This article was downloaded by: [National Chiao Tung University 國立交通大學]
On: 24 April 2014, At: 06:53
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

A scheduling model for the refurbishing process in recycling management

T.C.E. Cheng[a], B.M.T. Lin[b] & Y. Tian[b]
[a] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong.
[b] Department of Information Management and Finance, Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan.
Published online: 17 Oct 2013.

To cite this article: T.C.E. Cheng, B.M.T. Lin & Y. Tian (2013) A scheduling model for the refurbishing process in recycling management, International Journal of Production Research, 51:23-24, 7120-7139, DOI: 10.1080/00207543.2013.842024

To link to this article: http://dx.doi.org/10.1080/00207543.2013.842024

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

International Journal of Production Research, 2013
Vol. 51, Nos. 23–24, 7120–7139, http://dx.doi.org/10.1080/00207543.2013.842024

Taylor & Francis
Taylor & Francis Group

A scheduling model for the refurbishing process in recycling management

T.C.E. Cheng[a], B.M.T. Lin[b]* and Y. Tian[b]

[a] *Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong;* [b] *Department of Information Management and Finance, Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan*

(*Received 27 March 2013; accepted 20 August 2013*)

We propose from the perspective of operations scheduling a novel model of the refurbishing process in recycling management. We model the refurbishing process as a two-stage flowshop that dismantles products into parts in stage one and refurbishes the parts on dedicated machines in stage two. The model also features that the performance measure of a schedule is defined by operation-based completion times, which is different from the job-based performance measures traditionally adopted in the scheduling literature. We analyse the optimality properties and computational complexity of some special cases of the problem. We derive lower bounds on the optimal solution based on a disaggregation technique and the assignment problem, and develop dominance rules incorporating estimates of the effects of partial schedules on unscheduled jobs. We present a heuristic approach, based on LP relaxation, and analyse its performance ratio. We also develop two metaheuristic algorithms, based on iterated local search and ant colony optimisation, to produce approximate solutions. The results of computational experiments show that the metaheuristics generate better solutions than the simple weighted shortest processing time dispatching rule, and the NEH-based and CDS-based algorithms, which are commonly deployed to treat the classical two-machine flowshop scheduling problem.

Keywords: refurbishing flowshop; operation-based performance measure; weighted total completion time; approximation algorithm

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

1. Introduction

While Gantt (1919) and other practitioners drew initial research attention to scheduling issues in manufacturing (Pinedo 2012), the seminal paper of Johnson (1954) inaugurated mathematical and algorithmic developments in scheduling theory. The first paper published in the *International Journal of Production Research* (*IJPR*) that considered the mathematical aspect of scheduling was probably due to Alcalay and Buffa (1963). In the 1960s, several prominent works, including Maxwell (1964), Gupta (1968), Hwang, Tillman, and Fan (1967) and Charlton and Death (1968), on machine scheduling successively appeared in *IJPR* that had an important bearing on the future development of this emerging field of research. Throughout the past five decades of its history, *IJPR* has published numerous studies on models, theories, and algorithms that advance the theory and practice of scheduling. Adding to this large body of work, we introduce a new scheduling model arising from contemporary manufacturing with its focus on environmental sustainability.

This paper presents a model of the refurbishing process in recycling management, highlighting the necessity for environmental protection while seeking to minimise the operating cost. The disposing of TVs, computers, monitors, printers and mobile phones is causing great and growing environmental and public health concerns, especially when the world becomes more wired and companies introduce new products at an accelerating pace (Schwarzer et al. 2005; Roy and Geetha 2006). According to UNEP (2007), waste electrical and electronic equipment (WEEE) or e-waste accounts for 1–3% of the total municipal waste generation in the USA, and WEEE in EU has increased by 16–28% every five years, which is three times faster than average annual municipal solid waste generation. Only 10–15% of retired electronic products are recycled in the US and the rest merely collect dust in the corners of people's houses or are discarded to dump sites (US EPA 2007a, 2007b, 2007c). By 2020, e-waste from old computers will have jumped by 200–400% from 2007 levels in China and South Africa, and by 500% in India (UNEP 2010). Hazardous particles and toxic chemicals easily leak out from landfills and penetrate the surroundings. Therefore, as a result of consumers and e-waste advocates pushing producers to take responsibility, many technology firms are engaged in eliminating certain chemicals in their production processes and designing products that are eco-friendly and easy to be recycled, as well as offering recycling schemes to help their customers to dispose of obsolete equipment. For example, Hewlett-Packard has reached a milestone of responsibly recycling two billion pounds of electronic products and

*Corresponding author. Email: bmtlin@mail.nctu.edu.tw
Part of this paper was presented at the CORS-INFORMS International Meeting, June 14–17, 2009, Toronto, Canada.

© 2013 Taylor & Francis

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

supplies since 1987 (HP Environmental History 2012), and Dell provides several programmes to keep e-waste out of landfills and is two-thirds of the way towards its goal of recycling one billion pounds of e-waste by 2014 (Environmental Leader 2011).

Due to the necessity for environmental protection, as well as the potential of recovering valuable assets, the concept of closed-loop in supply chain management is attracting increasing attention of the operations management community (Stoner and Wankel 2008). Researchers have suggested that introducing refurbished products to a competitive market can be profitable or at least beneficial to the relationships with customers and supply chain partners (Ferrer and Swaminathan 2006; Vorasayan and Ryan 2006; Ferrer and Swaminathan 2010; Ovchinnikov 2011). There has been growing research on various aspects of remanufacturing in hopes of capturing the most immediate profitable return, such as timing control of used-product arrival (Ketzenberg, Souza, and Guide Jr 2003; Karamouzian, Teimoury, and Modarres 2011), acquisition policies (Galbreth and Blackburn 2006; Minner and Kiesmller 2011), lot sizing policies (Atasu and Centinkaya 2006; Tang and Teunter 2006; Andrew-Munot and Ibrahim 2012), and creating competitive advantage from strategic remanufacturing (Ferguson and Toktay 2006; Georgiadis, Vlachos, and Tagaras 2006).

How to efficiently manage the recycling process has become an important issue. This paper proposes a model of the refurbishing process from the perspective of operations scheduling that concerns making decisions on allocating limited resources to activities, subject to functional constraints, so as to optimise some performance measures. Research on scheduling and its applications has experienced phenomenal growth since the 1950s. Most of the performance measures that have been investigated in the scheduling literature are defined as a maximum function or a sum function. The most commonly known scheduling objectives include the makespan (maximum completion time), the maximum tardiness, the sum of completion times (total completion/flow time), the sum of tardiness and the number of tardy jobs. All of these so-called *classical* performance measures are job-based, i.e. they are defined in terms of job completion times. The notion of *aggregation* emerged in the past decade, according to which the objective function is defined in terms of the completion times of orders rather than jobs. For example, in Cheng, Kovalyov, and Lin (1997), the objective function depends on the delivery times of batches. Customer order scheduling (Lin and Kononov 2007; Leung, Li, and Pinedo 2007; Wang and Cheng 2007) and concurrent open shop scheduling (Lin and Cheng 2011; Mastrolilli et al. 2010; Roemer 2006) are two other examples. The demand for effective and efficient planning of the recycling process prompts a paradigm shift towards the reverse direction, i.e. *disaggregation*: a job comprises multiple operations or tasks and each task has its own identity; the objective function is defined in terms of the completion times of the tasks.

In a recycling process, a product enters the shop to be dismantled into parts, which will then be refurbished for reuse. The identity of a product vanishes when it is dismantled into parts. On the other hand, the parts pick up their own identities that readily induce processing requirements and implicit values. The refurbishing process can be modelled as a two-stage flowshop, in which stage one has a machine installed for dismantling products and stage two comprises a set of independent parallel dedicated machines for processing the different parts the dismantled products have engendered. The major distinctive characteristics of the proposed model are (1) the scheduling objective (performance measure) is defined in terms of parts rather than jobs emerging from the shop and (2) the number of input items (products) and the number of output items (parts) are different. In this paper, we seek to minimise the weighted sum of part completion times as an example of application of the model. To the best of our knowledge, the refurbishing flowshop model is not only new but may also constitute a new direction for scheduling research involving operation-based performance measures. The recycling of used printer toner cartridges provides a motivation for the refurbishing model. Recycling extends the life cycle of each cartridge, allowing for refill and remanufacture. One of the different approaches adopted for processing used cartridges starts with a thorough cartridge dismantling, followed by cleaning the precious parts, including the drums, primary charge rollers and various blades. Refurbished parts become independent items with their own values and will be manufactured into fully functioning cartridges with fresh toner refilled. Recycling computers is another example. But the process may consist of more stages. For example, the main board dismantled from a computer will be further dismantled into chips and precious metals of wiring. This paper focuses on the basic two-stage process with a view to demonstrating the refurbishing model in a concise way.

The remaining part of the paper is organised as follows: in Section 2, we introduce the model and some definitions and present a literature review of related or similar models. We devote Section 3 to complexity analysis of several special cases of the problem, some of which permit polynomial solvability. In Section 4, we discuss the development of lower bounds on the objective value and dominance rules. We give an approximation algorithm and analyse its performance in Section 5. We develop two metaheuristic algorithms and study their performance through computational experiments in Section 6. In Section 7, we conclude the paper and suggest topics for further research.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

2. Problem formulation and literature review

We formally state the problem as follows: a set of products $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ is to be dismantled and refurbished in a two-stage $(1 + m)$-machine flowshop. Installed in stage one is a dismantling machine M_0 that breaks products into parts and in stage two are m different dedicated machines for refurbishing the parts. Upon finishing its stage-one operation on M_0, each product $J_i \in \mathcal{J}$ becomes m different parts $\{J_{i1}, J_{i2}, \ldots, J_{im}\}$, which enter the second stage of the flowshop. Part J_{ik}, $1 \leq k \leq m$, will be processed on dedicated machine M_k. The m parts of a product are independent and thus can be processed in the second stage simultaneously. The processing time required by product J_i or its parts is positive and denoted by p_{ik}, $0 \leq k \leq m$. When $k = 0$, p_{ik} refers to the time for dismantling product J_i. A weight w_{ik}, $1 \leq i \leq n$, $1 \leq k \leq m$, is associated with part J_{ik} that reflects its value or importance. The major characteristic that distinguishes this model from conventional scheduling models is that there are n products (jobs, in scheduling terms) entering the refurbishing shop, which generate mn independent parts (operations). That is, the number of entities before the refurbishing process starts is n and that after the process is mn. Therefore, the objective function of the refurbishing model involves mn, instead of n, completion times because each individual part has its own identity. We denote the model by the three-field notation $\mathtt{RF}(1, m)||\gamma$, where $\mathtt{RF}(1, m)$ stands for a refurbishing flowshop consisting of one stage-one machine and m stage-two machines, and γ is the scheduling objective to be optimised.

The notation that will be used throughout the paper is summarised as follows:

$\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$: set of products;
$\{J_{i1}, J_{i2}, \ldots, J_{im}\}$: parts obtained from product $J_i \in \mathcal{J}$;
M_0: dismantling machine in stage one;
$M_1, M_2, \ldots, M_m$: dedicated machines for processing different part types;
p_{i0}: dismantling time of product J_i on M_0;
p_{ik}: processing time of part J_{ik} on M_k, $1 \leq k \leq m$;
w_{ik}: weight of part J_{ik};
C_{ik}: completion time of part J_{ik} on M_k;
$\sigma = (\sigma_1, \ldots, \sigma_n)$: a particular job sequence, which is a permutation of the job indices;
$Z(\sigma)$: objective value of sequence σ;
Z^*: optimal solution value.

With regard to the makespan objective, if we trace the critical path in the Gantt chart of a schedule of the assembly model studied in Lee, Cheng, and Lin (1993) and Potts et al. (1995), we see that the refurbishing flowshop can be considered as the reverse of the assembly mode, so they have the same makespan. It follows that $\mathtt{RF}(1, m)||C_{\max}$ is strongly NP-hard, too; and all the known results, such as exact and approximation algorithms, for the assembly model are applicable to the refurbishing model. As discussed earlier, a distinctive feature of the refurbishing model is that each part emerging from the shop has its individual value. Therefore, this paper studies the objective function of the sum of the weighted completion times of the parts, which reflects the feature well. The rest of the paper will focus on the development of properties and algorithms for the $\mathtt{RF}(1, m)|| \sum w_{ik}C_{ik}$ problem. Here, we further note that completion time instead of flow time is adopted because we implicitly assume that the value of a part is contained in (or inherited from) the product from which it is dismantled. When a product enters the shop for refurbishing, we expect that valuable parts will emerge from the shop through the dismantling and recycling processes. The value of a part exists at the very beginning and is realised when the part is refurbished.

