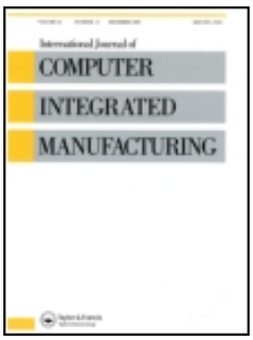

International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: http://www.tandfonline.com/loi/tcim20

Combining local search into non-dominated sorting for multi-objective line-cell conversion problem

Yang Yu , Jiafu Tang , Wei Sun , Yong Yin & Ikou Kaku

To cite this article: Yang Yu , Jiafu Tang , Wei Sun , Yong Yin & Ikou Kaku (2013) Combining local search into non-dominated sorting for multi-objective line-cell conversion problem, International Journal of Computer Integrated Manufacturing, 26:4, 316-326, DOI: 10.1080/0951192X.2012.717717

To link to this article: https://doi.org/10.1080/0951192X.2012.717717

Published online: 02 Nov 2012.

International Journal of Computer Integrated Manufacturing, 2013
Vol. 26, No. 4, 316–326, http://dx.doi.org/10.1080/0951192X.2012.717717

Taylor & Francis
Taylor & Francis Group

Combining local search into non-dominated sorting for multi-objective line-cell conversion problem

Yang Yu[a]*, Jiafu Tang[a], Wei Sun[b], Yong Yin[c] and Ikou Kaku[d]

[a]*Institute of Systems Engineering, State Key Laboratory of Synthetic Automation for Process Industries, Northeastern University, Shenyang 110819, P.R. China;* [b]*College of Business Administration, Liaoning University, Shenyang 110136, P.R. China;* [c]*Department of Economics and Business Management, Yamagata University, Yamagata 990-8560, Japan;* [d]*Faculty of Environmental and Information Studies, Tokyo City University, Yokohama 224-8551, Japan*

(*Received 27 August 2011; final version received 19 June 2012*)

Multi-objective decision-making is often used in line-cell conversion problem. It is the key issue of developing line-cell conversion successfully. In this article, a multi-objective optimisation model is proposed to investigate two line-cell conversion performances: the total throughput time and the total labour hours. It is proved that the model is a non-deterministic polynomial (NP)-hard problem and its Pareto-optimal front is non-convex. Owing to the properties of cell formation and cell loading in line-cell conversions, an approach of combining local search into non-dominated sorting genetic algorithm II is proposed to obtain better non-dominated solutions. Several numerical simulation experiments are performed to show the utility and performance of our approach.

Keywords: line-cell conversion; multi-objective optimisation; local search; non-dominated sorting genetic algorithm

1. Introduction

Line-cell conversion is a new innovation of a manufacturing system developed in Japan recently. Its essence is that converting a traditional conveyor assembly line to a cell (i.e. called *seru*, a Japanese word for cellular organism) type system where one (or multiple) worker carries out all of the operations of a job in a U-shaped station layout. *Seru* has also been adopted in the US, Europe and Korea, China and other countries (Yin *et al*. 2011). Equipment in *seru* such as machines is cheap, because most assembly tasks within a *seru* are manual so need only simple and cheap equipment, e.g. hand tools and workbenches. Duplicating this kind of equipment for multiple *serus* is usually not costly (Stecke *et al*. 2012, Yin *et al*. 2012). To be consistent with a previous research (Kaku *et al*. 2009), we use 'cell' in this article to represent *seru* and call the conversion from assembly lines to *serus* 'line-cell conversion'.

Line-cell conversion is a very topical and important decision-making problem, because the total productivity of manufacturers may be increased very largely by introducing line-cell conversion. For example, it can reduce throughput time and required labour hours. Some amazing cases related to these two *seru* performances are: the throughput time was reduced by 53% at Sony Kohda; and 35,976 required workers, equal to 25% of Canon's previous total workforce, have been saved (Yin *et al*. 2008, 2011, Liu *et al*. 2010). However, how to complete this conversion is a very complicated decision-making problem, because they must decide on how many cells should be formatted and which workers should be assigned in each cell. Such technical and decision-making problems in line-cell conversion had been defined as line-cell conversion problem (see Kaku *et al*. 2008, 2009).

By using a multi-objective mathematical model, in which two objectives of the total throughput time (TTPT) and the total labour hours (TLH) are considered, Kaku *et al*. (2009) investigated several operational factors. However, there is a weakness in Kaku's research. An enumeration for single-objective, but not non-dominated method, was used to analyse Kaku et al.'s multi-objective model. To overcome the weakness, a non-dominated sorting genetic algorithm II (NSGA-II) based algorithm (Yu *et al*. 2011) is used to solve the two-objective optimisation problem that minimises the TTPT and the TLH.

Without loss of generality, we simplified their model into a simple case in which an assembly conveyer line is converted into a pure cell system. However, even in the simplified model, its solution space is still huge. According to the property of the multi-objective line-cell conversion problem, this article develops an algorithm of combining local search into NSGA-II algorithm to obtain better non-dominated solutions. The role of the local search is to enhance the intensification process in the genetic

*Corresponding author. Email: yuyang@ise.neu.edu.cn

search, which had been reported good possibilities for hybridisation of a genetic algorithm (GA; Arroyo and Armentano 2005, Grosan and Abraham 2007). It may be expected that, by using our hybrid searching technique, the search efficiency (speed and accuracy of searching better non-dominated front) may be upped certainly. Several numerical simulation experiments are executed to illustrate the utility of our proposed approach.

