Ant colony optimization for sequence-dependent disassembly line balancing problem

Can B. Kalayci
Department of Industrial Engineering, Pamukkale University, Denizli, Turkey, and
Surendra M. Gupta
Department of Mechanical and Industrial Engineering, Northeastern University, Boston, Massachusetts, USA

Received February 2012
Revised March 2012
Accepted April 2012

Abstract
Purpose – The purpose of this paper is to introduce sequence-dependent disassembly line balancing problem (SDDLBP) to the literature and propose an efficient metaheuristic solution methodology to this NP-complete problem.

Design/methodology/approach – This manuscript utilizes a well-proven metaheuristics solution methodology, namely, ant colony optimization, to address the problem.

Findings – Since SDDLBP is NP-complete, finding an optimal balance becomes computationally prohibitive due to exponential growth of the solution space with the increase in the number of parts. The proposed methodology is very fast, generates (near) optimal solutions, preserves precedence requirements and is easy to implement.

Practical implications – Since development of cost effective and profitable disassembly systems is an important issue in end-of-life product treatment, every step towards improving disassembly line balancing brings us closer to cost savings and compelling practicality.

Originality/value – This paper introduces a new problem (SDDLBP) and an efficient solution to the literature.

Keywords Product recovery, Disassembly, Sequence-dependent disassembly line balancing, Combinatorial optimization, Metaheuristics, Ant colony optimization, Optimization techniques, Production planning and control

Paper type Research paper

1. Introduction
Product recovery seeks to obtain materials and parts from old or outdated products through recycling and remanufacturing in order to minimize the amount of waste sent to landfills. Gungor and Gupta (1999) and Ilgin and Gupta (2010) provide an extensive review of product recovery. The first crucial and the most time consuming step of product recovery is disassembly. Disassembly is defined as the systematic extraction of valuable parts and materials from discarded products through a series of operations to use in remanufacturing or recycling after appropriate cleaning and testing operations. Disassembly operations can be performed at a single workstation, in a disassembly cell or on a disassembly line. Although a single workstation and

This paper is part of the special issue on Advances in Production Line Systems guest edited by Dr Sabry Shaaban and Dr Sarah Hudson.

Journal of Manufacturing Technology Management
Vol. 24 No. 3, 2013
pp. 413-427
© Emerald Group Publishing Limited
1741-038X
DOI 10.1108/17410381311318909

disassembly cell are more flexible, the highest productivity rate is provided by a disassembly line and hence is the best choice for automated disassembly processes, a feature that will be essential in the future disassembly systems (Gungor and Gupta, 2001; Güngör and Gupta, 2002). A line can be paced (Caridi *et al.*, 2006) or unpaced (McNamara *et al.*, 2011; Shaaban and Hudson, 2011). In this paper we seek to balance a paced disassembly line (Güngör and Gupta, 2002) considering sequence dependency.

Disassembly operations have unique characteristics and cannot be considered as the reverse of assembly operations. The quality and quantity of components used in the stations of an assembly line can be controlled by imposing strict conditions. However, there are no such conditions of EOL products moving on a disassembly line. In a disassembly environment, the flow process is divergent; a single product is broken down into many subassemblies and parts while the flow process is convergent in an assembly environment. There is also a high degree of uncertainty in the structure, quality, reliability and the condition of the returned products in disassembly. Additionally, some parts of the product may be hazardous and may require special handling that will affect the utilization of disassembly workstations. Since disassembly tends to be expensive, disassembly line balancing becomes significant in minimizing resources invested in disassembly and maximizing the level of automation.

Disassembly line balancing problem (DLBP) is a multi-objective problem that is described in Güngör and Gupta (2002) and has mathematically been proven to be NP-complete in McGovern and Gupta (2007) making the goal to achieve the optimal balance computationally expensive. Exhaustive search works well enough in obtaining optimal solutions for small sized instances; however its exponential time complexity limits its application on the large sized instances. An efficient search method needs to be employed to attain a (near) optimal condition with respect to objective functions. Although some researchers have formulated the DLBP using mathematical programming techniques (Altekin and Akkan, 2011; Altekin *et al.*, 2008; Koc *et al.*, 2009), it quickly becomes unsolvable for a practical sized problem due to its combinatorial nature. For this reason, there is an increasing need to use metaheuristic techniques such as genetic algorithms (GA) (Kalayci and Gupta, 2011a; McGovern and Gupta, 2007), ant colony optimization (ACO) (Agrawal and Tiwari, 2008; Ding *et al.*, 2010; McGovern and Gupta, 2005; Tripathi *et al.*, 2009), simulated annealing (SA) (Kalayci *et al.*, 2011a), tabu search (TS) (Kalayci and Gupta, 2011b) and artificial bee colony (ABC) (Kalayci *et al.*, 2011b). McGovern and Gupta (2011) for more information on DLBP. In this paper, we define, mathematically formalize sequence-dependent disassembly line balancing problem (SDDLBP) and solve it by ACO metaheuristic.