The scheduling problem that is most similar to the refurbishing flowshop is the two-stage look-ahead scheduling (3MLA) model studied by Herrmann and Lee (1992). In the 3MLA model, there are two types of jobs to process in a two-stage three-machine flowshop. All the jobs are processed on the stage-one common machine and then jobs of each type will move to their stage-two dedicated machines for further processing. If there are n_1 type-one jobs and n_2 type-two jobs, then $n_1 + n_2$ operations are processed on the stage-one machine and the two dedicated machines in stage two are in charge of n_1 and n_2 stage-two operations, respectively. With $n_1 + n_2$ jobs entering the shop, $n_1 + n_2$ job completion times will arise. Herrmann and Lee (1992) analysed the time complexity and designed solution algorithms for three objectives, namely the makespan, total completion time and number of tardy jobs. Without knowing the results of Herrmann and Lee (1992), Yang (2010) gave another NP-hardness proof. Cheng, Lin, and Tian (2009) studied the same model to minimise the weighted sum of machine completion times. They designed a heuristic with a performance ratio of 4/3. Calling the 3MLA manufacturing setting the differentiation flow shop, Lin and Hwang (2011), Liu, Fang, and Lin (2013), and Huang and Lin (2013) studied the problem under the assumption that the job sequences on the machines are known a priori. They provided polynomial-time dynamic programming algorithms to produce optimal schedules for the problem. The reverse model of the 3MLA was independently investigated by Neumytov and Sevastianov (1993) and Oğuz, Lin, and Cheng (1997), where two dedicated machines are included in stage one for two types of jobs and a stage-two common machine processes all the jobs emerging from stage one.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

To tackle most flowshop scheduling problems, the assumption of permutation schedules is commonly made. A permutation schedule implies a consistent sequence of operations/tasks across all the machines. We recall the permutation property for the minimisation of the (weighted) sum of completion times in the traditional two-machine flowshop setting, $F2||\sum(w_i)C_i$. It was proved that a non-permutation schedule can be converted into a permutation one without increasing the objective value. Therefore, only permutation schedules need to be considered for $F2||\sum(w_i)C_i$ (Conway, Maxwell, and Miller 1967). For the $\mathrm{RF}(1,m)||\sum w_{ik}C_{ik}$ problem, examples exist (see Section 3) for which the optimal solution is not necessarily a permutation schedule. Due to technical constraints and machine processing modes, like discrete processing vs. batch processing (e.g. parallel-batching and serial-batching), it may not necessarily be optimal for all the machines to process the jobs and parts in the same sequence. Nevertheless, to simplify exposition without distracting the orientation of the paper, we assume that the schedules under consideration are permutation schedules. When computational complexity is concerned, we discuss both permutation schedules and non-permutation schedules in order to inspire future theoretical studies.

3. Special cases

Given that $\mathrm{RF}(1,m)||\sum w_{ik}C_{ik}$ is strongly NP-hard, we analyse the complexity status of some special cases with a view to gaining some insights into the optimality properties of the solution. We first note that when there is only one stage-two machine ($m = 1$), the problem is equivalent to the classical $F2||\sum(w_i)C_i$ problem (Garey, Johnson, and Sethi 1976), for which only permutation schedules need to be considered.

3.1 *Case 1: $\mathrm{RF}(1,m)|p_{i0} = p_0|\sum w_{ik}C_{ik}$*

In this case, the stage-one processing time is fixed, denoted as $p_{i0} = p_0$. It stems from the scenario that the dismantling operation is the same for all kinds of products. Consider a two-job instance with $m = 2$ second-stage machines: $p_{1,0} = 1, p_{1,1} = 5, p_{1,2} = 10, w_{1,1} = 1, w_{1,2} = 10$; $p_{2,0} = 1, p_{2,1} = 10, p_{2,2} = 5, w_{2,1} = 10, w_{2,2} = 1$. If job sequence (1, 2) is adopted on M_0, then the sequences on M_1 and M_2 are (2, 1) and (1, 2), respectively. On machine M_1, part $J_{1,1}$ yields part $J_{2,1}$ due to weight considerations. Optimal solutions are not permutational because machines M_0 and M_1 have different sequences in the optimal schedule.

Hoogeveen and Kawaguchi (1999) showed, by a reduction from Numerical Matching with Target Sum (SP17, Garey and Johnson 1979), that the classical $F2||\sum C_i$ problem is strongly NP-hard even if all the jobs have the same machine-one processing times. The $\mathrm{RF}(1,1)|p_{i0} = p_0|\sum C_{ik}$ setting corresponds to this hard problem and the following theorem follows:

Theorem 1 (Hoogeveen and Kawaguchi 1999) *Problem $\mathrm{RF}(1,1)|p_{i0} = p_0|\sum C_{ik}$ is strongly $\mathcal{NP}$-hard.*

Regarding the multiple-part setting, the following result follows:

Corollary 1 *The $\mathrm{RF}(1,m)|p_{i0} = p_0|\sum C_{ik}$ problem is strongly $\mathcal{NP}$-hard regardless of whether non-permutation schedules are allowed.*

3.2 *Case 2: $\mathrm{RF}(1,m)|p_{ik} = p_k|\sum w_{ik}C_{ik}$*

Consider the case where the parts of the same type require the same processing time, i.e. $p_{ik} = p_k$ for $k = 1,\dots,m$. The following example shows that the optimal solution is not a permutation schedule: $p_{1,0} = 1, p_{1,1} = 2, p_{1,2} = 2, w_{1,1} = 1, w_{1,2} = 10$; $p_{2,0} = 1, p_{2,1} = 2, p_{2,2} = 2, w_{2,1} = 10, w_{2,2} = 1$.

To deal with this case, we similarly need to address the related problem of minimising the weighted completion time in the traditional two-machine flowshop, where all the jobs have the same processing times on the second machine. If all the jobs are equally weighted, the problem can be solved by applying the shortest processing time (SPT) rule to the machine-one operations (Hoogeveen and Kawaguchi 1999). The weighted version, which is equivalent to this case with $m = 1$, however, has not been investigated in the literature. By a reduction from Odd-Even Partition (Garey and Johnson 1979), we have the following result:

Theorem 2 *The $\mathrm{RF}(1,1)|p_{i,1} = p_1|\sum w_{ik}C_{ik}$ problem is $\mathcal{NP}$-hard.*

Please refer to the supplementary http://web.it.nctu.edu.tw/~bmtlin/IJPR50/proof.pdf for the proof. Regarding the multiple-part setting, the following result follows:

Corollary 2 *The $\mathrm{RF}(1,m)|p_{i,k} = p_k|\sum C_{ik}$ problem is $\mathcal{NP}$-hard regardless of whether non-permutation schedules are allowed.*

3.3 *Case 3: Dismantling operation is dominant*

In this case, $\min_{1\le i\le n}\{p_{i0}\} \ge \max_{1\le i\le n, 1\le k\le m}\{p_{ik}\}$ and non-permutation schedules can be ignored. Assume that non-permutation schedules are permitted. Consider an optimal schedule in which the jobs on machine M_0 are sequenced as $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. Let i be the smallest index such that on some stage-two machine M_k, product $J_{\sigma_i k}$ does not occupy the i-th position. From the premise $\min_{1\le i\le n}\{p_{i0}\} \ge \max_{1\le i\le n, 1\le k\le m}\{p_{ik}\}$, we know that part $J_{\sigma_i k}$ can start immediately when product J_{σ_i} is dismantled and that machine M_k is idle during the interval when job $J_{\sigma_{i+1}}$ is being dismantled. As a consequence, the processing of part $J_{\sigma_i k}$ can be moved into the idle interval without deferring the processing of any other part. In other words, a non-permutation schedule can be transformed into a permutation one without increasing the weighted completion time. When this special case is considered, only permutation schedules are of interest.

Since the dismantling operation is dominant, all the parts of a product can be immediately processed on the dedicated machines when the product is dismantled. For simplicity in presentation, we let the jobs be indexed such that $C_{1,0} < C_{2,0} < \cdots < C_{n,0}$. Therefore, $C_{1k} = p_{1,0} + p_{1k}$ and $C_{ik} = C_{i0} + p_{ik} = \sum_{j=1}^{i} p_{j0} + p_{ik}$ for $i = 2, 3, \ldots, n$. The sum of weighted completion times is

$$\sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}C_{ik} = \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}\Big(p_{ik} + \sum_{j=1}^{i} p_{j0}\Big) = \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}p_{ik} + \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}\sum_{j=1}^{i} p_{j0}$$
$$= \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}p_{ik} + \sum_{i=1}^{n} W_i \sum_{j=1}^{i} p_{j0},$$

where $W_i = \sum_{k=1}^{m} w_{ik}$ is the total weight of the parts disassembled from product J_i. The first term is fixed. The second term can be rewritten as $\sum_{i=1}^{n}\sum_{j=1}^{i} W_i p_{j0}$. To minimise the second term, the problem becomes $1||\sum w_i C_i$, where the job processing times are p_{j0} and job weights are W_i. Therefore, the weighted shortest processing time (WSPT) rule can produce an optimal solution. Because $O(mn)$ is required for computing W_i for all the jobs J_i and $O(n \log n)$ for the sorting procedure, the overall time complexity is $O(n \max\{m, \log n\})$.

THEOREM 3 *The* $RF(1, m)||\sum w_{ik}C_{ik}$ *problem with dominant dismantling operations is solvable in* $O(n \max\{m, \log n\})$ *time.*

3.4 *Case 4: Dismantling operation is negligible*

In this case, $\max_{1\le i\le n}\{p_{i0}\} \le \min_{1\le i\le n, 1\le k\le m}\{p_{ik}\}$. The numerical example of Case 1 is valid here. Therefore, the optimal solution is not necessarily a permutation schedule.

We first consider permutation schedules only. In this case, all the parts on each dedicated machine are consecutively processed without idle time inserted between any two consecutive parts. Thus, $C_{1k} = p_{1,0} + p_{1k}$ and $C_{ik} = C_{i-1,k} + p_{ik} = p_{1,0} + \sum_{j=1}^{i} p_{jk}$ for $i = 2, 3, \ldots, n$. The sum of weighted completion times is computed as

$$\sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}C_{ik} = \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}\Big(p_{1,0} + \sum_{j=1}^{i} p_{jk}\Big) = \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}p_{1,0} + \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}\sum_{j=1}^{i} p_{jk}$$
$$= \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}p_{1,0} + \sum_{k=1}^{m} w_{1k}p_{1k} + \sum_{i=2}^{n}\sum_{j=1}^{i}\sum_{k=1}^{m} w_{ik}p_{jk}.$$

In the above equation, the term $\sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik}p_{10} + \sum_{k=1}^{m} w_{1k}p_{1k}$ is fixed once the first job is determined. So, we want to minimise the last term $\sum_{i=2}^{n}\sum_{j=1}^{i}\sum_{k=1}^{m} w_{ik}p_{jk}$. To the best of our knowledge, the complexity of this optimization problem remains open.[1]

Although the complexity status is unknown, we can develop a dynamic program to solve this case optimally without resorting to full enumeration, which takes $O(n!)$ time. For subset $\mathcal{J}'$ and job $J_j \in \mathcal{J}'$, define function $g(\mathcal{J}', J_j)$ as the optimal contribution made by the jobs of $\mathcal{J}'$ to the objective value, subject to the conditions that the jobs of $\mathcal{J}'$ form a prefix of a complete schedule and that job J_j is scheduled last within $\mathcal{J}'$.

For each job $J_j \in \mathcal{J}$, define the boundary condition

$$g(\{J_j\}, J_j) = \sum_{k=1}^{m}(p_{j0} + p_{jk})\sum_{i=1}^{n} w_{ik}.$$

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

If $|\mathcal{J}'| > 1$, then for each $J_j \in \mathcal{J}'$, we have the following recursion

$$g(\mathcal{J}', J_j) = \min_{J_i \in \mathcal{J}' \setminus \{J_j\}} \left\{ g(\mathcal{J}' \setminus \{J_j\}, J_i) + \sum_{k=1}^{m} p_{jk} \Big(w_{jk} + \sum_{J_l \in \mathcal{J} \setminus \mathcal{J}'} w_{lk} \Big) \right\}.$$

The goal is to find $\min_{J_j \in \mathcal{J}} \left\{ g(\mathcal{J}, J_j) \right\}$.

