

**A Mathematical Model and Solution Methodologies
for Optimal Process Planning
on Four- Axis CNC Turning Centers**

Vatin Chalermdamrichai ¹, Dharmaraj Veeramani ¹, and Robert Meyer ²

1. Department of Industrial Engineering

University of Wisconsin-Madison

1513 University Avenue, Madison, WI 53706

Tel : (608) 262-0861, Fax : (608) 262-8454

Email : raj@ie.engr.wisc.edu

2. Computer Sciences Department

University of Wisconsin-Madison

1210 W. Dayton St., Madison, WI 53706

Abstract

A new generation of Computer Numerical Control (CNC) machines, called four-axis turning centers has emerged and has been growing in use due to its versatility. Multiple turrets and spindles, automated material handling, and mill/turn capabilities on these machines enhance their processing capacity and lead to substantial improvement both in quality and productivity for turned components. However, optimal process planning on these machines is considerably complicated by their improved capabilities. No current commercial Computer-Aided Process Planning (CAPP) systems have considered constraint generation or process plan optimization for this class of machines. In this paper, the optimal process planning problem for four-axis turning centers is formulated as a Mixed-Integer Programming (MIP) model. The formulation is coded in GAMS (General Algebraic Modeling System) modeling language and the Branch-and-Bound (BB) algorithm is used to solve the problem. Experimental results with BB are presented as well as a comparison of results obtained from additional experiments with a partitioned-based heuristic, the Nested Partitions algorithm (Chalermdamrichai, Veeramani, and Shi 1999).

1. Problem Background and Motivation

Manufacturing today places a strong emphasis on the ability to produce highly customized and superior quality products. To maintain competitive advantage, therefore, a company needs to respond to customers in a timely and cost-effective manner. Within this context, a new class of CNC mill-turn centers known as four-axis turning centers was introduced in the late eighties and has been recognized as one of the most versatile CNC machines in the marketplace. Several

variations of these machines exist but the most widely used configuration is the dual-spindle, dual-opposing-turret turning center shown in Figure 1.

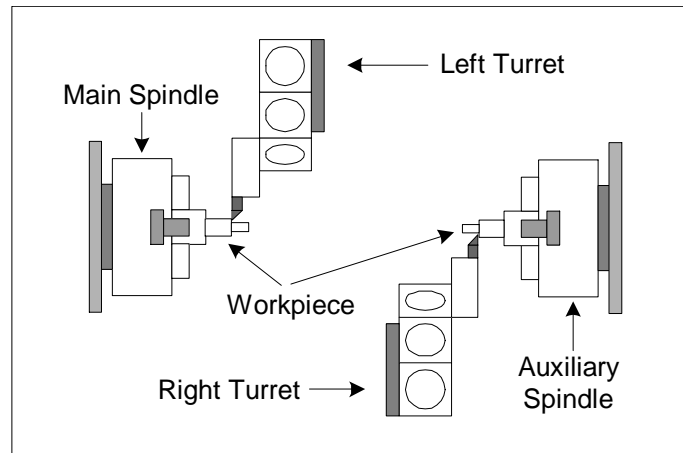


Figure 1: Dual-Spindle, Dual-Opposing-Turret Four-Axis Turning Centers

The distinctive features of these machines are multiple spindles and turrets, automated material handling, and live-tooling, which enable them to machine complex turned components in a single setup and perform both parallel and simultaneous machining. Parallel machining occurs when these turning centers perform machining operations on two different workpieces in parallel. Therefore, production throughput can potentially be increased two-fold (see Figure 1). In simultaneous machining, two cutting-tools held by two independent turrets are used concurrently to perform machining operations at a single workpiece, resulting in increased material removal rate. Two types of simultaneous machining exist, but one-feature simultaneous machining (Figure 2 in which both turrets work on one task) is used more extensively than two-feature simultaneous machining (Figure 3 in which the turrets perform distinct tasks) due to its added benefits of cutting-force balancing and reduced vibration and deflection, especially for long and slender parts.

In a typical batch-processing environment, a part is first held by the main spindle to machine one end of the part. The part is then transferred via an automated material handling mechanism to the auxiliary spindle to complete operations on the other end while, concurrently, the next part is loaded onto the main spindle to start operations on the new part. If the processing time at one of

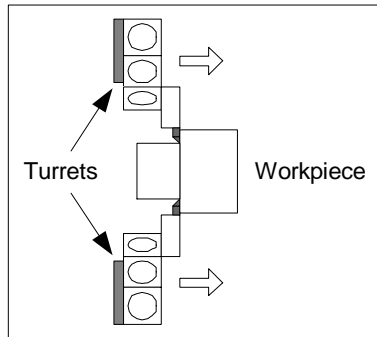


Figure 2: One-Feature Simultaneous Machining

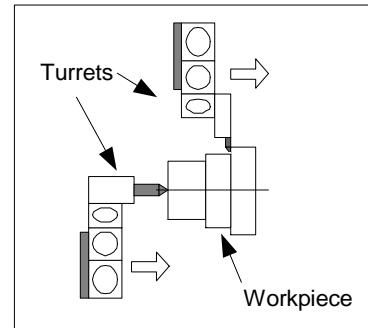


Figure 3: Two-Feature Simultaneous Machining

the spindles is shorter, idle time will be incurred since the part at that particular spindle has to wait for the part at the opposite spindle. Therefore, the overall cycle time is determined by the longer of the processing times at the two spindles. Figure 4 illustrates a typical manufacturing cycle of a four-axis turning center in the batch-processing mode.

2. Optimal Process Planning Problem Description

The goal of the process plan optimization problem is to minimize the overall cycle time. The problem is subject to the following principal constraints. (A full description of all constraints is presented in the appendix.)

1. Precedence Constraints: Precedence constraints govern the valid sequence of operations on both spindles. In essence, predecessor operations must be completed before successor operations can be performed.

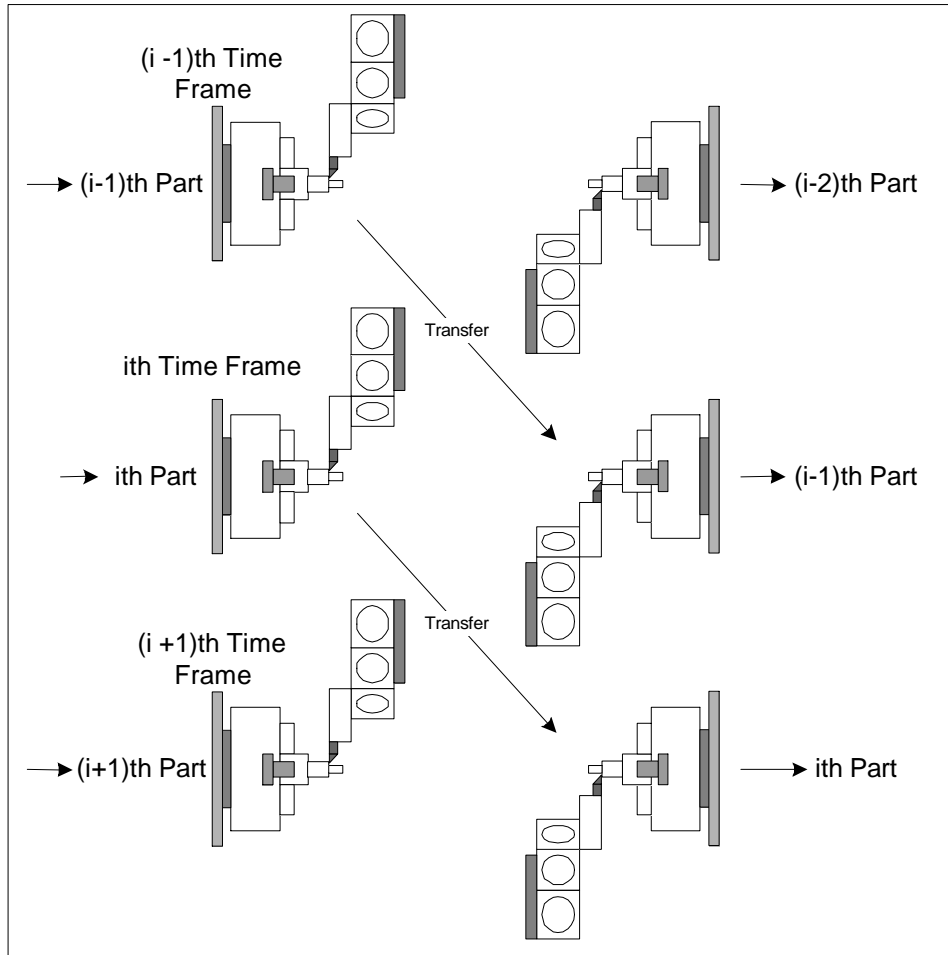


Figure 4: Typical Manufacturing Cycle of Four-Axis CNC Turning Centers

2. Setup Constraints: Setup constraints affect the assignment of operations to spindles due to fixturing and accessibility limitations. An operation may not be performed on a particular spindle if the tool is obstructed by the way that the chuck holds the workpiece. By reallocating unconstrained operations to the appropriate spindle, cycle time reduction may be achieved due to better balance of processing times at both spindles.

