ARTICLE IN PRESS

Journal of Cleaner Production xxx (2014) 1–15

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems

J. Mukund Nilakantan [a], George Q. Huang [b], S.G. Ponnambalam [a,*]

[a] Advanced Engineering Platform and School of Engineering, Monash University Malaysia, 46150 Bandar, Sunway, Malaysia
[b] Department of Industrial and Manufacturing Systems Engineering, Faculty of Engineering, The University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 5 May 2014
Received in revised form
12 November 2014
Accepted 13 November 2014
Available online xxx

Keywords:
Robotic assembly line balancing
Energy efficient assembly line
Cycle time
Particle swarm optimization

ABSTRACT

Manufacturing industries give importance to the reduction of energy consumption due to the increase in energy cost and to create an eco-friendly environment. Assembly line is considered to be one of the cost intensive systems. Robots are recently being used to perform the assembly tasks instead of manual labor. There is a requirement of efficiently balancing the assembly line by allocating equal amount of work to workstations and assignment of best fit robot to perform the tasks allocated to those workstations. The authors could not find any research on optimizing cycle time and total energy consumption concurrently in robotic assembly line systems to date. The objective of this paper is to propose models with dual focus on time and energy to minimize the cycle time and total energy consumption simultaneously, one model (time based model) with the primary focus to optimize cycle time and the other model (energy based model) with the primary focus to optimize total energy consumption. Particle swarm optimization is used as the optimization tool to solve this problem. Computational experiments are conducted on the proposed models using the benchmark problems available in the open literature and the results are presented. The two models proposed in this paper are very well applicable to automobile body shop with robot based lines. The models proposed have a significant managerial implication in real assembly line systems. Depending upon the priorities of the management, primary focus on reducing either cycle time or total energy consumption, suitable models could be selected. The proposed models are useful to reduce the total energy consumption and cycle time in robotic assembly lines. It is observed that the computation time for the time based model is less compared to energy based model.

1. Introduction

Energy is one of the most important vital resources in manufacturing scenario. Importance of energy efficiency has been realized in the recent years and stressed more than ever (Liu et al., 2014). Energy consumption is considered to be critical cost element in a manufacturing enterprise (Kilian, 2008). One of the major goal of many modern manufacturers in the recent years is to decrease the cost of production by any possible means while satisfying the environmental regulations and ensuring quality, and customer (Güngör and Gupta, 1999). Electricity is one of the important forms of energy which is used in a manufacturing sector. Production of electricity is a highly polluted process. Due to the consumption of electricity, amount of carbon dioxide emission generated would be around 20% (Dai et al., 2013). Thus manufacturing companies are required to reduce the energy consumption and become environment friendly. Due to the depletion of reserves of energy commodities such as petroleum and other fossils fuels and growing concern over global warming, there has been a growing interests for minimization of energy consumption by the industries (Mouzon and Yildirim, 2008).

Due to rise in energy price and increased demand for environmental compliant efficient energy management system and sustainable energy have become important factors for business competitive advantages. Reduced usage of energy helps the

Acronyms: ALB, Assembly Line Balancing; sALB, Simple Assembly Line Balancing; RALB, Robotic Assembly Line Balancing; NP, Non Polynomial; ACO, Ant Colony Optimization; EPC, Electric Power Cost; TOU, Time of use; FFS, Flexible Flow-shop Scheduling; GA, Genetic Algorithm; PSO, Particle Swarm Optimization.

* Corresponding author.
E-mail addresses: mukund.janardhanan@monash.edu (J. Mukund Nilakantan), gqhuang@hku.hk (G.Q. Huang), sgponnambalam@monash.edu (S.G. Ponnambalam).

http://dx.doi.org/10.1016/j.jclepro.2014.11.041
0959-6526/© 2014 Published by Elsevier Ltd.

industries to save cost and become more competitive. This is a key factor for promoting green and sustainable practices (Ngai et al., 2013). In a manufacturing industry, assembly line is considered to be one of the cost intensive processes. Using an energy efficient manufacturing system the energy consumption can be reduced (Chryssolouris, 2005).

Fysikopoulos et al. (2012) indicated that for manufacturing a car (Press, body, paint and assembly shops) could consume energy up to 700 kwh/vehicle. According to them, this energy cost is about 9–12% of the total manufacturing cost and a 20% reduction in energy cost shall be about 2–2.4% reduction in the final manufacturing cost. Extensive efforts are being under taken to improve the efficiency and cost effectiveness of assembly systems due to the economic importance of assembly as a manufacturing process (Sanderson et al., 1990).

Assembly line requires tedious, repetitive tasks which wearisome and some are dangerous for workers. Salveson (1955) mathematically formulated the model for Assembly Line Balancing (ALB) problem. Assembly Line Balancing (ALB) or simple Line Balancing (LB) problems mainly deals with assigning tasks to workstations in such a way that the assignment is in a balanced manner. While assigning the tasks to the workstations the precedence constraints are to be satisfied. Simple assembly line balancing (sALB) problems are of two types: simple ALB-I (sALB-I) and simple ALB-II (sALB-II). sALB-I mainly aims at assignment of tasks to workstations with the aim of minimizing the number of workstation whereas, sALB-II problem aims at minimizing the cycle time by assigning the tasks to the given set of workstations (Scholl, 1999).

Robots are being used extensively in assembly lines to perform the tasks and these assembly lines are called robotic assembly lines (Gao et al., 2009). The robots are programmed to perform different types of tasks and it can be used to work 24 hrs without worries of fatigue. Different types of robots are available in market to do the same task with different capabilities and efficiencies. Therefore, there is a requirement of judicious allocation of robots to workstations with certain specific objective. A typical robotic assembly line balancing (RALB) problem is to assign tasks to workstations and to allocate robot for each station in order to improve the productivity (Levitin et al., 2006).

Two types of RALB problems are addressed in the literature as type-I and type-II. Type I RALB problem aims at minimizing the number of workstations for a given cycle time of the assembly line. The type-II RALB problem aims at assigning tasks to workstations and to select the best fit robot for each workstation in such a way that cycle time is minimized (Gao et al., 2009). RALB problem is first formulated by Rubinovitz and Bukchin (1991) in which equal amount of tasks are allocated to the workstations in the assembly line and the best available robot to perform the tasks are assigned to the workstations with an objective of minimizing the number of workstations for a given cycle time.

Later, type-II RALB problem is developed with an aim of assigning tasks to work stations and select the best fit robot type for each workstation such that cycle time is minimized (Gao et al., 2009; Levitin et al., 2006). Yoosefelahi et al. (2012) formulated a multi-objective mixed integer linear programming model for type-II RALB problem aims to minimize the cycle time, robot setup costs and robot costs.

Literature on assembly line balancing problems shows that the key objectives evaluated are cost, time, and quality. However, the research on minimizing energy consumption in manufacturing systems has been rather limited (Dai et al., 2013). In case of RALB, most of the researchers considered only type-I and type-II robotic assembly line balancing problems for assigning tasks and allocating robots to workstations. Very limited researches related to assembly line problems in the context of minimizing energy consumption are available and few of them are briefly discussed below.

Fysikopoulos et al. (2012) showed that by modeling an assembly line in advance and by including energy considerations, one can possibly save energy and cost. An empirical study of the energy consumption of an automotive assembly line, under various scenarios and demand profiles is presented by them. (Luo et al., 2013) proposed a multi-objective ant colony optimization meta-heuristic to optimize not only production efficiency but also electric power cost (EPC) with the presence of time-of-use (TOU) electricity prices. Dai et al. (2013) proposed an energy-efficient model for flexible flow shop scheduling (FFS). A mathematical model for an FFS problem which is based on an energy-efficient mechanism is described by them. Due to NP-hard nature of the problem, an improved genetic-simulated annealing algorithm is adopted by them to make a significant trade-off between the make-span and the total energy consumption to implement a feasible scheduling.

Mouzon et al. (2007) developed several algorithms and a multi-objective mathematical programming model for investigating a problem of scheduling jobs on a single CNC machine in order to reduce energy consumption and completion time.

Shrouf et al. (2013) proposed a mathematical model to minimize the energy consumption cost for a single machine production system considering variable energy prices during a day. They proposed a genetic algorithm (GA) to obtain ‘near’ optimal solutions. They evaluated the performance of the proposed GA with an analytical solution generated. He et al. (2012) proposed a modeling method of task oriented energy consumption for machining manufacturing system by incorporating an event graph methodology. They solved the model using SIMULINK simulation environment.

Assembly line balancing problem belongs to the category of NP-hard (Gutjahr and Nemhauser, 1964). In the literature, researchers use optimization or simulation models to solve assembly line balancing problems. Researchers in the recent past have suggested that both optimum-seeking and heuristic algorithms can be used to solve single-model assembly line balancing problems. Exact methods guarantee an optimum solution, whereas heuristic methods only attempt to yield a good, but not necessarily optimum solution. But, the time taken by an exact method to find an optimum solution for NP-hard problem will be much greater than the time taken by any heuristic method. Hence, heuristic methods are used to solve real optimization problems which are generally complex (Martí and Reinelt, 2011).

There are limited literature available in both the categories. The available literature are presented in this section.

Rubinovitz et al. (1993) first introduced a branch and bound algorithm for RALB type I problem. An exact heuristic branch and bound algorithm is used to obtain optimal solutions for small and medium-sized ALB problem instances (Bukchin and Tzur, 2000). Rashid et al. (2012) presented a list of heuristic algorithms applied to ALB problem. According to them only limited heuristic algorithms are applied to RALB problems. So far, genetic algorithm (Levitin et al., 2006), hybrid genetic algorithm (Gao et al., 2009) and particle swarm optimization (Mukund Nilakantan and Ponnambalam, 2012) have been used to solve RALB problems to optimize the cycle time.

Daoud et al. (2014) proposed several evolutionary algorithms and a discrete event simulation model to solve robotic assembly line balancing problem and they considered an automated packaging line dedicated for dairy food products as the case study.

Villarreal and Alanis (2011) presented a simulation approach that could improve the efforts during the redesign of a traditional assembly line system applicable to an assembly system in Mexico.

Rajamani and Singh (1991) presented a case study of the design and balancing of TV assembly line. The authors apply an algorithm using a cost model in a stochastic environment for assignment of tasks to workstations. Using SLAM-II a model is developed for determining the in-process storage capacity for physical layout and also to estimate the work in process inventory levels for guaranteeing an adequate flow in the line. From the simulation model a regression model is also developed for finding out the average time spend on a television in the assembly system.