There are $O(n2^n)$ states in the recursive program. Each state can be computed in $O(n)$ time if we spend $O(mn2^n)$ time calculating the summation terms in advance. Therefore, the overall time complexity is $O(\max\{m, n\}n2^n)$.

We further develop a property that can help eliminate non-promising states and thus reduce the required computing time. Consider any two jobs J_i and J_j that occupy two consecutive positions, not including the first one. Regardless of whether J_i precedes J_j or J_j precedes J_i, the starting time of their immediate successor is not affected. If

$$\sum_{k=1}^{m} p_{jk} w_{jk} + \sum_{k=1}^{m} (p_{jk} + p_{ik}) w_{ik} \le \sum_{k=1}^{m} p_{ik} w_{ik} + \sum_{k=1}^{m} (p_{ik} + p_{jk}) w_{jk}, \tag{1}$$

then a schedule in which J_i immediately precedes J_j can be ignored. When the equality holds, we break ties arbitrarily. Job J_j is said to dominate job J_i if the inequality holds. Such dominance is static because the relationship between each pair of jobs can be determined when the job instance is input. Note that the relation is not transitive for otherwise a total order of all the jobs can be established and thus Case 4 is solvable in polynomial time. The dynamic program can incorporate the dominance relation. If J_j dominates J_i and J_i is considered to precede J_j during the recursion for $|\mathcal{J}'| > 2$, then we can skip this path of the recursion without sacrificing the optimal solution. We will further extend the dominance relation later for the development of a branch-and-bound algorithm.

We consider the further restricted case where all the parts on the same machine share a common weight, i.e. $w_{ik} = w_k$ for all k. In this case, the positional contribution of the i-th job is given by $\sum_{k=1}^{m} [p_{ik}(n - i + 1)w_k] = (n - i + 1) \sum_{k=1}^{m} p_{ik} w_k$, which is a decreasing function of index i. Therefore, arranging the jobs in non-decreasing order of $\sum_{k=1}^{m} p_{ik} w_k$ produces an optimal schedule. The required time is $O(\max\{m, \log n\}n)$.

If non-permutation schedules are allowed, the refurbishment of parts may be deferred as shown in the numerical example. This phenomenon is very similar to the single-machine scheduling problem to minimize the total completion time subject to release dates, $1|r_i|\sum C_i$, which is known to be strongly NP-hard (Lenstra, Rinnooy Kan, and Brucker 1977). We portray in the following theorem the hardness of the non-permutation version of Case 4 by a reduction from $1|r_i|\sum C_i$.

THEOREM 4 *When non-permutation schedules are allowed, the* $\mathrm{RF}(1, m)|p_{i,k} = p_k|\sum w_{ik} C_{ik}$ *problem with negligible dismantling operations is strongly NP-hard when* $m \ge 2$.

Please refer to the supplementary http://web.it.nctu.edu.tw/~bmtlin/IJPR50/proof.pdf for the proof.

4. Lower bounds and dominance rules

In this section, we develop two lower bounds on the weighted sum of completion times and design two dominance rules.

4.1 *Lower bounds*

The lower bounds are based upon the rearrangement technique introduced in Cheng, Lin, and Toker (2000). Lin and Wu (2005, 2006) applied this technique to develop branch-and-bound algorithms for solving $F2||\sum C_i$ and $F2||\alpha C_{\max}+\beta \sum C_i$.

Given an input instance $\mathcal{J}$, we create a corresponding instance $\mathcal{J}' = \{J'_1, J'_2, \ldots, J'_n\}$ by defining the products as follows: (1) Stage-one processing time: p'_{i0} = the i-th smallest element in $\{p_{1,0}, p_{2,0}, \ldots, p_{n0}\}$. (2) Stage-two processing time on M_k: p'_{ik} = the i-th smallest element in $\{p_{1k}, p_{2k}, \ldots, p_{nk}\}$. (3) Part weight: w'_{ik} = the i-th largest element in $\{w_{1k}, w_{2k}, \ldots, w_{nk}\}$.

In the above definitions, we break ties arbitrarily. In the derived set $\mathcal{J}'$, the products are indexed in non-decreasing order of their processing times and the order is consistent across all the machines. On the contrary, the job weights are arranged in non-increasing order. The instance $\mathcal{J}'$ satisfies the *agreeable condition* on processing times, i.e. $\forall J_i, J_j \in \mathcal{J}'(p_{i0} \le p_{j0} \Leftrightarrow p_{i1} \le p_{j1} \Leftrightarrow \cdots \Leftrightarrow p_{im} \le p_{jm})$. It is evident that the agreeable case is more general than the special Case 2 in Section 3, which is NP-hard. Therefore, even the derived set $\mathcal{J}'$ exhibits an "ideal" structure for the objective of weighted sum of completion times, we still have no clue as to how to solve it optimally. In the following, we give a procedure for deriving a lower bound. We develop the procedure by first improving the technique designed by Lin and Wu (2005) and then generalising the technique for $\mathrm{RF}(1, m)||\sum w_{ik} C_{ik}$.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Let the jobs of $\mathcal{J}'$ be sequenced in non-decreasing order of their indices. We calculate the completion times of all the parts. When the completion times are calculated, some stage-two operations could be pruned, if necessary. Denote $C_{ik}(\sigma)$ as the completion time of the part on machine M_k of the i-th job in the derived schedule σ.

Procedure Lower_Bound

Step 1: Create instance $\mathcal{J}'$ from instance $\mathcal{J}$ and set $LB = 0$.
Step 2: Sequence the jobs in non-decreasing order of their indices, i.e. $(J'_1, J'_2, \ldots, J'_n)$.
Step 3: **for** $k = 1$ to m **do**
- Find the smallest index $i \geq 2$ such that $C_{i0} > C_{i-1,k}$. /* Idle time occurs */
- **for** $j = i$ to $n - 1$ **do**
  if $C_{jk} > C_{j+1,0}$ **then** $\left\{ p'_{jk} = p'_{jk} - (C_{jk} - C_{j-1,0});\ p'_{jk} = p'_{jk} - (C_{jk} - C_{j-1,0}) \right\}$
- $LB = LB + \sum_{i=1}^{n} w'_{ik} C_{ik}$.

Step 4: **Return** lower bound LB.

Let σ^* be an optimal schedule for $\mathrm{RF}(1, m)||\sum w_{ik}C_{ik}$ and $C_{\sigma_i^* k}$ be the completion time of the part on machine M_k of the i-th job.

Lemma 1 *For any i and k, $C_{ik}(\sigma) \leq C_{\sigma_i^* k}$.*

Proof The validity is established from the proof in Lin and Wu (2005). □

With the above lemma, the inequality $LB \leq Z(\sigma^*)$ readily follows. Consider the development of a branch-and-bound algorithm. Auxiliary arrays are used to store the sorted lists of the parameters $\{p_{i0}\}, \{p_{ik}\}, k = 1, \ldots, m$, and $\{w_{ik}\}$, $k = 1, \ldots, m$. Then, the lower bound for the partial schedule associated with any node in the enumeration tree can be computed in $O(mn)$ time, which is linear with respect to the size of the input instance. Therefore, we have the following result:

Theorem 5 *Procedure Lower_Bound provides a lower bound LB on Z^* in $O(mn)$ time.*

We use the lower bound in the development of a branch-and-bound algorithm in the sequel. Preliminary computational studies of several flow time-related problems suggest that the lower bound is not tight at the root node, although its overall performance is good. In the development of the lower bound, the arrangement of part weights is rather conservative – part weights on all the machines are sorted in non-increasing order. In the following, we improve the bound by ordering the part weights in a more sophisticated way.

As indicated in Lemma 1, for any i and k, $C_{ik}(\sigma) \leq C_{\sigma_i^* k}$. Multiplying both sides by $w_{\sigma_i^* k}$ and summing over all i and k, we have $\sum_{i=1}^{n}\sum_{k=1}^{m} w_{\sigma_i^* k} C_{ik}(\sigma) \leq \sum_{i=1}^{n}\sum_{k=1}^{m} w_{\sigma_i^* k} C_{\sigma_i^* k}$. The left-hand side can be interpreted as assigning the n m-tuple vectors of weights $\mathtt{W}_i = (w_{\sigma_i^* 1}, \ldots, w_{\sigma_i^* m})$ to the n m-tuple vectors of completion times $\mathtt{C}_j = (C_{j1}(\sigma), \ldots, C_{jm}(\sigma))$. There are $O(n!)$ assignments by defining the cost of assigning weight vector $\mathtt{W}_i$ to the j-th completion time vector $\mathtt{C}_j$ as $\sum_{k=1}^{m} w_{\sigma_i k} C_{jk}(\sigma)$ for $1 \leq i, j \leq n$. That is, we can formulate an assignment problem, for which the minimum assignment cost is less than or equal to $\sum_{i=1}^{n}\sum_{k=1}^{m} w_{\sigma_i^* k} C_{\sigma_i k}$, yielding a lower bound, denoted by $LB_{assignment}$. We summarise the discussion into the following steps:

(1) Run Algorithm Truncation and obtain n completion time vectors, one for each job $J'_j \in \mathcal{J}'$: $\mathtt{C}_j \doteq (C_{j1}(\sigma), C_{j2}(\sigma), \ldots, C_{jm}(\sigma))$.
(2) Prepare from the input instance n weight vectors, one for each job $J_i \in \mathcal{J}$: $\mathtt{W}_i \doteq (w_{i1}, w_{i2}, \ldots, w_{im})$.
(3) Calculate the assignment cost of vectors $\mathtt{W}_i$ and $\mathtt{C}_j$: $d_{ij} = \mathtt{W}_i \bullet \mathtt{C}_j = \sum_{k=1}^{m} w_{ik} C_{jk}(\sigma)$.
(4) Solve the minimum assignment problem.

To prepare input (the cost matrix) to the assignment problem takes $O(mn^2)$ time and to solve the assignment problem needs $O(n^3)$ time. As the required time is not suitable for a light-weight process to be embedded in a branch-and-bound algorithm for each node to invoke, we derive this lower bound only at the root node as an estimate of the optimal solution.

4.2 *Dominance rules*

Dominance rules describe conditions under which certain branches can be skipped without sacrificing the optimal solution, thus reducing the required computing time. This subsection presents a dominance rule in a gradual manner, progressing from a coarse rule to a fine-tuned one. A second rule follows from the dominance rule of the dynamic program developed in Section 3.4.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Let σ denote some partial schedule and J_i and J_j be two unscheduled jobs. Schedule $\sigma' = \sigma \oplus (j, i)$ dominates schedule $\sigma'' = \sigma \oplus (i, j)$, where $\oplus$ is a sequence concatenation operation, if

(a) $\Delta = \sum_{k=1}^{m} \Big(w_{jk} C_{jk}(\sigma') + w_{ik} C_{ik}(\sigma') \Big) - \sum_{k=1}^{m} \Big(w_{ik} C_{ik}(\sigma'') + w_{jk} C_{jk}(\sigma'') \Big) \le 0$; and

(b) $C_{ik}(\sigma') \le C_{jk}(\sigma'')$ for all $1 \le k \le m$.

The reasoning is straightforward. If schedule σ' has a smaller weighted sum of completion times and shorter makespan values across all the stage-two machines, then we can prune the exploration of the subtree rooted at schedule σ'' because the best complete schedule developed from σ'' cannot have a total weighted completion time smaller than that of the best complete schedule developed from σ'. For each node corresponding to partial schedule σ, $O(m)$ time is required for examining the two conditions.

The probability that condition (b) is satisfied is, however, small because the inequality needs to be satisfied on all the refurbishing machines. The condition actually can be relaxed under some circumstances by examining the effects of partial schedules on the unscheduled jobs. To establish the dominance of a partial schedule over another one, two conditions are required to be met: (a) the cost already incurred is smaller and (b) the structures or environments for the remaining unscheduled elements are better. In the following, we propose a new concept to design dominance rules that allows relaxation on one of the two conditions. We first investigate the makespan values of partial schedules σ' and σ''. Denote $\delta_k = C_{ik}(\sigma') - C_{jk}(\sigma'')$ for machine index k. Two cases are discussed based upon the sign of δ_k. Case 1 ($\delta_k > 0$) describes the extra cost for the complete schedule developed from σ' when compared with σ''. On the other hand, Case 2 ($\delta_k < 0$) gives the offset in cost for the complete schedule developed from σ' when compared with σ''. In the following, x^+ denotes $\max\{x, 0\}$ for any real number x.

