ISSN: 2150-8194 Vol. 2, No. 8, August 2010

Scientific Research

Intelligent Information Management

Editor-in-Chief: Dr.Bin Wu

ISSN: 2150-8194
9 772150 819009 08

www.scirp.org/journal/iim

Journal Editorial Board

ISSN: 2150-8194 (Print), 2150-8208 (Online)

http://www.scirp.org/journal/iim

Intelligent Information Management, **2010, 2, 437-497**
Published Online August 2010 in SciRes (http://www.SciRP.org/journal/iim/)

TABLE OF CONTENTS

Volume 2 Number 8 August 2010

Intelligent Information Management (IIM)

Journal Information

SUBSCRIPTIONS

The *Intelligent Information Management* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: $50 per issue.

To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements

Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.

E-mail: sub@scirp.org

COPYRIGHT

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: iim@scirp.org

Intelligent Information Management, **2010, 2, 437-446**
doi:10.4236/iim.2010.28054 Published Online August 2010 (http://www.SciRP.org/journal/iim)

China Internet Based Academic Journals Digital Publishing Models Study

Xiangxing Shen[1], Xiangdong Li[1], Yuan Yang[1], Chong Shen[2,3], Kae Hsiang Kwong[3]
[1]*Centre for Studies of Information Resources, Wuhan University, Wuhan, China*
[2]*College of Information Science & Technology, Hainan University, Haikou, China*
[3]*Centre for Intelligent Dynamic Communications, University of Strathclyde, Glasgow, UK*
E-mail: *x.shen@whu.edu.cn*
Received April 1, 2010; *revised May* 7, 2010; *accepted July* 20, 2010

Abstract

The research takes three fundamental Internet Academic Journals (IAJ), which are China National Knowledge Infrastructure (CNKI), Chongqing VIP Information Co., Ltd. (VIP) and Beijing Wanfang Data Co., Ltd. (WFD), as subjects to discuss digital publishing mechanism innovations for China's science information communication system. After introducing main Internet environment, comparing journal characteristics, and explaining publishing & operation modes of the IAJs, the paper analyses important issues such as redundant contents, digitalization issues, illegal journals, standards construction, copyright and benefit distribution. By learning from research practices and open access, a new benefit distribution and copyright management mode based on the Third Party Agency (TPA) is proposed. Future IAJs operation modes are also discussed from the view of scientific research innovation and state of the art platform construction.

Keywords: China, Publishing Mode, Internet Academic Journals, Digital Library, Standard, Illegal Journals

1. Introduction

Science information communication is one of the most important stakeholders in digital publishing. In practice the information carrier plays a vital role while the traditional science information communication theory is based on the understanding of carrier's function. For example, H. Menzel's "information communication process" theory divides the process according to information carriers and A. Mikhailov's "science information communication" theory defines books, journal literatures and other paper carriers as the foundation for constructing science information communication system [1]. Each theory emphasizes that information carriers are the possibilities for science information communication. Traditional science information communication system, which was independent in form of printed publications, has gradually adapted to the coexistence with digital publications such as electronic books, IAJs, etc.

The paper is supported by National Natural Science Foundation of China Youth project "the Research of Similar Information Dissemination Dynamics under the Anti-Plagiarism Realization" (No. 60803080).

Digital publishing refers to publications published using digital technology. As digital technology evolves with the development of computer hardware and software as well communication technology, digital publishing has developed from early stage electronic publishing, desktop publishing, web publishing and network publishing to today's Internet publishing. Regardless the names, such publishing is a succeeding publications of books, journals, newspapers and magnetic medium publications including audio and video tapes, which is a brand-new cultural production and transmission mode. Supported by digital technology and carried by computer networks, digital publishing brings about a revolutionary transformation of traditional publishing forms and patterns [2].

Of various forms of digital publications, IAJ which shares traditional printed journals properties is a major tool in science information communication system. In this research the typical CNKI IAJ is taken by us as the main form of digital publishing to discuss mechanism innovation issues in China.

2. Status Quo of IAJs in China

2.1. Network Environment of IAJs in China

Two backbone networks, the China Education & Research Network (CERNET) and China Science & Technology Network (CSTNET) were constructed by the China state [3], as shown in **Figure 1** with a description on geographical topology. The two networks are now managed by the Ministry of Education and Ministry of Science and Technology, separately. The construction, operation & maintenance and service update of the two networks are undertaken by leading institutes such as Tsinghua University and Chinese Academy of Sciences, respectively. In 1996, the two networks were officially listed as two of four China backbone infrastructure networks by the China state council.

CERNET was finished construction in 1994 and is the first nationwide Internet backbone network in China. At present, the transmission speed of CERNET's main stream is up to 2.5 - 10 Gbps and that of its local substream is up to 155 Mbps - 2.5 Gbps. The network covers more than 200 cities of 32 state provinces including autonomous regions and Hong Kong special administrative region. It physically possesses over 30,000 km optical fiber with independent international outreaching throughput exceeding 5 G. 10 regional centers and 38 provincial nodes are in operation with one central network center located at Tsinghua University, Beijing. The giant network also links more than 2000 universities, educational institutions and scientific research units serving over 20 million users, which becomes a fundamental and indispensable educational information platform in China [4].

CSTNET was established in 1989 as a scientific research network of Chinese Academy of Sciences and national public network. The transmission speed up of CSTNET's main trunk is up to 2.5 Gbps to date and that of its local sub stream is up to 155 Mbps - 2.5 Gbps. The inter-network data exchange rate between CERNET and CSTNET is up to 2 Gbps while CSTNET's international outreaching throughput exceeds 4.5 G. Similar to CERNET, the network also covers 30 provinces, autonomous regions and Chinese Taiwan. The domestic backbone network is composed of 13 distributed city sub-centers including Beijing, Changchun, Shenyang, Shanghai, Guangzhou, Chengdu and Lanzhou, making itself another basic scientific information platform in China [5].

2.2. Development of IAJs in China

In 1989, Chongqing VIP issued the earliest academic digital journal in China in the form of floppy disk. In 1996, a large-scale integrated Chinese academic Journal (CD Edition) was developed by China Academic Journals Electronic Publishing House (CAJEPH) which marks the beginning of journal digital publishing in China. In 1997, WFD & company established the first technology information WWW website. The company focused on the association between the digital publishing and Internet. In June 1998, Chinese Academic Journal (CD Edition) was officially listed on the web. The Chinese academic Journal Full-Text Database marks an important milestone and initiates a new era of IAJ publishing in China.

The National Knowledge Infrastructure and its application such as knowledge based economy, as the fourth main infrastructure after energy, transportation and telecommunication, were identified and agreed in different degree by America, Britain and other developed countries governments. Increasing attention has been also paid on China National Knowledge Infrastructure (CNKI)

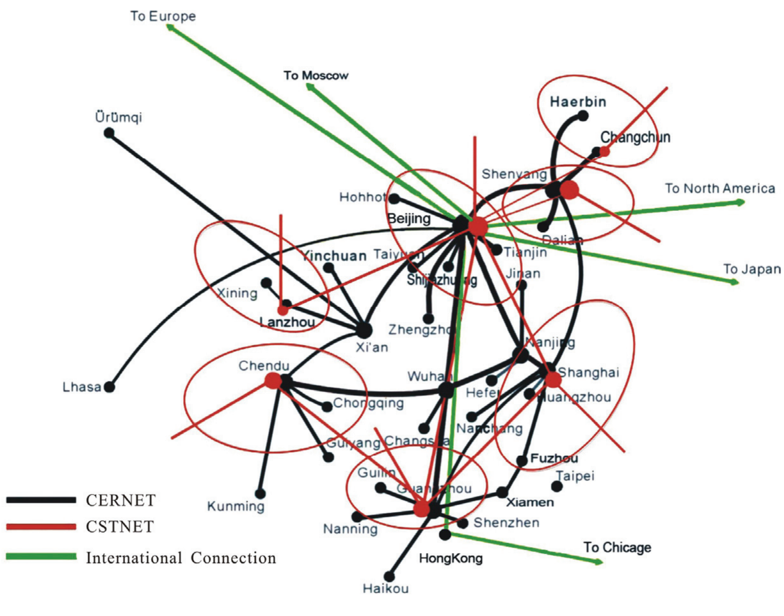

Figure 1. Topology of CERNET and CSTNET.

in China. Under the leaderships of the Ministry of Education of the People's Republic of China (MEC), the Ministry of Science and Technology of the People's Republic of China (MSTC), General Administration of Press and Publication of the People's Republic of China (GAPPC) and other ministries or commissions of China, relying on the support from libraries, information service centers and almost all Chinese academic journal editorial boards (CAJEB), Tong Fang Knowledge & Network Technology Co., Ltd (TKNT) has developed CNKI as a national development project with several internet editions, which promotes Chinese IAJs to an advanced level in terms of literatures volume and data collecting technologies.

The 3 IAJs VIP, CNKI (TKNT) and WFD so far have effectively collected printed journals into respective electronic journal databases by cooperating with printed journal editorial boards. These databases are formal and serialized Internet publications with electronic journal serial numbers approved by the GAPPC. Every CAJEBs is administrated by competent departments under one of the ministries or commissions under the China State Council which disburse funds to pay editorial staff's salaries and daily operation expenditures. In order to protect intellectual property right, coordinated by the GAPPC and the ministries and commissions the three organizations established copyright monitoring systems with various types of CAJEBs to mutually reuse resources between databases, which forms the basic science information communication mode based on Chinese IAJs as shown in **Figure 2**.

2.3. Status Quo of Construction of Chinese IAJ Literature Database

After more than 20 years development, Chinese IAJ literature database market has been divided and shared by VIP, CNKI and WFD with other small-sized specialized literature databases. China now owns totally 9,468 kinds of Chinese journals.

VIP's Chinese scientific and technical journals full-text database collects more than 20 million published literatures in over 8,000 kinds of domestic journals,

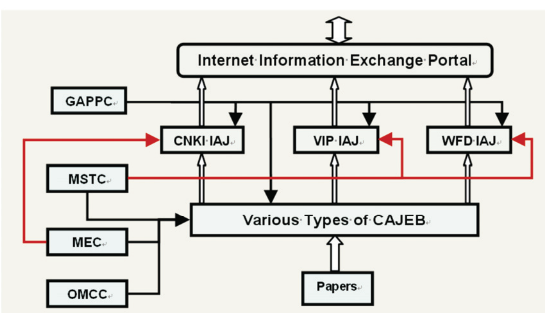

Figure 2. IAJ science information communication modes.

which are divided into eight special areas: social sciences, natural sciences, engineering technology, agricultural science, medicine and hygiene, economy and management, educational science, and library and information science. It is also an important strategic partner of Google, and one of the largest cooperative Chinese content websites of Google Scholar. However, the website itself does not present any English interface, as shown in **Table 1**.

CNKI has experienced 3 phases of development with corresponding name changes: earlier Chinese academic journal (CD Edition), then Chinese academic journal network and currently a brand new knowledge service platform named China's Knowledge Resources Database (CKRD). Apart from accommodating academic journals, it also provides Internet publishing for a series of databases including academic conference papers, doctoral and master's thesis, newspapers and other literatures using self-developed CNKI knowledge network platform for management and data retrieval. It collects 8,893 kinds of domestic journals containing more than 23.45 million of full-text documents, covers nearly all fields such as natural sciences, engineering technology, agriculture, philosophy, medicine and humanities, social sciences, etc., as shown in **Table 1**. By October 2006, CNKI had already owned more than 17,000 different customers covering universities, research institutions and government organizations. It is also subscribed and frequently used by university libraries outside China such as Harvard University library and Oxford University DL, and large-scale public libraries like National DL of Singapore [6]. CNKI is therefore identified as one of the most representative IAJs in China.

WFD's Chinese digital journal group specialized in eight categories with more than 100 sub-items: philosophy, politics and law, social sciences, economy and finance, education, science, culture and art; fundamental science, medicine and hygiene, agricultural science, and industrial technology. It includes 6065 kinds of domestic journals and more than 12.9 million of full-text literatures. The website provides use-friendly feathers such as an English interface while its Chinese journals can be easily linked to various classified English subjects. However, only the Chinese interface provides the list of all journal names.

2.4. Publishing and Operation Mode of CNKI IAJ

The central website and eight regional exchange service centers of CNKI IAJ are established and maintained by TKNT. **Figure 3** presents the topology framework and connection methods [7]. The website has been supported by two completely identical central sites established at

CERNET and China Broadband Network (CHINANET), separately. It provides services through "resource database" and "knowledge database". It also provides jointly-constructed resources and shared knowledge information. Each central website updates knowledge resources, full-text service for CNKI Knowledge Network Management Service Centers and CNKI Mirror sites in different cities through KNS5.0 (Knowledge Network Service 5.0) daily. Meanwhile, the website includes internet retrieval service for institutions and individuals using CNKI knowledge resource and online digital submission service. CNKI IAJ has the following operation modes: professional customers such as universities, research institutions, public libraries, government, enterprises, secondary and primary schools purchase services by database wholesale or remote packaged database installation. The databases then established on local servers in the form of mirror sites provide service for local area network users. The subscription fee is charged per network flow, individual reader purchase and accounts download. The users can pay by dedicated CNKI cards, normal bank cards or pre-paid telecommunication cards. The organization also gets revenue from paid information and analysis, advertising and government support.

2.5. CNKI IAJ's Publishing Institutions and Work Division

TKNT, CAJEPH and Tsinghua Tongfang Optical Disc Co., Ltd. (TTOD) have worked together to deliver the CNKI project. **Figure 4** shows the work division undertaken by different partners for CNKI's publishing and issuing [8].

For CNKI editing, publishing and journal distribution, CAJEPH is responsible for contacting various academic journal editorial boards, university libraries, national degree authorities and conference paper publishers to obtain electronic journal publications approvals, discuss potential product, optimize journal content, etc. CAJEPH has the high-level copyright to editing all journal parts, CNKI full-text journal database and other databases; TTOD makes use of the content resources provided by the press to conduct research & development, improve publishing process and provide online storage products and relevant techniques for CNKI distribution; TKNT provides technical support to network publishing platform promotes marketing and monitors sales.

3. Existing Problems of Chinese IAJs and Analysis

3.1. Duplicated Construction Issues

As shown in **Figure 2**, VIP, CNKI and WFD all fetch resources from CAJEBs. Depending on different company guidelines and operation models, they contact CAJEBs individually to obtain copyrights for publishing

Table 1. Internet database of 3 major Chinese academic journals.

Content	VIP database	CNKI database	WFD database
IAJ site	http://www.cqvip.com/	http://www.edu.cnki.net	http://www.wanfangdata.com.cn/Default.aspx
IAJ mirror site	http://vip.hbdlib.cn/	http://cnki2.lib.whu.edu.cn/kns50/index.aspx	http://g.wanfangdata. com.cn/Default.aspx
Site location	Chongqing	Beijing	Beijing
	CSTNET non-backbone	CERNET backbone	CSTNET backbone
	Network node	Network node	Network node
Date	1989 ～	1994 ～	1997 ～
Product forms	Web version	Web version	Web version
	Mirror version	Mirror version	Mirror version
	CD-ROM version	CD-ROM version	CD-ROM version
	Traffic billing	Traffic billing	Traffic billing
			Mobile phone billing
Document format	PDF	PDF or CAJ	PDF
Journal number	8,000	9,100	6,669
Journal papers	20,000,000	34,215,263	14,691,014
Conference papers	no	1,351,056	1,579,437
Academic dissertation	no	1,078,450	1,625,297
Google scholar	yes	no	no
Update frequency	Daily	Daily	Weekly

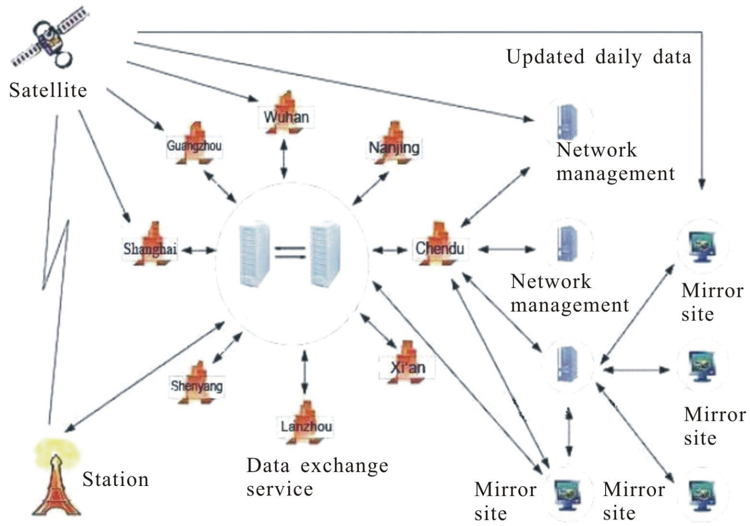

Figure 3. Topology framework of CNKI IA.

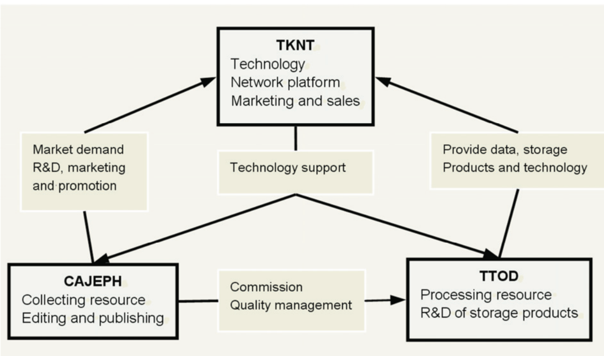

Figure 4. CNKI IAJ's publishing institutions and work division.

printed journal resources on the Internet, without know by each other. On the other hand, neither CAJEBs nor the authors aware of exclusive policy on duplicated publishing of journals and publications, which increase the possibility that more than one IAJ publish the journal and publication. The worse scenario is the same journal can be published simultaneously by three IAJs in this case, which causes resource repetition among the three IAJs. Investigating all printed journals collected by the three IAJs by the end of 2006, a literature [9] analyses the repetition through ISSN, journal names and serial numbers. It points out that 4,413 kinds of printed journals are collected at same time by all of the three IAJs, taking up 57%, 71% and 79% of the total journals of CNKI, VIP and WFD, respectively. Among the rest, 561, 849 and 123 kinds of printed journals are collected at same time by CNKI and VIP, CNKI and WFD, VIP and WFD, respectively. For journals which are collected only by one of the three IAJs, CNKI, VIP or WFD, the figures are 1,869, 1,119 and 170 kinds, respectively, as shown in **Figure 5**.

The duplicated resource construction indicates that every IAJ focuses on its own business scope under different leadership or relevant superior department. There is no mutually agreed planning or overall strategy, leading to waste in journal producing, purchasing and usage. From printed journals to electronic ones, each IAJ separately carries out editing, unpacking, scanning, checking and indexing, which directly causes duplicated production and waste in human resource, roll material and finance. The problem is there are many duplicated journals as well as exclusively collected ones simultaneously. With limited funds, universities and colleges libraries as the main purchasers have to spend a big amount to purchase electronic journals with a repetition rate over 50%. And users have to use more than one IAJ to improve the scope coverage which makes their burdens heavier.

3.2. Paper "Buy and Sell" and Illegal Academic Journal Issues

Occasionally, you may notice advertisements for buying and selling on university campus, Bulletin Board System (BBS) and instant messaging groups. Some occupations e.g. lecturer, scientific researcher, Ph.D. student do receive random inquire on journal buying. Without a law over cyberspace, lawbreakers can do illegal paper dealings through the Internet. Since they follow an attractive strategy that papers can be written before money is paid and can guarantee the publishing once submitted, someone would buy "the product" and those papers are never peer-reviewed or with any academic standards. The illegal academic journals producers charge expensive fees to gain enormous profits. The issue also becomes very common among undergraduates and postgraduates. The changes in searching "paper ghostwrites" in 2008 can be observed by Google Trends, as shown in **Figure 6**.

The map indicates that ghost written papers increased in the first half year when undergraduates, postgraduates and doctoral students were preparing for thesis/dissertation defenses, and the demand for ghost written papers decreased sharply after they graduated. The buyers are from research institutions including universities and colleges and the papers are mainly published in illegal academic journals. On average, one illegal journal publishes 169 papers per issue. From the distribution of original authors of the 169 papers, universities and colleges are the "disaster areas" which are deceived both actively and passively. The affiliation types of these first authors are shown in **Figure 6**. Many of them are from key Chinese universities and colleges, as shown in **Figure 7**.

The rapid development of the Internet and communication technology has boosted paper buy and sell business, and accelerated illegal journal publishing process. A collection of anonymous users is available behind Internet shelter, and the businesses are conducted in a hidden way. Advertisements and related links of paper selling and buying are reachable pervasively over the Internet (as shown in **Figure 8**, with dotted line). To date, China has neither formed a strict article responsibility check system nor popularized anti-plagiarism detection software and application. The GAPPC although has taken a series of measures to deal with the problems, e.g., it organizes stringent activities to check journals every year, published an updating list of the illegal academic journals catalogues and punished a group of lawbreakers. These measures fail to put an end or even a temporary stop of the business. Therefore, we urge major search engines and websites practically undertake their social responsibilities to enhance the capabilities of distinguishing between original papers and plagiarized ones for all journal publishers, crack down or ban paper selling and buying advertisement, and construct complete and efficient Chinese IAJ platforms.

3.3. IAJ XML Standard Problems

The standardization degree in current digital publishing industry in China is not high enough. Compared with those in developed countries, domestic digital publishing industry is far behind in terms of process establishment and implementation standards. China lacks an overall standardized protocol and even fails to establish the most basic standard system architecture, though we have introduced a variety of international standards for printing. Moreover, despite the improvement in establishment process for publication format, information-oriented publication and publication logistics, there still is some inefficiency. For example, for the publication format, comparatively systematic standards have been established only for printed books and journals, but not for audio, video products and electronic publications. Not to mention the online digital publications, the standards for the format are still to be determined and under hot debate. According to the requirements of press and publishing industry in the "11th China National Development Five-Year Plan", the research on press and publishing standard system has been listed as a key task of publishing science research. China Institute of Publishing Science has undertaken the task under the approval of the MSTC and has completed prove of concept by 2008. Besides, researches for Network Publishing Standards, Classified Standards for Publication Marketing and Normative Standards for Publication Logistics Procedure have already been initiated and will be finished in two to three years [10].

The indexing resources for full-text VIP, CNKI and WFD index are text documents such as Microsoft word documents received from CAJEBs or text documents

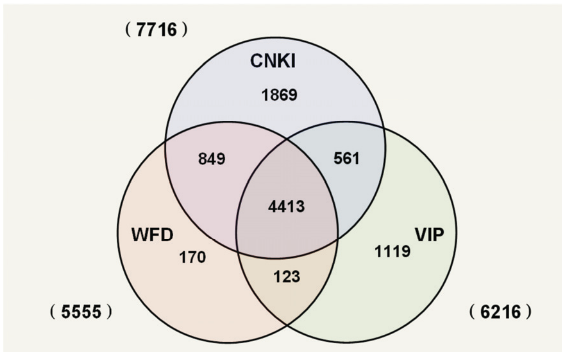

Figure 5. Repetitions of resource among CNKI, VIP and WFD.

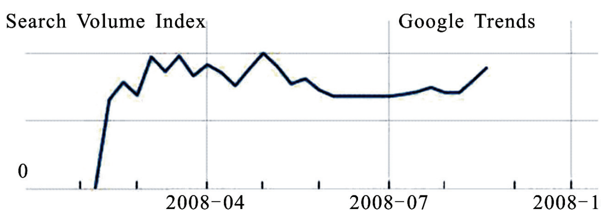

Figure 6. Search trend map of "ghost written" Chinese journal papers.

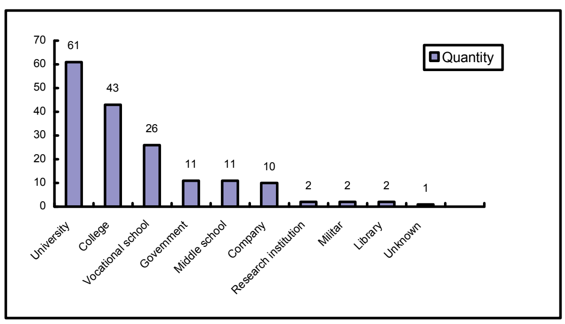

Figure 7. Statistical data on distribution of original authors.

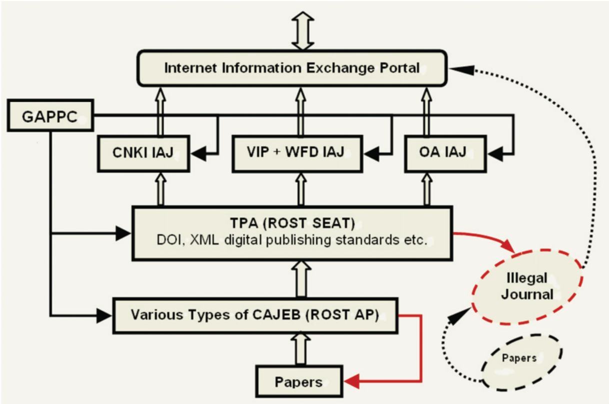

Figure 8. IAJ science information communication mode based on TPA.

obtained after scanning and OCR processing. The documents for displaying and download are in PDF format (with only CNKI has a unique CAJ format). When producing bibliographic information such as full-text indexes, authors, titles, affiliations, references, etc., no matter the process is conducted manually or assisted by software with manual inspection, the IAJs all choose to use different internal formats and never share with each other. Full-text reading is only limited to personal computers with PDF compatible software like Adobe Reader. Even the basic bibliographic information is only available to Internet Explorer (IE) based browsers, but does not support other browsers such as Firefox, Google Chrome or mobile Linux based browsers.

The three IAJs have experienced transition and expansion periods in aspects of scale & category collection, technical support and sales income, and have stepped into a steady development stage. How to make use of the existing successful experience in China and foreign countries to further improve and expend Chinese IAJ construction based on Extensible Markup Language (XML) along with Hypertext Markup Language (HTML) as the core, is a common problem faced by all three IAJs.

3.4. IAJ Management and Relevant Problems

IAJs mainly obtain their information resources from CAJEBs which have paper copyrights with control. Economic benefits are distributed only between IAJs and CAJEBs, and there is no systematic restriction or supervision mechanism to guarantee authors' financial benefits. Another problem closely related to copyright problem is that the financial benefits produced by the same paper that is published by several IAJs and they cannot be effectively returned to the original author and CAJEB.

According to relevant national policy, after peer review and formal collection by CAJEB before publishing, CAJEBs often charge a certain amount of money from authors in the name of page fee, publishing fee or manuscript review fee (generally called publishing fee below). IAJs completely adopt a market-oriented operation and charge libraries and information institutions in universities, research institutes, enterprises, public institutions and individuals in variety of ways including Internet database packages, mirror site purchase, network flow and etc. After several years of operation, these IAJs show that each one operates in a reasonable status and gets a certain amount of profit annually which the profit keeps increasing. Although IAJs such as CNKI pay copyright royalties to CAJEBs like copyright usage fees for collected articles. As a matter of fact, CAJEBs seldom provide any payment to authors [11].

As a responsible or correspondent author of a paper, he/she has to pay a certain amount of money not only at the time to publishing, but also at time when reading. The subscription fee is although usually paid by the libraries or organization they work for. CAJEBs collect both of publishing fees from authors and copyright usage fees from IAJs. Every IAJ makes full use authors' academic efforts, while CAJEBs gain good financial profits, create excellent social benefit, and boost science information communication. However, the profit distribution among authors, CAJEBs and IAJs is obviously unfair and improper, especially from author perspective.

4. IAJ Science Information Communication Mode Based on TPA

4.1. Improvement of IAJ Communication Mode

From the view of economic scale, CNKI now gets the best enterprise benefit among the three IAJs. In 2005, its sales revenue was RMB 140 million Yuan (21.5 million US dollar) with a net profit of RMB 26 million Yuan (4 million US dollar), in which the overseas sales amount was more than RMB 32 million Yuan (4.9 million US dollar). Its fixed assets valued RMB 190 million Yuan (29.2 million US dollar) after years of accumulation. WFD ranked second and its sales revenue in 2005 was nearly RMB 200 million Yuan (30.7 million US dollar) with a slight worse net profit when compared to CNKI. Since 2003, VIP has had a steady average annual sales nearly RMB 30 million Yuan (4.6 million US dollar) with around RMB 1 million Yuan (0.15 million US dollar) net profit. In order to maximize the enterprise benefit, the principle of "survive the fittest" is inevitable. Under the competition mechanism of market oriented economy, enterprises merge, combination or acquisition are normal economy activities. If WFD and VIP, both of whose net profit are relatively small and are all managed by MSTC, can be merged to a large-scale VIP + WFD IAJ enterprise, the new enterprise may achieve the economic benefit by size-effect which is unavailable in current state. The new combined enterprise would gather resources and techniques to compete with CNKI, and then it could also provide better services for the science information community while creating a favorable condition for its own development.

Chinese open access (OA) IAJs have had some achievements after years of construction. Science Paper Online is a good example which has included 65 kinds of OA journals. Among Chinese OA IAJs, 4 of them have already indexed by Directory of Open Access Journal (DOAJ). However, the current scale cannot meet the need of internet based science information communications. Because domestic OA IAJs are not popular and low in influence, scientific researchers tend to submit to CAJEBs instead of OA IAJs although the publication cycles are relatively long. For example the IAJs like CNKI aimed to provide a fast track service usual cannot make a paper available online up to 2-4 months after the printed paper version. In this regard, it could be wise a wise model that third-party agencies finance independent OA IAJs and cooperate with CAJEBs. By learning from the operation experience of OA journals from foreign successful publishers, e.g., open Choice. Depending on authors' objectives and financial status, after approved peer-review by a CAJEB, OA IAJ fast publishing could be realized for scientific research papers. However, the authors have to pay a relatively higher fee to OA IAJs and CAJEBs in this condition. Certainly, existing IAJs including CNKI could provide real-time fast publishing while offering other integrated services. We suggest using an improved IAJ communication mode as shown in **Figure 8**.

In this improved science information communication mode based on IAJ, the TPA is an independent, fair and non-profit organization which, under the leadership of GAPPC. The GAPPC drafts policies and instructions, conducts coordination, produces publishing contents, XML standards, copyright management and profit distribution, and supervises illegal journals. The TPA then organizes coordinates, supervises and directs the science information and it also play a core role to improve existing science information communication mode. The major responsibilities include:

- To properly distribute journal resources among IAJs and avoid repeated collection of CAJEB resources.
- To establish relevant XML standard for IAJ for paper submission, text editing and IAJs' publishing.
- To accelerate anti-plagiarism software & application development and restrict the further spread of paper dealing activates as well as illegal journals.

Although from the very beginning the three IAJs were supported by departments and commissions of the, they still substantially are non-state-owned IT enterprises taking sole responsibility for profits or losses. Their main commercial products are journals directed by GAPPC. In order to avoid repeated journal resources collection, one of the main tasks of a third-party agency is to coordinate and allocate journal resources according to different IAJs' expertise. For journals containing interdisciplinary subjects, mathematics and computer science which may serve as tools for other subjects, university journals and comprehensive journal resources, certain degree of reparation can be helpful. For example, the subscriber can find direct academic information and relevant information within one IAJ with an improved service quality like connection speed, cost-effectiveness and user experience. In 2007, in order to facilitate scientific researchers and IJ publishers' demand for a better XML capability, Microsoft provided an NLM DTD compatible add-in plug-in [12] for Word so the users can produce XML compatible documents when using normal Word documents. The implementation makes Word documents consistent with XML standard at any stage which is convenient for publishing, storing and knowledge transfer. For the issue that there are so many CAJEBs in China while the three IAJs have distinctive pushing methods, third-party agencies should agree to establish a common XML standard for software specifications, submission implementation, format conversion and IAJ historical data resources collection. The standard enables flexible editing, paper transfer and data share and also lays the foundation IAJ mobile computing applications.

The paper "buy and sell" is an indirect outcome of incomplete science information communication platform [13], which also reflects that China should make more effort on academic ethics system constructions. The lAJ management mode based on thirty-party agency should be improved, while current anti-plagiarism system (ROST AP) [14] can be applied and further developed to enhance the capabilities for all kinds of journals. The GAPPC has taken a series of measures to deal with the issue. For example, it organizes dedicated professionals checking journals every year and has listed online illegal academic journals catalogues [15] to alert a group of lawbreakers & public. However, these measures fail to put an end to paper dealing. We suggest third-party agency, together with academic institute, research related websites and search engines form an "Ally for Proper Academia Network", and jointly undertake the social responsibility for developing academic journals (ROST SEAT) [16] and cracking down the business. Law policies may be produced by organizations such as Chinese Law on Academic Norms and Global Academic Ethics to further regulate illegal journals.

