Online store of the manufacturer for remanufactured products

Untertitel der Diplomarbeit, falls vorhanden

Masterarbeit

zur Erlangung des akademischen Grades
eines Master of Science

an der Karl-Franzens-Universität Graz

vorgelegt von

Yanick Dickbauer, BSc.

am Institut für Produktion und Logistik

Begutachter: Univ.-Prof. Dr.rer.soc.oec. Marc Reimann

Graz, 2017

Abstract

xxx Titel der Arbeit Englisch xxx

xxx Untertitel der Arbeit englisch xxx

xxx Abstract auf Englisch xxx

Kurzfassung

xxx Titel der Diplomarbeit Deutsch xxx

xxx Untertitel der Diplomarbeit

xxx Abstract auf Deutsch xxx

Inhalt

Vorwort und Danksagung

xxx Optionales Vorwort mit Danksagung... xxx

Eidesstattliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen inländischen oder ausländischen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Die vorliegende Fassung entspricht der eingereichten elektronischen Version.

Graz, am xxx. xxx 201x

(Max Mustermann)

Abkürzungsverzeichnis (xxx optional xxx)

Die folgende Auflistung gibt einen Überblick der wichtigsten in dieser Arbeit vorkommenden Begriffe, Akronyme, Abkürzungen und Formelzeichen:

	Abkz.	**Erklärung**
A	AHS	Allgemeinbildende Höhere Schulen
B	Bac	Bachelor (Fachwissenschaft)
	BHS	Berufsbildende Höhere Schulen (HTL, HLW, HAK)
	BK	Brückenkurs
	BORG	Oberstufenrealgymnasium
	Bsp/Bspe	Beispiel/Beispiele (Übungsaufgaben)
E	ECTS	Leistungspunkte im European Credit Transfer System
G	Gym	Gymnasium
P	P & P	paper and pencil (Fragebogen in Papierform)
	PS	Lehrveranstaltungstyp Proseminar (siehe UE)
U	UE	Lehrveranstaltungstyp Übung
	UGO	Uni Graz Online - Onlinesystem der Uni Graz (LV-Anmeldungen etc.)
	Uni	Universität (z. B. Uni Graz: Karl-Franzens Universität Graz)
V	V.I.	Vollständige Induktion
	VO	Lehrveranstaltungstyp Vorlesung
	VU	Lehrveranstaltungstyp Vorlesung mit Übung
W	WS	Wintersemester

1. Introduction

The problem of scheduling jobs and machines in flexible manufactoring systems has been addressed in scientific papers for a vast amount of complex variations. For this thesis a currently relatively unexplored set-up is considered: The scheduling of vehicles in a cyclic flexible manufactoring system. Flexible manufactoring systems are used in a wide range of fields and in occur in a lot of different variations. Additional machines at stages allow for a bypass of possible bottlenecks by increasing throughput capacity. A cyclic system is distinct from the standard version in that the depot is starting point and finish for all jobs in the system.

As automation is becoming increasingly relevant each day, the problem tackled here sets the vehicles carrying jobs between the machines to be automatically guided (AGVs). The approach considering vehicles without driver has been addressed for non cyclic problems in papers like insertpaperhere with special consideration of problem1 and insertpaperhere focusing on problem2. First considerations of the cyclic variant have been made by Blazewicz and Pawlak citationSVCFMS in which they introduced the idea of the vehicles staying in a steady cycle without any waiting time for possible unfinished jobs and the machine schedule itself being given, allowing for focus on the scheduling of the AGVs alone. The first of these premises eliminates the potential of collisions between the AGVs while the second reduces the amount of changeable variables. This thesis builds upon some of the findings of Blazewicz and Pawlak and implements a meta-heuristical approach by introducing a local search.

While the goal of the paper by Blazewicz and Pawlak was to solely minimize the amount of AGVs, here the approach is changed a bit: By fixating the amount of machines in an iteration of the meta-heuristic and simply focusing on the scheduling of a given amount of machines to minimize the makespan, a set of solutions for different amount of AGVs is received and a cost function depending on the amount of machines and the amount of overall delay in the for a given deadline can be used to select a final solution. If the goal remains to minimize the amount of AGVs for production without delay, the cost function can be set to cost any constant c for each machine and Big-M per unit of delay.

