J Intell Manuf (2016) 27:1049–1065
DOI 10.1007/s10845-014-0934-3

Developing a multi-objective genetic optimisation approach for an operational design of a manual mixed-model assembly line with walking workers

Atiya Al-Zuheri · Lee Luong · Ke Xing

Received: 16 January 2013 / Accepted: 9 June 2014 / Published online: 24 June 2014

Abstract A walking worker assembly line (WWAL), in which each cross-trained worker travels along the line to carry out all required tasks, is an example of lean system, specifically designed to respond quickly and economically to the fluctuating nature of market demands. Because of the complexity of WWAL design problems, classical heuristic approaches are not capable of solving problematic design characteristic of WWAL of very large design space. This paper presents a new genetic approach to address the mixed model walking worker manual assembly line optimisation design problem with multiple objectives. The aim is to select a set of operational variables to perform to the required demand for two product models. The goal is to produce the required models at the lowest cost possible, whilst keeping within an ergonomically balanced operation. Genetic algorithms are developed to tackle this problem. This paper describes the fundamental structure of this approach, as well as the influence of the crossover probability, the mutation probability and the size of the population on the performance of the genetic algorithm. The paper also presents an application of a developed algorithm to the operational design problem of plastic electrical box assembly line.

Keywords Manual assembly line · Walking workers · Operational design · Multiple objectives · Genetic algorithms

A. Al-Zuheri (✉) · L. Luong · K. Xing
School of Engineering, University of South Australia, Mawson Lakes, SA 5095, Australia
e-mail: alyaa002@mymail.unisa.edu.au

A. Al-Zuheri
Ministry of Science and Technology, Al-Jadriya, 10070 Baghdad, Iraq

Introduction

The flexibility of assembly systems is essential in the assembly industry in order to be able to respond to the changeable characteristics of market demands. These demands, represented by increasing customisation, the shortening of a product lifecycle, and a high product variety produced in small batches (Shishir 2010). For this reason, it has become necessary to develop dynamic, flexible and reconfigurable assembly systems. The flexible labour line (or flexible assembly line), is one of the promising techniques to create the most effective and productive assembly systems to respond to the challenges in manufacturing environments (Moslemipour and Lee 2012). On the other hand, mixed-model assembly lines enable manufacturers to assemble multiple models in any order within a short delivery time and with the lowest possible cost. As a consequence, considering flexible workforce in designing a mixed-model assembly line is an innovative solution.

One attractive form of flexible workforce in mixed model assembly lines is the use of cross-trained workers who can perform all the assembly processes of products and also shifting their capacity to where it is needed (Sawhney 2012). This paper explores one such system that of staffing a line of assembly processes for mixed models of a product, with fewer workers than workstations on the line. This dynamic, flexible and reconfigurable system is so-called mixed model walking worker assembly line. In this type of assembly line, each worker travels along the line to carry out all required tasks (Wang et al. 2005). Walking worker assembly line (WWAL) results in a series advantages over a traditional lines, that is, fixed worker assembly lines (FWAL), where traditionally each worker has an assigned fixed task and continuously repeats that assigned task (see Fig. 1). In this context, rearranging assembly lines from the FWAL to the WWAL

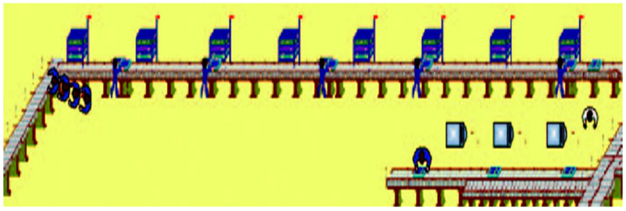

Fig. 1 Layout of walking worker line (Wang et al. 2005)

by a number of companies has led to achieve the followings (Wang et al. 2005):

- Greater ease of line balancing whilst contributing to reducing the number of buffers required;
- An adjustable number of line workers according to demand requirement; and
- A minimisation of labour and tooling costs.

Certainly, design of such assembly lines has significantly impacted workforce performance and consequently is more complex in comparison with single fixed worker assembly lines. This is because there is an essential need to consider human worker capabilities and their limitations in operational designs for these systems (Alzuheri et al. 2010). Also, due to the sources of complexities in manual assembly systems such as processing times at each workstation and skills of workers who perform these tasks, most of these systems are stochastic rather than deterministic (Longo et al. 2012). Considering the nature of the system thus, physical workers capabilities as well as the "multiple components" of WWAL can lead to large design space solutions and it is more difficult (if not impossible) to use exact mathematical methods such as integer-programming to solve problem design of such system.

Therefore, this difficulty requires the use of meta-heuristic approaches to solve the problem. The resulting design solutions are adjusted to create an acceptable design and the best resulting solution is implemented. Consequently, the design process must be regarded as an iterative, generative and test-involving process (Scholl et al. 2010).

For this purpose genetic algorithms (GAs) are proposed to solve the WWAL optimisation design problem, aiming at yielding optimal (or near-optimal) solution in large search spaces quite faster than any other optimisation technique owing to their parallel searching feature (Chambers 2010). Currently, GAs required for such a demonstration does not exist in literature on the topic. This paper describes development of a novel customised GAs that are used for solving the large scale optimisation problem that is typically can found in manual assembly line with walking workers.

In the next section of this paper, a literature review is conducted for research work that considers WWAL. Section "Design problem: mixed model assembly system optimisation problem", introduces the design problem that this paper attempts to solve. Section "The mathematical model" includes a summarised background about the mathematical model, which drives GAs comprising the objective functions. In "Genetic algorithms" section, GAs are reviewed as the approach for optimisation selected to handle the design problem presented in this paper, and there follows a description of the developed structure for this approach. Then the results of the GAs application are presented and analysed in "The results and analysis of design optimisation problem" section. Finally, in "Conclusions" section, the conclusion of this paper is presented.

Literature review

In last 15 years, research has investigated the subject of cross-trained walking (moving) workers performance in production systems. Mileham et al. (2000) and Nakade and Nishiwaki (2008) give a summary of this research. However, all this research was limited to the application of moving cross-trained workers in a cell in linear or U-shaped production line in the industry. In addition, most of the research referred to systems including such applications by names other than WWAL. The term WWAL is a recent concept (Bley et al. 2007; Mileham et al. 2008). Publications have focussed little on the subject of this type of assembly line, and even then, it has only been on the initial stages of the design phase. Table 1 summarises the number of research papers published by topic.

As the results of Table 1 show, most of the research carried out using both simulation technique and integer-programming approach for designing WWAL, so far has considered the single model type at early stage of the design phase. As a result, the design process was relatively simple due to the considerably large amount of extant research examining solution techniques offered on the problem structure of FWAL, which is also applicable in solving simple WWAL problems. Although WWAL is an important consideration in assembling different models as noted previously, there has been no research reported on the operational aspects of mixed model assembly line design with walking workers.

Since cross-trained workers have become a necessary requirement of WWAL, effective operational design is critical for enhancing productivity and maximising the usefulness of a given assembly system by optimising the utilisation of workers effectively. This improvement in system performance should be concerned with investigating the ergonomics consequences of any improvement processes, in terms of exposure to risk factors for work-related musculoskeletal injuries. This results in a multi-objective design problem.

Table 1 Summary of the research papers conducted on WWAL

The paper	Modelling descriptions	Issues addressed in study
Wang et al. (2005)	Witness, simulation package	The better of WWAL over fixed workers FWAL—in terms of line performance measures
Wang et al. (2007a)	Witness, simulation package	Influencing in line performance with varying the numbers of workers and workstations in WWAL
Wang et al. (2007b)	Witness, simulation package and Statistical Distribution Functions (SDFs) used for analyses collected data	Modelling and simulation of a linear walking worker assembly line considered being a random system due to the randomness of input parameters that can have a significant impact on performance of the system
Lassalle et al. (2007)	Witness, simulation software and an external user-friendly interface for its input and output data which were managed and controlled by a series of MS Excel™ spread sheets	Examined of the variable behaviour in-process waiting time that takes place on a linear WWAL based on two assumptions related workers performance: • Same performance during a period of production • Variable performance levels during a period of production
Mileham et al. (2008)	Mathematical modelling	Summarised the advantages and shortcomings of application of WWAL on short section on a semi-automated automotive engine assembly line
Wang et al. (2009)	A combination of computer simulation and mathematical analysis	Evaluated in-progress waiting time due to bottlenecks (e.g. machine with the longest processing time or a walking worker with variable performance) that affects the overall system performance

However, as noted from the review of literature, none of the researchers focused on operational design problems of mixed model lines adopted from the WWAL approach. Also, because of the nature of WWAL, and because the design problem is a multi-objective one, it is expected to be a large-scale one. This paper explores one such problem that of optimising operational design for mixed model assembly lines with walking workers in an industrial environment for made-to-order assemblages with some specific objectives, constrains and considerations about productivity and ergonomics.