4.2. IAJ Copyright Management Mode Based on TPA

The copyrights of various resources collected by IAJs are usually owned by domestic CAJEBs. IAJs must obtain rights from owners such as copyrights so as to legally publish these printed academic papers in IAJs.

Collective copyrights management means that Copyright owners including neighboring right owners authorize collective copyright management organization to manage their rights. Such an organization supervises the work usage, negotiates the conditions of usage with potential users, produces usage licenses, collects subscription fees and distributes the fees among copyright owners [17]. The academic paper copyright management can be regarded as one kind of CAJEBs based collective copyrights management. It facilitates flexible copyright management and realizes legal Internet variety publishing e.g. the same paper can be published by several publishers. It also increases science information communication channels, intensifies competition for service quality and embodies copyrights collective management advantages.

The core idea of the third-party copyright management mode is that authors possess copyrights but entrust third-party copyright management organization to manage their paper ownerships and related issues. The organization gives an exclusive publishing license to CAJEB: during a certain period, other publishers are not allowed to publish the papers in any form (printed or online edition, etc.), and the length of period can be changed according to factors such as publication type. The third-party copyright management organization is the only legal copyrights owner of all kinds of papers (printed or online edition, etc.) in terms of publishing, who takes charge of copyright management (including licenses, conditions and supervision), adjusts paper publishing priorities at CAJEBs and IAJs, and assists to establish the beneficiary payment mode of the TPA. Each IAJ must contact third-party copyright management organization to obtain a license for Internet publishing.

4.3. Third Party-Based Beneficiary Pay Mode in IAJ

The essence of beneficiary pay mode is that, beneficiaries jointly pay the publishing fees for scientific research efforts and rewards contributed authors. Based on beneficiary pay mode [18], a third party-based beneficiary pay mode has been moved forward. The following aspects are applied to boost science information communication system on both CAJEBs and IAJs:

- Determination of beneficiaries and beneficial degree: Authors, subscribers and IAJs are beneficiaries of published papers. For the subscribers' perspective, they may get information needed from IAJ websites directly or obtain the information from IAJ citation analysis platforms. Published papers from authors are the core resource and foundation for IAJ operations which attract users therefore IAJ and authors are also beneficiaries.
- Payment subjects and objects: The subjects of payment are beneficiaries and objects of payment are authors and CAJEBs in the science information domain. Printed journals determine the submitted papers academic values and claim publishing rights for authors' scientific research results. Authors first pay publication fees to CAJEB. When subscribers publish their own scientific research papers with references, the appreciation is expressed by paying the authors, who are cited in references a certain amount of money. It is seen as a tribute to others' scientific research efforts and is a necessity for establishing a people-oriented scientific research environment. The usage of scientific research results by enterprise customers is seldom embodied in published papers and it is hard to determine such kind of usage from references. IAJs should adopt a different charging tariff for those subscribers, properly raise the use price and promote science information communication mechanisms from which they benefit more. Although IAJs do not obtain copyrights directly from authors, the fact is that authors provide the core contents. IAJs can trust CAJEBs or use other methods to manage copyrights, but it is better to bypass CAJEBs and pay authors directly to reduce administration cost.
- Subscription fees distribution. Authors should pay publication fees to journal editorial boards and IAJs. The fees include two parts: manuscript review fee which should not be over-charged as many potential authors may withdraw; the page processing fee which is paid after peer-review, paper revision and acceptance notification. A public credible independent third-party agency is needed to collect basic IAJ usage fees from universities and colleges, research institutes, enterprises and government based on usage date fetched from mirror sites, Inter- net database packing and data flow. It also collects a certain amount of use fees from authors or subscribers who cite others' literatures according. The payment of operations, maintenance, new technology development and some commercial profit to IAJs are based on publication contents, papers download and references count.

The beneficiary pay mode introduction based on TPA requires a longer term follow up and should experience academic value verification. As for all contributors, there should be a fair benefit distribution mechanism to encourage authors and at the same time to guarantee the financial interests and brand effects for publishing institutions including IAJs and CAJEBs so that a healthy operation can be maintained. We suggest that a fair, credi-

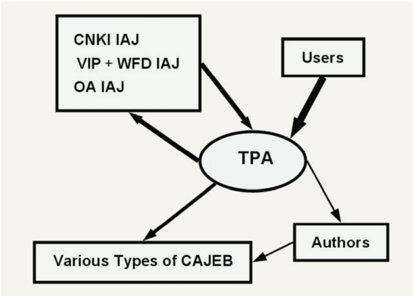

Figure 9. Beneficiary payment mode of TPA.

ble and independent TPA accelerate this progress to boost internet based science information communications. The TPA also prevents authors becoming the vulnerable group in the chain. Please refer to **Figure 9** for the beneficiary payment mode.

5. Conclusions

Building Chinese Internet academic journals for science information communications based on the TPA is a systematic engineering work, for which the advance in science and technology must be accelerated through system innovation. To deepen the technological system reform and all supporting system reforms, a vigorous system model that facilitates the innovation progress should be adopted increase productivity, embody Internet journals with Chinese characteristics, and conform to the technological development rule.

By learning the achievements of various Chinese IAJs and Chinese OA journal practice, several issues and operation models for digital publishing has been discussed. The benefit distribution and copyright management mode based on the TPA is proposed and studied. The future IAJs operation mode is also discussed from the view of innovative scientific research service and platform construction obstacles.

6. References

[1] L. F. Xu, "Digital Information Communication," Wuhan University Press, Wuhan, 2008.

[2] J. Willinsk, "Proposing Knowledge Exchange Model for Scholarly Publishing," *Current Issues in Education*, Vol. 3, No. 6, 2009. http://cie.asu.edu/ volume3/ number6/

[3] X. X. Shen, Z. Zheng and C. Shen, "A Review of the Major Projects Constituting the China Academic Digital Library," *The Electronic Library*, Vol. 26, No. 1, 2008, pp. 39-51.

[4] CERNET Network Center, "CERNET Brief Introduction," 2009. http://www.edu.cn/ cernet_jian_jie_1327/20060323/t20060323_91159.shtml

[5] STNET Network Center, "About CSTNET," 2009. http://www.cstnet.net.cn/ about.jsp?Type= zxgk

[6] J. K. Wang, "Web Publishing Resources out of a Possible Integration of the Road," *China Culture Daily*, 2006.

[7] CNKI, "CNKI Topology for Database Exchange Service Centre," 2009. http://www.dl.cnki.ne/tgycnki/daobao/cnkidaobao4/daobao2-5.htm

[8] CNKI, "About CNKI," 2009. http://www.global.cnki.net/Grid20/Aboutus/Aboutus.htm

[9] S. Tokizane, "Electronic Journal Publications in China," *Journal of Information Processing and Management*, Vol. 50, No. 1, 2007, pp. 2-10.

[10] Z. Hao, *et al*., "2006-2007 Interpretation for Annual Report on Publishing Industry in China," China Books Press, Beijing, 2007.

[11] S. Z. Chen, "On Copyright Protection of Journal," *Journal of Zhejiang Police College*, Vol. 8, No. 2, 2006, pp. 106-108.

[12] Microsoft, "Article Authoring Add-in for Microsoft Office Word," 2009. http://www.microsft.com/downloads/details.aspx?FamilyId=09C55527-0759-4D6D-AE02-51E90131997E&displaylang=en

[13] G. J. Shi, "On the Modern Science and the Exchange of Structure and Function of Innovation," *Science and Technology Progress and Policy*, Vol. 22, No. 1, 2005, pp. 14-18.

[14] Y. Shen, *et al*., "Research of Anti-Plagiarism Monitoring System Model," *Wuhan University Journal of Natural Sciences*, Vol. 12, No. 5, 2007, pp. 1202-1209.

[15] GAPPC, "Illegal Journals on the List of Notification," 2009. http://218.7.112.124/Article_Show.asp?ArticleID=627

[16] Y. Shen, *et al*., "Introducing a Metadata Earch Engine-ROST SEAT," 2009. http://hi.baidu.com/whusoft/blog/item/174090168e3b771c972b432e.html

[17] C. J. Wang, "Intellectual Property Right of Electronic Periodical in China," *Shanxi Library Journal*, Vol. 23, No. 1, 2007, pp. 3-6.

[18] T. Ohta and K. Hayashi, "Innovation-Oriented Publication System," *Journal of Information Processing and Management*, 2005, Vol. 48, No. 11, pp. 717-722.

Intelligent Information Management, 2010, 2, 447-456
doi:10.4236/iim.2010.28055 Published Online August 2010 (http://www.SciRP.org/journal/iim)

Semi-Markovian Model of Monotonous System Maintenance with Regard to Operating Time to Failure of Each Element

Yuriy E. Obzherin, Aleksey I. Peschansky
Sevastopol National Technical University, Sevastopol, Ukraine
E-mail: *vmsevntu@mail.ru*
Received May 12, 2010; *revised June* 18, 2010; *accepted July* 21, 2010

Abstract

An explicit form of reliability and economical stationary performance indexes for monotonous multicomponent system with regard to its elements' maintenance has been found. The maintenance strategy investigated supposes preventive maintenance execution for elements that has attained certain operating time to failure. Herewith for the time period of elements' maintenance or restoration operable elements are not deactivated. The problems of maintenance execution frequency optimization have been solved. For the model building the theory of semi-Markovian processes with a common phase field of states is used.

Keywords: Maintenance, Semi-Markovian Process, System Stationary Characteristics, System Performance Indexes Optimization

1. Introduction

In the process of a complex engineering system operation its components' characteristics are deteriorating. One of the methods of system stationary performance indexes improvement is the preventive maintenance of its components. From a large number of monographs and scientific papers dealing with mathematical aspects of systems' maintenance the works [1-8] should be singled out. Among other subjects, the single-component system maintenance strategy known as "Depending-on-age restoration" [2] or "The rule of preventive replacement" [1,5] is investigated in them. The essence of this strategy is the following. The system is restored after its failure. If it has been operating without failures for the given time period τ, its maintenance is carried out. For this strategy the rates of operation expenses and the availability function have been found, the problem of definition of maintenance execution optimal periodicity has been solved. In the present paper this maintenance strategy is generalized for multicomponent systems with monotonous structure [2]. The examples of the systems with monotonous structure are serial, parallel, parallel-serial, serial-parallel, and bridge ones.

The goal of the present article is to define optimal values of monotonous multicomponent system's components operating time for the maintenance execution with the purpose of gaining best stationary reliability and economical characteristics of the system.

2. The Problem Definition and Mathematical Model Building

Let us consider N-component system with a monotonous structure [2]. Such systems as serial, parallel, bridge systems, "P of N" systems, the ones with the whole-segregated redundancy fall into category of the system investigated.

The failure-free operation time of system's i-element is a random value (RV) α_i with distribution function (DF) $F_i(t) = P(\alpha_i \le t),\ i = \overline{1,N}$. The failure indication of element is carried out instantly and its emergency restoration (ER) which lasts random period of time β_i with DF $G_i(t) = P(\beta_i \le t),\ i = \overline{1,N}$ begins. If system's i-element has been operating without failures for the given time period τ_i, then its maintenance, the duration of which is RV β_i^p with DF $G_i^p(t) = P\left(\beta_i^p \le t\right)$ is to be executed. It is assumed that all the RV are independent and have an absolutely continuous DF and finite assembly averages $M\alpha_i, M\beta_i, M\beta_i^p$. There is no restoration queue. Both after maintenance and after ER all the reliability characteristics of elements are completely restored. The operating time to failure level τ_i for system's i-element maintenance is determined both after maintenance and after ER. Elements' deactivation and activation in the system are carried out instantly. The income per time unit of system's good state, expenses per time unit of emergency restoration and of system's i-element maintenance are equal respectively c_i^0, c_i and c_i^p, $i=\overline{1,N}$.

The system is in up state only then when at least one of the serial structures of minimal path [2] is also in up state. The system is considered to be in down state when at least one of the parallel structures of minimal section [2] is in down state (owing to its elements' maintenance or ER). It is assumed that neither ER nor maintenance of any element results in deactivation of the operable elements that are functionally connected with the failed component and do not belong to any other up-state path.

The task is to determine the following system's performance indexes: stationary steady state availability factor $K_u(\tau_1,...,\tau_N)$, mean specific income $S(\tau_1,...,\tau_N)$ per calendar time unit, and mean specific expenses $C(\tau_1,...,\tau_N)$ per time unit of system's good state. It is also necessary to determine the values of elements' operating time τ_i. On attaining these values elements' maintenance should be executed to optimize the above-mentioned system performance indexes.

Time diagram of system operation is shown in **Figure 1**.

Let us describe the system operation by means of semi-Markovian process $\xi(t)$ with discretely continuous phase field of states [9,10]

$$E=\left\{i\overline{d}\,\overline{x}^{(i)}\overline{u},\ \ i=\overline{1,N}\right\}$$

where the components of vector $\overline{d}=(d_1,...,d_N)$ indicate

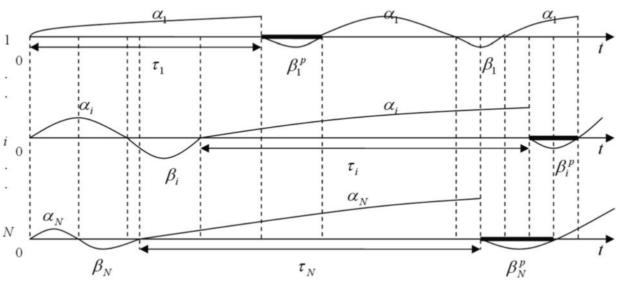

Figure 1. Time diagram of multicomponent system operation without deactivation of elements with regard to their maintenance in age.

physical state of elements: $d_k=1-k$ points out that k-element is in up state, $d_k=0$ indicates ER state of it, $d_k=2$ signifies its maintenance state; i is the number of element, which was last to change its physical state. The components of vector $\overline{x}^{(i)}$ record time period between the moment of i-element's last state change and the nearest moments of the rest of elements' state change respectively $(x_i=0)$. If $d_k=1$ then x_k is time till the next emergency failure of k-element. The components of vector $\overline{u}=(u_1,...,u_N)$ are equal to the values of respective elements' operating time from the moment of their ER or maintenance. If $d_k=2$ then it is considered that $u_k=\tau_k$. At the moment of i-element's up state restoration after its maintenance or ER its operating time is equal to zero: $u_i=0$.

Time periods of system's dwelling in its states are defined by ratios:

$$\theta_{i\overline{d}\,\overline{x}^{(i)}\overline{u}}=\gamma_i^{(d_i)}\wedge\underset{k\neq i}{\Lambda}x_k\underset{k\in\Omega_d^1}{\Lambda}\left(\tau_k-u_k\right),$$

where Λ is a sign of minimum; Ω_d^1 is a set of numbers of vector $\overline{d}$ components that are equal to 1,

$$\gamma_i^{(d_i)}=\begin{cases}\alpha_i,\ d_i=1,\\ \beta_i,\ d_i=0,\\ \beta_i^p,\ d_i=2.\end{cases}$$

Let us describe the probabilities (probability densities) of embedded Markovian chain (EMC) $\{\xi_n,\ n\geq 0\}$ transition. It is necessary to note that i-element can change its physical state 1 into the state 0 (ER) and into the state 2 (maintenance) but the states 0 and 2 can be changed only into the state 1.

Let us indicate $z_i=\underset{k\neq i}{\Lambda}x_k\wedge\underset{k\in\Omega_d^1}{\Lambda}\left(\tau_k-u_k\right)$ and let Ω_d^0, Ω_d^2 be sets of numbers of vector $\overline{d}$ components that are equal to 0 and 2, respectively. The state $i\overline{d}\,\overline{x}^{(i)}\overline{u},\ i=\overline{1,N}$ admits the following transitions:

1) to the set of states $i\overline{d'}\ \overline{x}^{(i)}\overline{u'},\ d_i'\neq 2$ with the probability density of transition $p_{i\overline{d}\,\overline{x}^{(i)}\overline{u}}^{i\overline{d'}\,\overline{x'}^{(i)}\overline{u'}}=\psi_i^{(d_i)}(z_i-y),$ where $y<z_i$, $\psi_i^{(d_i)}(\cdot)$ is the density of probability distribution of RV $\gamma_i^{(d_i)}$,

$d_k'=d_k,\ x_k'=x_k-(z_i-y),\ k\neq i,$

$$u_k' = \begin{cases} u_k + z_i - y, & k \in \Omega_d^1, \\ u_k, & k \in \Omega_d^0, \\ \tau_k, & k \in \Omega_d^2 \end{cases}$$

$k \neq i$,

$$u_i' = \begin{cases} z_i - y, & i \in \Omega_d^1, \\ 0, & i \in \Omega_d^0 \cup \Omega_d^2; \end{cases}$$

2) to the set of states $i\overline{d'}\,\overline{x'}^{(i)}\overline{u'}$, $d_i = 1$, $d_i' = 2$, with transition probability $P_{i\overline{d}\overline{x}^{(i)}\overline{u}}^{i\overline{d'}\overline{x'}^{(i)}\overline{u'}} = \overline{F}_i(\tau_i - u_i)$, where $d_k' = d_k$, $x_k' = x_k - \tau_i$, $k \neq i$,

$$u_k' = \begin{cases} u_k + \tau_i, & k \in \Omega_d^1, \\ u_k, & k \in \Omega_d^0, \\ \tau_k, & k \in \Omega_d^2; \end{cases}$$

3) to the set of states $j\overline{d'}\,\overline{x'}^{(j)}\overline{u'}$, $j \neq i$, with the probability density of transition $p_{i\overline{d}\overline{x}^{(i)}\overline{u}}^{j\overline{d'}\overline{x'}^{(j)}\overline{u'}} = \psi_i^{(d_i)}(z_i + y)$, where

$$y > 0,\ d_k' = d_k,\ k \neq j,\ x_i' = y,\ x_k' = x_k - z_i,\ k \neq i, j,$$

$$u_j' = \begin{cases} u_j + z_i, & j \in \Omega_d^1,\ d_j' \neq 2, \\ \tau_j, & j \in \Omega_d^1,\ d_j' = 2, \\ 0, & j \in \Omega_d^0 \cup \Omega_d^2, \end{cases}$$

$$u_k' = \begin{cases} u_k + z_i, & k \in \Omega_d^1, \\ u_k, & k \in \Omega_d^0, \quad k \neq j. \\ \tau_k, & k \in \Omega_d^2, \end{cases}$$

Let us assume that the conditions of stationary distribution [9,10] existence and uniqueness for EMC $\{\xi_n, n \geq 0\}$ are fulfilled. The following theorem takes place.

Theorem. The stationary distribution $\rho(\cdot)$ of EMC $\{\xi_n, n \geq 0\}$ is defined by the following expressions: as the Equation (1).

One can check this statement validity by the direct substitution of expression (1) to the set of integral equations, determining the stationary distribution of EMC.

3. Definition of System Stationary Characteristics

Let us divide the phase field E of system states into two non-overlapping subsets E_+ and E_-; E_+ is a subset of up states, E_- is a subset of down states:

$$E_+ = \left\{ i\overline{d}\overline{x}^{(i)}\overline{u},\ \overline{d} \in D_+,\ i = \overline{1,N} \right\},$$

$$E_- = \left\{ i\overline{d}\overline{x}^{(i)}\overline{u},\ \overline{d} \in D_-,\ i = \overline{1,N} \right\}.$$

Here $D_+(D_-)-$ is a set of vectors $\overline{d}$ the components of which are equal to the codes of physical states of (down) states $E_+(E_-)$. One should note that element's maintenance is referred to the down states.

Mean stationary operating time to failure T_+, mean stationary restoration time T_-, and stationary steady state availability factor (SSAF) K_u will be defined with the help of formulas [9,10]

$$T_+ = \frac{\int_{E_+} m(z)\rho(dz)}{\int_{E_+} \rho(dz)P(z, E_-)},$$

$$T_- = \frac{\int_{E_-} m(z)\rho(dz)}{\int_{E_-} \rho(dz)P(z, E_+)}, \tag{2}$$

$$K_u = \frac{T_+}{T_+ + T_-}$$

$$\rho\left(i\overline{d}\overline{x}^{(i)}\overline{u}\right) = \begin{cases} \rho \prod\limits_{k \in \Omega_d^0} f_k(u_k)\overline{G}_k(x_k) \prod\limits_{k \in \Omega_d^1} f_k(u_k + x_k) \prod\limits_{k \in \Omega_d^2} \overline{G}_k^p(x_k)\overline{F_k}(\tau_k), & i \notin \Omega_d^1,\ x_i = 0, \\ \rho \prod\limits_{k \in \Omega_d^0} f_k(u_k)\overline{G}_k(x_k) \prod\limits_{\substack{k \in \Omega_d^1 \\ k \neq i}} f_k(u_k + x_k) \prod\limits_{k \in \Omega_d^2} \overline{G}_k^p(x_k)\overline{F_k}(\tau_k), & i \in \Omega_d^1,\ i = \overline{1,N}, \end{cases} \tag{1}$$

$$\rho = \frac{1}{2}\left[\sum_{i=1}^{N} \prod_{\substack{k=1 \\ k \neq i}}^{N} \left(\int_0^{\tau_k} \overline{F_k}(u_k)du_k + M\beta_k^p \overline{F_k}(\tau_k) + M\beta_k F_k(\tau_k) \right)\right]^{-1}$$

where $\rho(\cdot)$ is the stationary distribution of EMC $\{\xi_n, n \ge 0\}$, $m(z)$ are mean time periods of system's dwelling in its states, $P(z, E_+)$ $\left(P(z, E_-)\right)$ are probabilities of EMC $\{\xi_n, n \ge 0\}$ transition from down (up) states to up (down) states.

With regard to the stationary distribution of EMC (1) the Formula (2) is transformed into:

$$T_+(\tau_1,...,\tau_N) = \frac{\sum\limits_{d\in D_+}\prod\limits_{k=1}^{N} T_k^{(d_k)}(\tau_k)}{\sum\limits_{d\in D'_+}\sum\limits_{j\in G(d)}\prod\limits_{\substack{k=1\\k\ne j}}^{N} T_k^{(d_k)}(\tau_k)}, \tag{3}$$

$$T_-(\tau_1,...,\tau_N) = \frac{\sum\limits_{d\in D_-}\prod\limits_{k=1}^{N} T_k^{(d_k)}(\tau_k)}{\sum\limits_{d\in D'_-}\left[\sum\limits_{j\in I_0(d)} F_j(\tau_j)\prod\limits_{\substack{k=1\\k\ne j}}^{N} T_k^{(d_k)}(\tau_k) + \sum\limits_{j\in I_2(d)} \overline{F}_j(\tau_j)\prod\limits_{\substack{k=1\\k\ne j}}^{N} T_k^{(d_k)}(\tau_k)\right]}, \tag{4}$$

$$K_u(\tau_1,...,\tau_N) = \frac{\sum\limits_{d\in D_+}\prod\limits_{k=1}^{N} T_k^{(d_k)}(\tau_k)}{\sum\limits_{d\in D}\prod\limits_{k=1}^{N} T_k^{(d_k)}(\tau_k)} = \frac{\sum\limits_{d\in D_+}\prod\limits_{k=1}^{N} T_k^{(d_k)}(\tau_k)}{\prod\limits_{k=1}^{N}\left(T_k^{(0)}(\tau_k)+T_k^{(1)}(\tau_k)+T_k^{(2)}(\tau_k)\right)}, \tag{5}$$

$$T_k^{(1)}(\tau_k) = \int_0^{\tau_k} \overline{F}_k(t)dt,$$
$$T_k^{(0)}(\tau_k) = F_k(\tau_k) M\beta_k,$$
$$T_k^{(0)}(\tau_k) = \overline{F}_k(\tau_k) M\beta_k^p.$$

Here D'_+ is a set of system's borderline physical up states, that is a set of such vectors $\overline{d} \in D_+$ that any component change from 1 to 0 or to 2 transfers vector $\overline{d}$ to the set D_-; $G(d)$ is a set of such numbers of vector $\overline{d} \in D'_+$ components that any component change from 1 to 0 or to 2 transforms vector $\overline{d}$ to the set D_-; D'_- is a set of system's borderline down states, that is a set of such vectors $\overline{d} \in D_-$ that any component change from 0 or 2 to 1 transforms vector $\overline{d}$ to the set D_+; $I_0(d)\left(I_2(d)\right)$ is a set of such numbers of vector $\overline{d} \in D'_-$ components that any component change from 0 (2) to 1 transforms vector $\overline{d}$ to the set D_+.

Remark. If passive strategy of elements' maintenance is executed, *i.e.*, elements' maintenance is not carried out, then the Formulas (3)-(5) coincide with respective stationary characteristics of restorable systems [9,10]. It becomes clear when taking into account that:

$$\lim_{\tau_k\to\infty} T_k^{(d_k)}(\tau_k) = \begin{cases} M\beta_k, & d_k = 0, \\ M\alpha_k, & d_k = 1, \\ 0, & d_k = 2. \end{cases}$$

Let us define system stationary characteristics $T_+(\tau_1, ..., \tau_N)$, $T_-(\tau_1,...,\tau_N)$, $K_u(\tau_1,...,\tau_N)$ using SSAF $K_i(\tau_i)$ of elements that are defined by Formulas [1,2]:

$$K_i(\tau_i) = \frac{\int_0^{\tau_k} \overline{F}_i(s)ds}{\int_0^{\tau_k} \overline{F}_i(s)ds + M\beta_i^p \overline{F}_i(\tau_i) + M\beta_i F_i(\tau_i)}, \ i = \overline{1,N}.$$

Let $M_1,...,M_\omega$ be all the different sets of elements of system paths [2]. One should note that according to the definition the elements not belonging to the set of elements of path is in down state, *i.e.*, are in a state 0 or 2. M'_i, $i = \overline{1,\omega'}$ are the sets of borderline paths elements; $G(M'_i)$, $i = \overline{1,\omega'}$ is a set of borderline path M'_i elements that correspond to the numbers of elements, the transition of which from up to down state, leads to the whole system failure. The sets Φ_i, $i = \overline{1,s}$ are the ones of section elements; Φ'_i, $i = \overline{1,s'}$ are the sets of borderline section elements; $I(\Phi'_i)$, $i = \overline{1,s'}$ is a set of borderline section Φ'_i elements that correspond to the numbers of the elements, the transition of which from down to up state leads to the whole system restoration.

Formulas (3)-(5) transformation of averages products' sums and some other elementary transformations lead to the following result:

$$T_+(\tau_1,...,\tau_N) = \frac{\sum\limits_{i=1}^{\omega}\prod\limits_{n\in M_i} K_n(\tau_n)\prod\limits_{n\notin M_i}^{N}\left(1-K_n(\tau_n)\right)}{\sum\limits_{i=1}^{\omega'}\sum\limits_{j\in G(M'_i)}\dfrac{\prod\limits_{n\in M'_i} K_n(\tau_n)\prod\limits_{n\notin M'_i}^{N}\left(1-K_n(\tau_n)\right)}{T_j^{(1)}(\tau_j)}}, \tag{6}$$

$$T_-(\tau_1,...,\tau_N) = \frac{\sum\limits_{i=1}^{s}\prod\limits_{n\notin \Phi_i}^{N} K_n(\tau_n)\prod\limits_{n\in \Phi_i}\left(1-K_n(\tau_n)\right)}{\sum\limits_{i=1}^{s'}\sum\limits_{j\in I(\Phi'_i)}\dfrac{\prod\limits_{n\notin \Phi'_i}^{N} K_n(\tau_n)\prod\limits_{n\in \Phi'_i}\left(1-K_n(\tau_n)\right)}{T_j^{(0)}(\tau_j)+T_j^{(2)}(\tau_j)}}, \tag{7}$$

$$K_u(\tau_1,...,\tau_N)=\sum_{i=1}^{\omega}\prod_{n\in M_i}K_n(\tau_n)\prod_{n\notin M_i}^{N}\left(1-K_n(\tau_n)\right)$$
$$=\phi\left(K_1(\tau_1),...,K_N(\tau_N)\right). \quad (8)$$

Here the structural function of the system $\phi(z_1,...,z_n)$ is given in a disjunctive normal form but it can be introduced in many different equivalent forms, for example, in a linear one [2,11].

To define mean specific income $S(\tau_1,...,\tau_N)$ per calendar time unit and mean specific expenses $C(\tau_1,...,\tau_N)$ per time unit of system's good state, the formulas [12] will be used

$$S=\frac{\int_E m(z)f_s(z)\rho(dz)}{\int_E m(z)\rho(dz)},\ C=\frac{\int_E m(z)f_c(z)\rho(dz)}{\int_{E_+} m(z)\rho(dz)} \quad (9)$$

where $f_s(z)$, $f_c(z)$ are the functions defining income and expenses respectively in every state.

With regard to indications introduced in the model building part of the article the functions $f_s(z)$ and $f_c(z)$ gain the bottom form:

After some transformations the Formula (9) will be as following:

$$S(\tau_1,...,\tau_N)=\sum_{i=1}^{N}S_i\left(\tau_i\right),$$

$$C\left(\tau_1,...,\tau_N\right)=\sum_{i=1}^{N}\frac{C_i(\tau_i)K_i(\tau_i)}{K_u\left(\tau_1,...,\tau_N\right)}, \quad (10)$$

where $S_i\left(\tau_i\right)=\frac{c_i^0T_i^{(1)}(\tau_i)-c_iT_i^{(0)}(\tau_i)-c_i^pT_i^{(2)}(\tau_i)}{T_i^{(1)}(\tau_i)+T_i^{(0)}(\tau_i)+T_i^{(2)}(\tau_i)}$ is mean specific income of i-element per calendar time unit and $C_i\left(\tau_i\right)=\frac{c_iT_i^{(0)}(\tau_i)+c_i^pT_i^{(2)}(\tau_i)}{T_i^{(1)}(\tau_i)}$ are mean specific expenses per time unit of i-element's good state. One should note that in [2] the value of mean specific expenses $C_i\left(\tau_i\right)$ is called "operating expenses rate" and is presented in the following form:

$$C_i\left(\tau_i\right)=c_i^pM\beta_i^pI_p(\tau_i)+c_iM\beta_iI_h(\tau_i),$$

where

$$I_h(\tau_i)=\frac{1}{M_h(\tau_i)}=\frac{F_i(\tau_i)}{\int_0^{\tau_k}\overline{F}_i(s)ds},$$

$I_p(\tau_i)=\frac{1}{M_p(\tau_i)}=\frac{\overline{F}_i(\tau_i)}{\int_0^{\tau_k}\overline{F}_i(s)ds}$ are respectively an average number of ER and maintenance per time unit; $M_h(\tau_i)$, $M_p(\tau_i)$ are assembly averages of time period between ER and maintenance of i-element.

Hereafter let us write down stationary characteristics of multicomponent systems with concrete structures and with regard to their elements' maintenance in age.

Stationary characteristics of serial system. The block scheme of N-component serial system is shown in **Figure 2**.