The remainder of this research is organised as follows. Section 2 presents the modified model, proves that the model is non-deterministic polynomial (NP)-hard and clarifies the properties of cell formation and cell loading. In Section 3, the Pareto-optimal front of the multi-objective line-cell conversion model is proved to be non-convex, and an algorithm of combining local search into NSGA-II is proposed to obtain better non-dominated solutions. According to the properties of line-cell conversion, several implement operators of NSGA-II and local search are modified. In Section 4, several comparison examples to illustrate the utility of our approach are given. Conclusions and future research are given in Section 5.

2. The line-cell conversion model

2.1. Problem description

For simplicity and without loss of generality, we just consider a simple case in which a traditional conveyor assembly line is converted to a pure cell system as shown in Figure 1. All of workers are assigned to cells (i.e. cell formation).

Figure 2 shows a cell loading example with six batches and two cells. The length of the rectangle charts in Figure 2 is the flow time of a product batch. We adopt a first come first serve (FCFS) principle. An arriving product batch is assigned to the empty cell with the smallest cell number. If all cells are occupied, the product batch is assigned to the cell with the earliest finish time.

For evaluating the system performance, two criteria are considered. Firstly, we define the TTPT to represent the system productivity, i.e. the time of all the product batches being assembled. Secondly, we define the TLH to represent the work efficiency, i.e. the cumulative working time of all workers. Therefore, our problem is to determine how many cells should be formed and how to assign workers and product batches to appropriate cells to minimise the two objectives, i.e. the TTPT and the TLH.

2.2. Problem features and assumption

The following assumptions are considered in this research to construct the model of a pure cell system:

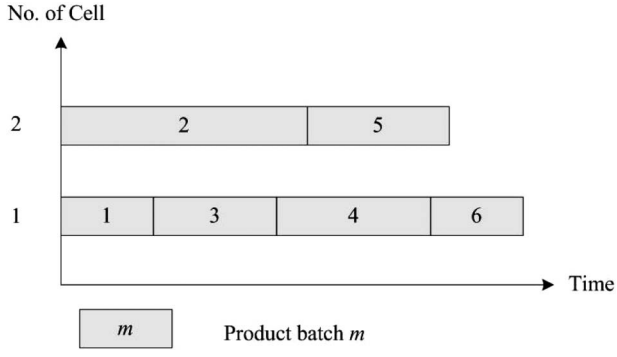

Figure 2. An example of FCFS scheduling rule in a cell system (cell loading).

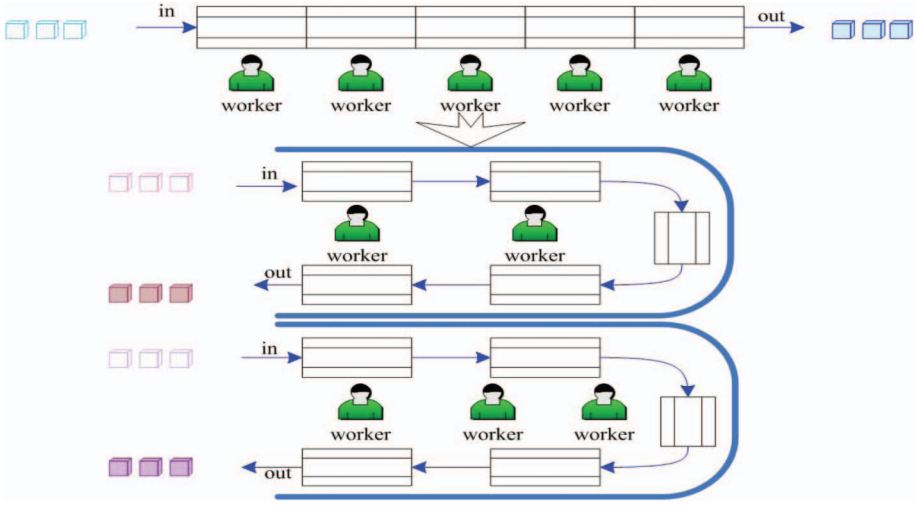

Figure 1. Converting an assembly line to a cell system (cell formation).

(1) There are N product types needed to be assembled. These types of products are divided into M product batches, and each batch consists of the same product type. The types and batches of products are known in advance.
(2) The cost of machines is not taken into account when adding machines during line-cell conversion, because most assembly tasks within a cell are manual so need only simple and cheap equipment. Duplicating them is usually not costly (Stecke *et al*. 2012, Yin *et al*. 2012).
(3) A product batch is just processed in a cell, without considering batch splitting.
(4) Every assembly task is performed within its exclusive station. If a product type does not need some task, the product skips the task's station.
(5) The number of tasks in each cell is the same as the number of tasks on assembly line.
(6) A worker only does one assembly task in the assembly line, but a worker in a cell can operate all the tasks in the cell, and there is no disruption or delay between adjacent tasks.
(7) The number of workers is the same as the number of tasks (or stations) in the assembly line.
(8) The number of workers in each cell may be different, but not more than the total number of workers.
(9) The set-up time is considered when two different types of products are processed consecutively; otherwise the set-up time is zero.

2.3. Notations

The following terms have been defined:

- Indices

i: Index set of workers ($i = 1, 2, \ldots, W$).
j: Index set of cells ($j = 1, 2, \ldots, J$).
n: Index set of product types ($n = 1, 2, \ldots, N$).
m: Index set of product batches ($m = 1, 2, \ldots, M$).
k: Index set of sequences of product batches in a cell ($k = 1, 2, \ldots, M$).

- Parameters

$$V_{mn} = \begin{cases} 1, & \text{if product type of product batch } m \text{ is } n \\ 0, & \text{otherwise.} \end{cases}$$

B_m: Size of product batch m.
T_n: Cycle time of product type n in the assembly line.
SCP_n: Set-up time of product type n in a cell.
η_i: Upper bound on the number of tasks for worker i in a cell. If the number of tasks assigned to worker i is more than η_i, worker i's average task time within a cell will be longer than her or his task time within the original assembly line.
C_i: Coefficient of variation of worker i's increased task time after the line-cell conversion, i.e. from a specialist to a completely cross-trained worker.
ϵ_i: Worker i's coefficient of influencing level of doing multiple assembly tasks.
β_{ni}: Skill level of worker i for one task of product type n.

