IRis

Northeastern University

Department of Mechanical and Industrial Engineering

January 01, 2005

Impact of different disassembly line balancing algorithms on the performance of dynamic kanban system for disassembly line

Surendra M. Gupta
Northeastern University

Elif A. Kizilkaya
Northeastern University

Recommended Citation

Gupta, Surendra M. and Kizilkaya, Elif A., "Impact of different disassembly line balancing algorithms on the performance of dynamic kanban system for disassembly line" (2005). . Paper 65. http://hdl.handle.net/2047/d10009899

This work is available open access, hosted by Northeastern University.

Laboratory for Responsible Manufacturing

Bibliographic Information

Kizilkaya, E. A. and Gupta, S. M., "Impact of Different Disassembly Line Balancing Algorithms on the Performance of Dynamic Kanban System for Disassembly Line", ***Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing V***, Boston, Massachusetts, pp. 103-110, October 23-24, 2005.

Copyright Information

Copyright 2005, Society of Photo-Optical Instrumentation Engineers.
This paper was published in Proceedings of SPIE (Volume 5997) and is made available as an electronic reprint with permission of SPIE. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Contact Information

Dr. Surendra M. Gupta, P.E.
Professor of Mechanical and Industrial Engineering and
Director of Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115, U.S.A.

(617)-373-4846 ***Phone***
(617)-373-2921 ***Fax***
gupta@neu.edu ***e-mail address***

http://www.coe.neu.edu/~smgupta/ ***Home Page***

Impact of Different Disassembly Line Balancing Algorithms on the Performance of Dynamic Kanban System for Disassembly Line

Elif A. Kizilkaya and **Surendra M. Gupta***
(*Corresponding Author)

Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115, USA.

Tel.: (617)-373-4846, Fax: (617)-373-2921, e-mail: gupta@neu.edu

ABSTRACT

In this paper, we compare the impact of different disassembly line balancing (DLB) algorithms on the performance of our recently introduced Dynamic Kanban System for Disassembly Line (DKSDL) to accommodate the vagaries of uncertainties associated with disassembly and remanufacturing processing. We consider a case study to illustrate the impact of various DLB algorithms on the DKSDL. The approach to the solution, scenario settings, results and the discussions of the results are included.

1.INTRODUCTION

The continuous growth in consumer waste seriously threatens the environment [3], [16]. Realizing this, many countries have passed laws that force manufacturers to take back their used products from consumers so that the components and materials may be retrieved from the products for reusing and recycling. Germany, for example, has passed a regulation that requires companies to remanufacture products until the product is obsolete. Japan has passed similar legislation requiring design and assembly methodologies that facilitate recycling of durable goods [2].

The first step in retrieving components and materials (for reusing, remanufacturing and recycling) from consumer products is disassembly [13]. Disassembly is the process of systematic removal of desirable constituents from the original assembly so that there is no impairment to any useful component. Our main focus is in the operational aspect of disassembly and, in particular, scheduling and shop floor control in a disassembly environment [1], [7], [8], [9], [11]. Although, the scheduling problems of assembly and disassembly share many characteristics (e.g., the dependent demand concept in discrete parts production systems), they also have their differences, and often the approaches to solve these two kinds of problems are very different [1]. Perhaps the most important difference is in the number of demand sources. In an assembly setting, the parts tend to converge to a single demand source (final product) as they move on the production floor. The governing principles are constrained by this "convergence" property. Under the disassembly setting, as the parts start moving away from their source of origin, they tend to diverge from each other leading to the "divergence" property. In addition to the "divergence property", each component item constitutes a source of demand, and fulfilling the demand of those component items cannot be done in an independent manner, since many of these component items share the same procurement source [8], [18], [19].

In this paper we consider the use of the JIT system for a disassembly line. In the disassembly line, more so than in a regular assembly line, optimum working conditions for a JIT system is hard to achieve because of the inherent uncertainties in a disassembly line. Thus the implementation of a JIT system in a disassembly line is fraught with numerous types of problems such as processing time variations due to the condition of the returned product, line imbalance, scrap rates, disposal rates and material handling system breakdowns.