3. Simultaneous Machining Constraints: Simultaneous machining constraints determine suitability for simultaneous machining. Technologically, cross-drilling and turning may not be

performed simultaneously. Other feasibility factors such as spindle speed requirements may not allow two operations to be performed in simultaneous machining mode. By applying simultaneous machining, the total processing time at the critical spindle (i.e., spindle with longer processing time) is decreased whereas the time at the uncritical spindle is increased. The better balance gained by application of simultaneous machining stems from more productive utilization of both turrets.

4. Tooling constraints: The inability to assign a particular tool to a turret because of capacity, size, and potential for interference is considered as a tooling constraint. Better allocations of tools to turrets may decrease overall cycle time. For instance, two operations that are performed at different spindles using the same turret cannot overlap. If the tool for one operation can be assigned to the other turret, the waiting or idle time can be eliminated.

Problem inputs include list of required machining operations (along with their associated cutting parameters and processing times) as well as relevant constraints due to setup restrictions, precedence relationships, suitability for simultaneous machining, and tooling related constraints.

3. Prior Work

University of Michigan's research group has employed a modified Giffler-Thompson approach to determine the assignment of operations to the turrets (Yip-Hoi 1994). The fundamental limitation of that work lies in the absence of one-feature simultaneous machining consideration, which is often used in the manufacturing industry as a significant means for cycle time reduction. In their more recent work, a genetic algorithm is adapted to solve the process planning problem (Yip-Hoi 1996). Nevertheless, it still fails to recognize the application of one-feature simultaneous

machining. In addition, unconstrained operations (i.e., operations that are not restricted to perform at either spindle) are not considered in their work.

The research group at University of Wisconsin has developed a comprehensive framework for a computer-intelligent hybrid user-interactive CAPP system for four-axis turning centers. The Tabu Search heuristic and the Nested Partitions (NP) algorithm yield promising results (Stinnes 1995, Sanghi 1997, Chalermdamrichai *et al* 1999). Tabu Search is a heuristic approach that prevents solutions from getting trapped in local optima by tabuing recently visited solutions for some time (Glover 1993). Those solutions are excluded from neighborhood search during the tabued period and non-improving solutions can be accepted if no improvements can be found. The NP algorithm is an adaptive randomized search algorithm that divides the solution space into subregions and focuses its computational effort on the most promising regions (Shi and Olafsson 1998).

4. Optimal Process Planning Problem Formulation

In this paper, the problem is formulated as a Mixed-Integer Programming (MIP) model. We consider both a complete model embodying all of the constraints described above, and a relaxed model that uses a subset of those constraints. The model is coded in the GAMS (General Algebraic Modeling System) modeling language and runs on a Unix workstation using CPLEX (GAMS CPLEX 6.5 User Notes 1999), which utilizes the Branch-and-Bound algorithm, as a solver. The complete formulation is described in Appendix A, which includes the overall concept of the formulation along with relevant decision variables, set and parameter definitions, and constraints.

5. Main Results

Small-scale problems (i.e., lower in number of operations compared to typical industry-based parts) are used to demonstrate the performance and validity of the complete model.

Table 1 shows the data for the tested part. The loading/unloading and part transfer time as well as the turret indexing time are machine specific and, thus, are treated as constants in our problem. The parameters used for the loading, unloading, and part transfer times are 5, 5, and 10 units respectively. The turret indexing time is 1 if the next operation uses the same tool, and 2 if it has to index a new tool. Further, the turret switch time (i.e., time for the turret to travel to the opposite spindle) is assumed to be 2 units. These parameters approximate the new generation of CNC turning centers with high speed indexing and part transfer mechanisms. All tools are unconstrained (i.e., can be allocated to either turret).

Several experiments have been conducted and the main results are summarized in

Table 2. Some Branch-and-Bound strategies available in CPLEX (GAMS CPLEX 6.5 User Notes, 1999) such as node selection strategies were also investigated. However, alternative strategies yielded no noticeable improvement in both solution quality and computing time. To ensure that optimality is attained, the tested part was initially run with 4 operations (i.e., the first 4 operations in

Table 1). The number of operations is gradually incremented in the subsequent experiments (5, 6, 7, and 8 operations) using data from Table 1.

Table 1: Part Data

Operation	Duration	Duration (one-feature)	Setup Restriction (main/aux/ unconstrained)	Partner Operation	Parent Operation	Tool
1	20	-	Main	2	-	1
2	5	-	Main	1	5	3
3	5	-	Unconstrained	-	-	4
4	5	-	Unconstrained	-	3	2
5	8	4	Unconstrained	-	6	2
6	15	8	Unconstrained	-	-	3
7	15	-	Main	-	1	5
8	5	-	Auxiliary	-	-	6

Table 2 Results from Complete MIP Model

Run	Number Of Operations	Variables	Constraints	Best Cycle Time Obtained	Time Used (secs)	Optimal Solution
1	4	172	973	40	5.73	Yes
2	5	247	1461	44	797.92	Yes
3	6	336	2049	49	23395.58	Yes
4	7	441	2739	88	50000*	No*
5	8	560	3529	84	50000*	No*

* 50,000 Second CPU Time Limit Reached

The Branch-and-Bound algorithm yield optimal solutions within the 50,000 second time limit when the problem size is small (in 4, 5, and 6 operations cases). However, the computing time

grows substantially as the problem size increases (the number of variables and constraints involved are $7n^2+12n+2t+4$ and $50n^2+38n+2t+13$ respectively where n and t are number of operations and tools) and, thus optimal results are not obtained within the time limit. The outcomes of the experiments suggest that the Branch-and-Bound algorithm has difficulty finding optimal solutions when the problem size is relatively large. The integration of the Branch-and-Bound algorithm with other heuristics/algorithms can potentially improve the computational time and results and is to be investigated.

In addition, lower bounds of cycle times obtained from a relaxed MIP formulation are used to measure solution quality for the NP algorithm to be discussed below. The relaxed MIP formulation does not consider precedence, setup, and tooling constraints, and, hence, yields cycle times that are not attainable in most cases. However, these lower bounds are close to optimal values and serve to establish the high quality of the NP results. Moreover, the run times for the relaxed MIP are in the order of a few seconds in all these cases.

Given the complete and relaxed models, an interesting alternative approach would be a “cutting plane” approach, starting with the fast solution of the relaxed MIP model, and then adding in a small number of violated constraints from the complete MIP. This process could be repeated until optimality is achieved by verifying feasibility for the complete problem (or terminated when computing limits were reached).

