Multiprocessor Scheduling

Theory and Applications

Multiprocessor Scheduling
Theory and Applications

Edited by
Eugene Levner

I-TECH Education and Publishing

Published by the I-Tech Education and Publishing, Vienna, Austria

Abstracting and non-profit use of the material is permitted with credit to the source. Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. Publisher assumes no responsibility liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained inside. After this work has been published by the I-Tech Education and Publishing, authors have the right to republish it, in whole or part, in any publication of which they are an author or editor, and the make other personal use of the work.

© 2007 I-Tech Education and Publishing
www.i-techonline.com
Additional copies can be obtained from:
publication@ars-journal.com

First published December 2007
Printed in Croatia

A catalogue record for this book is available from the Austrian Library.
Multiprocessor Scheduling: Theory and Applications, Edited by Eugene Levner

p. cm.
ISBN 978-3-902613-02-8
1. Scheduling. 2. Theory and Applications. 3. Levner.

Preface

Scheduling theory is concerned with the optimal allocation of scarce resources (for instance, machines, processors, robots, operators, etc.) to activities over time, with the objective of optimizing one or several performance measures. The study of scheduling started about fifty years ago, being initiated by seminal papers by Johnson (1954) and Bellman (1956). Since then machine scheduling theory have received considerable development. As a result, a great diversity of scheduling models and optimization techniques have been developed that found wide applications in industry, transport and communications. Today, scheduling theory is an integral, generally recognized and rapidly evolving branch of operations research, fruitfully contributing to computer science, artificial intelligence, and industrial engineering and management. The interested reader can find many nice pearls of scheduling theory in textbooks, monographs and handbooks by Tanaev et al. (1994a,b), Pinedo (2001), Leung (2001), Brucker (2007), and Blazewicz et al. (2007).

This book is the result of an initiative launched by Prof. Vedran Kordic, a major goal of which is to continue a good tradition - to bring together reputable researchers from different countries in order to provide a comprehensive coverage of advanced and modern topics in scheduling not yet reflected by other books. The virtual consortium of the authors has been created by using electronic exchanges; it comprises 50 authors from 18 different countries who have submitted 23 contributions to this collective product. In this sense, the volume in your hands can be added to a bookshelf with similar collective publications in scheduling, started by Coffman (1976) and successfully continued by Chretienne et al. (1995), Gutin and Punnen (2002), and Leung (2004).

This volume contains four major parts that cover the following directions: the state of the art in theory and algorithms for classical and non-standard scheduling problems; new exact optimization algorithms, approximation algorithms with performance guarantees, heuristics and metaheuristics; novel models and approaches to scheduling; and, last but least, several real-life applications and case studies.

The brief outline of the volume is as follows.

Part I presents tutorials, surveys and comparative studies of several new trends and modern tools in scheduling theory. Chapter 1 is a tutorial on theory of cyclic scheduling. It is included for those readers who are unfamiliar with this area of scheduling theory. Cyclic scheduling models are traditionally used to control repetitive industrial processes and enhance the performance of robotic lines in many industries. A brief overview of cyclic scheduling models arising in manufacturing systems served by robots is presented, started with a discussion of early works appeared in the 1960s. Although the considered scheduling problems are, in general, NP-hard, a graph approach presented in this chapter permits to reduce some special cases to the parametric critical path problem in a graph and solve them in polynomial time.

Chapter 2 describes the so-called multi-agent scheduling models applied to the situations in which the resource allocation process involves different stakeholders (“agents”), each having his/her own set of jobs and interests, and there is no central authority which can solve possible conflicts in resource usage over time. In this case, standard scheduling models become invalid, since rather than computing "optimal solutions", the model is asked to provide useful elements for the negotiation process, which eventually should lead to a stable and acceptable resource allocation. The chapter does not review the whole scope in detail, but rather concentrates on combinatorial models and their applications. Two major mechanisms for generating schedules, auctions and bargaining models, corresponding to different information exchange scenarios, are considered. Known results are reviewed and venues for future research are pointed out.

Chapter 3 considers a class of scheduling problems under unavailability constraints associated, for example, with breakdown periods, maintenance durations and/or setup times. Such problems can be met in different industrial environments in numerous real-life applications. Recent algorithmic approaches proposed to solve these problems are presented, and their complexity and worst-case performance characteristics are discussed. The main attention is devoted to the flow-time minimization in the weighted and unweighted cases, for single-machine and parallel machine scheduling problems.