Let us use the ratios (6)-(8). The system has a single borderline path: $M_1=M_1'=\{1,2,...,N\}$. Emergency failure of any element or its maintenance leads to the whole system failure, that is why $G(M_1')=\{1,2,...,N\}$. The structural function of the system $\phi(z_1,...,z_N)$ is the following one: $\phi(z_1,...,z_N)=\prod_{k=1}^{N}z_k$. Thus, mean stationary

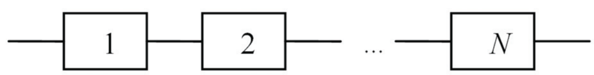

Figure 2. Block scheme of serial system.

$$f_s\left(z\right)=\begin{cases}-\sum_{k\in\Omega_d^0}c_k-\sum_{k\in\Omega_d^2}c_k^p,\ z\in\left\{i\overline{d}\overline{x}^{(i)}\overline{u}\right\},\ i=\overline{1,N},\ \text{if}\ \ \Omega_d^1=\varnothing,\\ \sum_{k\in\Omega_d^1}c_k^0-\sum_{k\in\Omega_d^0}c_k-\sum_{k\in\Omega_d^2}c_k^p,\ z\in\left\{i\overline{d}\overline{x}^{(i)}\overline{u}\right\},\ i=\overline{1,N},\ \text{if}\ \ \Omega_d^1\neq\varnothing,\end{cases}$$

$$f_c\left(z\right)=\begin{cases}\sum_{k\in\Omega_d^0}c_k+\sum_{k\in\Omega_d^2}c_k^p,\quad z\in\left\{i\overline{d}\overline{x}^{(i)}\overline{u}\right\},\ i=\overline{1,N},\quad \text{if}\ \ \Omega_d^0\bigcup\Omega_d^2\neq\varnothing,\\ 0,\quad z\in\left\{i\overline{d}\overline{x}^{(i)}\overline{u}\right\},\ i=\overline{1,N},\quad \text{if}\ \ \Omega_d^0\bigcup\Omega_d^2=\varnothing.\end{cases}$$

operating time to failure $T_+(\tau_1,...,\tau_n)$, mean stationary restoration time $T_-(\tau_1,...,\tau_n)$ and stationary steady state availability factor of the system $K_u(\tau_1,...,\tau_n)$ are defined by the following expressions:

$$T_+(\tau_1,...,\tau_N)=\left[\sum_{i=1}^{N}\frac{1}{T_i^{(1)}(\tau_i)}\right]^{-1}=\left[\sum_{i=1}^{N}\left[\int_0^{\tau_i}\overline{F}_i(t)dt\right]^{-1}\right]^{-1},$$

$$T_-(\tau_1,...,\tau_N)=\frac{\left[\prod_{i=1}^{N}K_i(\tau_i)\right]^{-1}-1}{\sum_{i=1}^{N}\frac{1}{T_i^{(1)}(\tau_i)}}$$

$$=\frac{\prod_{i=1}^{N}\frac{\int_0^{\tau_k}\overline{F}_i(s)ds+M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)}{\int_0^{\tau_k}\overline{F}_i(s)ds}-1}{\sum_{i=1}^{N}\left[\int_0^{\tau_k}\overline{F}_i(s)ds\right]^{-1}},$$

$$K_u(\tau_1,...,\tau_N)=\prod_{i=1}^{N}K_i(\tau_i)$$

$$=\prod_{i=1}^{N}\frac{\int_0^{\tau_k}\overline{F}_i(s)ds}{\int_0^{\tau_k}\overline{F}_i(s)ds+M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)}.$$

The system mean specific income per calendar time unit and mean specific expenses per time unit of system's good state are estimated with the help of ratios:

$$S(\tau_1,...,\tau_N)=\sum_{i=1}^{N}S_i(\tau_i),$$

$$C(\tau_1,...,\tau_N)=\sum_{i=1}^{N}\frac{C_i(\tau_i)}{\prod_{\substack{n=1\\n\neq i}}^{N}K_n(\tau_n)},$$

Stationary characteritics of parallel system. In **Figure 3** the block scheme of N-component redundant system of integer multiplicity with active reserve is shown.

The system has a single borderline section: $\Phi_1=\Phi_1'=I(\Phi_1')=\{1,...,N\}$. In this case the structural function of the system $\phi(z_1,...,z_N)$ will be $\phi(z_1,...,z_N)=1-\prod_{k=1}^{N}(1-z_k)$. According to the Formulas (6)-(8), (10) system stationary performance indexes are defined by the equations:

$$T_+(\tau_1,...,\tau_N)=\frac{\left[\prod_{n=1}^{N}\left(1-K_n(\tau_n)\right)\right]^{-1}-1}{\sum_{j=1}^{N}\frac{1}{T_j^{(0)}(\tau_j)+T_j^{(2)}(\tau_j)}}$$

$$=\frac{\prod_{i=1}^{N}\frac{\int_0^{\tau_k}\overline{F}_i(s)ds+M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)}{M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)}-1}{\sum_{i=1}^{N}\left[M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)\right]^{-1}},$$

$$T_-(\tau_1,...,\tau_N)=\frac{1}{\sum_{j=1}^{N}\frac{1}{T_j^{(0)}(\tau_j)+T_j^{(2)}(\tau_j)}}$$

$$=\left[\sum_{i=1}^{N}\left[M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)\right]^{-1}\right]^{-1},$$

$$K_u(\tau_1,...,\tau_N)=1-\prod_{n=1}^{N}\left(1-K_n(\tau_n)\right)$$

$$=1-\prod_{i=1}^{N}\frac{M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)}{\int_0^{\tau_k}\overline{F}_i(s)ds+M\beta_i^p\overline{F}_i(\tau_i)+M\beta_iF_i(\tau_i)},$$

$$S(\tau_1,...,\tau_N)=\sum_{i=1}^{N}S_i(\tau_i),$$

$$C(\tau_1,...,\tau_N)=\sum_{i=1}^{N}\frac{C_i(\tau_i)K_i(\tau_i)}{1-\prod_{n=1}^{N}\left(1-K_n(\tau_n)\right)}.$$

Stationary characteristics of parallel-serial system. The block scheme of parallel-serial system is shown in **Figure 4**.

It includes L serial chains with parallel connection. Each i-chain consists of N_i elements in series. In this instance the structural function of the system $\phi(z_{11}, ..., z_{LN_L})$ is as follows:

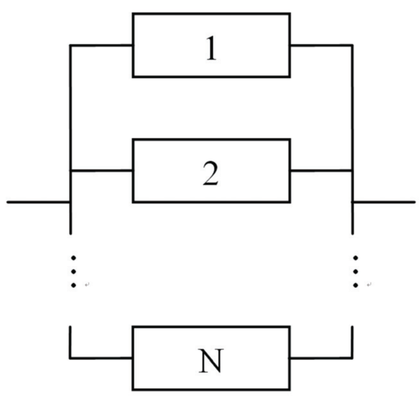

Figure 3. Block scheme of parallel system.

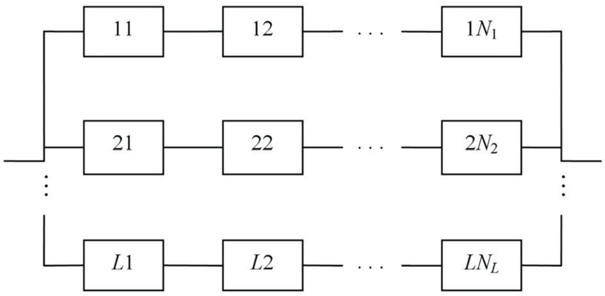

Figure 4. Block scheme of parallel-serial system.

$\phi(z_{11},...,z_{LN_L}) = 1-\prod_{i=1}^{L}\left(1-\prod_{n=1}^{N_i} z_{in}\right)$. The ratios (8), (10) for stationary characteristics estimation of this structure are transformed into the following expressions:

$$K_u(\tau_{11},...,\tau_{LN_L}) = 1-\prod_{i=1}^{L}\left(1-\prod_{n=1}^{N_i} K_{in}(\tau_{in})\right),$$

$$S(\tau_{11},...,\tau_{LN_L}) = \sum_{i=1}^{L}\sum_{n=1}^{N_i} S_{in}\left(\tau_{in}\right),$$

$$C(\tau_{11},...,\tau_{LN_L}) = \sum_{i=1}^{L}\sum_{n=1}^{N_i}\frac{C_{in}(\tau_{in})K_{in}(\tau_{in})}{1-\prod_{n=1}^{N}\left(1-K_{in}(\tau_{in})\right)},$$

where $K_{in}\left(\tau_{in}\right)$, $S_{in}\left(\tau_{in}\right)$, $C_{in}\left(\tau_{in}\right)$ are stationary SSAF, mean specific income of *i*-chain's n-element per calendar time unit and mean specific expenses per time unit of this element's good state respectively:

$$K_{in}\left(\tau_{in}\right) = \frac{T_{in}^{(1)}(\tau_{in})}{T_{in}^{(1)}(\tau_{in})+T_{in}^{(0)}(\tau_{in})+T_{in}^{(2)}(\tau_{in})},$$

$$S_{in}\left(\tau_{in}\right) = \frac{c_{in}^{0}T_{in}^{(1)}(\tau_{in})-c_{in}T_{in}^{(0)}(\tau_{in})-c_{in}^{p}T_{in}^{(2)}(\tau_{in})}{T_{in}^{(1)}(\tau_{in})+T_{in}^{(0)}(\tau_{in})+T_{in}^{(2)}(\tau_{in})},$$

$$C_{in}\left(\tau_{in}\right) = \frac{c_{in}T_{in}^{(0)}(\tau_{in})+c_{in}^{p}T_{in}^{(2)}(\tau_{in})}{T_{in}^{(1)}(\tau_{in})},$$

$$T_{in}^{(1)}(\tau_{in}) = \int_{0}^{\tau_{in}}\overline{F_{in}}(s)ds,$$

$$T_{in}^{(2)}(\tau_{in}) = M\beta_{in}^{p}\overline{F_{in}}(\tau_{in}),$$

$$T_{in}^{(0)}(\tau_{in}) = M\beta_{in}F_{in}(\tau_{in}).$$

Stationary characteristics of serial-parallel system. In **Figure 5** the block scheme of serial-parallel system is represented.

It consists of L units in series. Each *i*-unit includes N_i parallel elements. In this case the structural function of the system $\phi(z_{11},...,z_{N_LL})$ is

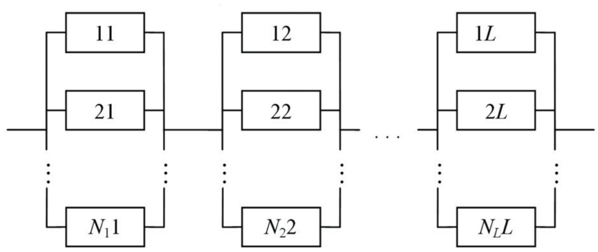

Figure 5. Block scheme of serial-parallel system.

$\phi(z_{11},...,z_{N_LL}) = \prod_{i=1}^{L}\left(1-\prod_{n=1}^{N_i}(1-z_{ni}\right)$. According to (8), (10) system stationary characteristics can be estimated in the following way:

$$K_u(\tau_{11},...,\tau_{N_LL}) = \prod_{i=1}^{L}\left(1-\prod_{n=1}^{N_i}(1-K_{ni}(\tau_{ni}))\right),$$

$$S(\tau_{11},...,\tau_{N_LL}) = \sum_{i=1}^{L}\sum_{n=1}^{N_i} S_{ni}\left(\tau_{ni}\right),$$

$$C(\tau_{11},...,\tau_{N_LL}) = \sum_{i=1}^{L}\sum_{n=1}^{N_i}\frac{C_{ni}(\tau_{ni})K_{ni}(\tau_{ni})}{\prod_{i=1}^{L}\left(1-\prod_{n=1}^{N_i}(1-K_{ni}(\tau_{ni}))\right)},$$

where $K_{ni}\left(\tau_{ni}\right)$, $S_{ni}\left(\tau_{ni}\right)$, $C_{ni}\left(\tau_{ni}\right)$ are stationary SSAF, mean specific income of *i*-unit's n-element per calendar time unit and mean specific expenses per time unit of this element's good state respectively:

$$K_{ni}\left(\tau_{ni}\right) = \frac{T_{ni}^{(1)}(\tau_{ni})}{T_{ni}^{(1)}(\tau_{ni})+T_{ni}^{(0)}(\tau_{ni})+T_{ni}^{(2)}(\tau_{ni})},$$

$$S_{ni}\left(\tau_{ni}\right) = \frac{c_{ni}^{0}T_{ni}^{(1)}(\tau_{ni})-c_{ni}T_{ni}^{(0)}(\tau_{ni})-c_{ni}^{p}T_{ni}^{(2)}(\tau_{ni})}{T_{ni}^{(1)}(\tau_{ni})+T_{ni}^{(0)}(\tau_{ni})+T_{ni}^{(2)}(\tau_{ni})},$$

$$C_{ni}\left(\tau_{ni}\right) = \frac{c_{ni}T_{ni}^{(0)}(\tau_{ni})+c_{ni}^{p}T_{ni}^{(2)}(\tau_{ni})}{T_{ni}^{(1)}(\tau_{ni})},$$

$$T_{ni}^{(1)}(\tau_{ni}) = \int_{0}^{\tau_{ni}}\overline{F_{ni}}(s)ds,$$

$$T_{ni}^{(2)}(\tau_{ni}) = M\beta_{ni}^{p}\overline{F_{ni}}(\tau_{ni}),$$

$$T_{ni}^{(0)}(\tau_{ni}) = M\beta_{ni}F_{ni}(\tau_{ni}).$$

4. Optimization of Elements Maintenance Terms

The task of defining optimal system's performance indexes is reduced to the definition of absolute extremums of the functions (8) and (10). It is necessary to note that for gaining maximum values of system SSAF K_u $(\tau_1,...,\tau_N)$ and mean specific income $S(\tau_1,...,\tau_N)$ it is obligatory and sufficient to optimize the value of each system element's operation time for its maintenance execution, which is not valid for system's minimum mean specific expenses $C(\tau_1,...,\tau_N)$.

Equaling the partial derivatives of the functions $K_u(\tau_1,...,\tau_N)$, $S(\tau_1,...,\tau_N)$ and $C(\tau_1,...,\tau_N)$ to zero we get the systems of equations defining optimal values of operating time τ_i^k, τ_i^s, τ_i^c, $i=\overline{1,N}$:

$$\lambda_i(\tau_i)\int_0^{\tau_i}\overline{F}_i(t)dt - F_i(\tau_i) = \frac{M\beta_i^p}{M\beta_i - M\beta_i^p},\quad i=\overline{1,N}, \tag{11}$$

as the Equations (12) and (13)

Here $\lambda_i(t)=\dfrac{f_i(t)}{\overline{F}_i(t)}$ is the continuous failure rate of i-element.

The sufficient condition for the systems of Equations (11), (13) finite solutions existence is respective inequalities fulfillment (it is supposed that $\lambda_i(0)=0$ for the system (13)).

$$\lambda_i(\infty)M\alpha_i > \frac{M\beta_i}{M\beta_i - M\beta_i^p},\quad i=\overline{1,N},$$

In case of systems of equations' unique solutions existence, optimal values of system performance indexes are defined by the following formulas:

$$\begin{aligned}
&K_{u\max} = \phi\left(K_1(\tau_1^k),...,K_N(\tau_N^k)\right),\\
&K_i(\tau_i^k) = \frac{\overline{F}_i(\tau_i^k)}{\overline{F}_i(\tau_i^k)+(M\beta_i - M\beta_i^p)f_i(\tau_i^k)},\\
&S_{\max} = \sum_{i=1}^N S_i(\tau_i^s),\\
&S_i(\tau_i^s) = \frac{c_i^0\overline{F}_i(\tau_i^s)-(c_iM\beta_i - c_i^pM\beta_i^p)f_i(\tau_i^s)}{\overline{F}_i(\tau_i^s)+(M\beta_i - M\beta_i^p)f_i(\tau_i^s)},\\
&C_{\min} = \frac{\sum_{i=1}^N C_i(\tau_i^c)K_i(\tau_i^c)}{K_u(\tau_1^c,...,\tau_N^c)}
\end{aligned} \tag{14}$$

$$\lambda_i(\tau_i)\left(\int_0^{\tau_i}\overline{F}_i(t)dt + \frac{(c_i - c_i^p)M\beta_iM\beta_i^p}{(c_i+c_i^0)M\beta_i-(c_i^p+c_i^0)M\beta_i^p}\right) - F_i(\tau_i) = \frac{(c_i^p+c_i^0)M\beta_i^p}{(c_i+c_i^0)M\beta_i-(c_i^p+c_i^0)M\beta_i^p},\quad i=\overline{1,N}, \tag{12}$$

$$\begin{aligned}
&\lambda_i(\tau_i)\left(\int_0^{\tau_i}\overline{F}_i(t)dt + \frac{(c_i-c_i^p)M\beta_iM\beta_i^p}{c_iM\beta_i - c_i^pM\beta_i^p}\right) - F_i(\tau_i) - \frac{\partial}{\partial K_i}\ln\phi\left(K_1(\tau_1),...,K_N(\tau_N)\right)\times\\
&\times\left[\sum_{j=1}^N C_j(\tau_j)K_j(\tau_j)\right]\cdot\left[\frac{M\beta_i^p - M\beta_i}{c_iM\beta_i - c_i^pM\beta_i^p}\left(\lambda_i(\tau_i)\int_0^{\tau_i}\overline{F}_i(t)dt - F_i(\tau_i)\right)+\frac{M\beta_i^p}{c_iM\beta_i - c_i^pM\beta_i^p}\right]=\\
&\quad = \frac{c_i^pM\beta_i^p}{c_iM\beta_i - c_i^pM\beta_i^p},\\
&\quad i=\overline{1,N}.
\end{aligned} \tag{13}$$

$$\begin{cases}
\lambda_i(\infty)\left(M\alpha_i + \dfrac{(c_i-c_i^p)M\beta_iM\beta_i^p}{(c_i+c_i^0)M\beta_i-(c_i^p+c_i^0)M\beta_i^p}\right) > \dfrac{(c_i+c_i^0)M\beta_i}{(c_i+c_i^0)M\beta_i-(c_i^p+c_i^0)M\beta_i^p},\\
\lambda_i(0)(c_i-c_i^p)M\beta_i < c_i^p + c_i^0, \qquad i=\overline{1,N},
\end{cases}$$

$$\begin{aligned}
&\left(c_iM\beta_i - c_i^pM\beta_i^p\right)\left(\lambda_i(\infty)M\alpha_i - 1\right) + \lambda_i(\infty)(c_i-c_i^p)M\beta_iM\beta_i^p -\\
&-\frac{\partial}{\partial K_i}\ln\phi\left(\frac{M\alpha_1}{M\alpha_1+M\beta_1},...,\frac{M\alpha_N}{M\alpha_N+M\beta_N}\right)\sum_{j=1}^N\frac{c_jM\beta_j}{M\alpha_j+M\beta_j}\left[\left(M\beta_i^p - M\beta_i\right)\left(\lambda_i(\infty)M\alpha_i - 1\right)+M\beta_i^p\right]>\\
&> c_i^pM\beta_i^p,\quad i=\overline{1,N}.
\end{aligned}$$

Table 1. System initial data.

№	γ_i	θ_i	$M\alpha_i$, h	$M\beta_i$, h	$M\beta_i^p$, h	c_i^0 , $c.u./h$	c_i , $c.u./h$	c_i^p , $c.u./h$
1	2	50	44.311	5	1	5	1	0.2
2	3	15	13.395	3	1	7	3	2
3	4	20	18.128	4	0.5	9	3	1

Table 2. Calculation results.

№	τ_i^k, h	$K_u^{\max}$	K_u^{∞}	τ_i^s, h	$S^{\max}$, $c.u./h$	S^{∞} , $c.u./h$	τ_i^c, h	$C^{\max}$, $c.u./h$	C^{∞} , $c.u./h$
1	25.533			23.131			15.608		
2	9.548	0.916	0.869	8.982	18.553	16.393	7.694	0.507	1.373
3	9.354			8.852			6.909		

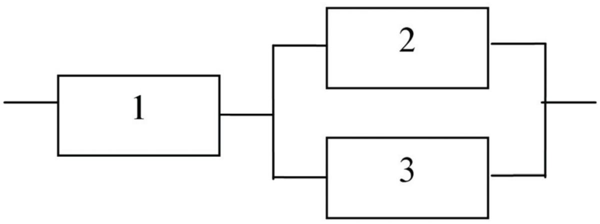

Figure 6. System block scheme for an example.

If the systems of equations have several solutions, optimal values of system performance indexes are found by substituting each one to the formula for the case of unique solution with subsequent choice of the best variant. The absence of roots of any i-equation of the systems (11), (12) denotes that the function $K_i(\tau_i)\left(S_i(\tau_i)\right)$ is a monotone one and its extremum is attained under $\tau_i \to \infty$. In this case it should be assumed that $K_i(\infty) = \frac{M\alpha_i}{M\alpha_i + M\beta_i}$, $S_i(\infty) = \frac{c_i^0 M\alpha_i - c_i M\beta_i}{M\alpha_i + M\beta_i}$ in the Formula (14).

If the system of Equation (13) doesn't have any solutions, all the possible functions that result from (10) after the substitution of

$$K_i(\infty) = \frac{M\alpha_i}{M\alpha_i + M\beta_i}, \quad C_i(\infty) = \frac{c_i M\beta_i}{M\alpha_i}$$

should be analyzed for absolute minimum. Extremum attainment under $\tau_i \to \infty$ signifies that it is not expedient to execute i-element's maintenance because it deteriorates system performance indexes.

In conclusion an example of optimal maintenance execution terms definition for three-component system with the structure represented in **Figure 6** will be given. Its elements' operating time to failure is disposed according to the law of Veibull-Gnedenko with densities:

$$f_i(t) = \frac{\gamma_i}{\theta_i} \cdot \left(\frac{t}{\theta_i}\right)^{\gamma_i - 1} e^{-\left(\frac{t}{\theta_i}\right)^{\gamma_i}}, \quad i = \overline{1,3}.$$

Initial data and calculation results are represented in the **Tables 1** and **2**.

Here K_u^{∞}, S^{∞}, C^{∞} denote system performance indexes in case if elements' maintenance is not carried out. The elements' maintenance execution increases these indexes for 5.406%, 13.178%, 63.068% respectively.

5. Conclusions

In the present paper semi-Markovian model of operation of multicomponent restorable system with monotonous structure, which takes into account its elements' maintenance execution depending on the values of operating time to failure, has been built. With the help of this model the reliability and economical stationary characteristics of the system with a general form of elements' time to failure and restoration time distributions have been defined. These characteristics are explicitly dependent on the periodicity of system of system elements' maintenance execution. This fact allows solving the problems of the above-mentioned characteristics' improvement. For a single-component system the quality indexes found coincide with the formerly known ones [1,2]. In case when elements' maintenance is not carried out ($\tau_k \to \infty$), the stationary indexes coincide with the ones found in [9,10] for restorable systems. In prospect the analogical model of maintenance of multicomponent system operating under the condition of elements' deactivation will be built by the authors.

6. References

[1] R. E. Barlow and F. Proschan, "Mathematical Theory of

Reliability," John Wiley and Sons, New York, 1965.

[2] F. Baichelt and P. Franken, "Reliability and Maintenance: A Mathematical Approach," Radio and Communication, Moscow, 1988.

[3] Y. Y. Barzilovich and V. A. Kashtanov "Some Mathematical Aspects of Complex Systems' Maintenance Theory," Soviet Radio, Moscow,1967.

[4] I. B. Gertsbakh, "Models of Preventive Maintenance," North Holland Publishing Company, Amsterdam, New York, 1977.

[5] R. E. Barlow and L. Hunter, "Optimum Preventive Maintenance Policies," *Operations Research*, Vol. 8, No. 1, 1960, pp. 90-100.

[6] C. Valdez-Flores and R. M. Feldman, "A Survey of Preventive Maintenance Models for Stochastically Deteriorating Single-Unit Systems," *Naval Research Logistics*, Vol. 36, No. 4, 1989, pp. 419-446.

[7] D. I. Cho and M. Parlar, "A Survey of Maintenance Models for Multi-Unit Systems," *European Journal of Operational Research*, Vol. 51, No. 2, 1991, pp. 1-23.

[8] R. Dekker and R. A. Wildeman, "A Review of Multi-Component Maintenance Models with Economic Dependence," *Mathematical Methods of Operations Research*, Vol. 45, No. 3, 1997, pp. 411-435.

[9] V. S. Korolyuk and A. F. Turbin, "Markovian Restoration Processes in the Problems of System Reliability," Naukova dumka, Kiev, 1982.

[10] A. N. Korlat, V. N. Kuznetsov, M. I. Novikov and A. F. Turbin, "Semi-Markovian Models of Restorable and Service Systems," Kishinev, Shtiintsa, 1991.

[11] A. M. Polovko and S. V. Gurov, "Bases of Reliability Theory," Saint Petersburg, 2006.

[12] V. M. Shurenkov, "Ergodic Markovian Processes," Nauka, Moscow, 1989.

Intelligent Information Management, **2010, 2, 457-474**
doi:10.4236/iim.2010.28056 Published Online August 2010 (http://www.SciRP.org/journal/iim)

Literature Review of Single Machine Scheduling Problem with Uniform Parallel Machines

Panneerselvam Senthilkumar[1], Sockalingam Narayanan[2]
[1]*Program Management Office, Product Development, Mahindra & Mahindra Ltd., Farm Division, Mumbai, India*
[2]*VIT University, Vellore, India*
E-mail: *psenthilpondy@gmail.com, snarayanan@vit.ac.in*
Received May 2, 2010; *revised June* 4, 2010; *accepted July* 18, 2010

Abstract

This paper presents a survey of single machine scheduling problem with uniform parallel machines. The single machine scheduling problem with uniform parallel machines consists of *n* jobs, each with single operation, which are to be scheduled on *m* parallel machines with different speeds. These parallel machines are also called proportional machines or related machines. There are several measures of performance which are to be optimized in uniform parallel machines scheduling. Since, this scheduling problem is a combinatorial problem; usage of a heuristic is inevitable to obtain solution in polynomial time. This paper gives a classification of the literatures of this scheduling problem in three major categories, viz. offline scheduling, online scheduling and miscellaneous scheduling. In total, the available literatures are classified into 17 subgroups. Under each of the first two categories, the available literatures are discussed under different groups based on different measures of performance and non-preemptive/preemptive nature of the jobs. In the last category, the literatures are discussed under three subgroups, namely non-preemptive jobs, preemptive jobs and periodic jobs.

Keywords: Uniform Parallel Machines, Measure of Performance, Heuristic, Model, Competitive Ratio

1. Introduction

In any company, production scheduling is an essential activity, which aims to prepare a schedule to produce a mix of products as per the production plan of the company. This in turn helps the company to improve its productivity.

Production scheduling can be classified into the following categories.

1) Single machine scheduling with single processor

2) Single machine scheduling problem with parallel processors (machines)

3) Flow shop scheduling

4) Job shop scheduling

5) Open shop scheduling

6) Batch scheduling

Single Machine Scheduling with Single Processor

The single machine scheduling problem with single processor (machine) consists of single machine to process n jobs. The objective of this problem is to schedule these n jobs on the single machine such that a given measure of performance is minimized. The jobs may be independent or dependent. If the set-up times of the jobs are independent of the process sequence of the jobs in the schedule, then the problem is termed as the single machine scheduling problem with independent jobs; otherwise it is termed as single machine scheduling problem with dependent jobs.

The different measures of performance of the single machine scheduling problem with independent jobs are as listed below.

- Minimizing the mean flow time
- Minimizing the maximum lateness
- Minimizing the total tardiness
- Minimizing the number of tardy jobs

In this scheduling problem, the makespan will be the same for all the sequences. Hence, it is not a part of the list of measures of performance.

Single Machine Scheduling with Parallel Machines

In the single machine scheduling problem, if the number of machines is more than one, then it is called as single machine scheduling problem with parallel machines. The parallel machines scheduling problem can be classified in to the following types.

1) Identical parallel machines scheduling problem.

2) Uniform/proportional parallel machines scheduling problem.

3) Unrelated parallel machines scheduling problem.

Let, t_{ij} be the processing times of the job j on the machine i, for i = 1, 2, 3, …., *m* and j = 1 2, 3, …, *n*.

Then the three types of parallel machines scheduling problem are defined using this processing time.

1) If $t_{ij} = t_{1j}$ for all i and j, then the problem is called as *identical parallel machines scheduling problem*.

This means that all the parallel machines are identical in terms of their speed. Each and every job will take the same amount of processing time on each of the parallel machines.

2) If $t_{ij} = t_{1j}/s_i$ for all i and j, where s_i is the speed of the machine i and t_{1j} is the processing time of the job j on the machine 1, then the problem is termed as *uniform (proportional) parallel machines scheduling problem*.

This means that the parallel machines will have different speeds. Generally, we assume s_1, s_2, s_3, …., and s_m for the parallel machines 1, 2, 3, …, and m, respectively with the relation $s_1 < s_2 < s_3 < …. < s_m$. That is the machine 1 is the slowest machine and the machine *m* is the fastest machine. For a given job, its processing times on the parallel machines will be in the ratios as listed below.

$1/s_1 : 1/s_2 : 1/s_3 : …. : 1/s_m$

3) If t_{ij} is arbitrary for all i and j, then the problem is known as *unrelated parallel machines scheduling problem*.

In this type of scheduling, there will not be any relation amongst the processing times of a job on the parallel machines. This may be due to technological differences of the machines, different features of the jobs, etc.

The measures of performance of this scheduling problem include the minimization of the makespan along with all the measures of performance as listed in Subsection 1.1. When *n* jobs with single operation are scheduled on *m* parallel machines, then each parallel machine will have its completion time of the last job in it. The maximum of such completion times on all the parallel machines is known as the makespan of the parallel machines scheduling problem, which is an important measure of performance, Panneerselvam [1].

Flow Shop Scheduling Problem

The flow shop scheduling problem consists of n jobs which require processing on m different machines. Each job has a process sequence. Further, the process sequences of all the jobs are one and same. The measures of performance of this problem are as listed below.

- Minimizing the mean flow time
- Minimizing the maximum lateness
- Minimizing the total tardiness
- Minimizing the number of tardy jobs
- Minimizing the makespan

Job Shop Scheduling

The job shop scheduling problem consists of *n* jobs which require processing on m different machines. Each job has process sequence. Further, the process sequences of the jobs are different from one another. The measures of performance of this problem are as listed under the flow shop scheduling problem.

Open Shop Scheduling

The open shop scheduling problem consists of n jobs which are to be scheduled on m different machines. There is no process sequence for each job, which means that the operations of that job can be performed in any order. The measures of performance of this problem are as listed under the flow shop scheduling problem.

Batch Scheduling

Consider the loading of a batch of jobs in a furnace for the purpose of heat treatment. The furnace will continuously function over a period of time. In between, different batches of jobs will be loaded into the furnace and taken out at different points in time. Frequent opening and closing of the door of the furnace will amount dissipation of heat from the furnace. To avoid such heat loss, one should group the jobs into different batches based on their periods of heat treatment and load them into the furnace accordingly. This will facilitate minimal disturbance (opening and closing of the door) to the furnace and thus reducing the heat loss in the furnace. Such scheduling is known as batch scheduling.

In this paper, a survey of a special case of the single machine scheduling problem which is known as *uniform parallel machines scheduling* is considered.

2. Uniform Parallel Machines Scheduling

The essential characteristics of the uniform parallel machines scheduling problem are as listed below.

- It has *n* single operation jobs.
- It has *m* parallel machines with different speeds ($s_1 < s_2 < s_3 < ….. < s_m$).
- *m* machines are continuously available and they are never kept idle while work is waiting.
- t_{1j} is the processing time of the job j on the machine 1 for j = 1, 2, 3, …, *n*.
- For each job, its processing times on the uniform parallel machines are inversely proportional to the speeds of those parallel machines ($1/s_1 : 1/s_2 : 1/s_3 : ….. : 1/s_m$), where s_1 is the unit speed.
- $t_{ij} = t_{1j}/s_i$ for j = 1, 2, 3, …., *n* and i = 2, 3, ….., *m*.

A sample data of the uniform parallel machines scheduling problem is shown in **Table 1**, in which the ratio between the machine 1 and the machine 2 is 1:2.

Table 1. Processing times of jobs on uniform parallel machines.

Machine i	Speed Ratio	Job j					
		1	2	3	4	5	6
1	1	10	8	5	12	4	6
2	2	5	4	2.5	6	2	3

The jobs may be classified into non-preemptive jobs and preemptive jobs. If the processing on a job which is assigned to a machine is continued till its completion, then such job is called non-preemptive job. If the processing of that job on a machine is discontinued before its completion and reassigned to either to the same machine or some other machine, that type of job is called preemptive job. If the processing times of the jobs do not depend on the sequence of assignment of the jobs on the machines, then the jobs are called independent jobs; otherwise, they are called as dependent jobs.

Further, if there is any dependency among the jobs which are to be scheduled on a set of uniform parallel machines, then they are called jobs with precedence constraints.

Periodic task is a recurring process/task which is characterized by two parameters, viz. execution requirement and a period. The execution time may be any non-negative number, deadline is assumed to be non-negative relational number. A periodic task $T_i = (e_i, p_i)$ with execution requirement parameter e_i and period parameter p_i generates a job at each instant $k.p_i$, which needs to execute for e_i units by a deadline of $(k+1).p_i$, for all non-negative integers k.

2.1. Basic Measures of Performance

The basic measures of performance of the single machine scheduling problem with uniform parallel machines are as listed below.

- Minimizing mean flow time
- Minimizing total tardiness
- Minimizing maximum lateness
- Minimizing number of tardy jobs
- Minimizing the maximum of the completion times of the last jobs on the uniform parallel machines (makespan)
- Maximizing minimum of the completion times of the last jobs on the uniform parallel machines

Mean flow time is the average of the completion times of the jobs. The minimization of total tardiness is an important measure, which means the minimization of the mean tardiness.

Let, n be the number of jobs with single operation

d_j be the due date of the job j

C_j be the completion time of the job j.

T_j be the tardiness of the job j.

