GENERATION OF PROCESS VARIANTS

Christiane Soika, Tobias Teich, Joerg Militzer and Daniel Kretz
Academic Department of Economics, University of Applied Sciences Zwickau, Zwickau, Germany

Keywords: CAD, CAM, STEP, Ant colony optimization.

Abstract: Small and medium sized enterprises (SME's) are commonly dependent on large-scale enterprises in their role as a supplier. In fact, growing international competition increases the pressure on SME's. Hence, it is enormously important, to react as fast and exact as possible on customer demands, while keeping high quality standards at the same time. In order to achieve these objectives, CAPP (Computer Aided Production Planning)-systems were introduced. This paper provides an integrated solution for automated process and production planning and we present our approach for generating process variants by using Ant Colony Optimization (ACO).

1 INTRODUCTION

Inefficient production planning in small and medium sized enterprises (SME's) is commonly known but still a significant problem. Generally, an integrated or automated production planning is entirely missing for small-series or single-part production. In their role as a supplier it is existential, not only to react fast and flexible on customer demands, but also to provide high quality products. Furthermore, it is very important to reduce or completely avoid additional costs by preventing the manufacturing of deficient products.

These issues require an intelligent and very efficient organized production planning. CAPP-systems are a fundamental component for an optimized production planning within SME's. This paper presents our approach for integrating computer aided production planning (CAPP) systems directly with computer aided design (CAD) and enterprise resource planning (ERP) systems for evaluating production variants within a given factory environment.

1.1 Process Model for Integrated and Automated Production Planning

Our intended solution for the introductory outlined problem consists of five modules that are visualized in terms of an extended event-driven process chain (eEPC) in Figure 1 and Figure 2. First of all, there is the *Design Module* (DM) that is used for providing a CAD model of either a single piece part or complex assemblies. Regarding this, we utilize the ISO standard 10303, most informally known as STEP '*Standard for the Exchange of Product model data*'. Besides a standardized exchange, application protocols of ISO 10303 provide reference models for a standardized product data representation. Thereby, we have focused our attention on the reference model of the application protocol 224 that is a special part of ISO 10303 for machining industries (South Carolina Research Authority, 2006). Within AP 224 '*Mechanical product definition for process planning using machining feature*' parts are modeled and parameterized by using feature based design with machining features.

Machining features are objects, whose semantics imply corresponding manufacturing operations, like a pocket feature implies a milling operation or an outer diameter feature could indicate a turning operation. After defining a base shape which defines an initial volume, the final part geometry is derived by applying manufacturing features which describe volumes that shall be removed by machining or shapes that result from machining. Manufacturing features of ISO 10303 are classified into machining, replicate and transition features. Furthermore, machining features are specialized as compound or multi-axis features. Compound features consist of two or more manufacturing features and multi-axis features are commonly manufactured by milling processes. Transition features define a transition area between at least two sur-

Soika C., Teich T., Militzer J. and Kretz D..
GENERATION OF PROCESS VARIANTS.
DOI: 10.5220/0003516100730078
In *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2011), pages 73-78
ISBN: 978-989-8425-74-4

faces and replicate features are identical copies of any arbitrary manufacturing feature or another replicate feature.

Second, there is the *Resource Description Module* (RDM) that is used for interpreting the design model and determining alternative sets of required resources and manufacturing operations for any involved manufacturing feature (Gaese and Winkler, 2009), (Teich et al., 2009). This module is additionally fed by an ERP system, in our case the SAP system. The RDM associates suitable manufacturing processes to the corresponding manufacturing features from the given product design model.

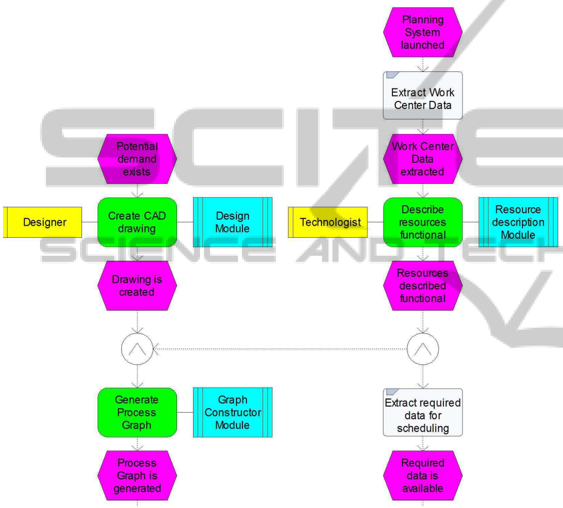

Figure 1: Automated process planning system (eEPC 1/2).