Chapter 4 is devoted to the analysis of scheduling problems with communication delays. With the increasing importance of parallel computing, the question of how to schedule a set of precedence-constrained tasks on a given computer architecture, with communication delays taken into account, becomes critical. The chapter presents the principal results related to complexity, approximability and non-approximability of scheduling problems in presence of communication delays.

Part II comprising eight chapters is devoted to the design of scheduling algorithms. Here the reader can find a wide variety of algorithms: exact, approximate with performance guarantees, heuristics and meta-heuristics; most algorithms are supplied by the complexity analysis and/or tested computationally.

Chapter 5 deals with a batch version of the single-processor scheduling problem with batch setup times and batch delivery costs, the objective being to find a schedule which minimizes the sum of the weighted number of late jobs and the delivery costs. A new dynamic programming (DP) algorithm which runs in pseudo-polynomial time is proposed. By combining the techniques of binary range search and static interval partitioning, the DP algorithm is converted into a fully polynomial time approximation scheme for the general case. The DP algorithm becomes polynomial for the special cases when jobs have equal weights or equal processing times.

Chapter 6 studies on-line approximation algorithms with performance guarantees for an important class of scheduling problems defined on identical machines, for jobs with arbitrary release times.

Chapter 7 presents a new hybrid metaheuristic for solving the jobshop scheduling problem that combines augmented-neural-networks with genetic algorithm based search.

In Chapter 8 heuristics based on a combination of the guided search and tabu search are considered to minimize the maximum completion time and maximum tardiness in the parallel-machine scheduling problems. Computational characteristics of the proposed heuristics are evaluated through extensive experiments.

Chapter 9 presents a hybrid meta-heuristics based on a combination of the genetic algorithm and the local search aimed to solve the re-entrant flowshop scheduling problems. The hybrid method is compared with the optimal solutions generated by the integer programming technique, and the near optimal solutions generated by a pure genetic algorithm. Computational experiments are performed to illustrate the effectiveness and efficiency of the proposed algorithm.

Chapter 10 is devoted to the design of different hybrid heuristics to schedule a bottleneck machine in a flexible manufacturing system problems with the objective to minimize the total weighted tardiness. Search algorithms based on heuristic improvement and local evolutionary procedures are formulated and computationally compared.

Chapter 11 deals with a multi-objective no-wait flow shop scheduling problem in which the weighted mean completion time and the weighted mean tardiness are to be optimized simultaneously. To tackle this problem, a novel computational technique, inspired by immunology, has emerged, known as artificial immune systems. An effective multi-objective immune algorithm is designed for searching the Pareto-optimal frontier. In order to validate the proposed algorithm, various test problems are designed and the algorithm is compared with a conventional multi-objective genetic algorithm. Comparison metrics, such as the number of Pareto optimal solutions found by the algorithm, error ratio, generational distance, spacing metric, and diversity metric, are applied to validate the algorithm efficiency. The experimental results indicated that the proposed algorithm outperforms the conventional genetic algorithm, especially for the large-sized problems.

Chapter 12 considers a version of the open-shop problem called the concurrent open shop with the objective of minimizing the weighted number of tardy jobs. A branch and bound algorithm is developed. Then, in order to produce approximate solutions in a reasonable time, a heuristic and a tabu search algorithm are proposed.. Computational experiments support the validity and efficiency of the tabu search algorithm.

Part III comprises seven chapters and deals with new models and decision making approaches to scheduling. Chapter 13 addresses an integrative view for the production scheduling problem, namely resources integration, cost elements integration and solution methodologies integration. Among methodologies considered and being integrated together are mathematical programming, constraint programming and metaheuristics. Widely used models and representations for production scheduling problems are reconsidered, and optimization objectives are reviewed. An integration scheme is proposed and performance of approaches is analyzed.

Chapter 14 examines scheduling problems confronted by planners in multi product chemical plants that involve sequencing of jobs with sequence-dependent setup time. Two mixed integer programming (MIP) formulations are suggested, the first one aimed to minimize the total tardiness while the second minimizing the sum of total earliness/tardiness for parallel machine problem.

Chapter 15 presents a novel mixed-integer programming model of the flexible flow line problem that minimizes the makespan. The proposed model considers two main constraints, namely blocking processors and sequence-dependent setup time between jobs.