- Decision variables

$$X_{ij} = \begin{cases} 1, & \text{if worker } i \text{ is assigned to cell } j \\ 0, & \text{otherwise} \end{cases}$$

$$Z_{mjk} = \begin{cases} 1, & \text{if product batch } m \text{ is assigned to cell } j \text{ in sequence } k \\ 0, & \text{otherwise.} \end{cases}$$

In addition, if $k = 0$, $Z_{mjk} = 0$.

- Variables

SC_m: Set-up time of product batch m in a cell.
TC_m: Assembly task time of product batch m per station in a cell.
FC_m: Flow time of product batch m in a cell.
FCB_m: Begin time of product batch m in a cell.

2.4. Problem formulation

We consider an assembly planning problem in which there is an assembly product mix with M product batches and N product types. W workers are assigned to assembly cells during the line-cell conversion. The batches are assigned to cells with the FCFS principle. We define the TTPT of the cell system following this FCFS principle.

Firstly, the cross-training process can be represented as a V-shaped learning curve. In other words, in the early period of the line-cell conversion, workers often cost more time on tasks she or he is not familiar with (Yin *et al*. 2012). So, it is reasonable to assume that a worker's skill level varies with the number of tasks assigned to her or him. In this article, we assume that if the number of worker i's tasks within a cell is over her or his upper bound η_i, i.e. $W > \eta_i$, then the worker will cost more average task time than her or his task time within the original assembly line. The details are given as Equation (1).

$$C_i = \begin{cases} 1 + \varepsilon_i(W - \eta_i), & W > \eta_i \\ 1, & W \geq \eta_i \end{cases}, \quad \forall i \tag{1}$$

Secondly, the task time of a product also varies with workers. Therefore, for a cell, the task time of a product is calculated by average task time of workers in the cell. The task time of product batch m per station in a cell can be represented via following equation:

$$\mathrm{TC}_m = \frac{\sum_{n=1}^{N}\sum_{i=1}^{W}\sum_{j=1}^{J}\sum_{k=1}^{M} V_{mn}T_n\beta_{ni}C_iX_{ij}Z_{mjk}}{\sum_{i=1}^{W}\sum_{j=1}^{J}\sum_{k=1}^{M} X_{ij}Z_{mjk}} \tag{2}$$

Finally, the set-up time SC_m, the flow time FC_m and the begin time FCB_m of product batch m are represented as below:

$$\mathrm{SC}_m = \sum_{n=1}^{N} \mathrm{SCP}_n V_{mn}\left(1 - \sum_{m'=1}^{M} V_{m'n}Z_{m'j(k-1)}\right), \quad (j,k)|Z_{mjk} = 1, \quad \forall j,k \tag{3}$$

$$\mathrm{FC}_m = \frac{B_m \mathrm{TC}_m W}{\sum_{i=1}^{W}\sum_{j=1}^{J}\sum_{k=1}^{M} X_{ij}Z_{mjk}} \tag{4}$$

$$\mathrm{FCB}_m = \sum_{s=1}^{m-1}\sum_{j=1}^{J}\sum_{k=1}^{m} (\mathrm{FC}_s + \mathrm{SC}_s)Z_{mjk}Z_{sj(k-1)} \tag{5}$$

Equation (3) states the set-up time of product batch m. The set-up time is considered when two different types of products are processed consecutively; otherwise the set-up time is zero. Equation (4) states the flow time of product batch m within a cell. Equation (5) states the begin time of each product batch. There is no waiting time between two product batches so that the begin time of one product batch is the aggregation of flow time and set-up time of all the previous product batches that are assembled in the same cell.

The comprehensive mathematical model is given in Equations (6)–(12).

Objective functions:

$$\mathrm{TTPT} = \mathrm{Min}\left\{\underset{m}{\mathrm{Max}}(\mathrm{FCB}_m + \mathrm{FC}_m + \mathrm{SC}_m)\right\} \tag{6}$$

$$\mathrm{TLH} = \mathrm{Min}\sum_{m=1}^{M}\sum_{i=1}^{W}\left(\sum_{j=1}^{J}\sum_{k=1}^{M}\mathrm{FC}_m X_{ij}Z_{mjk}\right) \tag{7}$$

Subject to:

$$\sum_{i=1}^{W} X_{ij} \leq W, \quad \forall j \tag{8}$$

$$\sum_{j=1}^{J} X_{ij} = 1, \quad \forall i \tag{9}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{M} Z_{mjk} = 1, \quad \forall m \tag{10}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{M} Z_{mjk} = 0, \quad \left\{j \middle| \sum_{i=1}^{W} X_{ij} = 0,. \quad \forall j\right\} \tag{11}$$

$$\sum_{j=1}^{J}\sum_{k=1}^{M} Z_{mjk} \leq \sum_{j'=1}^{J}\sum_{k'=1}^{M} Z_{(m-1)j'k'}, \quad m = 2,3,\ldots,M \tag{12}$$

Equation (6) states the objective to minimise the TTPT of all product batches. The TTPT is the finish time of the last completed product batch. Equation (7) states the objective to minimise the TLH of all product batches. The TLH is the cumulative working time of all workers in the cell system. Equation (8) is the constraint of number of workers in a cell. The maximum number of workers in one cell is the total number of workers. Equation (9) is the rule of worker assignment which ensures that a worker is only assigned to a cell. Equation (10) is the assignment rule in which a product batch is only assigned to a cell. Equation (11) ensures that a product batch cannot be assigned to an empty cell. Equation (12) means that product batches must be assigned sequentially.