This n:n relation provides the input information for our *Graph Constructor Module* (GCM) that generates process variants. Each process variant is scheduled and evaluated by a *genetic algorithm* (GA) of our *Evolutionary Module* (EM). After a first evaluation, the GCM constructs alternative process variants driven by the evaluated scheduling result and an ant colony algorithm within our *Swarm Intelligence Module* (SIM). The generation and evaluation of process variants is an iterative process.

It is the matter of fact, that every manufacturing feature that is used for destructive designing the part, exponentially increases the number of process variants because we do not only derive variants from manufacturing features themselves but also consider the resulting part as a whole. Consequently, we utilize artificial intelligence, especially *Ant Colony Optimization* (ACO) and GA, for the generation and evaluation of these variants. The ACO selects a set of process variants in cooperation with the GCM. Afterwards, each set is scheduled considering manufacturing capacity during a potential production time, claimed delivery date and further target criteria. This process requires information that is received from the SAP system via *Core Interface Function* (CIF) through the *SAP Java Connector* (SAP JCo).

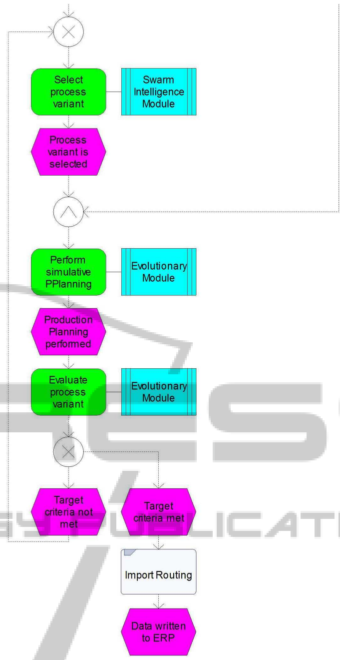

Figure 2: Automated process planning system (eEPC 2/2).

Accruing costs, the entire production time and adherence to delivery dates are primarily focused during our evaluation. Afterwards, these variants are compared with each other. This results into a steered selection and generation of new process variants for further evaluation. Consequently, these process steps are repeated iteratively until any determination criterion or certain target criteria are met. After deriving a process variant that fits the target criteria of costumer and management, this solution is preserved. Further details about the generation of process variants with ACO are presented in detail within section 3.

1.2 Sequencing in Production Planning

The objective of sequencing is the determination of a required manufacturing process sequence by using applicable machines (Loedding, 2005). Consequently, sequencing always influences logistic target values like delivery reliability, service level and yield. Finally, there are two approaches for creating an advantageous sequence. There are mathematic definite methods like branch and bound or the utilization of heuristics like ant colony optimization or genetic algorithms. The following sections discuss a conceptual solution for creating optimized sequences with heuristic methods.

2 PROCESS VARIANTS

In this issue, we want to focus our attention on creating sequences of manufacturing processes for producing single piece parts by turning, drilling or milling operations. Subsequently, scheduling will be involved for an evaluation of the generated process variants. There are different target criteria to consider during evaluation, like accruing production costs or adherence to delivery dates for avoiding contractual penalties. For our intended solution - to generate process plans directly from a CAD drawing - we utilize the feature-based design with manufacturing features because otherwise an efficient and automated interpretation of a complete and enhanced product design model is absolutely hampered (Kretz et al., 2009).

Figure 3 illustrates required manufacturing features of a roll-axis demonstrator. In fact, there are 21 features defined for the final shape. Considering, that there may be several machines which could manufacture the shape of one or many features with distinct parameters for set-up time, accruing costs or processing time, are facts increasing determination complexity. To give an example for alternative machines, feature *FE7* is an outer_round-outer_diameter that could be manufactured either by a CNC-machining center or a lathe. Furthermore, there could be technological dependencies between feature aspects which shall be minded.

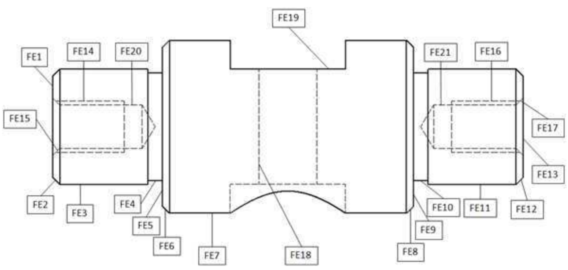

Figure 3: Roll-axis with manufacturing features.

