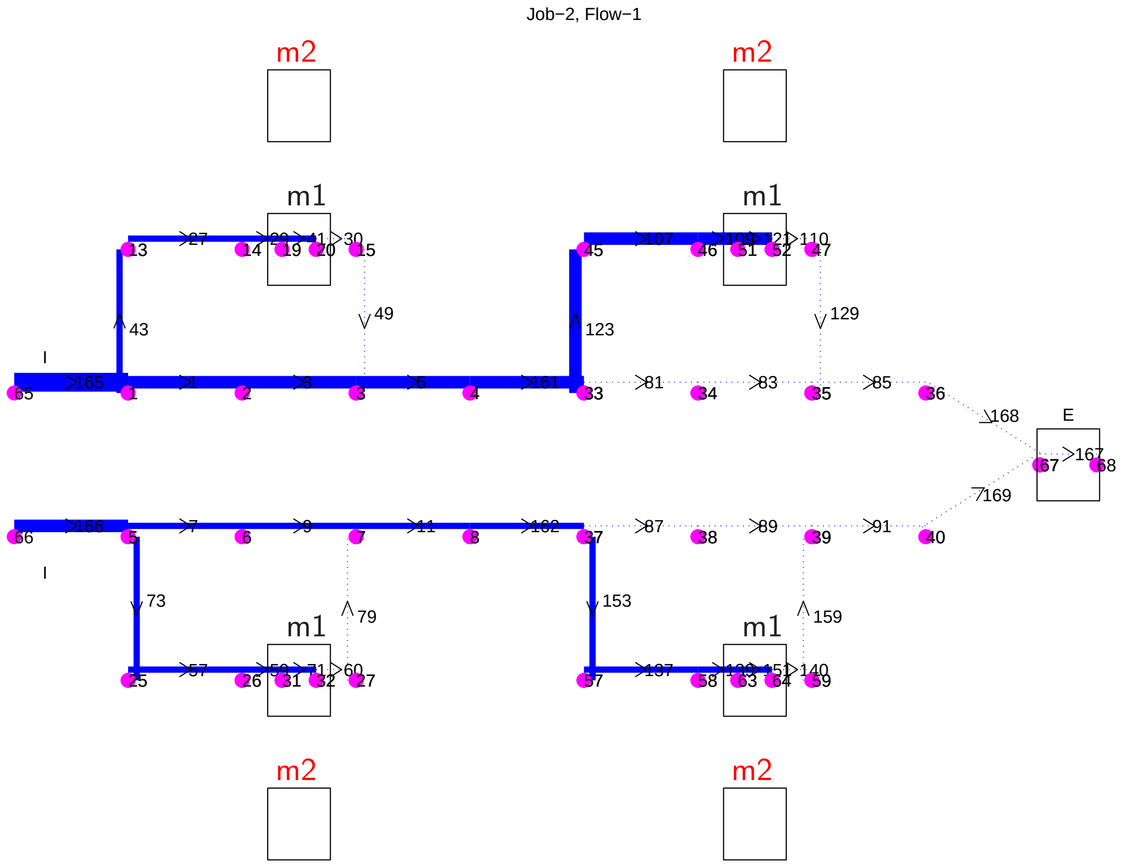
Job1: I → m1 → m2 → **m1 → E**



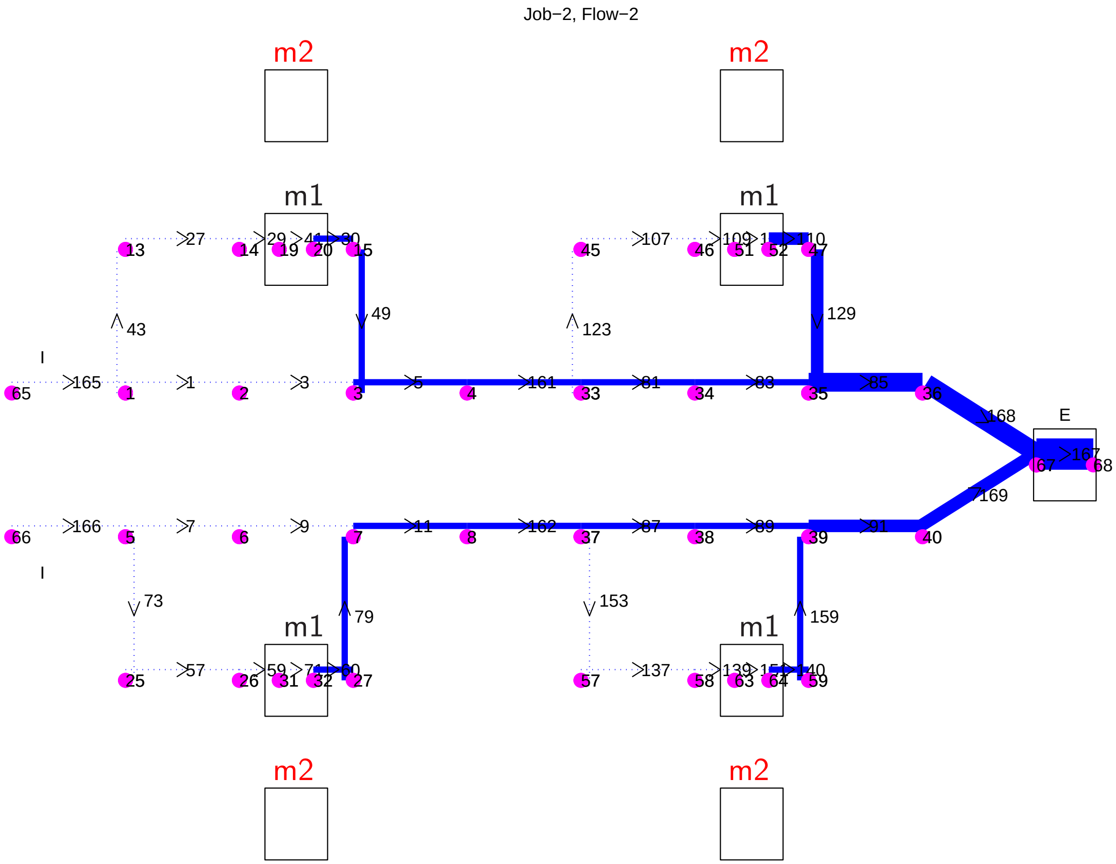
Job1: I $\rightarrow$ m1 $\rightarrow$ m2 $\rightarrow$ m1 $\rightarrow$ E