Case 1 $\delta_k > 0$

Comparing σ' and σ'', we see that the maximum extra cost to be incurred in the best complete schedule of σ' for the unscheduled jobs by the exceeded completion time, δ_k, on machine M_k is $\delta_k \times \sum_{l \notin \sigma'} w_{lk}$.

Case 2 $\delta_k < 0$

Denote I_k as the maximum possible total idle time of the unscheduled jobs on machine M_k subject to σ'. The value of I_k is obtained by appending the reverse of Johnson's sequence, using only M_0 and M_k, of the unscheduled parts to σ'. Then, the weighted sum of completion times of the unscheduled parts subject to σ' will be $(|\delta_k| - I_k)^+ \times \sum_{l \notin \sigma'} w_{lk}$ less than that subject to σ''.

Let $K_1 = \{k : \delta_k > 0\}$ and $K_2 = \{k : \delta_k < 0\}$. So, we have the following condition (c) for σ' to dominate σ''.

$$(c) : \Delta + \sum_{k \in K_1} \Big(\delta_k \times \sum_{l \notin \sigma'} w_{lk} \Big) - \sum_{k \in K_2} \Big((|\delta_k| - I_k)^+ \times \sum_{l \notin \sigma'} w_{lk} \Big) \le 0.$$

Note that in the literature on $F2||\sum C_i$, e.g. Della Croce, Ghirardi, and Tadei (2002), the unweighted versions of condition (a) and condition (c) need to be satisfied simultaneously. In fact, condition (c) can replace conditions (a) and (b). That is, dominance condition (c) can be applied even if the test of condition (a) fails. The required time for testing the dominance condition is $O(mn)$, which is also linear, because m Johnson's sequences and m sorted lists of part weights can be obtained in advance.

Inspecting the structures of the schedules of the unconsidered jobs, we can further relax condition (c) because the difference between two makespan will not necessarily affect all the unscheduled jobs. If the second term can be less positive and the third term can be more negative, the dominance condition will become more likely to hold. There are $r = n - |\sigma| - 2$ jobs that remain unconsidered. For any machine index k, $1 \le k \le m$, let $\psi^k = (\psi_{1k}, \ldots, \psi_{rk})$ (respectively, $\psi'^k = (\psi'_{1k}, \ldots, \psi'_{rk})$) denote Johnson's sequence (respectively, the reverse of Johnson's sequence), considering M_0 and M_k only, of the unscheduled jobs. Following the notation in Cheng and Lin (2009), part J_{ik} is called positive (respectively, negative) if $p_{ik} \ge p_{i0}$ (respectively, $p_{ik} < p_{i0}$). Let $r' \le r$ denote the number of positive unscheduled jobs.

Case 1 $\delta_k > 0$.

Let $\eta^k = (\eta_{1k}, \ldots, \eta_{rk})$ be a sequence of the unscheduled jobs such that $\sigma'' \oplus \eta^k$ is the optimal schedule in the subtree rooted at partial schedule σ''. Assume $J_{\eta_{i_1 k}}, \ldots, J_{\eta_{i_r k}}, i_1 < \cdots < i_r$ are the positive jobs as they appear in η^k. Denote $I_{\psi_{sk}}$ as the idle time before job $J_{\psi_{sk}}$ on machine M_k in schedule $\sigma'' \oplus \psi^k$. Idle time $I_{\eta_{sk}}$ is similarly defined for the jobs in schedule $\sigma'' \oplus \eta^k$. Then, we have the following property on cumulative idle times:

LEMMA 2 *Inequality* $\sum_{s=1}^{r''} I_{\psi_{sk}} \le \sum_{s=1}^{r''} I_{\eta_{sk}}$ *holds for any* r'', $1 \le r'' \le r'$.

Proof The arguments are based upon the properties of Johnson's sequence. Assume $\psi_{1k} = \eta_{i_s k}$ for some specific position s in $\sigma'' \oplus \eta^k$. Because part $J_{\eta_{i_s k}}$ is positive, insertion of it in front of $\eta_{i_1 k}$ will not increase the idle time of any job in $\sigma'' \oplus \eta^k$. Continuing insertions, if necessary, without increasing any idle time, we come up with a sequence in which all the positive parts are sequenced by Johnson's rule as in ψ^k. While the positive parts are not preceded by any negative part in ψ^k, each positive part in the derived sequence may be preceded by one or more negative parts. Comparing the derived sequence and sequence ψ_{1k}, we readily know that the inequality holds. □

With Lemma 2, it can be easily shown that if $\delta_k > 0$ (or, $k \in K_1$), the extra cost for the complete schedules developed from σ' when compared with σ'' on machine M_k is no more than $\texttt{influence}_k$, which is computed by the following program segment. Variable ω_{sk}, $1 \le s \le r'$, is the s-th largest weight of the unconsidered parts on M_k.

/* when $k \in K_1$ */
$\texttt{influence}_k = 0$;
for $s = 1$ **to** r'
{
  $\delta_k = (\delta_k - I_{\psi_{sk}})^+$;
  if $\delta_k \le 0$ **then** abort the **for**-loop;
  else $\texttt{influence}_k = \texttt{influence}_k + \delta_k \times \omega_{sk}$;
}
if $\delta_k > 0$ **then** $\texttt{influence}_k = \texttt{influence}_k +$ the weights of all the unconsidered negative parts.

Case 2 $\delta_k < 0$.
Let $\eta'^k = (\eta'_{1k}, \ldots, \eta'_{rk})$ be a sequence of the unscheduled jobs such that $\sigma' \oplus \eta'^k$ is the optimal schedule in the subtree rooted at partial schedule σ'. Assume $J_{\eta'_{i_1 k}}, \cdots, J_{\eta'_{i_r k}}$, $i_1 < \ldots < i_r$ are the positive jobs as they appear in η'^k. Denote $I_{\psi'_{sk}}$ as the idle time before job $J_{\psi'_{sk}}$ on machine M_k in schedule $\sigma' \oplus \psi'^k$. Idle time $I_{\eta'_{sk}}$ is similarly defined for the jobs in schedule $\sigma' \oplus \eta'^k$. The following property follows:

LEMMA 3 *Inequality* $\sum_{s=1}^{r''} I_{\psi'_{sk}} \ge \sum_{s=1}^{r''} I_{\eta'_{sk}}$ *holds for any* r'', $1 \le r'' \le r'$.

Proof Assume $\psi'_{1k} = \eta'_{i_s k}$ for some specific position s in $\sigma' \oplus \eta'^k$. Because part $J_{\eta_{i_s k}}$ is positive, forward insertion of it in the position immediately following $\eta_{i_r k}$ will not decrease the idle time of any job in $\sigma' \oplus \eta'^k$. Continuing insertions, if necessary, without decreasing any idle time, we come up with a sequence where all the positive parts are sequenced by the reverse of Johnson's rule as those in ψ'^k. All the positive parts are preceded by all the negative parts in ψ'^k. Comparing the derived sequence and sequence ψ'_{1k}, we readily know that the cumulative idle times in the corresponding positions of positive jobs in ψ'_{1k} would be no less than those in the derived sequence. The inequality thus follows. □

With Lemma 3, if $\delta_k < 0$ (or, $k \in K_2$), the maximum offset of cost for the complete schedules developed from σ' when compared with σ'' on machine M_k is less than $\texttt{influence}_k$, which is computed by the following program segment.

* /when $k \in K_2$ */
$\texttt{influence}_k = 0$;
$\delta_k = (|\delta_k| - \sum_{s=1}^{r-r'} I_{\psi'_{sk}})^+$;
for $s = r - r' + 1$ **to** r
{
  $\delta_k = (\delta_k - I_{\psi'_{sk}})^+$;
  if $\delta_k \le 0$ **then** abort the **for**-loop;
  else $\texttt{influence}_k = \texttt{influence}_k + \delta_k \times \omega_{(r-s+1)k}$;
}

Incorporating the above two limits into the second and third terms of condition (c), we obtain an updated condition for σ' to dominate σ'' as follows:

$$(c') : \Delta + \sum_{k \in K_1} \texttt{influence}_k - \sum_{k \in K_2} \texttt{influence}_k \le 0.$$

When the input instance is given, Johnson's sequences and the reverse of Johnson's sequences, and sorted lists of part weights over all the machines can be calculated in advance. Therefore, $O(mn)$ timesuffices to examine the dominance condition (c').

The second dominance rule is developed for the situation where, subject to partial schedule σ, no further idle time will occur for the unscheduled jobs on all the machines.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

(d): Subject to partial schedule σ, if no idle time occurs for the unscheduled jobs and inequality (1) holds, then the node rooted at $\sigma \oplus (i, j)$ should be fathomed.

Similarly, the time required to test condition (d) is also $O(mn)$. Note that condition (c') and condition (d) do not subsume each other. Thus, they will both be embedded in the branch-and-bound algorithm.

5. LP-based approximation and performance analysis

In this section, we present an approximation algorithm for the refurbishing flowshop problem. The *performance ratio* r of an approximation algorithm $\mathcal{A}$ for a computationally intractable minimisation problem $\mathcal{P}$ is defined as $\max_{x \in X}\{Z(\sigma_A(x))/Z^*(x)\}$, where X is the set of all the input instances and $\sigma_A(x)$ is the solution given by algorithm $\mathcal{A}$ on input x. An algorithm has a performance ratio r is called an r-approximate algorithm. Gonzalez and Sahni (1978) proved that the performance ratio of any heuristic for the $Fm||\sum C_i$ problem is no more than m. Hoogeveen and Kawaguchi (1999) further proved that when $m = 2$, the performance ratio of the approximation algorithm of Gonzalez and Sahni (1978) can be reduced to $2\beta/(\alpha + \beta)$, where α denotes the minimum processing time of the $2n$ operations and β the maximum processing time of the $2n$ operations. These results are restricted to the unweighted version of the objective function. For the weighted case $Fm||\sum w_i C_i$, Schulz (1996) presented a $(2m - \frac{2m}{n+1})$-approximate heuristic. In this section, we present an analysis of the performance ratio based upon a linear programming (LP) formulation. Our analysis follows Schulz's approach to developing a performance ratio for $Fm||\sum w_i C_i$.

The linear programming relaxation uses completion times C_{ik} of *part* J_{ik} as decision variables. Our analysis is similar to the techniques employed by Correa and Schulz (2005); Lenstra, Shmoys, and Tardos (1990), Shmoys and Tardos (1993), and Hall et al. (1997), which used completion times C_j of *jobs* J_j. The weighted sum of completion times can be formulated in the following way, where the constraints ensure that the variables $C_{1k}, \ldots, C_{nk}$, $0 \le k \le m$, specify a feasible set of completion times:

Program (LP):

$$\text{Minimize} \sum_{i=1}^{n}\sum_{k=1}^{m} w_{ik} C_{ik}$$

subject to

$$C_{i0} \ge p_{i0}, \qquad 1 \le i \le n; \tag{2}$$

$$C_{ik} \ge C_{i0} + p_{ik}, \qquad 1 \le i \le n, 1 \le k \le m; \tag{3}$$

$$C_{ik} \ge C_{jk} + p_{ik} \text{ or } C_{jk} \ge C_{ik} + p_{jk}, \qquad 1 \le i \ne j \le n, 0 \le k \le m. \tag{4}$$

The first constraint set ensures that every operation must start after time $t = 0$ and constraints (3) require that the dismantling of an item must be completed before the processing of any of its decomposed parts. Constraints (4) specify a disjunctive machine environment, i.e. any machine can process at most one operation at a time. Queyranne (1993) showed that such disjunctive constraints describe the convex hull of the feasible completion time vectors by defining linear transformations of a supermodular polyhedron. Based on this concept, assuming without loss of generality that $C_{1k} \le \cdots \le C_{nk}$, $1 \le k \le m$, constraints (4) are replaced by the following inequalities:

$$\sum_{i=1}^{n} p_{ik} C_{ik} \ge \frac{1}{2}\left(\left(\sum_{i=1}^{n} p_{ik}\right)^2 + \sum_{i=1}^{n} p_{ik}^2\right). \tag{5}$$