The rest of the paper is organized as follows: in Section 2, notation used in this paper is presented. Problem definition and formulation is given in Section 3. Section 4 describes the proposed ACO algorithm for the multi-objective SDDLBP. The computational experience to evaluate its performance on a numerical example is provided in Section 5. Finally some conclusions are pointed out in Section 6.

2. Notation

α	relative importance of the pheromone trail in path selection
β	relative importance of heuristic information in path selection
η	heuristic information (visibility) of task j (i.e. the priority rule value for task j)

ρ	evaporation coefficient
τ_0	initial pheromone level
τ_{ij}	pheromone trail intensity in the path "selecting task *j* after selecting task *i*"
a	ant count $(1, \ldots, an)$
an	number of ants
AS_i^a	the set of assignable tasks for ant *a* after the selection of task *i*
AV	available task list
c	cycle time (maximum time available at each workstation)
d_i	demand; quantity of part *i* requested
h_i	binary value; 1 if part *i* is hazardous, else 0
IP	set (i,j) of parts such that task *i* must precede task *j*
i	part identification, task count $(1, \ldots, n)$
j	part identification, task count $(1, \ldots, n)$
j_1	part identification, task count $(1, \ldots, n)$
j_2	part identification, task count $(1, \ldots, n)$
j_3	part identification, task count $(1, \ldots, n)$
k	workstation count $(1, \ldots, m)$
LB	line balance solution
m	number of workstations required for a given solution sequence
m^*	minimum possible number of workstations
M	sufficiently large number
n	number of parts for removal
N	the set of natural numbers
P_{ij}	probability of a task being selected, from the set of available tasks (AV)
PS_i	*i*th part in a solution sequence (i.e. for solution $\langle 1, 2, 3, 6, 5, 8, 7, 4\rangle, PS_4 = 6$)
q_0	selection probability parameter for the use of heuristic information
q_1	selection probability parameter for the use of pheromone trail information
r	uniformly distributed random number between 0 and 1
sd_{ij}	sequence-dependent time increment influence of *i* on *j*
$\lvert SUC_j \rvert$	the number of all successors of task *j*
t_i	part removal time of part *i*

t'_i	part removal time of part i considering sequence-dependent time increment
Q	constant parameter (amount of pheromone added if a path is selected)

3. Problem definition and formulation

The SDDLBP investigated in this paper is concerned with the paced disassembly line for a single model of product that undergoes complete disassembly. Model assumptions include the following: a single product type is to be disassembled on a disassembly line; the supply of the end-of-life product is infinite; the exact quantity of each part available in the product is known and constant; a disassembly task cannot be divided between two workstations; each part has an assumed associated resale value which includes its market value and recycled material value; the line is paced; part removal times are deterministic, constant, and discrete; each product undergoes complete disassembly even if the demand is zero; all products contain all parts with no additions, deletions, modifications or physical defects; each part is assigned to one and only one workstation; the sum of part removal times of all the parts assigned to a workstation must not exceed cycle time; the precedence relationships among the parts must not be violated.

The difference between DLBP and SDDLBP is task time interactions. As opposed to the DLBP, in SDDLBP whenever a task interacts with another task, their task times may be influenced. For example, consider the disassembly of a personal computer, where several components have to be disassembled at the same workstation or neighboring ones. Disassembling a particular component before another component from the same motherboard may prolong (or curtail) the task time, as opposed to disassembling them in reverse order, because one component could hinder the other since it requires additional movements and/or prevents it from using the most efficient disassembly process.

In order to model this very typical situation adequately, the concept of sequence-dependent task time increments is introduced. If task j is performed before task i, its standard time t_j is incremented by sd_{ij}. This sequence-dependent increment measures the prolongation of task j forced by the interference of already waiting task i. Obviously, tasks i and j only can interact in the described manner if they do not have any precedence relationships, i.e. there is no path in the precedence graph directly connecting i and j.

Illustrative example: the precedence relationships (solid line arrows) and sequence-dependent time increments (dashed line arrows) for an eight part PC disassembly process are shown in Figure 1 and their knowledge database is given in Table I. This example is modified from Güngör and Gupta (2002).

Sequence dependencies for the PC example are given as follows: $sd_{23} = 2,\ sd_{32} = 4,\ sd_{56} = 1,\ sd_{65} = 3.$

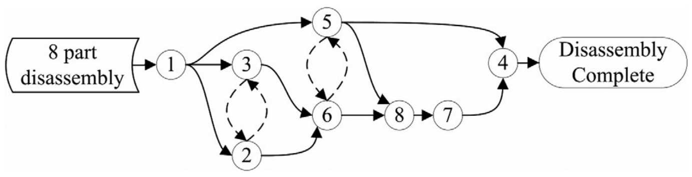

Figure 1. Precedence relationships (solid line arrows) and sequence-dependent time increments (dashed line arrows) for the PC example

For a feasible sequence $\langle 1, 2, 3, 6, 5, 8, 7, 4 \rangle$; since part 2 is disassembled before part 3, sequence dependency $sd_{32} = 4$ takes place because when part 2 is disassembled, the obstacle part 3 is still not taken out, i.e. the part removal time for part 2 is increased which results in $t_2^{'} = t_2 + sd_{32} = 14$; similarly since part 6 is disassembled before part 5, sequence dependency $sd_{56} = 1$ takes place because when part 6 is disassembled, the obstacle part 5 is still not taken out, i.e. the part removal time for part 6 is increased which results in $t_6^{'} = t_6 + sd_{56} = 17$.

