Makespan Scheduling for Assembly Tasks: Extended Abstract

Brian Drabble, David Clements
Computational Intelligence Research Laboratory,
University of Oregon,
1269 University of Oregon,
OR 97403-1269, USA.
Email: drabble@cirl.uoregon.edu, clements@cirl.uoregon.edu

Introduction

A number of applications in the space domain require complex assembly planning and scheduling. For example, assembly, integration and verification, satellite command sequencing and satellite assembly. One of the main objectives in each of these applications is to identify a minimum length schedule or makespan which accomplishs all the activities in the task. This is often difficult to achieve due to the complex interaction of resource and temporal constraints, e.g. variable shift patterns. The aim of this article is to provide a description of some of the techniques being developed at CIRL to develop schedules with minimum makespan. The techniques described in this article are Limited Discrepancy Search (LDS) (?) and the Doubleback Optimizer (?). The current scheduling system is acting as a test bed for further developments including a probablistic version of LDS and the ability to handle more complex concepts such as release times and delivery dates. The remainder of the article provides details of the techniques and describes the ways in which these could be used for applications such as assembly, integration and verification.

Scheduling Technologies

Overview

CIRL's scheduler is a research prototype designed to solve a particular class of problem. In particular, it attempts to generate minimum length, precedence and resource constrained job shop schedules. These problems typically involve manufacturing small numbers of fairly expensive items. To date the scheduler has been tested on a number of problems from airplane assembly manufacturing. The results obtained from these problems have shown that the CIRL scheduler is currently the best solution and manages to find schedules which are between one and three days better that other systems. This can be a significant saving when the costs of one days additional production vary between $100,000 and $1,000,000. The study of the generic features of the aircraft assembly problem has identified a number of similar tasks in the space domain. For example, in assembly, integration and verification (AIV) the aims are to integrate the different AIV schedules for the subsystems of the satellite into a single AIV schedule and minimize the overall cost of the combined schedule. By minimizing the length of the schedule the cost of using expensive test resources e.g. trollies, chambers, etc, is reduced. The AIV task has been shown to be a successful application for AI planning and scheduling technologies (?) with a deployed system for The European Space Agency being routinely used for payload checkout on the Ariane IV program. An additional application which would benefit from minimising makespan would be satellite imagery in which there are a large number of requests and limited time and resources.

Core Technology

The core technology of the CIRL scheduler is based around two different techniques to generate schedules. These are:

- The use of heuristics to attempt to generate a reasonably short "seed" schedule.
- The "seed" schedule produced by the heuristics is then fed to an optimizer that uses Doubleback optimization to shorten the schedule further.

The hueristic search strategy Limited Discrepancy Search (LDS) is used to generate good "seed" schedules using the heuristics, and does so in a manner that ensures that the schedules are not merely minor variants of each other. Work with the scheduler has discovered that by feeding multiple, different schedules to the optimizer that the scheduler can generally produce a better schedule than than can be obtained by feeding the single schedule that was derived using purely the heuristics.

Doubleback Optimization

Doubleback optimization, also known as schedule packing or bin packing, involves "sloshing" a candidate schedule, repeatedly, right and left within a scheduling window. This has a remarkable impact on the length of most schedules. The Doubleback process is analogous to filling a box with blocks and then shaking the box. Shaking the box will almost always result in a denser packing of blocks. Likewise, in schedules, Doubleback almost always results in a denser packing of tasks in a schedule.

Limited Discrepancy Search (LDS) and Heuristics

When generating a candidate schedule, many, many decision points are reached along the way about what task to place when. In scheduling, heuristics are used to attempt to predict which choices will result in the best (i.e. shortest) schedule. However, there are no known perfect scheduling heuristics. A perfect heuristic, if one existed, would enable the scheduler to always place tasks at the right time. An optimal schedule could be generated every time. LDS helps make up for the absence of a perfect heuristic, particularly in the case where you have a good heuristic. (In fact, the better your heuristic, the better LDS works.)

In scheduling problems of any size it is unlikely that always using a merely good heuristic will get you really close to an optimal schedule. A merely good heuristic will be incorrect some of the time. As the complexity of scheduling problems increases, the number of decisions guided by the heuristic also increases. The more decisions made, the more likely it is that some of them are going to be incorrect. How does LDS help address this problem? LDS is a systematic method for disregarding the recommendation of a heuristic a *limited* number of times (thus the name LDS) when generating a schedule. With LDS, schedules are generated repeatedly, each of them following the heuristic for all decisions except one. The decision at which the heuristic is ignored is different in each schedule.

If the heuristic leads to only one incorrect decision, then using LDS1 (the fastest form of LDS) will lead to a perfect schedule. Even if the heuristic leads to more than one incorrect decision (which is usually the case) then LDS1 will likely lead to a better solution then always following the heuristic.

Features of the CIRL Scheduler

The current CIRL scheduler currently supports a number of features and these are as follows:

- Fixed duration tasks with fixed resource usage for the duration of the task.
- Resource availability that is constant over time, or that varies cyclically over time, e.g. shifts
- Allocatable (i.e. non-consumable resources).
- Precedence constraints between tasks that state "this task cannot start until this other task has finished". An arbitrary network of precedences can be specified as long as there are no cycles present.

- A general constraint that all tasks running at each particular time must use no more than the amount of resources that are available at that time.

Research efforts are continuing to develop the functionality of the scheduler and these include using probablistic measures in seed generation and including release times and due dates in the schedulers representation.

References

Aarup, M. , Arentoft, M.M., Parrod, Y., Stokes, I., Vadon, H. and Stader, J. (1994) Optimum-AIV: A Knowledge-Based Planning and Scheduling System for Spacecraft AIV, in Intelligent Scheduling (eds. Zweben, M. and Fox, M.S.), pp. 451-469, Morgan Kaufmann.

Crawford, J., An Approach to Resource Constrained Project Scheduling, in the proceedings of the Artificial Intelligence and Manufacturing Research Planning Workshop, Alberquerque, New Mexico, June 24-26, 1996. Also available in `http://www.cirl.uoregon.edu/crawford/papers/albur.ps`

Harvey, W, and Ginsberg, M., Limited Discrepancy Search, in the proceedings of IJCAI-95, Montreal, Canada, July, 1995, Also available in `ftp://www.cirl.uoregon.edu/pub/users/ginsberg/papers/lds.ps.gz`.

Acknowledgements

Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under agreements numbered F30602-95-1-0023 and F30602-97-1-0294. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The work reported here was developed by a number of staff and students at CIRL and includes: Tania Bedrax-Weiss, Andrew Parkes, Bart Massey, Matt Ginsberg, David Joslin, David Etherington, Dave Clements, Jimi Crawford and others.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Rome Laboratory, or the U.S. Government.