NONPARAMETRIC (GENERALIZED) JACKSON NETWORKS

MJP models that preserve **product-form**

- (locally) *State-dependent service rates*: Allows one to incorporate multi-server and infinite-server nodes. The "bottom-line" is the same as before: in equilibrium, stations are independent, and in isolation, each "behaves *like*" the corresponding single-station model (e.g., M/M/m_j or M/M/∞).

  Single-class models with state-dependent arrivals, services, and transition probabilities, have been *approximated* by fluid and diffusion models.

- *Multi-class networks*: Allows a heterogeneous customer population. Product form is preserved if *service rates* are *associated with servers*, not with customer classes. (They are allowed to depend on the *total* number of customers in a queue.)

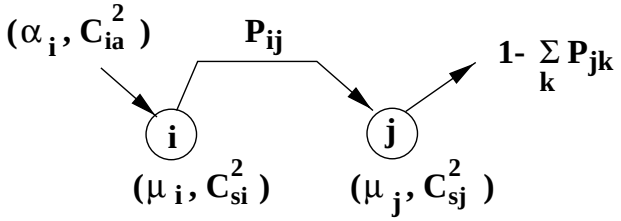
  Multi-class models with services that depend on the class are complicated, and provide a current research-challenge. (Even stability is not yet well understood!)

Nonproduct form: Nonparametric (Generalized) Jackson network

- Single-class;
- Arrival processes that are renewal (iid interarrival times);
- Services that are iid;
- Independence of mechanisms, as before.

Decomposition Approximations

- K service stations;
- External arrivals to station j: Renewal (α_j, C_{aj}^2), that is, iid interarrival times with mean $1/\alpha_j$ and $CV^2 = C_{aj}^2$;
- Services at station j: iid with mean $E(S^j) = 1/\mu_j,\ CV^2 = C_{sj}^2$;
- Routing: Markovian $P = [P_{ij}]$;
- Independence of mechanisms of arrivals, services, routing, as before.



Traffic equations, as before, with solution (λ_j) that exposes *bottlenecks*.

Assume $\lambda_j < \mu_j,\ j = 1, \ldots, K$, or $\rho_j = \lambda_j/\mu_j < 1$.

Then, under some (mostly technical) conditions, the network is ergodic. However, the network model is not amenable to exact analysis. Thus, one must resort to *approximations* (of an "exact" (general) model, vs. *exact* analysis of an approximate (special) model).

Generic Decomposition Approximation

$$\boxed{E(W_q^j) = E(S^j)\ \frac{\rho_j}{1-\rho_j}\ C_{Mj}^2} \qquad j = 1, \ldots, K \quad ,$$

where C_{Mj}^2 is an approximate measure of the variation at j, corresponding to the **M**ethod employed.

Average delay of a *marked customer*, the route of which constitutes V^j visits to station j, $j = 1, \ldots, K$, is given by

$$E(T) = \sum_{j=1}^{K} E(V^j)[E(S^j) + E(W_q^j)] \ .$$

Example: For a customer entering at station i,

$$E(T_i) = \sum_{j=1}^{K} R_{ij}[E(S^j) + E(W_q^j)], \quad R = [I - P]^{-1}.$$

Recall: R_{ij} = average number of visits to j by a customer who starts at i.

To approximate the *distribution* of delay, under FIFO, use *cautiously* the following guidelines:

- Exponential *law* of congestion: $W_q^j \sim$ exp (mean as above).
- Snapshot *principle*: Workloads (and queues) "do not" change over the duration of a visit (snapshot of the state upon arrival).
- Independence *assumption*: Given the route, delays at different stations are conditionally independent.

Example: The delay T during the route $3 \to 7 \to 2 \to 7$ is

$$T = S^3 + S_1^7 + S^2 + S_2^7 + W_q^3 + 2W_q^7 + W_q^2 \ ,$$

where S^j are service-times, W_q^j are exponentials as above, and the summands are all independent.

Remark: The above is justified if $C_a^2 = C_s^2 \equiv 1$, and *more*!

Possible Approximation Schemes:

0. Use data to approximate C_{Mj}^2.
1. QNA (Whitt, 1983–95; see Hall, §10.6, pg. 390–95).
2. Q-Net = Brownian approximations (Harrison; Nguyen, Dai,...).
3. Bottleneck decompositions (Reiman; Nguyen, Dai,...).

QNA = Queueing Network Analyzer (Whitt, 1983–95)

Approximation: each station "is" GI/G/m;
stations are independent;
customer flows are renewal processes;
Two-moment approximations $\Rightarrow (\lambda, C_a^2, \mu, C_s^2, m)$?

Primitives: exogenous arrivals (α_j, C_{0j}^2);
services (μ_j, C_{sj}^2);
routing $P = [P_{ij}]$ $(P_{0j} = \alpha_j/\alpha\ , \quad \alpha = \sum_{j=1}^K \alpha_j)$.

Parameters: (λ_j) via Traffic Equations;
(C_{aj}^2) via Variability Equations (below).

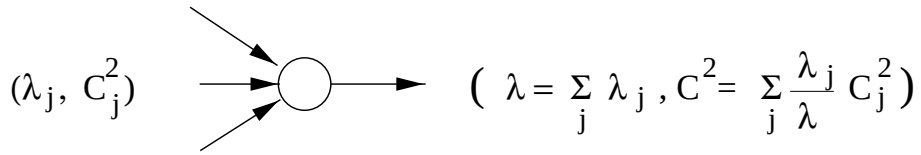
Network MOP's Restrict to Averages

$E(L)$ is deduced immediately from $E(W)$ via Little.