5.1 Comparison of Results from the Nested Partitions (NP) Algorithm

Chalermdamrichai (1999) has also developed an implementation of the Nested Partitions (NP) algorithm for four-axis CNC turning centers. The NP algorithm is an adaptive randomized search

that employs a divide-and-conquer strategy (like BB) but applies heuristics (rather than linear programming) in the subregions and focuses its computational effort in the most promising regions (Shi and Olafsson, 1998). The NP procedure generates a Markov chain and converges with probability of one to global optimum. To provide a comparison with the Branch-and-Bound algorithm, the same set of experiments was run using the NP optimizer. The NP procedure is implemented in JAVA™ and runs on a Pentium™ II, 233 MHz, Windows NT based workstation. The NP algorithm generates a total of 50,000 alternative process plans in each experiment and is run 15 times for each case, using different random seeds. The results are presented in Table 3.

Table 3 Comparison of Results from the NP Algorithm

# Ops	Relaxed MIP Lower Bound	Average Cycle Time (NP)	Standard Deviation (NP)	Best Cycle Time (NP)	CPU Time (seconds) (NP)	Best Cycle Time (MIP)	CPU Time (seconds) (MIP)
4	37*	40.00	0.00	40	2	40 (optimal)	5.73
5	42*	44.00	0.00	44	3	44 (optimal)	797.92
6	49	50.47	0.92	49	5	49 (optimal)	23395.58
7	58	61.33	1.29	61	5	88	50000**
8	60	62.20	2.08	61	4	84	50000**

* Lower bound from the relaxed MIP model is lower than the optimal value of the complete MIP model

** Time Limit Reached

The results indicate that the NP algorithm is much more effective both in terms of solution quality and computing time and is able to generate optimal or near-optimal results obtained from the

Branch-and-Bound method. This is true even though these methods run on different platforms, since the computing time difference is substantially large. As the problem size grows, NP still produces solutions that are close to the lower bounds. However, the relaxed MIP formulation serves to establish the near optimality of the NP results. In the next section, we consider more realistic problems.

5.2 Results from the Nested Partitions (NP) Algorithm for Large-Scale Problems

The NP algorithm is tested using eleven industry-based sample parts varying in physical balance, number of operations, and relevant constraints. Table 4 shows data for the sample parts. For each part, one million alternative plans were generated via NP and cycle times as well as computing times were recorded. Our computational experiments indicate that no further benefit would be obtained from additional plans. The experiment for each part was run 15 times with different random seeds. The main results from the NP algorithm are presented in Table 5.

To assess the performance of the NP algorithm, two criteria, namely, a compromised deviation (CD) and the average CPU time, are employed. The first criterion represents the average value of the pessimistic and optimistic deviations. The pessimistic deviation (PD) is computed from the difference between the average cycle time (ACT) and the MIP lower bound (LB) with respect to the lower bound (i.e., $(ACT-LB)/LB*100$). Since optimal values may be higher than the lower bound, the pessimistic deviation may convey an unfair impression for the algorithm's performance. The optimistic deviation (OD) uses the feasible cycle time as a basis for computation (i.e., $(ACT-LB)/ACT*100$). By the same token, it alone may not best represent the true deviation from optimum. Thus, an average value is more likely to measure the performance of the algorithm more accurately. The average computing time is intended only to give a general

idea of the computational effort required to generate 1 million number of plans within the NP framework. Performance measures are summarized in Tables 5 and 6. In addition, the relationship of the two performance measures to the problem size (i.e., number of operations and tools) are presented in Figures 5 and 6.

Table 4: Industry-Based Sample Parts

Part ID	# Operations	# Tools	Relaxed MIP Lower Bound
1	16	4	70
2	14	5	83
3	37	8	172
4	20	5	82
5	25	8	161
6	14	7	87
7	22	10	252
8	21	11	189
9	24	10	206
10	15	10	191
11	28	10	249

Table 5: Main Results from the NP Algorithm

Part ID	Best Cycle Time	Average Cycle Time	Standard Deviation of Cycle Time	Average CPU Time (seconds)	Standard Deviation of CPU Time
1	80	82.40	0.91	116	7.56
2	85	85.07	0.26	94	5.23
3	190	197.80	5.19	185	10.28
4	91	92.53	0.74	122	5.52
5	174	181.27	3.65	110	4.33
6	92	92.80	1.08	97	4.01
7	257	262.33	3.89	114	7.84
8	193	199.60	3.83	120	11.07
9	283	288.40	2.95	139	9.98
10	194	194.53	0.52	103	8.01
11	261	271.87	4.91	122	9.97

Table 6 Performance Measures of the NP Algorithm

Part ID	Relaxed MIP Lower Bound	Best Cycle Time	Average Cycle Time	%PD	%OD	%CD	Average CPU Time (seconds)
1	70	80	82.40	17.71	15.05	16.38	116
2	83	85	85.07	2.49	2.43	2.46	94
3	172	190	197.80	15.00	13.04	14.02	185
4	82	91	92.53	12.84	11.38	12.11	122
5	161	174	181.27	12.59	11.18	11.89	110
6	87	92	92.80	6.67	6.25	6.46	97
7	252	257	262.33	4.10	3.94	4.02	114
8	189	193	199.60	5.61	5.31	5.46	120
9	206	283	288.40	40.00	28.57	34.29	139
10	191	194	194.53	1.85	1.81	1.83	103
11	249	261	271.87	9.18	8.41	8.80	122

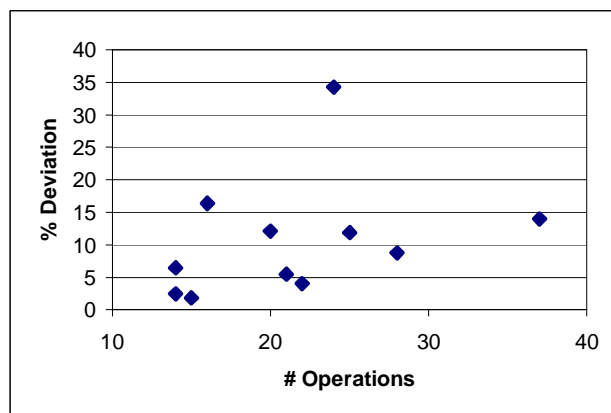


Figure 5: Problem Size (Number of Operations) Vs % Compromised Deviation

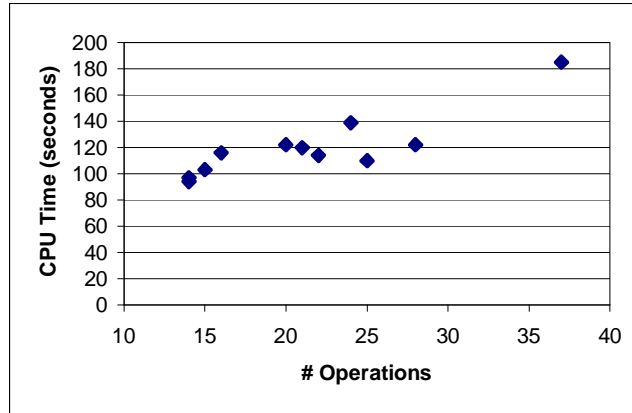


Figure 6: Problem Size (Number of Operations) Vs CPU Time

The average CPU times range from 94 to 185 seconds (standard deviations for CPU time are relatively small). As the problem size (number of operations) grows larger, the computing time increases (see Figure 6), but does not exhibit the rapid growth of the BB times.

The compromised deviations are between 1.83% and 34.29%, with 3 parts within 5%, 6 parts within 10%, and 9 parts (out of 11) are within 15% of their respective lower bounds. There is no evidence of correlation between problem size (i.e., number of operations) and % compromised deviation as illustrated in Figure 5. This suggests that solution quality does not rely on problem size.

6. Summary

Optimal process planning on four-axis turning centers is complex due to enhanced processing capabilities, which greatly increase the number of alternative process plans. A Mixed-Integer Programming (MIP) formulation of the problem has been developed and tested for validity. Although optimality was achieved in certain small cases, the results confirm that the Branch-and-Bound algorithm has difficulty obtaining good plans within reasonable computing

times. However, a smaller, relaxed MIP formulation was also generated by deleting several types of constraints. This relaxed MIP formulation was solved for process planning problems of realistic size, and served to provide high quality lower bounds. Benchmark results from the NP algorithm are also presented. The NP algorithm is a partition-based adaptive randomized search that focuses its computational effort on the most promising regions. Solution quality from the NP algorithm is very impressive compared to that of the complete MIP formulation. Additional experiments with eleven industry-based sample parts show that the NP method can efficiently provide high quality solutions rapidly. The results also indicate that while it is hard to guarantee optimal process planning for four-axis CNC turning centers, the combination of relaxed MIP and NP results serves to establish the near optimality of the NP results.