Although there are many alternative heuristic methods for solving multiple objective design problems, there is still no clear guidance available regarding the optimal choice for a particular problem. GAs seem particularly desirable to solve multi-objective optimisation problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows to find an entire set of Pareto-optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. On the other hand, GAs have been applied successfully in a wide variety of optimisation problems (Li et al. 2007). Therefore, developed GAs has been used to solve the problem presented in this paper. At the same time, a mathematical model is proposed for driving developed GAs. A detailed description of this model can be seen in Al-Zuheri et al. (2013). Thus, this paper focused mainly on the development of genetic algorithms to optimise multi-objective functions of performance measures in terms of productivity and ergonomics of mixed model assembly line using WWAL approach for assembling a plastic electrical box.

Design problem: mixed model assembly system optimisation problem

Let's assume, hypothetically speaking, an assembly cycle of a plastic electrical box produced in a design and manufacturing company. Figure 2 shows the plastic box assembly process and the models. The product is assembled on a flow line with seven workstations.

The line is used to assemble two different model types. To assemble model A requires processing on workstations (1, 3, 4, 6, 7) and model B requires processing on workstations (1, 2, 4, 5, 6, 7). Therefore, the models are nearly identical except that with model B process 2 is substituted for process 3 in model A. In addition, model B requires process 5 while model A does not. The processes must be run and completed in the exact sequence given for each model,

Table 2 gives average working time, and average expenditure for the metabolic energy requirement schedules of the two model types needed to complete the assembly tasks. The average working time are determined through method time measurement (MTM) and hence require re-calculation

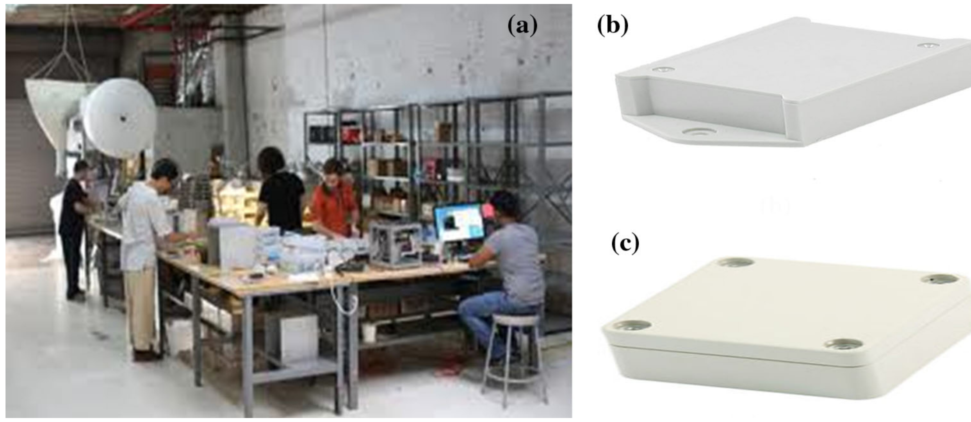

Fig. 2 Illustration of plastic electrical box assembly process; **a** the assembly line, **b** model A, **c** model B

Table 2 Task descriptions, average working time, required metabolic energy expenditure and the weight of the each model of product at each workstation on the line

Workstations	The task	Model type A			Model type B		
		Average working time (s)	Metabolic energy expenditure (kcal)	Weight of product after process (kg)	Average working time (s)	Metabolic energy expenditure (kcal)	Weight of product after process (kg)
$k1$	Preparing	40	1.581	3.67	40	1.581	4.57
$k2$	Trimming	–	–	–	30	1.511	4.30
$k3$	Fastening	55	1.811	5.22	–	–	–
$k4$	Inserting	86	1.632	8.52	86	1.632	4.82
$k5$	Assembling	–	–	–	70	1.293	6.52
$k6$	Testing	75	1.46	8.52	75	1.46	6.52
$k7$	Packing	44	1.163	9.65	44	1.163	7.34
		Total working time = 300 s	Total energy expenditure = 7.637 Kcal	Final weight of product = 9.65 kg	Total working time = 345 s	Total energy expenditure = 8.64 Kcal	Final weight of product = 7.34 kg

according to the variable requirements in the model proposed in Al-Zuheri et al. (2013).

Also Table 2 lists the product model data in terms of task descriptions and the weight of the product of each model at each workstation on the line.

Let's assume, that on this particular day the "make to order" WWAL is asked to assemble of 80 and 100 products from model A and model B, respectively. In this design problem, the assumptions that the available production time per that day for assembly of both models is one eight hour shift per day (480 min). Each model gets exactly half of this available time, then each model will have 90–240 min of assembly time to produce the required rate. The available production time is split between the two models. The required production demand from model A is assembled first and then production demand of model B. The workstations are located to ensure an efficient layout, and the most efficient assembly process arrangement maximises productivity of workers.

The design problem is to determine which number of workers and their skill levels (low and high) to select and which walking speed of workers (slow or fast) to assign so as to meet production demand requirements for the two product models as well as optimising the system operational design as mentioned in the objectives in "Design problem: mixed model assembly system optimisation problem" section. Also, the design problem includes determining the best available assembly time for both models to match the set design objectives.

Rather than the specific data that are given in Table 2, the parameters related to structure of the system and for workers gender, weight, and their cost, according skill levels are shown in Table 3, while the operational design variables that will affect the objective design functions are:

Variable x_1: skill level of slow walking workers;
Variable x_2: number of slow walking workers;
Variable x_3: walking speed of workers;

Table 3 Input parameters of the WWAL and their levels

Parameters	Notation	Values	Unit	
Average weight of workers	BW	82	kg	
Number of workers on the line	M	5	–	
Gender of workers	G	Male	–	
Age of workers	–	25–45	Years	
Variance of performing tasks	V_K	0.15	–	
Penalty function of fatigue	ε	0.97	–	
Factor of guaranteeing tasks	α	1.64	–	
The distance between workstation and other	DT	4	m	
Cost of worker per hour working	C_{mj}	Worker skill level	Cost	\$/h
		100 %	25	
		90 %	23	
		80 %	21	
		70 %	20	
Grade of the floor surface	g	0.9	–	

Variable x_4: the available assembling time for model A; and
Variable x_5: the available assembling time for model B.

These five critical variables are set to a different level as shown in Table 4. Based on these settings, there are a total of hundred thousands of different combinations and for each combination the model has a fitness function value is also equal to performance measure (objective function value).

The mathematical model

The model completes two objective functions for each of productivity and ergonomics. The four objective functions as follow:

- *Minimise the balance of labour blockage,*
- *Minimise the shift time labour cost,*
- *Minimise the metabolic energy consumption, and*
- *Maximise the rate of exposure variability*

A detailed formulation of these objective functions with assumptions, input parameters, variables and constraints can be seen in Al-Zuheri et al. (2013).

In conclusion, following the end-user equations that are used to describe these objective functions.

- *Minimise the balance of labour blockage:*

The balance of labour blockage η, is the ratio of in-process waiting time divided by the overall cycle time:

$$f_1(x) = \eta \tag{1}$$

$$Min.f_1(x) = Min\left(\frac{t_{Im}^{k_b}}{T_{ca}(n)}\right) \tag{2}$$

Subject to:

$$P_{Shift} = \sum_{m\in M} P_m \geq \left(\frac{T_{Shift}}{t_{Im}^{k_b}}\right) \tag{3}$$

This constraint ensures that every candidate solution has production rate equal or greater than to that with traditional assembly line FWAL.