2.1 Feature Dependencies

Dependencies are distinguished between technological and geometrical dependencies. The influence of geometrical dependencies for process variant generation is low. But they are very important for the resulting quality of a part. An example for geometrical dependencies are FE20 and FE21. Both shall be manufactured on exactly the same position on the left as well as the right side of the part and right angular to FE18. Inexactnesses and irregularities lead to a deficient part, which finally cannot be used.

Instead of this, a technological dependency is given, if at least one feature requires the manufacturing of shape aspects defined by other features. Against geometrical dependencies, technological dependencies influence process variants intensively. An example for technological dependencies are the features FE21 and FE16. FE21 is a round_hole with conical_hole_bottom and has to be manufactured first. FE16 typifies a thread feature and could be only applied after drilling the hole. Consequently, we have priorities that shall be considered for generating technological useful process variants.

Further restrictions are required for sequences which could be technological manufactured but the generated variant would be obviously inefficient. To give an example, a chamfer feature (FE12) could maybe manufactured before an associated outer diameter (FE11) feature, but this is extremely inefficient.

2.2 Graph Theory

For generating process variants we require a graph representation. Features that shall be applied to manufacture the final part are defined by the product model. Consequently, our sequence graph is given implicitly. Edges determine the application of manufacturing features. Weighting of an edge depends on the target criteria. Criteria are e.g. production costs, time or a weighted combination of them. Furthermore it has to be considered if a manufacturing feature could be applied with a single manufacturing operation.

Knots within our graph represent intermediate products during the manufacturing of the part. Given this fact, we have a dynamic graph. Depending on the previous selection of features, the succeeding intermediate products are different. Therefore, we construct our graph dynamically while generating a process variant. The initial knot of our graph is always defined by the base shape. Considering feature dependencies, there are still many variant edges remaining. Each intermediate product depends on the selected feature.

Feature selection and determination of the resulting intermediate product are repeated iteratively until the part is completely manufactured. Possibly the application of one manufacturing feature requires at least more than one manufacturing operation. Hence, a manufacturing operation only prepare part aspects and succeeding operations complete the shape. Generating process variants in this way is very similar to the Traveling Salesman Problem (TSP). Considering

TSP, knots are cities that have to be visited and edges dimension the distance between two cities. The objective is to find the shortest round trip by visiting each city exactly once. Both problems are very similar but instead of our dynamic graph representation, the TSP graph is static. Thus, the entire graph is traversable every time and paths are fixed. Best performance and efficiency for solving such represented minimization problems are achieved by ant colony optimization. In 1991, the Italian mathematician Marco Dorigo published the first ant algorithm for solving this problem (Dorigo et al., 1991) (Dorigo and Stuetzle, 2004).

3 ANT COLONY OPTIMIZATION

Because the analogy of both problems, traveling salesman and generation of process variants, we utilize ant colony optimization. ACO is suitable for solving difficult discrete optimization problems that could be described by a graph. Therefore, we use simple agents, in this case artificial ants who communicate with each other mediated by the environment.

Our previously discussed problems are classified as static and dynamic combinatorial problems. TSP is a static combinatorial problem, because the initially given information cannot change. Against this, the generation of process variants is a dynamic combinatorial problem. An example for changing information are the intermediate products. Depending on the preceding selected features, the resulting intermediate product is quite different. In fact, shape aspects of one feature could not be manufactured with a single operation because a particular machine is unavailable or against this, a compound-feature that consists of two or more atomic manufacturing features could be manufactured with a single operation on a special milling center. This forces our algorithm, to adapt the changing problem definition (Dorigo and Stuetzle, 2004). Ant colony optimization was already used for solving similar kinds of problems, like optimizing production plans (Liu et al., 2010) or ad-hoc-networks (Kamali and Opatrny, 2009). ACO summarizes a set of distinct algorithms that are based on the same approach but optimized for special problems. For our purposes, we apply the Ant Colony System (ACS).

3.1 Ant Colony System

Ant Colony Optimization is a nature analogue approach that imitates the behavior of Argentine ants. Ants have limited opportunities to communicate with each other. They use a chemical substance for communication called pheromones and deposit these on their way between anthill and source of food. Consequently, succeeding ants can orient themselves on the given trace. If there are two different ways with different lengths between anthill and source of food source, the first ant takes a random selection. A little later, the pheromone concentration on the shorter trace is higher than on the longer because of the length this route could be more often passed. Consequently, the probability of selecting the shorter route increases with the pheromone concentration. But there always remains a probability for selecting alternative routes that characterizes a heuristic method. In fact, there is never guaranteed that those algorithms find the optimal solution but they always have an optimizing nature. Hence, the natural approach was scientifically adapted for solving combinatorial problems. Ant algorithms consist of three phases, starting with solution construction, followed by pheromone update and optional daemon-activities.