For another feasible sequence $\langle 1, 3, 2, 5, 6, 8, 7, 4 \rangle$ with the same PC example; since part 3 is disassembled before part 2, sequence dependency $sd_{23} = 2$ takes place because when part 3 is disassembled, the obstacle part 2 is still not taken out, i.e. the part removal time for part 3 is increased which results in $t_3^{'} = t_3 + sd_{23} = 14$; since part 5 is disassembled before part 6, sequence dependency $sd_{65} = 3$ takes place because when part 5 is disassembled, the obstacle part 6 is still not taken out, i.e. the part removal time for part 5 is increased which results in $t_5^{'} = t_5 + sd_{65} = 26$.

The mathematical formulation of our DLBP is given as follows.

In this paper, the precedence relationships considered are of AND type and are represented using the immediately preceding matrix $[y_{ij}]_{n \times n}$, where:

$$y_{ij} = \begin{cases} 1 & \text{if task } i \text{ is executed after task } j \\ 0 & \text{if task } i \text{ is executed before task } j \end{cases} \tag{1}$$

In order to state the partition of total tasks, we use the assignment matrix $[x_{jk}]_{n \times m}$, where:

$$x_{jk} = \begin{cases} 1 & \text{if part } j \text{ is assigned to station } k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The mathematical formulation of SDDLBP is given as follows:

$$\min f_1 = m \tag{3}$$

$$\min f_2 = \sum_{i=1}^{m} \left(c - t_i^{'} \right)^2 \tag{4}$$

$$\min f_3 = \sum_{i=1}^{n} i \times h_{PS_i}, \quad h_{PS_i} = \begin{cases} 1 & \text{hazardous} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Part	Task	Time	Hazardous	Demand
PC top cover	1	14	No	360
Floppy drive	2	10	No	500
Hard drive	3	12	No	620
Back plane	4	18	No	480
PCI cards	5	23	No	540
RAM modules	6	16	No	750
Power supply	7	20	No	295
Motherboard	8	36	No	720

Table I. Knowledge database for the PC example

$$\min f_4 = \sum_{i=1}^{n} i \times d_{PS_i}, \quad d_{PS_i} \in N, \ \forall PS_i \tag{6}$$

Subject to:

$$\sum_{k=1}^{m} x_{jk} = 1, \quad j = 1, \ldots, n \tag{7}$$

$$\left\lfloor \frac{\sum_{i=1}^{n} t_i}{c} \right\rfloor \leq m^* \leq n \tag{8}$$

$$\sum_{j=1}^{n} \left(t_j + \sum_{i=1}^{n} sd_{ij} \times y_{ij} \right) \times x_{jk} \leq c \tag{9}$$

$$x_{ik} \leq \sum_{k=1}^{m} x_{jk}, \quad \forall (i,j) \in IP \tag{10}$$

The first objective given in equation (3) is to minimize the number of workstations for a given cycle time (the maximum time available at each workstation) (Baybars, 1986). It rewards the minimum number of workstations, but allows the unlimited variance in the idle times between workstations because no comparison is made between station times. It also does not force to minimize the total idle time of workstations.

The second objective given in equation (4) is to aggressively ensure that idle times at each workstation are similar, though at the expense of the generation of a non-linear objective function (McGovern and Gupta, 2007). The method is computed based on the minimum number of workstations required as well as the sum of the square of the idle times for all the workstations. This penalizes solutions where, even though the number of workstations may be minimized, one or more have an exorbitant amount of idle time when compared to the other workstations. It also provides for leveling the workload between different workstations on the disassembly line. Therefore, a resulting minimum performance value is the more desirable solution indicating both a minimum number of workstations and similar idle times across all workstations.

As the third objective (equation (5)), a hazard measure developed to quantify each solution sequence's performance, with a lower calculated value being more desirable (McGovern and Gupta, 2007). This measure is based on binary variables that indicate whether a part is considered to contain hazardous material (the binary variable is equal to 1 if the part is hazardous, else 0) and its position in the sequence. A given solution sequence hazard measure is defined as the sum of hazard binary flags multiplied by their position number in the solution sequence, thereby rewarding the removal of hazardous parts early in the part removal sequence.

As the fourth objective (equation (6)), a demand measure was developed to quantify each solution sequence's performance, with a lower calculated value being more desirable (McGovern and Gupta, 2007). This measure is based on positive integer values that indicate the quantity required of a given part after it is removed (or 0 if it is not desired) and its position in the sequence. A solution sequence demand measure is then defined as the sum of the demand value multiplied by the position of the part in the sequence, thereby rewarding the removal of high demand parts early in the part removal sequence.