Das et al. (2010) developed a simulation model for a realistic case problem and applied to a six-station assembly line. The objectives evaluated are: minimization of the total elapsed time; maximization of the average percentage of working time; and minimization of the average time in the system. Bowl phenomenon is studied and their applications in assembly line balancing problem are presented.

There are certain issues in simulation approaches. Simulation cannot naturally be used to find an optimal solution. There are methods which long to optimize the result, but simulation is not inherently an optimization tool. It is also quite expensive to develop simulation models. In any system (or model), you have parameters and other conditions (e.g. initial conditions) with make results different. In simulation, values of these effective parameters must be specified and result will be true just for that value of parameter. But optimization models result, give a general description of system for any value of parameters. Because of these issues, the motivation in this research is towards developing an optimization model and proposing a heuristic search algorithm to generate better solution for RALB problems.

The main objective of this paper is to propose an optimization model to optimize time and energy and to solve the model using a heuristic algorithm. Particle Swarm Optimization algorithm is used to optimize the objectives of two models proposed. Time based model optimizes the cycle time of the robotic assembly line as the primary objective and the total energy consumption as the secondary objective. Similarly, energy based model optimizes the total energy consumption of the robotic assembly line as the primary objective and cycle time as the secondary objective. The performance of the time based model and the energy based model are evaluated and the results are presented in tables and figures. The model proposed in this paper is very well applicable to automobile body shop with robot based lines.

The remainder of this paper is structured as follows. Section 2 provides the assumptions and the mathematical model of the problem. Section 3 explains PSO in detail. Section 4, presents the results of the experiments conducted. Finally, Section 5 concludes the finding of this research.

2. Mathematical formulation for RALB problems

In an assembly line, different assembly tasks are to be performed by each workstation to assemble and produce a given product, while precedence constraints of the tasks are specified. A set of workstations and robots are considered in the assembly line. In a balanced assembly line, tasks needs to be assigned to the workstations and best robot needs to be allotted to the station to perform the assembly tasks. The following assumptions are considered in the model formulation which are similar to those mentioned in (Gao et al., 2009; Levitin et al., 2006).

1. Robots power consumptions are assumed. Using the power of each robot, energy consumption is calculated.
2. Model is designed for a straight assembly line for a unique model of a single product.
3. Tasks cannot be subdivided among one or two work stations and precedence relationship should be met while assigning the tasks.
4. Time taken to perform a task depends on the type of robot assigned.
5. A robot type which could perform the tasks assigned to a station in the least time among other type of robots is considered to assign it to a station.
6. There is no limitations on assignment of a task or a robot to any workstation other than the precedence constraints and the robots ability to perform the task.
7. The number of work stations will be equal to number of robot types.
8. Material handling, loading and unloading time, as well as set-up and tool changing time are negligible, or are included in the task time. This assumption is realistic on a single-model assembly line. Tooling on such robotic line is usually designed such that tool changes are minimized within a work station. If tool change or other type of set-up task is necessary, it can be included in the task time.
9. The planning horizon is not included in the model. The proposed algorithm and the models are tested using the benchmark problems available in the literature. Hence, the maintenance operations are not considered in this study.

Levitin et al. (2006) used the fifth assumption listed above in the illustration presented in their paper. The two main objectives mentioned in their paper are: Optimal balance of the assembly line and allocation of the best fit robot to each workstation. Achieving these two objectives are possible with the fifth assumption. Since this consideration helps to reduce the cycle time of the assembly line and assign a best fit robot to workstations, it is followed in this paper.

Gao et al. (2009) presented a zero-one integer programming formulation for type-II RALB, which is presented below. Equation (5) of their model is relaxed in this paper to achieve the objectives mentioned by Levitin et al. (2006).

2.1. Zero-one integer formulation of RALB

The notations used in this paper are presented in the Appendix A.

- Decision variables

$$x_{is} = \begin{cases} 1 \text{ if task i is assigned to workstation } s \\ 0, \text{otherwise} \end{cases}$$

$$y_{sh} = \begin{cases} 1 \text{ if robot h is allocated to workstation s} \\ 0, \text{otherwise} \end{cases}$$

$$\min c = \max_{1\le s\le N_w} \left\{ \sum_{i=1}^{N_a} \sum_{i=1}^{N_w} t_{ih} \cdot x_{is} \cdot y_{sh} \right. \tag{1}$$

$$s.t. \sum_{s=1}^{N_w} s \cdot x_{is} - \sum_{s=1}^{N_w} s \cdot x_{js} \le, 0 \quad \forall i \in \text{pre}(j); j \tag{2}$$

$$\sum_{s=1}^{N_w} x_{is} = 1 \quad \forall i \tag{3}$$

$$\sum_{s=1}^{N_w} y_{sh} = 1 \quad \forall s \tag{4}$$

Please cite this article in press as: Mukund Nilakantan, J., et al., An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems, Journal of Cleaner Production (2014), http://dx.doi.org/10.1016/j.jclepro.2014.11.041

$$\sum_{s=1}^{N_w} y_{sh} = 1 \quad \forall h \tag{5}$$

$$x_{is} \in \{0,1\} \quad \forall s, i \tag{6}$$

$$y_{sh} \in \{0,1\} \quad \forall h, s \tag{7}$$

The objective 1 is to minimize the cycle time of the robotic assembly line. Equation (2) defines the precedence relationship among the tasks. It ensures that for a pair of tasks with precedence relation, the precedent cannot be assigned to a workstation after the one to which its successor is assigned. Equation (3) ensures that each task has to be assigned to one workstation and Equation (4) ensures that each workstation is equipped with one robot. Equation (5) ensures that each robot can only be assigned to one workstation. It is notable that objective 1 is non-linear, hence it is hard for traditional exact optimization techniques to solve the problem.

The objective of the energy based model (Equation (8)) is to minimize the total energy consumption. For the energy based model, Equation (1) is replaced by Equation (8).

$$\min E = \max_{1 \le s \le N_w} \left\{ \sum_{i=1}^{N_a} \sum_{i=1}^{N_w} e_{ih} x_{is} y_{sh} \right\} \tag{8}$$

3. PSO algorithm for robotic assembly line balancing

Since assembly line balancing problem is a well-known as an NP-hard problem, a particle swarm optimization algorithm, a swarm based heuristic algorithm is proposed in this paper. The implementation detail of PSO is discussed in this section.

PSO is a swarm-based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling (Eberhart and Kennedy, 1995). The swarm evolves continuously and tries to find the optimal solution. PSO consists of a number of particles moving around in the search space, each representing a possible solution. Each particle is searching for the optimal solution and they move around with certain velocity. Particle remembers the best fitness encountered by them (*Local best*) and exchanges information with other particles to determine the best particle (*Global best*) among the swarm. Pseudo code of PSO is shown in Fig. 1. Here v_i^t is the initial velocity, v_i^{t+1} is the updated velocity, ${}^eP^t{}_i$ is the Local best, G is the Global best and P_i^t is the current particle position.

$$
\begin{array}{l}
t \to 0; \\
\quad for(i = 1, N) \\
\quad\quad Generate\ \ P_i^t; \\
\quad\quad Evaluate\ Z\ (P_i^t); \\
\quad\quad {}^eP_i^t \ \to P_i^t; \\
\quad end\ i \\
\quad G \to P_i^t\ \ having\ min\,\{Z\left({}^eP_i^t\right);\ \ i = 1, N\ \} \\
\quad for(i = 1, N) \\
\quad\quad Initialize\ \ v_i^t; \\
\quad end\ i \\
\quad do\ \{ \\
\quad\quad\quad for(i = 1, N) \\
\quad\quad\quad\quad Update\ Position\ \ P_i^{t+1}\ (Eq.(19)\ ) \\
\quad\quad\quad\quad Update\ Velocity\ \ v_i^{t+1}\ \ (Eq.(20)) \\
\quad\quad\quad end\ i \\
\quad\quad\quad Evaluate\ all\ particles \\
\quad\quad\quad Update\ {}^eP_i^t\ \ and\ G, (i = 1, N) \\
\quad\quad\quad t\ \ \to t + 1 \\
\quad \}\ (while\ (t < t_{max})) \\
\quad Output\ G
\end{array}
$$

Fig. 1. Pseudo code of PSO.

3.1. Solution representation and fitness evaluation

A sequence represents a solution which corresponds to a particle. An example sequence (solution) for 11 tasks RALB problem is shown in Fig. 2a. Each integer in a sequence represents the task in an assembly line which needs to be performed by a robot in a particular work station. The number of tasks to be assigned to each work station is based on the cycle time for time based model and the energy consumption for energy based model. The tasks and robot allocation after decoding the sequence are also shown in Fig. 2b.

Each integer in the sequence shown in Fig. 2a is ordered according to their technological precedence constraints. Assigning the tasks to workstation and the robot assignment for the workstation is done using an approach named consecutive assignment procedure. Same procedure is used in time based model and in energy based model. Cycle time is used as the fitness value for time based model and energy consumption is used as the fitness value for energy based model. Tasks and robots are assigned to workstation using the consecutive heuristic procedure proposed by Levitin et al. (2006). The task and robot assignments for time based and energy based models are explained in Section 3.2 and 3.3.

3.2. Time based model – task and robot assignment procedure

The procedure starts with considering an initial cycle time for the assembly line, C_0. Tasks are assigned to the workstation until the sum of robot task times of any robot is less than or equal to C_0 and the respective robot is assigned to that workstation. If it is not possible to allocate all the tasks to workstations for the given initial C_0 value, C_0 is incremented by 'one' and the assignment procedure is repeated until all the tasks are assigned to all the workstations. An illustration is provided in this section which explains the task and robot allocation for the time based model. Precedence graph

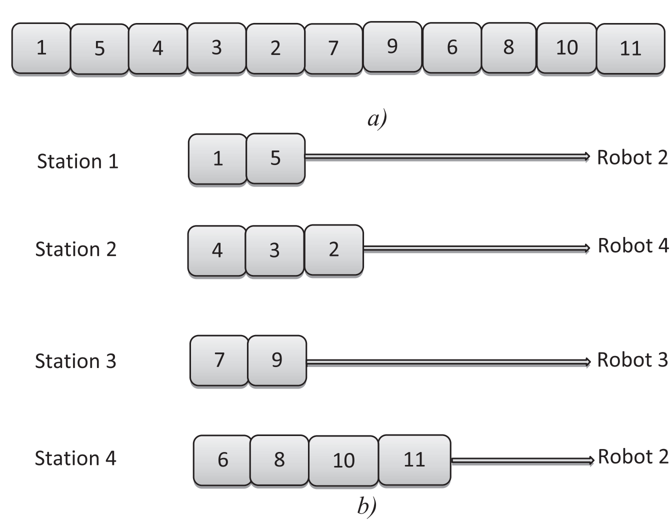

Fig. 2. a) Sample Task Sequence b) Tasks assigned after decoding the sequence.