Tardiness, $T_j = \text{Max}[0, C_j - d_j]$, if $C_j > d_j$

= 0, otherwise

$$\text{Total tardiness} = \sum_{j=1}^{n} T_j$$

Maximum lateness is defined as the maximum of the lateness values of the jobs. The lateness of a job is the difference between the completion time and the promised due date (deliver date) of a job, if the completion time is more than the due date.

The lateness of the job j be T_j or L_j.

Tardiness, $L_j = T_j = \text{Max}[0, C_j - d_j]$, if $C_j > d_j$

= 0, otherwise.

Here, the objective is to minimize the maximum lateness/tardiness (L_{max}) of the jobs.

That is, Min L_{max} = Min {Max L_j}

Minimizing the number of tardy jobs (N_T) is also another measure of performance in single machine scheduling problem. This means the minimum of the number of late jobs. A job is said to be late if it is completed beyond its due date.

The minimization of the maximum of the completion times of the last jobs on the uniform parallel machines is known as the makespan of the schedule.

The problem in which the maximization of the minimum of completion times of the last jobs on the parallel machines is known as machine covering problem. The objective of this measure is to keep all the machines alive (in working mode) during the makespan of the schedule of a given batch of jobs on the set of parallel machines.

2.2. Offline Scheduling vs. Online Scheduling

The scheduling problems are broadly classified into *offline scheduling* and *online scheduling*. In offline scheduling, the release time, processing time, due date and other necessary data of each of the jobs are known before determining the schedule of jobs on the uniform parallel machines. Online scheduling algorithms make scheduling decisions at each time-instant based upon the characteristics of the jobs that have arrived thus far with no knowledge of jobs that may arrive in the future.

The online scheduling is classified into the following types.

Scheduling jobs one by one

In this type of scheduling, the jobs which are available are ordered in some list. Then each of the jobs will be assigned to some uniform parallel machine or to some time slot one by one from the list before the next jobs are seen.

Unknown running time:

In some scheduling situation, the running time of each job may be unknown until the job finishes. An online algorithm only knows whether a job is still running or not. At any time, all currently available jobs are at the disposal of the algorithm.

Jobs arrive over time

In this type of scheduling, the running time of each job is known at the time of arrival of that job, but the arrival time of each job is not known in advance.

Interval Scheduling:

In this type of scheduling, each job is to be executed in a predetermined time interval. If it is impossible to execute the job in that time interval, then the job will be rejected. Here, the objective is to maximize the number of accepted jobs.

3. Classification of Uniform Parallel Machines Scheduling Problems

By taking the offline/online features, measures of performance and non-preemptive/preemptive nature of the jobs into account, the uniform parallel machines scheduling problems can be classified into the following categories.

1) Offline scheduling of non-preemptive jobs to minimize makespan.

2) Offline scheduling of preemptive jobs to minimize makespan.

3) Offline scheduling of non-preemptive jobs to minimize the sum of total completion times.

4) Offline scheduling of preemptive jobs to minimize the sum of total completion times.

5) Offline scheduling of non-preemptive jobs to minimize total earliness/tardiness.

(*Offline scheduling of preemptive jobs to minimize total earliness/tardiness is not included due to lack of reference*)

6) Offline scheduling of non-preemptive jobs to minimize maximum lateness.

7) Offline scheduling of preemptive jobs to minimize the maximum lateness.

8) Offline scheduling of non-preemptive jobs to minimize the number of tardy jobs.

9) Offline scheduling of preemptive jobs to minimize the number of tardy jobs.

10) Online scheduling of non-preemptive jobs to minimize the makespan.

11) Online scheduling of preemptive jobs to minimize the makespan.

12) Online scheduling of preemptive jobs to minimize the total tardiness/earliness.

13) Online scheduling of non-preemptive jobs to maximize the minimum completion time (*machine covering problem*).

14) Online scheduling of preemptive jobs to maximize the minimum completion time (*machine covering problem*).

15) Miscellaneous problems with non-preemptive jobs.

16) Miscellaneous problems with preemptive jobs.

17) Miscellaneous problems with periodic tasks/jobs.

A comparison of the above problems as per this classification is shown in **Table 2**.

4. Review of Literature

Identical parallel machines scheduling forms the basics for uniform parallel machines scheduling. Coffman and Graham [2] have formulated a general model of computation structures and exhibited an efficient algorithm for finding optimal non-preemptive schedules for the identical parallel machines scheduling with the objective of minimizing the makepsan. They proved that their algorithm gives optimal solution.

4.1. Offline Uniform Parallel Machines Scheduling

In this section, the literature review of the following classifications of the offline parallel machines scheduling problem is presented.

1) Offline scheduling of non-preemptive jobs to minimize makespan.

2) Offline scheduling of preemptive jobs to minimize makespan.

3) Offline scheduling of non-preemptive jobs to minimize sum of total completion times.

4) Offline scheduling of preemptive jobs to minimize sum of total completion times.

5) Offline scheduling of non-preemptive jobs to minimize total earliness/tardiness.

6) Offline scheduling of non-preemptive jobs to minimize maximum lateness.

7) Offline scheduling of preemptive jobs to minimize maximum lateness.

8) Offline scheduling of non-preemptive jobs to minimize number of tardy jobs.

9) Offline scheduling of preemptive jobs to minimize number of tardy jobs.

4.1.1. Offline Scheduling of Non-Preemptive Jobs to Minimize Makespan

Horowitz and Sahni [3] first presented dynamic programming algorithms for scheduling independent tasks in a multiprocessor environment in which the processors have different speeds. In this research, the objective is minimize the finish time (makespan) and weighted mean flow time on two processors. These algorithms have worst-case complexity functions which are exponential in the number of tasks. Hence, they next presented approximation algorithms of low polynomial complexity for the above problem. Ibarra and Kim [4] have developed a heuristic for scheduling independent tasks on nonidentical processors. In their study, particularly, for $m = 2$, an $n \log n$ time-bounded algorithm is given which

Table 2. Comparison of single machine scheduling problems with uniform parallel machines.

Problem No	Description	Distinguishing characteristics		Measure of performance
		Online/ offline	Preemptive/ non-preemptive	
1	Offline scheduling of non-preemptive jobs to the minimize makespan	Offline scheduling	Non-preemptive jobs	This category minimizes the makepan, which is the minimization of the maximum of the completion times of the last jobs on all the machines.
2	Offline scheduling of preemptive jobs to minimize the makespan	Offline scheduling	Preemptive jobs	This category minimizes the makepan, which is the minimization of the maximum of the completion times of the last jobs on all the machines.
3	Offline scheduling of non-preemptive jobs to minimize the sum of total completion times	Offline scheduling	Non-preemptive jobs	This category minimizes the sum of the total completion times of all the jobs and related other measures like sum of the flow time (weighted/unweighted)
4	Offline scheduling of preemptive jobs to minimize the sum of total completion times	Offline scheduling	Preemptive jobs	This category minimizes the sum of the total completion times of all the jobs and related other measures like sum of the flow time (weighted/unweighted).
5	Offline scheduling of non-preemptive jobs to minimize the total earliness/tardiness	Offline scheduling	Non-preemptive jobs	This category minimizes the total earliness/tardiness of all the jobs which are scheduled on all the machines.
6	Offline scheduling of non-preemptive jobs to minimize the maximum lateness	Offline scheduling	Non-preemptive jobs	This category minimizes the maximum lateness of the jobs on all the machines.
7	Offline scheduling of preemptive jobs to minimize the maximum lateness	Offline scheduling	Preemptive jobs	This category minimizes the maximum lateness of the jobs on all the machines.
8	Offline scheduling of non-preemptive jobs to minimize the number of tardy jobs	Offline scheduling	Non-preemptive jobs	This category minimizes the number of tardy jobs in the schedule.
9	Offline scheduling of preemptive jobs to minimize the number of tardy jobs	Offline scheduling	Preemptive jobs	This category minimizes the number of tardy jobs in the schedule.
10	Online scheduling of non-preemptive jobs to minimize the makespan	Online scheduling	Non-preemptive jobs	This category minimizes the makepan, which is the minimization of the maximum of the completion times of the last jobs on all the machines.
11	Online scheduling of preemptive jobs to minimize the makespan	Online scheduling	Preemptive jobs	This category minimizes the makepan, which is the minimization of the maximum of the completion times of the last jobs on all the machines.
12	Online scheduling of preemptive jobs to minimize the total tardiness/earliness	Online scheduling	Preemptive jobs	This category minimizes the total earliness/tardiness of all the jobs which are scheduled on all the machines.
13	Online scheduling of non-preemptive jobs to maximize the minimum the completion times (machine covering problem)	Online scheduling	Non-preemptive jobs	This category maximizes the minimum of completion times of the last jobs on the parallel machines. This problem is called as machine covering problem. The objective of this measure is to keep all the machines in working mode during the schedule period.
14	Online scheduling of preemptive jobs to maximize the minimum completion times (machine covering problem)	Online scheduling	Preemptive jobs	This category maximizes the minimum of completion times of the last jobs on the parallel machines. This problem is called as machine covering problem.
15	Miscellaneous problems with non-preemptive jobs	Miscellaneous problems	Non-preemptive jobs	The measures of performance which are not part of the standard list as mentioned in this in Subsection 2.1.
16	Miscellaneous problems with preemptive jobs	Miscellaneous problems	Preemptive jobs	The measures of performance which are not part of the standard list as mentioned in this in Subsection 2.1.
17	Miscellaneous problems with periodic tasks/jobs	Miscellaneous problems	Periodic tasks/ jobs	A periodic task T is characterized by two integer parameters – an execution requirement T.e and a period T.p with the interpretation that the task generates a job at each integer multiple of T.p and such job has an execution requirement of T.e execution units, which should be met by a deadline equal to the next integer multiple of T.p.

generates a schedule having a finishing time of at most $(\sqrt{5} + 1)/2$ of the optimal finishing time. They verified that the *LPT* algorithm applied to this problem gives schedules which are near optimal for larger *n*. Prabuddha De and Thomas E. Morton [5] have developed a new heuristic to schedule jobs on uniform parallel processors to minimize makespan. It is tested on a large number of problems for both uniform and identical processors. They found that the solutions given by the heuristic for the uniform parallel machines scheduling are within 5% of the solutions given by the branch and bound algorithm. Bulfin and Parker [6] have considered the problem of scheduling tasks on a system consisting of two parallel processors such that the makespan is minimized. In particular, they treated a variety of modifications to this basic theme, including the cases of identical processors, proportional (uniform) processors and unrelated processors. In addition, they suggested a heuristic scheme when precedence constraints exist.

Friesen and Langston [7] examined the non-preemptive assignment of *n* independent tasks to a system of *m* uniform processors with the objective of reducing the makespan. It is known that *LPT* (*longest processing time first*) schedules are within twice the length of the optimum makespan. Graham [8], they analyzed a variation of the MULTIFIT algorithm derived from the algorithm for bin packing problem and proved that its worst-case performance bound on the makespan is within 1.4 times of the optimum makepsan. Gregory Dobson [9] has given a worst-case analysis while applying the *LPT* (longest processing Time) heuristic to the problem of scheduling independent tasks on uniform processors with the minimum makepsan. In this research, a bound of 19/12 is derived on the ratio of the heuristic to the optimal makespan. Also, a generalization of the classic result of Graham [8] for the case of identical processors is given. The author has derived tight bounds for the ratio of the heuristic makespan to the optimal makespan which depends on the ratio of the longest task to the makespan. Friesen [10] examined the nonpreemptive assignment of independent tasks to a system of uniform processors with the objective of minimizing the makespan. The author showed that the worst case bound for the *largest processing time first* (*LPT*) algorithm for this problem is tightened to be in the interval (1.52 to 1.67). Hochbaum and Shmoys [11] devised a polynomial approximation scheme for the minimizing makespan problem on uniform parallel processors. They gave a family of polynomial-time algorithms such that each algorithm gives a solution that is within 40% relative error of the optimum. The technique employed is the dual approximation approach, where infeasible but super-optimal solutions for a related (dual) problem are converted to the desired feasible but possibly suboptimal solution.

Chen [12] has examined the non-preemptive assignment of independent tasks to a system of *m* uniform processors with the objective of minimizing the makespan. The author has examined the performance of *LPT* (*largest processing time*) schedule with respect to optimal schedules, using the ratio of the fastest speed to the slowest speed of the system as a parameter. Two well-known heuristics LPT (largest processing time first) and MULTIFIT obtains schedules having makespan with ¾ and 11/13, respectively, of the minimum possible makespan, when the *m* parallel processors are identical. The best known worst-case performance ratio bounds are 1.583 and 1.40, respectively. The author has tightened bound to 1.382 for MULTIFIT algorithm for the uniform parallel processors system.

Mireault, Orlin, Vohra [13] have considered the problem of minimizing the makespan when scheduling independent tasks on two uniform parallel machines. Out of the two machines, the efficiency of one machine is q times as that of the other machine. They computed the maximum relative error of the *LPT* (*largest processing time first*) heuristic as a function of *q*. For the special case in which the two machines are identical ($q = 1$), their problem and heuristic are identical to the problem and heuristic analyzed by Graham [8], respectively.

Burkard and He [14] derived the tight worst case bound $\sqrt{6}/2 + (1/2)^k$ for scheduling jobs using the *MULTIFIT* heuristic on two parallel uniform machines with *k* calls of *FFD* (first fit decreasing) within *MULTIFIT*. As per *FFD*, the tasks are sorted in non-increasing order from left to right. They concluded that when *MULTIFIT* is combined with *LPT* as an incumbent algorithm, the worst case bound decreases to $\sqrt{2} + ½ + (1/2)^k$.

Kovalyov and Shafransky [15] have studied the uniform machine scheduling of unit-time jobs subject to resource constraints. Some jobs may require a unit of an additional single resource during their execution. The resource is renewable, but the total resource consumption is limited by the same value at each time instant. They presented an $O(m \log m)$ algorithm for the problem with no machine idle times to minimize the maximum job completion time, that is the makespan. They also presented a linear time algorithm for the problem with identical machines to minimize the maximum job completion time. Burkard, He and Kellerer [16] have developed a linear compound algorithm for scheduling jobs on uniform parallel machines with the objective of minimizing makespan. This algorithm has three subroutines, which run independently in order to choose the best assignment among them. Panneerselvam and Kanagalingam [17] have presented a mathematical model for parallel machines scheduling problem with varying speeds in which the objective is to minimize the makespan. Also, they discussed industrial applications of such scheduling problem. Panneerselvam and Kanagalingam [18] have given a heuristic to minimize the makespan for scheduling *n* independent jobs on *m* parallel processors with different speeds.

Chandra Chekuri and Michael Bender [19] designed a new and efficient polynomial 6 approximation algorithm for scheduling precedence-constrained jobs on uniform parallel machines. In their work, the objective is to find a non-preemptive schedule to minimize the makespan. From their work, Woeginger [20] combined some straightforward observations and thereby derived an extremely simple 2 approximation algorithm for this problem. Chudak and Shmoys [21] gave an algorithm with an approximation ratio of $O(\log m)$, significantly improving the earlier ratio of $O(\sqrt{m})$ due to Jaffe [22]. Their algorithm is based on solving a linear programming relaxation. Building on some of their ideas, Chandra Chekuri and Michael Bender [23] have presented a combinatorial algorithm that achieves a similar approximation ratio but in $O(n^3)$ time. Ching-Jong Liao and Chien-Hung Lin [24] have considered the two uniform parallel machines problem with the objective of minimizing makespan. In this work, the two uniform parallel machines problem is converted into a special problem of two identical parallel machines from the viewpoint of workload instead of completion time. An optimal algorithm is developed for the transformed special problem. The proposed algorithm has an exponential time complexity. Inspite of this fact, the authors claim that their algorithm can find the optimal solution for large sized problems in a short time.

Epstein and Sgall [25,26] derived a polynomial approximation scheme for the problem of scheduling on uniformly related parallel machines for a large class of objective functions as listed below that depend only on the machine completion times This generalizes and simplifies many previous results in this area.

1) Minimize the sum of the completion times on all the machines.

2) Minimize the maximum of the completion times on all the machines, which amounts to minimizing the makespan.

3) Maximize the sum of the completion times on all the machines.

4) Maximize the minimum of the completion times on all the machines.

Christos Koulamas and George J. Kyparisis [27] investigated the makespan minimization problem on uniform parallel machines in the presence of release times. They developed a heuristic for this NP-hard problem and derived a tight worst-case ratio bound for this heuristic independent of the machines speeds. Agarwal, Colak, Jacob and Pirkul [28] have proposed new heuristics along with an augmented-neural-netwrok (AugNN) formulation for solving the makespan minimization task-scheduling problem for the non-identical machine environment. They explored four task and three machine-priority rules, resulting in 12 combinations of single-pass heuristics. They gave the AugNN formulation for each of the 12 heuristics and showed computational results on 100 randomly generated problems of sizes ranging from 20 to 70 tasks and 2 to 5 machines. The results clearly showed that AugNN provides significant improvement over single-pass heuristics. The reduction in the gap between the obtained solution and the lower bound due to AugNN over single-pass heuristics ranges from 24.4% to 50%. Chein-Hung Lin and Ching-Jong Liao [29] have considered a classical scheduling problem with makespan minimization on uniform parallel machines. From the viewpoint of workload, instead of completion time, two important theorems are developed for the problem. The first theorem provides an improved lower bound as the starting point of the search and the second theorem further accelerates the search speed in the algorithm. Incorporating the two useful theorems, an algorithm is developed for obtaining the optimal solution. Although the developed algorithm has an exponential time complexity, extensive computational experiments demonstrate that it is quite efficient for various sizes of the problem. With the optimal algorithm, they also examined the effectiveness of the popular *LPT* heuristic.

Kis and Kapolnai [30] researched the scheduling of groups of identical jobs on uniform machines with sequence independent setup times. They provide a 2-approximation algorithm for minimizing the makespan. The second result is truthful, polynomial time, randomized mechanism for the batch scheduling problem with a deterministic approximation guarantee of 4.

For this scheduling problem, most of the authors aimed at developing dynamic programming algorithm, mathematical model, branch and bound algorithm, heuristics based on LPT ordering, etc. Since, this problem comes under combinatorial category, obtaining the optimal solution will take exponential time. But, one can try metaheuristics, viz. simulated annealing algorithm, genetic algorithm, etc., which will give better solution tending towards global optimum, when compared to the single pass heuristics. Further, the researchers may consider multi-objective function for this problem. The possible combinations of the multi-objective problems are as listed below.

- Minimization of makespan and mean flow time
- Minimization of makespan and total tardiness
- Minimization of makespan and number of tardy jobs

While investigating such problems, enough care is to be taken to fix weights for the components of the objective functions, because they have differences in terms of interpretations.

4.1.2. Offline Scheduling of Preemptive Jobs to Minimize Makespan

McCormic and Pinedo [31] have considered the uniform parallel machines scheduling problem with preemption at no cost. They have generated the entire tradeoff curve of schedules which are Pareto-optimal (undominated) for the flow time and makespan objectives. To achieve this, they first developed an $O(mn)$ algorithm that produces a schedule with minimum flow time, subject to a fixed makespan deadline. This algorithm alternates between the Shortest Processing Time on Fastest Machine (SPT-FM) rule and the Longest Remaining Processing Time on Fastest Machine (LRPT-FM) rule. Their knowledge of the structure of optimal schedules allows them to characterize breakpoints on the tradeoff curve, and then to compute all of the $O(mn)$ breakpoints in $O(m^3n)$ time.

Martel [32] describes a fast parallel algorithm for preemptive scheduling of independent jobs on uniform machines to minimize the makepsan. Gonzalez and Sahni [33] have developed a sequential algorithm which solves this problem in $O(n + m \log m)$ time. Martel [32] has developed a parallel version of this algorithm for a concurrent Read Exclusive Write (*CREW*) shared memory computer. This algorithm runs in $O(\log n + \log^3 m)$ time using n processors. Shachnai, Tamir and Woeginger [34] studied the problem of minimizing the makespan with preemption costs on a system of uniform machines scheduling. Pandelis [35] has considered the problem of scheduling independent jobs on uniform parallel machines. Each job has a deterministic processing time and a weight associated with it. In this research, the objective is to minimize the sum of the discounted flow time of the jobs. The author has shown that for scheduling on uniform machines, assigning the job with the shortest remaining time to the fastest available machine (SRPT-FM rule) is optimal in the case preemptive schedules. The author's straightforward extension of this result is that if jobs are assigned weights that satisfy a certain agreeability condition (shortest processing time corresponds to largest weight, second shortest processing time corresponds to second largest weight, and so on), the SRPT-FM rule minimizes discounted weighted flow time.

Though the researchers have used Pareto-optimal planes and other heuristics, only limited researches are carried out in this category of scheduling problem. So, the researchers may concentrate on using the advanced algorithms to the problems with single as well as multi-objective function.

4.1.3. Offline Scheduling of Non-Preemptive Jobs to Minimize Sum of Total Completion Times

Dessouky and Marcellus [36] have developed methods to optimize the expected sum of weighted completion times and the probability of meeting a common due date while scheduling identical jobs on a set of uniform parallel machines with random processing times. Meral Azizoglu and Omer Kirca [37] have considered the NP-hard problem of scheduling jobs on identical parallel machines to minimize total weighted flow time. They also discussed the properties that characterize the structure of an optimal solution, presented a lower bound and proposed a branch and bound algorithm. They also extended the algorithm to uniform parallel machines. Zhi-Long Chen and Warren B. Powell [38] have considered a class of problems of scheduling n jobs on m identical, uniform, or unrelated parallel machines with an objective of minimizing an additive criterion, viz. the total weighted completion times of the jobs. They proposed a decomposition algorithm for solving there problems exactly. The decomposition algorithm first formulates these problems as an integer program, and then reformulates the integer program, using Dantzig-Wolfe decomposition, as a set partitioning problem.

The minimization of the sum of the total completion times of the jobs in turn minimizes the mean flow time, for which there is an exact algorithm namely SPT rule in the single machine scheduling problem with single processor. The minimization of this measure under parallel machines environment makes the problem combinatorial in nature. Hence, the researchers may concentrate on the development of meta-heuristics for this problem. If one uses simulated annealing, an improved version of SPT rule may be designed as the seed generation algorithm.

4.1.4. Offline Scheduling of Preemptive Jobs to Minimize Sum of Total Completion Times

Gonzalez, Leung and Pinedo [39] have analyzed n independent jobs and m uniform machines in parallel. Each job has a processing time and a due date. Job j may complete its processing before or after its due date and preemptions are allowed. A set of jobs is said to be feasible if there exists a schedule that meets all the due dates. They presented a polynomial-time algorithm that given a feasible set of jobs, which constructs a schedule that minimizes the total completion time of the jobs.

Leung, Li, Pinedo and Jiawei Zhang [40] researched the scheduling of orders in an environment with m uniform machines in parallel. Each order requests certain amounts of k different product types. Each product type can be produced by any one of the m machines. No setup is required if a machine switches over from one product type to another. Different product types intended for the same order can be produced at the same time (concurrently) on different machines. Each order is released at time zero and has a positive weight. The completion time of an order is the finish time of the product type that is completed last for that order. The objective of this research is to minimize the total weighted completion time of orders. They proposed heuristics for the non-preemptive as well as preemptive case and obtained worst-case bounds that are functions of the number of machines as well as the difference in the speeds of the machines.

As discussed in the previous section, the indirect objective of this problem is to minimize the mean flow time. Since, this is less complex problem, when compared to the non-preemptive type, the researchers may try to develop mathematical models for different cases of the problem. In turn, the optimal solutions through a model for small or moderate size problems may be used for bench marking purpose while developing heuristics.

4.1.5. Offline Scheduling of Non-Preemptive Jobs to Minimize Total Earliness/Tardiness

Kun-si Lin [41] developed an integer programming algorithm to solve the parallel machines problem with the objective of minimizing total tardiness in which different machines are allowed to have different processing speeds on a job. In this work, a solution procedure is introduced beginning with the single machine problem and then extended to the general case of m machines. Alain Guinet [42] considered the problem of scheduling n jobs on m uniform parallel machines with the objective of minimizing the mean tardiness or the weighted sum of tardiness with weights based on jobs, periods or both. For the mean tardiness criterion in the non-preemptive case, the problem is NP-hard, except for the cases with equal job processing times or with job due dates equal to job processing times. They have developed a heuristic to solve the non-preemptive scheduling problem with unrelated job processing times. The heuristic was experimented with 576 problems which consist of 18 problem sizes.

Meral Azizoglu and Omer Kirca [43] have considered the NP-hard problem of scheduling jobs on identical parallel machines to minimize the total tardiness. They presented the properties that characterize the structure of an optimal schedule. They proposed a branch and bound algorithm that incorporates the properties along with an efficient lower bounding scheme. They found that optimal solutions can be obtained in reasonable times for problems with size up to 15 jobs. In the last part of the study, they extended the results to uniform parallel machines. Naofumi, Shunji and Mitsuhiko [44] have considered a class of problems to minimize total tardiness on uniform parallel machines with human resource constraints, in which each machine requires an operator to process a job. They constructed a branch and abound algorithm for this problem and examined the efficiency of it by numerical examples.

Viniclus Amaral Armentano and Moacir Fellzardo de Franca Fllho [45] have considered the problem of scheduling jobs in uniform parallel machines with sequence dependent setup times in order to minimize the total tardiness of the jobs. They proposed GRASP versions that incorporate adaptive memory principles to solve the problem. Long-term memory is used in the construction of an initial solution and in a post optimization procedure which connects high quality local optima by means of path re-linking. They carried out computational tests on a set of benchmark instances and compared the proposed GRASP versions with heuristic methods from the literature. Toung, Soukhal and Jean-Charless Billaut [46] have developed algorithm for uniform parallel machines scheduling problem with a common due date to minimize total weighted tardiness. Gur Mosheiov and Assaf Sarig [47] have shown that the two machines case of the uniform parallel machines scheduling with earliness and tardiness and due-date costs is solvable in constant time. Also, they stated that the problem remains polynomially solvable for a fixed number of machines.

The minimization of the total earliness/tardiness is considered to be a challenging measure of performance even in the single machine scheduling problem with single processor. This magnifies the complexity of the algorithm to obtain the optimal solution. The researchers have used integer programming technique, branch and bound technique, single pass heuristic, GRASP, etc to minimize the measures of performance of this problem. As mentioned earlier, since it is a challenging measure, which magnifies the complexity of the problem, use of any exact algorithms like integer programming, branch and bound technique, etc. will take too much computational time. So, the researchers should concentrate on the development of meta-heuristics to obtain near-optimal/global optimum solutions for this scheduling problem.

4.1.6. Offline Scheduling of Non-Preemptive Jobs to Minimize Maximum Lateness

Federgruen and Groenevelt [48] considered the problem of scheduling n jobs, each with a specific processing requirement, release time and due date on m uniform parallel machines. They have obtained a feasible schedule by using network flow technique which determines the maximum flow in network. The complexity of the procedure used by them is $O(tn^3)$ operations, where t is the number of distinct machine types. Previous algorithm solves the feasibility problem in $O(m + \log n)\ (m^2n^3 + n^4)$ operations. In this research, they further described algorithms for minimizing the maximum lateness.

Dessouky [49] has considered the problem of scheduling n identical jobs with unequal ready times on m parallel uniform machines to minimize the maximum lateness. In this research, a branch and bound procedure and six simple single-pass heuristic procedures are presented. The branch and bound procedure uses the heuristics to establish an initial upper bound. On sample problems, the branch and bound procedure in most instances was able to find an optimal solution within 100,000 iterations when the number of jobs is less than or equal to 80 and the number of machines is less than or equal to 3. For larger values of the number of machines, this heuristics provided approximate solutions close to optimal values.

Chhajed [50] has analyzed the problem of simultaneous determination of a common due-date and a sequence of n jobs to minimize the maximum deviation of job completion time around the common due-date. It is assumed that all the jobs are available at time zero and their processing times are known in advance. He gave some results for the case when splitting and preemption are allowed.

Chudak and Shmoys [51] presented new approximation algorithms for the problem of scheduling precedence-constrained jobs on parallel machines that are uniformly related without preemption. Here, they considered two objective functions: $C_{max} = \max_j C_j$, where C_j denotes the completion time of the job j and $\sum w_j C_j$, where w_j is a weight that is given for each job j. For the first objective, the best previously known result is an $O(\sqrt{m})$-approximation algorithm, which was shown by Jaffe [22]. They have given an $O(\log m)$ approximation algorithm. They also showed how to extend this result to obtain an $O(\log m)$-approximation algorithm for the second objective. There results also extend to settings in which each job has a release date r_j before which the job may not begin processing.

Christos Koulamas and George J. Kyparisis [52] considered the uniform parallel machines scheduling problem with the objective of minimizing maximum lateness. They showed that an extension of EDD rule to a uniform parallel machines setting yields a maximum lateness value which does not exceed the optimal value by more than p_{max}, where p_{max} is the maximum job processing time.

The researchers have used network flow technique, branch and bound technique, heuristics, etc in the past. The minimization of the maximum lateness improves the goodwill of the customers, because the maximum delay is minimized, for which there is an exact algorithm namely EDD rule in the single machine scheduling problem with single processor. The minimization of this measure under parallel machines environment makes the problem combinatorial in nature. Hence, the researchers may concentrate on the development of meta-heuristics for this problem. If one uses simulated annealing, an improved version of EDD rule may be designed as the seed generation algorithm.

4.1.7. Offline Scheduling of Preemptive Jobs to Minimize Maximum Lateness

Drozdowski, Blazewicz, Formanowicz, Kubiak and Schmidt [53] have studied the problem of scheduling n preemptive tasks with ready times and due-dates on m uniform processors available in q time windows for minimizing maximum lateness criterion. The problem is reduced to a sequence of network flow problems. The complexity of the algorithm is $O(n+q)^3(\log n + \log q + \log m + \log \max\{b_i\})$, where b_i is speed of the processor i.

Only few researchers have concentrated on this problem. The researchers may extend the contributions made in the non-preemptive case to this problem.

4.1.8. Offline Scheduling of Non-Preemptive Jobs to Minimize Number of Tardy Jobs

As already stated in one of the earlier sections, Zhi-Long Chen and Warren B.Powell [38] have considered a class of problems of scheduling n jobs on m identical, uniform, or unrelated parallel machines with an objective of minimizing an additive criterion, viz. the total weighted completion times of the jobs. In addition to this measure, they also considered the objective of minimizing the weighted number of tardy jobs. They proposed a decomposition algorithm for solving these problems exactly.

Ruiz-Torres, Lopez and Ho [54] investigated the uniform parallel machines scheduling problem subject to a secondary resource constraint to minimize the number of tardy jobs. The secondary resource is fixed in quantity and is to be allocated to the machines at the start of the schedule. Two versions of the problem are analyzed. The first version assumes that the jobs are pre-assigned to the machines, while the second one takes into consideration the task of assigning jobs to the machines. They proposed an integer programming formulation to solve the first case and a set of heuristics for the second case.

One can use Hodgson's algorithm to optimize the number of tardy jobs in the single machine scheduling problem with single processor with un-weighted jobs. When the jobs are with weights, the optimality is not guaranteed. Since, this measure is a derived measure from tardiness, it is a challenging measure. So, the researchers may use meta-heuristics to obtain global optimal solution. Further, the seed generation algorithm for the simulated annealing algorithm may be designed using an improved version of Hodgson's algorithm.

4.1.9. Offline Scheduling of Preemptive Jobs to Minimize Number of Tardy Jobs

Lawler and Martel [55] have considered the problem of scheduling n preemptive jobs on uniform parallel machines. In this research, they formed a schedule such that the number of tardy jobs is minimized. They presented an algorithm with the complexity of $O(n^3)$ for the special case of two uniform machines. Also, they gave a fully polynomial scheme for the weighted case.

A very few researches have been carried out in this problem. So, there is a scope for further work in this problem with the guidelines as given in the previous section.

4.2. Review of Online Uniform Parallel Machines Scheduling

The performance of an online algorithm is measured by its competitive ratio which is the ratio of its worst case performance and the performance of an optimal algorithm with total prior knowledge. If the makespan is the measure of performance, alternatively, competitive ratio is the ratio between the cost of online schedule (make- span) and the cost of offline schedule (optimal makepsan).

The literature review of the following online uniform parallel machines scheduling problems is presented in this section.

1) Online scheduling of non-preemptive jobs to minimize makespan.

2) Online scheduling of preemptive jobs to minimize makespan.

3) Online scheduling of preemptive jobs to minimize total tardiness/earliness

4) Online scheduling of non-preemptive jobs to maximize the minimum completion time (*machine covering problem*).

5) Online scheduling of preemptive jobs to maximize the minimum completion time (*machine covering problem*).