The constraints given in: equation (7) ensure that all tasks are assigned to at least and at most one workstation (the complete assignment of each task), equation (8) guarantees that the number of work stations with a workload does not exceed the permitted number, equation (9) ensures that the work content of a workstation cannot exceed the cycle time and equation (10) imposes the restriction that all the disassembly precedence relationships between tasks should be satisfied.

4. Proposed ACO approach

The decision version of DLBP was proven to be NP-complete (and hence, the optimization version is NP-hard) (McGovern and Gupta, 2007). Since SDDLBP is a generalization of DLBP (setting all sequence-dependent time increments to zero, SDDLBP reduces to DLBP), the decision version of SDDLBP is NP-complete and its optimization versions are also NP-hard.

Since SDDLBP falls into the NP-Complete class of combinatorial optimization problems, when the problem size increases, the solution space is exponentially increased and an optimal solution in polynomial time cannot be found as it can be time consuming for optimum seeking methods to obtain an optimal solution within this vast search space. Therefore, it is necessary to use alternative methods in order to reach (near) optimal solutions faster. For this reason, a fast and effective ACO approach is used to solve the problem.

4.1 Solution representation

One of the most important decisions in designing a metaheuristic lies in deciding how to represent solutions and relate them in an efficient way to the searching space. Also, solution representation should be easy to decode to reduce the cost of the algorithm. In the proposed ACO algorithm, permutation based representation is used, so elements of a solution string are integers. Each element represents a task assignment to work station. The value of the first element of the array shows which task is assigned to workstations first, the second value shows which task is assigned second and so on. For example, if there are eight tasks to be assigned to workstations then the length of the solution string is eight. Figure 2 shows assignment of tasks to workstations as an example.

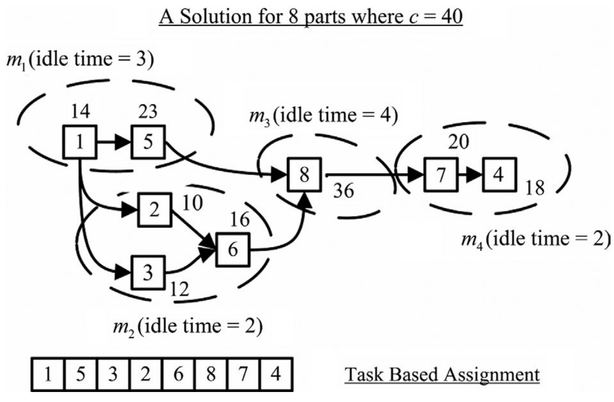

Figure 2. Assignment of tasks to workstations

4.2 Feasible solution construction strategy

The strategy of building a feasible balancing solution is the key issue to solve the line balancing problems. We use station-oriented procedure for a solution constructing strategy in which solutions are generated by filling workstations successively one after the other (Ding *et al.*, 2010). The procedure is initiated by the opening of a first station. Then, tasks are successively assigned to this station until more tasks cannot be assigned and a new station is opened. In each iteration, a task is selected according to the probabilistic selection criteria from the set of candidate tasks to assign to the current station. When no more tasks may be assigned to the open station, this is closed and the following station is opened. The procedure finalizes when there are no more tasks left to assign.

In order to describe the process to build a feasible balancing solution, AV and assignable task are defined as follows.

A task is an AV if and only if it has not already been assigned to a workstation and all of its predecessors have already been assigned to a workstation.

A task is an assignable task if and only if it belongs to the set of AV and if the idle time of current workstation is higher than or equal to the processing time of the task.

The generation procedure of a feasible balancing solution is given as follows:

- *Step 0.* Initialization; open the first workstation for task assignment.
- *Step 1.* According to the precedence constraints construct the AV set.
- *Step 2.* According to the cycle time construct the assignable task set.
- *Step 3.* If the set of candidate task is null, go to step 5.
- *Step 4.* Select a task from the assignable task set according to the probabilistic selection criterion and assign the task to the current workstation; go back to step 1.
- *Step 5.* If the set of AV is null, go to step 7.
- *Step 6.* Open a new workstation, go back to step 1.
- *Step 7.* Stop the procedure.

4.3 Task selection strategy

First, at the initialization step, the pheromone trails, the heuristic information and the parameters are initialized. Second, in the iterative step, each ant repeatedly applies the state transition rule to select a task to assign to a station such that the precedence relations of each model are satisfied until a complete assignment is constructed. When building an assignment, both the heuristic information and the pheromone amount are used to determine the tasks to be chosen.

The role of visibility values (heuristic information) in ant algorithm is important since it provides problem-specific knowledge to manage the ants' probabilistic solution process. The visibility η_j of task j (the priority rule) which is analogous to providing the ants with sight, can be calculated as follows (Ding *et al.*, 2010):

$$\eta_j = \frac{t_j}{c} + \frac{|SUC_j|}{Max_{1 \leq i \leq n}\{|SUC_i|\}} \tag{11}$$

where $|SUC_j|$ is the number of all successors of task j.