1	3	2	4	5	6	7	9	8	10	11

Fig. 3. Example feasible sequence for 11 task problem.

shown in Fig. 5 and robot processing time shown in Table 1 is used for illustration purpose.

Step 1. The following feasible sequence of tasks (Fig. 3) is considered for illustration. The sequence meets the precedence constraints shown in Fig. 5.

Initial value of C_0 is the mean of the minimum processing time of robots for all the tasks, which is calculated using Equation (9).

$$C_0 = \left[\sum_{j=1}^{N_a} \min_{1 \le i \le N_r} t_{ih} \middle/ N_w \right] \tag{9}$$

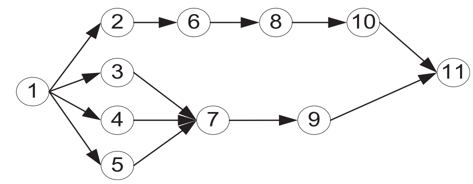

Fig. 5. Precedence Graph of the sample problem with 11 tasks.

The robot processing times are shown in Table 1. The initial C_0 for the example is found out to be 109. $C_0 = [37 + 42+38 + 40+25 + 65+40 + 34+33 + 41+38]/4 = 108.25$.

Step 2. Open a station and allocate the tasks in the sequence in the order of occurrence, if one or more robot could perform the allocated tasks within C_0.

Each workstation‘s’ has a set of preferred/allotted robots H which is defined as follows:

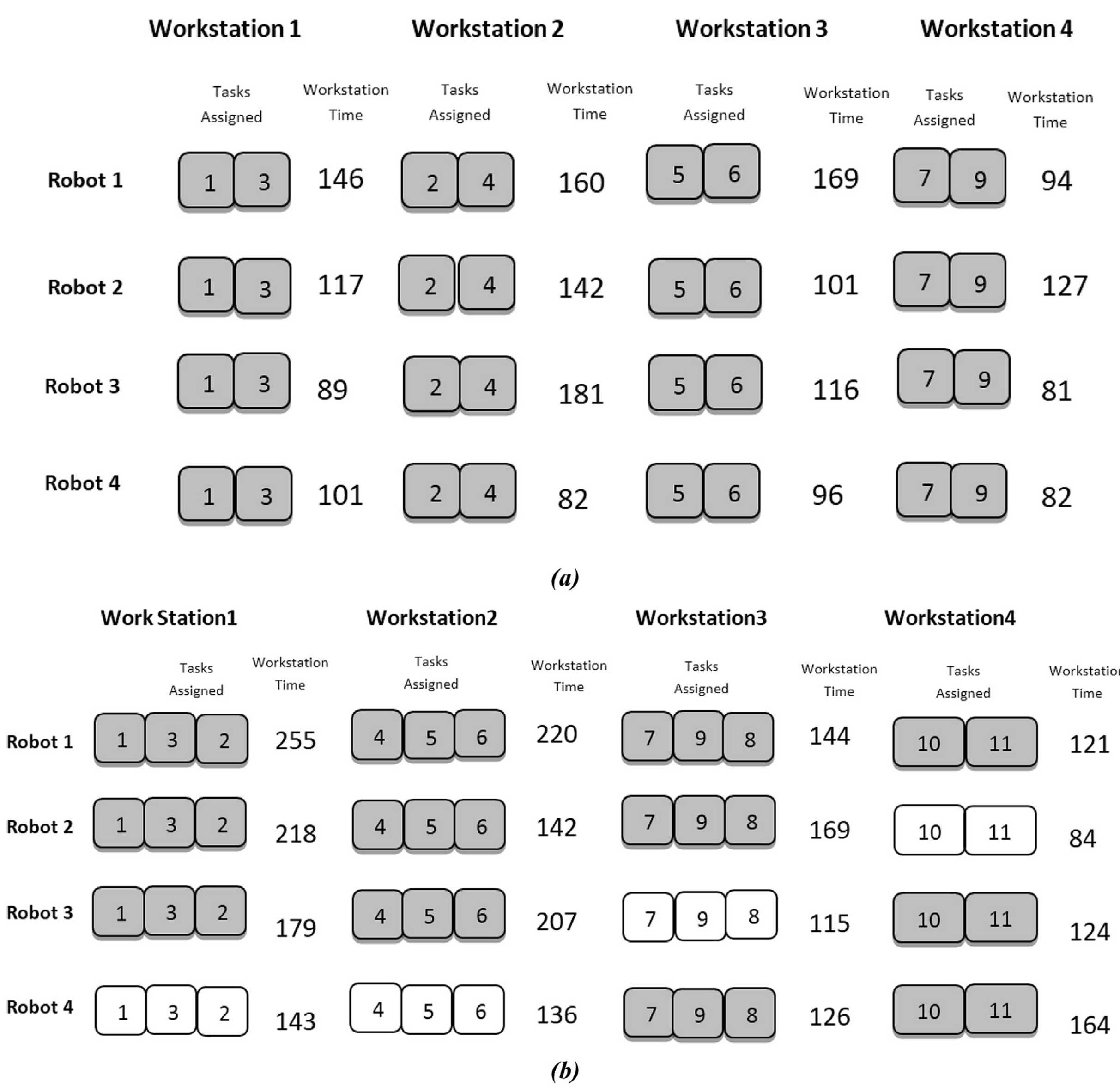

Fig. 4. a) Task and Robot allocation using time based consecutive method with initial C_0 = 109 b) Completed Task and Robot allocation for the given sequence using time based consecutive method when C_0 = 143.

Table 1
Processing times for 11 tasks by 4 robots.

Tasks	Robot 1	Robot 2	Robot 3	Robot 4	Average time
1	81	37	51	49	54.5
2	109	101	90	42	85.5
3	65	80	38	52	58.75
4	51	41	91	40	55.75
5	92	36	33	25	46.5
6	77	65	83	71	74
7	51	51	40	49	47.75
8	50	42	34	44	42.5
9	43	76	41	33	48.25
10	45	46	41	77	52.25
11	76	38	83	87	71

$$k \in H, \text{if } m(k) \geq m(h), \text{for } 1 \leq h \leq N_r \tag{10}$$

where, $m(h)$ is the maximal number of tasks a robot h can perform in the given sequence sq during a time lesser than C_0.

$$T_s(h) = \sum_{k=p1_s}^{p1_s+m(h)} t_{h,sq(k)} < C_0 \leq \sum_{k=p1_s}^{p1_s+m(h)+1} t_{h,sq}(k) \tag{11}$$

Next, it defines the robot to be assigned to the workstation s as:

$$h(s) = k \text{ if } T_s(k) < T_s(h) \quad \forall h \in H \tag{12}$$

If no further assignment of tasks and robots to the current workstation is possible, go to Step 3.

Step 3. The start of position of the next workstation is calculated as follows:

$$p1_{s+1} = pr_s + 1 = p1_s + m(h(s)) + 1 \tag{13}$$

Repeat Step 2, until all tasks are assigned to given number of workstations.

Step 4. If all task are not possible to be assigned to the given number of workstations, increment C_0 *by 'one'* and repeat step 2 to 3 until all tasks are allotted to the given number of workstations.

Step 5. Cycle time is calculated. The maximum of the workstation time is the cycle time for the assignment made. Workstation time is the sum of robot processing times of the tasks by the allotted robots.

Step 6. The total energy consumption for the assignment made is calculated. Energy consumption for a workstation is calculated by multiplying the total robot task time by the power consumed by that robot. Total energy consumption for the assignment is obtained by adding the energy consumed by all workstations.

In the example, when C_0 is 109, it is found that tasks 8, 10 and 11 are left unassigned as shown in Fig. 4a. To accommodate all these tasks C_0 is incremented by one and procedure is repeated. When C_0 reaches 143 as shown in Fig. 4b, all tasks are assigned to the four workstations.

Energy consumption of each workstation is calculated as follows:

Energy consumption at a workstation = Workstation time× Power Consumption of the robot.
Energy consumption at Workstation *1* = 143 × 0.35 = 50.05 kJ.
Energy consumption at Workstation *2* = 136 × 0.35 = 47.6 kJ.
Energy consumption at Workstation 3 = 115 × 0.3 = 34.5 kJ.
Energy consumption at Workstation 4 = 84 × 0.4 = 33.6 kJ.

Total energy consumption of the assignment = 50.05 + 47.6+34.5 + 33.6 = 165.7 kJ and the cycle time of the assignment is 143. Refer Fig. 4b for the workstation times and Appendix B for the power consumption values of the robots. The results obtained for the 11 tasks problem using time based model is shown in Fig. 6.

3.3. Energy based model- Task and Robot assignment procedure

Procedure starts with considering an initial energy consumption of the assembly line, E_0. Tasks are assigned to the workstation until the sum of the energy consumption of the tasks is less than or equal to E_0 and that respective robot is assigned to the workstation to perform the tasks. If it is not possible to assign the tasks to the workstation within the given initial E_0 value, E_0 is incremented by *'one'* and the assignment procedure is repeated until all the tasks are assigned to the workstations. Stepwise illustration is provided for explaining the task and robot allocation using the energy based model. Precedence graph shown in Fig. 5 and energy consumption of the robots shown in Table 2 are used for the illustration.

Step 1. Feasible sequence of tasks (Fig. 3) which meets the precedence constraints is considered for illustration. Initial value of E_0 is the mean of minimum energy consumption of the robots for the tasks. The initial energy consumption (E_0) of the assembly line is calculated using Equation (14).

$$E_0 = \left[\sum_{j=1}^{N_a} \min_{1 \leq i \leq N_r} e_{ih} / N_w \right] \tag{14}$$

Using the energy consumption of the robots shown in Table 2, the initial E_0 for the example is found out to be 35. $E_0 = [15 + 15+11 + 13+9 + 19+12 + 10+11 + 11+15]/4 = 35$.