The effect of processing times, line imbalance and material handling system breakdowns for a regular production system using TKS (Traditional Kanban System) and FKS (Flexible Kanban System) has been addressed by Gupta and Al-Turki [6]. In TKS the number of production and withdrawal Kanbans are kept constant even though it is a known fact that the

production control managers at times increase the number of Kanbans in the system on an ad hoc basis. The JIT system and the Kanban control mechanism are applicable to stable environments while the classical manufacturing firm has a very unstable environment. In order to deal with these uncertainties in the manufacturing environment Gupta and Al-Turki proposed a Flexible Kanban System where they systematically increased or decreased the number of production and withdrawal kanbans in the system [4], [5], [6], [9]. Following Gupta and Al-Turki's work other researchers such as Tardif and Maaseidvaag [20], Takahashi and Nakamura [17] also proposed and showed techniques where the manipulation of the number of Kanbans in unstable manufacturing environments led to superior performance.

In their work, Kizilkaya and Gupta [11] introduced a modified JIT system adapted to the disassembly environment and studied the behavior of the proposed MKSD (Modified Kanban System for Disassembly) model. They also showed that MKSD outperformed the regular push system such as the MRP system in the disassembly environment. In their recent work they also compared the MKSD model to DKSDL. They compared the performance of both models using simulation and showed DKSDL is far more flexible in handling operational uncertainties than the MKSD model. In this paper, we compare the impact of different disassembly line balancing (DLB) algorithms, such as Greedy and 2-Optimal [14], [15], on the performance of DKSDL to accommodate the vagaries of uncertainties associated with disassembly and remanufacturing processing. We consider a case study to illustrate the impact of various DLB algorithms on the DKSDL. The approach to the solution, scenario settings, results and the discussions of the results are included.

2. DYNAMIC KANBAN SYSTEM FOR DISASSEMBLY LINE (DKSDL)

In this section we briefly describe the Dynamic Kanban System for Disassembly Line (DKSDL).

In a classical push system each station is pushed to disassemble and fill the parts inventory buffers whether there is immediate demand for these parts or not. On the other hand, in DKSDL, the disassembled parts inventories are controlled and kept at a low level. They are limited by the number of the disassembly Kanbans (DKs), employed in the system at a given time. In DKSDL the number of disassembly Kanbans are varied based on the variable demand rate between a previously calculated base number of DKs and a maximum number of DKs. Additional products are disassembled in response to the demands of parts.

Figure 1 presents a Disassembly Station of a balanced DKSDL composed of N disassembly stations in series. Each Disassembly Station (DS) has one processing machine or operator, an input buffer and two types of output buffers. Depending on the end-of life product disassembled and applied DLB algorithm or heuristic there can be one or more types of parts disassembled at each station. The material flow is controlled via the use of disassembly withdrawal and disassembly production Kanbans. The production Kanban of a DS always circulates within that station, controlling disassembly at that station. The withdrawal Kanbans of a DS always circulate between input buffer of that station and the partially disassembled product buffer of its preceding station. Disassembly station in DKSDL has two types of production Kanbans, viz. a disassembly production Kanban for partially disassembled unit (DPKAN) due to demand occurring at the succeeding stations and a disassembly production Kanban for the disassembled parts (DPKAND) to satisfy the demand for individual parts at that station. Station N may have only DPKAND to satisfy demand for the disassembled parts if the product is completely disassembled at that station based on the DLB. The number of production Kanbans at each station limits the number of disassembled parts. There are two types of work-in-process (WIP) (types I and II) in the disassembly system. Type I WIP represents the partially disassembled product anywhere in the system and type II WIP represents the disassembled parts at any station's parts buffers. Parts buffers at station j are represented by $1j$, $2j$, $3j$, …., nj (See Figure 1).