From the set of AVs, an ant selects one task for assignment to the current workstation, according to a selection rule that takes into account the heuristic information about each AV and the pheromone trail intensity between the previously selected task and each AV (Vilarinho and Simaria, 2006). An ant which has selected task i in the previous iteration will select task j by applying the following rule:

$$j = \begin{cases} j_1 : \arg\max_{j \in AS_i^a} \{(\tau_{ij})^{\alpha}(\eta_j)^{\beta}\} & \text{if } 0 \le r \le q_0 \quad \text{(exploitation)} \\ j_2 : P_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_j)^{\beta}}{\sum_{j_2 \in AS_i^a}(\tau_{ij_2})^{\alpha}(\eta_{j_2})^{\beta}} & \text{else if } q_0 < r < q_1 \quad \text{(biased exploration)} \\ j_3 : \text{Random select } j \in AS_i^a & \text{else if } q_1 < r \le 1 \quad \text{(random selection)} \end{cases} \tag{12}$$

4.4 Pheromone release strategy

The pheromone release strategy is based on the one used by Dorigo *et al.* (1996). While making the assignment, to vary other ants' assignments and avoiding leading to local optima, an ant also decreases the amount of pheromone by applying the following local updating rule:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\tau_0 \tag{13}$$

where τ_0 is the initial pheromone level and ρ is the local pheromone evaporating parameter $(0 < \rho < 1)$.

Then, for the current iteration, the global updating rule is applied to increase the pheromone for the task of the best assignment and to decrease the pheromone for tasks that are assigned to other stations. Thus, all the ants will focus on a better assignment. The pheromone trail level is updated by the following global updating rule:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij} \tag{14}$$

where:

$$\Delta\tau_{ij} = \begin{cases} Q/f_2 & \text{if } (i,j) \in \text{best schedule} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Although this is a multi-criteria problem, only a single criterion (the measure of balance, f_2) is being used in the basic ACO calculations and trail selection. The other objectives are only considered after balance and then only at the completion of each cycle, not as part of the probabilistic node selection. That is, an ant's tour solution is produced based on f_2, while at the end of each cycle the best overall solution is then updated based on f_1, f_2, f_3, f_4 in that order. This is done because for the purpose of this study, the balance is considered be the overriding requirement, as well as to be consistent with previous multi-criteria DLBP studies by the authors (McGovern and Gupta, 2006).

5. Numerical results

The proposed algorithm was coded in MATLAB and tested on Intel Core2 1.79 GHz processor with 3GB RAM. After engineering, the program is investigated on two different scenarios for verification and validation purposes.

The first scenario is for a given product consisting of $n = 10$ components. The knowledge database and precedence relationships for the components are given in Table II and Figure 3, respectively. The problem and its data were modified from McGovern and Gupta (2006) with a disassembly line operating at a speed which allows $c = 40$ s for each workstation to perform its required disassembly tasks.

The sequence dependencies for the ten part product are given as the following:

$$sd_{14} = 1,\ sd_{23} = 2,\ sd_{32} = 3,\ sd_{41} = 4,\ sd_{45} = 4,\ sd_{54} = 2,\ sd_{56} = 2,\ sd_{65} = 4,\ sd_{69} = 3,\ sd_{96} = 1$$

Here the objective is to create feasible solutions for the complete disassembly of the product.

Table II.
Knowledge database for the ten part product

Task	Time	Hazardous	Demand
1	14	No	0
2	10	No	500
3	12	No	0
4	17	No	0
5	23	No	0
6	14	No	750
7	19	Yes	295
8	36	No	0
9	14	No	360
10	10	No	0

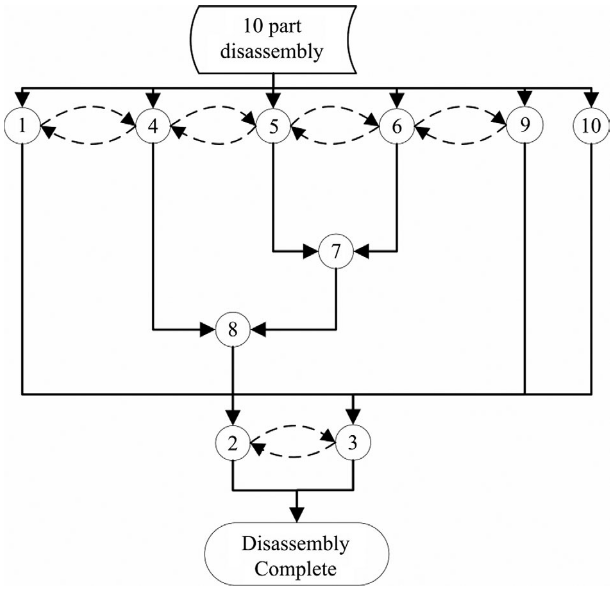

Figure 3.
Precedence relationships (solid line arrows) and sequence-dependent time increments (dashed line arrows) for the ten part product

While the exhaustive search method was able to find optimal solution in 215t time on average, the proposed approach was able to successfully find the optimal solution in just over 5t time on average under the restriction of the system specifications given above. The proposed approach rapidly found optimal solutions in an exponentially large search space (as large as 10! = 3,628,800). Table III depicts an optimal solution sequence. The fitness function values of the optimal solution are found to be: $f_1 = 5,\ f_2 = 67,\ f_3 = 5,\ f_4 = 9,605$. According to this sequence, sequence-dependent time increments $sd_{56},\ sd_{96},\ sd_{41},\ sd_{45},\ sd_{32}$ are added to the part removal times of part 6, 6, 1, 5, 2, respectively.