Acknowledgement

This research is funded by the National Science Foundation (grant No. DMI 9622530). In addition, DP Technology and Traub-Hermle have kindly provided needed data and information.

References

Chalermdamrichai V., "A Hybrid Computer-Intelligent, User-Interactive Process Plan Optimizer for Four-Axis CNC Turning Centers", *Ph.D. Thesis*, Industrial Engineering Department, University of Wisconsin-Madison, 1999.

Chalermdamrichai V., Veeramani D., and Shi L. "A Nested Partitions Algorithm for Optimal Process Planning on Four-Axis CNC Turning Centers", to be submitted 1999.

Koepfer C., "<http://www.mmsonline.com/articles/099503.html>", August 1999a.

Koepfer C., "<http://www.mmsonline.com/articles/099803.html>", September 1999b.

GAMS CPLEX 6.5 User Notes, "<http://www.gams.com/solvers/cplex/cplexman.htm#nodesel>", April, 1999.

Levin, J. B. and Dutta, D., "Computer-Aided Process Planning for Parallel Machines", *Journal of Manufacturing Systems*, Vol. 11, No. 2, pp. 79-92, 1991.

Levin, J. B., "PMPS - A Process Planning System for Parallel Machine Tools", *Master's Thesis*, University of Michigan, 1992

Sanghi, D., Generalized Process Plan Optimization Problem for Four-Axis CNC Turning Centers, *Master's Thesis*, University of Wisconsin-Madison, 1997.

Shi, L., and Olafsson, S., "Nested Partitions Method for Global Optimization", *Operations Research*, 1998 (in press).

Shi, L., Olafsson, S., and Sun, N., "A New Parallel Randomized Algorithm for Traveling Salesman Problem", *Computer and Operations Research*, 1999, Vol. 26, pp. 371-394.

Shi, L., Olafsson, S., and Chen, Q., "A New Hybrid Optimization Algorithm", *Computers and Industrial Engineering*, 1999, Vol. 36, pp.409-426.

Shi, L., and Olafsson, S., "Convergence Rate of the Nested Partitions Method for Stochastic Optimization", *Methodology and Computing in Applied Probability*, 1999 (accepted).

Stinnes, A. H., Intelligent and Optimized Process Planning for Four-Axis Turning Centers, *Master's Thesis*, University of Wisconsin-Madison, 1995.

Veeramani D. and Chalermdamrichai V., "A Novel Search Algorithm to the Process Plan Optimization Problem for Four-Axis Turning Centers," *2nd International Conference on Engineering Design and Automation*, August 9-12, 1998, Maui, Hawaii.

Veeramani, D., and Sanghi D., "Selection of an Optimal Family of Process Plans to Minimize Batch Processing Time on Four-Axis CNC Turning Centers", *Journal of Manufacturing Systems*, submitted, 1997a.

Veeramani, D., and Sanghi D., "Generalized Process Plan Optimization Problem for Four-Axis CNC Turning Centers ", *IIE Transactions on Design and Manufacturing*, submitted, 1997b.

Veeramani D., and Stinnes A., "Assessment of Manufacturability on Four-Axis Turning Centers", *Proceedings of the 1996 ASME Design-for-Manufacturability Conference*, Chicago, IL, March 18-20, 1996.

Veeramani D., and Stinnes A., "A Hybrid Computer-Intelligent and User-Interactive Process Planning Framework for Four-Axis CNC Turning Centers", *5th Industrial Engineering Research Conference Proceedings*, Minneapolis, MN, pp. 187-192, May 18-19, 1996a.

Veeramani, D., and Stinnes A., "A Hybrid Optimal Process Planning System for Four-Axis CNC Turning Centers", *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, CA, August 18-22, 1996b.

Veeramani, D., Stinnes, A., and Brown S. "Optimal Selection of Four-Axis CNC Turning Machine Configuration", *for submission to Journal of Manufacturing Systems*.

Veeramani D., Stinnes. A., and Sanghi D., "Application of Tabu Search for Process Plan Optimization of Four-Axis CNC Turning Centers", *International Journal of Production Research*, Vol 37, No. 16, pp. 3803-3822, 1999.

Yip-Hoi, D., and Dutta, D., "An Introduction to Parallel Machine Tools and Related CAPP Issues", *Advances in Features Based Manufacturing*, pp. 261-285, 1994.

Yip-Hoi, D., and Dutta, D., "A Genetic Algorithm Application for Sequencing Operations in Process Planning for Parallel Machining", *IIE Transactions*, 28, pp. 55-68, 1996.

Appendix A: The Mixed-Integer Programming Formulation

A1: Problem Formulation

The process plan optimization problem is, in essence, a combination of operations sequencing and setup assignment problems. It involves the following decisions:

1. Spindle Allocation of Operations
2. Tool Allocation to Turrets, hence Turret Assignment of Operations
3. Sequence of Operations on Both Spindles and Turrets
4. Applications of Simultaneous Machining

The formulation incorporates several variables to handle the decisions above. Three binary (zero-one) variables, namely $reg(i, s, t)$, $onesim(i, s, t)$, and $twosim(p, s, q)$ are used to account for spindle and turret allocations along with modes of operation (i.e., regular, one-feature simultaneous machining, and two-feature simultaneous machining). For instance, binary variable $reg(i, s, t)$ is one when operation i is assigned to spindle s and turret t and it is not performed in one-feature simultaneous machining mode. Similarly, variable $onesim(i, s, t)$ equals one when operation i is performed at spindle s by turret t in one-feature simultaneous machining mode. In application of two-feature simultaneous machining, zero-one variable $twosim(p, s, q)$ is used in conjunction with $reg(p, s, t_1)$ and $reg(q, s, t_2)$. When partner operations p and q are performed in two-feature simultaneous machining mode at spindle s , thus, $twosim(p, s, q)$, $reg(p, s, t_1)$ and $reg(q, s, t_2)$ will all be one's ($t_1, t_2 \in TURRET$ and $t_1 \neq t_2$). The reasons that variable $onesim(i, s, t)$ is not associated with $reg(p, s, t)$ is due to the fact that operations performed in one-feature simultaneous machining mode have duration changes due to increased material removal rate (i.e.,

processing times are shortened due to higher feed rate). On the other hand, operations performed in two-feature simultaneous machining mode result in no change of duration for involved operations. The operation sequences at the two turrets and spindles can be determined by two other binary variables which are $x(i, t, j)$ and $y(p, s, q)$ respectively. Variable $x(i, t, j)$ equals one when operations i and j are adjacent operations performed by turret t whereas $y(p, s, q)$ is one when operation p is performed prior to operation q at spindle s . Binary variable $tool(tl, t)$ is responsible for tool allocation to turret (i.e., $tool(tl, t) = 1$ when tool tl is allocated to turret t).

Since our goal for the optimization is to minimize cycle time, three positive variables that involves time including 1) $start(i, t)$, 2) $end(i, t)$, and 3) $indexing(i, j)$ are introduced into the formulation. The first variable defines the starting time of operation i on turret t . This implies that each operation has its own starting time on both the turrets. However, $start(i, t)$ is only meaningful when operation i is assigned to turret t . Variable $end(i, t)$ refers to the corresponding end time of operation i on turret t . Due to its dependency on operation sequence at each turret and spindle, turret indexing time cannot be predetermined. The turret indexing time includes the time for the turret to index a new tool, move to the desired cutting position, and travel to the appropriate spindle (i.e., when two operations requiring the same turret are performed at opposite spindles). Hence, variable $indexing(i, j)$ representing the total turret indexing time of operation i that is immediately followed by operation j is introduced. Free variables LT and RT refer to the processing times at the left and right spindles respectively. To define the objective function, another free variable CT which is the maximum value of LT and RT is used to determine the overall cycle time of a process plan. However, CT does not consider the part transfer time which is machine specific and is treated as a constant in the formulation. Therefore, another variable

CycleTime that includes part transfer time is used in the objective function which is reduced to minimizing *CycleTime*.

