Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Computer-Integrated Manufacturing 22 (2006) 256–266
Robotics and Computer-Integrated Manufacturing
www.elsevier.com/locate/rcim

On setup level tool sequence selection for 2.5-D pocket machining

Roshan M. D'Souza

Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA

Received 24 May 2005; received in revised form 24 May 2005; accepted 13 June 2005

Abstract

This paper describes algorithms for efficiently machining an entire setup. Previously, the author developed a graph based algorithm to find the optimal tool sequence for machining a single 2.5-axis pocket. This paper extends this algorithm for finding an efficient tool sequence to machine an entire setup. A setup consists of a set of features with precedence constraints, that are machined when the stock is clamped in a particular orientation. The precedence constraints between the features primarily result from nesting of some features within others. Four extensions to the basic graph algorithm are investigated in this research. The first method finds optimal tool sequences on a feature by feature basis. This is a local optimization method that does not consider inter feature tool-path interactions. The second method uses a composite graph for finding an efficient tool sequence for the entire setup. The constrained graph and subgraph approaches have been developed for situations where different features in the setup have distinct critical tools. It is found that the first two methods can produce erroneous results which can lead to machine crashes and incomplete machining. Illustrative examples have been generated for each method.

Keywords: CAPP; Tool sequence selection

1. Introduction

Process planning for milling consists of three main tasks. The first identifies removal volumes/machining features/pockets and various access directions to machine them [1–3]. The second clusters them into setups based on the feasibility of machining these removal volumes in a particular direction, and clamping the stock [4,5]. The final task consists of selecting appropriate tool sequences to either minimize machining time or total cost.

Current state of the art process planning systems [6,7] allow users to select 2 or more tools for machining pockets. The actual tool sequence selection is left to the human process planner. The process of time or cost optimization is one of trial and error where complete process planning has to be done in order to validate the plan and calculate costs using NC-Verify systems.

The issue of selecting tool sequences has been addressed by several researchers [8–16]. All these researchers have focused on a single contiguous feature. However, in real life situations, several features are machined in a single setup. Moreover, some of these features may be nested and therefore can have precedence constraints for machining. Tool sequence selection thus becomes a very complex problem because of the various interactions between features in the setup.

A problem similar to the setup level tool sequence optimization has been addressed by Balasubramaniam et al. and Yao et al. [17,18]. Balasubramaniam et al. have developed a graph based tool sequence selection method for rough machining of 3-axis pockets. 3-axis rough machining is converted to a 2.5 axis problem by dividing the pocket into 2.5 D slices. In the graph representation, the nodes represent the tools and the weights of the edges, the cost of machining. The cost of machining is calculated based on the numerical value of the area of the cross sections at each depth of cut of the accessible region and not actual tool-paths. This method

E-mail address: rmdsouza@mtu.edu.

0736-5845/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.rcim.2005.06.001

Nomenclature

$f(p,h)$	2.5-D feature/pocket represented by an area p and depth h
t_i	end milling cutter with diameter d_i and cutting length l_i
$A_i(f)$	accessible area of tool t_i in feature f. This essentially is the area that t_i traverses at each depth of cut (*doc*) to machine whatever it can in f
$D_{ij}(f)$	area that t_j traverses in feature f at each *doc* after t_i is done machining to the extent of $A(f)_i$
$T_{\text{feas}}(f)$	set of feasible tools to machine f
$T_{\text{opt}}(f)$	set of tools that form the cheapest tool sequence for f
$X(p,h)$	solid obtained by sweeping 2-D area p through distance h
C_{mn}	precedence constraint resulting from f_m nesting in f_n

of cost calculation is grossly inaccurate as it does not account for geometric complexity. It is stated that the shortest path in the graph is the optimal sequence if the numerical value of the accessible areas monotonically increases down the tool-ordering sequence, assuming that the tools are arranged in the decreasing order of diameters. This assumption works only if the machining cost is a function of the numerical value of the machined areas. However, in real life situations, machining cost is a function of tool-path lengths. Tool-paths are generated from the geometry of the accessible regions. Therefore, the graph approach can be used if and only if, for any two tools, the accessible region of the larger tool is a strict subset of the accessible region of a smaller tool. Only then can the weight of any edge in the graph be independent of the path in which it occurs. Yao et al. have formulated a multipart milling problem using the graph approach. The objective is to select a set of tools to be mounted on a machine for machining several distinct parts from several distinct stock pieces. Essentially, this approach is the same as the composite graph approach described in this paper. An unstated assumption in this formulation is that the critical tool for all the parts is the same. In other words, the smallest tool in the available tool set can completely machine each and every pocket in all the parts. We will show that this approach can lead to incomplete machining when this assumption does not hold.

In this paper we have extended the graph based algorithm for selecting the cheapest tool sequence developed earlier [19]. Four approaches were tried out. In the first approach, tool sequence graphs are solved for individual features. Tool-paths for the tools in the resulting tool sequences are connected on a per tool basis to minimize airtime. This is in a sense a local optimization method. This method can lead to tool crashes in certain situations. The latter methods optimize tool sequences by grouping features in sibling levels. The composite tool sequence graph method is similar to the approach adopted by Yao et al. The constrained graph method forces all possible solutions to pass through critical tools. This constraining can lead to sub-optimal solutions. The subgraph method eliminates this constraint while still generating solutions that completely machine the setup.

2. Tool sequence selection for a single pocket

2.1. Accessible area

Accessible area $A_i(f)$ of a tool t_i in a feature $f(p,h)$ (Fig. 1(a)) is the area that the tool t_i traverses at each *doc*, to machine whatever it can without gouging. If p shares an edge with the stock boundary (i.e. there is an *open edge*), then $A_i(f)$ should sufficiently cover the open-edge for complete machining. The area within p that the tool traverses is given by $A(f)_i \cap p$. Smaller tools have larger accessible areas inside the pocket as compared to larger tools. In fact, for any two tools t_i, t_j with diameters $d_i > d_j$, $(A_i(f) \cap p) \subseteq (A_j(f) \cap p)$. The volume that the tool machines inside the pocket is given by $X(A_i(f) \cap p, h)$. In machining this volume, the tool traverses a volume given by $X(A_i(f), h)$ (Fig. 1(b)). If h is larger than the *doc* of the tool, the volume is removed in layers, each of whose thickness is less than or equal to the *doc* of the tool. Appendix A illustrates the algorithm to calculate accessible area.

2.2. Decomposed area

Consider the case where two tools $t_i, t_j : d_i > d_j$ are used to machine the feature $f(p,h)$. The tool t_i will traverse a region given by $A_i(f)$ at each *doc*. Actual

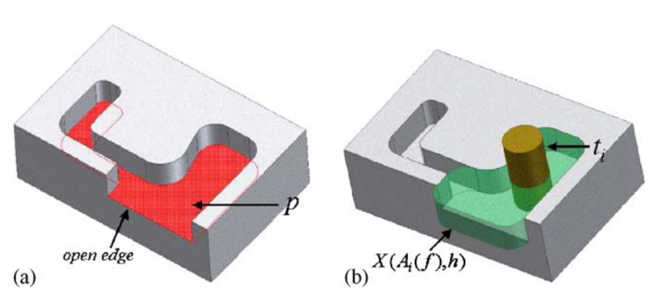

Fig. 1. Finding accessible volume: (a) feature with open-edge, (b) accessible volume given by $X(A_i(f), h)$.