Generally speaking, even if we simplify the line-cell conversion model developed by Kaku *et al*. (2009), the resulted problem is still difficult.

Theorem 1. Line-cell conversion is NP-hard.

Proof. Line-cell conversion includes cell formation and cell loading. Cell formation is to partition W workers of an assembly line into a set of non-empty cells; each cell may have one or several workers, and each worker is only assigned to one cell. If the term 'worker' is generalised as element, cell formation is to partition W elements of a set into unordered pairwise disjoint non-empty subsets. Therefore, cell formation is an instance of the unordered set partition problem. It is well known that the set partitioning problem is NP-hard (Garey and Johnson 1979). Line-cell conversion includes cell formation which is NP-hard. Therefore, line-cell conversion is NP-hard. □

2.5. Properties of cell formation and cell loading in line-cell conversion

The line-cell conversion with the FCFS rule is in effect a two-stage decision process. The first step is the cell

formation shown in Figure 1, and the decision variable X_{ij} decides the result of cell formation. The second step is the cell loading shown in Figure 2, and the decision variable Z_{mjk} decides the result of cell loading.

Property 1. TTPT and TLH are in independent of the order of workers in any cell.

Explanation. Consider $C_j\{1, 2, \ldots, n\}$ denotes cell j in which n workers are assigned, obviously $X_{ij} = 1\ \forall\ i(i \in C_j)$, that is to say that X_{ij} is independent of the order of n workers. So, according to Equations (3)–(5), if other conditions remain unchanged, SC_m, FC_m and FCB_m are not influenced by the order of workers assigned to each cell. Therefore, according to Equations (6) and (7), TTPT and TLH are independent of the order of workers assigned to each cell. □

Cell loading is the batch assignment to cells after the cell formation. Without given scheduling rule, cell load is a scheduling problem and a NP-hard problem. Therefore, line-cell formation is a more complex NP-hard problem and consists of two NP-hard problems. Yin *et al*. (2011) have proved that even a simple cell loading problem is NP-hard. For simplicity and without loss of generality, we consider using the classical types of scheduling rule applied in many companies, i.e. FCFS.

Property 2. During cell loading with the FCFS rule, TTPT and TLH are dependent on the sequence of the cells.

Explanation. Consider $\{C_1, C_2, \ldots, C_J\}$ be a sequence of J formatted cells. If there are empty cells which are not assigned to any batch, an arriving product batch is assigned to the empty cell with the smallest cell number. When all cells are occupied, a product batch is assigned to the cell with the earliest finish time as shown in Figure 2. Given the sequence of batches and the FCFS rule, the changing sequence numbers of the J cells will produce the different Z_{mjk}s that influence the values of SC_m, FC_m and FCB_m (see Equations (3)–(5)). So, TTPT and TLH are dependent on the sequence numbers of the formatted cells (see Equations (6) and (7)). □

3. Combining local search into NSGA-II for multi-objective line-cell conversion

Theorem 1 means that the model described in this article is in general computationally intractable. For small size problems with no more than eight workers, an enumeration algorithm can be used to obtain the non-dominated solutions (i.e. exact Pareto-optimal solutions). For large size problems, however, there is no efficient algorithm for solving the problem to optimality, unless $P = \text{NP}$.

Theorem 2. The Pareto-optimal front of the multi-objective line-cell conversion with the FCFS rule is non-convex.

Proof. The Pareto-optimal front of line-cell conversion with six workers with the FCFS rule is shown in Figure 3. From the graphs of non-dominated solutions in Figure 3, it is easy to observe that the Pareto-optimal front of the multi-objective line-cell conversion is non-convex. □

Since the multi-objective line-cell conversion problem has a non-convex Pareto-optimal or non-dominated front (see Theorem 2), converting multi-objective problem into a single-objective problem by linear weighted approach cannot be applied to find Pareto-optimal solutions (Goicoechea *et al*. 1982). On the contrary, the multi-objective evolutionary algorithms are getting immense popularity (Li and Huang 2009), and the well-known NSGA-II proposed by Deb *et al*. (2002) is widely used, because it provides excellent results as compared with other multi-objective GAs.

However, NSGA-II is one typical type of GAs. Even if GAs usually can find the area of good fitness quite easily, its completely probabilistic search strategy may cause time-consuming and inaccurate in finding the global optimal solutions (Hakimi-Asiabar *et al*. 2009, Chaariabc *et al*. 2011). For example, when a point is close to the Pareto-optimal front, the size of proper descent/ascent cone is extremely narrow and there may exist a low probability of improving the objective functions (Brown and Smith 2005). Therefore, it can be concluded that convergence rate and diversity of GAs need to be improved.

In general, exploration and exploitation are the two main goals of a heuristics algorithm. Exploration allows extensive search to determine the part of the solution space that has a higher possibility of

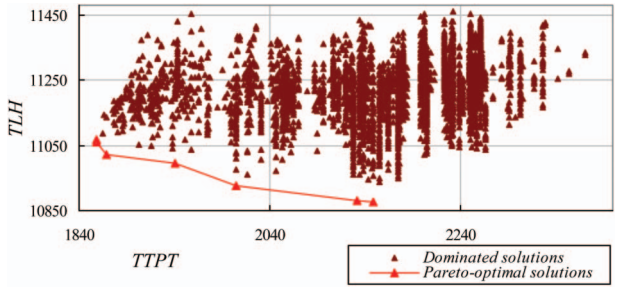

Figure 3. Pareto-optimal front of the example with six workers.

containing the global optimum and can be obtained via population-based heuristics such as GA, ant colony optimisation (ACO) and particle swarm optimisation (PSO) (Mohammadia *et al*. 2011), while exploitation refines the search and focuses on a special part of the space and would be obtained by local search heuristics, simulated annealing (SA) and tabu search (TS; Moslehi and Mahnam 2011). Combining a local search mechanics into GAs has complementary advantages and usually be used to overcome the weaknesses of GAs. A role of the local search is to enhance the intensification process in the genetic search (Arroyo and Armentano 2005, Grosan and Abraham 2007), which has been reported with good performance for hybridisation of a GA.