A2: Set Definitions

This section explains the definitions of sets used in the formulation.

1. $OPS \in \{ source, 1, 2, 3, \dots, n, sink \}$

Set of n operations with *source* and *sink*. Both *source* and *sink* are dummy operations required in the formulation.

2. $OP \in \{ 1, 2, 3, \dots, n \}$

Set of n operations.

3. $SPINDLE \in \{ main, aux \}$

Set of available spindles which are the main and auxiliary spindles.

4. $TURRET \in \{ left, right \}$

Set of available turrets which are the left and right turrets.

5. $MODE \in \{ regg, onef \}$

Set of modes of operations. This set is used to define processing times of operations. The processing times of operations performed in one-feature simultaneous are reduced due to higher material removal rate. However, the duration does not change when operations are performed in regular or two-feature simultaneous machining modes. Hence, there are only 2 members in this set.

6. $TOOLS \in \{ 1, 2, 3, \dots, m \}$

Set of m required tools.

A3: Variable Definitions

Decision variables that are employed in the formulation are described in this section. Three types of variables, namely binary (zero-one), positive, and free variables are involved.

1. Binary variable $reg(i, s, t)$; $i \in OPS, s \in SPINDLE, t \in TURRET$

$reg(i, s, t)$ is 1 when operation i is assigned to spindle s and is operated by turret t in regular mode and 0 otherwise.

2. Binary variable $onesim(i, s, t)$; $i \in OPS, s \in SPINDLE, t \in TURRET$

$onesim(i, s, t)$ is 1 when operation i is assigned to spindle s and is operated by turret t in one-feature simultaneous machining mode and 0 otherwise.

3. Binary variable $twosim(p, s, q)$; $p, q \in OP, s \in SPINDLE$

$twosim(p, s, q)$ is 1 when operation p is assigned to spindle s and q is the partner operation in two-feature simultaneous machining mode and 0 otherwise.

4. Binary variable $x(i, t, j)$; $i, j \in OPS, t \in TURRET$

$x(i, t, j)$ is 1 when operation i is operated by turret t and is immediately followed by operation j which is also operated by turret t and 0 otherwise.

5. Binary variable $y(p, s, q)$; $p, q \in OP, s \in SPINDLE$

$y(p, s, q)$ is 1 when operation p precedes operation q on spindle s and they are both performed at spindle s and is 0 otherwise. This variable is used to avoid the situation of having two operations that are allocated to the same spindle but different turrets been performed during the same time period unless they are performed in two-feature simultaneous machining mode.

6. Binary variable $tool(tl, t)$; $tl \in TOOLS, t \in TURRET$

$tool(tl, t)$ is 1 when tool tl is allocated to turret t and 0 otherwise.

Positive variable $start(i, t); i \in OPS, t \in TURRET$

7. $start(i, t)$ is the starting time of operation i on turret t .

Positive variable $end(i, t); i \in OPS, t \in TURRET$

8. $end(i, t)$ is the end time of operation i on turret t .

Positive variable $indexing(i, j); i, j \in OPS$

9. $indexing(i, j)$ refers to the indexing time of operation i when it is immediately followed by operation j .

10. Free variable LT

Cycle Time (processing time) at the main spindle (left cycle time).

11. Free variable RT

Cycle Time (processing time) at the auxiliary spindle (right cycle time).

12. Free variable CT

Overall cycle time excluding Transfer time.

13. Free variable $CycleTime$

Over all cycle time.

A4: Parameter Definitions

This section summarizes the parameters used in the formulation.

1. $REGULAR(i, s, t); i \in OPS, s \in SPINDLE, t \in TURRET$

$REGULAR(i, s, t) = 1$ if operation i can be assigned to spindle s and turret t and 0 otherwise.

2. $ONEFEATURE(i, s, t); i \in OPS, s \in SPINDLE, t \in TURRET$

$ONEFEATURE(i, s, t) = 1$ if operation i can be assigned to spindle s and turret t and is performed in one-feature simultaneous machining mode and 0 otherwise.

3. $TWOFEATURE(p, q); p, q \in OP$

$TWOFEATURE(p, q) = 1$ if operation p can be performed in two-feature simultaneous machining mode with operation q as a partner and 0 otherwise.

4. $PRECEDENCE(i, j); i, j \in OPS$

$PRECEDENCE(i, j) = 1$ if operation i must be performed before operation j and 0 otherwise.

5. $INDEX(i, j); i, j \in OPS$

$INDEX(i, j)$ is the turret Indexing time between operation i and j excluding turret switch. This parameter can be predetermined since tool requirements for all operations are known. The turret switch (i.e., travel time from one spindle to another) due to different spindle assignments for operations i and j will be taken into account by variable $indexing(i, j)$.

6. $DURATION(i, m); i \in OPS, m \in MODE$

Duration of operation i performed in mode m .

7. $TURRETOK(tl, t); tl \in TOOLS, t \in TURRET$

$TURRETOK(tl, t) = 1$ if tool tl can be assigned to turret t and 0 otherwise.

8. $OPTOOL(p); p \in OP$

$OPTOOL(p) = k$ when operation p requires tool k .

9. $bigM$

A large number used in the formulation.

10. $LOAD$

Loading time.

11. $UNLOAD$

Unloading time.

12. $TRANSFER$

Transfer time.

13. MAXTOOLS

Turret capacity (per turret).

14. TURRETSWITCH

Turret travel time from one spindle to another.

A5: Constraints

The complete formulation entails all relevant constraints including setup restrictions, precedence relationships, suitability for simultaneous machining, and tooling related constraints. The constraints can be divided into 9 categories as follow:

1. Objective Function Related Constraints
2. Setup Constraints
3. Precedence Relationships
4. Two-Feature Simultaneous Machining Constraints
5. Tooling Constraints
6. Operation Sequence Constraints
7. Time Sequence Related Constraints
8. Turret Synchronization Related Constraints
9. Miscellaneous Constraints

A5.1: Objective Function Related Constraints

$$\min \text{CycleTime}$$

Equation 1: Objective Function

$$CT \geq LT$$

Equation 2

$$CT \geq RT$$

Equation 3

$$\text{CycleTime} = CT + \text{TRANSFER}$$

Equation 4

The overall cycle time for four-axis CNC turning centers is determined by the longer of the processing times at both spindles. The above objection function and Equations 2 through 4 are collectively used to define the goal of optimization. Equations 2 and 3 define CT as the longer of the cycle times at the main spindle (LT) and the auxiliary spindle (RT). Equation 4 includes the transfer time which is a constant to define the overall cycle time.

A5.2 Setup Assignment Related Constraints

$$2 * \sum_{\substack{s \in SPINDLE \\ t \in TURRET \\ s.t. REGULAR(p,s,t)=1}} reg(p,s,t) + \sum_{\substack{s \in SPINDLE \\ t \in TURRET \\ s.t. ONEFEATURE(p,s,t)=1}} onesim(p,s,t) = 2$$

For $p \in OP$

Equation 5

$$reg(p,s,t) \leq REGULAR(p,s,t)$$

For $p \in OP, s \in SPINDLE, t \in TURRET$

Equation 6

$$onesim(p,s,t) \leq ONEFEATURE(p,s,t)$$

For $p \in OP, s \in SPINDLE, t \in TURRET$

Equation 7

These constraints are used to restrict setup assignments of operations. Equation 5 guarantees that each operation is performed exactly once at a spindle and turret and in one mode of operation only. In Equations 6 and 7, parameters associated with the decision variables are used to restrain the values of the variables. For instance, parameters $REGULAR(p, main, left)$ and $REGULAR(p, main, right)$ would be set to one's whereas $REGULAR(p, aux, left)$ and $REGULAR(p, aux, right)$ are set to zero's if operation p is allowed to be performed at the main spindle only. Thus, variables $reg(p, aux, left)$ and $reg(p, aux, right)$ are both forced to become zero so that operation p may not be allocated to the auxiliary spindle. The same notion applies to operations that are not suitable to perform in one-feature simultaneous machining mode as well (Equation 7).