In the DKSDL, the arrival of demand at any DS for any one of the part(s) triggers a pull action at the preceding DS. Therefore whether there is a need for the preceding DS's part(s) or not, one more product will be disassembled, possibly causing the disassembled part(s) buffer(s) (PB) for those other parts disassembled at that station and disassembled part(s) buffer(s) of the preceding DS to exceed their capacity. On the other hand, if there is no demand for the succeeding DS's parts, the partially disassembled products buffer (PDPB) for the current DS may exceed its capacity. In order to eliminate excess inventory at these buffers, DKSDL disposes additional products and/or part(s) (at some cost) when disassembly Kanban (DK) capacities are exceeded. Figure 1 shows in detail the inner workings of a Disassembly station j in DKSDL, in addition to depicting the newly introduced Kanban concepts for the disassembly line.

In order to model the disassembly environment more realistically we also had to take into account factors such as quality and quantity of parts in incoming products, stochastic disassembly task times, early leaving work pieces, self skipping work pieces, skipping work pieces, disappearing work pieces revisiting work pieces, exploding work pieces, variable demand of multiple parts at various disassembly work stations as well as the precedence relationships, hazardous parts, tasks requiring similar working conditions, tasks requiring special tooling and tasks requiring directional changes.

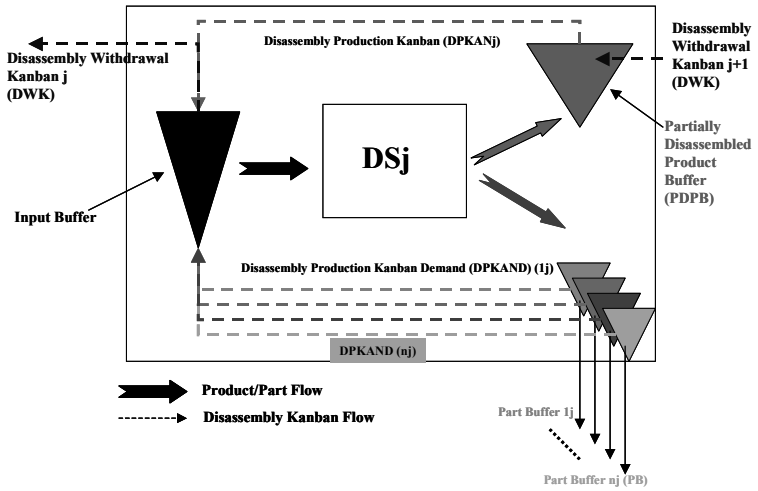

Figure 1. A Disassembly Station in DKSDL

Our approach to manipulating the number of production kanbans at each DS has been motivated by Gupta and Al-Turki's studies of an adaptive kanban control system for a production environment where the number of kanbans are varied with respect to the demand and the capacity of the system [4], [5], [6], [9]. The number of kanbans in their model never reaches below a base number of kanbans and the adjustments to the number of kanbans in the system are done once a period. The limitations of their model due to this inflexibility of not being able to modify the number of kanbans more than once in a period are, potential increase in WIP levels or backorders. Tardiff and Maaseidvaag's study of an adaptive kanban system allows the numbers of kanbans to be varied based on the current inventory levels and backorders in the system [20]. They assume to have extra kanbans available in the system, which they introduce to the system or take them back, based on pre-calculated capture and release inventory level thresholds. This way they are able to adapt to changing demand levels within a planning period. However their model is limited to a single-stage, single-product environment.