Let C_{ik}^{LP} denote the completion time of product (part) J_{ik} on machine $M_k, 0 \le k \le m$, from the relaxed linear programming formulation. Applying Lemma 1 of Schulz (1996), we obtain relationships between the completion time of the product in position i and the sum of the processing times of its predecessors as follows:

$$\sum_{j=1}^{i} p_{j0} \le 2C_{i0}^{LP} - \frac{\sum_{j=1}^{i} p_{j0}^2}{\sum_{j=1}^{i} p_{j0}},$$

$$\sum_{j=1}^{i} p_{jk} \le 2C_{ik}^{LP} - \frac{\sum_{j=1}^{i} p_{jk}^2}{\sum_{j=1}^{i} p_{jk}}. \tag{6}$$

Consider the natural Heuristic H that schedules the products such that J_i precedes J_j if $C_{i0}^{LP} \le C_{j0}^{LP}$, and breaks ties arbitrarily. For any operation of product J_i on machine M_k in the flowshop environment, the completion time, denoted as

C_{ik}^H, in the solutions obtained by Heuristic H is bounded above as follows:

$$C_{ik}^H \leq \sum_{j=1}^{i} p_{j0} + \sum_{j=1}^{i} p_{jk}. \tag{7}$$

Let Z_{LP}^* denote the optimal solution value of the relaxed linear program, subject to constraints (2), (3) and (5). Since the LP relaxation provides a lower bound on the optimal objective function value of the $\mathrm{RF}(1,m)||C_{\max}$ problem, we have $Z_{LP}^* \leq Z^*$ and may use it for the performance analysis of Heuristic H. Combining inequalities (6) and (7), we obtain, for $1 \leq k \leq m$,

$$\begin{aligned} C_{ik}^H &\leq \sum_{j=1}^{i} p_{j0} + \sum_{j=1}^{i} p_{jk} \\ &\leq 2C_{i0}^{LP} - \frac{\sum_{j=1}^{i} p_{j0}^2}{\sum_{j=1}^{i} p_{j0}} + 2C_{ik}^{LP} - \frac{\sum_{j=1}^{i} p_{jk}^2}{\sum_{j=1}^{i} p_{jk}} \\ &\leq 4C_{i0}^{LP} - \frac{\sum_{j=1}^{i} p_{j0}^2}{\sum_{j=1}^{i} p_{j0}} - \frac{\sum_{j=1}^{i} p_{jk}^2}{\sum_{j=1}^{i} p_{jk}} - 2p_{ik}. \end{aligned} \tag{8}$$

Thus, summing inequality (8) with the weights incorporated over all the products and all the refurbishing machines, we obtain the following theorem:

THEOREM 6 *Heuristic H, based upon program LP, is a 4-approximation algorithm.*

From the negative terms in inequality (8), we can reduce the ratio of 4 by a certain amount. The following derivations lead to a function of problem size n for expressing the reduction from the ratio of 4. As shown in the supplementary `http://web.it.nctu.edu.tw/ bmtlin/IJPR50/proof.pdf`, the performance ratio Heuristic H can be reduced to $4 - \frac{4}{n+1}$.

6. Approximation methods and computational study

This section presents the development of heuristics and metaheuristic algorithms for producing approximate solutions for the problem. We adopt two metaheuristics, namely iterated local search (ILS) and ant colony optimization (ACO). We compare the performance of the WSPT rule, an NEH-based (Nawaz, Enscore, and Ham 1983) heuristic, a CDS-based (Campbell, Dudek, and Smith 1970) heuristic and the two metaheuristics through computational experiments. The WSPT dispatching heuristic constructs a product sequence in which product J_i precedes product J_j if

$$\frac{p_{i0} + \max_{k \in 1..m}\{p_{ik}\}}{\sum_{k=1}^{m} w_{ik}} \leq \frac{p_{j0} + \max_{k \in 1..m}\{p_{jk}\}}{\sum_{k=1}^{m} w_{jk}}.$$

Ties are broken arbitrarily. The WSPT sequence serves not only as the initial solution for the ILS approach but also the base for performance comparisons. The CDS and NEH algorithms were developed for minimising the makespan in a multiple-stage flowshop. They have also been widely adopted for minimising the total flow time in flowshop scheduling (Brah and Loo 1999; Gupta and Schaller 2006; Framinan, Leistenb, and Ruiz-Usanoa 2002). Lanna, Mosheiov, and Rinott (1998) adapted the CDS algorithm for minimising the total completion time in concurrent open shops. In our design, the NEH-based algorithm starts with a sorted list of jobs arranged in decreasing order of job loads, defined as $\sum_{k=0}^{m} p_{ik}$ for each job J_i. The CDS-based algorithm uses processing times aggregated from the processing times on the second-stage machines.

6.1 *ILS and ACO*

ILS (Baum 1986; Lourenço, Martin, and Stützle 2010) is a simple metaheuristic that can be applied to deal with general computationally intractable problems. It iteratively applies local searches to modifications of the current solution. Research has been done on applying ILS to flowshop problems, such as Stützle (1998) on the classical $Fm||C_{\max}$ problem, Yang, Kreipl, and Pinedo (2000) on a flexible flowshop with multi-stages in series and Dong, Huang, and Chen (2009) on permutation flowshop scheduling to minimise the total completion time. An ILS algorithm mainly consists of four procedures: generating an initial solution, modifying the current solution (perturbation), finding a local extreme (local minimum in our case) by a local search procedure and accepting a solution value that will be perturbed in the next iteration. In our experiments,

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Table 1. Optimal solutions, metaheuristic solutions and lower bounds for $n = 10$.

m	*OPT*	*ACO*		*ILS*		*LB*		$LB_{assignment}$	
	obj	obj	error%	obj	error%	obj	error%	obj	error%
	131,729	131,729	0.00	131,729	0.00	73,140	44.48	95,701	27.35
	173,768	173,768	0.00	173,768	0.00	123,516	28.92	130,066	25.15
	158,763	158,763	0.00	158,763	0.00	128,468	19.08	137,965	13.10
	213,213	213,213	0.00	213,213	0.00	155,939	26.86	163,908	23.12
	150,254	150,254	0.00	150,254	0.00	93,746	37.61	106,808	28.92
2	120,323	120,323	0.00	120,323	0.00	91,087	24.30	105,605	12.23
	110,718	110,718	0.00	110,718	0.00	54,956	50.36	73,377	33.73
	180,198	180,198	0.00	180,198	0.00	103,434	42.60	141,865	21.27
	143018	143,018	0.00	143,018	0.00	97,685	31.70	108,893	23.86
	99,787	99,787	0.00	99,787	0.00	60,269	39.60	74,253	25.59
	147,353	147,353	0.00	147,353	0.00	90,765	38.40	107,353	27.15
	314,452	314,452	0.00	314,452	0.00	187,758	40.29	221,113	29.68
	325,519	325,519	0.00	325,519	0.00	168,496	48.24	238,593	26.70
	297,807	297,807	0.00	297,807	0.00	165,601	44.39	206,403	30.69
	266,137	266,137	0.00	266,137	0.00	174,419	34.46	205,442	22.81
	339,231	339,231	0.00	339,231	0.00	207,093	38.95	269,014	20.70
4	235,669	235,669	0.00	235,669	0.00	132,692	43.70	197,410	16.23
	228,478	228,478	0.00	228,478	0.00	122,290	46.48	170,232	25.49
	303,543	303,543	0.00	303,543	0.00	174,014	42.67	253,335	16.54
	273,992	273,992	0.00	273,992	0.00	155,755	43.15	196,035	28.45
	273,164	273,164	0.00	273,164	0.00	152,304	44.24	200,658	26.54
	602,259	602,259	0.00	602,259	0.00	402,179	33.22	518,514	13.91
	698,103	698,103	0.00	698,103	0.00	560,287	19.74	661,761	5.21
	361,238	361,238	0.00	361,238	0.00	181,213	49.84	269,786	25.32
	538,764	538,764	0.00	538,764	0.00	368,600	31.58	455,015	15.54
	499,534	499,534	0.00	499,534	0.00	296,599	40.62	413,504	17.22
6	363,971	363,971	0.00	363,971	0.00	198,239	45.53	298,031	18.12
	406,053	406,053	0.00	406,053	0.00	202,655	50.09	308,583	24.00
	396,992	396,992	0.00	396,992	0.00	230,393	41.97	293,093	26.17
	411,057	411,057	0.00	4110,57	0.00	215,600	47.55	313,271	23.79
	537,595	537,595	0.00	5375,95	0.00	352,307	34.47	480,094	10.70

we used the WSPT-based heuristic to construct an initial solution even though the ILS algorithm could start with a random solution. The neighbourhood structure of a local search is defined by how solutions can be generated from a specific solution. For permutational solutions as in sequencing problems, a neighbourhood can be defined by swapping the positions of two jobs. A neighbourhood constructed from swapping two arbitrary jobs usually outperforms the neighbour defined by adjacent jobs in terms of solution quality and efficiency. For the perturbation procedure, we considered a series of swapping along with swapping of two sectors in a sequence. Such an arrangement prevents cycling that may occur in the succeeding local search procedure. The choice of the acceptance criterion may be critical to the performance of the ILS algorithm. For example, accepting every new local optimum is similar to a random walk over the local optima, which could incur extra computing efforts. Martin and Otto (1996) suggested a simulated annealing type of acceptance criterion to improve the computational results for the travelling salesman problem. We simply chose to take the best solution as the acceptance criterion for each iteration, and then applied the succeeding perturbation to the sequence of the accepted solution. In order to achieve diversification of the search and to avoid the trap of local minima, we also shuffled the sequence after a certain number of iterations.

The second metaheuristic approach we adopted was ant colony optimisation (ACO), which was proposed by Colorni, Dorigo, and Maniezzo (1991) and Dorigo (1992). The fundamental concept of ACO is the indirect communication of a colony of artificial ants mediated by pheromone trails with a knowledge-sharing mechanism during the food-seeking process. ACO has been widely deployed to deal with computationally intractable optimisation problems. Recent research on applying ACO to flowshop scheduling includes T'kindt et al. (2002), Rajendran and Ziegler (2004), Gajpal and Rajendran (2006), and Lin et al. (2008). In this paper, we used the ACO algorithm with the dual pheromone control developed by Lin et al. (2008) for flowshop scheduling. In addition to the pheromone used to measure the preferences of different job orderings, a dual pheromone mechanism introduces a new type of pheromone for dictating the preference of assigning a certain job to some specific position in a schedule. The second type of pheromone is motivated by the fact that ants deposit multiple types of pheromone for different purposes when they travel. For scheduling problems, the second type of pheromone

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Table 2. Optimal solutions, metaheuristic solutions and lower bounds for $n = 15$.

m	OPT	ACO		ILS		LB		$LB_{assignment}$	
	obj	obj	error%	obj	error%	obj	error%	obj	error%
2	292,477	292,477	0.00	292,477	0.00	191,807	34.42	232,145	20.63
	205,646	205,646	0.00	205,646	0.00	127,562	37.97	163,667	20.41
	226,830	226,830	0.00	226,830	0.00	114,629	49.46	161,488	28.81
	235,085	235,085	0.00	235,085	0.00	149,392	36.45	167,465	28.76
	265,881	265,881	0.00	265,881	0.00	159,173	40.13	193,167	27.35
	307,280	307,280	0.00	307,280	0.00	183,239	40.37	226,727	26.21
	314,861	314,861	0.00	314,861	0.00	208,433	33.80	241,872	23.18
	280,473	280,558	0.03	280,511	0.01	173,892	38.00	234,248	16.48
	285,593	285,593	0.00	285,593	0.00	179,088	37.29	207,464	27.36
	190,681	190,681	0.00	190,681	0.00	100,101	47.50	132,923	30.29
4	626,733	626,733	0.00	626,733	0.00	329,881	47.36	467,415	25.42
	691,961	691,961	0.00	691,961	0.00	425,311	38.54	586,451	15.25
	546,877	546,877	0.00	546,877	0.00	304,008	44.41	434,190	20.61
	783,626	783,626	0.00	783,626	0.00	515,920	34.16	641,914	18.08
	582,121	582,121	0.00	582,121	0.00	329,525	43.39	431,412	25.89
	641,345	641,345	0.00	641,345	0.00	387,158	39.63	501,411	21.82
	516,403	516,403	0.00	516,403	0.00	288,380	44.16	402,414	22.07
	583,101	583,101	0.00	583,101	0.00	339,086	41.85	439,692	24.59
	591,532	591,532	0.00	591,532	0.00	350,397	40.76	455,147	23.06
	812,557	812,557	0.00	812,557	0.00	514,655	36.66	679,027	16.43
6	976,377	976,377	0.00	976,377	0.00	515,168	47.24	747,186	23.47
	899,796	899,796	0.00	899,796	0.00	518,443	42.38	721,230	19.85
	874,100	874,100	0.00	874,100	0.00	451,689	48.33	663,524	24.09
	1,101,410	1,101,410	0.00	1,101,410	0.00	733,557	33.40	928,860	15.67
	724,410	724,410	0.00	724,410	0.00	363,122	49.87	524,033	27.66
	855,570	855,570	0.00	855,570	0.00	412,994	51.73	623,749	27.10
	982,586	982,586	0.00	982,586	0.00	546,303	44.40	735,041	25.19
	899,860	899,860	0.00	899,860	0.00	522,414	41.94	714,898	20.55
	902,933	903,052	0.01	902,933	0.00	446,072	50.60	613,085	32.10
	1,325,830	1,325,920	0.01	1,325,830	0.00	894,259	32.55	1,159,758	12.53

signifies the observation that the preference of letting job J_i precede J_j in the front part of a schedule can be different from that in the rear part. This phenomenon is especially common in flowshop scheduling.