Chapter 16 considers the so-called hybrid jobshop problem which is a combination of the standard jobshop and parallel machine scheduling problems with the objective of minimizing the total tardiness. The problem has real-life applications in the semiconductor manufacturing or in the paper industries. Efficient heuristic methods to solve the problem, namely, genetic algorithms and ant colony heuristics, are discussed.

Chapter 17 develops the methodology of dynamical gradient Artificial Neural Networks for solving the identical parallel machine scheduling problem with the makespan criterion (which is known to be NP-hard even for the case of two identical parallel machines). A Hopfield-like network is proposed that uses time-varying penalty parameters. A novel time-varying penalty method that guarantees feasible and near optimal solutions for solving the problem is suggested and compared computationally with the known LPT heuristic.

In Chapter 18 a dynamic heuristic rule-based approach is proposed to solve the resource constrained scheduling problem in an FMS, and to determine the best routes of the parts, which have routing flexibility. The performance of the proposed rule-based system is compared with single dispatching rules.

Chapter 19 develops a geometric approach to modeling a large class of multithreaded programs sharing resources and to scheduling concurrent real-time processes. This chapter demonstrates a non-trivial interplay between geometric approaches and real-time programming. An experimental implementation allowed to validate the method and provided encouraging results.

Part IV comprises four chapters and introduces real-life applications of scheduling theory and case studies in the sheet metal shop (Chapter 20), baggage handling systems (Chapter 21), large-scale supply chains (Chapter 22), and semiconductor manufacturing and photolithography systems (Chapter 23).

Summing up the wide range of issues presented in the book, it can be addressed to a quite broad audience, including both academic researchers and practitioners in halls of industries interested in scheduling theory and its applications. Also, it is heartily recommended to graduate and PhD students in operations research, management science, business administration, computer science/engineering, industrial engineering and management, information systems, and applied mathematics.

This book is the result of many collaborating parties. I gratefully acknowledge the assistance provided by Dr. Vedran Kordic, Editor-in-Chief of the book series, who initiated this project, and thank all the authors who contributed to the volume.

References

Bellman, R., (1956). Mathematical aspects of scheduling theory. Journal of Society of Industrial and Applied Mathematics 4, 168–205.

Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., and Weglarz (2007), Handbook on Scheduling, From Theory to Applications, Springer. Berlin.

Brucker, P. (2007), Scheduling Algorithms, Springer, 5th edition, Berlin.

Chretienne, P., Coffman, E.G., Lenstra, J.K., Liu, Z. (eds.) (1995), Scheduling Theory and its Applications, Wiley, New York.

Coffman, E.G., Jr. (ed.), (1976), Scheduling in Computer and Job Shop Systems, Wiley, New York.

Gutin, G. and Punnen, A.P. (eds.) (2002), The Traveling Salesman Problem and Its Variations, Springer, Berlin, 848 p.

Johnson, S.M. (1954). Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly 1, 61–68.

Lawler, E., Lenstra, J., Rinnooy Kan, A., and Shmoys, D. (1985) The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Wiley, New York.

Leung, J.Y.-T. (ed.) (2004), Handbook of Scheduling: Algorithms, Models, and Performance Analysis, Chapman & Hall/CRC, Boca Raton

Pinedo, M. (2001), Scheduling: Theory, Algorithms and Systems, Prentice Hall, Englewood Cliffs.

Tanaev, V.S., Gordon, V.S., and Shafransky, Ya.M. (1994), Scheduling Theory. Single-Stage Systems, Kluwer, Dordrecht.

Tanaev, V.S., Sotskov,Y.N. and Strusevich, V.A.. (1994), Scheduling Theory. Multi-Stage Systems, Kluwer, Dordrecht.