In this research, according to the property of line-cell conversion, an algorithm of combining local search with NSGA-II is proposed to obtain better non-dominated solutions of the multi-objective line-cell conversion problem. Local search is applied to enhance the exploitation capacity after generating offspring population (Q) of NSGA-II. The algorithm of combining local search with NSGA-II is described as follows.

3.1. Chromosome coding

The chromosome with n workers is represented by a sequence vector with $2n - 1$ integer elements, where the element presents the worker number if it is not more than n; otherwise, it presents the separating character. In addition, there are two fixed separating characters in the start and end locations of the vector. If there exists at least one worker between two separating characters, a cell that contains the worker(s) is constructed.

Such chromosome coding determines not only the number of the cells, the sequence of the cells and the workers assigned to each cell, but also the batch assignment to cells. For example, consider the chromosome as '8 1 7 5 3 9 6 2 4'. In the chromosome, 8, 7, 9 and 6 are the separating characters, so 3 cells are set up. The first cell contains worker 1, the second cell contains worker 5 and worker 3, and the third cell contains worker 2 and worker 4. Obviously, for the chromosome, the sequence of the cells is known. Therefore, given the product batches the FCFS rule, according to Property 2, the batch assignment with the FCFS rule to cells is also represented in the chromosome.

3.2. Evaluation of solutions

A solution for the multi-objective line-cell conversion is computed by the values of two objectives of the TTPT (see Equation (6)) and the TLH (see Equation (7)). Its fitness is evaluated by the non-domination rank and crowding distance of NSGA-II (Deb *et al*. 2002).

3.3. Selection, crossover and mutation

The individuals are selected by using a binary tournament selection (Beyer and Deb 2001). Use the order crossover (OX) operator proposed by Davis (1985). It is implemented by determining two random crossover points. Mutation is the swapping of two unique elements in the solution sequence.

3.4. The strategy of neighbourhood

If neighbourhood is produced by swapping two unique elements in the sequence, then for the solution x with n elements there are $n(n-1)/2$ neighbourhoods. However, to the rule of chromosome coding as described in Section 3.1, it is easy to know that nearly half of the elements in the sequence are separating characters. Therefore, if the two elements are separating characters, then the neighbourhood generated is the same as the original solution. In addition, according to Property 1, it is easy to know that if the two elements are in a cell, then the neighbourhood generated is also the same as the original solution. To protect the diversity of neighbourhood, a new strategy of neighbourhood is proposed as follows.

Definition 1. For the solution x of line-cell conversion, its neighbourhood is defined: worker i assigned in a cell is swapped with any element which is not in the cell.

Example. Consider the solution x with five workers as '8 1 7 5 3 9 6 2 4', where elements labelled 8, 7, 9 and 6 represent separating characters, and elements 1, 5, 3, 2 and 4 represent workers. Worker 1 can be swapped with any element, and worker 5 can be swapped with any element except worker 3.

According to Definition 1, for the solution with n elements, there is less than $n(n-1)/2 - \{[(n-1)/2][(n-1)/2 - 1]\}/2$ neighbourhoods.

Based on the strategy of neighbourhood, a local search is applied to enhance the exploitation capacity after generating offspring population (Q) of NSGA-II. Some good solutions in offspring population are selected to produce neighbourhoods, and the generated neighbourhood survives if it is not dominated by non-dominated set of Q. However, combining local search into NSGA-II may increase the calculation time. Therefore, only the solutions near the optimum sets (with smaller number in the population Q) will be permitted to generate their neighbourhoods. Local

population (L) generated by offspring population (Q) can be described in detail as follows:

3.5. Description of the algorithm of combining local search with NSGA-II

The algorithm of combining local search with NSGA-II for the multi-objective line-cell conversion problem can be expressed as follows:

Input: offspring population Q.
Output: local population L.
(1) Set L = non-dominated solutions of Q;
(2) Select the solutions in the first ff fronts of Q to construct the selected set S;
(3) For each $x_k \in S$ do
  Generate $B_{\max}$ neighbourhoods of x_k, $N(x_k)$.
  For each $y \in N(x_k)$ do
   If y is not dominated by L then
    Set $L = L \cup y$.
(4) Output L.

Input: P (a set of initial solutions).
  N_{iter} (maximum iteration of unchanging non-dominated front).
Output: set of non-dominated solutions.
(1) Initialisation. Set $N = 0$ (iteration of unchanging non-dominated front), sort P based on non-domination and get non-dominated solutions of P, nd(P).
(2) Generate the offspring population (Q) by P in tournament selection, OX crossover and mutation of swapping of two unique elements.
(3) Generate the local population (L) by Q using local search.
(4) Combine P, Q and L into new population C and sort C based on non-domination.
  Initialise $C = P$;
  For each individual $q \in Q$ do
   If $q \notin P$ then
    Add q into C;
  For each individual $l \in L$ do
   If $l \notin P$ then
    Add l into C.
(5) Produce new P by using non-dominated individual in C and get non-dominated solutions of new P, nd(NP).
(6) Judge if to output result:
  If ($N = N_{\text{iter}}$) then
   Go to step (7);
  Else
  If nd(NP) = nd(P) then
   $N = N + 1$;
  Else
   $N = 0$;
   nd(P) = nd(NP);
  Return to step (2).
(7) Output non-dominated solutions of P.