Teil I.

xxx Hauptteil 1 xxx

2. Characteristics of the problem

The problem of cylcic flexible manufactoring systems has certain limitations, which will be expanded and described in this chapter. The baseline of the overall problem is that vehicles are used to carry jobs from one machine to the next. This paper will only consider a flow shop approach in which the machines are already sorted in the order, in which the jobs need to be processed. Due to the cyclic nature of this special case of the problem the AGVs are routed to drive on a steady course and through start and depot being the same, the AGVs are constantly repeating laps of said track. This in turn enables us to know the position of an AGV at any given point in the system solely by setting a starting time at which the AGV is "released". The actual position in the system is then acquired by calculating the overall time minus the starting time modulo the lap time.

Since the schedule of the jobs on the machines themselves is considered to be given, after setting the AGVs to automatically pick up the job which is next in the schedule on the next machine and waiting in the outgoing queue of the current machine, the only input needed to be able to achieve a solution is a starting time for each of the AGVs.

Before it is possible to formulate an approach to solve this problem, some rules have to be set first, especially considering the handling of jobs in outgoing queues (jobs which have been completed at a machine but have not yet been picked up by an AGV). Since the system is set to be in constant movement, a machine cannot simply wait for a job no yet finished. Therefore a rule has to be set as to which job is to be picked out of those waiting in line. This rule requires the machine to pick up the next job in chronological order that is both scheduled to follow the job currently being worked on at the next stage, and waiting in the outgoing queue of the current machine. This is proven in the following lemma.

Lemma 1: an AGV passing a machine has to pick up the job which is waiting in the current queue which is set to be the chronologically next job on the following machine.

Proof: Consider a problem with a single AGV and a schedule on machine n+1 set to be [1,2,3]. Jobs 2 and 3 are currently waiting to be picked up in the outgoing queue of machine n, while job 1 is still currently being processed on machine n. Considering the lemma an incoming AGV has to pick up job 2 as it is the next job which is supposed to be processed on machine n+1 and sitting in the outgoing queue of machine n. In this case the AGV now picks up job 3 instead and delivers it to machine n+1, where job 3 is now waiting in the incoming queue. While the AGV is continuing its lap job 1 has now been finished on machine n and is ready for pick up. As the AGV returns to machine n the job 1 is now picked up and delivered to the machine n+1, where it can immediately start to be processed. Somewhere during the continuation of the lap of the AGV job1 is finished on machine n+1 and the machine is free to process a new job. Here however it is unable to do so as in the incoming queue there is still only the job 3, which is supposed to be processed only after the job 2. If the AGV had picked up job 2 in the first round the job could have been started now, so the decision to load job 3 led to an increase in makespan that is at least the minimum of the processing time of job 2 on the second machine and the time it takes the AGV to pick up and deliver job 2.

3. Choosing an approach

Due to the NP-completeness of the problem the general approach was to find a suitable heuristic that would be able to provide a relatively decent solution, while maintaining a low running time. There is a wide choice of heuristical approaches described in detail in a vast amount of papers, the idea was to find one which would be able to provide an approach useful to the problem of scheduling AGVs in the circular system. For this reason the local search meta-heuristic was chosen as an underlying approach, in part because a neighbourhood can be relatively easily defined in a scheduling problem and partly because a special form of the neighbourhood search, the variable neighbourhood search, opened up a new approach to the general problem.

The variable neighbourhood search, as closer described in PAPERHERE by AUTHOR, tries to improve the solution of the standard approach by limiting an iteration of the search to a predefined neighbourhood and switching to a different one for the next iteration. This leads to higher coverage of the whole solution space and a reduced chance to get stuck in a local optimum. While the problem of minimizing the amount of AGVs needed proved to be tricky to divert into neighbourhoods to apply a variable neighbourhood search, it offered up the possibility of a reformulation of the original problem. In the original problem the goal was solely to minimize the amount of AGVs needed to finish on time. If however the problem is changed to have two objectives, being the minimization of the makespan and the minimization of the amount of vehicles needed, the approach of the variable neighbourhood search can be easily applied by setting the different neighbourhoods to be all scheduling possibilities while varying the amount of machines available. This opens up a new problem: Because a higher amount of machines will generally result in a lower makespan, the solutions gained from different neighbourhoods are now ineligible for unreflected comparison.