Eugene Levner

September 10,2007

Contents

Part IV. Real-Life Applications and Case Studies

1

Cyclic Scheduling in Robotic Cells: An Extension of Basic Models in Machine Scheduling Theory

Eugene Levner[1], Vladimir Kats[2] and David Alcaide López De Pablo[3]
[1]Holon Institute of Technology, Holon, [2]Institute of Industrial Mathematics, Beer-Sheva,
[3]University of La Laguna, La Laguna, Tenerife
[1, 2] Israel, [3]Spain

1. Introduction

There is a growing interest on cyclic scheduling problems both in the scheduling literature and among practitioners in the industrial world. There are numerous examples of applications of cyclic scheduling problems in different industries (see, e.g., Hall (1999), Pinedo (2001)), automatic control (Romanovskii (1967), Cohen et al. (1985)), multi-processor computations (Hanen and Munier (1995), Kats and Levner (2003)), robotics (Livshits et al. (1974), Kats and Mikhailetskii (1980), Kats (1982), Sethi et al. (1992), Lei (1993), Kats and Levner (1997a, 1997b), Hall (1999), Crama et al. (2000), Agnetis and Pacciarelli (2000), Dawande et al. (2005, 2007)), and in communications and transport (Dauscha et al. (1985), Sharma and Paradkar (1995), Kubiak (2005)). It is, perhaps, a surprising thing that many facts in scheduling theory obtained as early as in the 1960s, are re-discovered and re-rediscovered by the next generations of researchers. About two decades ago, this fact was noticed by Serafini and Ukovich (1989).

The present survey uniformly addresses cyclic scheduling problems through the prism of the classical machine scheduling theory focusing on their features that are common for all aforementioned applications. Historically, the scheduling literature considered periodic machine scheduling problems in two major classes – called *flowshop* and *jobshop* - in which setup and transportation times were assumed insignificant. Indeed, many machining centers can quickly switch tools, so the setup times for these situations may be small or negligible. There are a lot of results about cyclic flowshop and jobshop problems with negligible setup/transportation times. Advantages of cyclic scheduling policies over conventional (non-cyclic) scheduling in flexible manufacturing are widely discussed in the literature, we refer the interested reader to Karabati and Kouvelis (1996), Lee and Posner (1997), Hall et al. (2002), Seo and Lee (2002), Timkovsky (2004), Dawande et al. (2007), and numerous references therein.

At the same time, modern flexible manufacturing systems are supplied by computer-controlled hoists, robots and other material handling devices such that the transportation and setup operation times are significant and should not be ignored. Robots have become a standard tool to serve cyclic transportation and assembling/disassembling processes in manufacturing of airplanes, automobiles, semiconductors, printed circuit boards, food products, pharmaceutics and cosmetics. Robots have expanded production capabilities in the manufacturing world making the assembly process faster, more efficient and precise than ever before. Robots save workers from tedious and dull assembly line jobs, and increase production and savings in the processes. As larger and more complex robotic cells are implemented, more sophisticated planning and scheduling models and algorithms are required to perform and optimize these processes.

The cyclic scheduling problems, in which setup operations are performed by automatic transporting devices, constitute a vast subclass of cyclic problems. Robots or other automatic devices are explicitly introduced into the models and treated as special purpose machines. In this chapter, we will focus on three major classes of cyclic scheduling problems – flowshop, jobshop, and parallel machine shop.

The chapter is structured as follows. Section 2 is a historical overview, with the main attention being paid to the early works of the 1960s. Section 3 recalls three orthodox classes of scheduling theory: flowshop, jobshop, and PERT-shop. Each of these classes can be extended in two directions: (a) for describing periodic processes with negligible setups, and (b) for describing periodic processes in robotic cells where setups and transportation times are non-negligible. In Section 4 we consider an extension of the cyclic PERT-shop, called the cyclic FMS-shop and demonstrate that its important special case can be solved efficiently by using a graph approach. Section 5 concludes the chapter.

2. Brief Historical Overview

Cyclic scheduling problems have been introduced in the scheduling literature in the early 1960s, some of them assuming setup/transportation times negligible while other explicitly treating material handling devices with non-negligible operation times.

Cyclic Flowshop. Cuninghame-Greene (1960, 1962) has described periodic industrial processes, which in today's terminology might be classified as a *cyclic flowshop* (without setups and robots), and suggested an algebraic method for finding minimum cycle time using matrix multiplication in which one writes "addition" in place of multiplication and operation "max" instead of addition. This (max, +)–algebra has become popular in the 1980s (see, e.g. Cuninghame-Greene (1979), Cohen et al. (1985), Baccelli et al. (1992)) and is presently used for solving the cyclic flowshop without robots, see, e.g., Hanen (1994), Hanen and Munier (1995), Lee (2000), and Seo and Lee (2002).