Step 2. The first station is opened and the procedure tries to allocate the tasks according to the sequence in the order of occurrence, if one or more robot could perform the allocated tasks within E_0. Each workstation s has a set of preferred/ allotted robots H which is defined as follows:

$$k \in H, \text{if } m(k) \geq m(h), \text{for } 1 \leq h \leq N_r \tag{15}$$

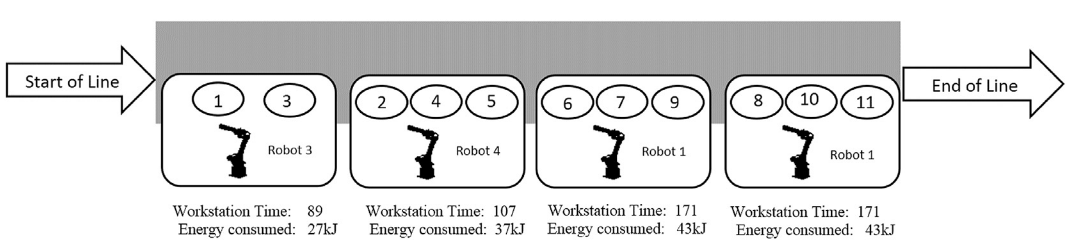

Fig. 6. Solution for the 11 task problem (time based model).

Table 2
Energy consumption for 11 tasks by 4 robots.

Tasks	Robot 1	Robot 2	Robot 3	Robot 4
1	20	15	15	17
2	27	40	27	15
3	16	32	11	18
4	13	16	27	14
5	23	14	10	9
6	19	26	25	25
7	13	20	12	17
8	13	17	10	15
9	11	30	12	12
10	11	18	12	27
11	19	15	25	30

where $m(h)$ is the maximal number of tasks a robot h can perform in the given sequence sq within E_0.

$$E_s(h) = \sum_{k=p1_s}^{p1_s+m(h)} e_{h,sq(k)} < E_0 \leq \sum_{k=p1_s}^{p1_s+m(h)+1} e_{h,sq}(k) \quad (16)$$

Next, it defines the robot to be assigned to the workstation s as:

$$h(s) = k \text{ if } E_s(k) < E_s(h) \quad \forall h \in H \quad (17)$$

Step 3. The start of position of the next workstation is calculated as follows:

$$p1_{s+1} = pr_s + 1 = p1_s + m(h(s)) + 1 \quad (18)$$

Step 2 is to be repeated until all tasks are assigned to given number of workstations.

Step 4. If the tasks are not possible to be assigned within the given E_0, increment E_0 by *'one'* and repeat the steps 2 and 3 until all the tasks are allotted.
Step 5. The total energy consumption for the assignment is calculated by adding the energy consumption of all workstation.
Step 6. The workstation time of the assignment made is calculated by finding out the time taken by the robot at each workstation. The maximum workstation time is the cycle time of the assignment made.

In the example for energy based model, it is found that tasks 10 and 11 are left unassigned as when $E_0 = 35$ as shown in Fig. 7a. E_0 is incremented till 43 for accommodating all the tasks in the four workstations and the completed allocation is shown in Fig. 7B. The total energy consumption of the assembly line when allocated based on the energy model is found to be 150 kJ (27 + 37+43 + 43). The workstation time of the each workstation is calculated by adding the processing time of the tasks assigned to that workstation for the robot assigned. In this example workstation time of the assignment made is calculated by using the processing time given in Table 1.

Time at Workstation 1(*Robot* 3) = 51 + 38 = 89.
Time at Workstation 2(*Robot* 4) = 42 + 40+25 = 107.
Time at Workstation 3(*Robot* 1) = 77 + 51+43 = 171.
Time at Workstation 4(*Robot* 1) = 50 + 45+76 = 171.

The maximum workstation time is the cycle time of the assignment made and the cycle time is 171. The results obtained for the 11 tasks problem using time based model is shown in Fig. 8.

3.4. *Energy consumed by robots during standby mode*

Standby energy is the energy consumed by the robots or any electrical equipment when it is switched off or not performing its main function. Standby power consumption represents an increasing fraction of energy use in Organization for Economic Cooperation and Development (OECD) countries. Increased usage of new technologies results in growth of stand-by power usage. Reduction of stand-by energy consumption worldwide could reduce CO_2 emissions by one percent. For calculation of energy consumed during standby mode the power consumption of robot is considered to be 10% of the original power (Bertoldi et al., 2002). This is an assumption considered in this paper to calculate the energy consumed during standby time. In an assembly line, line is said to be balanced if total slack (i.e., the sum of the stand-by times of all the stations along the line) is as low as possible. But in practical cases it is difficult to get 100% efficiently balanced assembly line. In this research, the energy consumed by the workstations during the (standby mode) is also considered. Standby time is calculated as follows:

Standby time of the work station
= Cycle Time of the assembly line − workstation time

The steps involved in calculation of energy consumed in an assembly line is as follows:

1. Calculate Cycle Time of the assembly line.
2. Find the standby time of each workstation.
3. Calculate the standby energy consumed by the robots allotted to the workstation. Energy is calculated by using ($E = P \times t$) here P is the power of the robot allotted to the workstation and t corresponds to the standby time of the workstation. Sum up all the standby energy of all the workstations in an assembly line.

Table 3 shows an example problem with 9 workstations. Column 1 shows the workstation number, Column 2 shows the robots assigned after task allocation. Column 3 and 4 shows the workstation time and standby time of the workstation. In this problem cycle time is found to be 110. Using the standby time and power of the robot (Refer Appendix B), standby time energy is calculated. The standby time energy is evaluated for both the models and it is added to the energy consumption during the production time to get the total energy consumed in an assembly line. Both the models employ the same procedure for the evaluation of the standby energy consumption.

3.5. *Key steps in PSO*

The key steps involved in PSO are explained in this section.

3.5.1. *Initial particle generation*

PSO starts the search process with a group of randomly generated sequences called particles. Initial swarm size used in this paper is 25. Initially six particles are generated using the six heuristic rules mentioned in (Ponnambalam et al., 2000). The remaining particles satisfying the precedence constraints are generated randomly. Table 4 shows the six particles generated using the heruristic rules. Precedence graph shown in Fig. 5 and processing time shown in Table 1 are used for illustration purpose.

3.5.2. *Velocity generation*

Inital velocity for the particles are randomly generated and they represent the number of pairs of transpositions. Table 5 shows the

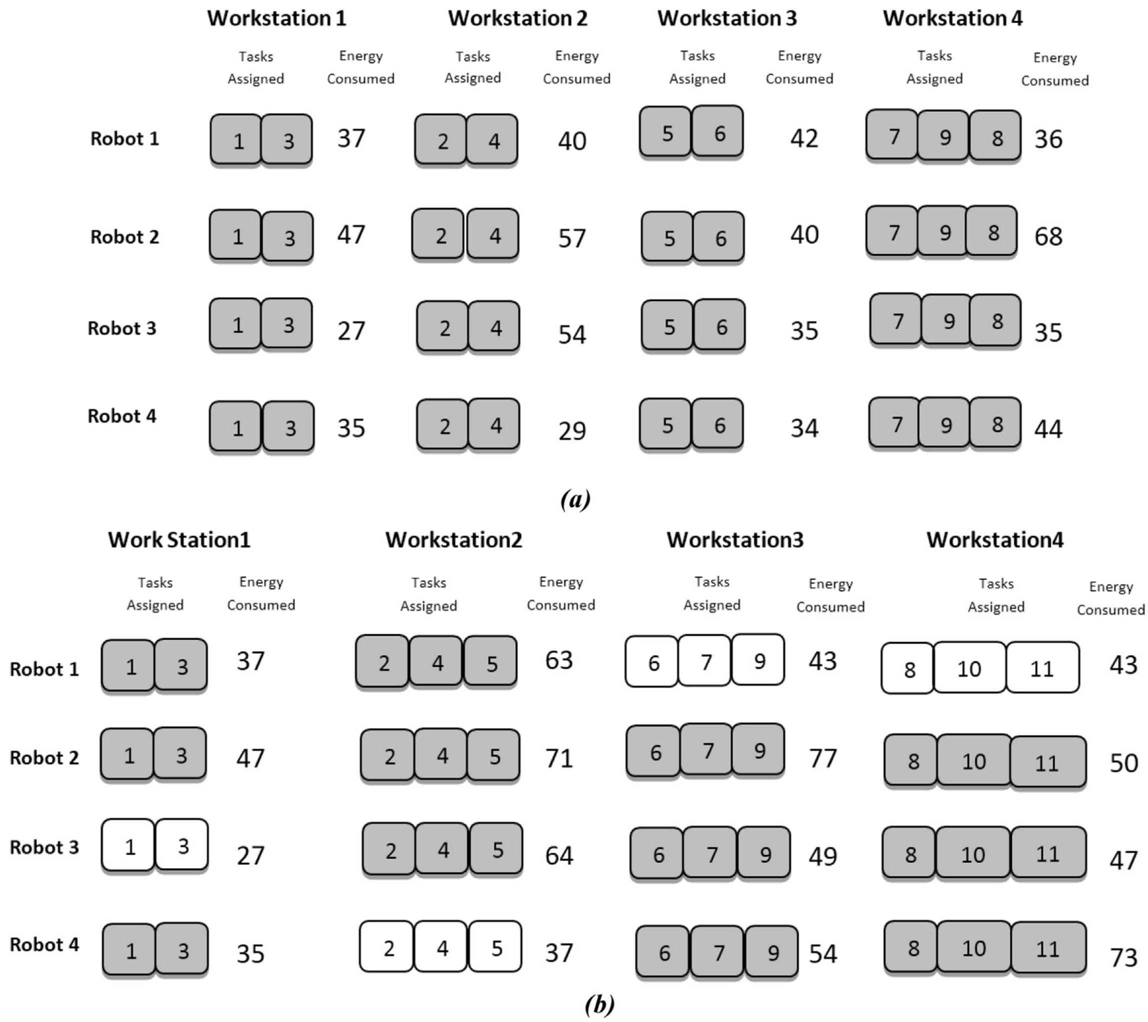

Fig. 7. a) Task and Robot allocation using energy based consecutive method with initial $E_0 = 35$ b) Completed Task and Robot allocation for the given sequence using energy based model when $E_0 = 43$.

maximum number of velocity pairs used in this paper. The velcoity update Equation (20) is used from the second iteration onwards.