In step (3), local search is used to enhance the intensification process in NSGA-II as described in Section 3.4. Step (4) ensures that there are no same solutions in population C for protecting the diversity of population. From the description of the algorithm, it can be easily observed that local population (L) produced by local search increases the chance of obtaining better non-dominated solutions of the multi-objective line-cell conversion problem.

4. Computational experiments

4.1. Test instances

To test the proposed algorithm, the following simulation examples are used. They are generated by simulating line-cell conversion applications investigated (e.g. Canon and Fujitsu). The experimental data are described in Tables 1–5; they show the

Table 1. The parameters of line-cell conversion example.

Factor	Value
Product types	5
Batch size	N(50,5)
SCP_n	1.0
T_n	1.8
η_i	10
ε_i	N(0.2,0.05)

Note: N(50,5): Normal distribution ($\mu = 50$, $\sigma = 5$)

Table 2. The data distribution of worker's level of skill (β_{ni}).

Product type				
1	2	3	4	5
N(1,0.05)	N(1.05,0.05)	N(1.1,0.05)	N(1.15,0.05)	N(1.2,0.05)

Table 3. The data of worker's level of skill (β_{ni}).

Worker/product	1	2	3	4	5
1	0.92	0.96	1.04	1.09	1.2
2	0.95	0.97	1.09	1.12	1.18
3	0.99	1.01	1.05	1.09	1.21
4	1.03	1.07	1.09	1.12	1.25
5	0.96	1.02	1.05	1.1	1.18
6	1.01	1.1	1.1	1.15	1.23
7	1.04	1.07	1.09	1.17	1.24
8	0.98	1.02	1.1	1.11	1.2
9	0.97	1.03	1.12	1.19	1.26
10	0.98	1.06	1.13	1.18	1.28
11	0.95	1.04	1.03	1.14	1.19
12	0.98	1.07	1.07	1.15	1.15
13	0.99	0.95	1.11	1.17	1.1
14	1.01	1.1	1.05	1.13	1.18
15	1.04	1.1	1.05	1.15	1.11
16	0.99	0.97	1.08	1.11	1.22
17	1.04	1.01	1.11	1.15	1.24
18	0.93	1.06	1.07	1.13	1.14
19	0.96	0.98	1.12	1.14	1.21
20	1.08	1.04	1.09	1.11	1.13

Table 4. The coefficient of influencing level of skill to multiple stations for workers (ε_i)

Worker	1	2	3	4	5	6	7	8	9	10
ε_i	0.18	0.19	0.2	0.21	0.2	0.2	0.2	0.22	0.19	0.19
Worker	11	12	13	14	15	16	17	18	19	20
ε_i	0.18	0.23	0.24	0.22	0.16	0.24	0.18	0.18	0.21	0.18

Table 5. The data of batches.

Batch number	Product type	Batch size (B_m)
1	3	55
2	5	53
3	3	54
4	4	49
5	1	49
6	4	55
7	1	54
8	2	48
9	2	48
10	3	48
11	2	46
12	4	58
13	3	48
14	4	52
15	5	48
16	5	51
17	1	54
18	4	57
19	2	54
20	5	49
21	1	53
22	3	46
23	4	45
24	5	46
25	2	45
26	3	44
27	1	53
28	4	47
29	2	53
30	3	52

parameters, the data distribution and the detail data of level of skill of workers, the coefficient of influencing level of skill to multiple stations for workers and data of batches used in the test, respectively.

From Tables 1–5, it can be observed that five types of product are manufactured, the lot size of each batch is $N(50,5)$ and the ability of workers is different with multiple stations and $N(0.2,0.05)$.

For the instance with W workers, the following data set from Tables 1–5 are used: the entire Table 1, the first W rows of Tables 3 and 4, and the entire Table 5.

4.2. Parameter setting

The crossover probability (P_c) and the mutation probability (P_m) are fixed to 0.5 and 0.9, respectively. The first ff (is set to 5) fronts of non-dominated solutions in offspring population are selected to produce neighbourhoods. The maximum number (B_{max}) of neighbourhoods of a solution is 20.

4.3. Hardware and software specifications

The proposed algorithm, NSGA-II and the numeration algorithm were coded in C# and executed on an Intel Core(TM) 2 processor at 3.0 GHz under Windows XP using 992 MB of RAM.

4.4. Computational results

By repeatedly running the algorithm 100 times to solve the instances with five and six workers, and performances comparative results to NSGA-II are shown in Tables 6 and 7, respectively.

The larger Ac Rate and Av RNI, the better the performance of non-dominated solutions produced by algorithm; the less Av D_{av} and Av D_{max}, the closer the non-dominated front from the algorithm to the true non-dominated front. For example, if Ac Rate is equal to 100%, then Av RNI, Av D_{av} and Av D_{max} will be 1, 0 and 0, respectively, which means each non-dominated front produced by the algorithm in 100 times is the same as the result from the enumeration algorithm.

From Tables 6 and 7, it can be observed that for the small-scale problems of multi-objective line-cell conversion, the performance of our proposed algorithm is much better than that of NSGA-II. For example, for the instance with six workers, when $N = 60$, $n = 90$, each non-dominated front produced by our proposed algorithm in 100 times is the same as the result from the enumeration algorithm, however, there are only 51 non-dominated front produced by NSGA-II which is the same as the result from the enumeration algorithm.

In addition, the proposed algorithm is used to solve larger instances with 10, 15 and 20 workers, respectively. The performance comparative results to NSGA-II of the larger instances are shown in Table 8. True non-dominated solutions are not known, the performance is evaluated with respect to the reference set R, which consists of the non-dominated solutions

Table 6. Performance comparative results to NSGA-II for the instance with five workers.