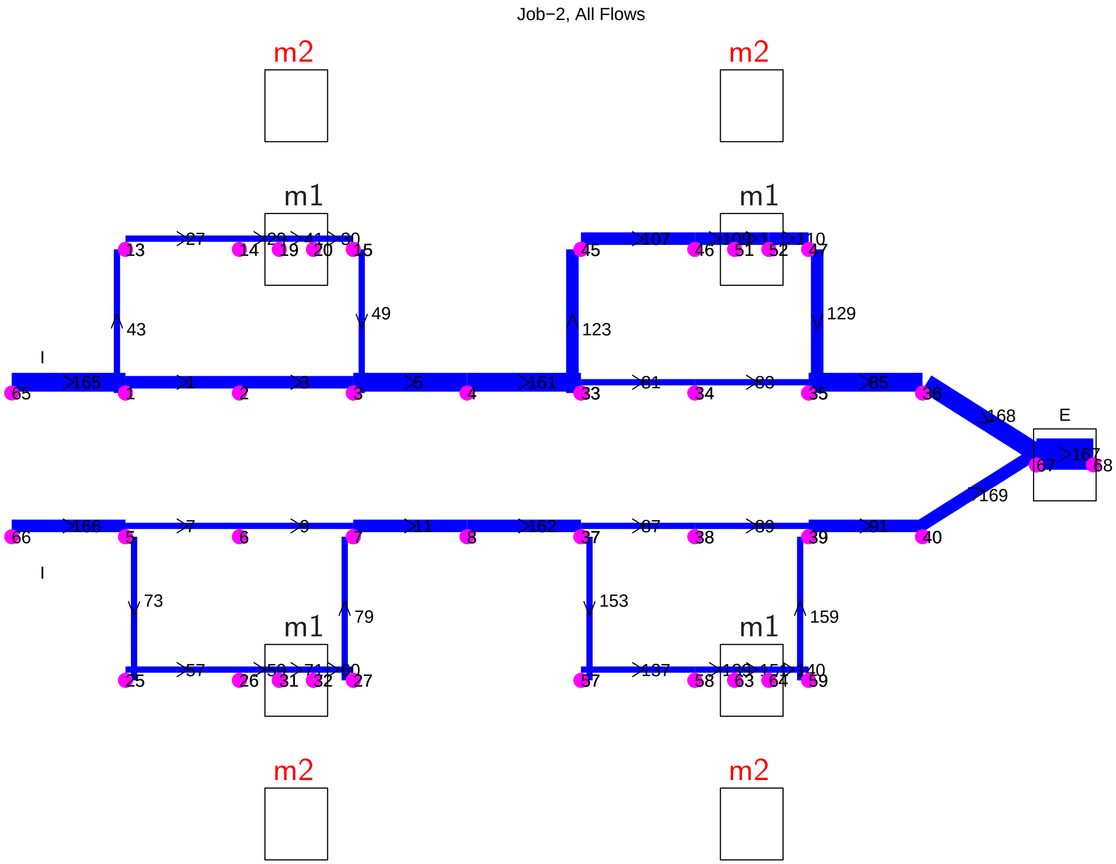
Job2: I → m1 → E



Job2: I → m1 → E



Job2: I → m1 → E



Conclusion

Presented multistage multicommodity network flow model for FMS

- captures notion of steady state throughput
- gives bounds on best achievable performance
- system design tool for large complex FMS networks
- based on efficient convex optimization
- for large capacity interconnects MMNF produces simple intuitive routes
- could use as heuristic in discrete planner
- enables rapid evaluation of failure scenarios & system architectures & sensitivity analysis
- other applications $e.g.$: transportation systems, reconfigurable FMS, etc