Independently of the latter research, Degtyarev and Timkovsky (1976) and Timkovsky (1977) have studied so-called *spyral cyclograms* widely used in the Soviet electronic industry; they introduced a generalized shop structure which they called a "*cycle shop*". Using a more standard terminology, we might say that these authors have been the first to study a *flowshop with reentrant machines* which includes, as special cases, many variants of the basic flowshop, for instance, the reentrant flowshop of Graves et al. (1983), V-shop of Lev and Adiri (1984), cyclic robotic flowshop of Kats and Levner (1997, 1998, 2002). The interested reader is referred to Middendorf and Timkovsky (2002) and Timkovsky (2004) for more details.

Cyclic Robotic Flowshop. In the beginning of 1960s, a group of Byelorussian mathematicians (Suprunenko et al. (1962), Aizenshtat (1963), Tanaev (1964), and others) investigated cyclic processes in manufacturing lines served by transporting devices. The latters differ from other machines in their physical characteristics and functioning. These authors have introduced a cyclic robotic flowshop problem and suggested, in particular, a combinatorial method called *the method of forbidden intervals* which today is being developed further by different authors for various cyclic robotic scheduling problems (see, for example, Livshits et al. (1974), Levner et al. (1997), Kats et al. (1999), Che and Chu (2005a, 2005b), Chu (2006), Che et al. (2002, 2003)). A thorough review in this area can be found in the surveys by Hall (1999), Crama et al. (2000), Manier and Bloch (2003), and Dawande et al. (2005, 2007).

Cyclic PERT-shop. The following cyclic PERT-shop problem has originated in the work by Romanovskii (1967). There is a set S of n partially ordered operations, called *generic operations*, to be processed on machines. As in the classic (non-cyclic) PERT/CPM problem, each operation is done by a dedicated machine and there is sufficiently many machines to perform all operations; so the question of scheduling operations on machines vanishes. Each operation i has processing time $p_i > 0$ and must be performed periodically with the same period T, infinitely many times.

For each operation i, let $<i, k>$ denote the kth execution (or, repetition) of operation i in a schedule (here k is any positive integer). *Precedence relations* are defined as follows (here we use a slightly different notation than that given by Romanovskii). If a generic operation i precedes a generic operation j, the corresponding *edge* (i, j) is introduced. Any edge (i,j) is supplied by two given values, L_{ij} called the *length*, or *delay*, and H_{ij} called the *height* of the corresponding edge (i, j). The former value is any rational number of any sign while the latter is integer. Then, for a pair of operations i and j, and the given length L_{ij} and height H_{ij}, the following relations are given: for all $k \geq 1$, $t(i,k) + L_{ij} \leq t(j, k + H_{ij})$, where $t(i,k)$ is the starting time of operation $<i, k>$. An edge is called *interior* if its end-nodes belong to the same iteration (or, one can say "to the same block, or pattern") and *backward* (or, *recycling*) if its end-nodes belong to two consecutive blocks.

A schedule is called *periodic* (or *cyclic*) with *cycle time T* if $t(i, k) = t(i,1) + (k-1)T$, for all integer $k \geq 1$, and for all $i \in S$ (see Fig. 1). The problem is to find a periodic schedule (i.e., the starting time $t(i,1)$ of operations) providing a minimum cycle time T, in a graph with the infinite number of edges representing an infinitely repeating process.

Figure 1. The cyclic PERT graph (from Romanovskii, (1967))

In the above seminal paper of 1967, Romanovskii proved the following claims which have been rediscovered later by numerous authors.

- **Claim 1**. Let the heights of interior edges be 0 and the heights of backward edges 1. The minimum cycle time in a periodic PERT graph with the infinite number of edges is equal to the maximum circuit ratio in a corresponding double-weighted finite graph in which the first weight of the arc is its length and the second is its height: $T_{\min} = \max_C \sum L_{ij}/\sum H_{ij}$, where maximum is taken over all circuits C; $\sum L_{ij}$ denotes the total circuit length, and $\sum H_{ij}$ the total circuit height.

- **Claim 2**. The max circuit ratio problem and its version, called the max mean cycle problem, can be reformulated as linear programming problems. The dual to these problems is the parametric critical path problem.
- **Claim 3.** The above problems, namely, the max circuit ratio problem and the max mean cycle problem, can be solved by using the iterative Howard-type dynamic programming algorithm more efficiently than by linear programming. (The basic Howard algorithm is published in Howard (1960)).
- **Claim 4.** Mean cycle time counted for n repetitions of the first block in an optimal schedule differs from the optimal mean cycle time by $O(1/n)$.