- *Minimise the shift time labour cost:*

The cost of shift time labour requires calculation for each design of WWAL. The cost considered is direct labour cost. Given the number of workers, their efficiencies, their cost per hour and the shift time hours, the labour cost in terms of these variables are:

$$f_2(x) = C_{TI} = \sum_{m\in M} (Cost\ of\ worker\ per\ hour\ shift\ time \times Shift\ time\ hours) \tag{4}$$

Mathematically, the objective function is to minimise the total shift time labour cost:

$$Min.f_2(x) = Min.(C_{TI}) = Min.\left(\sum_{m\in M}\sum_{j\in J} C_{mj} \times T_{Shift}\right) \tag{5}$$

Subject to:

$$C_{TI} \leq C_{Tl}FW \tag{6}$$

where C_{mj} is the cost of worker per shift time hour. $C_{Tl}FW$ is the shift time labour cost for traditional assembly line-FWAL, and normally in this line the number of workers equal the number of workstations and the cost of shift time hour is low.

Table 4 Operational design variables and their levels for the WWAL design

Design variables	Notation	Level/code				Number of levels	Units
		1	2	3	4		
Variable x_1—skill level of slow walking workers	γ	70 %	80 %	90 %	–	3	–
Variable x_2—number of slower walking workers	S	2	3	4	5	4	–
Variable x_3—walking speed of workers	v_o	0.7	1.4	–	–	2	m/s
Variable x_4—available assembling time for model A	T_A	90	240	–	–	151	min
Variable x_5—the available assembling time for model B	T_B	90	240	–	–	151	min

- *Minimise the metabolic energy consumption:*

Since in WWAL, the worker needs to work standing up rather than sitting down also, he or she moves along with the workpiece. Mathematically then, the Garg's model (Garg et al. 1978) for the prediction of average metabolic energy expenditure requirement to perform the entire assembly tasks in one cycle time $E_m(n)$ in WWAL is as follows:

$$f_3(x) = E_m(n) = Min.f_3(x) = Min.E_m(n) \tag{7}$$

$$Min.E_m(n) = \frac{\left[(E_{mp} \times T_c(n)) + \left(\sum_{k\in K} \Delta E^k_{Om}\right) + Min.\left(\sum_{k\in K} \Delta E^k_{Wm}\right) + Min.(\Delta E^{k_b}_{Im})\right]}{T_{ca}(n)} \tag{8}$$

Subject to:

$$E_m(n) \le 3.2\frac{Kcal}{min.} \tag{9}$$

This constraint ensures that in addition to the energy cost of walking or in-process waiting to work, the metabolic energy requirement to assemble one product by the worker in each candidate solution should not exceed the limitation value of energy consumption 3.2 kcal/min. in a working day (8 h) (Waters et al. 2011; Zhu et al. 2010).

- *Maximise rate of exposure variability*

This measure is the ratio of overall walking time between workstations to the amount of time worker spends to perform all tasks of assembly job during the work cycle with standing position.

$$f_4(x) = \varphi(n) \tag{10}$$

$$Max.f_4(x) = Max.(\varphi(n)) = \frac{W_c(n)}{S_C(n) + t^{k_b}_{Im}} \tag{11}$$

Subject to:

$$L_w = \frac{W_C(n) \times n \times v_o}{1000} < 4.32km \tag{12}$$

The OSHA regulations (2009) related to the amount of maximum walking distance for assembly line worker cannot have more than 4.32 km (2.7 miles) at the shift time.

Genetic algorithms

GAs are meta-heuristic stochastic approach for finding the global optimal solution for a combinatorial optimisation problem (Liu et al. 2011). They mimic the mechanism of genetics and natural selection (survival of the fittest) natural genetics as described by Charles Darwin (Li et al. 2007). Holland was introduced GAs in the 1960s and 1970s as an approach to evolve an optimal solution from a population of initial feasible solutions available for solving an optimisation problem (Goldberg 2012). The GAs search for optimal or near-optimal solutions starts with the generation of a number of individuals for the initial random population of solutions. Each individual in the population is a chromosome representing a solution of the problem. The chromosome is a string of symbols or (set of genes) can be coded in different forms; binary, integer, real, etc. With taking into account specified selection rules, the initial population evolves towards a population expected to include the optimal solution, eventually accomplished through successive iterations called generations.

Within a generation, GAs undertake to select subsets (usually two) of chromosomes from the current population called parents for mating them to produce new chromosome called children (or offspring). Chromosomes are selected according to their fitness relative to the current population. Based on that, the chromosomes or solutions that are of higher fitness then, have a greater chance of selection and then "mating" with another high-fitness chromosome for producing new chromosomes. Selected chromosomes are subjected to rules of combination to yield children; genetic operators—crossover and mutation. The crossover consists of the exchange of parental genes through passing it on to the new chromosome. Through applying mutation, further genetic diversity is introduced into the chromosomes of a population, and by further mating between parents; a set of new generation chromosomes is formed. After several of generations, the algorithm tends to the optimal chromosome, which globally, has a better fitness in comparison with other chromosomes.

Design of the GAs structure for the optimisation of the WWAL design problem

GAs are responsible for inserting the changes of values for design variables into the mathematical model. For reaching optimal design of WWAL, the approach works by iteration, and every result given by the model corresponds a design solution of WWAL. The chromosome in solution spaces of WWAL includes a number of genes (design variables), that describe operational aspect of the system. These variables are as listed in "Design problem: mixed model assembly system optimisation problem" section. The WWAL design corresponding to each chromosome is characterised by its fitness, which is measured by its resultant objective function value (will be derived later in this section).

Herein, a generation consisting of evolving chromosomes of the previous population and new chromosomes are made through reproduction process by means of crossover, mutation, and selection of their parent's chromosomes. Figure 3 shows optimisation procedure using GAs and mathematical model. Details are presented on the designed GAs components in the forthcoming subsections of this paper.

Chromosome representation

For the problem under consideration in this paper, real coded value variables are adopted to present chromosomes. Using real coded variables makes it possible to use large domains (even unknown domains) for variables. There is also a chance to increase the efficiency of GAs by exploiting the incremental nature of the functions with continuous variables. Herein, each chromosome (solution) of WWAL design comprises a vector of real design variables, and it is very close to the real structure of the practical problem. In this regard, considering the alternative design solution is a, let n the number of design variables that are allocated to design solution, the whole chromosome code which represents all design variables, is shown by the following equation:

$$x_a = [x_a(1), x_a(2), \ldots, x_a(n)], \qquad a = 1, \ldots\ldots A \quad (13)$$

where A is the number of available feasible solutions to the problem. The design variables $x_n (n = 1, 2, 3, \ldots, n)$ are operational variables. Also these variables is not unconstrained and must be within a level from x_{min} to x_{max}.

Generation of the initial population

Initialisation of the population of chromosomes (set of possible solutions to the problem) may be randomly created, or created according to heuristic procedures (Zhan and Zhang 2013). Yu and Yin (2010) have been indicated that the performance of the GAs algorithms is not as good from well-adapted (seeded) population as it is from a random start. Based on that, in this paper the chromosomes in an initial population are randomly generated ones. The number of chromosomes within the population is called population size Pz.

Fitness evaluation of the chromosomes in the population

Once the initial population is developed, fitness evaluation for each chromosome is performed. Fitness function is the key performance index in GAs application and can determine which of the chromosomes will reproduce and survive into the next generation. The better performing chromosomes are selected as candidates for evolving them using the genetic operations. In general, derivation of fitness function is through the objective functions, and is used in successive genetic operations (Garg 2010). Since the objectives $f_1(x)$, $f_2(x)$ and $f_3(x)$ must be minimised and $f_4(x)$ must be maximised (as stated in "Design problem: mixed model assembly system optimisation problem" section), it is then required to convert these multiple objectives into a single overall objective function, called a resultant objective function. The resultant objective function is considered as a fitness function requiring final optimisation. The desirability function approach (Derringer and Suich 1980) is used to achieve the converting from multiple objectives to single objective in two stages, as described below.