Given that the disassembly environment is a multi-stage and multi-product (part) system, the Dynamic Kanban System for Disassembly Line changes the number of kanbans at each DS with respect to the following developed logic:

A base and a ceiling number of kanbans are calculated for each type of production kanban, namely DPKAN and DPKAND, at every DS *i* and for every part *j*. The number of withdrawal kanbans is set to 1. They are the types of kanbans that trigger the disassembly process between the disassembly stations. As long as there is a production kanban available, the withdrawal kanban can be freed immediately. Because in a disassembly environment, each DS acts independently with its own processing, finished parts queues (PB) and individual demand source points, we need to calculate individual release and capture inventory level thresholds for the two types of production kanbans at each disassembly station. The overall numbers of DPKAN and DPKAND (for each of the parts) at each DS never go above their respective ceiling levels or below the base number of kanban levels.

In our model we not only change the number of production kanbans in the system but also the capacity of the PBs based on the same logic. If the number of DPKAND for a part is increased in a DS, the queue capacity is also increased by the number of parts disassembled at that station. In Tardiff and Maaseidvaag's research [20] the finished queue capacities are assumed to be infinite, which obviously is not be a desirable scenario as it would cause the queues to grow indefinitely for parts and partially disassembled products that are not demanded or have a low demand rate. This is why

the parts in our model are disposed of at a disposal cost, after they exceed certain limit. For the formulation of DKSDL, see our previous paper [12].

3. CASE STUDY

We chose a 10-part product for the case study that was also considered by McGovern and Gupta [14]. Table 1 shows the data for the product considered in the case study. Returned products arrive at the processing center according to Poisson distribution with a mean rate of 8. There are two material handling devices that carry partially disassembled products; one between disassembly stations 2 and 3 and the other between disassembly stations 3 and 4. The material handling devices are assumed to be operating without breakdowns.

Table 1. Data for the Case Study Product

Part	Precedence	Disassembly Time	Hazardous	Scrap Rate - Part	Demand
A		14	No	0.25	No
B	A, H, I, J	10	No	0.25	500
C	A, H, I, J	12	No	0.25	No
D		17	No	0.25	No
E		23	No	0.25	No
F		14	No	0.25	750
G	E, F	19	No	0.25	295
H	G, D	36	Yes	0.25	No
I		14	No	0.25	360
J		10	No	0.25	No

3.1 DLB Using Greedy Algorithm

In this scenario, the disassembly line for processing the product is balanced using the Greedy algorithm reported by McGovern and Gupta [15]. The Greedy algorithm uses a first fit decreasing method modified to take the precedence relationships and multiple objectives into account. Based on the calculations, we end up with five disassembly stations. Table 2 shows the parts disassembled at each station, their disassembly times and the estimated idle times. Idle times are based on a defined cycle time of 40 seconds. Each station has one operator who disassembles the product and retrieves the parts assigned to that station. Demand for each part is shown in Table 1.

Table 2. Balanced Disassembly Line Design and Task Assignments Using Greedy Algorithm

	DS-1	DS-2	DS-3	DS-4	DS-5
Parts	D, E	F, G	H	A, I, J	B, C
Disassembly Time (sec)	40	33	36	38	22
Idle Time	0	7	4	2	18

We calculated and obtained the necessary parameters for the DKSDL model using the formulation detailed in [12]. The buffer capacity is assumed to be 3 for each demanded part and 0 for non-demanded parts. Ceiling constant is assumed to be 2. The calculated parameter values (see [12]) are presented in Table 3.

Table 3. Greedy Algorithm Parameter Setting for the Case Study

Parameter	DS-1 D, E	DS-2 F, G	DS-3 H	DS-4 A, I, J	DS-5 B, C
$N_{DPKAND_{ij}}$	0, 0	17, 10	0	0,10,0	12,0
N_{DPKAN_j}	49	0	23	2	N/A
$Ceiling_{DPKAND_{ij}}$	0, 0	34,20	0	0,20,0	24, 0
$Ceiling_{DPKAN_j}$	98	0	46	4	N/A
PB_{ij}	0, 0	57, 30	0,	0,30,0	36,0
$C_{DPKAND_{ij}}$	1, 1	18,11	1	1,11,1	13,1
C_{DPKAN_j}	50	1	24	3	N/A
$R_{DPKAND_{ij}}$	0, 0	17,10	0,	0,10,0	12,0
R_{DPKAN_j}	49	0	23	2	N/A
X_i for +1	0.1, 0.05	0.1, 0.1	0.15	0.1,0.05,0.05	0.05, 0.05
Y_i for -1	0.1, 0.1	0.1, 0.1	0.1	0.1, 0.1	0.1, 0.1