The interested reader can find these or similar claims discovered independently, for example, in Reiter (1968), Ramchandani (1973), Karp (1978), Gondran and Minoux (1985), Cohen et al. (1985), Hillion and Proth (1989), McCormick et al. (1989), Chretienne (1991), Lei and Liu (2001), Roundy (1992), Ioachim and Soumis (1995), Lee and Posner (1997), Hanen (1994), Hanen and Munier (1995), Levner and Kats (1998), Dasdan et al. (1999), Hall et al. (2002). In recent years, the cyclic PERT-shop has been studied for more sophisticated modifications, with the number of machines limited and resource constraints added (Lei (1993), Hanen (1994), Hanen and Munier (1995), Kats and Levner (2002), Brucker et al. (2002), Kampmeyer (2006)).

3. Basic Definitions and Illustrations

In this section, we recall several basic definitions from the scheduling theory. Machine scheduling is the allocation of a set of machines and other well-defined resources to a set of given jobs, consisting of operations, subject to some pre-determined constraints, in order to satisfy a specific objective. A problem instance consists of a set of m machines, a set of n jobs is to be processed sequentially on all machines, where each operation is performed on exactly one machine; thus, each job is a set of operations each associated with a machine.

Depending on how the jobs are executed at the shop (i.e. what is the routing in which jobs visit machines), the manufacturing systems are classified as:

- flow shops, where all jobs are performed sequentially, and have the same processing sequence (routing) on all machines, or
- job shops, where the jobs are performed sequentially but each job has its own processing sequence through the machines,
- parallel machine shop, where sequence of operations is partially ordered and several operations of any individual job can be performed *simultaneously* on several parallel machines.

Formal descriptions of these problems can be found in Levner (1991, 1992), Tanaev et al. (1994a, 1994b), Pinedo (2001), Leung (2004), Shtub et al. (1994), Gupta and Stafford (2006), Brucker (2007), Blazewicz et al. (2007). We will consider their cyclic versions.

The cyclic shop problems are an extension of the classical shop problems. A problem instance again consists of a set of m machines and a set of n jobs (usually called *products,* or *part types*) which is to be processed sequentially on all machines. The machines are requested to process repetitively a *minimal part set*, or *MPS*, where the MPS is defined as the smallest integer multiple of the periodic production requirements for every product. In other words, let $r = (r_1, r_2, \dots, r_n)$ be the production requirements vector defining how many units of each product (j=1,…,n) are to be produced over the planning horizon. Then the MPS is the vector $r_{MPS} = (r_1/q, r_2/q, \dots, r_n/q)$ where q is the greatest common divisor of integers $r_1, r_2, \dots, r_n$. Identical products of different, periodically repeated, replicas of the MPS have the same processing sequences and processing times, whereas different products within an MPS may require different processing sequences of machines and the processing times. The replicas of the MPS are processed through equal time intervals T called *cycle time* and in each cycle, exactly one MPS's replica is introduced into the process and exactly one MPS's replica is completed.

An important subclass of cyclic shop problems are the robotic scheduling problems, in which one or several robots perform transportation operations in the production process. The robot can be considered as an additional machine in the shop whose transportation operations are added to the set of processing operations. However, this "machine" has several specific properties: (i) it is *re-entrant* (that is, any product requires the utilization of the same robot several times during each cycle) and (ii) its setup operations, that is, the times of empty robots between the processing machines, are *non-negligible*.

3.1. Cyclic Robotic Flowshop

In the cyclic robotic flowshop problem it is assumed that a technological processing sequence (route) for n products in an MPS is the same for all products and is repeated infinitely many times. The transportation and feeding operations are done by robots, and the sequences of the robotic operations and technological operations are repeated cyclically. The objective is to find the cyclic schedule with the maximum productivity, that is, the minimum cycle time. In the general case, the robot's route is not given and is to be found as a decision variable.

A possible layout of the cyclic robotic flowshop is presented in Fig. 2.