Stage 1—defining individual desirability functions for objective functions: as a result from computational experiments, assuming that there are a different alternative solution for the candidate problem, each objective (productivity or ergonomics) have desirability at a solution. Accordingly η_a, C_a, E_a and φ_a denote, respectively the $\eta(n)$, C_{TI}, $E_m(n)$ and $\varphi(n)$ objectives values for solutions $(a = 1, \ldots, A)$. In this paper, the target is for three objectives to be minimised and one objective to be maximised

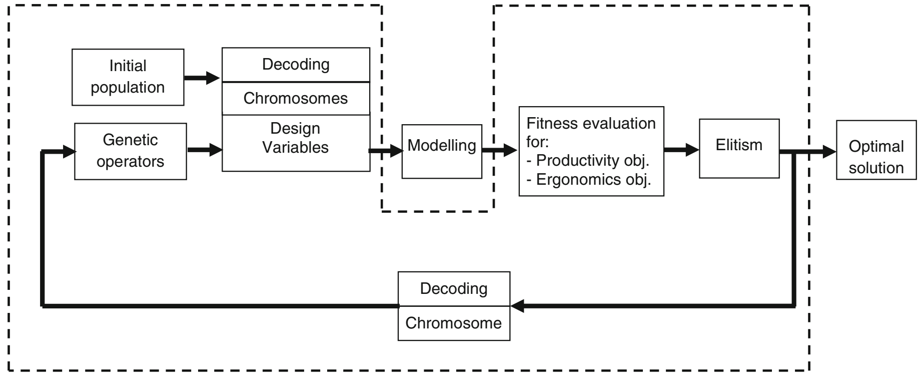

Fig. 3 A schematic diagram of the integrated GAs and the mathematical for conducting the design problem

(for the one-sided case), as in the case when the objective is productivity measure (balance of labour blockage), the individual desirability d_a for η_a is defined as follows:

$$d_a(\eta_a) = \begin{cases} 1, & \eta_a \leq L_n \\ \left(\frac{\eta_a - L_\eta}{U_\eta - \eta_a}\right)^{W1} & L_\eta \leq \eta_a \leq U_\eta \quad a = 1, \ldots\ldots, A \\ 0, & \eta_a \geq U_n \end{cases} \tag{14}$$

For second productivity objective (shift time labour cost), C_a:

$$d_a(C_a) = \begin{cases} 1, & C_a \leq L_C \\ \left(\frac{C_a - L_C}{U_C - C_a}\right)^{W2} & L_C \leq C_a \leq U_C \quad a = 1, \ldots\ldots, A \\ 0, & C_a \geq U_C \end{cases} \tag{15}$$

For ergonomics objective (average metabolic energy expenditure), E_a:

$$d_a(E_a) = \begin{cases} 1, & E_a \leq L_E \\ \left(\frac{E_a - L_E}{U_E - E_a}\right)^{W3} & L_E \leq E_a \leq U_E \quad a = 1, \ldots\ldots, A \\ 0, & E_a \geq U_E \end{cases} \tag{16}$$

For second ergonomic objective (rate of exposure variability),φ_a:

$$d_a(\varphi_a) = \begin{cases} 1, & \varphi_a \leq L_\varphi \\ \left(\frac{\varphi_a - L_\varphi}{U_\varphi - \varphi_a}\right)^{W4} & L_\varphi \leq \varphi_a \leq U_\varphi \quad a = 1, \ldots\ldots, A \\ 0, & \varphi_a \geq U_\varphi \end{cases} \tag{17}$$

where $U_\eta(L_\eta),\ U_C(L_C),\ U_E(L_E)$ and $U_\varphi(L_\varphi)$ are the upper (lower) limits of the four objective functions, respectively. The $w1,\ w2, w3$ and $w4$ indicates the weight. In the design problem of this paper, the assigned weights imply that balance of labour blockage η_a and average metabolic energy expenditure E_a are twice as important compared with other objectives important C_a and φ_a, the shift time labour cost and rate of exposure variability.

Stage 2—calculation of the overall desirability: an overall desirability function D_a can be obtained by using the geometric mean of the individual desirability d_a for each measure. It reflects the composite desirable grade of the a_{th} alternative solution desirability with respect to the objective functions.

$$D_a = \left(\prod_{v=1}^{v} d_{a,v}^{r_v}\right)^{1/\sum r_v} \quad a = 1, \ldots\ldots.A \tag{18}$$

where: $d_{a,v}$ = the v_{th} objective function of alternative solution (a); and

r_v = the relative importance that is assigned subjectively and, respectively to each objective function. In problem design of this paper, $v = 4$ and for more simple form of the equation, the expressions for the individual desirability of each objective function : $d_a\left(\eta_a\right) = \hat{\eta}_a$, $d_a\left(C_a\right) = \hat{C}_a$, $d_a\left(E_a\right) = \hat{E}_a$, and $d_a\left(\varphi_a\right) = \hat{\varphi}_a$. Accordingly, the overall desirability functions as in the following:

$$D_a = \left(\hat{\eta}_a \times \hat{C}_a \times \hat{E}_a \times \hat{\varphi}_a\right)^{1/4} \quad a = 1, \ldots\ldots.A \tag{19}$$

In GAs of this paper, the resultant objective function (the overall (or total) desirability, D_a given in formula 19, is adapted as a fitness function. Then, the goal of WWAL design optimisation problem is to find the levels of design variables that maximise the overall desirability D_a, in the design level of interest, that is;

$$Max.(D_a) = Max.\left[\left(\hat{\eta}_a \times \hat{C}_a \times \hat{E}_a \times \hat{\varphi}_a\right)^{1/4}\right] \quad a = 1, \ldots\ldots.A \tag{20}$$

Subject to:

$$x_{min} \leq x \leq x_{max} \tag{21}$$

where, formula 21 can be stated as the primary or initial design levels. x_{min}, x_{max} are the minimum and maximum design levels of operational variables, respectively. This constraint of optimising design of WWAL is the last constraints to be met in a set of feasible solutions before consider one of them as superior choice that meets the previous four objectives.

Selection strategy

Since the best chromosomes (solutions) in the current generation are used to reproduce the next generation, this requires establishing a strategy to select those chromosomes. When implementing GAs, there are two selection strategies in choosing the chromosomes for reproduction of new chromosomes; elitist and non-elitist. Elitist strategy is used herein to avoid losing best solutions in population by saving a number of them and later copying it into the next generation for the process of evolution.

Genetic operators

Several types of operators are known currently. Please note however, that these operators are beyond the scope of this paper. Following is a brief background about the rule of each operator in evolving process of GAs, along with the types of operators selected for this paper.

Selection process: Darwin's evolutionary theory of the survival of the fittest is applied in the parent selection process for chromosomes in GAs applications. Based on this law, chromosomes with high fitness function values are selected for genetic operations using stochastic selection process (Goldberg 2012). Many types of selection techniques are suggested with which to select the best chromosomes to be copied into the next generation (Martinec and Bundzel 2013). The biased roulette wheel technique is the most popular from these techniques. This paper applies the roulette wheel technique because it is easy to implement and mimics nature more faithfully and consequently is much more appealing (Tang and Tseng 2012).

Crossover: crossover or recombination is the process most widely used by which a chromosome pair recombines to generate a new chromosome pair (offspring). As the name implies, this process involves swapping some parts (genes) between the pair selected chromosomes. The crossover operators have a very big chance of reproducing the parent chromosomes' desirable features, and consequently expect to improve the solution quality of the problem undertaken. Probabilistic recombination is commonly used to improve or repair the offspring, rather than using complex or intelligent heuristics for this purpose (Kucukkoc and Yaman 2013). In this regards, the recombination process is not applied to every pair of chromosomes selected for recombination. The process is controlled by a crossover probability (P_c) (Javadian et al. 2011). This probability indicates the number of chromosome pairs that will be involved in the crossover operation. Typically, the probability for crossover ranges from 0.6 to 0.95.

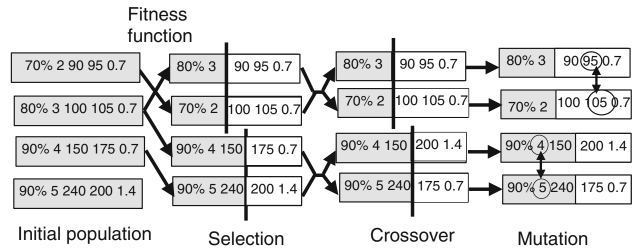

Fig. 4 Evolving process by GAs operators for optimisation WWAL problem in this paper

From several types of crossover, single-point crossover technique is used in this paper.