The second scenario is for a cellular telephone instance consisting of $n = 25$ components. The knowledge database and precedence relationships for the components are given in Table IV and Figure 4, respectively. The problem and its data were modified from Gupta *et al.* (2004) with a disassembly line operating at a speed which allows $c = 18$ for each workstation to perform its required disassembly tasks.

The sequence dependencies for the 25 part product are given as the following:

$$sd_{45} = 2,\ sd_{54} = 1,\ sd_{67} = 1,\ sd_{69} = 2,\ sd_{76} = 2,\ sd_{78} = 1,\ sd_{87} = 2,\ sd_{96} = 1,$$
$$sd_{13,14} = 1,\ sd_{14,13} = 2\quad sd_{14,15} = 2,\ sd_{15,14} = 1,\ sd_{20,21} = 1,\ sd_{21,20} = 2,$$
$$sd_{22,25} = 1,\ sd_{25,22} = 2$$

Since within the vast search space (25!), the exhaustive search is useless due to the exponential growth of the time complexity, i.e. the optimal solution is unknown. The proposed ACO approach was able to find the solution given in Figure 5 as the best performance. It took less than 100t time on average under the restriction of the system specifications given above.

6. Conclusions

The main objective of this paper was to define and formulate SDDLBP which aimed to minimize the number of disassembly workstations, minimize the total idle time of

		Workstations I	II	III	IV	V	
Part removal sequence ↓	6	14(+2 + 1)					Time to remove parts (in seconds)
	1	14(+4)					
	10		10				
	5		23(+4)				
	7			19			
	4			17			
	8				36		
	9					14	
	2					10(+3)	
	3					12	
Total time		35	37	36	36	39	
Idle time		5	3	4	4	1	

Table III. An optimal line balancing solution (*LB*) for ten part product disassembly

Part	Task	Part removal time	Hazardous	Demand
Antenna	1	3	Yes	4
Battery	2	2	Yes	7
Antenna guide	3	3	No	1
Bolt (Type 1) A	4	10	No	1
Bolt (Type1) B	5	10	No	1
Bolt (Type2) 1	6	15	No	1
Bolt (Type2) 2	7	15	No	1
Bolt (Type2) 3	8	15	No	1
Bolt (Type2) 4	9	15	No	1
Clip	10	2	No	2
Rubber seal	11	2	No	1
Speaker	12	2	Yes	4
White cable	13	2	No	1
Red/blue cable	14	2	No	1
Orange cable	15	2	No	1
Metal top	16	2	No	1
Front cover	17	2	No	2
Back cover	18	3	No	2
Circuit board	19	18	Yes	8
Plastic screen	20	5	No	1
Keyboard	21	1	No	4
LCD	22	5	No	6
Sub-keyboard	23	15	Yes	7
Internal IC board	24	2	No	1
Microphone	25	2	Yes	4

Table IV.
Knowledge base of cellular telephone instance

all workstations by ensuring similar idle time at each workstation considering sequence-dependent time increments, maximize the removal of hazardous components as early as possible in the disassembly sequence and maximize the removal of high demand components before low demand components. A fast, near-optimal, ACO approach to the multi-objective deterministic SDDLBP was developed and presented in this paper. The ACO rapidly provides a feasible solution to the SDDLBP using an ant system/ant-cycle model algorithm based on the ACO metaheuristic modified to meet multiple objectives. The DLBP ACO method appears well suited to the multi-criteria decision-making problem format as well as for the solution of problems with non-linear objectives. The DLBP ACO provides a near-optimal minimum number of workstations, with the level of optimality increasing with the number of constraints. In addition, the DLBP ACO method is suitable for integer problems, a requirement of many disassembly problems, which generally do not lend themselves to rapid or easy solution by traditional optimum solution generating mathematical programming techniques. Since SDDLBP is new to the literature; as a future research, different combinatorial optimization techniques can be applied to solve the problem. Furthermore, the comparison of these proposed methods can be very useful to demonstrate the solution performance in terms of time complexity and fitness measures.