4.2.1. Online Scheduling of Non-Preemptive Jobs to Minimize Makespan

Wein and Williamson [56] have studied the problem of scheduling jobs on parallel machines in an online fashion in which the processing requirement of a job is not known until that job is completed. For this problem, they scheduled jobs such that the makespan is minimized. They studied the following two models, viz. scheduling on identical parallel machines and scheduling on uniform parallel machines. Their results include the following.

- Matching upper and lower bounds on the competitive ratio for the case of identical machines.
- Upper and lower bounds that differ by a constant factor for uniformly related machines.
- A lower bound for randomized algorithms for identical machines that nearly matches the deterministic upper bound.
- Several upper and lower bounds for variations on these models.

A study on establishing a lower bound for makespan in on-line scheduling on uniformly related machines was done by Epstien and Sgall [57]. They considered the problem of on-line scheduling of jobs one by one on uniformly related machines, with or without preemption. They proved a lower bound of 2, both with and without preemption for randomized algorithms working for an arbitrary number of machines.

Tan and He [58] have investigated the semi-online scheduling problem with ordinal data on two uniform machines where the order of jobs by their processing times is known as priori. They presented a comprehensive lower bound which is a piecewise function of the speed ratio *s*. The algorithm gives optimal makespan for majority of s $(1, \infty)$. In this paper, the total length of the intervals of s where the competitive ratio does not match the lower bound is less than 0.7784 and the biggest gap between them never exceeds 0.0521. Kontogiannis [59] has studied the problem of assigning unit size tasks to related machines when only limited online information is provided to each task. The author has proved that the missing information for an oblivious scheduler to perform almost optimally is the amount of tasks to be inserted into the system. In particular, the author provided an oblivious scheduler that only uses $O(\log \log n)$ polls along with the additional information of the size of the input sequence, in order to achieve a constant competitive ratio. Finally, this oblivious scheduler is used in an adaptive scheduler that does not demand the knowledge of the input sequence and yet achieves almost the same performance. Epstein and Favrholdt [60] have considered non-preemptive semi-online scheduling problem in which jobs with non-increasing sizes arrive one by one which are to be scheduled on two uniformly related machines, with the goal of minimizing the makespan. They analyzed both the optimal overall competitive ratio and the optimal competitive ratio as a function of the speed ratio ($s \geq 1$) between the two machines. They showed that the greedy algorithm LPT has optimal competitive ratio of $\frac{1}{4}(1+\sqrt{17}) \approx 1.28$ overall, but does not have optimal competitive ratio for every value of *s*. They gave a tight analysis of the competitive ratio for every speed ratio.

Cheng, Ng and Vladirmir Kotov [61] have researched the online scheduling problem with m-1, $m \geq 2$, uniform machines each with a processing speed of 1 and one machine with a speed of s, $1 \leq s \leq 2$, to minimize the makespan. The well known list scheduling (LS) algorithm has a worst-case bound of (3m-1)/(m+1) (Sahni and Cho, [62]). An algorithm with a better competitive ratio was proposed by Li and Shi [63]. It has a worst-case bound of 2.8795 for large value of m and n = 2. In the note by Cheng, Ng and Vladirmir Kotov [61], they presented an algorithm with a competitive ratio of 2.45 for $m \geq 4$ and any s, $1 \leq s \leq 2$.

Angelelli, Speranza and Tuza [64] have considered the problem of online scheduling on two uniform processors where the total sum of the tasks is known in advance. In this research, tasks arrive one at a time and each task is to be assigned to one of the two processors before the next task arrives. The assignment can be changed later. The objective is the minimization of the makespan. By assuming *s* as the speed of the fast processor and 1 as the speed of the slow processor, they derived general lower bounds on the competitive ratio achievable with respect to offline optimum and designed on-line algorithms with guaranteed upper bound on their competitive ratio.

The researchers have concentrated on the analysis of lower bound for the makespan of this problem as well as on the development of heuristics with different competitive ratios. In future, researchers may aim to develop algorithms which yield better competitive ratio.

4.2.2. Online Scheduling of Preemptive Jobs to Minimize Makespan

Wen and Du [65] have considered the problem of preemptive on-line scheduling for two processors in which one of the processors has speed 1 and the speed of the other processor is greater than or equal to 1. In this research, the objective is to minimize the makespan. They proposed an algorithm with competitive ratio of $(1+s)^2/(1+s+s^2)$ for this problem.

Vestjens [66] considered the online scheduling problem in which *n* jobs, where *n* is unknown are to be scheduled on *m* uniform parallel machines with preemption of jobs such that the makespan is minimized. The processing time of the job is known on its arrival. The author has shown that if only a finite number of preemptions is allowed, then there exists an algorithm that solves the problem if and only if $s_{i-1}/s_i \leq s_i/s_{i+1}$ for all i = 2, …, *m*-1, s_i is the largest machine speed. It is shown that if this condition is to be satisfied, then $O(mn)$ preemptions are necessary.

Epstein, Noga, Seiden, Sgall and Woeginger [67] [68], studied the problem of on-line scheduling on two uniform machines with speeds 1 and $s \geq 1$. Here, the objective is to obtain a schedule, which minimizes the makespan. First, they presented randomized results for this problem. Then, they showed a simple memory-less algorithm with competitive ratio $(4-s)(1+s)/4 \leq 1.5625$. Also, they analyzed other randomized algorithms which demonstrate that the randomized competitive ratio is at most 1.52778 for any s. Finally, they presented a deterministic algorithm with competitive ratio of $1+s/(s^2 + s + 1)$ for the preemptive version of this problem. Epstein [69,70] studied the preemptive scheduling on uniformly related processors, where jobs are assigned one by one in an on-line fashion such that the makespan is minimized. This paper discusses the class of machine sets where the speed ratios are non-decreasing as speed increases. For each set of machines in this class, an algorithm of optimal competitive ratio is deigned in this paper. This generalizes the known result for identical machines and solves other interesting cases.

A study on preemptive semi-online scheduling was carried out by Epstein and Favrholdt [71]. In this study, jobs with non-increasing sizes arrive one by one which are to be scheduled on two uniformly related machines. They analyzed the algorithms as a function of the speed ratio ($s \geq 1$) between the two machines. Here, the objective is to minimize the makespan. Then, they designed algorithms of optimal completive ratio for all values of s and showed that for $s > 2$, idle time needs to be introduced. This is the first preemptive scheduling problem over list, where idle time is provably required. Donglei Du [72] also investigated the same problem within a certain range [1,r] ($r \geq 1$) with the objective of minimizing the makespan. The author has characterized the optimal competitive ratio as a function of both *s* and *r* by devising a deterministic on-line scheduling algorithm.

The researchers have concentrated on the development of algorithms with different competitive ratios. In future, researchers may aim to develop algorithms which yield better competitive ratio.

4.2.3. Online Scheduling of Preemptive Jobs to Minimize Sum of Tardiness/Earliness

Balakrishnan, Kanet and Sridharan [73] have researched the problem of scheduling jobs uniform parallel machines to minimize the sum of earliness and tardiness costs. Jobs are assumed to arrive in a dynamic albeit deterministic manner and have non-identical due dates. Any job completion beyond its due date results in earliness or tardiness penalty, which may be different for different jobs. Setup times are job-sequence dependent and may be different on different machines based on the characteristics of the machines. For this problem, they presented a mixed integer formulation that has substantially fewer zero-one variables than typical formulations for scheduling problems of this type. They have reported their computational experience in using this model to solve small size problems and presented solution procedures for solving larger size problems. Bilge, Kirac, Kurtulan and Pekgun [74] have considered the problem of scheduling a set of dependent jobs with sequence dependent setups on a set of uniform parallel machines such that the total tardiness is minimized. Jobs have non-identical due dates and arrival times. They have used tabu search based algorithm for this problem. They have used several key components of tabu search such as candidate list strategies, tabu classifications, tabu tenure and intensification and diversification strategies in their algorithm. Alternate approaches to each of these issues are developed and extensively tested on a set of problems obtained from the literature. The results obtained are considerably better than those reported previously and constitute the best solutions known for the benchmark problems up to date.

As stated earlier, the minimization of the total earliness/tardiness is considered to be a challenging measure of performance even in the single machine scheduling problem with single processor. This magnifies the complexity of the algorithm to obtain the optimal solution. The researchers have concentrated on the development of mathematical model and tabu search based algorithm for this problem. As mentioned earlier, since it is a challenging measure magnifying the complexity of the problem, use of any model will take too much computational time. So, the researchers should concentrate on the development of meta-heuristics to obtain near-optimal/global optimum solutions for this scheduling problem.

4.2.4. Online Scheduling of Non-Preemptive Jobs to Maximize Minimum Completion Time (Machine Covering Problem)

Azar and Epstein [75,76] have considered the problem of scheduling a sequence of jobs on m parallel machines such that the *minimum load over the machines is maximized* in an online environment. Such problem is called as machine covering problem. This situation corresponds to a case that a system consists of m machines. They have given several theorems for identical parallel machines as well as for related parallel machines along with proofs. It is well known that any on-line deterministic algorithm for identical machines has a competitive ratio of at least m. In contrast they designed an on-line randomized algorithm which is $O(\sqrt{m})$ competitive and a matching lower bound of $\sqrt{m}$ for any online randomized algorithm. In the case where the jobs are polynomially related, they designed an optimal $O(\log m)$ competitive randomized algorithm and a matching tight lower bound for any on-line randomized algorithm. For related machines, they showed that there is no online algorithm, whose competitive ratio is a function of the number of machines. However for the case where the value of the optimal assignment is known in advance and for the case where jobs arrive in non-increasing order, they showed that the exact competitive ratio is m. Azar and Epstein [77] have developed a polynomial approximation scheme for the machine covering problem. They reported that the previous best approximation algorithm has a performance ratio of 2. They provided an approximation scheme for the related machines scheduling. Their algorithm can be adapted to provide a simpler approximation scheme for the related machines scheduling too.

Tan, He and Epstein [78] have considered the non-preemptive ordinal on-line scheduling of n independent jobs (p_1, p_2, p_3, $\ldots p_n$) on two uniform related machines. They have developed optimal algorithms for *maximizing the minimum machine completion time and minimizing the* l_p *norm* of the completion times. They assumed that the values of the processing times of the jobs are unknown at the time of assignment. However, it is known in advance that processing times of arriving jobs are sorted in a non-increasing order. They have constructed an assignment of all jobs to the machines at time zero by utilizing only ordinal data rather than actual magnitudes of jobs. For the problem of maximizing the minimum completion time, they first presented a comprehensive lower bound on the competitive ratio which is a piecewise function of machine speed ratio s. Then they proposed an algorithm which is optimal for any s which is greater than or equal to 1. For minimizing the l_p norm, they studied the case of identical machines ($s = 1$) and present tight bounds as a function of p.

This is special class of problem. The researchers have concentrated on the development of algorithms with different competitive ratios. In future, researchers may aim to develop algorithms which yield better competitive ratio.

4.2.5. Online Scheduling of Preemptive Jobs to Maximize Minimum Completion Time (Machine Covering Problem)

He and Jiang [79] considered the semi-online preemptive scheduling problem with decreasing job sizes on two uniform machines. The goal of this paper is to maximize the continuous period of time (starting from time zero) when both machines are busy, which is equivalent to maximizing the minimum machine completion time if idle time is not introduced before all the jobs are completed. They designed optimal deterministic semi-online algorithms for every machine speed ratio s in the range from 1 to ∞ and showed that idle time is required during the assignment procedure of algorithms for any s greater than $\sqrt{6}/2$.The competitive ratio ratios of the algorithms match the randomized lower bound for every s in the range from 1 to 3.

A very few researchers have concentrated on this problem on the development of algorithms and analysis of competitive ratio. In future, researchers may aim to develop algorithms which yield better competitive ratio.

4.3. Miscellaneous Uniform Parallel Machines Scheduling

This section gives the review of articles on uniform parallel machines scheduling problem which are not grouped under any of the Subsection 4.1 and Subsection 4.2. Since, miscellaneous problem are discussed in the subsections, no specific comment is given in each of them.

4.3.1. Miscellaneous Problems with Non-Preemptive Jobs

Bahram Alidaee and Ahmad Ahmadian [80] have considered the problem of scheduling n single-operation jobs on 2 non-identical parallel machines where the sequencing of the jobs and their processing times are decision variables. In this research, the authors have assumed that the cost of processing a job is directly proportional to its processing time. The objectives of their research are as listed below.

1) Minimizing the total processing cost plus total flow time.

2) Minimizing the total processing cost plus weighted earliness and weighted tardiness.

They have reduced each of these problems to a transportation problem, which can be solved by a polynomial time algorithm.

Bahram Alidaee and Ahmad Ahmadian [81] have considered the single machine scheduling problem with variable speed. They have established polynomial time algorithms to minimize the makespan, total flow time and sum of the deviations of jobs from a common due date.

Vahid, Mahmoud and Mohsen [82] have presented a new deadline-based algorithm for the online uniform parallel machines scheduling problem with overloading situations. Then, they compared its performance with that of earliest due date first (EDF) algorithm. It is shown that their algorithm not only demonstrated a performance close to that of EDF in non-overloaded conditions but also has supremacy over EDF in overloaded situations in many aspects. Furthermore, it imposes much less overhead on the system. Bekki and Meral Azizoglu [83] studied an operational fixed interval scheduling problem on uniform parallel machines. Here, the objective is to maximize the total weight of the jobs processed. They showed that the problem is NP-hard in the strong sense and develop polynomial time algorithms for some special cases. They proposed a branch and bound algorithm that employs dominance conditions and tight bounds. He and Min [84] have considered the problem of online uniform machine scheduling with rejection. For the two machine case and a special three machine case, they presented the best possible online algorithms for certain values of speed ratio *s*.

4.3.2. Miscellaneous Problem with Preemptive Jobs

Blazewicz [85] has considered the problem of minimizing mean weighted information loss. In this paper, the preemptive case for identical as well as for a fixed number of uniform processors was considered. Blazewicz and Finke [86] have proposed a strongly polynomial algorithm based on a network flow technique, which minimizes the mean weighted execution time loss for an arbitrary number of identical processors as well as uniform processors. The upper bound on the number of preemptions in each of the cases is also reported.

Ishi, martel, Masuda and Nishida [87] considered a scheduling problem in which the objective is to determine both the optimal speeds of processors and an optimal schedule in a preemptive multiprocessor environment. The jobs are independent and each processor can be assigned any speed. The cost associated with each processor is a function of the speed of the processor. They presented polynomial algorithms to find the optimal speed assignments for a variety of cost functions such that the total cost is minimized in each case. Blazewicz, Bouvry, Guinand and Trystram [88] presented an optimal algorithm for scheduling a complete k-ary tree on two uniform processors of different speeds in order to minimize the makespan. They considered the basic case of unit standard execution times and unit communication times.

Shakhlevich and Strusevich [89] have provided a unified approach to solve preemptive scheduling problems with uniform parallel machines and controllable processing times. They demonstrated that a single criterion problem of minimizing total compression cost subject to the constraint that all due dates should be met can be formulated in terms of maximizing a linear function over a generalized polymatriod. This justified the applicability of the greedy approach and allowed them to develop fast algorithms for solving the special case with zero release dates and a common due date. For the bicriteria counterpart of the latter problem, they developed an efficient algorithm that constructs the trade-off curve for minimizing the compression cost and the makepsan.

Kubiak, Penz and Trystram [90] showed that the problem of scheduling chains of unit execution time (UET) jobs on uniform processors with communication delays to minimize the makespan is NP-hard in the strong sense. They also presented a heuristic that generates solutions with known and relatively small, absolute error for this problem. The NP–hardness result holds even for the case without communication delays and complements the earlier result of Gonzalez and Sahni [33] who gave a polynomial time algorithm for preemptive jobs of arbitrary length. They also studied the structure of optimal solutions for the two processor problem scheduling chains of UET jobs with communication delays, where one processor is an integer times faster than the other. This investigation resulted with a linear time optimization algorithm for this case.

4.3.3. Miscellaneous Problem with Periodic Tasks

Baruah [91] has considered scheduling systems of real-time tasks that are specified according to periodic model. In the periodic model of hard real-time tasks, a task *T* is characterized by two integer parameters – an execution requirement *T.e* and a period *T.p* with the interpretation that the task generates a job at each integer multiple of *T.p* and such job has an execution requirement of *T.e* execution units, which should be met by a deadline equal to the next integer multiple of *T.p*. A periodic task system consists of several such periodic tasks that are to execute on specified processor architecture. In this research, the job execution on a processor may be preempted. The author has designed an optimal algorithm for scheduling such periodic tasks on uniform parallel machines. Goossens and Baruah have considered the online scheduling of hard-real-time systems in which all jobs must be completed by specified deadlines on uniform multiprocessor machines. In this paper, they provided resource-augmentation techniques that permit on-line algorithms to perform better given the inherent limitations. Results derived here are applied to the scheduling of periodic task systems on uniform multiprocessor machines. Baruah and Goossens [92] considered the rate-monotonic scheduling on uniform multiprocessors. For systems comprised of periodic tasks that are to execute upon a single shared processor, a very popular static-priority run-time scheduling algorithm is the rate-monotonic scheduling algorithm (Algorithm RM). It assigns each task a priority inversely proportional to its period. The authors have obtained the first non-trivial feasibility test for a static-priority scheduling algorithm that adopts a global approach to task allocation on uniform multiprocessors. They have obtained simple sufficient conditions for determining whether any given periodic task system will be successfully scheduled by Algorithm RM upon a given uniform multiprocessor platform.

5. Conclusions

In this paper, an extensive review of literature of the single machine scheduling problem with uniform parallel processors is presented. First, an introduction of this problem is presented with its different measures of performance, viz. minimizing mean flow time, minimizing total tardiness, minimizing maximum lateness, minimizing the number of tardy jobs and maximizing the minimum of the completion times of the last jobs on the uniform parallel machines. This problem is reviewed with the primary classification of offline scheduling and online scheduling. Under each such scheduling, the next classification is non-preemptive scheduling and preemptive scheduling. Under each of the above combinations, the applicable measures of performance are considered to classify the literatures. In this way, the authors have classified the literatures into 14 categories. Finally, under miscellaneous problems, three problems, viz. non-preemptive, preemptive and periodic tasks are considered. This resulted with a total of 17 categories. For each category of literatures, a comprehensive review of literature has been presented.

In different cases of the offline scheduling problems, in future, the researchers may concentrate the following.

- For the problem under non-preemptive and preemptive cases to minimize the makespan, the researchers may concentrate on the development of meta-heuristics, viz. simulated annealing algorithm, genetic algorithm, etc., which will give better solution tending towards global optimum, when compared to the single pass heuristics. Further, effort may be directed on the development of algorithms with multi-objective function.
- For the problems under preemptive case with the minimization of the sum of the total completion times, the researchers may concentrate on the development of meta-heuristics for this problem. If one uses simulated annealing, an improved version of SPT rule may be designed as the seed generation algorithm. For the non preemptive case of this problem, the researchers may try to develop mathematical models for different cases of the problem. In turn, the optimal solutions through a model for small or moderate size problems may be used for benchmarking purpose while developing heuristics.
- Since, the minimization of the total earliness/tardiness under non-preemptive case is a challenging measure magnifying the complexity of the problem, use of any exact algorithms like integer programming, branch and bound technique, etc will take too much computational time. So, the researchers should concentrate on the development of meta-heuristics to obtain near-optimal/global optimum solutions for this scheduling problem.
- The minimization of the maximum lateness under the non-preemptive case under parallel machines environment makes the problem combinatorial in nature. Hence, the researchers may concentrate on the development of meta-heuristics for this problem. If one uses simulated annealing, an improved version of EDD rule may be designed as the seed generation algorithm. Under the preemptive case of this problem, only few researchers have concentrated on this problem. The researchers may extend the contributions made in the non-preemptive case to this problem.
- For the problems with non-preemptive jobs to minimize number of tardy jobs, the researchers may use meta-heuristics to obtain global optimal solution. Further, the seed generation algorithm for the simulated annealing algorithm may be designed using an improved version of Hodgson's algorithm. For the problems with preemptive case, researchers may aim to develop algorithms which yield better competitive ratio.

In different cases of the online scheduling problems, the directions for future are as listed below.

- For the problems of scheduling with non-preemptive as well as preemptive jobs to minimize makespan, the researchers may aim to develop algorithms which yield better competitive ratio.
- For the problem of scheduling with preemptive jobs to minimize sum of tardiness/earliness, the researchers should concentrate on the development of meta-heuristics to obtain near-optimal/ global optimum solutions.
- For the machine covering problem with non-preemptive as well as preemptive jobs, the researchers may aim to develop algorithms which yield better competitive ratio.

6. Acknowledgements

We sincerely thank the anonymous referees for their constructive suggestions, which helped us to improve the paper.

7. References

[1] R. Panneerselvam, "Production and Operations Management," 2nd Edition, Prentice-Hall of India, New Delhi,

2005.

[2] E. G. Jr. Coffman and R. L. Graham, "Optimal Scheduling for Two-Processor Systems," *Acta Informatica*, Vol. 1, No. 3, 1972, pp. 200-213.

[3] E. Horowiz and S. Sahni, "Exact and Approximate Algorithms for Scheduling Nonidentical Processors," *Journal of the ACM*, Vol. 23, No. 2, 1976, pp. 317-327.

[4] O. H. Ibarra and C. E. Kim, "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors," *Journal of the ACM*, Vol. 24, No. 2, 1977, pp. 280-289.

[5] D. Prabuddha and E. M. Thomas, "Scheduling to Minimize Makespan on Unequal Parallel Processors," *Decision Sciences*, Vol. 11, No. 4, 1980, pp. 586-602.

[6] R. L. Bulfin, and R. G. Parker, "Scheduling Jobs on Two Facilities to Minimize Makespan," *Management Science*, Vol. 26, No. 2, 1980, pp. 202-214.

[7] D. K. Friesen and M. A. Langston, "Bounds for Multifit Scheduling on Uniform Processors," *SIAM Journal on Computing*, Vol. 12, No. 1, 1983, pp. 60-70.

[8] R. L. Graham, "Bounds for Multiprocessing Timing Anomalies," *SIAM Journal of Applied Mathematics*, Vol. 17, No. 2, 1969, pp. 416-429.

[9] D. Gregory, "Scheduling Independent Tasks on Uniform Processors," *SIAM Journal of Computing*, Vol. 13, No. 4, 1984, pp. 705-716.

[10] D.K. Friesen, "Tighter Bounds for LPT Scheduling on Uniform Processors," *SIAM Journal on Computing*, Vol. 16, No. 3, 1987, pp. 554-560.

[11] D. S. Hochbaum and D. B. Shmoys, "A Polynomial Approximation Scheme for Scheduling on Uniform Processors: Using the Dual Approximation Approach," *SIAM Journal on Computing*, Vol. 17, No. 3, 1988, pp. 539-551.

[12] B. Chen, "Parametric Bounds for LPT Scheduling on Uniform Processors," *Acta Mathematicae Applicateae*, Sinica, Vol. 7, No. 1, 1991, pp. 67-73.

[13] P. Mireault, J. B. Orlin and R. V. Vohra, "A Parametric Worst Case Analysis of the LPT Heuristic for Two Uniform Machines," *Operations Research*, Vol. 45, No. 1, 1997, pp. 116-125.

[14] R. E. Burkard and Y. He, "A Note on MULTIFIT Scheduling for Uniform Machines," *Computing*, Vol. 61, No. 3, 1998, pp. 277-283.

[15] M. Y. Kovalyov and Y. M. Shafransky, "Uniform Machine Scheduling of Unit-Time Jobs Subject to Resource Constraints," *Discrete Applied Mathematics*, Vol. 84, No. 1-3, 1998, pp. 253-257.

[16] R. E. Burkard, Y. He and H. Kellerer, "A Linear Compound Algorithm for Uniform Machine Scheduling," *Computing*, Vol. 61, No. 1, 1998, pp. 1-9.

[17] R. Panneerselvam and S. Kanagalingam, "Modelling Parallel Processors with Different Processing Speeds of Single Machine Scheduling Problem to Minimize Makespan," *Industrial Engineering Journal*, Vol. 17, No. 6, 1998, pp. 16-19.

[18] R. Panneerselvam and S. Kanagalingam, "Simple Heuristic for Single Machine Scheduling Problem with Two Parallel Processors Having Varying Speeds to Minimize Makespan," *Industrial Engineering Journal*, Vol. 18, No. 6, 1999, pp. 2-8.

[19] C. Chekuri and M. Bender, "An Efficient Approximation Algorithm for Minimizing Makespan on Uniformly Related Machines," *Proceedings of IPCO*'99, *LNCS*, 1999, pp. 383-393.

[20] G. J. Woeginger, "A Comment on Scheduling on Uniform Machines under Chain-Type Precedence Constraints," *Operations Research Letters*, Vol. 26, No. 3, 2000, pp. 107-109.

[21] F. A. Chudak and D. B. Shmoys, "An Efficient Approximation Algorithm for Minimizing Makespan on Uniformly Related Machines," *Proceedings of the* 8*th Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, 1997, pp. 581-590.

[22] J. Jaffe, "Efficient Scheduling of Tasks without Full Use of Processor Resources," *Theoretical Computer Science*, Vol. 26, No. 1, 1980, pp. 22-35.

[23] C. Chekuri and M. Bender, "An Efficient Approximation Algorithm for Minimizing Makespan on Uniformly Related Machines," *Journal of Algorithms*, Vol. 41, No. 2, 2001, 212-224.

[24] C. J. Liao and C.-H. Lin, "Makespan Minimization for Two Uniform Parallel Machines," *International Journal of Production Economics*, Vol. 84, No. 2, 2003, pp. 205-213.

[25] L. Epstein and J. Sgall, "Approximation Schemes for Scheduling on Uniformly Related and Identical Parallel Machines," *Proceedings of* 7*th Annual European Symposium on Algorithms*, 1999, pp. 151-162.

[26] L. Epstein and J. Sgall, "Approximation Schemes for Scheduling on Uniformly Related and Identical Parallel Machines," *Algorithmica*, Vol. 39, No. 1, 2004, pp. 43-57.

[27] C. Koulamas and G. J. Kyparisis, "Makespan Minimization on Uniform Parallel Machines with Release Times," *European Journal of Operational Research*, Vol. 157, No. 1, 2004, pp. 262-266.

[28] A. Agarwal, S. Colak, V. S. Jacob and H. Pirkul, "Heuristics and Augmented Neural Networks for Task Scheduling with Non-Identical Machines," *European Journal of Operational Research*, Vol. 175, No. 1, 2006, pp. 296-317.

[29] C.-H. Lin and C.-J. Liao, "Makespan Minimization for Multiple Uniform Machines," *Computers & Industrial Engineering*, Vol. 54, No. 4, 2008, pp. 983-992.

[30] T. Kis and R. Kapolnai, "Approximation and Auctions for Scheduling Batches on Related Machines," *Operations Research Letters*, Vol. 35, No. 1, 2007, pp. 61-68.

[31] S. T. McCormic and M. inedo, "Scheduling n Independent Jobs on m Uniform Machines with both Flow Time and Makespan Objectives: A Parametric Analysis," *ORSA Journal of Computing*, Vol. 7, No. 1, 1995, pp. 63-77.

[32] C. U. Martel, "A parallel Algorithm for Preemptive Scheduling of Uniform Machines," *Journal of Parallel*

and Distributed Computing, Vol. 5, No. 6, 1988, pp. 700-715.

[33] T. Gonzalez and S. Sahni, "Preemptive Scheduling of Uniform Processor Systems," *Journal of Association of Computing Machinery*, Vol. 25, No, 1, 1978, pp. 92-101.

[34] H. Shachnai, T. Tamir and G. J. Woeginger, "Minimizing Makespan and Preemption Costs on a System of Uniform Machines," *Proceedings of the* 10*th European Symposium on Algorithms*, 2002, pp. 859-871.

[35] D. G. Pandelis, "Optimal Preemptive Scheduling on Uniform Machines with Discounted Flow Time Objectives," *European Journal of Operational Research*, Vol. 177, No. 1, 2007, pp. 630-637.

[36] M. I. Dessouky and R. L. Marcellus, "Scheduling Identical Jobs on Uniform Parallel Machines with Random Processing Times," *Computers and Industrial Engineering*, Vol. 35, No. 1-2, 1998, pp. 109-112.

[37] M. Azizoglu and O. Kirca, "On the Minimization of Total Weighted Flow Time with Identical and Uniform Parallel Machines," *European Journal of Operational Research*, Vol. 113, No. 2, 1999, pp. 91-100.

[38] Z.-L. Chen and W. B. Powell, "Solving Parallel Machine Scheduling Problems by Column Generation," *INFORMS Journal on Computing*, Vol. 11, 1999, pp. 78-94.

[39] T. F. Gonzalez, Y. T. L. Joseph and M. Pinedo, "Minimizing Total Completion Time on Uniform Machines with Deadline Constraints," *ACM Transactions on Algorithms*, Vol. 2, No. 1, 2006, pp. 95-115.

[40] J. Y. T. Leung, H. B. Li, M. Pinedo and J. W. Zhang, "Minimizing Total Weighted Completion Time when Scheduling Orders in a Flexible Environment with Uniform Machines," *Information Processing Letters*, Vol. 103, No. 3, 2007, pp. 119-129.

[41] K.-S. Lin, "Scheduling with Parallel Machines to Minimize Total Job Tardiness," *Engineering Costs and Production Economics*, Vol. 5, No. 3-4, 1981, pp. 289-296.

[42] A. Guinet, "Scheduling Independent Jobs on Uniform Parallel Machines to Minimize Tardiness Criteria," *Journal of Intelligent Manufacturing*, Vol. 6, No. 2, 1995, pp. 95- 103.

[43] M. Azizoglu and O. Kirca, "Tardiness Minimization on Parallel Machines," *International Journal of Production Economics*, Vol. 55, No. 2, 1998, pp. 163-168.

[44] I. Naofumi, T. Shunji and A. Mitsuhiko, "Minimizing Total Tardiness on Uniform Parallel Machines under Human Resource Constraint," *Proceedings of the Annual Conference of the Institute of Systems*, *Control and Information Engineers*, Vol. 47, 2003, pp. 419-420,.

[45] V. A. Armentano and M. F. de F. Filho, "Minimizing Total Tardiness in Parallel Machine Scheduling with Setup Times: An Adaptive Momory-Based GRASP Approach," *European Journal of Operations Research*, Vol. 183, No. 1, 2007, pp. 100-114.

[46] N. H. Tuong, A. Soukhal and J.-C. Billaut, "Uniform Parallel Machine Scheduling Problem with a Common Due Date to Minimize Total Weighted Tardiness," *MAPSP Conference*, 2007.

[47] G. Mosheiov and A. Sarig, "Due-Date Assignment on Uniform Machines," *European Journal of Operational Research*, Vol. 193, No. 1, 2009, pp. 49-58.

[48] A. Federgruen and H. Groenevelt, "Preemptive Scheduling of Uniform Machines by Ordinary Network Flow Techniques," *Management Science*, Vol. 32, No. 3, 1986, pp. 341-349.

[49] M. M. Dessouky, "Scheduling Identical Jobs with Unequal Ready Times on Uniform Parallel Machines to Minimize the Maximum Lateness," *Computers & Industrial Engineering*, Vol. 34, No. 4, 1998, pp. 793-806.

[50] D. Chhajed, "A Note on Minimizing the Maximum Deviation of Job Completion Time about a Common Due-Date," *Applied Mathematics Letters*, Vol. 1, No. 2, 1988, pp. 161-163.

[51] F. A. Chudak and D. B. Shmoys, "Approximation Algorithm for Precedence-Constrained Scheduling Problems on Parallel Machines that Run at Different Speeds," *Journal of Algorithms*, Vol. 20, No. 2, 1999, pp. 323-343.

[52] C. Koulamas and G. J. Kyparisis, "Scheduling on Uniform Parallel Machines to Minimize Maximum Lateness," *Operations Research Letters*, Vol. 26, No. 3, 2000, pp. 175-179.

[53] M. Drozdowski, J. Blazewicz, P. Formanowicz, W. Kubiak and G. Schmidt, "Scheduling Preemptable Tasks on Uniform Processors with Limited Availability for Maximum Lateness Criterion," 7*th International Workshop on Project Management Scheduling*, Osnabrück, 2000, pp. 118-120.

[54] A. J. Ruize-Torres, F. J. Lopez and J. C. Ho, "Scheduling Uniform Parallel Machines Subject to a Secondary Resource to Minimize the Number of Tardy Jobs," *European Journal of Operational Research*, Vol. 179, No. 2, 2007, pp. 302-315.

[55] E. L. Lawler and C. U. Martel, "Preemptive Scheduling of Two Uniform Machines to Minimize the Number of Late Jobs," *Operations Research*, Vol. 37, No. 2, 1989, pp. 314-318.