6.2 *Computational experiments and results*

The computational experiments consisted of two parts. The first part was dedicated to small-scale problems for which optimal solutions could be derived. The second part contained experiments on instances involving more jobs so as to examine the performance of the approximation heuristics and metaheuristics.

For the small-scale problems, optimal solutions were obtained by the branch-and-bound approach. We implemented the lower bounds and the dominance rules proposed in this paper and used them to examine if further branching from a partial solution could be curtailed.

For finite values of n and $1 + m$, we computed the lower bounds and heuristic solutions for a set of randomly generated test problems, which were generated as follows: (1) number of products $n \in \{10, 15, 20, 50, 100\}$; (2) number of machines $1 + m \in \{3, 5, 7\}$; (3) processing times p_{ik} were independently drawn from a uniform integer distribution over the interval [1, 100]; and (4) the weights w_{ik} associated with each operation of various products were selected from the interval [1, 50].

Ten sets of test problems were generated and calculated for each combination of n and m. The results are summarised in Tables 1–5. Relative errors between the approximate solutions and the exact solutions are presented for the small-scale problems. As the number of products increased, the computation of the exact solutions by the branch-and-bound algorithm became impossible, so the approximate solutions produced by the NEH-based algorithm, the CDS-based algorithms, and the two metaheuristics were compared with the solutions generated by the WSPT-based heuristic to show the improvement ratios.

The metaheuristics easily produced optimal solutions for the tested problem instances when the number of products was no more than 15 with various numbers of machines, as illustrated in Tables 1 and 2. The branch-and-bound approach

Table 3. Approximate solutions for $n = 20$.

m	$WSPT$	NEH	$impr\%$	CDS	$impr\%$	ILS	$impr\%$	ACO	$impr\%$
	670,288	639,860	4.54	738,585	−10.19	602,406	10.13	602,406	10.13
	537,860	504,623	6.18	668,685	−24.32	486,452	9.56	487,124	9.43
	544,880	495,779	9.01	622,856	−14.31	479,356	12.03	478,979	12.09
	427,939	424,475	0.81	520,386	−21.60	406,926	4.91	407,099	4.87
	497,293	479,539	3.57	653,026	−31.32	451,337	9.24	451,734	9.16
2	463,921	432,979	6.67	571,813	−23.26	429,097	7.51	429,097	7.51
	423,666	412,408	2.66	545,501	−28.76	401,247	5.29	401,247	5.29
	458,519	456,194	0.51	602,365	−31.37	440,459	3.94	441,179	3.78
	586,237	610,271	−4.10	676,037	−15.32	574,683	1.97	574,683	1.97
	410,884	424,662	−3.35	538,856	−31.15	394,087	4.09	395,465	3.75
	994,375	971,062	2.34	1,141,600	−14.81	950,471	4.42	950,471	4.42
	1,070,630	1,052,890	1.66	1,220,350	−13.98	1,019,660	4.76	1,019,660	4.76
	1,146,570	1,123,790	1.99	1,353,490	−18.05	1,106,300	3.51	1,106,300	3.51
	844,314	800,571	5.18	1,118,800	−32.51	766,084	9.27	766,119	9.26
	1,076,760	1,076,160	0.06	1,271,900	−18.12	1,052,830	2.22	1,057,530	1.79
4	980,884	952,010	2.94	1,078,970	−10.00	930,460	5.14	929,970	5.19
	1,086,890	1,060,230	2.45	1,192,920	−9.76	1,010,440	7.03	1,010,400	7.04
	941,458	919,687	2.31	1,089,270	−15.7	886,918	5.79	886,918	5.79
	768,041	758,072	1.30	954,425	−24.27	747,148	2.72	746,520	2.80
	917,818	913,123	0.51	1,095,850	−19.4	882,199	3.88	882,199	3.88
	1,375,420	1,361,960	0.98	1,635,800	−18.93	1,344,510	2.25	1,344,220	2.27
	1,483,500	1,464,040	1.31	1,636,610	−10.32	1,442,990	2.73	1,442,990	2.73
	1,803,360	1,685,800	6.52	1,991,850	−10.45	1,642,250	8.93	1,642,250	8.93
	1,358,270	1,423,560	−4.81	1,498,790	−10.35	1,328,520	2.19	1,327,580	2.26
	1,499,460	1,423,830	5.04	1,648,240	−9.92	1,376,700	8.19	1,377,070	8.16
6	1,871,360	1,806,080	3.49	2,015,540	−7.70	1,782,220	4.76	1,781,710	4.79
	1,899,780	1,793,220	5.61	2,030,430	−6.88	1,781,460	6.23	1,781,460	6.23
	1,496,940	1,495,150	0.12	1,684,450	−12.53	1,438,380	3.91	1,438,380	3.91
	2,008,780	1,946,200	3.12	2,084,010	−3.75	1,907,590	5.04	1,908,120	5.01
	1,696,320	1,704,380	−0.48	1,982,520	−16.87	1,644,380	3.06	1,644,380	3.06

required double-digit seconds to tackle problem instances with fewer than ten jobs, and took only seconds when the number of machines were fewer than seven. The time to compute the optimal solutions multiplied when the number of products increased. We did not successfully solve instances of more than 15 jobs within the 180-min running time limit.

Table 1 shows that significant optimal results are obtained by ACO and ILS while the performance of the lower bounds appears comparatively weak. Similar results could be expected when the number of products increases. It is noted that the improvement of lower bound $LB_{assignment}$ over lower bound LB is significant, although the former is too computationally demanding to be embedded in the branch-and-bound algorithm. It is worth noting that when the number of machines equals two, the studied problem reduces to the classical two-machine flowshop problem, $F2||\sum C_i$. Research on the two-machine flowshop problem has already indicated difficulty in obtaining tight lower bounds for $F2||\sum C_i$ (e.g. Ahmadi and Bagchi 1990; Della Croce, Ghirardi, and Tadei 1996, 2002; Hoogeveen and Kawaguchi 1999; van de Velde 1990; Lin and Wu 2005; Hoogeveen, van Norden, and van de Velde 2006). Therefore, such a time-consuming optimal solution seeking process for the $\mathrm{RF}(1, m)||\sum w_{ik}C_{ik}$ problem is to be expected. Meanwhile, in order to eschew the complicated mathematical calculations required by Lagrangian relaxation, we derived our lower bound for ease of comprehension and simplicity of implementation. Although the lower bound could not help the branch-and-bound algorithm to solve problems with more than 15 jobs in the computational experiments, such a lower bound assuredly improved the computational effort for finding the optimal solutions.

When the number of products exceeded 15, we turned to evaluating the performance of the proposed approximation methods in terms of their improvement ratios over the objective values produced by a simple heuristic. Due to the resemblance of the problem under study to the classical flowshop problem, the WSPT-based algorithm was used as a proper approximation approach. We consequently used the sequences reported by the WSPT-based method as initial solutions for ILS and also took the solution values as the basis for comparisons.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Table 4. Approximate solutions for $n = 50$.

m	$WSPT$	NEH	$impr\%$	CDS	$impr\%$	ILS	$impr\%$	ACO	$impr\%$
	3,072,940	3,026,700	1.50	4,216,630	−37.22	2,929,740	4.66	2,931,930	4.59
	2,954,470	2,892,690	2.09	4,000,700	−35.41	2,781,480	5.86	2,782,060	5.84
	2,044,900	2,029,470	0.75	2,962,220	−44.86	1,918,770	6.17	1,923,090	5.96
	2,769,290	2,762,930	0.23	3,794,270	−37.01	2,599,510	6.13	2,600,700	6.09
	2,938,900	2,765,680	5.89	3,759,860	−27.93	2,678,680	8.85	2,678,950	8.85
2	2,684,630	2,517,780	6.22	3,808,630	−41.87	2,481,200	7.58	2,481,620	7.56
	2,579,490	2,558,230	0.82	3,589,840	−39.17	2,418,580	6.24	2,420,230	6.17
	2,563,860	2,486,800	3.01	3,766,180	−46.89	2,369,080	7.60	2,367,110	7.67
	2,780,000	2,718,420	2.22	3,753,990	−35.04	2,642,510	4.95	2,646,800	4.79
	2,057,710	2,068,420	−0.52	3,246,010	−57.75	1,954,310	5.03	1,954,180	5.03
	5,835,570	5,797,820	0.65	6,942,570	−18.97	5,545,400	4.97	5,544,280	4.99
	5,691,870	5,464,780	3.99	6,929,170	−21.74	5,298,320	6.91	5,302,600	6.84
	5,283,960	5,289,830	−0.11	6,069,790	−14.87	5,092,120	3.63	5,089,380	3.68
	5,199,380	5,045,570	2.96	6,458,850	−24.22	4,948,630	4.82	4,957,630	4.65
	6,391,050	6,315,030	1.19	7,769,400	−21.57	6,210,480	2.83	6,211,270	2.81
4	6,227,320	6,118,480	1.75	7,741,960	−24.32	5,974,140	4.07	5,976,120	4.03
	5,788,820	5,650,730	2.39	7,093,330	−22.53	5,503,520	4.93	5,501,730	4.96
	5,768,990	5,519,360	4.33	6,689,480	−15.96	5,430,640	5.86	5,439,030	5.72
	4,949,010	4,919,530	0.60	6,129,440	−23.85	4,750,040	4.02	4,744,940	4.12
	6,542,810	6,148,780	6.02	7,291,740	−11.45	6,026,210	7.90	6,018,690	8.01
	9,757,180	9,728,580	0.29	11,134,400	−14.11	9,454,240	3.10	9,457,860	3.07
	9,434,230	9,304,660	1.37	10,842,700	−14.93	9,155,210	2.96	9,157,920	2.93
	8,100,640	8,041,130	0.73	9,351,820	−15.45	7,775,860	4.01	7,783,070	3.92
	8,893,470	8,790,880	1.15	10,792,300	−21.35	8,565,560	3.69	8,575,410	3.58
	9,587,830	9,241,780	3.61	11,403,500	−18.94	8,986,370	6.27	9,004,260	6.09
6	8,894,430	8,741,740	1.72	10,277,800	−15.55	8,380,780	5.77	8,385,820	5.72
	9,062,880	8,918,550	1.59	10,615,900	−17.14	8,575,330	5.38	8,581,570	5.31
	9,167,490	9,013,850	1.68	10,112,200	−10.31	8,643,970	5.71	8,636,250	5.79
	8,015,840	7,874,370	1.76	9,611,100	−19.90	7,700,750	3.93	7,712,080	3.79
	8,794,100	8,560,540	2.66	10,027,800	−14.03	8,404,400	4.43	8,404,920	4.43

Tables 3–5 exhibit the comparisons among the four approaches in various settings, in terms of their improvement ratios over the solution values produced by the WSPT-based heuristic. For most of the test instances, the NEH-based algorithm outperformed the WSPT-based heuristic. The performance of the CDS-based algorithm was, however, found wanting. The CDS algorithm was originally designed for makespan minimisation ($Fm||C_{\max}$) based on aggregation of the processing times along the serial stages. In the refurbishing flowshop, aggregation is performed using the processing times on parallel machines. Therefore, the aggregation concept used in the CDS algorithm is not effective in the refurbishing flowshop. With regard to ILS and ACO, Tables 1 and 2 show that they were able to produce optimal solution values in the experiments when the problem size was small. It is evident from Tables 3 to 5 that ILS and ACO have comparable performance. The overall computational efforts favour ILS over ACO. Despite fluctuations in their improvement outcomes, both ILS and ACO can meliorate the initial WSPT-based heuristic outcomes by up to 10%. While the heuristic method may still deliver admissible objective values for large-scale problems, the computational efforts, especially of ILS, are surprisingly insignificant.