3.5.3. Position and velocity update

Position update is done iteratively using the Equation (19).

$$P_i^{t+1} = P_i^t + v_i^{t+1} \tag{19}$$

The velocity of each particle is updated iteratively using the Equation.

$$v_i^{t+1} = v_i^t + c_1 U_1 \left({}^eP_i^t - P_i^t\right) + c_2 U_2 \left(G - P_i^t\right) \tag{20}$$

Where U_1 and U_2 are the velocity coefficients (random numbers between 0 and 1), v_i^t is the initial velocity, ${}^eP^t_i$ is the Local best, G is the Global best and P_i^t is the current particle position, c_1 and c_2 are the learning coefficients, Local best and Global best, respectively. Each particle adjusts its trajectory toward its own best position and the best position attained by swarm, namely Local Best and Global Best (Marinakis and Marinaki, 2010).

An example is shown to explain the updates of velocity and position.

$$\textit{Local Best}\ {}^eP_i^t: \ (1, 2, 6, 3, 4, 5, 7, 8, 10, 9, 11),$$
$$\textit{Global Best}\ G: \ (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11),$$

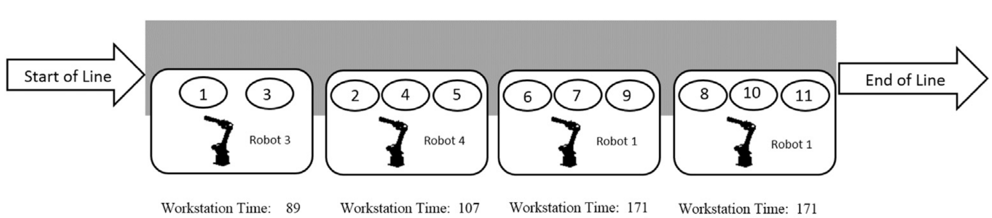

Fig. 8. Solution for the 11 task problem (time based model).

Table 3
Standby time Energy Evaluation.

Workstation number	Robot assigned	Workstation time	Standby time	Standby energy (kilojoules)
1	4	59	51	2.04
2	7	109	1	0.025
3	4	92	18	0.72
4	9	**110**[a]	0	0
5	7	104	6	0.15
6	4	109	1	0.04
7	7	87	23	0.575
8	7	98	12	0.3
9	7	98	12	0.3
				Total standby energy: 4.15 kJ.

[a] Cycle Time

Table 4
Initial population generated using the heuristic rules.

Rule number	Rule	Task sequence
1	Maximum ranked positional weight	1 2 6 3 4 5 7 8 10 9 11
2	Minimum reverse positional weight	1 5 4 3 2 7 9 6 8 10 11
3	Minimum total number of predecessor tasks	1 2 3 4 5 6 8 10 7 9 11
4	Maximum total number of follower tasks	1 2 3 4 5 6 7 8 9 10 11
5	Maximum task time	1 5 2 6 3 4 7 8 10 9 11
6	Minimum task time	1 4 3 2 5 7 9 6 8 10 11

$$Particle\ P_i^t : (1,2,3,6,5,4,7,8,10,9,11)\ and\ Initial\ velocity : (2,\ 3)\ (4,\ 5).$$

Learning coefficients: $c_1 = 1$, $c_2 = 2$ and U_1 and U_2 are generated randomly. Assuming $U_1 = 0.8$ and $U_2 = 0.3$.

Velocity of the particle is calculated using Equation (20).

$$\begin{aligned} v_i^{t+1} &= (2,3)(4,5) + 0.8[(1,2,6,3,4,5,7,8,10,9,11) \\ &\quad - (1,2,3,6,5,4,7,8,10,9,11)] \\ &\quad + 0.6[(1,2,3,4,5,6,7,8,9,10,11) \\ &\quad - (1,2\ ,3,6,5,4,7,8,10,9,11)] \\ &= (2,3)(4,5) + 0.8(2,3)(4,5) + 0.6(3,5)(8,9) \\ &= (2,3)(4,5)(8,9) \end{aligned}$$

Using Equation (19) position of the particle is updated.

$$\begin{aligned} P_i^{t+1} &= (1,2,3,6,5,4,7,8,10,9,11) + (2,3)(4,5)(8,9) \\ &= (1,3,2,5,6,4,7,10,\ 8,9,11) \end{aligned}$$

Table 5
Maximum size of velocity pairs.

Task range	Maximum velocity pairs
0-20	4
20-40	8
40-60	10
60-80	25
80-100	30
100-120	40
120-140	50
140-200	65
200-300	75

Table 6
PSO Parameters selected for evaluating the models.

Time based Model	Energy based Model
Population size: 25 Number of iterations: 30 Learning coefficients: c_1-1 and c_2-2	Population size 25 Number of iterations:40 Learning coefficients: c_1-2 and c_2-2

Table 7
Total energy consumption and cycle time evaluated using two models for small datasets.

Small datasets	Time based Model		Energy based model	
	Total energy consumption (kilojoules)	Cycle time	Total energy consumption (kilojoules)	Cycle time
25-3	514	503	494	641
25-4	347	293	342	314
25-6	420	221	365	235
25-9	265	110	248	142
35-4	1091	341	1072	516
35-5	959	357	929	424
35-7	1180	226	1015	342
35-12	755	105	697	160
53-5	2707	454	2700	587
53-7	2197	293	1989	343
53-10	2513	224	2215	273
53-14	2237	146	2177	200
70-7	4218	446	4146	463
70-10	3228	259	3069	290
70-14	4092	194	3871	290
70-19	3732	139	3323	255

4. Results and Discussion

The computational experiments are conducted in order to test the performance of the two proposed models for RALB problem. PSO is proposed to solve RALB. The following section describes the experiments conducted.

4.1. Data for time based model and energy based model

Gao et al. (2009) generated 32 test problems for RALB using 8 precedence graphs available in http://www.assembly-line-balancing.de/and the time data used by (Scholl, 1999). These

Table 8
Total energy consumption evaluated using two models for large datasets.

Large datasets	Time based model		Energy based model	
	Total energy consumption (kilojoules)	Cycle time	Total energy consumption (kilojoules)	Cycle time
89-8	5078	464	5043	562
89-12	6314	317	5683	430
89-16	5191	219	5119	340
89-21	4734	176	4250	206
111-9	4734	526	4250	735
111-13	8207	317	7267	396
111-17	7403	250	6945	280
111-22	7400	185	6909	255
148-10	10,166	556	9840	678
148-14	12,045	420	10,654	461
148-21	11,467	272	10,131	335
148-29	9290	190	8606	263
297-19	26,849	594	25,232	809
297-29	26,161	428	24,970	466
297-38	25,450	295	22,862	348
297-50	24,870	256	22,243	348

datasets are used in this research to evaluate the two models. Power consumption by robots are randomly generated and it is shown in Appendix B for small datasets and for large datasets it is shown in Appendix C. The energy consumption of a task i by robot h is calculated as follows:

$$E = P_h \times t_{ih} \tag{21}$$

Where, t_{ih} is processing time of the task i by robot h. P_h is the power consumption of robot h. The datasets are divided into two groups: small and large datasets. Small dataset contains problems with 25 tasks to 70 tasks. Large datasets consists of problem with tasks 89 tasks to 297 tasks.

4.2. Performance of two models proposed

The 32 test problems are solved for the two model proposed using the PSO algorithm. The parameters used in PSO are chosen experimentally in order to get a satisfactory solution quality in an acceptable time span. It is well recognized that the parameter values significantly affect the solution quality. Experiments are performed to find the optimal parameters. Three data sets of different sizes are chosen to find the parameters which yielded best solution. Different combinations of the parameters are tested until the best combination is achieved. Quality of solution is given importance compared to the computational time in selecting the parameters. PSO Parameters chosen in this paper are shown in Table 6.

The two models are evaluated on the two factors using cycle time and total energy consumption. Tables 7 and 8 show the total energy consumption and cycle time obtained using the two models. Table 7 shows the results for small datasets and Table 8 shows the results for larger datasets. Total energy consumption evaluated is the sum of energy consumption during the production mode and standby mode. Figs. 9 and 10 show the graphical comparison of the energy consumption for both small and large sized datasets. From the graphs and tables it is observed that the total energy consumption is lower for energy based model for all the datasets reported here.

The average of the difference in the energy consumption between the models is taken to find out the average energy saving. And the average energy saving is found to be around 113 kJ for small datasets The average energy savings for the large datasets is found to be 960 kJ Fig. 11 represents the difference in energy consumption between the time based model and energy based model and the graph represents the energy saved by using energy based model when compared to the time based model. It is observed could be analyzed from the graph (Figs. 11 and 12) that datasets with more amount of tasks, the energy saved is high. Fig. 12 represents the average energy saving for large datasets and it is observed that higher the number of task, energy saving is more in case of energy based model.

Fig. 13 and Fig. 14 shows the performance of time based model. This model performs better for both small and large datasets when compared with the energy based model in case of minimizing the cycle time. Average reduction in cycle time for the time based model is found to be 73 units for small datasets and for large datasets average cycle time reduction is found to be 109 units.

The objective of this work is to propose models with the dual focus on time and energy. These two factors are very important in a manufacturing setup to increase productivity and also to reduce the energy consumption. Depending upon the priority of the management, the primary focus between time and energy could vary at different time horizon. The appropriate model could be selected based on the priority. These comparison results could be used for important managerial implications in real life assembly line systems.