[56] J. Wein and D. P. Williamson, "On-Line Scheduling of Parallel Machines," *Technical Report*, Institute of Technology, Cambridge Lab for Computer Science, Massachusetts, 1990, p. 18.

[57] L. Epstein and J. Sgall, "A Lower Bound for On-Line Scheduling on Uniformly Related Machines," *Operations Research Letters*, Vol. 26, No. 1, 2000, pp. 17-22.

[58] Z. Tan and Y. He, "Semi-On-Line Scheduling with Ordinal Data on Two Uniform Machines," *Operations Research Letters*, Vol. 28, No. 5, 2001, pp. 221-231.

[59] S. Kontogiannis, "Multiple-Choice Assignments on Uniformly Related Machines," *ALCOM-FT Technical Report Series*, 2002, pp. 2-39.

[60] L. Epstein and L. M. Favrholdt, "Optimal Non-Preemptive Semi-Online Scheduling on Two Related Machines," *Journal of Algorithms*, Vol. 57, No. 1, 2005, pp. 49-73.

[61] T. C. E. Cheng, C. T. Ng and V. Kotov, "A New Algorithm for Online Uniform-Machine Scheduling to Minimize the Makespan," *Information Processing Letters*, Vol. 99, No. 3, 2006, pp. 102-105.

[62] S. Sahni and Y. Cho, "Scheduling Independent Tasks with Due Times on a Uniform Processor System," *Journal of the ACM*, Vol. 27, No. 3, 1980, pp. 550-563.

[63] R. Li and L. Shi, "An On-Line Algorithm for Some Uniform Processor Scheduling," *SIAM Journal of Computer*, Vol. 27, No. 2, 1998, pp. 414-422.

[64] E. Angelelli, M. G. Speranza and Z. Tuza, "Semi-Online Scheduling on Two Uniform Processors," *Theoretical Computer Science*, Vol. 393, No. 1-3, 2008, pp. 211-219.

[65] J. Wen, and D. Du, "Preemptive On-Line Scheduling for Two Uniform Processors," *Operations Research Letters*, Vol. 23, No. 3, 1998, pp. 113-116.

[66] A. Vestjens, "Scheduling Uniform Machines On-Line Requires Nondecreasing Speed Ratios," *Mathematical Programming*, Vol. 82, No. 2, 1998, pp. 225-234.

[67] L. Epstein, J. Noga, S. S. Seiden, J. Sgall and G. J. Woeginger, "Randomized On-Line Scheduling on Two Uniform Machines," *Proceedings of the* 10*th Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, 1999, pp. 317-326.

[68] L. Epstein, J. Noga, S. S. Seiden, J. Sgall and G. J. Woeginger, "Randomized On-Line Scheduling on Two Uniform Machines," *Journal of Scheduling*, Vol. 4, No. 2, 2001, pp. 71-92.

[69] L. Epstein, "Optimal Preemptive On-Line Scheduling on Uniform Processors with Non-Decreasing Speed Ratios," *Operations Research Letters*, Vol. 29, No. 2, 2001(a), pp. 93-98.

[70] L. Epstein, "Optimal Preemptive Scheduling on Uniform Processors with Non-Decreasing Speed Ratios," *Proceedings of* 18*th STACS*, Dresden, 2001(b), pp. 230-237.

[71] L. Epstein and L. M. Favrholdt, "Optimal Preemptive Semi-Online Scheduling to Minimize Makespan on Two Related Machines," *Operations Research Letters*, Vol. 30, No. 4, 2002, pp. 269-275.

[72] D. L. Du, "Optimal Preemptive Semi-Online Scheduling on Two Uniform Processors," *Information Processing letters*, Vol. 92, No. 5, 2004, pp. 219-223.

[73] N. Balakrishnan, J. J. Kanet and S. Sridharan, "Early/ Tardy Scheduling with Sequence Dependent Setups on Uniform Parallel Machines," *Computers & Operations Research*, Vol. 26, No. 2, 1999, pp. 127-141.

[74] U. Bilge, F. Kirac, M. Kurtulan and P. Pekgun, "A Tabu Search Algorithm for Parallel Machine Total Tardiness Problem," *Computers and Operations Research*, Vol. 31, No. 3, 2004, pp. 397-414.

[75] Y. Azar and L. Epstein, "On-Line Machine Covering," *Proceedings of* 5*th ESA conference*, Graz, 1997, pp. 23-36.

[76] Y. Azar and L. Epstein, "On-Line Machine Covering, *Journal of Scheduling*," Vol. 1, No. 2, 1998(a), pp. 67-77.

[77] Y. Azar and L. Epstein, "Approximation Schemes for Covering and Scheduling on Related Machines," *Proceedings of the* 1*st Workshop on Approximation Algorithms for Combinatorial Optimization Problems* (*APPROX*'98), *in Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Vol. 1444, 1998(b), pp. 39-47.

[78] Z. Tan, Y. He and L. Epstein, "Optimal On-Line Algorithms for the Uniform Machine Scheduling Problem with Ordinal Data," *Information and Computation*, Vol. 196, No. 1, 2005, pp. 57-70.

[79] Y. He and Y. Jiang, "Optimal Semi-Online Preemptive Algorithms for Machine Covering on Two Uniform Machines," *Theoretical Computer Science*, Vol. 339, No. 2-3, 2005, pp. 293-314.

[80] B. Alidaee and A. Ahmadian, "Two Parallel Machine Sequencing Problems Involving Controllable Job Processing Times," *European Journal of Operational Research*, Vol. 70, No. 3, 1993, pp. 335-341.

[81] B. Alidaee and A. Ahmadian, "Scheduling on a Single Processor with Variable Speed," *Information Processing Letters*, Vol. 60, No. 4, 1996, pp. 189-193.

[82] S. Vahid, N. Mohmoud and K. Mohsen, "Deadline Scheduling with Processor Affinity and Feasibility Check on Uniform Machines," *Computer and Information Technology*, Vol. 16-19, 2007, pp. 793-798.

[83] O. B. Bekki and M. Azizoglu, "Operational Fixed Interval Scheduling Problem on Uniform Parallel Machines," *International Journal of Production Economics*, Vol. 112, No. 2, 2008, pp. 756-768.

[84] Y. He and X. Min, "On-Line Uniform Machine Scheduling with Rejection," *Computing*, Vol. 65, No. 1, 2000, pp. 1-12.

[85] J. Blazewicz, "Minimizing Mean Weighted Information Loss with Preemptable Tasks and Parallel Processors," *Technology and Science of Informatics*, Vol. 3, No. 6, 1984, pp. 415-420.

[86] J. Blazewicz and G. Finke, "Minimizing Mean Weighted Execution Time Loss on Identical and Uniform Processors," *Information processing Letters*, Vol. 24, No. 4, 1987, pp. 259-263.

[87] H. Ishi, C. Martel, T. Masuda and T. Nishida, "A Generalized Uniform Processor System," *Operations Research*, Vol. 33, No. 2, 1985, pp. 346-362.

[88] J. Blazewicz and P. Bouvry, F. Guinand and D. Trystram, "Scheduling Complete Intrees on Two Uniform Processors with Communication Delays," *Information Processing Letters*, Vol. 58, No. 5, 1996, pp. 255-263.

[89] N. V. Shakhlevich and V. A. Strusevich, "Preemptive Scheduling on Uniform Parallel Machines with Controllable Job Processing Times," *Algorithmica*, Vol. 51, No. 4, 2007, pp. 451-473.

[90] B. Kubiak, B. Penz and D. Trystram, "Scheduling Chains on Uniform Processors with Communication Delays," *Journal of Scheduling*, Vol. 5, No. 6, 2002, pp. 459-476.

[91] S. Baruah, "Scheduling Periodic Tasks on Uniform Multiprocessors," *Information Processing Letters*, Vol. 80, No. 2, 2001, pp. 97-104.

[92] S. Baruah and J. Goossens, "Rate-Monotonic Scheduling on Uniform Multiprocessors," *IEEE Transactions on Computers*, Vol. 57, No. 7, 2003, pp. 966-970.

Intelligent Information Management, **2010, 2, 475-482**
doi:10.4236/iim.2010.28057 Published Online August 2010 (http://www.SciRP.org/journal/iim)

Information Aggregation of Group Decision-Making in Emergency Events*

Kefan Xie, Qian Wu, Gang Chen, Chao Ji
Management school, Wuhan University of Technology, Wuhan, China
E-mail: *xkf@whut.edu.cn, shashavio@sohu.com,*{*cg224*, *jichaowuqiao*2003}@163.*com*
Received May 14, 2010; *revised June* 22, 2010; *accepted July* 19, 2010

Abstract

Information is a key factor in emergency management, which helps decision makers to make effective decisions. In this paper, aiming at clarifying the information aggregation laws, and according to the characteristic of emergency information, information relative entropy is applied in the information aggregation to establish the information aggregation model of emergency group decision-making. The analysis shows that support and credibility of decision rule are the two factors in information aggregation. The results of four emergency decision-making groups in case study support the analysis in the paper.

Keywords: Emergency, Group Decision-Making, Information Aggregation, Relative Entropy

1. Introduction

As an important part of the emergency management, decision-making system for emergency response is one of the most important research areas. Emergency decision-making is a group decision-making process. Emergency managers make decision in the situation of constraint time, incomplete and inaccurate information. They brainstorm and coordinate the interests of multiple decision-makers, in order to handle variety of emergency for achieving timely, effective and smooth goal. Group decision-making behavior in emergency management attracts more and more attention of the whole society.

When making decisions on complex issues, in order to scientific decision-making and avoid individual subjective judgments, choice and preference impact on the decision results, emergency managers should analysis, judgment and decision by adopting integrated experience and wisdom of expert groups and method of group decision-making. Information aggregation has a significant effect in the process of group decision-making, which becomes a hot topic on the emergency management research.

According to information aggregation research, American scholar Yager [1] proposes Ordered Weighted Averaging Operator. Bodily [2] gives information aggregation method based on principal process. Brock [3] puts forward information aggregation method based on Nash-Harsanyi negotiation model. Krzyszto and Duck [4] propose the information aggregation on the basis of group value judgment. Chen Dongfeng *et al*. [5] make an uniformization of four common preference information given by decision makers, which are real value, interval values, language phrases and intuitionistic fuzzy value, based on conversion of formula between deferent fuzzy preference information. Li Bingjun and Liu Sifeng [6] construct a defining-number judgment matrix of certain credibility which is equivalent to group information of interval number reciprocal judgment matrix based on-set-valued statistics principle. Wu Jiang and Huang Dengshi [7] construct the relation functions based on four interval number preference information, namely interval number preference orderings, interval number utility values, interval number complementary judgment matrices and interval number reciprocal judgment matrices, which uniforms different uncertain preference information. Zhou Shizhong *et al*. [8] analysis characteristics of four kinds uncertain preference and propose a goal programming model to aggregate the group preference. Zhu Jianjun [9] studies the group aggregation approach of interval number reciprocal comparison matrix and interval number complementary comparison matrix by using UOWA method. Feng Xiangqian *et al*. [10] construct an aggregating of interval number judgment matrixes with maximum satisfaction. Guiwu Wei [11] proposes intui- tionistic fuzzy ordered weighted geometric operator and interval-valued intuitionistic fuzzy ordered weighted geometric operator to study multiple attribute group decision making issue. H. J. Zimmermann and P. Zysno [12] representing the criteria (subjective categories)

*Project No. 90924010 supported by NSFC.

by fuzzy sets which is possible to use aggregation operators as models for the amalgamation of those criteria. Xie Kefan and Chen Gang [13] research on the entrepreneurial Team's Risk Decision-Making Process Based on Group Learning.

In short, most scholars have focused on the various structural preference aggregation methods. However, the information aggregation has been little studied in emergency cases.

2. Information Aggregation Mechanism of Group Decision-Making in Emergency Management

Whether emergency decision making is a scientific decision, depends on whether there is a consistence between quality of information and preference information of expert decision-making. Quality of decision-making information is critical to the ways of group decision-making. Quality of decision-making information refers to the authenticity and credibility of the information. Information aggregation refers to the process of gathering, compressing, screening, refining and integrating to decision-making information, which includes aggregation to decision-making information and aggregation to expert preference information. Aggregation to decision-making information composes of information awareness and screening.

Information aggregation of group decision-making in emergency management has two characteristics as follow: 1) Emergency decision-making actors are large groups, which are collections of various groups. Therefore, information aggregation among groups should be taken into account when study on group decision-making; 2) There are two factors to be considered in group decision-making: statutory decision-making power and expert decision-making power. Statutory decision-making power is mainly used in emergency decision-making mechanism, and expert decision-making power is mainly used in credibility and support of expert decision-making.

2.1. Information Perception

After the outbreak of unexpected events, the interests of all groups would like to obtain the original information as much as possible to make a scientific decision-making to respond to emergency in order to protect their own interests, reduce the loss of harm. Information perception is the first act of information aggregation stage, which refers to the process of getting information by interest groups through various channels and means. The amount of information increases exponentially and generates information associated with variation and deviation after the outbreak of unexpected events, which may bring a lot of information garbage. The information in this stage is incompleteness, and transmission route owns of characteristics of plurality, asymmetry and mutagenicity etc. In this stage, information aggregation of interest groups reflects in the "quantity", which means that obtaining information the more the better and as soon as possible.

2.2. Information Screening

Information screening is the second stage of information aggregation. Information screening refers to information process of screening, identification, verification of its authenticity, validity to the first stage perceptible information through various technical means which based on laws and rigorous, serious, scientific workflow. Information screening includes two parts of information process control and information content control. Information screening is the screening authenticity of information in essence in order to ensure the validity of the information. At this stage, each group pays more attention to false, unrealistic and not useful information. After screening the authenticity and validity of information greatly increased and becomes an important basis for group decision-making. Information screening methods can be summarized into two parts. On the one hand it can be judged by perceptions and experiences of decision makers; on the other hand, it can be judged by analyzing, summarizing and organizing the various historical data of past emergency events to find all kinds of information regularity. Information aggregation at this stage of information gathering reflected in the "quality", which means that the more real the obtained information, the better the efficiency will be.

2.3. Preference Information Aggregation

In the process of emergency decision-making, in order to ensure scientific decision-making, group decision-making is usually required, that is to focus and synthesize the experience and knowledge of experts, to analyze the problem, judging and decision-making. Group decision making is the essence of decision-makers in each assembly group preferences for the group of decision makers' preferences, then depending on the group's preferences on a set of programs to sort, select the group from which is the most preferred option. Decision information can be incomplete and asymmetric, as well as decision-makers may have different knowledge structure, judgment level and personal preference etc., therefore, when facing the same decision issue experts may give difference preference information of difference structure form, which need to have aggregation process to the preference information. The usual method is to do different forms of preference information consistent processing. The common decision-making preference information can be summarized into four types, interval preference order, interval utility value, interval number reciprocal comparison matrix and interval number complementary comparison matrix. Currently, the main methods to synthesis expert opinion and preference in group decision-making can be divided into two categories: Comprehensive Judgment Matrix and Comprehensive sequencing vector. Preference information aggregation is a method of decision makers to take some form of the above or any particular form of individual preferences for the group preference aggregation process. However, the However, the different structure of preference information of the process the same information to generate distortion is inevitable, how to ensure the validity of the information would be an important issue.

3. Information Relative Entropy Aggregation Model of Group Decision-Making in Emergency Events

3.1. Model Assumes

1) In the process of decision-making in emergency management, assuming that n groups decision of the emergency response form n decision table. Suppose T is decision table set $T_j = \left(U_j, C \cup \{d\}\right)$, $j = 1, 2, \cdots, n$, $T = \{T_1, T_2, \cdots, T_n\}$, where decision table T_j of certain group j corresponds to m_j decision-making rule, any of which $Ru_{ij} = \left\{rule_{ij} = des\left([x]_{C_{ij}}\right) \to des\left([x]_{d_{ij}}\right)\right\}$ has support SD_{ij} and credibility CD_{ij};

Then $SD_{ij} = \left|[x]_{C_{ij} \cap d_{ij}}\right| / \left|U_j\right|$ (1)

$$CD_{ij} = \left|[x]_{C_{ij} \cap d_{ij}}\right| / \left|[x]_{C_{ij}}\right| \quad (2)$$

2) In the process of decision-making in emergency management, there are communication and games among all the groups, which can aggregate m decision rule of group, where any decision-making rule Ru_i of group has support SD_i and credibility CD_i.

3) In the process of decision-making in emergency management, assuming that any group weight is ω_j, then, group decision-making weight distribution is

$$\omega = \left\{\omega_1, \omega_2, \cdots, \omega_j, \cdots, \omega_n\right\}.$$

Theorem 1: The number of group decision rule m is greater than or equal to the number of any group decision-making rule m_j in group, that is $m_j \le m$.

Certification: Assuming that m_j decision rules of group j can form decision-making system of $S_j = \left(U_j, C \cup \{d\}\right)$. Therefore, in the decision-making system, if $\exists x_{1j}, x_{2j} \in U_j$ as well as $x_{2j} \in \left[x_{1j}\right]_C$, then it is inevitable that $x_{2j} \notin \left[x_{1j}\right]_d$, that is the decision system is incompatible, and there are m_j decision rules to meet the conditions. Assume that group decision-making system is $S = \left(U, C \cup \{d\}\right)$, then $S_j \subseteq S$ is inevitable. Assuming that $S - S_j$ is one of decision-making system, if $\exists x \in U - U_j$, $\exists x_j \in U_j$ as well as $x \in \left[x_j\right]_C$, $x \notin \left[x_j\right]_d$, then group decision-making system $S = \left(U, C \cup \{d\}\right)$ is incompatible and there are more than $m_j + 1$ decision rules to meet the condition, that is $m = m_j + 1 > m_j$; If $\forall x \in U - U_j$, $\exists x_j \in U_j$ as well as $x \in \left[x_j\right]_C, x \in \left[x_j\right]_d$, then decision-making system $S - S_j$ and $S_j = \left(U_j, C \cup \{d\}\right)$ is fully compatible, but there are at least m_j decision-making rules, that is $m = m_j$, so $m_j \le m$.

3.2. Support and Credibility of Decision Rule

For $m_j \le m$, group j in the decision-making gruops may not has the whole decision-making rule which groups possesses, that is parts of the decision rule are not owned by group j, which is called loss of decision-making rule. If there is a decision-making rule $Ru_i = \left\{rule_i = des\left([x]_{C_i}\right) \to des\left([x]_{d_i}\right)\right\}$ in group, any one decision-making rule $Ru_{ij} = \left\{rule_{ij} = des\left([x]_{C_{ij}}\right) \to des\left([x]_{d_{ij}}\right)\right\}$ of certain one group j does not satisfy the conditions: $des\left([x]_{C_{ij}}\right) = des\left([x]_{C_i}\right)$ and $des\left([x]_{d_{ij}}\right) = des\left([x]_{d_i}\right)$, then, decision-making rule Ru_i is an loss decision rule for group j. According to the formula (1) and (2), the support and credibility of decision rule are $SD_{ij} = 0$ and $CD_{ij} = 0$. However, certain support and credibility of decision-making rules is essentially not Zero. For example, when $des\left([x]_{C_{ij}}\right) \cap des\left([x]_{C_i}\right) \ne \varnothing$

and $des\left([x]_{d_{ij}}\right)\cap des\left([x]_{d_i}\right)=\varnothing$, the support and credibility of decision rule losses do not always equals Zero. Therefore, the amount of support and credibility of decision-making rule loss should depend upon the state of affair.

1) If group j has one decision-making rule Ru_{ij} which can contain loss decision-making rule Ru_i , and satisfies the condition of $des\left([x]_{C_{i'j}}\right)\supset des\left([x]_{C_i}\right)$ and $des\left([x]_{d_{ij}}\right)=des\left([x]_{d_i}\right)$, then the support and credibility of loss decision-making rule Ru_i is in accordance with the support and credibility of decision-making rule $Ru_{i'j}$, that is $SD_{ij}=SD_{i'j}$, $CD_{ij}=CD_{i'j}$. For the purpose of group j , if there are m' decision-making rule $Ru_{i'j}$ which can contain the loss decision-making rule Ru_i , then the support and credibility of loss decision-making rule Ru_i are as follow:

$$SD_{ij}=\sum_{i'=1}^{m'}SD_{i'j}\ ,\quad CD_{ij}=\sum_{i'=1}^{m'}CD_{i'j} \tag{3}$$

2) If group j has one decision-making rule Ru_{ij} which satisfies the condition of $des\left([x]_{C_{i'j}}\right)=des\left([x]_{C_i}\right)$ and $des\left([x]_{d_{ij}}\right)\cap des\left([x]_{d_i}\right)=\varnothing$, and the decision-making rule $Ru_{i'j}$ is restrict and incompatible to the loss decision-making rule Ru_i , then the support and credibility of loss decision-making rule Ru_i are:

$$SD_{ij}=0\ ,\quad CD_{ij}=0 \tag{4}$$

3) If group j has no decision-making rule Ru_{ij} which satisfies the condition of $des\left([x]_{C_{i'j}}\right)\supseteq des\left([x]_{C_i}\right)$, $des\left([x]_{d_{ij}}\right)=des\left([x]_{d_i}\right)$ or $des\left([x]_{d_{ij}}\right)\cap des\left([x]_{d_i}\right)=\varnothing$, then the loss decision-making rule is not relevant to the Ru_{ij} , that is in the condition of $[x]_{C_{i'j}}$, group j has neither decision nor strategy, which is regarded as abstention in decision-making process. Therefore, the support and credibility of those kinds of loss decision-making rules to group j , are in accordance with emergency decision-making mechanism. Assuming to the above, the support and credibility of decision-making strategy to abstention in emergency decision-making process are meeting the condition as follow:

$$SD_{ij}=0\ ,\quad CD_{ij}=0 \tag{5}$$

3.3. Decision-Making Weight

In the emergency response decision-making process, a number of factors influence the decision-making groups in decision weights, in which legal decision-making weight and expert decision-making weight are the two main factors. The mechanism of group decision-making to the legal decision-making weight is related with emergency decision-making mechanism; the mechanism of group decision-making to the expert decision-making weight is related with the credibility of expert groups. Those two weight response the emergency decision-making process, and demonstrate the laws of the following points:

1) The strategies of expert decision-making groups have more credibility and less ambiguous decision. The expert credibility enhances the decision-making weight of expert decision-making groups. The less ambiguous decision shows that there is less flexibility but more rigid in the decision-making, resulting in the more firm weight.

2) No matter how big groups of legal decision-making power, when it has ambiguous type of strategy and more decision-making room, legal decision-making weight will be greatly reduced. In order to meet decision-making target, decision makers may randomly adjust their strategy in order to influence their decision-making weight.

Therefore, in the process of emergency decision-making, group decision-making weight is in accordance with credibility of decision-making strategy. The more the credibility, the higher the decision-making weight, the less the credibility, the lower the decision-making weight. Hence, assuming that decision-making weight ω_{ij} of group j in decision strategy Ru_{ij} is proportionate to credibility CD_{ij} of the decision-making rule. Decision-making weight ω_{ij} of group j in decision strategy Ru_{ij} can be represented by the relative credibility of decision-making rule Ru_{ij} to group j :

$$\omega_{ij}=e_{ij}\Big/\sum_{j=1}^{n}e_{ij}=CD_{ij}\Big/\sum_{j=1}^{n}CD_{ij} \tag{6}$$

Then, the decision-making weight expectation of group j in emergency decision-making process is:

$$e_j = \sum_{i=1}^{m} \omega_{ij} \cdot SD_{ij} \tag{7}$$

After normalization of decision-making weight to each decision-making group in the group, it is available to get decision weights of the groups:

$$\omega = (\omega_1, \omega_2, \cdots, \omega_n), \text{ and } \omega_j = \frac{e_j}{\sum_{j=1}^{n} e_j} \tag{8}$$

3.4. Model of Information Relative Entropy Aggregation

In process of emergency group decision-making, in order to obtain more efficient decision strategy, every group in the group want to choose the minimized deviation values of decision-making rules according to group decision-making rule. Therefore, the support SD_i of group decision-making rule Ru_i can be computed by relative entropy aggregation model as Equation (9).

Where ω_j in (9) is the decision-making weight of group j, which can be computed by (8).

With nonlinear programming (9), local optimal solution $SD^* = (SD_1^*, SD_2^*, \cdots, SD_m^*)^T$ can be calculated, where $SD_i^* = \prod_{j=1}^{n} (b_{ij})^{\omega_j} \Big/ \sum_{i=1}^{m} \prod_{j=1}^{n} (b_{ij})^{\omega_j}, i = 1, 2, \cdots, m$, and $b_{ij} = SD_{ij} \Big/ \sum_{i=1}^{m} SD_{ij}$.

In the process of emergency group decision-making, $Ru_i = \{rule_i = des([x]_{C_i}) \to des([x]_{d_i})\}$ is one of group decision-making rules, where group credibility is CD_i.

If $\exists m^*$ decision-making rule meet the conditions as follows: $Ru_i^* = \{rule_i^* = des([x]_{C_i^*}) \to des([x]_{d_i^*})\}$ and $des([x]_{C_i^*}) = des([x]_{C_i}), des([x]_{d_i^*}) \neq des([x]_{d_i})$, and assuming that credibility of any decision-making rule Ru_i^* to group j is CD_{i^*j}. Then, credibility CD_i of any decision-making rule Ru_i^* can be calculated by relative entropy aggregation model (10) as follows:

Where ω_j in (10) is the decision-making weight of group j, which can be computed by (8).

With nonlinear programming (10), local optimal solution $CD^* = (CD_1^*, CD_2^*, \cdots, CD_{m^*}^*)^T$ can be calculated, where $CD_i^* = \prod_{j=1}^{n} (c_{ij})^{\omega_j} \Big/ \sum_{i=1}^{m^*} \prod_{j=1}^{n} (c_{ij})^{\omega_j}, i = 1, 2, \cdots, m^*$, and $b_{ij} = SD_{ij} \Big/ \sum_{i=1}^{m^*} SD_{ij}$.

From (9) and (10) it is known that $SD_{ij} \neq 0$ and $CD_{ij} \neq 0$ is not in conformity with the two kinds of decision-making loss situation, that is, in the situation of decision-making rule loss, support and credibility of group decision-making rule cannot be calculated by (9) and (10). To circumvent this problem, it is better to make infinitesimal positive real numbers to support and credibility of loss decision-making rule in decision-making rule loss situation as follows: $\ni \varepsilon \to 0$ and $\varepsilon \neq 0$, let $SD_{ij} = CD_{ij} = \varepsilon$.

4. Case Study

In the process of emergency decision-making, there are

$$\begin{cases} \min(SD) = \sum_{j=1}^{n} \omega_j \sum_{i=1}^{m} \left[SD_i \cdot \left(\log(SD_i) - \log\left(\frac{SD_{ij}}{\sum_{i=1}^{m} SD_{ij}} \right) \right) \right] \\ s.t. \sum_{i=1}^{m} SD_i = 1 \end{cases} \tag{9}$$

$$\begin{cases} \min(CD) = \sum_{j=1}^{n} \omega_j \sum_{i=1}^{m^*} \left[CD_i \cdot \left(\log(CD_i) - \log\left(\frac{CD_{ij}}{\sum_{i=1}^{m^*} CD_{ij}} \right) \right) \right] \\ s.t. \sum_{i=1}^{m^*} CD_i = 1 \end{cases} \tag{10}$$

Table 1. Decision-making table T_1.

Decision-making rules	style	Position	Type	Decision	CD	SD
ω_1	Y	S	D	accept	75.00%	9.09%
ω_2	Y	S	D	reject	25.00%	3.03%
ω_3	Y	J	D	accept	50.00%	9.09%
ω_4	Y	J	D	reject	50.00%	9.09%
ω_5	Y	J	X	accept	33.33%	6.06%
ω_6	Y	J	X	reject	66.67%	12.12%
ω_8	P	S	D	reject	100.00%	9.09%
ω_9	P	S	X	accept	100.00%	3.03%
ω_{11}	P	J	D	accept	100.00%	27.27%
ω_{13}	P	J	X	accept	75.00%	9.09%
ω_{14}	P	J	X	reject	25.00%	3.03%

Table 2. Decision-making table T_2.

Decision-making rules	style	Position	Type	Decision	CD	SD
ω_1	Y	S	D	accept	100.00%	20.59%
ω_3	Y	J	D	accept	71.43%	14.71%
ω_4	Y	J	D	reject	28.57%	5.88%
ω_5	Y	J	X	accept	100.00%	2.94%
ω_{10}	P	S	X	reject	100.00%	8.82%
ω_{11}	P	J	D	accept	25.00%	2.94%
ω_{12}	P	J	D	reject	75.00%	8.82%
ω_{14}	P	J	X	reject	100.00%	5.88%
ω_{15}	Y	S	X	accept	90.00%	26.47%
ω_{16}	Y	S	X	reject	10.00%	2.94%

Table 3. Decision-making table T_3.

Decision-making rules	style	Position	Type	Decision	CD	SD
ω_1	Y	S	D	accept	85.71%	14.29%
ω_2	Y	S	D	reject	14.29%	2.38%
ω_3	Y	J	D	accept	50.00%	4.76%
ω_4	Y	J	D	reject	50.00%	4.76%
ω_5	Y	J	X	accept	50.00%	7.14%
ω_6	Y	J	X	reject	50.00%	7.14%
ω_7	P	S	D	accept	25.00%	2.38%
ω_8	P	S	D	reject	75.00%	7.14%
ω_{12}	P	J	D	reject	100.00%	9.52%
ω_{13}	P	J	X	accept	85.71%	14.29%
ω_{14}	P	J	X	reject	14.29%	2.38%
ω_{15}	Y	S	X	accept	90.00%	21.43%
ω_{16}	Y	S	X	reject	10.00%	2.38%

four groups in the decision-making group, and the decision-making rules of the four groups are respectively listed in **Tables 1-4**. According to the four decision-making tables we can find that the four groups constituting a group have 15 decision-making rules, which are $\omega_1, \omega_2, \cdots, \omega_{15}$ respectively. The four groups all have decision-making loss in parts of their decision, therefore the support and credibility of four groups loss deci sion-making can be calculated by model (3),(4) and (5). According to model (6), (7) and (8), the results of decision making weight of four groups are 0.301, 0.239, 0.213, 0.247, respectively. Due to the fact that parts of support and credibility of loss decision-making are Zero, it influences the results of support and credibility of decision-making to the groups by using model (9) and (10). In order to eliminate this impact, we define that support and credibility of loss decision-making rule are alternated by 10^{-6}. The results are shown in **Table 5** as follows:

According to **Table 5**, after aggregating by relative entropy, the support and credibility of group decision-

Table 4. Decision-making table T_4.

Decision-making rules	style	Position	Type	Decision	CD	SD
ω_1	Y	S	D	accept	28.57%	5.71%
ω_2	Y	S	D	reject	71.43%	14.29%
ω_3	Y	J	D	accept	25.00%	2.86%
ω_4	Y	J	D	reject	75.00%	8.57%
ω_5	Y	J	X	accept	50.00%	5.71%
ω_6	Y	J	X	reject	50.00%	5.71%
ω_9	P	S	X	accept	16.67%	2.86%
ω_{10}	P	S	X	reject	83.33%	14.29%
ω_{12}	P	J	D	reject	100.00%	8.57%
ω_{13}	P	J	X	accept	100.00%	11.43%
ω_{15}	Y	S	X	accept	100.00%	20.00%

Table 5. The group decision-making rules.