The improvement in solution quality obtained by the metaheuristics suggests that metaheuristics can be applied to solve large-scale problems because of their low-cost implementation and maintenance, as well as their demonstrated ability in producing solutions of adequate quality. Although we may not be able to firmly declare the expediency or superiority of ILS or ACO in tackling the problem under study, simplicity in representation and flow control favour ILS for implementation.

7. Conclusions

Recycling has made an increasing impact on human life as a result of growing concern about the environmental damage caused by the manufacturing industry. We propose a refurbishing model as a two-stage flowshop consisting of one common

Table 5. Approximate solutions for $n = 100$.

m	$WSPT$	NEH	$impr\%$	CDS	$impr\%$	ILS	$impr\%$	ACO	$impr\%$
	10,240,900	10,022,800	2.13	14,486,600	−41.46	9,628,070	5.98	9,616,690	6.10
	11,132,900	10,617,700	4.63	14,801,100	−32.95	10,067,000	9.57	10,067,900	9.57
	9,619,300	9,509,040	1.15	13,344,600	−38.73	9,160,850	4.77	9,161,290	4.76
	9,026,010	8,675,470	3.88	12,427,800	−37.69	8,287,780	8.18	8,293,550	8.11
	10,312,000	10,027,200	2.76	13,759,100	−33.43	9,802,580	4.94	9,819,260	4.78
2	10,405,000	10,299,600	1.01	14,256,000	−37.01	10,028,800	3.62	10,029,500	3.61
	9,067,820	9,146,930	−0.87	13,922,400	−53.54	8,703,930	4.01	8,705,630	3.99
	9,057,190	8,927,760	1.43	13,092,500	−44.55	8,691,440	4.04	8,710,920	3.82
	10,425,100	10,272,800	1.46	14,673,600	−40.75	9,991,610	4.16	9,993,390	4.14
	10,119,300	10,145,000	−0.25	14,818,600	−46.44	9,466,130	6.45	9,463,040	6.49
	23,864,300	23,375,800	2.05	28,333,500	−18.73	22,749,600	4.67	22,760,000	4.63
	21,711,500	21,338,000	1.72	27,617,500	−27.20	20,527,500	5.45	20,559,700	5.31
	21,608,300	21,417,700	0.88	27,667,500	−28.04	20,642,800	4.47	20,697,800	4.21
	24,538,700	24,207,500	1.35	30,016,500	−22.32	22,986,300	6.33	22,980,600	6.35
	22,727,800	21,686,300	4.58	26,562,900	−16.87	21,223,400	6.62	21,216,200	6.65
4	23,384,400	22,943,600	1.89	28,956,300	−23.83	22,361,000	4.38	22,367,100	4.35
	24,970,000	24,314,600	2.62	29,502,700	−18.15	23,395,800	6.30	23,383,100	6.36
	25,000,700	24,467,900	2.13	29,730,300	−18.92	23,404,100	6.39	23,369,500	6.52
	23,019,800	22,258,500	3.31	27,959,000	−21.46	21,570,900	6.29	21,601,400	6.16
	23,428,900	22,959,200	2.00	28,908,000	−23.39	22,141,400	5.50	22,239,100	5.08
	35,410,600	33,573,400	5.19	38,855,800	−9.73	32,963,900	6.91	32,982,900	6.86
	32,297,800	31,090,100	3.74	38,163,000	−18.16	30,547,600	5.42	30,547,300	5.42
	36,039,800	34,472,700	4.35	40,989,400	−13.73	33,733,900	6.40	33,783,100	6.26
	34,809,500	34,347,100	1.33	40,915,700	−17.54	33,757,800	3.02	33,859,700	2.73
	33,875,900	32,287,300	4.69	37,851,000	−11.73	31,376,100	7.38	31,370,200	7.40
6	36,821,400	36,005,600	2.22	42,804,300	−16.25	35,367,700	3.95	35,486,300	3.63
	34,000,000	33,540,900	1.35	38,664,200	−13.72	32,574,200	4.19	32,570,400	4.20
	32,925,500	32,183,900	2.25	38,927,600	−18.23	31,452,100	4.47	31,535,600	4.22
	33,394,600	32,761,300	1.90	40,044,200	−19.91	32,207,500	3.55	32,205,600	3.56
	35,337,300	33,598,200	4.92	40,486,200	−14.57	32,819,200	7.13	32,878,500	6.96

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

dismantling machine, followed by several independent dedicated machines that handle the different disassembled parts. The objective function depends on operation-based completion times, rather than the objective function involving job-based completion times that has been commonly adopted in decades of scheduling research. While the studied $\mathtt{RF}(1, m)||\sum w_{ik}C_{ik}$ problem is strongly NP-hard, we examine several special cases and make some interesting observations on the optimality properties of the solutions. We identify an open problem in the case where the dismantling operations are negligible. We propose a lower bound on the optimal solution and dominance rules. In view of the difficulty in obtaining a tight lower bound even for the classical $F2||\sum w_iC_i$ problem, we consider the proposed lower bound reasonably helpful in alleviating the computational burden required by complicated mathematical calculations and in facilitating the implementation of exact and approximate solution algorithms. Such effects, along with the comparisons of heuristic solutions, are demonstrated in the computational results. We propose an LP-based heuristic algorithm and show its performance ratio to be less than 4. We develop two metaheuristics, based on ILS and ACO, an NEH-based heuristic, and a CDS-based heuristic to produce quality approximate solutions in a reasonable time. Computational results show that ACO, ILS and the NEH-based heuristic perform significantly better than the simple, intuitive WSPT rule commonly deployed to treat the classical two-machine flowshop scheduling problem. The solutions obtained by the CDS-based heuristic are nevertheless less impressive.

For further research, we note that the use of operation-based objective functions in scheduling research is justified from the observation that any item, be it an WIP part or a final product, has its own identity and thus possesses its own value and cost, regardless of its position in the value chain. Therefore, operation-based objective functions open up a new avenue for scheduling research. Specific to the problem setting studied in the paper, we highlight two potential research problems. The first is to incorporate the concept of flow time in the objective function. The flow time of a part is defined as the difference between the time it is refurbished and the time at which the product it belongs to is dismantled. The second topic is to consider missing second-stage operations. This setting arises when not all the products produce the full set of the parts. When the objective function is the total (weighted) completion time, the assumptions of missing operations and zero processing time lead to significantly different results for the corresponding scheduling problems.

Acknowledgements

Lin was supported in part by the National Science Council of Taiwan under the Grant number NSC100-2410-H-009-015-MY2.

Note

1. We thank Dr S.V. Sevastyanov (the Sobolev Institute of Mathematics, Russia) for discussions on this case.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

References

Ahmadi, R. H., and U. Bagchi. 1990. "Improved Lower Bounds for Minimizing the Sum of Completion Times of n Jobs Over m Machines in a Flow Shop." *European Journal of Operational Research* 44 (3): 331–336.

Alcalay, J. A., and E. S. Buffa. 1963. "A Proposal for a General Model of a Production System." *International Journal of Production Research* 2 (1): 73–88.

Andrew-Munot, M., and R. N. Ibrahim. 2012. "Development and Analysis of Mathematical and Simulation Models of Decision-making Tools for Remanufacturing." *Production Planning and Control*. doi:10.1080/09537287.2012.654667.

Atasu, A., and S. Centinkaya. 2006. "Lot Sizing for Optimal Collection and Use of Remanufacturable Returns Over a Finite Life-cycle." *Production and Operations Management* 15 (4): 473–487.

Baum, E. B. 1986. *Iterated descent: A better algorithm for local search in combinatorial optimization problems. Manuscript*.

Brah, S. A., and L. L. Loo. 1999. "Heuristics for Scheduling in a Flow Shop with Multiple Processors." *European Journal of Operational Research* 113 (1): 113–122.

Campbell, H. G., R. A. Dudek, and M. L. Smith. 1970. "A Heuristic Algorithm for the n-Job m-Machine Sequencing Problem." *Management Science* 16 (10): 630–637.

Charlton, J. M., and C. C. Death. 1968. "A General Method for Machine Scheduling." *International Journal of Production Research* 7 (3): 207–217.

Cheng, T. C. E., M. Y. Kovalyov, and B. M. T. Lin. 1997. "Single Machine Scheduling to Minimize the Batch Delivery and Job Earliness Penalty." *SIAM Journal on Optimization* 7 (2): 547–559.

Cheng, T. C. E., and B. M. T. Lin. 2009. "Johnson's Rule, Composite Jobs and the Relocation Problem." *European Journal of Operational Research* 192 (3): 1008–1013.

Cheng, T. C. E., B. M. T. Lin, and Y. Tian. 2009. "Scheduling of a Two-stage Differentiation Flowshop to Minimize Weighted Sum of Machine Completion Times." *Computers and Operations Research*. 36 (11): 3031–3040.

Cheng, T. C. E., B. M. T. Lin, and A. Toker. 2000. "Flowshop Batching and Scheduling to Minimize the Makespan." *Naval Research Logistics* 47 (2): 128–144.

Colorni, A., M. Dorigo, and V. Maniezzo. 1991. *The Ant System: An Autocatatlytic Process*. Technical Report 91–016, Italy: Politecnico di Milano.

Conway, R. W., W. L. Maxwell, and L. W. Miller. 1967. *Theory of Scheduling*. Reading, MA: Addison-Wesley.

Correa, J. R. and A. S. Schulz. 2005. "Single-Machine Scheduling with Precedence Constraints." *Operations Research* 30 (4): 1005–1021.

Della Croce, F., M. Ghirardi, and R. Tadei. 2002. "An Improved Branch-and-bound Algorithm for the Two Machine Total Completion Time Flow Shop Problem." *European Journal of Operational Research* 139 (2): 293–301.

Della Croce, F., V. Narayan, and R. Tadei. 1996. "The Two-machine Total Completion Time Flow Shop Problem." *European Journal of Operational Research* 90 (2): 227–237.

Dong, X., H. Huang, and P. Chen. 2009. "An Iterated Local Search Algorithm for the Permutation Flowshop Problem with Total Flowtime Criterion." *Computers and Operations Research* 36 (5): 664–1669.

Dorigo, M., 1992. "Optimization, Learning and Natural Algorithms." PhD Thesis, Politecnico di Milano, Italy.

Environmental Leader. 2011. "Dell Closing in on Recycling Goal." Accessed September 10, 2012. http://www.environmentalleader.com/2011/04/19/dell-closing-in-on-recycling-goal

Ferguson, M. E., and L. B. Toktay. 2006. "The Effect of Competition on Recovery Strategies." *Production and Operations Management* 15 (3): 351–368.

Ferrer, G., and J. M. Swaminathan. 2006. "Managing New and Remanufactured Products." *Management Science* 52 (1): 15–26.

Ferrer, G., and J. M. Swaminathan. 2010. "Managing New and Differentiated Remanufactured Products." *European Journal of Operational Research* 203 (2): 370–379.

Framinan, J. M., R. Leistenb, and R. Ruiz-Usanoa. 2002. "Efficient Heuristics for Flowshop Sequencing with the Objectives of Makespan and Flowtime Minimisation." *European Journal of Operational Research* 141 (3): 559–569.

Gajpal, Y., and C. Rajendran. 2006. "An Ant-colony Optimization Algorithm for Minimizing the Completion-time Variance of Jobs in Flowshops." *International Journal of Production Economics* 101 (2): 259–272.

Galbreth, M. R., and J. D. Blackburn. 2006. "Optimal Acquisition and Sorting Policies for Remanufacturing." *Production and Operations Management* 15 (3): 384–392.

Gantt, H. L. 1919. "Efficiency and Democracy." *Transactions of the American Society of Mechanical Engineers* 40: 799–808.

Garey, M. R., and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. San Francisco, CA: Freedman.

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. "The Complexity of Flowshop and Jobshop Scheduling." *Mathematics of Operations Research* 1 (2): 117–129.

Georgiadis, P., D. Vlachos, and G. Tagaras. 2006. "The Impact of Product Lifecycle on Capacity Planning of Closed-loop Supply Chains with Remanufacturing." *Production and Operations Management* 15 (4): 514–527.