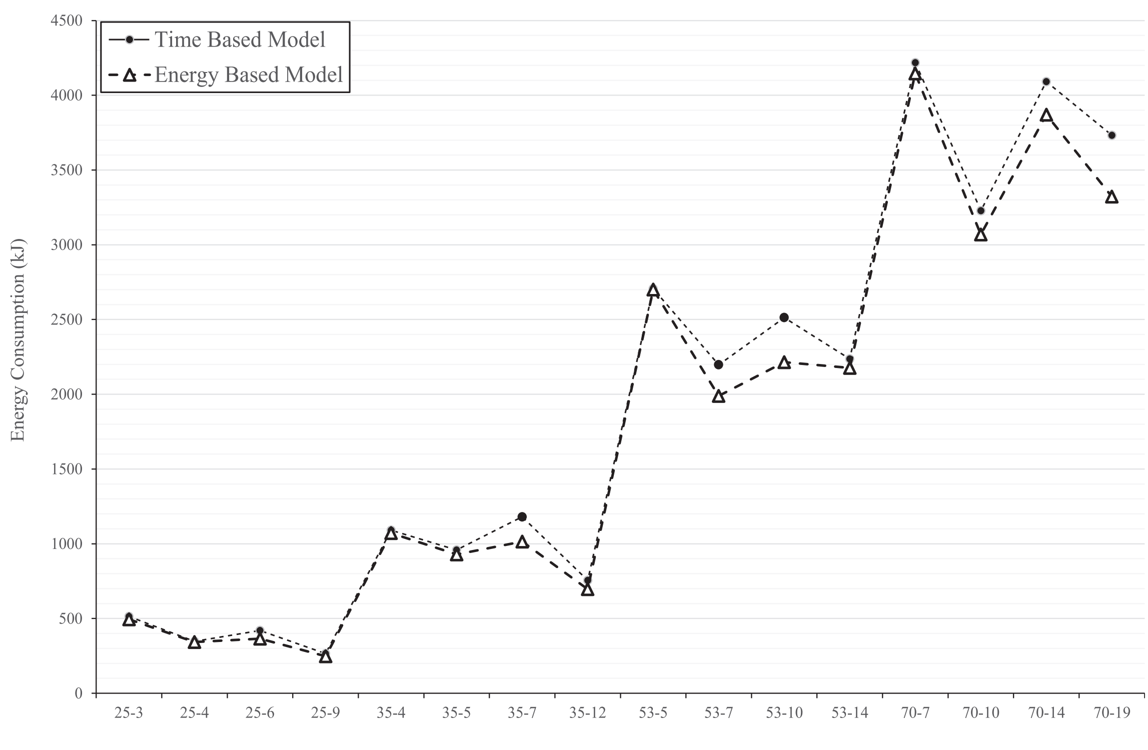

Fig. 9. Energy consumption comparison in small sized datasets between two models.

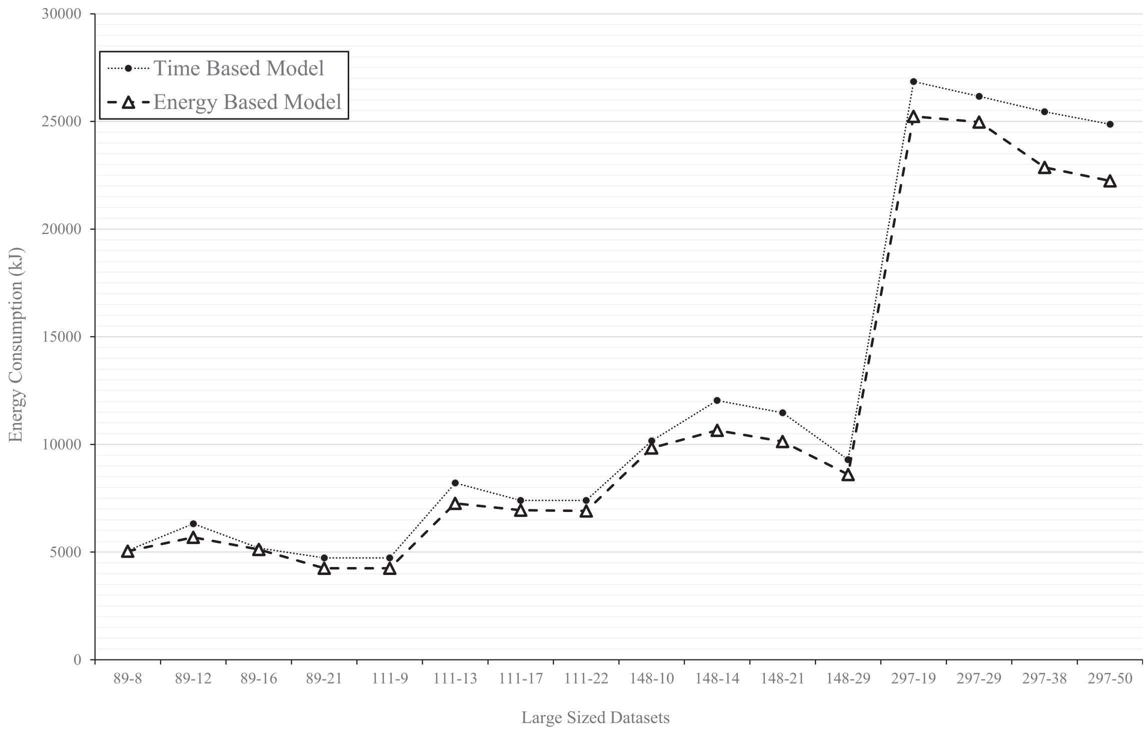

Fig. 10. Energy consumption comparison in large sized datasets between two models.

4.3. *Computation time*

The computation time for the time based model and energy based model for the 32 problems considered is presented in Table 9. The quality of solution is given importance compared to the computation time. Table 9 presents the average computation time of both models for each problem. And it could be noticed that time taken for computing the time based model is very less compared to that of the energy based model.

5. Conclusions

Optimizing cycle time and energy consumption simultaneously is an important problem in manufacturing systems. Reducing

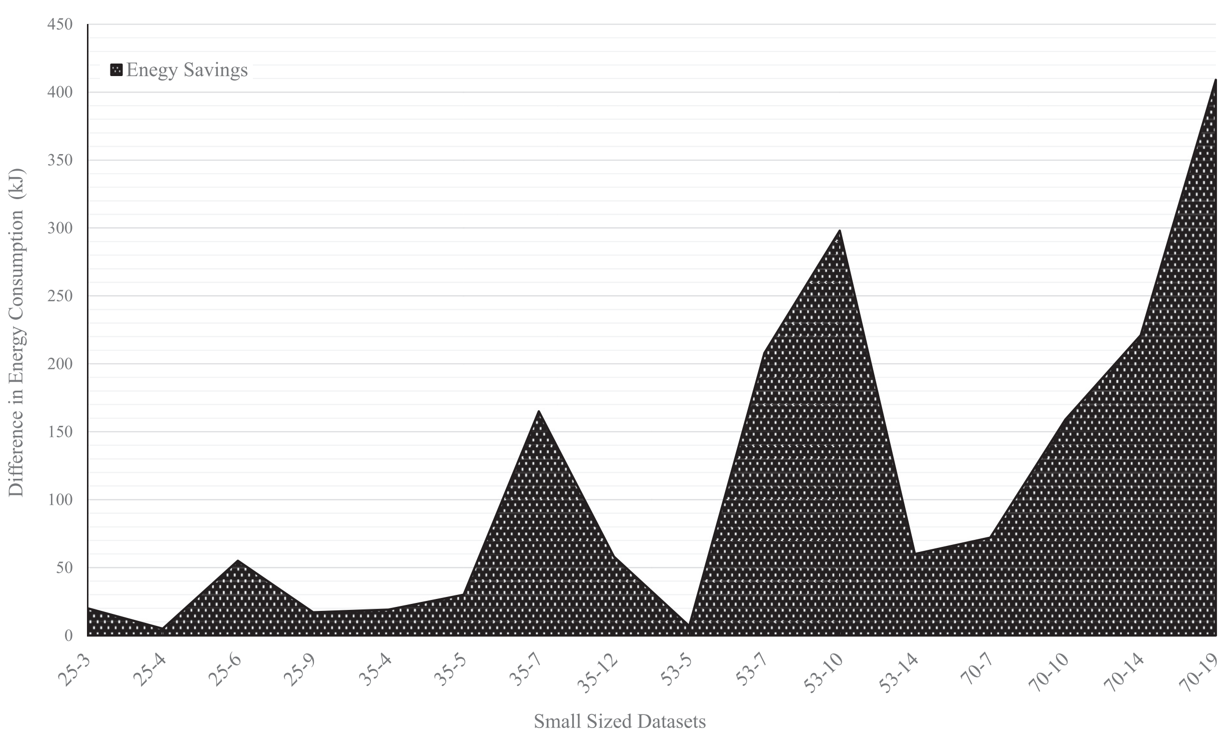

Fig. 11. Energy saving potential in small datasets for energy based model.

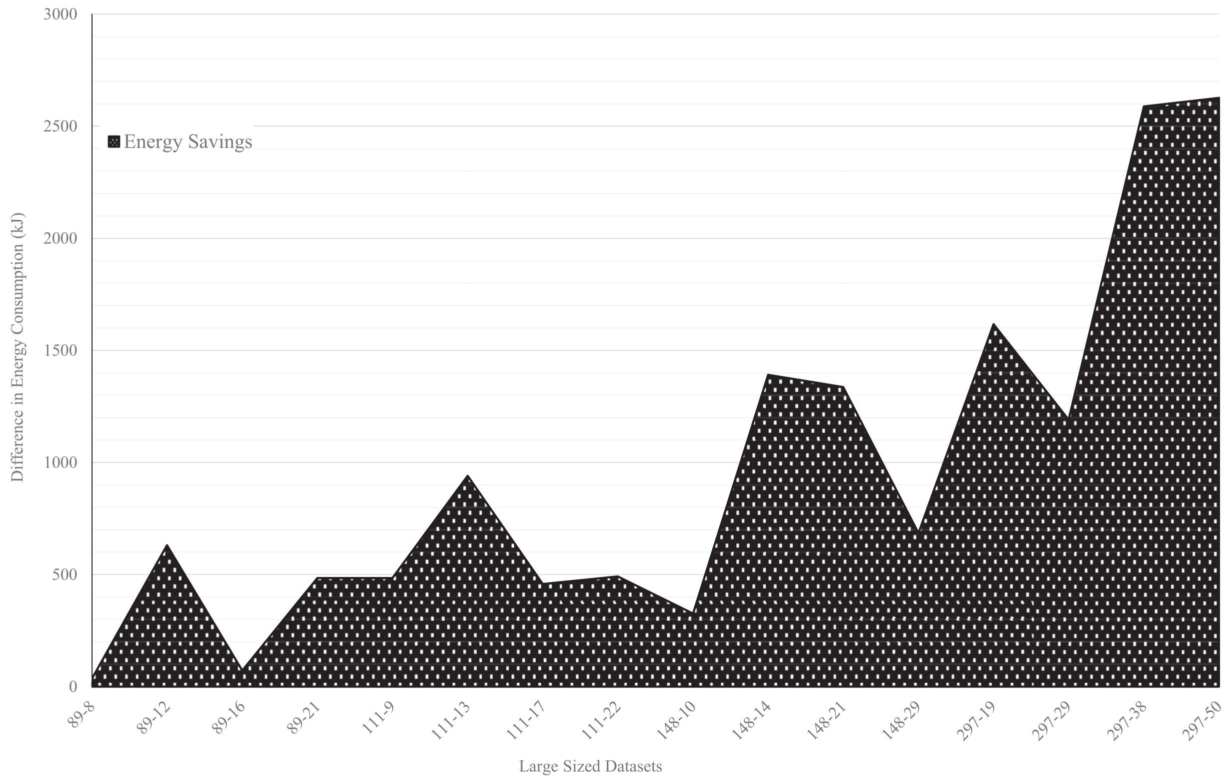

Fig. 12. Energy saving potential in large datasets for energy based model.