		NSGA-II					The proposed algorithm				
N	n	Ac Rate (%)	Av RNI	Av D_{av}	Av D_{max}	Av Time	Ac Rate (%)	Av RNI	Av D_{av}	Av D_{max}	Av Time
6	50	12	0.74	0.039	0.22	0.17	96	0.994	0.00008	0.00056	0.60
	65	25	0.82	0.022	0.13	0.19	99	0.999	0.00002	0.00014	0.72
	80	48	0.89	0.013	0.087	0.20	100	1	0	0	0.76
20	50	54	0.92	0.006	0.039	0.21	100	1	0	0	1.11
	65	73	0.85	0.001	0.01	0.27	100	1	0	0	1.21
	80	89	0.98	0.0002	0.002	0.39	100	1	0	0	1.45
40	50	72	0.95	0.0007	0.004	0.43	100	1	0	0	2.12
	65	85	0.98	0.0003	0.002	0.45	100	1	0	0	2.34
	80	100	1	0	0	0.48	100	1	0	0	2.74

Notes: N: iteration of unchanging non-dominated front. n: population size of the algorithm. Ac Rate: accuracy rate of non-dominated front produced by the algorithm in 100 times. Av: average. RNI (ratio of non-dominated individuals) proposed by Tan *et al*. (2001) defines the proportion of the number of efficient solutions produced by the algorithm to the number of true Pareto-optimal solutions. D_{av} is the average distance from a solution to its closest solution in ND (non-dominated set), and D_{max} is the maximum of the minimum distance from a solution to any solution in ND (see Czyak and Jaskiewicz 1998, Ulungu *et al*. 1998, Arroyo and Armentano, 2005). Av Time: average computational time. The unit of Av Time: seconds.

Table 7. Performance comparative results to NSGA-II for the instance with six workers.

		NSGA-II					The proposed algorithm				
N	n	Ac Rate (%)	Av RNI	Av D_{av}	Av D_{max}	Av Time	Ac Rate (%)	Av RNI	Av D_{av}	Av D_{max}	Av Time
40	80	26	0.89	0.013	0.098	1.05	96	0.996	0.00048	0.00444	3.58
	90	46	0.93	0.007	0.062	1.76	96	0.996	0.00048	0.00444	4.47
	100	47	0.93	0.007	0.061	1.92	98	0.998	0.00024	0.0022	4.98
60	80	44	0.93	0.007	0.063	2.43	99	0.987	0.00018	0.00166	5.39
	90	51	0.93	0.006	0.053	2.56	100	1	0	0	5.99
	100	53	0.95	0.006	0.050	2.64	100	1	0	0	6.33
80	80	61	0.96	0.005	0.043	2.49	100	1	0	0	6.23
	90	71	0.97	0.003	0.031	2.95	100	1	0	0	6.78
	100	73	0.97	0.003	0.029	3.12	100	1	0	0	7.01

Table 8. Performance comparative results to NSGA-II for the larger instances.

	NSGA-II					The proposed algorithm				
W	$\|R\|$	Av RNI	Av D_{av}	Av D_{max}	Av Time	$\|R\|$	Av RNI	Av D_{av}	Av D_{max}	Av Time
10	17	0.24	0.096	0.378	9.73	17	0.689	0.02	0.17	20.5
15	19	0.08	0.116	0.38	15.9	16	0.14	0.08	0.23	39.8
20	33	0.01	0.183	0.31	22.9	34	0.07	0.059	0.16	54.2

Note: $|R|$: number of non-dominated solutions.

generated by running the algorithm 100 times for the instance.

Table 8 states each problem size (W), the number of non-dominated solutions ($|R|$), Av RNI, Av D_{av}, Av D_{max} and average computational time of the two algorithms with $N(100)$ and $n(100)$.

From Table 8, it can be concluded that, for the large-scale problems of multi-objective line-cell conversion, the performance of our proposed algorithm is also much better than that of NSGA-II. For example, for the instance with 10 workers, the Av RNI (0.689) of our proposed algorithm is much larger than that (0.24) of NSGA-II, and Av D_{av} (0.02) and Av D_{max} (0.17) are smaller than those (0.096 and 0.378) of NSGA-II, respectively. The other instances are in the similar way.

Running more times (T) to combine the all non-dominated solutions has the probability of producing better non-dominated solutions. Figure 4 shows the relationship between non-dominated front and running times (T) of NSGA-II for the instance with 10 workers. Figure 5 shows the relationship between non-dominated front and running times (T) of our proposed algorithm for the instance with 10 workers.

From Figures 4 and 5, it can be easily observed that better non-dominated solutions can be obtained with the increasing of running times (T). Moreover, it can be observed that the convergence of the proposed algorithm is better than that of NSGA-II by comparing Figures 4 and 5.

The two algorithms with $T(10)$, $N(100)$ and $n(100)$ are used to evaluate the performances of the proposed algorithm for the larger instances, and the performance comparative results to NSGA-II are shown in Table 9.

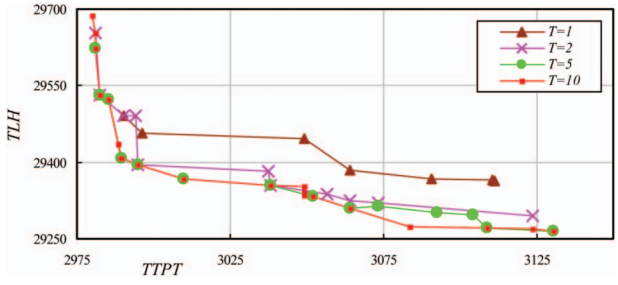

Figure 4. Current non-dominated front of different Ts of NSGA-II.