3.2 DLB Using 2-Opt Algorithm

In this scenario, the disassembly line for processing the product is balanced using the 2-Opt algorithm reported by McGovern and Gupta [14]. Based on the calculations, we end up with five disassembly stations. Table 4 shows the parts disassembled at each station, their disassembly times and the estimated idle times. Idle times are based on a defined cycle time of 40 seconds. Each station has one operator who disassembles the product and retrieves the parts assigned to that station. Demand for each part is shown in Table 1.

Table 4. Disassembly Line Design and Task Assignments Using 2-Opt

	DS-1	DS-2	DS-3	DS-4	DS-5
Parts	E, J	F, G	D, I	H	A, B, C
Disassembly Time (sec)	33	33	31	36	36
Idle Time	7	7	9	4	4

We calculated and obtained the necessary parameters for the DKSDL model using the formulation detailed in [12]. The buffer capacity is assumed to be 3 for each demanded part and 0 for non-demanded parts. Ceiling constant is assumed to be 2. The calculated parameter values (see [12]) are presented in Table 5.

Table 5. 2-Opt Algorithm Parameter Setting for the Case Study

Parameter	DS-1 E, J	DS-2 F, G	DS-3 D, I	DS-4 H	DS-5 A, B, C
$N_{DPKAND_{ij}}$	0, 0	17, 10	0,10	0	0,12,0
N_{DPKAN_j}	49	0	2	12	N/A
$Ceiling_{DPKAND_{ij}}$	0, 0	34,20	0,20	0	0,24,0
$Ceiling_{DPKAN_j}$	98	0	4	24	N/A
PB_{ij}	0, 0	57,30	0,30	0	0,36,0
$C_{DPKAND_{ij}}$	1, 1	18,11	1,11	1	1,13,1
C_{DPKAN_j}	50	1	3	13	N/A
$R_{DPKAND_{ij}}$	0, 0	17,10	0,10	0	0,12,0
R_{DPKAN_j}	49	0	2	12	N/A
X_i for +1	0.05, 0.05	0.1, 0.1	0.1,0.05	0.15	0.1, 0.05, 0.05
Y_i for -1	0.1, 0.1	0.1, 0.1	0.1,0.1	0.1, 0.1	0.1, 0.1

3.3 DLB Using AEHC Algorithm

In this scenario, the disassembly line for processing the product is balanced using the AEHC algorithm reported by McGovern and Gupta [15]. Based on the calculations, we end up with five disassembly stations. Table 6 shows the parts disassembled at each station, their disassembly times and the estimated idle times. Idle times are based on a defined cycle time of 40 seconds. Each station has one operator who disassembles the product and retrieves the parts assigned to that station. Demand for each part is shown in Table 1.

Table 6. Disassembly Line Design and Task Assignments Using AEHC

	DS-1	DS-2	DS-3	DS-4	DS-5
Parts	E, F	D, G	H	A, I, J	B, C
Disassembly Time (sec)	37	36	33	38	22
Idle Time	3	4	7	2	18

We calculated and obtained the necessary parameters for the DKSDL model using the formulation detailed in [12]. The buffer capacity is assumed to be 3 for each demanded part and 0 for non-demanded parts. Ceiling constant is assumed to be 2. The calculated parameter values (see [12]) are presented in Table 7.