A5.3: Precedence Relationships Related Constraints

$$\begin{aligned}
 & bigM * \left(2 - \left(2^* \sum_{t \in TURRET} reg(p, s, t) + \sum_{t \in TURRET} onesim(p, s, t) \right) \right) + \\
 & bigM * \left(2 - \left(2^* \sum_{t \in TURRET} reg(q, s, t) + \sum_{t \in TURRET} onesim(q, s, t) \right) \right) + \\
 & start(q, right) \geq end(p, left) \\
 & For \ p, q \in OP, \ s \in SPINDLE \ s.t. \ PRECEDENCE(p, q) = 1
 \end{aligned}$$

Equation 8

$$\begin{aligned}
 & bigM * \left(2 - \left(2^* \sum_{t \in TURRET} reg(p, s, t) + \sum_{t \in TURRET} onesim(p, s, t) \right) \right) + \\
 & bigM * \left(2 - \left(2^* \sum_{t \in TURRET} reg(q, s, t) + \sum_{t \in TURRET} onesim(q, s, t) \right) \right) + \\
 & start(q, left) \geq end(p, right) \\
 & For \ p, q \in OP, \ s \in SPINDLE \ s.t. \ PRECEDENCE(p, q) = 1
 \end{aligned}$$

Equation 9

$$\begin{aligned}
 & bigM * \left(2 - \left(2^* \sum_{t \in TURRET} reg(p, s, t) + \sum_{t \in TURRET} onesim(p, s, t) \right) \right) + \\
 & bigM * \left(2 - \left(2^* \sum_{t \in TURRET} reg(q, s, t) + \sum_{t \in TURRET} onesim(q, s, t) \right) \right) + \\
 & start(q, right) \geq end(p, right) \\
 & For \ p, q \in OP, \ s \in SPINDLE \ s.t. \ PRECEDENCE(p, q) = 1
 \end{aligned}$$

Equation 10

$$\begin{aligned}
& bigM * \left(2 - \left(2 * \sum_{t \in TURRET} reg(p, s, t) + \sum_{t \in TURRET} onesim(p, s, t) \right) \right) + \\
& bigM * \left(2 - \left(2 * \sum_{t \in TURRET} reg(q, s, t) + \sum_{t \in TURRET} onesim(q, s, t) \right) \right) + \\
& start(q, left) \geq end(p, left) \\
& For p, q \in OP, s \in SPINDLE s.t. PRECEDENCE(p, q) = 1
\end{aligned}$$

Equation 11

$$\begin{aligned}
& 2 * \sum_{t \in TURRET} reg(q, main, t) + \sum_{t \in TURRET} onesim(q, main, t) \\
& \leq bigM * \left(2 - \left(2 * \sum_{t \in TURRET} reg(p, aux, t) + \sum_{t \in TURRET} onesim(p, aux, t) \right) \right) \\
& For p, q \in OP and PRECEDENCE(p, q) = 1
\end{aligned}$$

Equation 12

These constraints concern with precedence relationships among operations and are applied whenever precedence relationships exist between two operations p and q (i.e., when parameter $PRECEDENCE(p, q)$ is 1). Equations 8 through 11 ensure that when precedence relationship exists between operations p and q and they are both assigned to the same spindle s , operation q may not start before operation p .

As an example, when operations p and q are both allocated to spindle s and left turret and they are not performed in simultaneous machining mode, variables $reg(p, s, left)$ and $reg(q, s, left)$ would

both be 1's. Hence, the first two terms of Equations 8, 9, 10, and 11 become $bigM*(2-(2*1+0))$ which amounts to zero and the starting time for operation q at turret t (i.e., $start(q, t)$) has to be later than that of operation p ($start(p, t)$).

Equation 12 eliminates the possibility of operation q being assigned to the main spindle if its parent operation p is performed at the auxiliary spindle. When that occurs, the $bigM$ term on the right hand side of the equation becomes zero and $reg(q, main, t)$ as well as $onesim(q, main, t)$ are consequently forced to be zero as well. This is due to the way that each part gets machined by first being held at the main spindle and later gets transferred to the auxiliary spindle in a batch-processing environment. Thus, operations performed at the auxiliary spindle are taking place at a later time than at the main spindle.

A5.4: Two-Feature Simultaneous Machining Related Constraints

$$\sum_{s \in SPINDLE} twosim(p, s, q) + \sum_{s \in SPINDLE} twosim(q, s, p) \leq 1$$

For $p, q \in OP$ and $TWOFEATURE(p, q) = 1$

Equation 13

$$\sum_{s \in SPINDLE} twosim(p, s, q) = 0$$

For $p, q \in OP$ and $TWOFEATURE(p, q) \neq 1$

Equation 14

$$twosim(p, aux, q) \leq \sum_{t \in TURRET} reg(q, aux, t)$$

For $p, q \in OP$ and $TWOFEATURE(p, q) = 1$

Equation 15

$$twosim(p, aux, q) \leq \sum_{t \in TURRET} reg(p, aux, t)$$

For $p, q \in OP$ and $TWOFEATURE(p, q) = 1$

Equation 16

$$twosim(p, main, q) \leq \sum_{t \in TURRET} reg(q, main, t)$$

For $p, q \in OP$ and $TWOFEATURE(p, q) = 1$

Equation 17

$$twosim(p, main, q) \leq \sum_{t \in TURRET} reg(p, main, t)$$

For $p, q \in OP$ and $TWOFEATURE(p, q) = 1$

Equation 18

$$reg(q, s, right) + bigM * (1 - twosim(p, s, q)) + bigM * (1 - reg(p, s, left)) \geq 1$$

For $p, q \in OP, s \in SPINDLE$ and $TWOFEATURE(p, q) = 1$

Equation 19

$$\begin{aligned} ®(q, s, left) + bigM * (1 - twosim(p, s, q)) + bigM * (1 - reg(p, s, right)) \geq 1 \\ &For\ p, q \in OP, s \in SPINDLE\ and\ TWOFEATURE(p, q) = 1 \end{aligned}$$

Equation 20

This set of constraints is applicable to all two-feature simultaneous machining capable operations. Equation 13 emphasizes the fact that $twosim(p, s, q)$ and $twosim(q, s, p)$ are not treated similarly in the formulation and two-feature simultaneous machining can only occur in either case. If the applications of two-feature simultaneous machining are deemed infeasible or unsuitable to the two operations p and q (i.e., parameter $TWOFEATURE(p, q) \neq 1$), variable $twosim(p, s, q)$ or $twosim(q, s, p)$ will be set to zero as shown in Equation 14.

Equations 15 through 18 ensure that operations performed in two-feature simultaneous machining mode are allocated to the appropriate spindles and turrets. In addition, the partner operations have to be allocated to different turrets. Equation 19 states that when operations p and q are performed in two-feature mode and operation p uses the left turret, operation q must use the right turret. This condition is guaranteed because when $twosim(p, s, q)$ and $reg(p, s, left)$ equal one, the terms $bigM*(1-twosim(p, s, q))$ and $bigM*(1-reg(p, s, left))$ become zero. Hence, it forces variable $reg(q, s, right)$ to be greater or equal to one. Since $reg(q, s, right)$ can only be either one or zero, it is forced to be 1 which means that partner operation q is allocated to the right turret as desired. Similar logic applies to Equation 20 as well.