Mutation: mutation is another genetic operator is applied to single chromosome for creating a modified mutant—new offspring being added to the population. In mutation operation, one or more gene values of the created chromosome, randomly chosen for replacement to create another offspring. With this new offspring, the GAs may be able to reach a more optimal solution than was previously possible using crossover operator. For real-coded genetic algorithm, like the one used in this thesis, uniform mutation is applied. In real coded GAs, uniform mutation operation replaces a gene to a real number within a fixed solutions space if a gene of a chromosome is mutated. This type of mutation is preferable in research because it can configure the searching solutions spaces with generations (Abdul-Rahman et al. 2011). Like crossover operator, mutation can be done based on heuristics or on probability (P_m) (Gen and Lin 2013; Kucukkoc and Yaman 2013). Typically, this probability ranges from 0.001 to 0.01. A simplified scheme of GAs for chromosomes representing and operators is given in Fig. 4.

Mechanism of constraints handling

Before evaluating all design objectives as outcomes of modelling process it is not known if any design solution is feasible or not, especially if any conflict exists. With the mathematical model of the overall desirability function (formula 15) if any value of d_a for anyone from the objective functions (η_a, C_a, E_a and φ_a) = 0, then the overall desirability function, $D_a = 0$. Hence, the exploration of GAs to reach an optimal solution can not include infeasible solutions that involve measure values falling outside of the acceptable limits (e.g. $d_a = 0$). This mechanism is beneficial to the optimisation process as it ensures the optimal solutions will generally locate on the boundaries between feasible and infeasible solutions. The general form of this mechanism in this paper as follow:

$$D_{a(i)} = \begin{cases} D_{a(i)}, & \text{if } a(i) \text{feasible;} \\ 0, & \text{otherwise;} \end{cases} \tag{22}$$

Stopping criteria

Usually, GAs are stopped if the quality of the solution to the problem on hand is not improved significantly per generation. Thus, the designer of the algorithms has to decide the stopping criteria. Because if the algorithms are let to run for too long, time and computational power are wasted and they will also revisit all the solutions it has previously searched out. The proposed stopping criteria are as follows:

- Stop the algorithmic search process if it reaches the acceptable design solution for WWAL. Usually, when the GAs search continues with negligible improvement in solution quality for a number of generations, it has either found a good answer or has become trapped in the local optima.
- The GAs search should be stopped when no significant variations are appreciated in the average of fitness function values for a certain number of generations.

Application of developed GAs to the WWAL design

As described in the preceding sections, the developed GAs are aimed to find WWAL design solution that is supposed to be optimal in terms of setting design objectives from a range of alternatives conducted by using proposed mathematical model in "The mathematical model" section. As illustrated before, finding optimal design of WWAL involves determination of the best combination of decision variables so as to optimise objective functions. By integrating the mathematical model with developed GAs, optimal design solution can be obtained. In application of GAs, the candidate design solutions would be the array $< n_1, n_2, \ldots, n_k >$ in one line and number of columns equals number of design variables. Each column indicates a value to design variable.

Based on this general description, following, in Fig. 5, the basic pseudo-code of real genetic algorithms to maximise "D_a", as an approach to solve the WWAL design problem in this paper.

Setting GAs parameters for the WWAL optimisation design

When applying GAs, one needs careful settings for the values of the basic parameters, such as crossover rate, mutation rate, population size, selection strategy, max generation…etc. Since the performance of a GAs is greatly dependent on setting the values of these parameters. In this regard, selecting appropriate parameters might enable the algorithm to find optimal solutions in a short time while local optimum solutions may prevail and survive throughout the algorithm run with improper parameter settings (Jäntschi 2010). Because of this, the question may be raised as to which of the basic parameters and their level-setting values significantly influence the performance of GAs.

Factorial experimental design

Computational experiments are commonly used to determine which of the basic operators is most significant when compared with the others, and what its value is Mosadegh et al. (2012). Therefore, in this subsection, factorial experiments are conducted in order to obtain the parametric levels that will be set when implementing the GAs. Using a full factorial experiment extracts the maximum amount of unbiased information regarding the parameters affecting performance of GAs and also their interactions (Jebari et al. 2013). The experiments are conducted to investigate the effects of the population size N, maximum number of generation G_{max}, crossover probability P_c, mutation probability P_m. Rather than G_{max}, the level of each parameter was adopted from previous studies. Table 5 includes the number of levels and the settings of each parameter. The selection process is used a roulette wheel with elitist strategy and setting size of two. The experiment with these five parameters and three replicates are conducted giving a total of 48 runs.

The settings of these parameters are based on characteristics of fitness function. In other words, the best parameter values must be expected to have great improvement on fitness function statistics (maximum fitness function).

The results and analysis of experimenting GAs parameters

Factorial experiments are conducted for the GAs basic parameter combinations on the design problem in this paper (presented in "Design problem: mixed model assembly system optimisation problem" section) with Table 6 showing results of these experiments. The result is the fitness function of optimal solution of operational design of the research problem.

Analysis of variance (ANOVA) is used to investigate the effects of the GAs parameters and their interactions on searching performance in terms of finding the best fitness function value. Table 7 shows these effects when GAs are applied on the design problem in this paper. P value (or significant probability) from ANOVA elements is used as an informal measure about their significant effects. If the value of "P" less than 0.05 this statistically indicates that the parameter is significant.

From Table 7, P_m and G_{max} have significant effect on the performance of GAs as their P-values are below 0.05. This given conclusion from the above analysis of ANOVA applied to the experiments of the GAs parameters in the design problem of this paper can only serve as guidelines. Reasonable conclusions may need more levels of parameters for experimenting on the problem at hand and on the basis of the experimental observation an appropriate value should be selected. It

Begin

Set initial generation number $G = 0$;

Select a set of initial of initial population size including initial feasible random variables (Pz);

Evaluate Pz; % asses the fitness function "D_a" of every design solution "a" in population size (Pz);

Record best solution $a = a^*$ and best $D_a = D_{a*}$

Select and record best half of pop (Pz), e.g. pop' (Pz); best half selected based on D_a values

Repeat

G=G+1;

Select and record best half of pop' (Pz) as pop'' (Pz); reproducing same number of new offspring using predefined genetic operators; selection, crossover, and mutation on pop'' (Pz)

Record all new offspring in set pop''' (Pz);

Select from pop'' (Pz) and pop''' (Pz) to form pop' (Pz); % the size of pop' (Pz) remains same

Set pop' $(Pz) \longleftarrow$ {pop'' (Pz), pop' (Pz)};

Evaluation pop' (Pz); % assess the fitness function (D_a) of chromosomes (design solutions) in pop (Pz)

IF best "D_a"of pop' $(Pz) > D_{a*}$ **THEN**

Replace and record new best $a = a^*$ and best $D_a = D_{a*}$

END

Until G=G_{max}

Return the best a^* visited and final $D_{a*} = D(a*)$ value

Fig. 5 The pseudo-code of GAs for maximisation of D_a

Table 5 The levels of basic parameters for GAs

Parameters	Number of levels	Setting
Crossover probability P_c	2	0.6, 0.90
Mutation probability P_m	2	0.005, 0.01
Population size N	2	40, 50
Maximum number of generations $G_{\max}$	2	100, 300

is therefore advisable to run further experiments with different settings of each parameter and compare the performance of the GAs. This procedure aims to improve the stability of the algorithm and avoid setting parameters randomly so as to achieve the goal of quantifying parameter settings and significantly advance the performance of the algorithm.

Further experimentation and analysis with GAs parameters

For further experimentation analysis and consequently selecting the best values of P_c, P_m and population size N a comparison of the effect of changing the values of these parameters on GAs performance is conducted. Except for the changing parameter all other parameters are fixed during experimentation. In this context, in order to compare with the result of each case, basic parameters were assigned so that the size of entity population is 40, the crossover probability is 0.8, and the mutation probability is 0.005. That is, when there is a change in N value from 40 to 200 to reproduce chromosomes, the P_c is fixed to 0.8 and the P_m is fixed to 0.005. Also when there is a change in the P_c value then the N and the P_m values are fixed.