Table 7. AEHC Algorithm Parameter Setting for the Case Study

Parameter	DS-1 E, F	DS-2 D, G	DS-3 H	DS-4 A, I, J	DS-5 B, C
$N_{DPKAND_{ij}}$	0, 17	0,10	0	0,10,0	12,0
N_{DPKAN_j}	15	13	23	2	N/A
$Ceiling_{DPKAND_{ij}}$	0, 34	0,20	0	0,20,0	24,0
$Ceiling_{DPKAN_j}$	30	26	46	4	N/A
PB_{ij}	0, 57	0,30	0	0,30,0	36,0
$C_{DPKAND_{ij}}$	1, 18	1,11	1	1,11,1	13,1
C_{DPKAN_j}	16	14	24	3	N/A
$R_{DPKAND_{ij}}$	0, 17	0,10	0	0,10,0	12,0
R_{DPKAN_j}	15	13	23	2	N/A
$X_{i \text{ for } +1}$	0.05, 0.1	0.1, 0.1	0.15	0.1,0.05,0.05	0.05, 0.05
$Y_{i \text{ for } -1}$	0.1, 0.1	0.1, 0.1	0.1,0.1	0.1, 0.1	0.1, 0.1

4. EXPERIMENTATION AND RESULTS

We modeled the DKSDL using Arena 8.0 [10]. The models were run for 24000 minutes to warm up and data was collected for the next 480 minutes of simulation representing one day's production. We ran the model under variable demand rates. We used Poisson distribution with a mean of demand/hr calculated for each one of the items being disassembled and demanded. We ran three different scenarios to compare the effect of different DLB algorithms. Table 8 shows the results of the different scenarios. Overall we achieved the best WIP and demand satisfaction levels with the AEHC algorithm compared to the other two. One of the major reasons is that the disassembly station causing the biggest imbalance was the last disassembly station (DS-5) and therefore the overall processing time was reduced. Given that one of the highly demanded parts was disassembled in the last station actually helped the process. However, AEHC had higher disposal rates compared to the other two. That is, the higher demand satisfaction rate leads to higher disposal rate.

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper we have compared three different Disassembly Line Balancing Algorithms using our Dynamic Kanban System for Disassembly Line model. We have found the Greedy first then AEHC Algorithm to work better under the conditions of our case study defined. In future research different product types and configurations will be applied to the model to compare further into these DLB algorithms. Overall AEHC is a better fit for real time dynamic environments where immediate disassembly line balancing decisions can be made and the disassembly line can me configured quickly.

Table 8. Results

	DLB Using		
	Greedy	**2-Opt**	**AEHC**
WIP Type I	157.64	150.43	138.22
WIP Type II	19.272	96.348	31.036
Shortage-A	0	0	0
Shortage-B	423	475	232
Shortage-C	0	0	0
Shortage-D	0	0	0
Shortage-E	0	0	0
Shortage-F	673	712	274
Shortage-G	231	227	17
Shortage-H	0	0	0
Shortage-I	274	296	98
Shortage-J	0	0	0
Scrap-A	21	6	9
Scrap-B	11	5	10
Scrap-C	8	3	11
Scrap-D	14	22	43
Scrap-E	36	31	103
Scrap-F	26	26	122
Scrap-G	16	30	47
Scrap-H	9	7	9
Scrap-I	15	18	12
Scrap-J	11	29	9
Dispose-A	183	90	336
Dispose-B	136	0	5
Dispose-C	0	0	340
Dispose-D	137	102	412
Dispose-E	171	134	615
Dispose-F	0	0	0
Dispose-G	73	38	82
Dispose-H	140	89	334
Dispose-I	8	0	31
Dispose-J	143	136	341
Product Disposal Station 1	0	0	0
Product Disposal Station 2	0	0	309
Product Disposal Station 3	63	48	130
Product Disposal Station 4	1	38	1
Product Disposal Station 5			

REFERENCES

[1] Brennan, L., Gupta, S. M., and Taleb, K. N. "Operational Planning Issues in an Assembly/ Disassembly Environment", *International Journal of Operations and Production Management*, Vol.14, No.9, 57-67, 1994.