For creating a solution, an ant starts from the current initial knot with a specific probability to a neighbored knot. Afterwards, this task is repeated until a termination criterion is met. During pheromone update, the ant deposits pheromone by leaving the edge. Daemon-activities are further optional activities e.g. deposition of pheromones on the entire final ant path. Certain aspects of Ant Colony System (ACS) differ from ant algorithm. To give an example, ACS uses a global and a local pheromone update. A global pheromone update addresses the deposition and evaporation on the entire path of the current best solution. Consequently, only the best evaluated ant is authorized for a global pheromone update. Furthermore, there is a local pheromone update where every ant reduces the pheromone concentration after leaving an edge. This approach supports a wide search that avoids a concentration on local optima as well as premature convergence (Fischer, 2008).

```
Start
  Determination of raw material
  For every ant in colony
    REPEAT
      IF pheromone exists on edge
        Pheromone-steered and random selection of  the next feature
      ELSE
        Randomly selection of the next feature
      IF no technological dependencies between selected and other feature
        Calculation of location and orientation of the feature
        Demand at RDM
        Answer with possible process steps
        Selection of one process step
      IF feature could manufactured completely
        Deleting of feature out of feature-list
        IF pheromone existing at the edge
          Evaporation
    UNTIL process variant is generated completely
    Information about process variant is stored in data base
    Evaluation of current process variant
    IF current process variant is best so far solution
      Deposition over the hole path
  ENDFOR
END
```

Figure 4: Pseudo code for generating process variants.

3.2 Algorithm for Generating Process Variants

Listing 4 presents the pseudo code of our process variant generation approach. First of all, a suitable raw material and the base position of the final part are determined. Against the default ant algorithm, our approach upgrades the ants with a memory about the manufacturing features of the part and their technological dependencies (feature list). After determining the raw material, each ant decides which feature shall be manufactured first. Therefore, it has to be checked whether there is already pheromone on this edge. Hence, we check for every feature in the list, whether there is a pheromone value on the edge between initial shape and resulting intermediate product. If there is no pheromone value for orientation, the selection is straight randomly. Otherwise the selection is weighted as a pheromone-steered random selection.

The probability $p(x_{ij})$, for selecting an edge from either an initial base shape or an intermediate product i to a further intermediate product j is derived by the following formula:

$$p(x_{ij}) = \left(\frac{\tau_{ij}^{\alpha} \cdot \nu_{ij}^{\beta}}{\sum_{k \in X} \tau_{ik}^{\alpha} \cdot \nu_{ik}^{\beta}} \right) \tag{1}$$

τ_{ij} represents the pheromone concentration on the edge between intermediate product i and j. The parameters α and β influence the concentration of pheromone and heuristic information (Kramer, 2009).

After selecting an edge for the next step, it has to be checked whether technological dependencies prevent the manufacturing of the involved feature or not. If there are not any technological dependencies, the location and orientation of a feature are derived and requested as a demand to our RDM. The corresponding data structure is illustrated as class diagram in Figure 5.

The demand consists of two parts. First, there is the feature information like identifier, name, description, location, orientation and other required parameters. Those are used by the RDM for determining a set of suitable manufacturing operations. Second, their is an information about the concrete previous selection from this set. Hence our RDM updates the current intermediate product.

After determining suitable manufacturing operations, the set is replied as an answer as illustrated in Figure 6. Possibly, there are no suitable manufacturing operations, e.g. if there currently is not any machine available or a single manufacturing operation could apply more than one features. If an answer

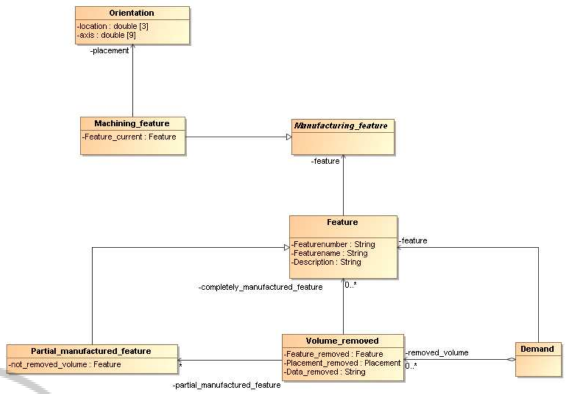

Figure 5: UML class diagram: feature request.

contains at least one operation, then it supplies the following information:

- removed volume,
- state of manufacturing,
- manufacturing method,
- selected machine,
- used tool,
- required tool clamping devices,
- required work piece clamping,
- production time (execution, mounting and set-up)
- production costs (execution, mounting and set-up).