energy consumption in addition to increase in productivity is given importance in manufacturing sectors due to sequence of serious environmental impacts and rising energy cost. Creating an eco-friendly manufacturing system by minimizing energy consumption is very important in present day context. In this paper, a study on robotic assembly line balancing problem with an objective of minimizing cycle time and energy consumption simultaneously is considered. This paper is an important addition

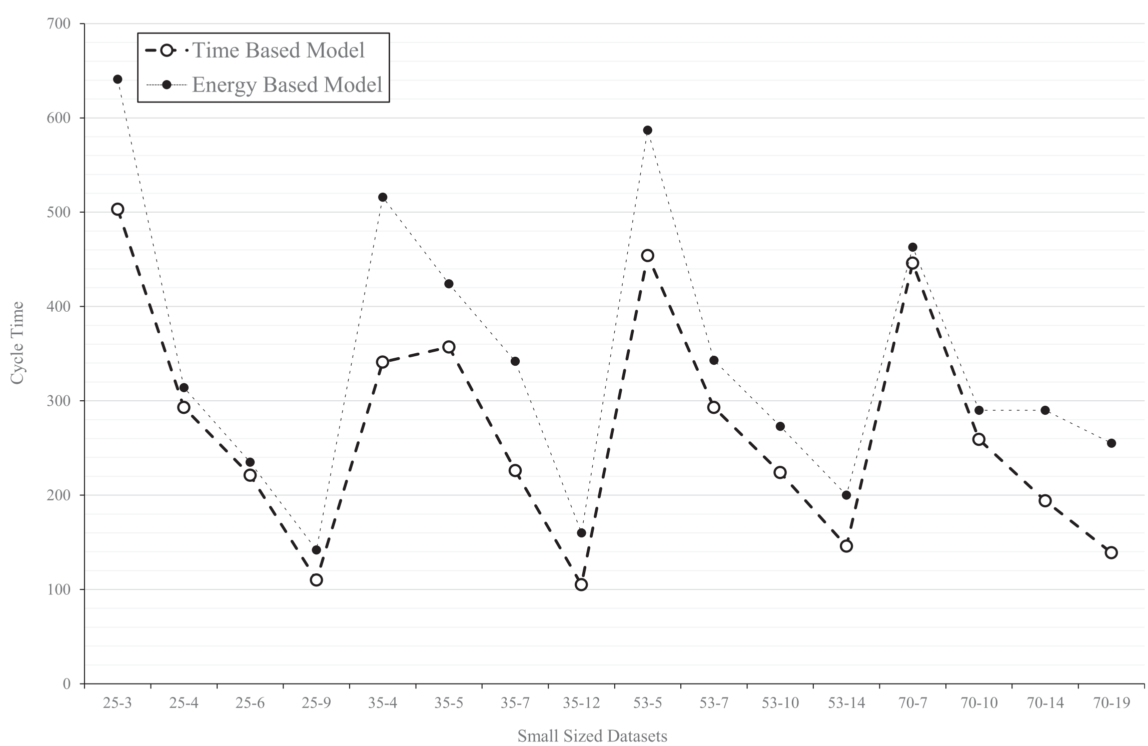

Fig. 13. Cycle Time comparison in small sized datasets between two models.

Please cite this article in press as: Mukund Nilakantan, J., et al., An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems, Journal of Cleaner Production (2014), http://dx.doi.org/10.1016/j.jclepro.2014.11.041

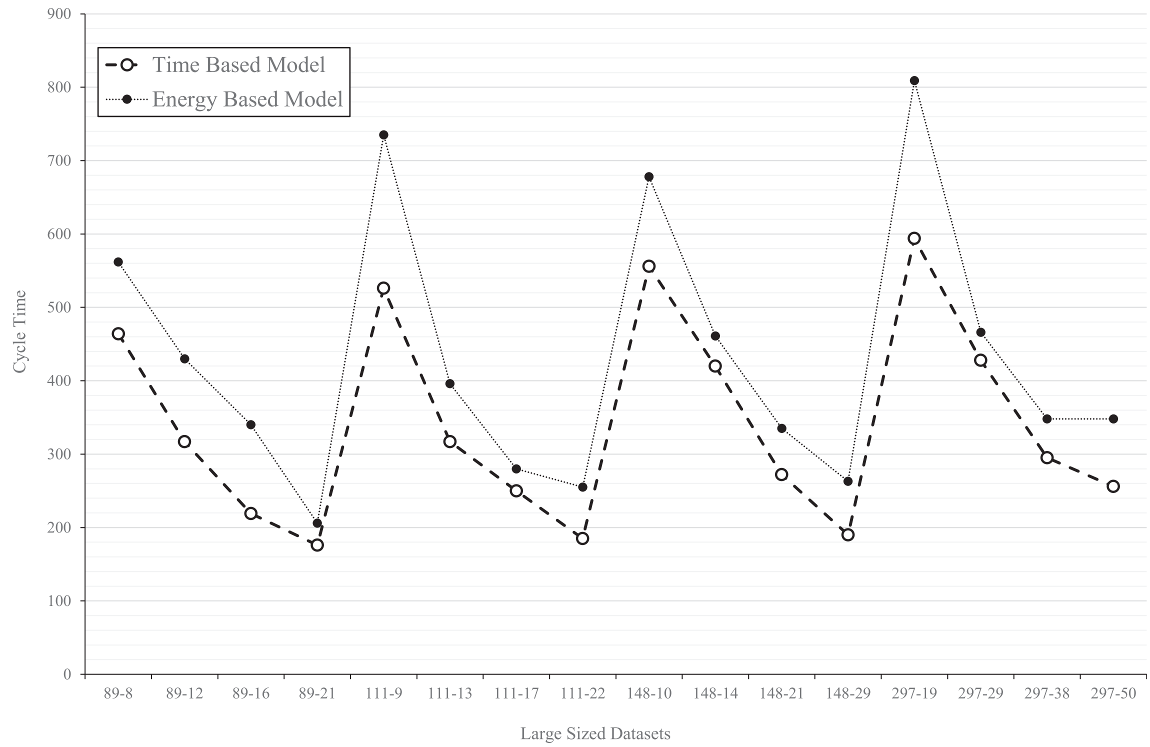

Fig. 14. Cycle Time comparison in large sized datasets between two models.

to the literature where the majority of robotic assembly line balancing researches till date focused on objectives of minimizing cycle time and minimizing cost while energy-related optimization is ignored. The research mainly aims at developing a heuristic to minimize cycle time and total energy consumption in a robotic assembly line. A particle swarm optimization algorithm is proposed to optimize cycle time and also energy consumption. Thirty-two datasets available in the literature has only time and precedence information. The energy data is embedded into the existing datasets for the experiments conducted. The computational experiments are conducted on the two models proposed in this paper. The objective of this work is to propose models with the dual focus on time and energy. These two factors are very important in a manufacturing setup to increase productivity and also to reduce the energy consumption. Depending upon the priority of the management, the primary focus between time and energy could vary at different time horizon. The appropriate model could be selected based on the priority of the management.

Table 9
Average Computation Time in seconds for RALB problems.

Problem set	Tasks	No. of problems	CPU time	
			Time based evaluation	Energy based evaluation
1	25	4	3.2	5.3
2	35	4	7.5	9.2
3	53	4	18.2	21.3
4	70	4	45.6	54.3
5	89	4	62.3	78.5
6	111	4	105.9	130.3
7	148	4	245.5	295.2
8	297	4	1245.6	1512.2

In terms of computation time, time based method performs better compared to the energy based model for evaluating the objectives. In future work, the algorithm could be tested on by considering a specific time horizon, where the factors such as maintenance operation and effect of failures of the resources in the system could be included.

Appendix A

The following notations are used in this paper:

- Indices

i, j: Index of assembly tasks, $i, j = 1,2, \ldots N_a$
h: Index of robots, $h = 1,2, \ldots N_r$
s: Index of workstation, $s = 1,2, \ldots N_w$

- Parameters

N_w: total number of workstations.
N_a: total number of tasks.
N_r: Number of robots.
H: Set of preferred/allotted robots.
C: Cycle time.
E: Energy consumption ($E = P.t_{hi}$)
sq: Sequence of tasks represents feasible solution.
t_{ih}: processing time of task i by robot h.
e_{ih}: energy consumption of task i by robot h.
$pre(i)$: Set of immediate predecessors of task i in the precedence network.
Ts: total execution time for workstation s.
Es: total energy consumption for workstation s.

