Chapter 6
Multiple Product Factory Models

Most manufacturing facilities are setup to produce more than a single product. Even in the case of single product facilities, if the product visits a workstation more than once with different processing times at each visit, then the workstation sees the equivalent of multiple products. Such revisiting production schemes, called re-entrant flow systems, are prevalent in the semiconductor industry where it is not unusual for a product to be routed to the same machine group for distinct processing 20 or more times.

Modeling multiple product facilities is not significantly more difficult than single product models. There are two basic principles to keep in mind. First, the workload on a workstation is, as before, the sum of all the visits multiplied by the processing time per visit. This concept was introduced in the previous chapter (see p. 143) and since we use it in a more general setting here, we give a formal definition.

Definition 6.1. The offered workload (or simply the workload) of a workstation is the total amount of work that is required of a workstation per unit of time. The workload is determined by the sum of the total arrival rate (per hour) for each product type multiplied by its associated mean processing time (in hours). For purposes of determining workload, when a specific product type revisits a workstation, it is considered as a separate product type.

The second basic principle is that job flow needs to be maintained by product type. That is, the number of visits to each workstation by product class is needed. Different products can have different probabilistic flows through the production facility as well as different processing time characteristics. Hence, the number of visits to each workstation by product needs to be developed. This analysis requires the solution of a network flow system of equations by product. Here again as was done in the preceding chapter, the processing time is assumed to follow the same distribution for each product on each visit to a given workstation (of course due to randomness, the actual processing times will vary even though the distribution is the same). The re-entrant flow situation with different processing distributions per visit requires a different modeling paradigm that is taken up in Sect. 6.5.
6.1 Product Flow Rates

To compute the workload on the workstation, the number of visits to the workstation by each product is computed first. This requires an analysis for each product similar to that performed in Sect. 5.4.1 for a single product. A method of distinguishing between products visiting the same workstation is required. Previously a subscript was used to denote the workstation visited by a product so that \( \lambda_k \) denoted the arrival rate of jobs to Workstation \( k \). Two subscripts will now be used to distinguish among the various product types; thus, \( \lambda_{i,k} \) is the arrival rate of Product Type \( i \) to Workstation \( k \). Since a single subscript refers to a workstation, we will use a superscript when a single index refers to a product type; thus, \( \lambda^i \) is a vector giving the total arrival rates of Product Type \( i \) into each workstation so that the \( k^{th} \) component of the vector \( \lambda^i \) is \( \lambda_{i,k} \).

Arrivals from an external source are denoted as before by \( \gamma \) so that \( \gamma_{i,k} \) is the external arrival rate of Product \( i \) into Workstation \( k \). Additionally a workstation branching probability matrix for each product type will be needed. Since it is standard to already use two subscripts for this matrix of probabilities, the product type will be denoted by a superscript such as \( p_{i,jk} \) meaning the probability that an individual item of Product \( i \) leaving Workstation \( j \) goes to Workstation \( k \). The matrix of these probabilities for Product \( i \) is denoted as \( P^i \).

With the above notation, we can rewrite Property 5.7 so that it applies to more than one product type.

**Property 6.1.** Consider a factory of \( n \) workstations where Product Type \( i \) follows the switching rule defined by the routing matrix \( P^i \) and assume that the sum of at least one row of \( P^i \) is strictly less than one (i.e., jobs exit the network from at least one workstation). Let \( \gamma^i = (\gamma_{i,1}, \ldots, \gamma_{i,n}) \) denote a vector consisting of the mean arrival rate of Type \( i \) jobs from an external source to the workstations. Both \( P^i \) and \( \gamma^i \) are known. Let \( \lambda^i = (\lambda_{i,1}, \ldots, \lambda_{i,n}) \) be the (unknown) vector denoting mean arrival rate of all Type \( i \) jobs to the workstations. The vector \( \lambda^i \) is given by

\[
\lambda^i = (I - (P^i)^T)^{-1} \gamma^i,
\]

where \( I \) is an \( n \times n \) identity matrix and \( (P^i)^T \) is the transpose of \( P^i \).

Once the arrival rates for the various product types have been determined, the total arrival rate of jobs to Workstation \( k \) is given by the sum of the different product types; that is

\[
\lambda_k = \sum_{i=1}^{m} \lambda_{i,k},
\]

where \( m \) is the total number of product types within the factory.