The G_{max} is not considered in further experimentation as the effect of this parameter on GAs performance was well known from previous studies—if it is large it will enhance the opportunities of obtaining the optimum solution but will augment the cost of operation and the computational time is not limited. The assumption is that the simulation model of GAs is run for 300 generations whilst taking into account the constraints of the design problem, also the dependent vari-

Table 6 Computational results for combinations of GAs basic parameters

Experiment no.	GAs parameters				Fitness function
	P_c	P_m	N	G_{max}	
1	0.6	0.005	50	100	0.656382
2	0.6	0.01	40	100	0.635648
3	0.6	0.005	40	300	0.640247
4	0.6	0.005	40	100	0.616175
5	0.9	0.01	50	300	0.65348
6	0.6	0.01	50	100	0.661875
7	0.6	0.01	50	300	0.649159
8	0.6	0.01	40	300	0.632918
9	0.9	0.005	40	300	0.660785
10	0.6	0.01	40	100	0.651889
11	0.9	0.005	40	100	0.657718
12	0.9	0.005	40	300	0.644749
13	0.9	0.01	50	100	0.605704
14	0.9	0.005	50	100	0.65348
15	0.9	0.01	40	100	0.605704
16	0.6	0.005	50	100	0.660785
17	0.6	0.005	40	300	0.551074
18	0.9	0.005	50	100	0.544543
19	0.6	0.005	50	300	0.660785
20	0.9	0.005	40	100	0.644749
21	0.6	0.005	40	100	0.642614
22	0.6	0.01	40	300	0.635648
23	0.9	0.005	50	300	0.657718
24	0.9	0.01	40	300	0.656382
25	0.9	0.01	40	300	0.616175
26	0.6	0.005	50	300	0.649159
27	0.9	0.01	50	300	0.651889
28	0.6	0.005	50	100	0.551074
29	0.9	0.005	50	300	0.561576
30	0.6	0.01	50	300	0.661875
31	0.6	0.005	50	300	0.551074
32	0.9	0.01	40	300	0.65348
33	0.9	0.01	50	100	0.616175
34	0.9	0.01	50	100	0.649159
35	0.9	0.01	40	100	0.661875
36	0.9	0.01	50	300	0.661875
37	0.6	0.01	50	100	0.626139
38	0.9	0.01	40	100	0.630947
39	0.6	0.01	50	300	0.568407
40	0.9	0.005	40	300	0.649159
41	0.6	0.01	40	300	0.635648
42	0.6	0.005	40	100	0.65348
43	0.6	0.005	40	300	0.65348
44	0.6	0.01	40	100	0.626139
45	0.9	0.005	40	100	0.65348

Table 6 continued

Experiment no.	GAs parameters				Fitness function
	P_c	P_m	N	G_{max}	
46	0.6	0.01	50	100	0.644749
47	0.9	0.005	50	300	0.657718
48	0.9	0.005	50	100	0.647301

Table 7 ANOVA table of factorial experiments of basic parameters on design problem

Parameters	Sum of squares	Df	Mean square	F value	P value	Significance
P_c	1.268×10^{-8}	1	1.268×10^{-8}	1.829×10^{-3}	0.9661	–
P_m	2.984×10^{-5}	1	2.984×10^{-5}	4.31	0.0450	*
N	5.316×10^{-6}	1	5.316×10^{-6}	0.77	0.3868	–
G_{max}	5.601×10^{-5}	1	5.601×10^{-5}	8.08	0.0072	*
$P_c \times P_m$	6.576×10^{-9}	1	6.576×10^{-9}	9.488×10^{-4}	0.9756	–
$P_c \times N$	9.967×10^{-6}	1	9.967×10^{-6}	1.44	0.2381	–
$P_c \times G_{max}$	5.420×10^{-7}	1	5.420×10^{-7}	0.078	0.7813	–
$P_m \times N$	1.003×10^{-6}	1	1.003×10^{-6}	0.14	0.7058	–
$P_m \times G_{max}$	2.136×10^{-5}	1	2.136×10^{-5}	3.08	0.0874	–
$N \times G_{max}$	1.104×10^{-7}	1	1.104×10^{-7}	0.016	0.9002	–

* The parameter has an effect

able in these experiments is the maximum fitness function value. In consideration of the variation in the fitness function value that may be caused by changes to the initial population, three replications of GAs run are executed and it was decided that the GAs run showing the best performance (in terms of convergence speed and fitness function value) can be selected as an optimal GAs with respect to the values of experimented basic parameter. The goal is to select the GAs parameters in which peak fitness values are more likely to be found and also the convergence to an optimal solution occurs in a short time period.

Experimentation of crossover operator:the crossover probabilities used in this experimentation are 0.5, 0.6, 0.8, 0.9, and 1.0 and test individually. The algorithm has been run for each crossover probability P_c with the other parameters fixed. The results summarised in Fig. 6 give a figure that compares only the average of best fitness function value found in a generation.

As Fig. 6 shows, when P_c is 0.5, the optimisation model converges to fitness a function value of 0.644749, from generation 112. In case a P_c is 0.6 the model converges to value, 0.644749, after generation 56. The model converges to fitness value 0.661875, after generation 45, when a P_c is 0.8. The model converges to fitness value 0.65348, after generation 145 when P_c is 0.9 and the model converges to fitness value 0.616175, after 229 generations when P_c is 1.0. The experimental result, clearly indicates that when a P_c is 0.8 then the fastest model converges to highest fitness values of the P_c values, 0.661875.

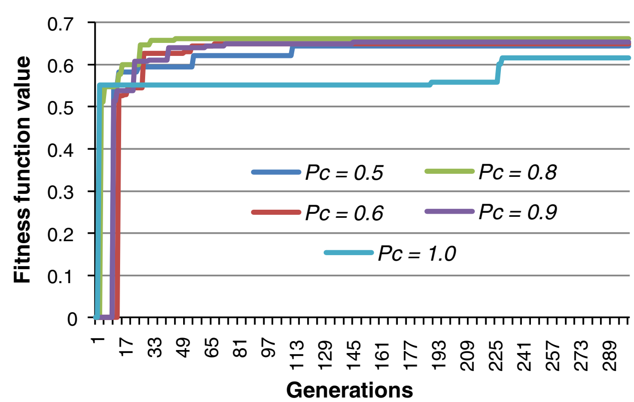

Fig. 6 Comparison of crossover probability value on converging to best fitness function value

Experimentation of mutation operator: according to the results in Fig. 7, in case a P_m is 0.005, the model converges to fitness value 0.6402472, after 43 generations while other values of P_m; 0.007, 0.008, and 0.009 converged the optimisation model to the low values of fitness function, 0.6110067, 0.6329181, and 0.5445833 with more generations; 97, 84, and 48, respectively. At P_m is 0.01 the model converged to fitness values 0.621217 after 32 generations. Hence, P_m at 0.005 has the highest performance.

Experimentation of population size: the tested population size in experimentation is 40, 50, 60, 80 and 200. Results in are summarised in Fig. 8 which shows the value of fitness function value per each generation after

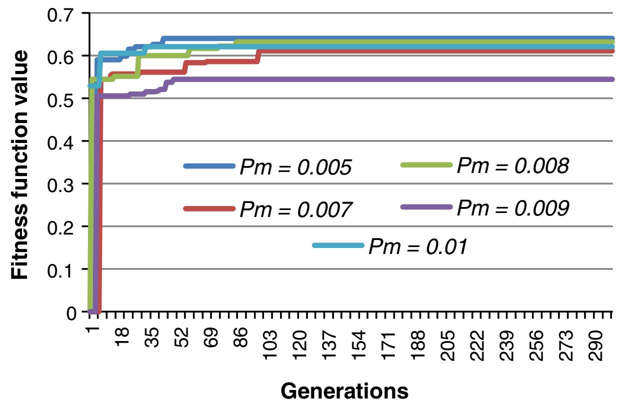

Fig. 7 Comparison of mutation probability value on converging to best fitness function value

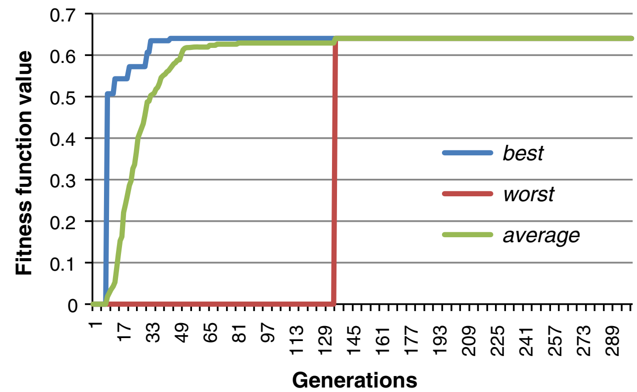

Fig. 9 The optimisation process for maximizing fitness function for operational design solutions of mixed model WWAL during 300 generations