Decision-making rules	$j=1$		$j=2$		$j=3$		$j=4$		group	
	0.301		0.239		0.213		0.247			
	CD_{i1}	SD_{i1}	CD_{i2}	SD_{i2}	CD_{i3}	SD_{i3}	CD_{i4}	SD_{i4}	CD_i	SD_i
ω_1	75.00%	9.09%	100.00%	20.59%	85.71%	14.29%	28.57%	5.71%	97.79%	33.26%
ω_2	25.00%	3.03%	**0%**	**0%**	14.29%	2.38%	71.43%	14.29%	2.21%	1.10%
ω_3	50.00%	9.09%	71.43%	14.71%	50.00%	4.76%	25.00%	2.86%	48.69%	20.48%
ω_4	50.00%	9.09%	28.57%	5.88%	50.00%	4.76%	75.00%	8.57%	51.31%	21.57%
ω_5	33.33%	6.06%	100.00%	2.94%	50.00%	7.14%	50.00%	5.71%	95.66%	15.95%
ω_6	66.67%	12.12%	**0%**	**0%**	50.00%	7.14%	50.00%	5.71%	4.34%	1.68%
ω_7	**0%**	**0%**	**0%**	**0%**	25.00%	2.38%	**0%**	**0%**	1.22%	**0**.00%
ω_8	100.00%	9.09%	**0%**	**0%**	75.00%	7.14%	**0%**	**0%**	98.78%	0.10%
ω_9	100.00%	3.03%	**0%**	**0%**	**0%**	**0%**	16.67%	2.86%	61.34%	0.09%
ω_{10}	**0%**	**0%**	100.00%	8.82%	**0%**	**0%**	83.33%	14.29%	38.66%	0.09%
ω_{11}	100.00%	27.27%	25.00%	2.94%	**0%**	**0%**	**0%**	**0%**	7.93%	0.16%
ω_{12}	**0%**	**0%**	75.00%	8.82%	100.00%	9.52%	100.00%	8.57%	92.07%	0.88%
ω_{13}	75.00%	9.09%	**0%**	**0%**	85.71%	14.29%	100.00%	11.43%	69.49%	2.12%
ω_{14}	25.00%	3.03%	100.00%	5.88%	14.29%	2.38%	**0%**	**0%**	30.51%	0.81%
ω_{15}	**0%**	**0%**	90.00%	26.47%	90.00%	21.43%	100.00%	20.00%	98.79%	1.68%
ω_{16}	**0%**	**0%**	10.00%	2.94%	10.00%	2.38%	**0%**	**0%**	1.21%	0.03%

making rule change both. The support of decision-making ω_1 is greatest, and its result is 33.26%. The reason why it support is so high is because that supports of any groups in group to the decision-making rules are high, and the groups have higher degree of recognition acceptance. However, the support of decision-making ω_7 is lowest, and its result is 0.003%. The reason why it support is so low is because that recognitions of the decision-making rule in group lacks of consensus, and it only support by group $j=3$ but its support is just 2.38%. The credibility of decision-making rule ω_8 is greatest to group and its result is 98.78%. Though group lacks of consensus in recognition to the decision-making rule, two groups $j=1$ and $j=3$ in decision-making groups have high support, even to a firm degree. The credibility of decision-making rule ω_7 and ω_{16} is lowest to groups and their results are 1.22% and 1.21%. The reason of low credibility in ω_7 is that there is lacking of consensus in groups. The reason of low credibility in ω_{16} is that there is lacking of consensus in groups as well as believing that credibility of group is low in decision-making rule. The analysis result shows that group decision aggregation by relative entropy is feasible, scientific and reliable.

5. Conclusions

In the process of group decision-making in emergency management, information plays an important role to make effect decision strategies. This paper analyzes the information aggregation mechanism by using information relative entropy method, and believes that the law of information aggregation can be found out according to the information aggregation model.

6. References

[1] R. R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 18, No. 6, 1988, pp. 183-190.

[2] S. E. Bodily, "A Delegation Process for Combining Individual Utility Function," *Management Science*, Vol. 25, No. 10, 1979, pp. 1035-1047.

[3] H. W. Brock, "The Problem of Utility Weights in Group Preference Aggregation," *Operations Research*, Vol. 18, No. 1, 1980, pp. 176-187.

[4] F. R. Krzyszto and S. L. Duck, "Preference Criterion for Flood Control under Uncertainty," *Water Resources Research*, Vol. 15, No. 3, 1979, pp. 513-520.

[5] D. F. Chen, Y. J. Lei and Y. Tian, "Aggregation and Application of the Information about Attribute Weights with Different Forms in Group Decision Making," *Journal of Air Force Engineering University* (*Natural Science Edition*) (*in Chinese*), Vol. 7, No. 6, 2006, pp. 51-53.

[6] B. J. Li, S. F. Liu, "A New Method on Group Decision-Making with Interval-Number Judgment Matrices," *Chinese Journal of Management Science* (*in Chinese*), Vol. 12, No. 6, 2004, pp. 109-112.

[7] J. Wu and D. S. Huang, "The Uniform Methods for Interval Number Preference Information Multi-Attribute Decision Making," *Systems Engineering-Theory Methodololgy Application* (*in Chinese*), Vol. 12, No. 4, 2003, pp. 359-362.

[8] S. Z. Zhou, J. J. Zhu, S. F. Liu and H. H. Wang, "Goal Programming Model on Multiple Kinds of Uncertain Preference Aggregation Approach," *Systems Engineering-Theory & Practice* (*in Chinese*), Vol. 3, 2008, pp. 118-124.

[9] J. J. Zhu, "Group Aggregation Approach of Two Kinds of Uncertain Preference Information," *Control and Decision* (*in Chinese*), Vol. 21, No. 8, 2006, pp. 889-892.

[10] X. Q. Feng, C. P. Wei, Z. Z. Li and G. Hu, "Aggregating of Interval Number Judgment Matrixes with Maximum Satisfaction," *Systems Engineering* (*in Chinese*), Vol. 24, No. 11, 2006, pp. 42-45.

[11] G. W. Wei, "Some Induced Geometric Aggregation Operators with Intuitionistic Fuzzy Information and their Application to Group Decision Making," *Applied Soft Computing*, Vol. 10, No. 2, 2010, pp. 423-431.

[12] H. J. Zimmermann and P. Zysno, "Decisions and Evaluations by Hierarchical Aggregation of Information," *Fuzzy Sets and Systems*, Vol. 10, No. 3, 1983, pp. 243-260.

[13] K. F. Xie and G. Chen, "Entrepreneurial Team's Risk Decision-Making Process Based on Group Learning," *Computational Intelligence and Software Engineering* (*in Chinese*), Vol. 2, No. 4, 2009, pp. 1-4.

Intelligent Information Management, **2010, 2, 483-486**
doi:10.4236/iim.2010.28058 Published Online August 2010 (http://www. SciRP.org/journal/iim)

Real-Time Traffic Signal Timing for Urban Road Multi-Intersection

Lin Dong, Wushan Chen
College of Mechanical Engineering, Shanghai University of Engineering Science, Shanghai, China
E-mail: *Donglin*1985*@hotmail.com*
Received April 19, 2010; *revised June* 5, 2010; *accepted July* 19, 2010

Abstract

Abstract—this paper develops a real-time traffic signal timing model which is to be integrated into a single intersection for urban road, thereby solving the problem of traffic congestion. We analyze the current situation of the traffic flow with release matrix firstly, and then put forward the basic models to minimize total delay time of vehicles at the intersection. The optimal real-time signal timing model (non-fixed cycle and non-fixed split) is built with the Webster split optimal model. At last, the simulated results, which are compared with conventional model, manifest the promising properties of proposed model.

Keywords: Traffic Signal Control, Traffic Flows, Real-Time Signal Timing, Release Matrix, Split, Passion Distribution

1. Introduction

Traffic congestion is a severe problem at an intersection in urban, having been causing many critical problems and challenges in major and most populated cities around the world. Indeed, signal timing is the technique to appropriately decide the signal cycle and effective green time or other factors, with respect to various intersections, phases, traffic flows and others. The conventional signal timing model by using fixed cycle and fixed split algorithm is less efficiencies to cause congestion problem. Therefore, the design of real-time signal timing model for a single intersection in urban traffic networks is important to improve service function of road. A lot of ground can be gained in this area. Indeed, a model of signal timing, proposed in [1-3], is constructed with respect to minimization of vehicle average delay and number of stops at an intersection. Other study, for example [4], the reinforcement learning method, which defining the state sets, action sets and reward function, is introduced to the intersection signal control system. Critical Intersection Control (CIC) [5], is using real-time traffic data to better assign green times to conflicting movements. A real-time dynamic model on multiphase traffic flows, as in [6-13], is that each phase time in next cycle is determined with GA optimizing method. Using data of the traffic flow in the current cycle and the cycle before, and the traffic flows in the next cycle may be estimated through the linear pre-estimation method.

Based on the above research, this paper develops a real-time signal timing model (non-fixed cycle and non-fixed split). According to real-time traffic flows, the model decides the signal cycle and effective green to relieve the congestion problem. Due to the random and complexity, the accurate models is hard to find to describe the traffic flow system. So, the release matrix is introduced to analyze the current situation of the traffic flow. The fixed cycle and non-fixed split model is put forward to minimize total delay time of vehicles, which use the maximum road capacity as the control level. Finally, integrated with the Webster split optimal model, the optimal real-time signal timing model (non-fixed cycle and non-fixed split) is obtained.

2. Modeling of Real-Time Signal Timing

This section is to focus on presenting the real-time optimized signal timing model, including signal state description, delay estimation, mathematical formulation and others.

2.1. Fixed Cycle and Non-Fixed Split Model

As is shown in **Figures 1(a-c)**, a movement state of a vehicle at a single intersection, denoted by the release matrix p_{ijk}, is defined by the follow information: 1) a phase (i), 2) a direction (j), and 3) a road-lane (k). The release matrix p_{ijk}, which describes the current traffic situation, is given by:

$$p_{ijk} = \begin{cases} 1, \text{the vehicle at } (i, j, k) \text{ stays at the same place} \\ 0, \text{the vehicle at } (i, j, k) \text{ can advance} \end{cases} \quad (1)$$

where i values 1, 2, 3 and 4 represent respectively the first, second, third and fourth phase. Correspondingly, the j values 1, 2, 3 and 4 represent respectively the direction of south, west and north respectively. And k values 1, 2 and 3 represent respectively turning left, directing and turning right.

Therefore, the release matrix p_{ijk} in **Figure 1** can be written as:

$$p_{ijk} = \begin{Bmatrix} \{1,0,1,0,0,1,1,0,1,0,0,1\}, \\ \{0,1,1,0,0,1,0,1,1,0,0,1\}, \\ \{0,0,1,1,0,1,0,0,1,1,0,1\}, \\ \{0,0,1,0,1,1,0,0,1,0,1,1\}, \end{Bmatrix} \quad (2)$$

Now that vehicle travel delay is the main index for estimating intersection Level of Service (LOS), accuracy of the delay estimation directly affects the effectiveness of the signal control. So, a calculation for the total delay of vehicles uses mathematical analysis for traffic flow and stop equation. We assume that t_i (I = 1, 2, 3, 4) is the green phase at the single intersection. λ_{ijk} is the arrive ratio of vehicle at (i, j, k). Then, the total number of vehicles at (i, j, k), which flow-into the intersection, is as follow:

$$s_1 = \lambda_{ijk} t_i \quad (3)$$

During the green time, we assume that the ratio of flow-out vehicle at (i, j, k) is u_{ijk}. In one cycle, the total number of flow-out vehicles at (i, j, k) can be given by:

$$s_2 = p_{ijk} u_{ijk} t_i \quad (4)$$

The total number of delay vehicles at (i,j,k) in one cycle is s^l_{ijk}, then:

$$\begin{cases} S^l_{ijk} = S^{l-1}_{ijk} + \lambda_{ijk} t_i - p_{ijk} u_{ijk} t_i, S^{l-1}_{ijk} + \lambda_{ijk} t_i \\ S^l_{ijk} = 0, S^{l-1}_{ijk} + \lambda_{ijk} t_i < p_{ijk} u_{ijk} t_i \end{cases} \quad (5)$$

where $i = 1,2,3,4; j = 1,2,3,4; k = 1,2,3$ and s^{l-1}_{ijk} is the total number of delay vehicles at (i, j, k) in l-1 cycle. Indeed, total number of delay vehicles at the end of l cycle at the single intersection can be written as:

$$S = \sum_{j=1}^{4} \sum_{k=1}^{3} S^l_{4jk} \quad (6)$$

Based the above analysis, we put forward the fixed cycle and non-fixed split model to minimize total delay time of vehicles to maximize the capacity of traffic flow at the intersection. Therefore, the objective function realizing

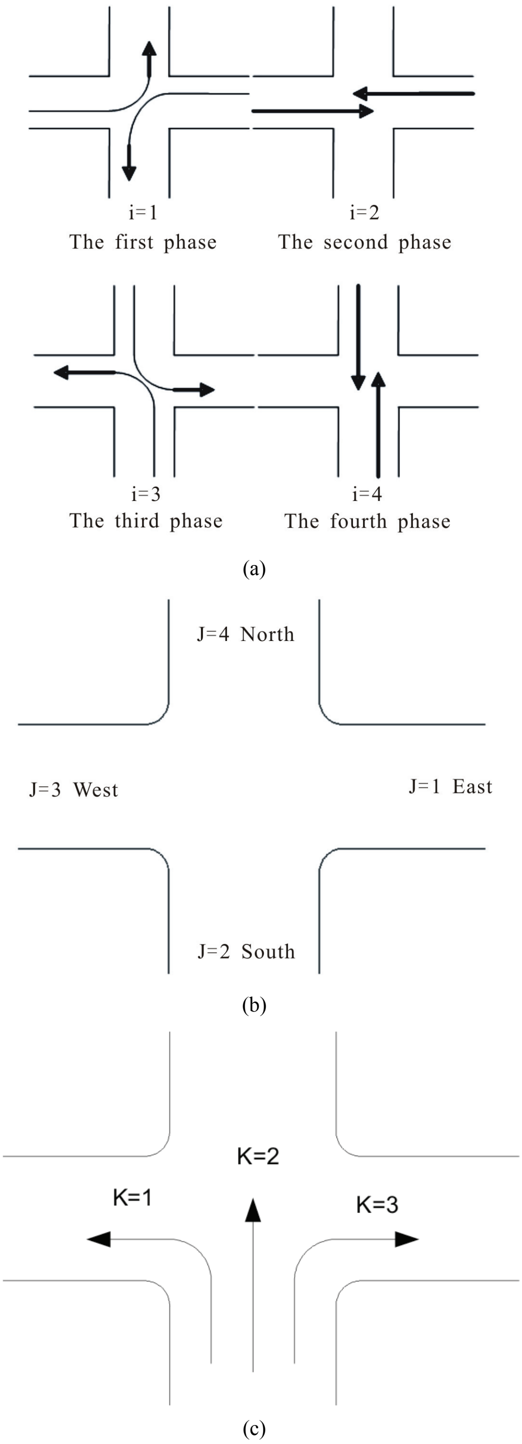

Figure 1. Current traffic flow analysis at a single intersection.

the minimum delay can be given by:

$$S^t = \min\left(\left(\sum_{j=1}^{4} \sum_{k=1}^{3} S^l_{3jk} + \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{3} \lambda_{ijk} t_i - \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{3} p_{ijk} u_{ijk} t_i \right) (T - t_i) \right) \quad (7)$$

Subject to:

$$t_1 + t_2 + t_3 + t_4 = T \tag{8}$$

where T denotes the cycle time at the single intersection.

Taking the security of pedestrians crossing the road at the intersection into account, the green time of one phase is supposed to be less than a specific value e ($e \geq 6s$, generally). So, the green time of each phase is supposed to be subject to the (9):

$$6 \leq t_i \leq T - 18, i = 1,2,3,4 \tag{9}$$

The u_{ijk}, which describe the maximum vehicle at the road, is a constant up to 1.5 veh/s. and T is up to 120 s. Generally, the effective green time will follow on the yellow time. So, the yellow time is supposed to be considered as a part of green time to simplify the calculation.

2.2. Optimized Model

The "Webster" split algorithm is that the green time is determined by the traffic flow ratio in adjoining phases after determining the signal cycle. The equation can be written as:

$$t_i = \frac{(T - L) \times (\dfrac{v}{s_{isaturation}})_i}{\sum_{i=1}^{n} (\dfrac{v}{s_{isaturation}})_i} \tag{10}$$

where t_i is the green time of i phase. $s_{saturation}$ is the constant up to 1.5 veh/s. $(\dfrac{v}{s_{isaturation}})_i$ is traffic flow ratio at i phase and key road-lane. $V = \max(\lambda_{ijk}, \lambda_{i(j+2)k})$. T denotes the cycle time, and, L is the lost time.

We simplified the (10) to (11) for easy calculation:

$$t_i = \frac{T \times (\dfrac{v_i}{s_{isaturation}})}{\sum_{i=1}^{4} (\dfrac{v_i}{s_{isaturation}})} \tag{11}$$

Therefore, the relationship between T and traffic flow can be written as:

$$T = \frac{t_i \cdot \sum_{i=1}^{4} \dfrac{v_i}{s_{isaturation}}}{\dfrac{v_i}{s_{isaturation}}} \tag{12}$$

Based on the above calculation, the ultimate goal function can be written as:

$$S^t = \min\left(\left(\sum_{j=1}^{4}\sum_{k=1}^{3} S^l_{3jk} + \sum_{j=1}^{4}\sum_{k=1}^{3}\sum_{i=1}^{4} \lambda_{ijk} t_i - \sum_{j=1}^{4}\sum_{k=1}^{3}\sum_{i=1}^{4} p_{ijk} u_{ijk} t_i\right) \left(\frac{t_i \cdot \sum_{i=1}^{4} \dfrac{\max(\lambda_{ijk}, \lambda_{i(j+2)k})}{s_{isaturation}}}{\dfrac{\max(\lambda_{ijk}, \lambda_{i(j+2)k})}{s_{isaturation}}} - t_i\right)\right) \tag{13}$$

where t_i is The green time of *i* phase, $\dfrac{\max(\lambda_{ijk}, \lambda_{i(j+2)k})}{s_{isaturation}}$ is traffic flow ratio at i phase and key road-lane, $s_{saturation}$ = 1.5 veh/s, T is the cycle time, and L is lost time in a cycle respectively.

2.3. Simulation

Because the traffic pattern in the simulation is stochastic, the traffic flow based on the Poisson distribution is applied to obtain the estimates of model performance. The parameters of Poisson distribution are accurately determined by the analysis of traffic flow at high demand level from Songjiang road, Shanghai. Simulation results will be presented as follow.

The model 1, 2 and 3, as in **Tables 1**, **2** and **3**, represent the fixed cycle and fixed split model, fixed cycle and non-split cycle model and real-time signal timing model respectively. The detailed performance improvement at high demand compare the results of three models based on the total vehicle delay for all movement at each time step and the cumulated delay over three cycles. From **Table 1**, the simulation results indicate that the real-time signal timing model limit the ratio of delay vehicle within 30%. Indeed, the optimal model out performance conventional model by approximately 10%. The simulation results present that the proposed real-time model is more efficient and fairness than the others at the high demand level.

3. Conclusions

The paper developed the real-time signal timing model (non-fixed cycle and non-fixed split) to relieve the traffic congestion. Introducing the release matrix to analyze the situation of traffic flow, the basic model was proposed to minimize total delay time of vehicles. Then, the optimal real-time signal model was realized by combining the Webster split optimal model. Finally, based on simulated

Table 1. the simulation of fixed cycle and fixed split model.

Cycle No.	Cycle Time (s)	Phase time				the ratio of delay vehicles
		*t*1	*t*2	*t*3	*t*4	
1	80	20	20	20	20	41%
2	100	25	25	25	25	39%
3	120	30	30	30	30	39%

Table 2. the simulation of fixed cycle and non-fixed split model.

Cycle No.	Cycle Time(s)	Phase time				the ratio of delay vehicles
		*t*1	*t*2	*t*3	*t*4	
1	80	27	20	20	13	32%
2	100	51	18	17	14	36%
3	120	52	27	27	14	32%

Table 3. the simulation of non-fixed cycle and non-fixed split model.

Cycle No.	Cycle Time (s)	Phase time				the ratio of delay vehicles
		*t*1	*t*2	*t*3	*t*4	
1	102	30	28	29	15	25%
2	105	24	27	31	23	27%
3	95	19	23	16	37	28%

data using Passion distribution, result was showed that the "non-fixed cycle, non-fixed split" model out performance and is more cost effective than "fixed cycle, fixed split" model and "fixed cycle, non-fixed timing". This research gives useful suggestions on signal control to prevent the traffic congestion.

4. References

[1] G. F. Qu, S. Y. Chen, Q. B. Zhu and Y. Zhang, "Determine Traffic Signal Timing Plan with Particle Swarm Optimization," *Journal of Computational Information Systems*, Vol. 4, No. 6, December 2008, pp. 2809-2816.

[2] D. Mariagrazia, F. M. Pia and M. Carlo, "A Signal Timing Plan Formulation for Urban Traffic Control," *Control Engineering Practice*, Vol. 14, No. 11, November 2006, pp. 1297-1311.

[3] E. D. Ferreira, E. Subrahmanian and D. Manstretten, "Intelligent Agent in Decentralized Traffic Control," *IEEE Intelligent Transportation System Conference Proceedings*, Oakland, August 2001, pp. 707-711.

[4] K. Lu, J. M. Xu and Y. S. Li, "An Optimization Method for Single Intersection's Signal Timing Based on SARSA (λ) Algorithm," *Control and Decision Conference*, July 2008, pp. 5146-5150.

[5] S. Alexander, B. R. Gallagher and K. P. Patel, "Determining Capacity Benefits of Real-Time Signal Control at an Intersection," *Transportation Research Record*, Vol. 16, No. 1, November 2007, pp. 78-83.

[6] Y. X. Huang and K. C. Shi, "Real-Time Control of Traffic Signal with Genetic Algorithms Optimizing in Urban Intersection," *Systems Engineering-Theory & Pratice*, Vol. 21, No. 3, 2001, pp. 102-106.

[7] B. Wolshon and W. C. Taylor, "Analysis of Intersection Delay under Real-Time Adaptive Signal Control," *Transportation Research Part C: Emerging Technologies*, Vol. 7, No. 1, February 1999, pp. 53-72.

[8] D. A. Roozemond and J. L. H. Rogier, "Agent Control Traffic Lights," *ESIT*2000, Aachen, 2000.

[9] M. Nazmi, N. Hanim and S. Takaba, "Simulation and Evaluation for Solving the Traffic Flows Problem at Four Intersection in Malaysia," *Proceedings for the* 24*th JSST Conference*, Chester, July 2005.

[10] M. Higashikubo, T. Hinenoya and K. Takeuchi, "Traffic Queue Length Measurement Using an Image Processing Sensor," *The* 3*rd ITSWorld Conference*, Budapest, 1996.

[11] R. L. Kelsey and K. R. Bisset, "A Simulation Environment for Fuzzy Control of Traffic Systems," 12*th IFAC-World Congress*, Sydney, Austria, Vol. 5, 18-23 July 1993, pp. 553-556.

[12] C. C. Lee, "Fuzzy Logic in Control System: Fuzzy Logic Controller - Part I & II," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, 1990, pp. 404-435.

[13] J. Favilla, A. Machion and F. Gomide, "Fuzzy Traffic Control: Adaptive Strategy," *Proceedings of the* 2*nd IEEE International Conference on Fuzzy Systems*, San Francisco, March 1993, pp. 1371-1376.

Intelligent Information Management, 2010, 2, 487-497
doi:10.4236/iim.2010.28059 Published Online August 2010 (http://www.SciRP.org/journal/iim)

Analysis and Design of Distributed Pair Programming System*

Wanfeng Dou, Yifeng Wang, Sen Luo
School of Computer Science and Technology, Jiangsu Research Center of Information Security & Confidential Engineering, Nanjing Normal University, Nanjing, China
E-mail: *douwanfeng@njnu.edu.cn*
Received June 7, 2010; *revised July* 10, 2010; *accepted August* 11, 2010

Abstract

Pair Programming (PP) that has gained extensive focus within pedagogical and industrial environments is a programming practice in which two programmers use the same computer to work together on analysis, design, and programming of the same segment of code. Distributed Pair Programming (DPP) system is a programming system to aid two programmers, the driver and the navigator, to finish a common task such as analysis, design and programming on the same software from different locations. This paper first reviews the existing DPP tools and discusses the interaction and coordination mechanism in DPP process. By means of activity theory and language-action theory, some basic requirements of the DPP system are presented. Then, a design framework of such system and functions of each sub-system are deeply analyzed. Finally, a system prototype is implemented by plug-in style in Microsoft Visual Studio environment.

Keywords: Pair Programming, Distributed Pair Programming, Software Engineering, Extreme Programming

1. Introduction

In recent years, agile software methodologies have attracted increasing interest within pedagogical and industrial environments, with extreme programming being considered the most important of these agile methodologies [1]. In the agile manifesto, the authors state twelve general principles that all highlight the importance of flexibility and collaboration. One of these techniques, which are being adopted by software development group, is known as Pair Programming (PP), in which two developers work side by side, on the same computer, to collaboratively produce a design, an algorithm, a code, etc [2]. Taking these principles would imply a distributed application of agile methods, such as distributed extreme programming. Although some tools have been developed to better support distributed agile software development, there is still a need for additional research on tools and processes for distributed extreme programming, especially for solutions that extend the most obvious solution of providing a shared code editor. As the trend towards global software development continues, pair programming in which two developers are required to work in face-to-face interaction don't meet the need of global software development. This needs to create computer programs through pair programming practice where developers are located in different workstation but they collaborate simultaneously to solve the same problem. This approach is called Distributed Pair Programming (DPP). This paper focuses on reviewing the existing distributed pair programming systems, and presents system design and implementation. This paper has six sections. After this introduction, Section 2 gives a related work about DPP tools. Section 3 discusses analysis approach of DPP based on activity theory and language theory. The requirements of DPP tool are presented in Section 4. Section 5 describes the design and implementation of prototype system. Section 6 draws conclusions.

*This project was financed partly by the eleventh five-years plan from Jiangsu Education Science in 2009, No: D/2009/01/093.

2. Related Work

2.1. Pair Programming

Extreme programming, also known as XP, includes a set of principles and practices for the rapid development of high quality software. XP identifies 12 best practices of software development and takes them to an extreme. Pair programming originated in industry as a key component of the XP development methodology. As the name suggests, pair programming involves two programmers working at the same computer to create code or analyze requirements and develop design and etc. This provides a mechanism for real-time problem solving and real-time quality control [2]. One programmer acts as the driver, who actively writes code or design document and has control of the keyboard and mouse. The other partner acts as the navigator, who helps plan as well as identify and prevent any syntactic or strategic deficiencies in code or design document, thinks of alternatives and asks questions [3]. The collaborator may exchange roles at any time during the pair programming session, or not at all.

The concepts underlying pair programming are not new [4,5], but pair programming has only recently attracted significant attention and interest within the software industry and academia. Several previous controlled experiments have concluded that pair programming has many benefits over solo programming [6]. Pair programming has significant improvements in code reviewing and various others measures of quality of the programs being developed including lower duration with only minor additional overhead in terms of a measure of cost or effort [4-5]. But, with respect to time taken and improvement of functional correctness of the software product compared with Solo programming showed no positive effects of pair programming [7]. The reasons are the difference in sample populations (e.g., students or professionals), study settings (e.g., amount of training in pair programming), lack of power (e.g., few subjects), and different ways of treating the development variables (e.g., how correctness was measured and whether measures of development times also included rework), and task complexity (e.g., simple dependent tasks, or complicated projects) [8-9].

Pair programming originated in industry as a key component of the extreme programming development methodology. As the name suggests, pair programming involves two programmers working at the same computer to create code or analyze requirements and develop design and etc. This provides a mechanism for real-time problem solving and real-time quality control [2]. One programmer acts as the driver, who actively writes code or design document and has control of the keyboard and mouse. The other partner acts as the navigator, who helps plan as well as identify and prevent any syntactic or strategic deficiencies in code or design document, thinks of alternatives, and asks questions. The collaborator may exchange roles at any time during the pair programming session, or not at all. Pair programming has been shown to be an effective pedagogical approach in teaching courses such as introductory computer science [10-11]. Undergraduate software engineering [12], and graduate object-oriented software development [13]. Studies have shown that pair programming creates an environment conducive to more advanced, active learning and social interaction, leading to students being less frustrated, more confident and more interested in IT [14], and also improve retention of women in computer science [15]. Pair programming encourages students to interact and cooperate with partners in their classes and laboratories, or development teams, thereby creating a more collaborative environment in which pairs discuss problems, brainstorm each other, and share knowledge. Pair programming also benefits the teaching staff. A pair of students can always analyze and discuss the low-level technical or procedural questions that typical burden the teaching staffs in the laboratory, hence there are fewer questions to be dealt with.

Distributed pair programming is a style of programming in which two programmers who are geographically-distributed and synchronously collaborating over the Internet work together on the same software artifact. Comparing with pair programming, DPP decreases the scheduling issues that arise for developers trying to schedule collocated pair programming. Making DPP technology available to students increases the likelihood that they will pair program. Trying distributed pair programming increases the likelihood that students will pair program remotely in the future. While DPP has been shown to be better than distributed non-pair programming, DPP is not perfect. The main reason is to require a better tool to support the DPP process.

2.2. Tools of Distributed Pair Programming

In pair programming environment, however, obstacles such as limited facilities, geographic separation, and scheduling often present challenges to collocated pair programming. DPP enables students or developers to collaborate from different locations to work on their programming projects remotely. One of the main trends in software development has been the globalization of the software industry. Motivating factors behind this trend include hiring qualified programmers in different cities and countries for software companies, placing development group closer to their client's location, creating quickly virtual development groups, and working continuously on critical projects by working on different time zones for groups [16].

Researchers have proposed several tools to better support distributed pair programming [17-22].These existing tools adopt either an application sharing approach to enhance an existing editor suite or provide customized tools that include various groupware features such shared awareness [17]. Customized groupware tools do not support all of the features needed by pair programming and thus limit partner's ability to successfully accomplish their work. On the other hand, application sharing solutions lack process support and thus met collaboration awareness.

2.2.1. Application Sharing Tools

JAZZ system is an example with an application approach [18]. It is an extension of eclipse that supports the XP and workflows in asynchronous interaction. JAZZ allows users to stay aware of co-workers and initiate chat sessions, and be invited to a synchronous pair programming session using an application sharing system. JAZZ implements a shared editing plug-in that provides a synchronous shared code editor based on operation transformation approach. But this plug-in is not integrated into the workflow of pair programming, and thus does not provide awareness and has no explicit switching of roles.

MILOS is another application sharing system [19]. IT provides awareness of co-present users and allows users to initiate pair programming sessions using application sharing like JAZZ. MILOS makes use of existing IDEs and integrates single-user development environments into pair programming settings. But, application sharing approach does not support flexible pairing such as one-to-many pairing way, and role switching. TUKAN is a special purpose groupware for all phrase of the XP process [20]. It provides a shared editor for manipulating code together and users can highlight important code using a remote selection. Moomba extends the awareness tools of TUKAN and support Java IDE where the users can use a shared java editor [21]. However, TUKAN and Moomba use ENVY environment and are built as a proprietary tool and thereby cannot provide the same domain specific tool support as it is present in modern IDEs. This is one of the reasons why they have not gain high popularity.

Eclipse is a popular and more open environment that allows closer coupling of the developing IDEs [4]. Coordination work can be integrated into Eclipse in the internal browser window or special-purpose planning plug-in. The Eclipse Communication Framework (ECF) aims at integration a collaboration infrastructure with the IDE. Sangam is an Eclipse plug-in for Eclipse users to share workspace so that developers may work as if they were using the same computer [22]. Sangam use an event-driven design for this plug-in. There are three basic components in Sangam: event interceptor, message server, and event reproducer. The responsibility of the event interceptor is to capture the event when the driver does something in Eclipse and then send it to the message server. When the event reproducer receives a message and interacts with Eclipse to perform the driver's action. Saros plug-in supports driver-navigator interaction in Eclipse in a distributed pair programming session, and provides awareness on files that are opened at the driver's site [4]. Saros includes a shared editor that allows collaborative code creation, and remote selections that allow the navigator to point at relevant pieces of code. Xpairtise is an Eclipse plug-in that offers shared editing, project synchronization, shared program, test execution, user management, built-in chat communication, and a shared whiteboard [4].

RIPPLE is a plug-in for the popular Eclipse integrated development environment in which data on collaborative programming is collected. RIPPLE is designed for use in educational setting to facilitate various forms of collaborative programming problem solving including distributed pair programming and distributed one-to-one tutoring [23]. RIPPLE extends the architecture implemented in Sangam. Compilation and execution of code, as well as generation of console message, are performed directly by Eclipse. However, RIPPLE currently only supports Java programming because the event-driven behavior of it requires that language-specific messages be transmitted between users. The textual dialogue of RIPPLE is an instant-message-style chat program that supports enforced turn-taking in dialogue.

2.2.2. Customized Tools

COLLECE, developed using Java technology, is a groupware system to support synchronous distributed pair programming practices. COLLECE provides a set of tools including editor, session panel, coordination panels, and structured chat [24]. The editor provides a workspace in which the driver inserts or modifies the source code of the program that is being built. The session panel provides a simple awareness of partner that shows the photo and name of each pair. The coordination panels include three coordination tools that allow a collaboration protocol to be established: edition coordination panel, compilation coordination panel, and execution coordination panel. The structured chat is used to express conversational acts that are usually used during program coding, compilation and execution.