Because of this problem a new approach, developed for multidimensional target functions, is introduced: A pareto frontier. AUTHOR2 wrote about the idea of a pareto frontier extensively and showing that it offered the possibility of displaying the optima of a multidimensional problem and allowed for the choice of a preferred solution through implementation of an additional target function (e.g. a cost function of the two dimensions displayed) or simply by personal preference. The pareto frontier was chosen as an instrument for this thesis, because of a reason already touched previously: Due to the nature of scheduling problems increasing the number of AGVs available to transport the jobs between the machines will in its optimal configuration always yield a makespan that is better or as a worst case equal to the makespan received when calculating with one less AGV. This can easily be proven by finding the optimal solution for a given amount of AGVs n and then increasing the amount. All AGVs are assigned exactly the same starting time as the previously attained optimal solution for n AGVs. The newly added AGV can be freely allocated to any starting time. This results in all jobs being picked up either at the exact same time as previously or, due to the newly added AGV, earlier than previously. Therefore the total makespan will at the worst case stay the same(e.g. the newly added machine does not transport any job in the system or the earlier transport of a job simply results in longer time in queue due to the processing time of the previous job)

In this thesis the received pareto frontier will not be filtered through the introduction of a cost function, as this simply picks one of the solutions from the list, instead a comparison will be made for each amount of machines up to a set limit between the heuristical solution and an optimal solution obtained through a brute-force algorithm. The difference in run time and target value will then be used to determine whether the heuristic offers a significant reduction in the first, while maintaining only a slight reduction in the latter.

4. Implementation of the model

Before we are able to try and approach the problem with a local search algorithm, the local search itself has to be defined more clearly. As stated in the previous chapter, the idea is to vary the amount of machines available and find optima for all of those cases to be able to display a pareto frontier. To be able to achieve this it is necessary to adjust the original variable neighbourhood search, in that there is little to no gain by constantly swapping between the neighbourhoods, as the solutions received for the maximum makespan with a given amount of AGVs cannot be directly compared. The swapping of neighbourhoods is therefore reduced to be only once after a given amount of iterations, which is set beforehand, for a given amount of AGVs. This in turn means that each swap of neighbourhood means that the previous neighbourhood is assigned a solution that is then saved in the pareto frontier. As this modified approach has now drifted away considerably from the original approach of variable neighbourhood search, the author deems it necessary to rename the heuristical approach to be more simply called ïterated neighbourhood search".

For the actual implementation of the model a few parameters have to be set. These include the number of iterations that the local search is going to run for per amount of machines, the maximum amount of machines and of course all the parameters set by a specific test environment have to be saved, namely the amount of production stages, the production machines for each stage, the amount of jobs, the lap time of an AGV, the time needed by an AGV to traverse between machines, the schedule of jobs on the production machines and of course the process time for each job on each machine. As noted in a previous chapter, the only input variable is the actual starting time for each of the AGVs.

To venture a bit further into the comparability of the heuristic, this paper introduces multiple variations working on the same base construct. The main difference is in the pertubation step which happens after each iteration that has ßtrandedïn a local optimum. The base version was a simple random jump. The starting time of every AGV, except for AGVs starting at "0", is set to a random starting point and the neighbourhood search starts anew. Before each of these pertubations the currently reached solution is compared to the best so far reached solution for the given amount of AGVs and is saved if it is an improvement. Different from this probabilistic approach, the second variation of the heuristic set out to be deterministic, except for the generation of the starting solution. The idea behind the differences in pertubations is to be able to achieve comparability between efficiency and reliability. The deterministic approach sets all starting times of AGVs not starting at "0"to be starting a set amount of units of time early(if the starting time would be negative, the time is instead counted down starting from the last possible start). As local searches with deterministic pertubations generally suffer from the same problem, namely the pertubation either being too high, resulting in almost random jumps, or being too low, resulting in getting stuck at a local optimum, not being able to leave, the heuristic is run multiple times with varying pertubation lengths.

Teil II.

xxx Hauptteil 2 xxx

5. Kapitel 1 des Hauptteils 2

6. Kapitel 1 des Hauptteils 2

Teil III.

Resümee

7. Zusammenfassung

8. Ausblick

Literatur

Verzeichnisse

Abbildungen

Tabellen

Anhang

Anhang A.

Erster Teil vom Anhang

Anhang B.

Zweiter Teil vom Anhang

Anhang C.

Oben beginnende Überschrift, falls nötig