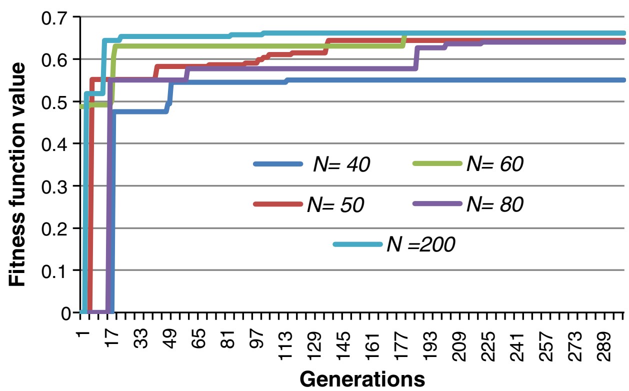

Fig. 8 Comparison of the effect of population size on converging to best fitness function value

Table 8 The optimal operational design solution of mixed model WWAL found by GAs search

Operational design variables value					Performance measures			
x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4
80 %	1	0.7	128	158	0.229	706	2.87	0.141

optimising the mathematical model of the design problem during generation 300. Results show that convergence to better solutions happens sooner than with bigger population sizes. The population size of 60 reached best fitness value of 0.661875 at generation 179. On the other hand, the fitness value to population size of 50, 80 reached to 0.644749 and 0.640247 during generation 136 and 221, respectively.

Summary of experimentation

Examining the effects of GAs parameters values on the performance of the algorithm to find the best parameter combination for the design problem of this paper tends to conclude the best combination of parameters are the following:

- Crossover probability is 0.8;
- Mutation probability is 0.005;
- Population size is 60; and
- Maximum number of generations is 300.

The above selected parameter values for optimising the operational design of mixed model WWAL; are consistent with the results obtained from De Jong's (1975) study for off-line setting of GAs parameters. In that study, DeJong selected low values for crossover probability and mutation probability, 0.6 and 0.001, while the selected population size was 50 and selection strategy was elitist.

The results and analysis of design optimisation problem

The developed GAs in this paper is executed using the best parameter values gained from the experiments to solve the problem of optimising operational design for mixed model WWAL presented in "Design problem: mixed model assembly system optimisation problem" section. Figure 9 shows the convergence to best function value of optimising problem, 0.640247. It also shows the change in combinations for the design variables obtained through the operational design variables levels, which not changed the fitness function values among the solutions composing the last 258 generations. Therefore, the final optimal design variable combinations that simultaneously satisfy the requirements placed (i.e. optimisation objectives) on each one of the measures and operational design variables (i.e. multiple-objective optimisation) is determined. The final optimal operational design variable combinations of WWAL are presented in Table 8.

Assembly line design assessment

Although optimised operational design for mixed model WWAL are generally reliable as shown in Table 5, it has the following drawbacks:

- Low efficiency due to high of balance of labour blockage. The value of this design objective was 0.111 indicated the optimised operational design generates blockage and starvation in the previous and successive workstations (5, 6) causing a reduction in the worker productivity and consequently in line throughput. This is because the worker productivity depends on the overall cycle time for assembling the product and as usual the line efficiency varies considerably with the cycle time.
- High expenditure energy cost. Since the expenditure in one walk cycle was 3.28 kcal, the current design of the system where a worker carries out the assembly tasks by standing, carrying, walking and waiting during blockage in an 8 hr. workday, was found to almost exceed the physiological capacity of worker to maintain suitable physical fitness and work ability. In WWAL operational design, the worker's metabolic response to the amount of work assigned was affected by two variables;

– Walking distance: as expected, walking distance had the greatest effect on walking time; and
– Behavioural factors of workers, in particular walking speed.

Therefore, implementing a WWAL approach needs "some time" involving due consideration of both productivity and ergonomics in the early design stage of manufacturing systems. Manual assembly systems can be designed using different structural designs. In general, different structural system designs have a profound impact on the performance of the system itself in terms of ergonomics and productivity. Structural system designs are determined by different variables; (1) the arrangement of the workstations (layout) (2) the number of workstations (3) the movement distance between workstations (Hu et al. 2011). Also, the structural design of the system should assure safety of movement especially in dynamic systems like WWAL where workers slipping while performing their job is a commonly reported due to flooring surface problems. These variables, therefore, should be considered satisfactorily in order to ensure that an assembly system is appropriately designed for WWAL approach implementation requirements.

The mathematical model developed in Al-Zuheri et al. (2013) and the optimisation approach developed in this paper, may allow prediction of those structural designs that maximise the flexibility in the system when future changes in product demand are required together with maximised performance and reduced costs by considering both productivity and ergonomics together.

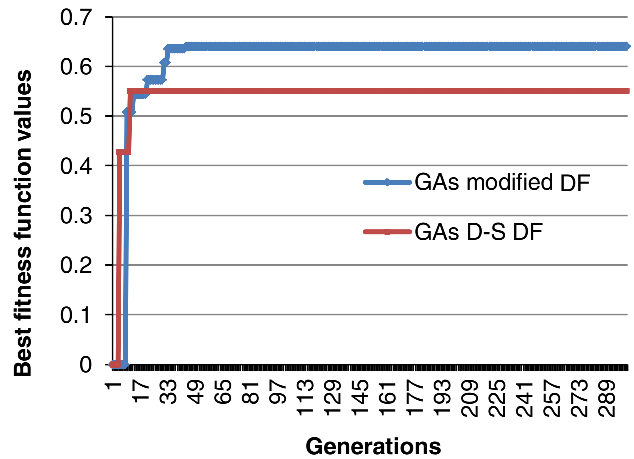

Fig. 10 Comparison of running a GAs using the modified desirability function (GAs modified DF) and running GAs using Derriger and Suich desirability function (GAs D-S DF) only

Comparison of GAs performances using the modified desirability function versus using Derriger and Suich desirability function

To illustrate the effectiveness of using the modified desirability function, Fig. 10 illustrates a comparison of running the GAs on illustrative example problem using this desirability function and Derriger and Suich desirability function.

After 45 generations, the GAs with modified desirability function converged to near the global optimal design solution with value of fitness was 0.640247. While the Derriger and Suich optimal design solution in searching GAs had the fitness value of 0.551074 and converted to it after 12 generations, after that, the local search failed to increase the fitness value. The design solutions which found by GAs search with the type of Derriger and Suich desirability function is listed as follows.

$$x_1 = 70\,\%, x_2 = 2, x_3 = 0.7, x_4 = 121,$$
$$x_5 = 152, y_1 = 0.227, y_2 = 637, y_3 = 2.923, y_4 = 0.149.$$

Since the Derriger and Suich desirability function cannot evaluate the relative fitness of infeasible chromosomes the GAs cannot determine which parents are better for reproduction. Evidence of this is that the GAs ran with the Derriger and Suich desirability function trapped in local optimal as shown in Fig. 9. While the GAs that ran with the modified desirability function converged to a value near the global optimal in same number of generations 300. This is because GAs searching in this case is able to consider even infeasible solutions that as possible consist of feasible regions.

Conclusions

This paper has described the development of the optimisation design approach of manual assembly line with walking worker. As GAs were selected in this paper as the appropriate optimisation approach for solving the WWAL design problem, the fundamental structure of GAs was outlined. It included chromosome representation, generation of initial population, fitness function, selection strategy, genetic operators (e.g. selection, crossover and mutation operators), handling of constraints and stopping criteria. Real number coded chromosome representation was adopted as the internal object representation for the solutions search in the GAs of this paper. The algorithm is designed to implement the developed GAs for finding a WWAL design depending on the given input parameters and design variables to make this task flexible and systematic as possible. Setting basic parameters values for developed GAs for best performance was based on a premature convergence of fitness function to optimal solution. Furthermore, the development process of GAs also included introducing a penalty function as a scheme to handling the constraints of optimisation in algorithm.

For demonstrating the effectiveness of an algorithm developed to handle the WWAL optimisation design problem, GAs were run to find an optimal operational design for the designed problem; mixed model manual assembly line intended to be run with walking workers.

