Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 50 (2006) 503–527
computers & industrial engineering
www.elsevier.com/locate/dsw

A genetic algorithm for the optimisation of assembly sequences

Romeo M. Marian *, Lee H.S. Luong, Kazem Abhary

School of Advanced Manufacturing and Mechanical Engineering, University of South Australia, Mawson Lakes, SA 5095, Australia

Available online 10 August 2006

Abstract

This paper describes a Genetic Algorithm (GA) designed to optimise the Assembly Sequence Planning Problem (ASPP), an extremely diverse, large scale and highly constrained combinatorial problem. The modelling of the ASPP problem, which has to be able to encode any industrial-size product with realistic constraints, and the GA have been designed to accommodate any type of assembly plan and component. A number of specific modelling issues necessary for understanding the manner in which the algorithm works and how it relates to real-life problems, are succinctly presented, as they have to be taken into account/adapted/solved prior to Solving and Optimising (S/O) the problem. The GA has a classical structure but modified genetic operators, to avoid the combinatorial explosion. It works only with feasible assembly sequences and has the ability to search the entire solution space of full-scale, unabridged problems of industrial size. A case study illustrates the application of the proposed GA for a 25-components product.

Keywords: Assembly sequence planning; Representation; Precedence relations; Genetic algorithms; Genetic operators; Combinatorial optimisation

1. Introduction

Assembly is an obligatory process for all multi-component manufactured goods. Assembly contributes significantly to both cost and lead-time of a product (20–50%, sometimes even more (Nof, Wilbert, & Warnecke, 1997), approaching an astounding 90% in specific areas in micro-technologies and electronics).

Assembly Sequence Planning is part of Assembly Planning. An assembly sequence is the most important part of an assembly plan and it affects other aspects of the assembly process – resources, assembly line layout, efficiency and cost – as well as various details in the product design. Automating the generation of assembly sequences and their optimisation can ensure the competitivity of manufactured goods and increase profit margins.

This paper focuses on a GA designed to optimise the assembly sequence of any type of mechanical products for problems of industrial size by overcoming the difficulties associated with the size and character of the ASPP. Other, associated aspects, necessary for the understanding of the manner in which the algorithm works

* Corresponding author. Tel.: +618 83025275; fax: +618 83023380.
E-mail address: romeo.marian@unisa.edu.au (R.M. Marian).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2005.07.007

and how it relates to real-life problems, are only succinctly presented here as they were detailed in previously published papers.

ASPP is a large-scale combinatorial problem and is highly constrained. The number of potential assembly sequences is proportional to the factorial of the number of parts in the assembly–Combinatorial Explosion (CE) (Wolter, 1989, 1991). Absolute constraints – geometrical, precedence, accessibility and other types of constraints – severely limit the number of feasible assembly sequences. In general terms, automatically generating feasible assembly sequences is, in its full generality, an extraordinarily difficult task, shown to be NP-complete (Wilson & Watkins, 1990) in both two-dimensional and tri-dimensional cases (Kavraki, Latombe, & Wilson, 1993; Wilson, Kavraki, & Perez, 1995). As a result, most of the past and present work in this area have focused on restricted variants of the problem (Kaufman, Wilson, Jones, & Calton, 1996; Romney, Goddard, Goldwasser, & Ramkumar, 1995).

Assembly has an extraordinary diverse character. Assembly can address sequential or non-sequential, linear or non-linear, monotone or non-monotone, coherent or non-coherent assembly sequences or plans (Wolter, 1989) or any combination of those, involving any combination of rigid, elastic, non-elastic, solid, liquid or gaseous components or subassemblies. To be applicable in practice and useful, an assembly sequence planning and optimisation algorithm has to be general enough to accommodate any type of assembly plan and component.

In order to S/O the ASPP, all relevant aspects of the problem have to be properly modelled and the information has to be represented and stored. The quality of the modelling and representations directly impacts on the quality of the results of the subsequent algorithms.

Solving the ASPP is an essential step prior to its optimisation. It can also be considered a standalone problem in itself, when the required output is an automatically generated assembly sequence. The problem is solved, here, by generating a feasible sequence to assemble an n-part product given its description and a number of supplementary constraints (in Section 5.1.).

The optimisation algorithm – a Genetic Algorithm (GA) designed to accommodate the specific requirements of the ASPP – is a population-based search algorithm in the space of solutions and its output is a population of optimal or near-optimal assembly sequences from which the best one/s are selected.

A brief literature review in the next section shows a number of previous attempts to S/O the ASPP and pinpoints critical, limiting issues, not properly adapted to a constrained combinatorial problem. Section 3 presents the research methodology for S/O the ASPP. Section 4 succinctly shows the modelling and representation issues necessary to consider prior to S/O the ASPP.

The GA is able to S/O any type of product of industrial size and realistic constraints. It is presented, along with its operators and its implementation, in Section 5. A case study illustrates the application of the GA for a product with 25 components, in Section 6.

2. A literature review on assembly sequence planning

S/O the ASPP has been attempted, using various approaches, with mixed results. Due to CE, classic optimisation methods failed to optimise industrial-size problems.

AND/OR graphs and a hybrid A^* algorithm were used to represent and select, iteratively and interactively, the optimal assembly sequence in Archimedes, a software package (Kaufman et al., 1996; Jones, Wilson, & Calton, 1998). The planner uses Non-Directional Blocking Graphs (NDBG (Wilson, 1992)) of each subassembly to determine the assembly operations that might be performed to construct a subassembly then it adds constraints.

AND/OR graphs with weighted hyperarcs were also used to store feasible assembly operations for assembly in HighLAP – High Level Assembly Planning, another assembly software package. It selects an assembly plan minimising the costs from the initial nodes up to the goal node (Rohrdanz, Mosemann, & Wahl, 1996).

The representations used for the optimisation approaches presented above either generate and utilize the explicit AND/OR graphs to store and evaluate *all* assembly sequences for a given assembly or interactively select an assembly sequence by successively applying a number of constraints. They are limited either by the maximum number of sequences that can be stored (CE) or risk to lose valuable solutions by artificially