COPPER is a synchronous source code editor that allows two distributed software engineers to write a program using pair programming. Its functions include communication mechanisms, collaboration awareness, concurrency control, and a radar view of the documents, among others. COPPER system is based on the C/S architecture. It is composed of three subsystems: collaborative editor, user and document presence, and audio subsystems. The editor is further decomposed into the Editor module and the document server. The Editor module implements a turn-taking synchronous editor and the document server provides document storage, document editing access control, user authentication and permissions, and document presence extensions.

However, low display refresh rate can sometimes be confusing or something significant may be lost in the remote display. The trace of the mouse pointer is another problem if both developers use no same resolution for their monitors. Hence, next-generation tool is still analyzed and studied in terms of requirements of distributed pair programming.

3. Analysis and Interaction in DPP System

3.1. Analysis Based on Activity Theory in DPP System

Activity theory, as a social psychological theory on human self-regulation, is a well suited epistemological foundation for design. Activity theory was first used to design the user interface by Bodker. Later it has been extended and refined by numerous other authors. In particular, activity theory is used to understand cooperative work activities supported by computers [25,26]. Pair programming is a social activity involved two programmers, driver and navigator. This paper use activity theory as a theoretical basis for understanding the cooperative work activities in DPP.

Broadly defined, activity theory is a philosophical framework for studying different forms of human praxis as development processes, with both individual and social levels interlinked. Three of the key ideas of activity theory can be highlighted here: activities as basic unit of analysis, the historical development of activities and internal mediation with activities [25]. Activities—an individual can participate in several at the same time—have the following properties: 1) an activity has a material object; 2) an activity is a collective phenomenon; 3) an activity has an active subject, who understands the motive of the activity; 4) an activity exists in a material environment and transforms it; 5) an activity is a historically developing phenomenon; 6) contradiction is realized through conscious and purposeful actions by participants; 7) the relationships within an activity are culturally mediated.

Y. Engestrom has made an attempt to establish a structural model of the concept activity and culturally mediated relationships within it (**Figure 1**). This structure includes three components, namely subject, object and community, and forms three relationships: subject-object, subject-community and object-community.

This activity model contains three mutual relationships between subject, object and community: the relationship between subject and object is mediated by tools, that between subject and community is mediated by rules and that between object and community is mediated by the division of labor. Each of the mediating terms is historically formed and opens to further development. In this activity model, four subsystems are formed: production subsystem, communication subsystem, assignment subsystem and consumption subsystem.

The production subsystem is used by the subject (e.g., driver and navigator) to manipulate the object into outcome (e.g., analysis, design or programming for a code). In **Figure 1**, the production subsystem involves three components: subject, object and tool. In DPP, this subsystem is a shared editor that can support the synchronous editing, role switching, test execution and file sharing, etc.

Communication subsystem, in **Figure 1**, involves also three components: subject, community and rule. For instance, In DPP, the driver and navigator use this subsystem to communicate each other so as to solve the problems met during pair programming. The driver and navigator as a community should stand by rules. For example, a partner as a role of driver, another must be a navigator. They switch role at intervals. The communication subsystem that includes the relationship and interaction between subject and community should provide chat session, whiteboard and audio or video communication. The communication subsystem must be designed for users to easy discussion on problems and suggestions on their task and further focus on the shared code.

Assignment subsystem builds the relationship between object and community through establishment of the division of labor, that is to say, it assign activity according to social rules and expectation. In DPP, the pair with a driver role is responsible for writing the code using keyboard and mouse, and the other with a navigator role is responsible for reviewing the code written by the partner and gives some suggestions. During DPP, they should switch the role at intervals.

Consumption subsystem describes how subject and community to cooperate to act on object. Consumption process stands for the inherent contradictions of activity system. Although the goal of the production subsystem is to transform the object into outcome, it also consumes the energy and resources of subject and community. The consumption subsystem may plan arrangement and provide the resources for DPP.

In **Figure 1**, the emphasis of analysis of activity system is production subsystem. The production of object is leaded by the results or intention of activity system. For example, the activities of DPP lead to produce the code with high quality. Production subsystem is usually considered to be the most important subsystem. Hence, understanding the production subsystem will be a good start for design of DPP system.

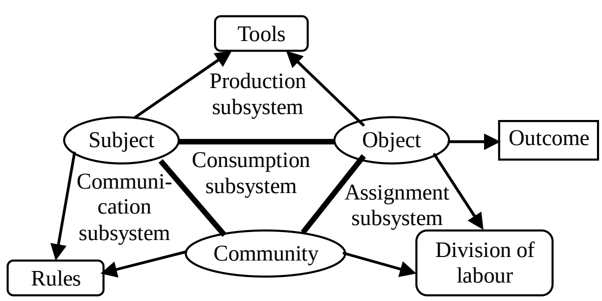

Figure 1. Basic structure of an activity.

3.2. Conversation Model of DPP

In a DPP system, two programmers, the driver and the navigator, work commonly on the same task such as a code, or a design, or an analysis by network and related tools. In order to the efficiency of their programming, the communication of pairs is important to effective cooperation for them. When the driver is editing, the navigator may observe the code or design remotely and at any moment to give suggestions about it, or think about optional solution or strategy. In other one, the driver may request acknowledgement of the pair to it during he/she writes the code. The conversation model is to describe the communicating process between the driver and the navigator so that we clarify how to communicate between them during pair programming. In follow section, we construct a conversation model of DPP by means of language-action theory.

In designing a DPP system for practical situations, we need to consciously focus on the context and application. The structure of the system determines the possibilities for action by the people who use it, and it is this action structure that is ultimately important. Design is ontological. That is what we are participating in the larger design of the organization and collection of practices in which it plays a role. In describing or constructing any system, we are guided by a perspective. This perspective determines the kinds of questions that will be raised and the kinds of solution that will be sought. One can consciously apply a perspective as a guide to design. It will not provide answers to all of the specific design questions, but serves to generate the question that will demand answers.

The language/action perspective is one of the relevant theoretical contributions that have appeared within cooperative work. Cooperative work is coordinated by the performance of language actions, in which the partner become mutually convinced to the performance of future actions and they make declarations creating social structures in which those acts are gathered and interpreted [27]. The language/action perspective has had a significant role with computer supported cooperative work. The PP or DPP is a cooperative activity with two actors, which can be modeled by language/action perspective.

The language/action perspective emphasis pragmatics, not the form of language, but what people do with it. The language/action has five fundamental illocutionary points —things you can do with an utterance [27]: 1) Assertive that commits the speaker to something being the case –to the truth of the expressed proposition; and 2) Directive that attempts to get the header to do something; and 3) Commission that commits the speaker to future course of action; and 4) Declaration that brings about the correspondence between the propositional content of the speech act and reality; and 5) Expressive that expresses a psychological state about a state of affairs.

The need of supporting DPP with suitable computer based tools implies the investigation of the deep aspects of cooperation and clarification. Cooperation clarification, to the extent that is made up of communication and negotiation, can be fully characterized under the assumption that the DPP can be viewed as a special linguistic action game between the driver and the navigator, constituted by asset of rules defining the conversations possible within it. The results of conceptual and experimental research motive the following answer: the driver and navigator spend their time taking commitments for future activities each other, coordinating the programming work, switching role according to the situation, explaining the problems they encounter during pair programming, reviewing the code. This needs to precisely develop conversation between the driver and navigator in order to take commitments for an effective negotiation and coordination of the activities.

A conversation between the driver and the navigator during a DPP process is a sequence of related utterances. The utterance within a conversation can be classified from the pragmatic point of view in some basic categories of speech acts on the basis of their illocutionary point namely, directives (e.g., Request, Acceptance or Rejecting of a promise), commission (e.g., Promise, Count-offer, Acceptance or Rejecting of a commitment, Declaring of commitment fulfillment). Each conversation involves two actors in the DPP: the driver and the navigator, and follows the pattern which defines the possible sequences of speech acts characterizing the specific type of conversation. In accordance with language/action theory, there are also three main types of conversation occurring in any PP. The first is the conversation for action, characterized by the definition of a commitment for doing an action. The driver in the PP can recognize, e.g., the conversations opened by a request, where the driver opening the conversation asks the partner for some activities; the conversation by a promise, where the navigator agrees and provides the support for its fulfillment. The second is the conversation for possibilities, where the pairs discuss a new possibility for the code, in terms of requirements, code structure, language and related knowledge these conversations, when successful and devoted to topics under the competence of the pair, end with a declaration explaining the concept and agreeing with the code. The third is the conversation for clarification, where the pairs cope with or anticipate breakdowns concerning interpretations of conditions of satisfaction for action. The conditions are always interpreted with request to an implicit shared background, but sharing is partial and needs to be negotiated. There is no sharp line between them, but they are accompanied by different moods.

The PP is characterized by a specific organizational rule, which define the roles of pair programming, role switching and compatibility of pairs. These rules can be expressed in terms of conversation possibilities open to that role. For example, two roles are defined in pair programming mode, the driver and navigator. The driver is responsible for writing the code by the mouse and keyboard and the navigator can view and test the code, and think about the structure and some strategies. The navigator cannot enter any code but can point out the existing problems and request the discussion with the driver. If possible, the pairs can periodically switch role.

The conversation for action forms the central fabric of a DPP. In a simple conversation for action, the driver (A) in a pair programming makes a request to the navigator (B). The request is interpreted as having certain conditions of satisfaction, which characterize a future course of action by B. After the initial utterance (the request), B can accept (satisfaction for action); reject (end of the conversation); or counter-offer with alternative conditions. Each of these in turn has its own possible continuations (e.g., after a counter offer by B, A can accept, reject, or counter-offer again).

The meaning of a language/action exists in the interpretations of a driver and a navigator in a particular situation with their particular backgrounds. The request is an initial utterance, for the driver there are several kinds of request: 1) Help (request for collection of some materials or testing for the codes); and 2) Negotiation (request for clarification of some problems; and 3) Question (request for the design of programming).

Reducing the complexity of a work process and of communicative mode going on within it is that they need to be supported in copying with that complexity [28]. This means that any tools supporting practices of a conversation must broaden and not restrict the range of all kinds of possibilities of its participants. The relationship between conversation and commitments is not a one-to-one one. Making a commitment explicit is sometimes very useful, in particular when we must ensure that it will be completed satisfactorily. Considering a conversation as a sequence of communication events to which can be attached not only documents of any types but also any numbers of commitment negotiations. A DPP procedure includes a set of conversations. Each conversation with a commitment and a title includes a set of events. Each event is a structured message characterized by its completion time, content, associated code, attached documents, its sender and its receiver.

In a DPP procedure, there are a lot of conversation occurring between the driver and navigator. For example, the driver may request a help for some materials with the code from the navigator, or hope to discuss some uncertain programming problems. In some time the driver may request the navigator to test the code written by him/her. The navigator can point out the existing problems during reviewing the code.

A support system of commitment negotiation is required to help the user to understand the context where he/she is negotiating each other, as well as the state of the negotiation. The goal of this conversation model is to develop a theoretical framework for understanding communication within a DPP process.

Conversations are just sequences of communicative events involving two participants, driver and navigator in PP, where each participant is free to be as creative as he/she wants. Conversations can be supported by a system making accessible the sequence of records of the communicative events, together with the documents generated and/or exchanged and with the commitment negotiations steps which occurred during them. Within this model a commitment may be viewed as the respective negotiation steps performed within a conversation by the driver and navigator and by the documents that are attached to them. Any negotiation step of a commitment is characterized by its object, its time and its state. Commitment negotiations are therefore fully transparent to their actors within conversations without imposing any normative constraints upon them comparing to fully scheduleed model of conversation.

4. Requirements of DPP System

Distributed pair programming means that two developers synchronously collaborate on the same design or code from different locations. The results of experiments indicate that DPP is competitive with collocated pair programming in terms of software quality and development time [13], and it is feasible and effective to develop software code using distributed pair programming [29]. Considering the trend of globalization in software development we have aimed at finding out how programmers could effectively apply DPP technique with the use of appropriate groupware tools, and what would be the requirements of such tools. For this purpose we defined a set of requirements of distributed pair programming tool in terms of the analysis of the existing groupware tools and DPP tools [16,22-24], and features of pair programming. According to the technology of Computer Supported Cooperative Work (CSCW) we have identified the following requirements of the DPP tool.

4.1. Shared Editing Integrating Existing Editor

As a source code editor it should highlight keywords based on the programming language being used and not only provide conventional editing tools such as: Cut, Copy, Paste, Find, and Replace, but also the options of compilation and execution of the source code being edited and should notify the users of the error messages reported by the compiler. On the other hand, the existing editors with the integration of developing environment supporting a specific language have very powerful functionalities. Moreover, developers hope to use their familiar editor or integrating development environment to pair programming, and for some language, for example Java language has several editors or developing environment support editing and compiling source code such as Eclipse, JDK, JBuilder, Visual Café, and etc. It is required that collaborative pair programming tool can integrate these editors or developing environments. However, there are some problems to be solved when paring developers use different editors or developing environments. For example, interoperation is one of the main problems in information exchange between two different editors with the same language due to the differences of their format of editing commands and the parameter options of compiling and executing the source code.

4.2. Shared File Repository

The source code files and related documents being edited should be controlled at the repository. These files and documents should be shared among all members of the development team. Furthermore, configuration management tools are available to control the version change of code flies and documents. Mechanisms to request and obtain shared resources need to be provided so that developers invite their partner for pair programming.

In the DPP setting, users hope to share intermediate results by passing to one or more users. A shared file repository is provided for users to place and retrieve files. Users can browser files and pars on these files. The shared file repository allows users to organize the files in folders.

Pair programming tool should support text and audio or video-based communication so that the pairs discuss questions and selection of solutions or know the partner's sensibility and intention through these communicating tools.

4.3. Activity Indicator

Users need time to perform a task but only the results are shared among them. In the DPP setting, users need to be aware of other user's activities, which can use a peripheral place. The interface of the DPP also should support the presence of the role state of pairs and the function of role exchange.

4.4. Role Switch and Concurrent Control

When a navigator wants to own the role of driver and write the code the system should support to apply for and release the token. Once there is the occurrence of role exchange, the DPP tool should support the file locking to control the change of the code.

Concurrent operations to shared artifacts can lead to conflicting actions that confuse the interacting users and result in inconsistency on the artifacts, make interaction difficult. By means of a token and only let user holding the token modify or access the shared resources. In DPP setting, the user with the driver role can hold a token and allow modifying the code, and the user with a navigator role only browser the code written by the partner. Role switch can allow them switch the role each other and change the token holder.

4.5. DPP Communication Session

Pair programming process is a negotiation process for programming problems such as design strategy, code specification, and collaborative testing. Its goal is to improve code quality and increase programming efficiency. Hence, distributed pair programming tool should support free and natural problem negotiations with a set of communicative events associated a conversation.

For distributed interaction, communication between pairs poses an important role in DPP. There are all kinds of communication channels, such as text chat, whiteboard, remote selection, and audio or video channel. The text chat is a simplest communication style in which users can send short text messages and distribute these messages at the pair's site. The driver or navigator initiates a conversation at any time aiming at a code segment or a design. The conversation with a title is composed of events. Those events are mutually related to the same conversation with a sequence of occurring of them. Each event is represented a message format organized with complete time, content, sender, receiver, and optional code segment and attached documents. But the disadvantage of textual chat communication for a DPP is that the driver needs his or her hands to produce code. Normally, coding and talking goes hand in hand. Thus, the textual chat will not be the most important communication medium [4]. Whiteboard chat is similar to textual chat, but the only difference is that whiteboard uses graphical object to support their interaction. Whiteboard is usually used to discuss design problems of software. For example, pairs in DPP use UML (Unified Modeling Language) to finish the design and analysis of the software.

As an alternative or addition to the communication functionality an audio or video channel can be embedded in the DPP. An audio or video channel supports parallel communication and coding. But the disadvantages of these channels are that they will consume too much network bandwidth, not be stable enough, and establishing connections will not be quick and easy.

Remote selection shows remote user's selection to a local user. Make sure that other pair is aware of his or her partner who has selected the object or edited the code.

5. Design of DPP System

The goal of distributed pair programming system between heterogeneous code editors or developing environments is to enhance pair programming ability and cooperation capacity among partners. As a result of programmers daily use different kinds of single-user code editors or developing environments during their designing or programming task, the functions of existing distributed pair programming system is inferior to the one of the commercial or open-source single-user code editors or developing environments.

Figure 2 shows a framework of distributed pair programming system with the same or different code editors or environments, compatible to the specific program language, between driver and navigator. The system is implemented by the client/server architecture. The communication management module is responsible for transferring of operation information and event or message between the driver and navigator.

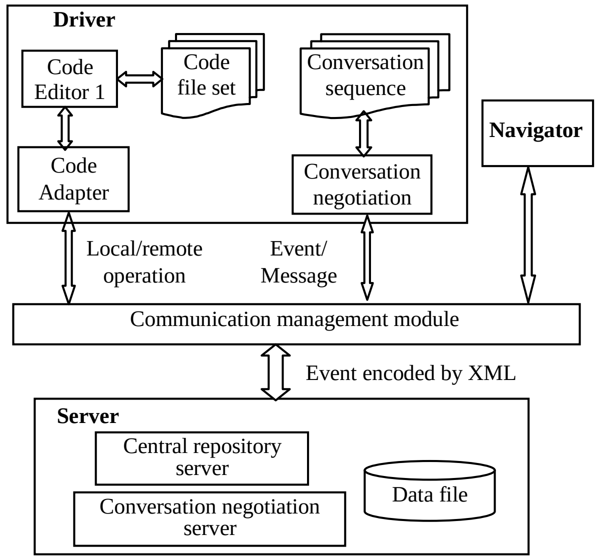

Figure 2. A framework of DPP system.

5.1. Collaborative Editing Subsystem

Moreover, the existing code editors or environments lack good compatibility with the commercial single-user systems, and its usability is poor. It is impossible for programmers to accept these systems to support their development task concurrently. In order to solve this problem, the collaboration transparency technology emerges as the times require. Collaboration transparency technology causes group of users to be possible of no revision to the single-user code editors or developing environments, allows them directly to use familiar single-user code editors or developing environments for distributed pair programming tasks, thus the research of collaboration transparency technology has a vital value.

In **Figure 2**, the driver can select any code editors or developing environments that support a specific program language. The Code Adapter component can capture any local operations from the driver, filter any inessential information, and recombine into useful operation information in a common or standard format. Similarly, the navigator can select the same or different editor with the driver. This is due to the like or experiments of the navigator. The Code Adapter component also transforms the operation information received by Information transfer component into suitable format according to the requirements of local editor, and executes it to the local editor.

In server site, the Central repository server is a resource repository. These resources include source code files, design documents, users, pair information. The design of Central repository operates on the client/server architecture. The clients reveal these resources, and the server is responsible for updating of them. In a one-to-many pair mode, the core programmer needs to know new changes to the code when he/she switches to previous partner.

5.2. Conversation Negotiation Subsystem

The conversation for negotiation subsystem is responsible for the initiating, maintenance, organization, and storage of conversation. The messages of conversation are transferred by specific format between the driver and navigator. Its role is to aid pairs to communication and negotiation for some coding or design problem. Each conversation corresponds to a commitment.

In DPP procedure, there are a lot of conversations occurring between the driver and navigator. For example, the driver may request a help for some materials with the code from the navigator, or hope to discuss some uncertain programming problems. In some time the driver may request the navigator to test the code written by him/her. On the other hand, the navigator may point out the existing problems by conversation negotiations during reviewing the code.

Each conversation is associated a sequence of message which is composed of title, time, source, destination, content, attached documents, associated code segments. **Figure 3** shows the model of conversation for the DPP process. The conversation negotiation server is responsible for recording all conversation information between the driver and the navigator so as to querying and indexing for later usage.

5.3. A DPP System Prototype

We have implemented a preliminary prototype system which adopts the client/server architecture. The system can provide the basic functions of distributed pair programming in plug-in form integrated MS Visual Sudio environment. The XPPlugin based on client/server architecture consists of three subsystems: real-time communication, code synchronization and pairing management. In client site, communication and code synchronization are implemented in MS Visual studio plug-in style. In server site, all management tasks of DPP are finished by a solo program. The network communication between client and server is implemented by a XMPP (Extensible Messaging and Presence Protocol) which is an open instant-massage protocol based on XML (Extensible Markup Language). This prototype uses open source software, agsXMPP, under .net environment to support the interaction between pairs.

The client program exists in plug-in form which conforms to the specification of MS Visual Studio plug-in. **Figure 4** shows the window of our prototype system. The window consists of three sub-windows: code sharing and editing window, communication window and role switching and control window.

The system architecture, as showed in **Figure 5**, is divided into four layers:

- User interface layer provides the functionalities such as login in, text chat, code control and role switch. User interface is implemented by using LoginForm, chatControl, CodeMonitorcontrol class.
- Middle layer is decided by MS Visual Studio. Only using this layer, the XPplugin can support the tool window pane as a visual studio standard tool pane to be used freely. The goal of design is that DPP tools are allowed to be embedded in visual studio environment, thus increase the efficiency of the prototype system.
- Interaction layer implements interaction between the XPPlugin and internal data of visual studio, including XPPluginPachage and SccService class. XPPluginPackage inherits Package class to allow the whole program as a plug-in to be loaded into Visual studio environment. SccService implements the management of code encapsulated as a service which can be freely called by either internal of the program or other plug-in or programs of Visual studio.
- Network interface layer encapsulates a network communication class using a XMPP protocol to implement the interaction between client and server. Datahandler is an instance of such network communication class.

6. Conclusions

In this paper, we have reviewed the features of the exist-

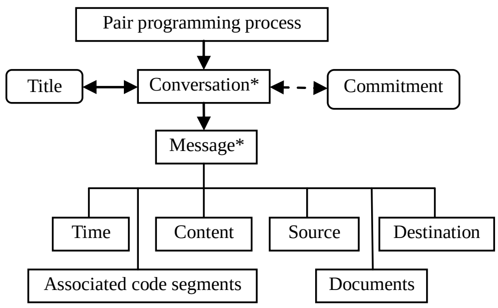

Figure 3. Model of conversation for the DPP process.

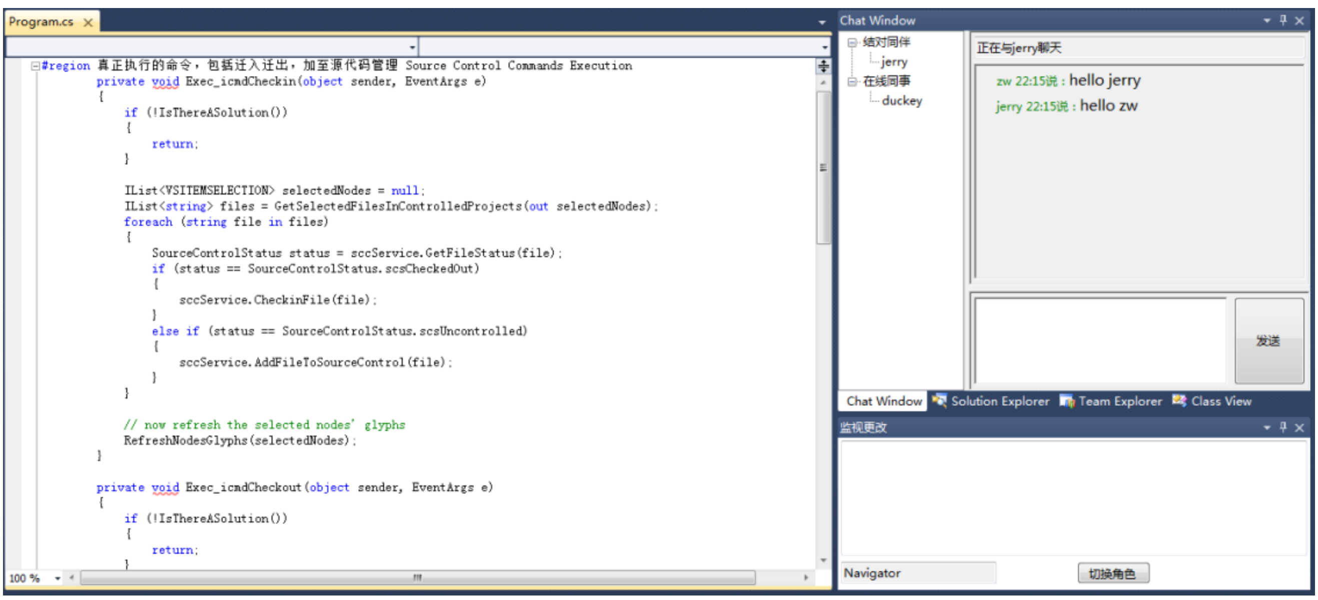

Figure 4. Main window of prototype system.

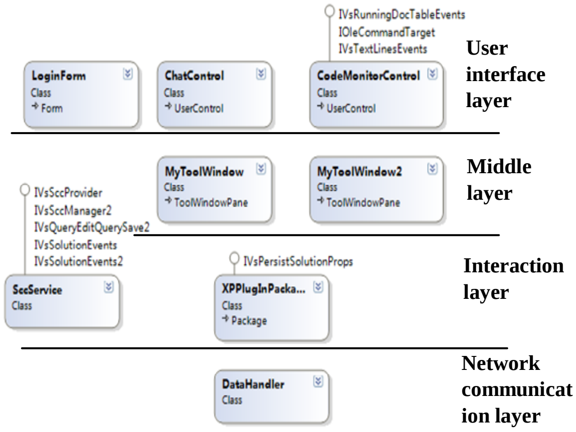

Figure 5. Layered structure of prototype system.

ing distributed pair programming tools, analyzed their advantages and disadvantages. An activity theory is introduced to analyze the process model of DPP and the related main subsystem. A conversation model with commitments is presented based on language/action perspective as a framework for understanding communication within DPP processes. We have analyzed the requirements of distributed pair programming system, presented four important aspects in designing distributed pair programming system: 1) interoperation between heterogeneous editors corresponding to the same language; 2) file sharing at the repository and awareness of pair programming information; 3) role switch and control, and 4) conversation pattern with negotiation. Finally, we have presented a framework supporting distributed pair programming with heterogeneous editors or developing environments. In the future, we will improve our current preliminary system with new collaborative tools to support communication with audio and video channels. We also hope to integrate the existing developing environment, such as J++, JBuilder, Visual Cafe, which is relative to Java language, into our system.

7. References

[1] R. Duque and C. Bravo, "Analyzing Work Productivity and Program Quality in Collaborative Programming," *The* 3*rd International Conference on Software Engineering Advances*, Sliema, 2008, pp. 270-276.

[2] D. Preston, "Using Collaborative Learning Research to Enhance Pair Programming Pedagogy," *ACM SIGITE Newsletter*, Vol. 3, No. 1, January 2006, pp. 16-21.

[3] L. Williams, R. Kessler, W. Cunningham and R. Jefferies, "Strengthening the Case for Pair Programming," *IEEE Software*, Vol. 17, No. 11, 2000, pp. 19-21.

[4] T. Schummer and S. Lukosch, "Understanding Tools and Practices for Distributed Pair Programming," *Journal of Universal Computer Sciences*, Vol. 15, No. 16, 2009, pp. 3101-3125.

[5] M. M. Muller, "Two Controlled Experiments Concerning the Comparison of Pair Programming to Peer Review," *Journal of Systems and Software*, Vol. 78, No. 2, 2005, pp. 166-179.

[6] L. Williams, R. R. Kessler, W. Cuningham and R. Jeffries, "Strengthening the Case for Pair Programming," *IEEE Software*, Vol. 17, No. 4, 2000, pp. 19-25.

[7] J. Ncwroclci and A. Wojciechowski, "Experimental Evaluation of Pair Programming," *Proceedings of European Software Control and Metrics Conference*, London, 2001.

[8] E. Arishoim, H. Gallis, T. Dyba and D. I. K. Sjoberg, "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise," *IEEE Transactions on Software Engineering*, Vol. 33, No. 2, 2007, pp. 65-86.

[9] H. Gallis, E. Arishoim and T. Dyba, "An Initial Framework for Research on Pair Programming," *Proceedings of the* 2003 *ACM-IEEE International Symposium on Empirical Software Engineering*, Toronto, 2003, pp. 132-142.

[10] C. McDowell, L. Werner, H. Bullock and J. Fernald, "The Effects of Pair Programming on Performance in an Introductory Programming Course," *Proceedings of the* 33*rd Technical Symposium on Computer Science Education*, Cincinnati, 2002, pp. 38-42.

[11] N. Nagappan, L. Williams, *et al.*, "Improving the CS1 Experience with Pair Programming," *Proceedings of the* 34*rd Technical Symposium on Computer Science Education*, Reno, 2003, pp. 359-362.

[12] L. Williams and R. L. Upchurch, "In Support of Student Pair Programming," *Proceedings of the* 32*nd Technical Symposium on Computer Science Education*, Charlotte, 2001, pp. 327-331.

[13] P. Baheti, E. Gehringer and D. Stotts, "Exploring the Efficacy of Distributed Pair Programming," *Proceedings of XP Universe*, Springer-Verlag, Chicago, 2002, pp. 208-220.

[14] L. Williams, D. M. Scott, L. Layman and K. Hussein, "Eleven Guidelines for Implementing Pair Programming in the Classroom," *Agile* 2008 *Conference*, Kopaonik, 2008, pp. 445-451.

[15] L. Werner, B. Hanks and C. McDowell, "Pair Programming Helps Female Computer Science Students," *ACM Journal of Education Resources in Computing*, Vol. 4, No. 1, 2004, pp. 1-8.

[16] H. Natsu, J. Favela, *et al.*, "Distributed Pair Programming on the Web," *Proceedings of the* 4*th Mexican International Conference on Computer Science*, Los Alamitos, 2003, pp. 81-88.

[17] B. Hanks, "Tool Support for Distributed Pair Programming: An Empirical Study," *Proceedings of Conference Extreme Programming and Agile Methods*, Calgary, 2004, pp. 1-18.

[18] S. Hupfer, L. T. Cheng, S. Ross and J. Patterson, "Introduction Collaboration into an Application Development Environment," *Proceedings of the Computer Supported Cooperative Work*, ACM Press, New York, 2004, pp. 21-24.

[19] F. Maurer, “Supporting Distributed Extreme Programming,” *Proceedings of Conference on Extreme Programming and Agile Methods*, Springer Verlag, London, 2002, pp. 13-22.

[20] T. Schummer and J. Schummer, “Support for Distributed Teams in Extreme Programming,” In: G. Succi and M. Marchesi, Eds., *Extreme Programming Examined*, Addison Wesley, Boston, 2001, pp. 355-377.

[21] M. Reeves and J. Zhu, “Moomba: A Collaborative Environment for Supported Extreme Programming in Global Software Development,” In: *Lecture Notes in Computer Science*: *Extreme Programming and Agile Process in Software Engineering*, Springer, London, 2004, pp. 38-50.

[22] C. W. Ho, S. Raha, E. Gehringer and L. William, “Sangam: A Distributed Pair Programming Plug-in for Eclipse,” *Proceedings of the* 2004 *OOPSLA Workshop on Eclipse Technology Exchange*, New York, 2004, pp. 73-77.

[23] K. Elizabeth, A. Dwight, *et al*., “A Development Environment for Distributed Synchronous Collaborative Programming,” *Proceedings of the* 13*th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Madrid, 2008, pp. 158-162.

[24] R. Duque and C. Bravo, “Analyzing Work Productivity and Program Quality in Collaborative Programming,” *The* 3*rd International Conference on Software Engineering Advances*, 2008, pp. 270-276.

[25] K. Kuutti, “The Concept of Activity as a Basic Unit for CSCW Research,” In: L. J. Bannon, M. Robinson and K. Schmid, Eds., *Proceedings of the* 2*nd ECSCW*, Kluwer Academical, Amsterdam, 1991, pp. 249-264.

[26] K. Kuutti, “Identifying Potential CSCW Applications by Means of Activity Theory Concepts: A Case Example,” *Proceedings of CSCW*, ACM Press, New York, 1992, pp. 233-240.

[27] T. Winograd, “A Language/Action Perspective on the Design of Cooperative Work,” *Human Computer and Interaction*, Vol. 3, No. 30, 1998, pp. 203-220.

[28] G. D. Michelis and M. A. Grasso, “Situation Conversations within the Language/Action Perspective: The Milan Conversation Model,” *Proceedings of the* 5*th Conference on CSCW*, Chapel hill, North Carolina, 1994, pp. 1-12.

[29] P. Baheti, L. Williams, *et al*., “Exploring Pair Programming in Distributed Object-Oriented Team Projects,” *Proceedings of OOPSLA Educators Symposium*, Seattle, 2002, pp. 1-6.

Intelligent Information Management, **2010, 2, 437-497.**
Published Online August 2010 in SciRes (http://www.SciRP.org/journal/iim/).

TABLE OF CONTENTS

Volume 2 Number 8 **August 2010**