Since the solution design (chromosome representation) in any GAs is quite problem domain dependant, hence the developed GAs procedure here cannot be used to effectively solve different optimisation problems elsewhere. Based on the above this paper successfully introduces a unique GAs.

Acknowledgments The authors would like to express their appreciation to anonymous referees for their helpful comments.

References

Abdul-Rahman, O. A., Munetomo, M., & Akama, K. (2011). An adaptive resolution hybrid binary-real coded genetic algorithm. *Artificial Life and Robotics*, *16*(1), 121–124.

Al-Zuheri, A., Luong, L., & Xing, K. (2013). Mathematical modelling for process design of walking worker assembly line (WWAL) in productivity and ergonomics perspectives. *International Journal of Industrial and Systems Engineering*, *14*(1), 104–137.

Alzuheri, A., Luong, L., & Xing, K. (2010). Ergonomics design measures in manual assembly work. *Paper presented at the second international conference on engineering systems management and its applications (ICESMA)*, Sharjah, UAE, 30 March–1 April 2010.

Bley, H., Zenner, C., & Bossmann, M. (2007). Walking worker assembly lines—A contribution to lean production. *Paper presented at the international conference competitive manufacturing—The challenge of digital manufacturing*, Stellenbosch, South Africa, 31 January–2 February 2007.

Chambers, L. D. (2010). Practical handbook of genetic algorithms: Complex coding systems (3rd ed). Boca Raton: CRC Press.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. Ann Arbor, MI: University of Michigan.

Derringer, G., & Suich, R. (1980). Simultaneous optimization of several response variables. *Journal of Quality Technology*, *12*(4), 214–219.

Garg, A., Chaffin, D., & Herrin, G. (1978). Prediction of metabolic rates for manual materials handling jobs. *American Industrial Hygiene Association Journal*, *39*(8), 666–764.

Garg, P. (2010). Evolutionary computation algorithms for cryptanalysis: A study. (IJCSIS) International Journal of Computer Science and Information. *Security*, *7*(1), 306–310.

Gen, M., & Lin, L. (2013). Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey. *Journal of Intelligent Manufacturing*, *1–18*. doi:10.1007/s10845-013-0804-4.

Goldberg, D. E. (2012). Genetic algorithms in search, optimization, and machine learning (30th ed.). Cincinnati, OH: Addison-Wesley.

Hu, S. J., Ko, J., Weyand, L., ElMaraghy, H. A., Lien, T. K., Koren, Y., et al. (2011). Assembly system design and operations for product variety. *CIRP Annals—Manufacturing Technology*, *60*(2), 715–733.

Jäntschi, L. (2010). *Genetic algorithms and their applications*. Cluj-Napoca: University of Agricultural Sciences and Veterinary Medicine Cluj-Napoca.

Javadian, N., Aghajani, A., Rezaeian, J., & Sebdani, M. J. G. (2011). A multi-objective integrated cellular manufacturing systems design with dynamic system reconfiguration. *International Journal of Advanced Manufacturing Technology*, *56*(1), 307–317.

Jebari, K., Bouroumi, A., & Ettouhami, A. (2013). Parameters control in GAs for dynamic optimization. *International Journal of Computational Intelligence Systems*, *6*(1), 47–63.

Kucukkoc, I., & Yaman, R. (2013). A new hybrid genetic algorithm to solve more realistic mixed-model assembly line balancing problem. *International Journal of Logistics Systems and Management*, *14*(4), 405–425.

Lassalle, S., Wang, Q., Owen, G., & Mileham, A. (2007). A study of in-process waiting time on a linear walking worker assembly line proceedings of the institution of mechanical engineers. *Part B, Journal of engineering manufacture*, *221*(12), 1763–1770.

Li, J.-P., Balazs, M. E., & Parks, G. T. (2007). Engineering design optimization using species-conserving genetic algorithms. *Engineering Optimization*, *39*(2), 147–161.

Liu, Z., Wong, Y. S., & Lee, K. S. (2011). A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm. *Journal of Intelligent Manufacturing*, *22*(6), 891–907.

Longo, F., Massei, M., & Nicoletti, L. (2012). An application of modeling and simulation to support industrial plants design. *International Journal of Modeling, Simulation, and Scientific Computing*, *3*(1). doi:10.1142/S1793962312400016.

Martinec, D., & Bundzel, M. (2013). Evolutionary algorithms and reinforcement learning in experiments with slot cars. *Paper presented at the international conference on process control (PC)*, Strbske Pleso, Slovakia, 18–21 Jun 2013.

Mileham, A., Jeffries, A., Owen, G., & Pellegrin, F. (2000). The design and performance of walking fitter assembly lines. *Paper presented at the international conference on CADCAM robotics and factories of the future (CARS&FOF2000)*, Trinidad and Tobago, 26–28 June 2000.

Mileham, A., Wang, Q., Lassalle, S., & Owen, G. (2008). Modelling a walking worker assembly system. *Paper presented at the 18th international conference on flexible automation and intelligent manufacturing*, Skövde, Sweden, 30 June–2 July 2008.

Mosadegh, H., Zandieh, M., & Ghomi, S. M. T. (2012). Simultaneous solving of balancing and sequencing problems with stationdependent assembly times for mixed-model assembly lines. *Applied Soft Computing*, *12*(4), 1359–1370.

Moslemipour, G., & Lee, T. S. (2012). Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing systems. *Journal of Intelligent Manufacturing*, *23*(5), 1849–1860.

Nakade, K., & Nishiwaki, R. (2008). Optimal allocation of heterogeneous workers in a U-shaped production line. *Computers & Industrial Engineering*, *54*(3), 432–440.

OSHA. (2009). *Ergonomic guidelines for manual material handling*. Cincinnati, OH: Walking of Assembly Workers.

Sawhney, R. (2012). Implementing labor flexibility: A missing link between acquired labor flexibility and plant performance. *Journal of Operations Management*, *31*(1–2), 98–108.

Scholl, A., Fliedner, M., & Boysen, N. (2010). ABSALOM: Balancing assembly lines with assignment restrictions. *European Journal of Operational Research*, *200*(3), 688–701.

Shishir, B. N. (2010). Changing from conveyor to work-cell-based systems to deal with fluctuations in demand. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *224*(1), 169–176.

Tang, P.-H., & Tseng, M.-H. (2012). Adaptive directed mutation for real-coded genetic algorithms. *Applied Soft Computing*, *13*(1), 600–614.

Wang, Q., Lassalle, S., Mileham, A. R., & Owen, G. W. (2009). Analysis of a linear walking worker line using a combination of computer simulation and mathematical modeling approaches. *Journal of Manufacturing Systems*, *28*(2–3), 64–70.

Wang, Q., Owen, G., & Mileham, A. (2005). Comparison between fixed- and walking-worker assembly lines. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *219*(11), 845–848.

Wang, Q., Owen, G., & Mileham, A. (2007a). Determining numbers of workstations and operators for a linear walking-worker assembly line. *International Journal of Computer Integrated Manufacturing*, *20*(1), 1–10.

Wang, Q., Pan, L., Mileham, A. R., Owen, G. W., & Newnes, L. B. (2007b). Modelling randomness of a manual assembly system with walking workers. *International Journal of Industrial and Systems Engineering*, *2*(2), 195–210.

Waters, T., Lu, M., Piacitelli, L., Werren, D., & Deddens, J. (2011). Efficacy of the revised NIOSH lifting equation to predict risk of low back pain due to manual lifting. *Journal of Occupational and Environmental Medicine*, *53*(9), 1061–1067.

Yu, J., & Yin, Y. (2010). Assembly line balancing based on an adaptive genetic algorithm. *International Journal of Advanced Manufacturing Technology*, *48*(1–4), 347–354.

Zhan, Y., & Zhang, R. (2013). A genetic algorithm for the balancing and sequencing problem in the mixed-model U-line. In *2013 IEEE symposium on computational intelligence in production and logistics systems (CIPLS)*, Singapore, 16–19 April 2013.

Zhu, H., Wu, D., & Fan, X. (2010). Interactive assembly tool planning based on assembly semantics in virtual environment. *International Journal of Advanced Manufacturing Technology*, *51*(5–8), 739–755.