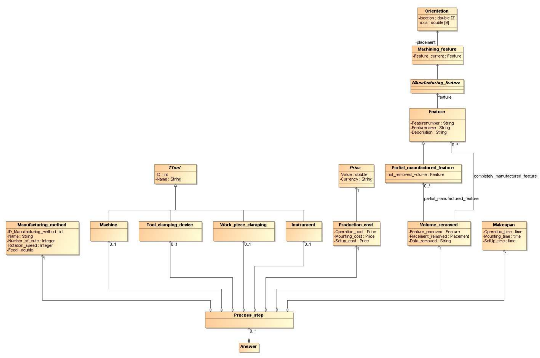

Figure 6: UML class diagram: RDM answer.

Driven by the target criteria, one of the operations are selected. The selection depends on time- or cost-minimization or a composition of them. Another aspect is the state of manufacturing that indicates whether a feature was manufactured completely or not. If it was only prepared, hence a further process step is later required for completion. Otherwise, it was manufactured completely and consequently removed from feature list. Afterwards, evaporation is utilized, if there is pheromone on the edge. Without evaporation, the search would result in a premature convergence on a local optimum. The evaporation is calculated with the following formula:

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} \tag{2}$$

φ is a parameter which describes the evaporation rate, defined by $0 < \varphi < 1$ (Fischer, 2008).

Selection of the next feature, request to and response from RDM, as well as the selection of a manufacturing operation from the response set are repeated until the entire process variant is generated and thus the final part produced. Afterwards, the process variant is preserved and evaluated with our GA. If the current process variant is the best, then pheromones are deposited on the entire path. Currently, we have only focused production time and accruing costs of the manufacturing operations. In fact, there are still another aspects. To given an example, two features have to be manufactured on two different machines. The resulting intermediate product has to be transported from current machine to another. Hence, this additional time and consequently it could be more efficient to execute both operations on the same machine.

4 CONCLUSIONS

We have stated our problem of generating process variants in an automated integrated production planning within this paper. Therefore, we have geometrical and technological dependencies defined which derive from the feature based design. For solving our generation problem, we utilize heuristic approaches, especially ant colony optimization and genetic algorithms. This paper provided a short introduction about Ant Colony Optimization and our intention for their usage. Finally, we have explained our approach for generating process variants with the Ant Colony System. Future work deals with the implementation and integration of this approach into our planing solution as a software module.

REFERENCES

Dorigo, M., Maniezzo, V., and Colorni, A. (1991). Ant system: An autocatalytic optimizing process.

Dorigo, M. and Stuetzle, T. (2004). *Ant Colony Optimization*. MIT Press.

Fischer, M. (2008). *Partnerauswahl in netzwerken*. Verlag Dr. Kovač.

Gaese, T. and Winkler, S. (2009). Entwicklung eines ressourcenmodells fr die automatische angebotsgenerierung auf grundlage featurebasierter produktbeschreibung. In *Journal of the University of Applied Sciences Mittweida*, pages pp. 18–21. Hochschule Mittweida.

Kamali, S. and Opatrny, J. (2009). A hybrid ant-colony routing algorithm for mobile ad-hoc networks. In *Complex Sciences*, pages pp. 1337–1354. Springer, Berlin Heidelberg.

Kramer, O. (2009). *Computational Intelligence*. Springer Verlag.

Kretz, D., Teich, T., Militzer, J., Jahn, F., and Neumann, T. (2009). Step standardized product data representation and exchange for optimized product development and automated process planning. In *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*, pages pp. 41–56. Springer.

Liu, X., Yi, H., and Ni, Z. (2010). Application of ant colony optimization algorithm in process planning optimization. In *Journal of Intelligent Manufactoring*, pages pp. 1–13.

Loedding, H. (2005). *Verfahren der fertigungssteuerung*. Springer Verlag.

South Carolina Research Authority (2006). *STEP Application Handbook ISO 10303*. SCRA/ISG, 5300 INTERNATIONAL BOULEVARD. NORTH CHARLESTON, SC 29418, 3. version edition.

Teich, C., Duerr, H., Militzer, J., Neumann, T., and Tran, N. (2009). Automatische grobkalkulation auf basis funktionaler beschreibung der nachfrage. In *Logistics and Supply Chain Management - Modern Trends in Germany and Russia*, pages pp. 294–305. Cuvillier Verlag Goettingen.