ISSN : 1978-774X

The objective is to minimize the total tme necessary to process all the assemblies. Constraint (2) states that the completion time of the last operation (k) for $P_{i(j-1)}$ on machine m has to be less than or equal to the start time of the first operation ($k=1$) for P_{ij} on machine m' (provided that P_{ij} is the direct parent of $P_{i(j-1)}$ and $m' = m_{ijk}$. Constraint (3) expresses the operation relationship, while constraint (4) states that the completion time of the K^{th} operation for P_{i0} on machine m has to be less than or equal to production flow time F if K is the last operation. Constraint (5) expresses the processing time, while expression (6) and (7) ensure that no two jobs can be processed simultaneously on the same machine. Expression (8) is the integrality requirement on $X_{ijkqrsm}$ and (9) express that all product batch sizes are integer.

SOLUTION METHODOLOGY

The goal of the solution methodology is to generate efficient optimal or near optimal solutions to problems of practical size. A way of accomplishing this is by fixing all of the 0-1 variables. Fixing these variables would cause the disjunctive constraints in the MILP formulation to be replaced with simple precedence constraints (consisting of start and completion times). Disjunctive constraints arise naturally in scheduling problems where jobs have to share a machine and the order in which they are to be processed is not specified. Replacing all of the disjunctive constraints causes a reduction in the size of the problem and thus making it easier to solve.

<u>Heuristic</u>

Step 1. Collect the data for all jobs to be scheduled

Step 2. Decompose each job into its individual sub-job and their operations

Step 3. Calculate all Rt_{ijkm}

Step 4. Group the operations according to machine assignment. If the jobs are to be processed on M machines, create M columns and label the head of each column accordingly. A column represents a machine. Within each column, create three sub-columns and label them P_{ijkm}, Rt_{ijkm} and t_{ijkm} respectively.

Step 5. Partition the operations into sub-levels within each column.

Step 6. On a column by column basis, arrange the operations (within each sub-level only) in non-increasing order of Rt_{ijkm}. In the event of a tie, the selection is arbitrary.

Step 7. On a column by column basis, within the lowest sub-level, consider each operation and its position relative to its parallel related operation (operation are considered to be parallel related if their individual paths lead to the same direct parent). If both operations are competing on the same machine, rearrange them according to shortest processing time order by swapping their positions if necessary. Otherwise their relative positions remain the same. When the ordering is completed, calculate the completion time.

Step 8. For the higher sub-level, rearrange the operations for sequencing according to the completion time of their respective predecessors. When the ordering is completed, calculate the completion time. Always check the current sub-level and the ones below to see if there are operations that can be moved ahead of another to fill machine idle time.

Step 9. Repeat step 8 for all remaining sub-levels and final level.

Step 10. For each machine (column) convert the operation assignments into simple precedence constraints consisting of completion and start time.

Step 11. Formulate the problem using the MILP formulation, but replace the disjunctive constraints with the equation in step 10.

Step 12. Solve the problem using a LP solver.