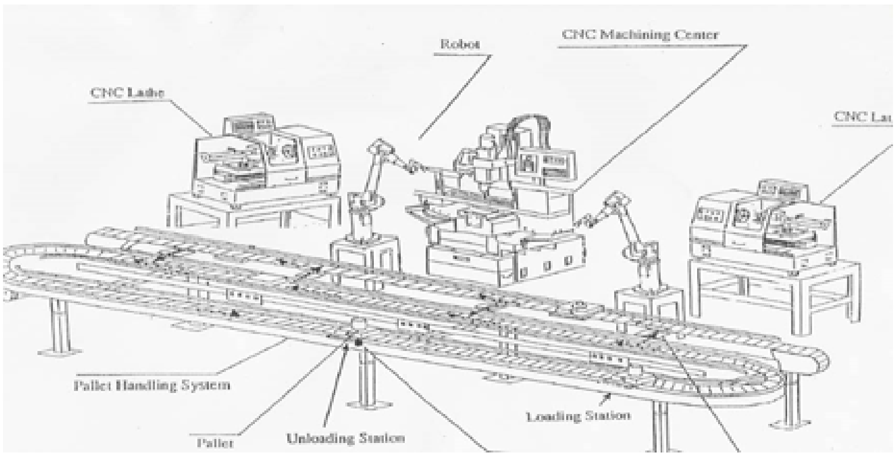

Figure 2. Cyclic Robotic Flowshop

A corresponding Gantt chart depicting coordinated movement of parts and robot is given in Fig. 3. Machines 0 and 6 stand for the loading and unloading stations, correspondingly. Three identical parts are introduced into the system at time 0, 47 and 94, respectively. The bold horizontal lines depict processing operations on the machines while a thin line depicts the route of a single robot between the processing machines. More details can be found in Kats and Levner (1998).

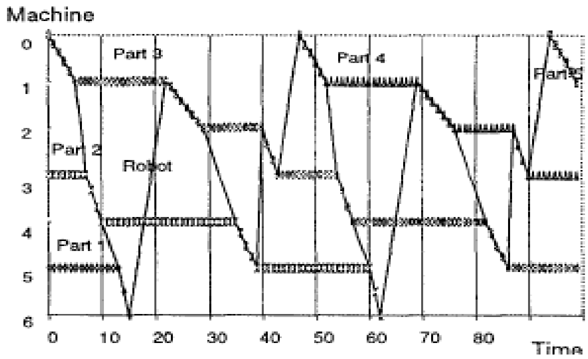

Figure 3. The Gantt chart for cyclic robotic flowshop (from Kats and Levner (1998))

3.2 Cyclic Robotic Jobshop

The cyclic robotic jobshop differs from cyclic robotic flowshop only in that each of n products in MPS has its own route as depicted in Fig. 4.

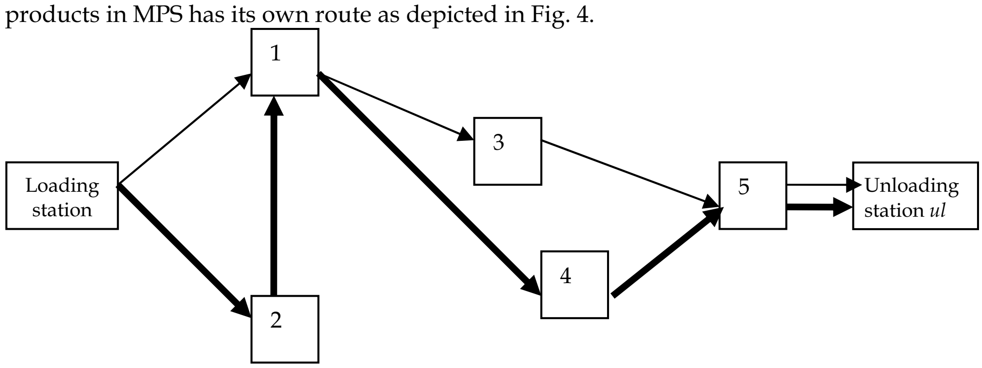

Fig. 4. An example of a simple technological network with two linear product routes and five processing machines, depicted by the squares, where denotes the route for product *a*, and denotes the route for product *b* (from Kats et al. (2007))

The corresponding graphs depicting the sequence of technological operations and robot moves in a jobshop frame are presented in Fig. 5 and 6 .

The corresponding Gantt chart depicting coordinated movement of parts and robots in time is in Fig. 7, where stations 1 to 5 stand for the processing machines and stations 0 and 6 are, correspondingly, the loading and unloading ones. In what follows, we refer to the machines and loading/unloading stations simply as the *stations*.

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below