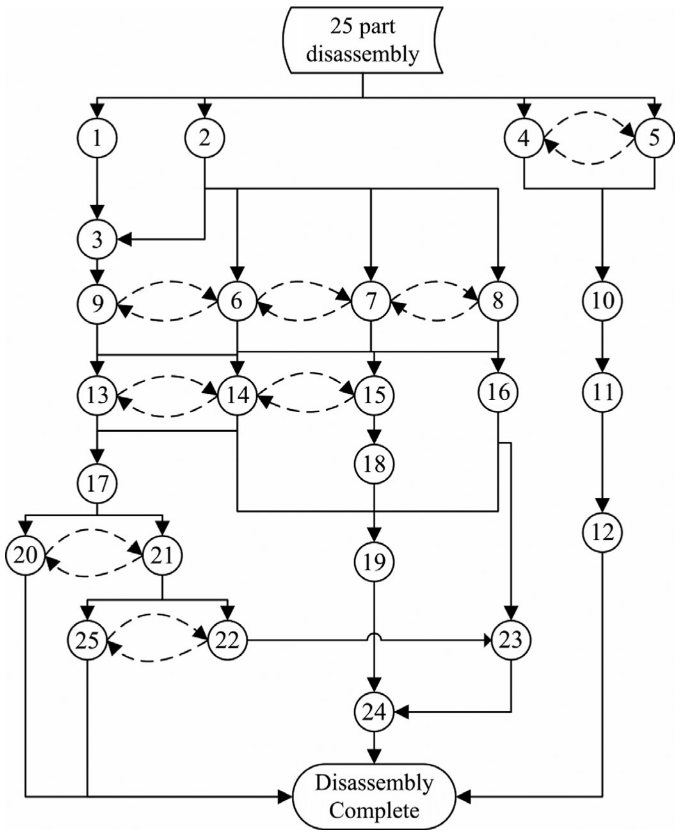

Figure 4. Cellular telephone precedence relationships

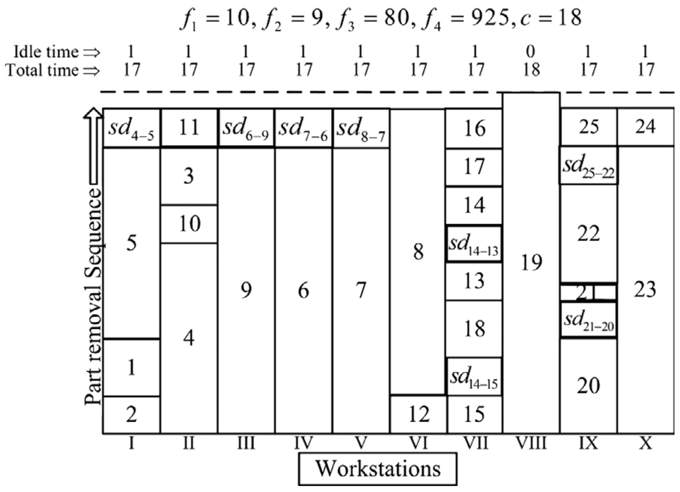
$f_1 = 10, f_2 = 9, f_3 = 80, f_4 = 925, c = 18$
Idle time ⇒ 1 1 1 1 1 1 1 0 1 1
Total time ⇒ 17 17 17 17 17 17 17 18 17 17
Part removal Sequence
I: 2, 1, 5, sd_{4-5}
II: 4, 10, 3, 11
III: 9, sd_{6-9}
IV: 6, sd_{7-6}
V: 7, sd_{8-7}
VI: 12, 8
VII: 15, sd_{14-15}, 18, 13, sd_{14-13}, 14, 17, 16
VIII: 19
IX: 20, sd_{21-20}, 21, 22, sd_{25-22}, 25
X: 23, 24
I II III IV V VI VII VIII IX X
Workstations

Figure 5. The best SDDLBP solution found by ACO using the cellular telephone instance

References

Agrawal, S. and Tiwari, M.K. (2008), "A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem", *International Journal of Production Research*, Vol. 46, pp. 1405-29.

Altekin, F.T. and Akkan, C. (2011), "Task-failure-driven rebalancing of disassembly lines", *International Journal of Production Research*, Vol. 46 No. 10, pp. 1-22.

Altekin, F.T., Kandiller, L. and Ozdemirel, N.E. (2008), "Profit-oriented disassembly-line balancing", *International Journal of Production Research*, Vol. 46, pp. 2675-93.

Baybars, I. (1986), "A survey of exact algorithms for the simple assembly line balancing problem", *Management Science*, Vol. 32, pp. 909-32.

Caridi, M., Cigolini, R. and Farina, V. (2006), "Designing unbalanced paced lines: a conceptual model and an experimental campaign", *Production Planning & Control*, Vol. 17, pp. 464-79.

Ding, L.-P., Feng, Y.-X., Tan, J.-R. and Gao, Y.-C. (2010), "A new multi-objective ant colony algorithm for solving the disassembly line balancing problem", *The International Journal of Advanced Manufacturing Technology*, Vol. 48, pp. 761-71.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 26, pp. 29-41.

Gungor, A. and Gupta, S.M. (1999), "Issues in environmentally conscious manufacturing and product recovery: a survey", *Computers & Industrial Engineering*, Vol. 36, pp. 811-53.