Please cite this article in press as: Mukund Nilakantan, J., et al., An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems, Journal of Cleaner Production (2014), http://dx.doi.org/10.1016/j.jclepro.2014.11.041

Appendix B

Power data for small datasets (Power in kW)

Problem (number of tasks-number of robots)	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19
11-4	0.25	0.4	0.3	0.35	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
25-3	0.4	0.35	0.3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
25-4	0.25	0.4	0.3	0.3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
25-6	0.3	0.4	0.4	0.3	0.3	0.35	–	–	–	–	–	–	–	–	–	–	–	–	–
25-9	0.2	0.3	0.25	0.4	0.35	0.4	0.25	0.3	0.3	–	–	–	–	–	–	–	–	–	–
35-4	0.5	0.8	0.8	0.9	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
35-5	0.7	0.5	0.8	0.9	0.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–
35-7	0.9	0.8	0.5	0.9	0.7	0.8	0.5	–	–	–	–	–	–	–	–	–	–	–	–
35-12	0.9	0.8	0.6	0.8	0.9	0.5	0.5	0.6	0.7	0.5	0.7	0.9	–	–	–	–	–	–	–
53-5	1.1	1.2	1.5	0.9	1.3	–	–	–	–	–	–	–	–	–	–	–	–	–	–
53-7	1.5	0.9	1.2	1.1	1.2	1.3	1.5	–	–	–	–	–	–	–	–	–	–	–	–
53-10	1.1	0.9	1.2	1.5	1.3	0.9	1.2	1.4	1.1	0.9	–	–	–	–	–	–	–	–	–
53-14	0.9	1.2	1.5	1.1	1.4	0.9	1.2	0.9	1.5	1.3	1.2	1.4	1.3	0.9	–	–	–	–	–
70-7	1.4	1.4	1.7	1.5	1.3	1.5	1.4	–	–	–	–	–	–	–	–	–	–	–	–
70-10	1.2	1.6	1.1	1.7	1.2	1.5	1.4	1.8	1.6	1.7	–	–	–	–	–	–	–	–	–
70-14	1.8	1.4	1.5	1.7	1.1	1.4	1.5	1.3	1.8	1.6	1.1	1.7	1.4	1.2	–	–	–	–	–
70-19	1.5	1.8	1.1	1.7	1.3	1.6	1.6	1.2	1.8	1.3	1.7	1.6	1.3	1.1	1.3	1.1	1.1	1.5	1.7

Appendix C

Power data for large datasets (Power in kW)

Problem (number of tasks-number of robots)	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	R 21	R 22	R 23	R 24	R 25	R 26	R 27	R 28	R 29	R 30	R 31	R 32	R 33	R 34	R 35	R 36	R 37	R 38	R 39	R 40	R 41	R 42	R 43	R 44	R 45	R 46	R 47	R 48	R 49	R 50
89-8	1.1	1.8	1.5	1.2	1.9	1.4	1.2	1.6	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
89-12	1.8	1.2	1.4	1.2	1.4	1.4	1.6	1.7	1.9	1.5	1.1	1.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
89-16	1.5	1.6	1.7	1.4	1.6	1.7	1.4	1.5	1.8	1.4	1.2	1.5	1.6	1.4	1.3	1.3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
89-21	1.5	1.1	1.7	1.8	1.6	1.4	1.7	1.8	1.3	1.9	1.9	1.5	1.4	1.3	1.6	1.6	1.4	1.9	1.6	1.1	1.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
111-9	2.1	1.7	2.4	1.8	1.2	2.2	1.5	1.8	2.3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
111-13	1.8	2.1	1.5	2.2	2.3	2.4	1.9	1.7	1.9	2.2	2.1	2.3	2.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
111-17	1.7	1.9	1.6	2.2	2.1	1.8	1.6	1.5	2.1	1.6	2.3	1.9	1.8	2.4	1.7	2.2	2.1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
111-22	2.1	1.5	1.5	1.6	1.7	1.6	2.1	2.4	1.8	1.9	1.5	1.7	1.6	2.2	1.8	1.6	2.3	2.1	1.6	1.7	2.2	2.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
148-10	2.1	2.2	1.8	2.4	1.6	1.5	2	2.3	2.4	2.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
148-14	2.2	2	2.1	2.4	1.8	2.4	1.7	1.5	1.7	2.1	2.2	2	1.7	2.3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
148-21	2.4	1.5	1.6	2.4	2.4	1.5	1.9	1.6	2	2	2.1	2.2	2	2.1	1.5	1.9	2	2.1	1.8	2.1	2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
149-29	2.2	2.3	1.6	1.6	2.4	2.3	2	1.6	2.2	1.6	1.6	2.1	1.7	1.6	1.7	1.7	2.4	2.2	2.2	1.8	1.8	1.5	2.1	1.9	1.8	1.9	2.4	1.7	1.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
297-19	2.8	2.2	2.3	2.4	2.6	2.4	1.9	2	2.2	2.3	1.9	2.6	3	2.7	2.5	2.5	2.2	2.1	2.7	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
297-29	2.3	1.9	2.3	2.1	2.1	2	2.9	2.1	2.2	2.8	2.7	2.5	2.1	2.8	2.1	3	2.4	2.7	2.9	2.2	2.2	2.1	2.1	2.8	2.1	2.7	2.6	2.7	2.9	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
297-38	2.3	2.3	1.8	2	2.9	1.8	2.4	2.3	2.9	2	2.8	2	2.7	2.9	2.7	2.2	2.3	2.5	2.6	2.5	2.1	1.8	2.1	2.9	2.5	2.8	1.9	1.9	2.1	2.7	2.4	2.2	1.9	2.6	2.7	3	2.5	2.8	–	–	–	–	–	–	–	–	–	–	–	–
297-50	2.4	2.3	2.5	2.4	2.2	2	2.5	2.3	2.3	2.4	2	2.7	2.3	2.6	2.4	2.4	2.5	2.1	2.3	1.9	2	2.3	2.9	2.2	2.4	2.2	2.7	1.8	2.9	1.8	1.8	1.8	2.4	2.3	2.2	2.2	1.9	1.8	2.6	2.1	2.6	2.5	2.6	2.5	1.8	2.8	2.3	2.4	2.9	2.7

References

Bertoldi, P., Aebischer, B., Edlington, C., Hershberg, C., Lebot, B., Lin, J., Marker, T., Meier, A., Nakagami, H., Shibata, Y., 2002. Standby Power Use: How Big Is the Problem? what Policies and Technical Solutions Can Address it?.

Bukchin, J., Tzur, M., 2000. Design of flexible assembly line to minimize equipment cost. IIE Trans. 32, 585–598.

Chryssolouris, G., 2005. Manufacturing Systems: Theory and Practice. Springer.

Dai, M., Tang, D., Giret, A., Salido, M.A., Li, W.D., 2013. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robotics Computer-Integrated Manuf. 29, 418–429.

Daoud, S., Chehade, H., Yalaoui, F., Amodeo, L., 2014. Solving a robotic assembly line balancing problem using efficient hybrid methods. J. Heuristics 20, 235–259.

Das, B., Sanchez-Rivas, J.M., Garcia-Diaz, A., MacDonald, C.A., 2010. A computer simulation approach to evaluating assembly line balancing with variable operation times. J. Manuf. Technol. Manag. 21, 872–887.

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory, micro machine and human science, 1995. MHS'95. In: Proceedings of the Sixth International Symposium on. IEEE, pp. 39–43.

Fysikopoulos, A., Anagnostakis, D., Salonitis, K., Chryssolouris, G., 2012. An empirical study of the energy consumption in automotive assembly. Procedia CIRP 3, 477–482.

Gao, J., Sun, L., Wang, L., Gen, M., 2009. An efficient approach for type II robotic assembly line balancing problems. Comput. Indust. Eng. 56, 1065–1080.

Güngör, A., Gupta, S.M., 1999. Issues in environmentally conscious manufacturing and product recovery: a survey. Comput. Indust. Eng. 36, 811–853.

Gutjahr, A.L., Nemhauser, G.L., 1964. An algorithm for the line balancing problem. Manag. Sci. 11, 308–315.

He, Y., Liu, B., Zhang, X., Gao, H., Liu, X., 2012. A modeling method of task-oriented energy consumption for machining manufacturing system. J. Clean. Prod. 23, 167–174.

Kilian, L., 2008. The economic effects of energy price shocks. J. Econ. Literature 871–909.

Levitin, G., Rubinovitz, J., Shnits, B., 2006. A genetic algorithm for robotic assembly line balancing. Eur. J. Oper. Res. 168, 811–825.

Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N., 2014. An investigation into minimising total energy consumption and total weighted tardiness in job shops. J. Clean. Prod. 65, 87–96.

Luo, H., Du, B., Huang, G.Q., Chen, H., Li, X., 2013. Hybrid flow shop scheduling considering machine electricity consumption cost. Int. J. Prod. Econ. 146, 423–439.

Marinakis, Y., Marinaki, M., 2010. A hybrid genetic-particle swarm optimization algorithm for the vehicle routing problem. Expert Syst. Appl. 37, 1446–1455.

Martí, R., Reinelt, G., 2011. The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization. Springer.

Mouzon, G., Yildirim, M.B., 2008. A framework to minimise total energy consumption and total tardiness on a single machine. Int. J. Sustain. Eng. 1, 105–116.

Mouzon, G., Yildirim, M.B., Twomey, J., 2007. Operational methods for minimization of energy consumption of manufacturing equipment. Int. J. Prod. Res. 45, 4247–4271.

Mukund Nilakantan, J., Ponnambalam, S.G., 2012. An efficient PSO for type II robotic assembly line balancing problem. In: Automation Science and Engineering (CASE), 2012 IEEE International Conference on. IEEE, pp. 600–605.

Ngai, E.W., Ng, C.D., Huang, G., 2013. Energy sustainability for production design and operations. Int. J. Prod. Econ. 146, 383–385.

Ponnambalam, S.G., Aravindan, P., Naidu, G.M., 2000. A multi-objective genetic algorithm for solving assembly line balancing problem. Int. J. Adv. Manuf. Technol. 16, 341–352.

Rajamani, D., Singh, N., 1991. A simulation approach to the design of an assembly line: a case study. Int. J. Oper. Prod. Manag. 11, 66–75.

Rashid, M.F.F., Hutabarat, W., Tiwari, A., 2012. A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. Int. J. Adv. Manuf. Technol. 59, 335–349.

Rubinovitz, J., Bukchin, J., 1991. Design and balancing of robotic assembly lines. In: Proceedings of the Fourth World Conference on Robotics Research Pittsburgh, PA, SME 1991.

Rubinovitz, J., Bukchin, J., Lenz, E., 1993. RALB—A heuristic algorithm for design and balancing of robotic assembly lines. CIRP Annals-Manufacturing Technol. 42, 497–500.

Salveson, M.E., 1955. The assembly line balancing problem. J. Industrial Eng. 6, 18–25.

Sanderson, A.C., de Mello, L.S.H., Zhang, H., 1990. Assembly sequence planning. AI Mag. 11, 62.

Scholl, A., 1999. Balancing and Sequencing of Assembly Lines. Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).

Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., Ortega-Mier, M., 2013. Optimizing the production scheduling of a single machine to minimize total energy consumption costs. J. Clean. Prod. 67, 197–207.

Villarreal, B., Alanis, M., 2011. A simulation approach to improve assembly line performance. Int. J. Industrial Eng. Theory, Appl. Pract. 18, 283–290.

Yoosefelahi, A., Aminnayeri, M., Mosadegh, H., Ardakani, H.D., 2012. Type II robotic assembly line balancing problem: an evolution strategies algorithm for a multi-objective model. J. Manuf. Syst. 31, 139–151.