A5.5: Tooling Related Constraints

$$\sum_{tl \in TOOL} tool(tl, t) \leq MAXTOOLS$$

For $t \in TURRET$

Equation 21

$$tool(tl, t) \leq TURRETOK(tl, t)$$

For $tl \in TOOL, t \in TURRET$

Equation 22

$$\sum_{s \in SPINDLE} reg(p, s, t) \leq tool(OPTOOL(p), t)$$

For $p \in OP, t \in TURRET$

Equation 23

$$bigM * (2 - \sum_{\substack{s \in SPINDLE \\ t \in TURRET}} onesim(p, s, t) + \sum_{t \in TURRET} tool(OPTOOL(p), t) \geq 2$$

For $p \in OP, t \in TURRET$

Equation 24

For tooling related constraints, Equation 21 is used to limit turret capacity. Hence, the number of tools allocated to each turret may not exceed turret capacity (*MAXTOOLS*). In the same way that parameters *REGULAR*(*p*, *s*, *t*) and *ONEFEATURE*(*p*, *s*, *t*) restrict decision variables *reg*(*p*, *s*, *t*) and *onesim*(*p*, *s*, *t*), parameter *TURRETOK*(*tl*, *t*) is used to constrain tool allocations to turrets (Equation 22). If *TURRETOK*(2, *left*) is zero, variable *tool*(2, *left*) has to be zero as well. This means that tool number 2 may not be assigned to the left turret.

Equation 23 ensures that when operation *p* is allocated to turret *t*, its required tool (parameter *OPTOOL*(*p*) refers to the tool used by operation *p*) has to be present at turret *t* as well. Similarly, the needed tool has to be present at both the turrets in case of one-feature simultaneous machining applications. This constraint is represented by Equation 24.

A5.6: Operation Sequence Related Constraints

$$\sum_{\substack{j \in OPS \\ s.t. i \neq j, j \neq source}} x(i, t, j) = \left(\sum_{s \in SPINDLE} reg(i, s, t) + \sum_{s \in SPINDLE} onesim(i, s, t) \right)$$

For $i \in OPS, t \in TURRET$

Equation 25

$$\sum_{\substack{i \in OPS \\ s.t. i \neq j, i \neq sink}} x(i, t, j) = \left(\sum_{s \in SPINDLE} reg(j, s, t) + \sum_{s \in SPINDLE} onesim(j, s, t) \right)$$

For $j \in OPS, t \in TURRET$

Equation 26

$$\begin{aligned}
& bigM * (1 - \sum_{t \in TURRET} reg(p, s, t)) + bigM * (1 - \sum_{t \in TURRET} reg(q, s, t)) + \\
& y(p, s, q) + y(q, s, p) \geq 1 \\
& \text{For } p, q \in OP \text{ and } p \neq q, s \in SPINDLE
\end{aligned}$$

Equation 27

$$\begin{aligned}
& bigM * (1 - y(p, s, q)) + bigM * (twosim(p, s, q) + twosim(q, s, p)) + \\
& start(q, left) \geq end(p, left) \\
& \text{For } p, q \in OP \text{ and } p \neq q, s \in SPINDLE
\end{aligned}$$

Equation 28

$$\begin{aligned}
& bigM * (1 - y(p, s, q)) + bigM * (twosim(p, s, q) + twosim(q, s, p)) + \\
& start(q, right) \geq end(p, right) \\
& \text{For } p, q \in OP \text{ and } p \neq q, s \in SPINDLE
\end{aligned}$$

Equation 29

$$\begin{aligned}
& bigM * (1 - y(p, s, q)) + bigM * (twosim(p, s, q) + twosim(q, s, p)) + \\
& start(q, left) \geq end(p, right) \\
& \text{For } p, q \in OP \text{ and } p \neq q, s \in SPINDLE
\end{aligned}$$

Equation 30

$$\begin{aligned}
& bigM * (1 - y(p, s, q)) + bigM * (twosim(p, s, q) + twosim(q, s, p)) + \\
& start(q, right) \geq end(p, left) \\
& \text{For } p, q \in OP \text{ and } p \neq q, s \in SPINDLE
\end{aligned}$$

Equation 31

These constraints facilitate the determination of operations sequence at spindles and turrets. Equations 25 and 26 guarantee that when an operation i is assigned to turret t (i.e., $reg(i, s, t)$ or $onesim(i, s, t)$ is equal to 1), it would be present on the sequence of operations at turret t (i.e., $x(i, t, j)$ and $x(k, t, i)$ are both 1, $i, j,$ and $k \in OPS$ and $i \neq j \neq k$). Variable $y(p, s, q)$ is used to order the sequence of operations p and q at spindle s . Thus, when operations p and q are allocated to spindle s , either $y(p, s, q)$ or $y(q, s, p)$ must be one (Equation 27).

Since a spindle is also considered a resource, two operations assigned to the same spindle must be performed in order (i.e., no two operations assigned to the same spindle can be performed simultaneously) with the exception of two-feature simultaneous machining mode. Hence, barring the instances of two-feature simultaneous machining for two operations p and q ($twosim(p, s, q)$ and $twosim(q, s, p)$ are both zero's), operation q must be performed after operation p is completed (i.e., $start(q, t)$ must be greater than $end(p, t)$) when $y(p, s, q)$ is one. Equations 28 through 31 represent the above constraints.

A5.7: Time Sequence Related Constraints

$$\begin{aligned} end(i,t) + indexing(i,j) &\leq start(j,t) + bigM * (1 - x(i,t,j)) \\ \text{For } i, j \in OPS, t \in TURRET \text{ s.t. } i \neq j \end{aligned}$$

Equation 32

$$\begin{aligned} start(i,t) + duration(i,regg) * \sum_{s \in SPINDLE} reg(i,s,t) + \\ duration(i,onef) * \sum_{s \in SPINDLE} onesim(i,s,t) \\ = end(i,t) \\ \text{For } i \in OPS, t \in TURRET \end{aligned}$$

Equation 33

$$\begin{aligned} bigM * (1 - x(p,left,sink)) + bigM * (1 - x(source,left,q)) + \\ LT \geq start(sink,left) + indexing(p,q) \\ \text{For } p, q \in OP \end{aligned}$$

Equation 34

$$\begin{aligned} bigM * (1 - x(p,right,sink)) + bigM * (1 - x(source,right,q)) + \\ RT \geq start(sink,right) + indexing(p,q) \\ \text{For } p, q \in OP \end{aligned}$$

Equation 35

$$\begin{aligned}
& \text{indexing}(i, j) \geq \text{INDEX}(i, j) \\
& \text{For } i, j \in \text{OPS}
\end{aligned}$$

Equation 36

$$\begin{aligned}
& \text{bigM} * (1 - x(p, t, \text{sink})) + \text{bigM} * (1 - x(\text{source}, t, q)) + \\
& \text{bigM} * (2 - (2 * \sum_{t \in \text{TURRET}} \text{reg}(p, \text{main}, t) + \sum_{t \in \text{TURRET}} \text{onesim}(p, \text{main}, t))) + \\
& \text{bigM} * (2 - (2 * \sum_{t \in \text{TURRET}} \text{reg}(q, \text{aux}, t) + \sum_{t \in \text{TURRET}} \text{onesim}(q, \text{aux}, t))) + \\
& \text{indexing}(p, q) \geq \text{INDEX}(p, q) + \text{TURRET_SWITCH} \\
& \text{For } p, q \in \text{OP and } p \neq q, t \in \text{TURRET}
\end{aligned}$$

Equation 37

$$\begin{aligned}
& \text{bigM} * (1 - x(p, t, \text{sink})) + \text{bigM} * (1 - x(\text{source}, t, q)) + \\
& \text{bigM} * (2 - (2 * \sum_{t \in \text{TURRET}} \text{reg}(p, \text{aux}, t) + \sum_{t \in \text{TURRET}} \text{onesim}(p, \text{aux}, t))) + \\
& \text{bigM} * (2 - (2 * \sum_{t \in \text{TURRET}} \text{reg}(q, \text{main}, t) + \sum_{t \in \text{TURRET}} \text{onesim}(q, \text{main}, t))) + \\
& \text{indexing}(p, q) \geq \text{INDEX}(p, q) + \text{TURRET_SWITCH} \\
& \text{For } p, q \in \text{OP and } p \neq q, t \in \text{TURRET}
\end{aligned}$$