[2] Guide V.D.R, Srivastava, R. M., Spencer, S. "Are Production Systems Ready for The Green Revolution", *Production and Inventory Management Journal,* Vol.37, No.4, 70-76, 1996.

[3] Gungor, A. and Gupta, S. M. "Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey", *Computers and Industrial Engineering*, Vol. 36, No.4, 811-853, 1999.

[4] Gupta, S. M. and Al-Turki, Y. "Adapting Just-In-Time Manufacturing Systems to Preventive Maintenance Interruptions", *Production Planning and Control,* Vol. 9, N0.4, 349-359, 1998.

[5] Gupta, S. M. and Al-Turki, Y. "An Algorithm to Dynamically Adjust the Number of Kanbans in Stochastic Processing Times and Variable Demand Environment", *Production Planning and Control,* Vol. 8, No.2, 133-141, 1997.

[6] Gupta, S. M. and Al-Turki, Y. "The Effect of Sudden Material Handling System Breakdown on the Performance of a JIT System", *International Journal of Production Research,* Vol. 36, No.7, 1935-1960, 1998.

[7] Gupta, S. M. and McLean, C. R. "Disassembly of Product", *Computers and Industrial Engineering*, Vol.31, No.1-2, 225-228, 1996.

[8] Gupta, S. M. and Taleb, K. N. "Scheduling Disassembly", *International Journal of production Research*, Vol.32, No.8, 1857-1866, 1994.

[9] Gupta, S. M., Al-Turki, Y. and Perry, R. F. "Flexible Kanban System", *International Journal of Operations and Production Management,* Vol. 19, No. 10, 1065-1093, 1999.

[10] Kelton, D., Sadowski, R. P. and Sadowski, D. A., "Simulation with ARENA", McGraw-Hill, New York, NY, 1998.

[11] Kizilkaya, E. A. and Gupta, S. M., "A Pull Control System for the Disassembly Environment", *Proceedings of the 2000 World Symposium on Group Technology/ Cellular Manufacturing*, San Juan, Puerto Rico, March 27-29, pp. 233-238, 2000.

[12] Kizilkaya, E. A. and Gupta, S. M., "Modeling Operational Behavior of a Disassembly Line", *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing IV*, Philadelphia, Pennsylvania, pp. 79-93, October 26-27, 2004.

[13] Lambert, A. J. D. and Gupta, S. M., "Disassembly Modeling for Assembly, Maintenance, Reuse, and Recycling", Boca Raton, Florida, CRC Press, 2005.

[14] McGovern, S. M. and Gupta, S. M. "2-Opt Heuristic for the Disassembly Line Balancing Problem", *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing III*, Providence, 71-84, October 29-30, 2003.

[15] McGovern, S. M. and Gupta, S. M., "Local Search Heuristics and Greedy Algorithm for Balancing a Disassembly Line", *International Journal of Operations and Quantitative Management*, Vol.11, No.2, 2005.

[16] Moyer, L. and Gupta, S. M. "Environmental Concerns and Recycling/Disassembly Efforts in the Electronics Industry", *Journal of Electronics Manufacturing*, Vol.7, No.1, 1-22, 1997.

[17] Takahashi, K. and Nakamura, N. "Agile Control in JIT Ordering Systems", *International Journal of Agile Management Systems,* Vol. 2, No. 3, 242-252, 2000.

[18] Taleb, K. and Gupta, S. M. "Disassembly of Multiple Product Structures", *Computers and Industrial Engineering*, Vol.32, No.4, 949-961, 1997.

[19] Taleb, K., Gupta, S. M. and Brennan, L. "Disassembly of Complex Products with Parts and Materials Commonality", *Production Planning and Control*, Vol.8, No.3, 255-269, 1997.

[20] Tardiff, V. and Maaseidvaag, L. "An Adaptive Approach to Controlling Kanban Systems", *European Journal of Operations Research,* Vol.132, 411-424, 2001.