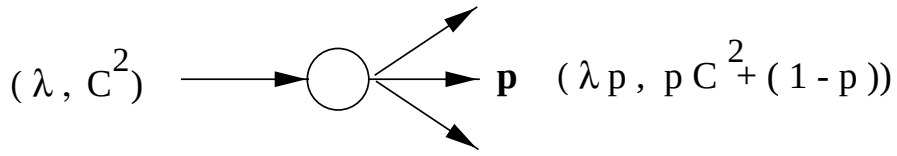
Delay statistics computed based on the laws and principles, and making the assumptions, all as described above.

Network Calculus (QNA)

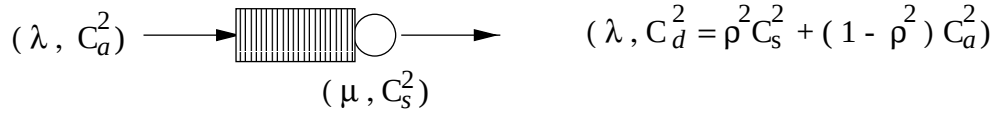
Superposition (merging)



Decomposition (splitting)



Departures (flow through)



Traffic equations $\lambda_j = \alpha_j + \sum_{i=1}^{K} \lambda_i P_{ij}\ , \quad j = 1, \ldots, K.$

Variability equations $C_{aj}^2 = a_j + \sum_{i=1}^{K} C_{ai}^2 b_{ij}$, where a_j, b_{ij} determined as follows:

First, assume $P_{ii} = 0$ (Whitt has a reduction scheme for that.)

Let $\boxed{q_{ij} = \dfrac{\lambda_i P_{ij}}{\lambda_j}}$ fraction of arrivals to j coming directly from i
$(q_{0j} = \alpha P_{0j}/\lambda_j = \alpha_j/\lambda_j\ ;\ P_{0j} = \alpha_j/\alpha)$.

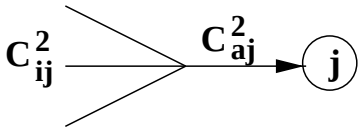
$C_{aj}^2 = CV^2$ of arrivals to $j,\ j = 1, \ldots, K$.

$C_{dj}^2 = CV^2$ of departures from $j,\ j = 1, \ldots, K$.

$C_{ij}^2 = CV^2$ of flow between i and $j,\ i = 0, \ldots, K,\ j = 1, \ldots, K$.

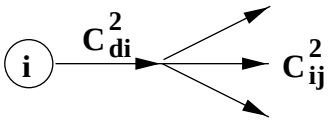
($C_{0j}^2 = CV^2$ of exogeneous interarrival times.)

1. *Arrivals* to j are *superposition* of flows from $i \neq j$ to j.



$$\boxed{C_{aj}^2 = \sum_{i=0}^{K} q_{ij} C_{ij}^2}$$

2. *Flows* between i and j are *decomposition* of departures from i.



$$\boxed{C_{ij}^2 = P_{ij} C_{di}^2 + (1 - P_{ij})}$$

3. *Departures* from j are related to arrivals by

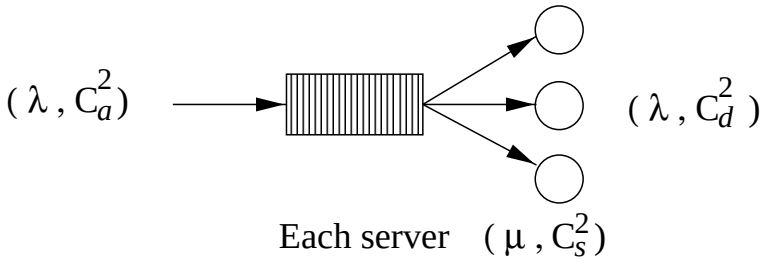
$$\boxed{C_{dj}^2 = \rho_j^2 C_{sj}^2 + (1 - \rho_j^2) C_{aj}^2}$$

(1) + (2) + (3) yield the Variability Equations.

Remarks

1. With $C_{0j}^2 = C_{sj}^2 \equiv 1$ (as in Jackson), we get $C_{aj}^2 \equiv 1$, as we should.

2. For a *multi-server* station, one must modify only the departure scheme:



where $\boxed{C_d^2 = 1 + (1-\rho^2)(C_a^2 - 1) + \dfrac{\rho^2}{\sqrt{m}}\,(C_s^2 - 1)}$

This reduces to the previous scheme when $m = 1$.

For M/M/m and M/G/∞ we get $C_d^2 = 1$, as we should.

3. *Heavy Traffic*: $C_{dj}^2 \approx C_{sj}^2$ when $\rho_j \approx 1$, in which case

$$\boxed{C_{aj}^2 = \frac{\alpha_j}{\lambda_j}\, C_{0j}^2 + \sum_{i \neq j} \frac{\lambda_i P_{ij}}{\lambda_j}\, [P_{ij} C_{sj}^2 + 1 - P_{ij}]}$$

is an *explicit* expression for C_{aj}^2 (a Brownian approximation). Thus, there is no need to solve any equations here. Simply use the parameters

$$(\lambda_j, C_{aj}^2, \mu_j, C_{sj}^2, m_j)\ , \quad j = 1, \ldots, k\ ,$$

to approximate individually each station. Then calculate system performance as articulated previously.

4. *History*: QNA has had 3 stages of development, as far as I know:

   4.1 The original version, in [1983], which is a *refined* version of the above.

   4.2 Refinements tailored at multi-type models (1988–1994).

   4.3 Enhancements that develop $C_a^2(\rho)$ (vs. C_a^2), where ρ is the traffic intensity of the queue which the arrivals join [1995].