Gungor, A. and Gupta, S.M. (2001), "A solution approach to the disassembly line balancing problem in the presence of task failures", *International Journal of Production Research*, Vol. 39, pp. 1427-67.

Güngör, A. and Gupta, S.M. (2002), "Disassembly line in product recovery", *International Journal of Production Research*, Vol. 40, pp. 2569-89.

Gupta, S.M., Erbis, E. and McGovern, S.M. (2004), "Disassembly sequencing problem: a case study of a cell phone", in Gupta, S.M. (Ed.), *Environmentally Conscious Manufacturing IV*, SPIE-International Society for Optical Engineering, Bellingham.

Ilgin, M.A. and Gupta, S.M. (2010), "Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art", *Journal of Environmental Management*, Vol. 91, pp. 563-91.

Kalayci, C.B. and Gupta, S.M. (2011a), "A hybrid genetic algorithm approach for disassembly line balancing", *Proceedings of the 42nd Annual Meeting of Decision Science Institute* (*DSI 2011*)*, Boston, MA, USA*.

Kalayci, C.B. and Gupta, S.M. (2011b), "Tabu search for disassembly line balancing with multiple objectives", *41st International Conference on Computers & Industrial Engineering* (*CIE41*), University of Southern California, Los Angeles, CA.

Kalayci, C.B., Gupta, S.M. and Nakashima, K. (2011a), "A simulated annealing algorithm for balancing a disassembly line", *Proceedings of the Seventh International Symposium on Environmentally Conscious Design and Inverse Manufacturing* (*EcoDesign 2011*)*. Kyoto, Japan*.

Kalayci, C.B., Gupta, S.M. and Nakashima, K. (2011b), "Bees colony intelligence in solving disassembly line balancing problem", *Proceedings of the 2011 Asian Conference of Management Science and Applications* (*ACMSA2011*)*, Sanya, Hainan, China*.

Koc, A., Sabuncuoglu, I. and Erel, E. (2009), "Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph", *IIE Transactions*, Vol. 41, pp. 866-81.

McGovern, S.M. and Gupta, S.M. (2005), "Ant colony optimization for disassembly sequencing with multiple objectives", *The International Journal of Advanced Manufacturing Technology*, Vol. 30, pp. 481-96.

McGovern, S.M. and Gupta, S.M. (2006), "Ant colony optimization for disassembly sequencing with multiple objectives", *The International Journal of Advanced Manufacturing Technology*, Vol. 30, pp. 481-96.

McGovern, S.M. and Gupta, S.M. (2007), "A balancing method and genetic algorithm for disassembly line balancing", *European Journal of Operational Research*, Vol. 179, pp. 692-708.

McGovern, S.M. and Gupta, S.M. (2011), *The Disassembly Line: Balancing and Modeling*, McGraw-Hill, New York, NY.

McNamara, T., Shaaban, S. and Hudson, S. (2011), "Unpaced production lines with three simultaneous imbalance sources", *Industrial Management & Data Systems*, Vol. 111, pp. 1356-80.

Shaaban, S. and Hudson, S. (2011), "Transient behaviour of unbalanced lines", *Flexible Services and Manufacturing Journal*, December, pp. 1-28.

Tripathi, M., Agrawal, S., Pandey, M.K., Shankar, R. and Tiwari, M.K. (2009), "Real world disassembly modeling and sequencing problem: optimization by algorithm of self-guided ants (ASGA)", *Robotics & Computer-Integrated Manufacturing*, Vol. 25, pp. 483-96.

Vilarinho, P.M. and Simaria, A.S. (2006), "ANTBAL: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations", *International Journal of Production Research*, Vol. 44, pp. 291-303.

About the authors

Can B. Kalayci received his BS degree in Computer Engineering from Sakarya University, and his MS degree in Industrial Engineering from Pamukkale University. Being awarded with a full scholarship, he started his PhD studies in Industrial Engineering at Northeastern University, Boston, USA in January 2009. He is currently a PhD candidate in the Department of Industrial Engineering at Northeastern University. Can B. Kalayci's research interests are in the areas of combinatorial optimization, algorithms, heuristics, metaheuristics, scheduling, assembly and disassembly. He has co-authored several papers published in various conference proceedings. Can B. Kalayci has been employed as a research assistant at Pamukkale University since 2005.

Dr Surendra M. Gupta is a Professor of Mechanical and Industrial Engineering and Director of the Laboratory for Responsible Manufacturing at Northeastern University in Boston. He received his BE in Electronics Engineering from Birla Institute of Technology and Science, MBA from Bryant University and MSIE and PhD in Industrial Engineering from Purdue University. Dr Gupta is a recipient of the Outstanding Research Award and the Outstanding Industrial Engineering Professor Award. Dr Gupta's research interests are in the areas of production/manufacturing systems and operations research. He has authored/co-authored over 400 technical papers published in prestigious journals, books and conference proceedings. Surendra M. Gupta is the corresponding author and can be contacted at: gupta@neu.edu