Gonzalez, T., and S. Sahni. 1978. "Flowshop and Jobshop Schedules: Complexity and Approximation." *Operations Research* 26 (1): 36–52.

Gupta, J. N. D. 1968. "A General Method for the $n \times M$ Flowshop Scheduling Problem." *International Journal of Production Research* 7 (3): 241–247.

Gupta, J. N. D., and J. E. Schaller. 2006. "Minimizing Flow Time in a Flow-line Manufacturing Cell with Family Setup Times." *Journal of the Operational Research Society* 57 (2): 163–176.

Hall, L. A. 1996. "Approximation Algorithms for Scheduling." In *Approximation Algorithms for NP-hard Problems*, edited by Dorit S. Hochbaum, 1–45. Boston, MA: PWS.

Herrmann, J. W. and C. Y. Lee. 1992. Three-Machine Look-ahead Scheduling Problems. Research Report 92–93, Department of Industrial Engineering, University of Florida, Florida.

Hoogeveen, J. A., and T. Kawaguchi. 1999. "Minimizing Total Completion Time in a Two-machine Flowshop: Analysis of Special Cases." *Mathematics of Operations Research* 24 (4): 887–913.

Hoogeveen, H., L. van Norden, and S. L. van de Velde. 2006. "Lower Bounds for Minimizing Total Completion Time in a Two-machine Flow Shop." *Journal of Scheduling* 9 (6): 559–568.

Hoogeveen, J. A., and S. L. van de Velde. 1995. "Stronger Lagrangian Bounds by Use of Slack Variables: Application to Machine Scheduling Problems." *Mathematical Programming* 70 (1–3): 173–190.

Hall, L. A., A. S. Schulz, D. B. Shmoys, and J. Wein. 1997. "Scheduling to Minimize Average Completion Time: Off-line and On-line Approximation Algorithms." *Mathematics of Operations Research* 22 (3): 513–544.

HP Environmental History. 2012. "Milestones." Accessed September 10, 2012. http://www8.hp.com/us/en/hp-information/environment/our-history.html

Huang, T. C., and B. M. T. Lin. 2013. "Batch Scheduling in Differentiation Flow Shops for Makespan Minimisation." *International Journal of Production Research*. 51 (17): 5073–5082.

Hwang, C.-L., F. A. Tillman, and L. T. Fan. 1967. "Optimization of Production Scheduling." *International Journal of Production Research* 5 (4): 341–346.

Ignall, E., and L. E. Schrage. 1965. "Application of the Branch-and-bound Technique to Some Flow-shop Scheduling Problems." *Operations Research* 13: 400–412.

Johnson, S. M. 1954. "Optimal Two- and Three-stage Production Schedules with Setup Times Included." *Naval Research Logistics Quarterly* 1 (1): 61–67.

Karamouzian, A., E. Teimoury, and M. Modarres. 2011. "A Model for Admission Control of Returned Products in a Remanufacturing Facility Using Queuing Theory." *International Journal of Advanced Manufacturing Technology* 54 (1–4): 403–412.

Ketzenberg, M. E., and G. C. Souza. 2003. "Mixed Assembly and Disassembly Operations for Remanufacturing." *Production and Operations Management* 12 (3): 320–335.

Lanna, A., G. Mosheiov, and Y. Rinott. 1998. "Asymptotic Optimality in Probability of a Heuristic Schedule for Open Shops with Job Overlaps." *Operations Research Letters* 22 (2–3): 63–68.

Lee, C. Y., T. C. E. Cheng, and B. M. T. Lin. 1993. "Minimizing the Makespan in Three-machine Assembly Type Flow Shop Problem." *Management Science* 39 (5): 616–625.

Lenstra, J. K., and A. H. G. Rinnooy Kan. 1977. "Complexity of Machine Scheduling Problems." *Annals of Discrete Mathematics* 1: 343–362.

Lenstra, J. K., D. B. Shmoys, and E. Tardos. 1990. "Approximation Algorithms for Scheduling Unrelated Parallel Machines." *Mathematical Programming* 46 (1–3): 259–271.

Leung, J. Y.-T., H. Li, and M. Pinedo. 2007. "Scheduling Orders for Multiple Product Types to Minimize Total Weighted Completion Time." *Discrete Applied Mathematics* 155 (8): 945–970.

Lin, B. M. T., and T. C. E. Cheng. 2011. "Concurrent Open Shop Scheduling with Centralized and Decentralized Batching Policies." *Naval Research Logistics* 58 (1): 17–27.

Lin, B. M. T., and F. J. Hwang. 2011. "Total Completion Time Minimization in a 2-Stage Differentiation Flowshop with Fixed Sequences Per Job Type." *Information Processing Letters* 111 (5): 208–212.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014

Lin, B. M. T., and A. V. Kononov. 2007. “Customer Order Scheduling to Minimize the Number of Late Jobs.” *European Journal of Operational Research* 183 (2): 944–948.
Lin, B. M. T., C. Y. Lu, S. J. Shyu, and C. Y. Tsai. 2008. “Development of New Features of Ant Colony Optimization for Flowshop Scheduling.” *International Journal of Production Economics* 112 (2): 742–755.
Lin, B. M. T., and J. M. Wu. 2005. “A Simple Lower Bound for Two Machine Flowshop Scheduling to Minimize Total Completion Time.” *Asia Pacific Journal of Operational Research* 22 (3): 391–408.
Lin, B. M. T., and J. M. Wu. 2006. “Bicriteria Scheduling in a Two-machine Permutation Flowshop.” *International Journal of Production Research* 44 (12): 2299–2312.
Liu, Y. C., K. T. Fang, and B. M. T. Lin. 2013. “A branch-and-bound Algorithm for Makespan Minimization in Differentiation Flow Shops.” *Engineering Optimization*. doi:10.1080/0305215X.2012.737783.
Lourenço, H. R., O. C. Martin, and T. Stützle. 2010. “Iterated Local Search: Framework and Applications.” In *Handbook of Metaheuristics*, edited by M. Gendreau, J.-Y. Potvin, 321–354. New York: Springer.
Martin, O. C., and S. W. Otto. 1996. “Combining Simulated Annealing with Local Search Heuristics.” *Annals of Operations Research* 63 (1): 57–75.
Mastrolilli, M., M. Queyranne, A. S. Schulz, O. Svensson, and N. A. Uhan. 2010. “Minimizing the Sum of Weighted Completion Times in a Concurrent Open Shop.” *Operations Research Letters* 38 (5): 390–395.
Maxwell, W. L. 1964. “The Scheduling of Single Machine Systems: A Review.” *International Journal of Production Research* 3 (3): 177–198.
Minner, S., and G. P. Kiesmller. 2011. “Dynamic Product Acquisition in Closed Loop Supply Chains.” *International Journal of Production Research* 50 (11): 2836–2851.
Nawaz, M., E. E. Enscore, and I. Ham. 1983. “A Heuristic Algorithm for the *m*-Machine, *n*-Job Flowshop Sequencing Problem.” *Omega* 11 (1): 91–95.
Neumytov, Y. D., and S. V. Sevastianov. 1993. “Approximation Algorithm with Tight Bound for Three-machine Counter-routes Problem (in Russian).” *Upravlyaemye Sistemy* 31: 53–65.
Oguz, C., B. M. T. Lin, and T. C. E. Cheng. 1997. “Two-stage Flowshop Scheduling Problem with a Common Second-stage Machine.” *Computers and Operations Research* 24 (12): 1169–1174.
Ovchinnikov, A. 2011. “Revenue and Cost Management for Remanufactured Products.” *Production and Operations Management* 20 (6): 824–840.
Pinedo, M. 2012. *Scheduling: Theory, Algorithms, and Systems*. New York: Springer.
Potts, C. N., S. V. Sevast’janov, V. A. Strusevich, L. N. Van Wassenhove, and C. M. Zwaneveld. 1995. “The Two-stage Assembly Scheduling Problem: Complexity and Approximation.” *Operations Research* 43 (2): 346–355.
Queyranne, M. 1993. “Structure of a Simple Scheduling Problem.” *Mathematical Programming* 58 (1–3): 263–285.
Rajendran, C., and H. Ziegler. 2004. “Ant-colony Algorithms for Permutation Flowshop Scheduling to Minimize Makespan/Total Flowtime of Jobs.” *European Journal of Operational Research* 155 (2): 426–438.
Roemer, T. A. 2006. “A Note on the Complexity of the Concurrent Open Shop Problem.” *Journal of Scheduling* 9 (4): 389–396.
Roy, S., and N. Geetha. 2006. “eWaste.” Express Channel Business, 2006. Accessed February 04, 2008. http://www.channelbusiness.in/index.php?option=com_content&task=view&id=302&Itemid=78
Schulz., A. S. 1996. “Scheduling to Minimize Total Weighted Completion Time: Performance Guarantees of LP-based Heuristics and Lower Bounds.” In *Integer Programming and Combinatorial Optimization, Lecture Notes in Artificial Intelligence 1084*, edited by W. H. Cunningham and M. Queyranne, Berlin, Heidelberg: Springer.
Schwarzer, S., A. De One, P. Peduzzi, G. Giuliani, and S. Kluser. 2005. “E-waste, the hidden side of IT equipment’s manufacturing and use. Envirenment Alert Bulletin, UNEP.” Accessed April 08, 2007. http://www.grid.unep.ch/product/publication/download/ew_ewaste.en.pdf
Stoner, F., and C. Wankel. 2008. *Global Sustainability Initiatives: New Models and New Approaches*. Charlotte, North Carolina: Information Age Publishing.
Shmoys, D. B., and E. Tardos. 1993. “An Approximation Algorithm for the Generalized Assignment Problem.” Mathematical Programming 62 (1–3): 461–474.
Stützle, T., 1998. *Applying Iterated Local Search to the Permutation Flow Shop Problem*. Technical Report AIDA-98-04, FG Intellektik, TU: Darmstadt.
Tang, O., and R. Teunter. 2006. “Economic Lot Scheduling Problem with Returns.” *Production and Operations Management* 15 (4): 488–496.
T’kindt, V., N. Monmarché, F. Tercinet, and D. Laügt. 2002. “An Ant Colony Optimization Algorithm to Solve a 2-Machine Bicriteria Flowshop Scheduling Problem.” *European Journal of Operational Research* 142 (2): 250–257.
UNEP. 2007. “E-Waste: Inventory Assessment Manual.” United Nations Environment Programme. http://www.unep.org/ietc/Portals/136/Publications/Waste%20Management/Ewaste%20Manual%203.pdf
UNEP. 2010. *Urgent Need to Prepare Developing Countries for Surge in E-Wastes*. Bali, Indonesia: United Nations Environment Programme press release.
US EPA. 2007a. “Management of Electronic Waste in the United States: Approach One. U.S. Environmental Protection Agency, eCycling, EPA 530-R-07-004a.” Accessed September 11, 2012. http://www.epa.gov/osw/conserve/materials/ecycling/docs/app-1.pdf

US EPA. 2007b. "Management of Electronic Waste in the United States: Approach two. U.S. Environmental Protection Agency, eCycling, EPA530-R-07-004b." Accessed September 11, 2012. http://www.epa.gov/osw/conserve/materials/ecycling/docs/app-2.pdf

US EPA. 2007c. "Management of Electronic Waste in the United States: Fact Sheet. U.S. Environmental Protection Agency, eCycling." Accessed September 11, 2012. http://www.epa.gov/osw/conserve/materials/ecycling/docs/fact7-08.pdf

van de Velde, S. L. 1990. "Minimizing the Sum of Job Completion Times in the Two-machine Flow-shop by Lagrangean Relaxation." *Annals of Operations Research* 26 (1–4): 257–268.

Vorasayan, J., and S. M. Ryan. 2006. "Optimal Price and Quantity of Refurbished Products." *Production and Operations Management* 15 (3): 369–383.

Wang, G., and T. C. E. Cheng. 2007. "Customer Order Scheduling to Minimize Total Weighted Completion Time." *Omega* 35 (5): 623–626.

Yang, J. 2010. "A New Complexity Proof for the Two-stage Hybrid Flow Shop Scheduling Problem with Dedicated Machines." *International Journal of Production Research* 48 (5): 1531–1538.

Yang, Y., S. Kreipl, and M. Pinedo. 2000. "Heuristics for Minimizing Total Weighted Tardiness in Flexible Flow Shops." *Journal of Scheduling* 3 (2): 89–108.

Downloaded by [National Chiao Tung University] at 06:53 24 April 2014