Equation 38

$$\begin{aligned}
& \text{bigM} * (1 - x(p, t, q)) + \\
& \text{bigM} * (2 - (2 * \sum_{t \in \text{TURRET}} \text{reg}(p, \text{aux}, t) + \sum_{t \in \text{TURRET}} \text{onesim}(p, \text{aux}, t))) + \\
& \text{bigM} * (2 - (2 * \sum_{t \in \text{TURRET}} \text{reg}(q, \text{main}, t) + \sum_{t \in \text{TURRET}} \text{onesim}(q, \text{main}, t))) + \\
& \text{indexing}(p, q) \geq \text{INDEX}(p, q) + \text{TURRET_SWITCH} \\
& \text{For } p, q \in \text{OP and } p \neq q, t \in \text{TURRET}
\end{aligned}$$

Equation 39

$$\begin{aligned}
& bigM * (1 - x(p, t, q)) + \\
& bigM * (2 - (2 * \sum_{t \in TURRET} reg(p, main, t) + \sum_{t \in TURRET} onesim(p, main, t))) + \\
& bigM * (2 - (2 * \sum_{t \in TURRET} reg(q, aux, t) + \sum_{t \in TURRET} onesim(q, aux, t))) + \\
& indexing(p, q) \geq INDEX(p, q) + TURRET_SWITCH \\
& For p, q \in OP \text{ and } p \neq q, t \in TURRET
\end{aligned}$$

Equation 40

$$\begin{aligned}
& bigM * (1 - (\sum_{t \in TURRET} reg(p, main, t) + onesim(p, main, left))) + start(p, right) \geq LOAD \\
& For p \in OP
\end{aligned}$$

Equation 41

$$\begin{aligned}
& bigM * (1 - (\sum_{t \in TURRET} reg(p, main, t) + onesim(p, main, left))) + start(p, left) \geq LOAD \\
& For p \in OP
\end{aligned}$$

Equation 42

$$\begin{aligned}
& bigM * (1 - x(p, right, sink)) + \\
& bigM * (2 - (2 * \sum_{t \in TURRET} reg(p, aux, t) + \sum_{t \in TURRET} onesim(p, aux, t))) + \\
& RT \geq start(sink, right) + UNLOAD \\
& For p \in OP
\end{aligned}$$

Equation 43

$$\begin{aligned}
& bigM * (1 - x(p, left, sink)) + \\
& bigM * (2 - (2 * \sum_{t \in TURRET} reg(p, aux, t) + \sum_{t \in TURRET} onesim(p, aux, t))) + \\
& RT \geq start(sink, left) + UNLOAD \\
& For p \in OP
\end{aligned}$$

Equation 44

In addition to operation sequence at both spindles and turrets, we need to determine the starting and end time of each operation to be able to evaluate the overall cycle time of a process plan. This set of constraints establishes the starting time, end time, and sequence-dependence turret indexing time for each operation.

Equation 32 indicates that when two operations i and j are performed in immediate order at turret t (or $x(i, t, j) = 1$), operation j may not start until operation i is completed and the turret has indexed and move to the ready-to-cut position. The end time of an operation is defined as the starting time plus duration shown in Equation 33. Equations 34 and 35 are used to adjust the cycle times (or processing times) at both main and auxiliary spindles with the indexing time of the last operations on each turret sequence. This requires the knowledge of the first operation on each sequence since after the last operation at each turret is performed, the turret has to index to perform the first operation of a new part.

Equations 36 through 40 cope with the turret indexing time. The turret indexing time can be predetermined to a certain degree since the tool requirements for all machining operations are

known beforehand (i.e., parameter $INDEX(i, j)$ can be preset) . However, spindle allocations have to be established before the total indexing times (i.e., variable $indexing(i, j)$ can be determined. This is due to the fact that when two operations using the same turret are performed at different spindles, the turret has to travel from one spindle to another after it has completed the first operation. The turret movement time from one spindle to the other is defined as the turret switching time in this formulation.

Equation 36 allows the possibility for variable $indexing(i, j)$ to incorporate the turret switching time. In Equations 37 through 40, the turret indexing time is adjusted for two operations performed in immediate succession by the same turret. The turret switching time is added only when the two adjacent operations are allocated to different spindles.

If an operation is allocated to the main spindle, it may not be start until the part has been loaded to the spindle. This fact is encapsulated in Equations 41 and 42. By the same token, the unloading time has to be taken into account for the last operation allocated to the auxiliary spindle (Equations 43 and 44).

A5.8: Turret Synchronization Related Constraints

$$\begin{aligned}
 &bigM * (1 - twosim(p, s, q)) + start(p, left) \geq start(q, right) \\
 &For p, q \in OP, s \in SPINDLE \text{ and } TWOFEATURE(p, q) = 1
 \end{aligned}$$

Equation 45

$$bigM * (1 - twosim(p, s, q)) + start(q, right) \geq start(p, left)$$

For $p, q \in OP, s \in SPINDLE$ and $TWOFEATURE(p, q) = 1$

Equation 46

$$bigM * (1 - twosim(p, s, q)) + start(p, right) \geq start(q, left)$$

For $p, q \in OP, s \in SPINDLE$ and $TWOFEATURE(p, q) = 1$

Equation 47

$$bigM * (1 - twosim(p, s, q)) + start(q, left) \geq start(p, right)$$

For $p, q \in OP, s \in SPINDLE$ and $TWOFEATURE(p, q) = 1$

Equation 48

$$bigM * (1 - onesim(i, s, t)) + start(i, right) \geq start(i, left)$$

For $i \in OPS, s \in SPINDLE, t \in TURRET$, and $ONEFEATURE(i, s, t) = 1$

Equation 49

$$bigM * (1 - onesim(i, s, t)) + start(i, left) \geq start(i, right)$$

For $i \in OPS, s \in SPINDLE, t \in TURRET$, and $ONEFEATURE(i, s, t) = 1$

Equation 50

Equations 45 through 50 constitute turret synchronization constraints for the process plan optimization problem. When operations are performed in simultaneous machining mode, the turrets have to be synchronized to ensure a starting condition that prevents turret collision and improves force balancing. Equations 45 through 48 force the starting times for partner operations

p and q to be synchronized at both turrets when they are performed in two-feature simultaneous machining mode. Similarly, Equations 49 and 50 take care of turret synchronization for operations performed in one-feature simultaneous machining mode.

A5.9: Miscellaneous Constraints

$$reg(sink, main, left) = 1$$

Equation 51

$$reg(sink, aux, right) = 1$$

Equation 52

$$reg(source, main, left) = 1$$

Equation 53

$$reg(source, aux, right) = 1$$

Equation 54

$$\sum_{q \in OP} x(source, t, q) = 1$$

For $t \in TURRET$

Equation 55

$$\sum_{p \in OP} x(p, t, sink) = 1$$

For $t \in TURRET$

Equation 56

$$\sum_{t \in TURRET} x(i, t, i) = 0$$

For $i \in OPS$

Equation 57

$$x(p, t, source) = 0$$

For $p \in OP, t \in TURRET$

Equation 58

$$x(sink, t, q) = 0$$

For $q \in OP, t \in TURRET$

Equation 59

$$onesim(p, s, left) = onesim(p, s, right)$$

For $p \in OP, s \in SPINDLE$

Equation 60

Miscellaneous constraints are used to force variables whose values can be predetermined to minimize the number of unknown variables involved in the problem. Equations 51 through 54 set the spindle and turret allocations for *source* and *sink* which are two dummy (i.e., meaningless)

operations. In the formulation, *source* precedes all other operations whereas *sink* is preceded by all other operations. In other words, each operation sequence starts with *source* and ends with *sink*. These dummy operations are required since some variable such as $x(i, t, j)$ involves a pair of operations. Hence, the first operation in the sequence has to be preceded by a dummy operation (i.e., *source*). Likewise, the last operation in the sequence has to be succeeded by a dummy operation (i.e., *sink*). These constraints are presented by Equations 55 and 5 Equations 57, 58, and 59 prevent nonsense variables to be assigned meaningful values. In one-feature simultaneous machining, both turrets are involved hence $onesim(p, s, left)$ has to be equal to $onesim(p, s, right)$. This is taken care of by Equation 60.