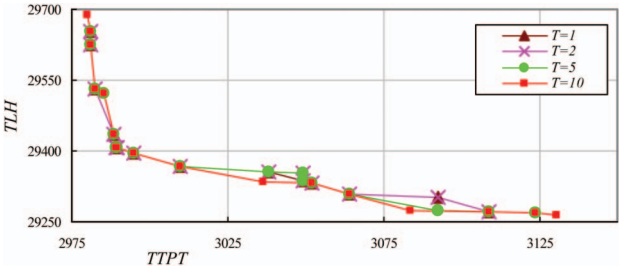

Figure 5. Current non-dominated front of different Ts of our proposed algorithm.

From Table 9, it can be concluded that for the large-scale problems of multi-objective line-cell conversion, when $N = 100$, $n = 100$ and $T = 10$, the performance of our proposed algorithm is much better than that of NSGA-II. For example, for the instance with 15 workers, the Av RNI (0.52) of our proposed algorithm is larger than that (0.28) of NSGA-II, and Av D_{av} (0.017) and Av D_{max} (0.078) are smaller than those (0.026 and 0.11) of NSGA-II, respectively. The other instances are in the similar way.

All comparison results show that the performance (except computational time) of our proposed algorithm is much better than that of NSGA-II. Moreover, the increasing of computational time is acceptable.

5. Conclusions

Our contributions in this article are as follows. Firstly, we simplified the model developed by Kaku *et al.* (2009) into a simple case in which an assembly line is converted to a cell system and proved that is NP-hard and the Pareto-optimal front is non-convex. Secondly, owing to the non-convex Pareto-optimal front and the weakness of generic algorithm, an approach of combining local search into NSGA-II is proposed to obtain better non-dominated solutions. According to the property of line-cell conversion, several implement operators of NSGA-II and local search are modified. Compared with the results obtained by an enumeration algorithm and NSGA-II, our proposed algorithm works well.

The line-cell conversion is a real working problem in Japan industry, and there are still a lot of work should be performed. For example, we only consider the line-cell conversion problem with FCFS rule, and a comparison study of scheduling rules in line-cell conversion problem will be discussed in the future. In addition, we will consider the more complex and practical situations: the line-cell conversion where workers cannot operate all the tasks in assembly line, and the number of assembly tasks is different to the different product types; cross-training; human and psychology factors; saving manpower by line-cell conversion, and so on.

Table 9. Performance comparative results to NSGA-II for the larger instances with $T(10)$.

	NSGA-II					The proposed algorithm				
W	$\|R\|$	Av RNI	Av D_{av}	Av D_{max}	Av Time	$\|R\|$	Av RNI	Av D_{av}	Av D_{max}	Av Time
10	16	0.85	0.013	0.10	98	17	0.96	0.0028	0.047	198
15	20	0.28	0.026	0.11	160	16	0.53	0.017	0.078	401
20	32	0.28	0.034	0.15	240	34	0.35	0.024	0.10	513

Acknowledgements

This research has been supported by the National Natural Science Foundation of China (71021061 and 70971019), the Fundamental Research Funds for the Central Universities (N110404021), the China Postdoctoral Science Foundation (2011M500827), the Startup Foundation for Doctors of Liaoning Province (20101039) and the project of Sanqin Scholars of Shaanxi Province.

References

Arroyo, J.E.C. and Armentano, V.A., 2005. Genetic local search for multi-objective flowshop scheduling problems. *European Journal of Operational Research*, 167 (3), 717–738.

Beyer, H.G. and Deb, K., 2001. On self-adaptive features in real-parameter evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 5 (3), 250–270.

Brown, M. and Smith, R.E., 2005. Directed multi-objective optimisation. *International Journal of Computers Systems and Signals*, 6 (1), 3–17.

Chaariabc, T., *et al*., 2011. A genetic algorithm for robust hybrid flow shop scheduling. *International Journal of Computer Integrated Manufacturing*, 24 (9), 821–833.

Czyak, P. and Jaskiewicz, A., 1998. Pareto simulated annealing – a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multi-Criteria Decision Making*, 6 (7), 34–47.

Davis, L., 1985. Applying adaptive algorithms to epistatic domains. *Proceedings of International Joint Conference on Artificial Intelligence*, 234–243.

Deb, K., *et al*., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions Evolutionary Computation*, 6 (2), 182–197.

Garey, M.R. and Johnson, D.S., 1979. *Computers and intractability: a guide to the theory of NP-completeness*. New York: W.H. Freeman.

Goicoechea, A., Hansen, D.R., and Duckstein, L., 1982. *Multiobjective decision analysis with engineering and business applications*. New York: Wiley.

Grosan, C. and Abraham, A., 2007. Hybrid evolutionary algorithms methodologies architectures and review's. *Studies in Computational Intelligence*, 75, 1–17.

Hakimi-Asiabar, M., Ghodsypour, S.H., and Kerachian, R., 2009. Multi-objective genetic local search algorithm using Kohonen's neural map. *Computers & Industrial Engineering*, 56 (4), 1566–1576.

Kaku, I., *et al*., 2008. A mathematical model for converting conveyor assembly line to cellular manufacturing. *International Journal of Industrial Engineering and Management Science*, 7 (2), 160–170.

Kaku, I., *et al*., 2009. Modeling and numerical analysis of line-cell conversion problems. *International Journal of Production Research*, 47 (8), 2055–2078.

Li, L. and Huang, G.Q., 2009. Multiobjective evolutionary optimisation for adaptive product family design. *International Journal of Computer Integrated Manufacturing*, 22 (4), 299–314.

Liu, C.G., *et al*., 2010. *Seru seisan* – an innovation of the production management mode in Japan. *Asian Journal of Technology Innovation*, 18 (2), 89–113.

Mohammadia, M., Fatemi, G.S.M.T., and Jafaric, N., 2011. A genetic algorithm for simultaneous lotsizing and sequencing of the permutation flow shops with sequence-dependent setups. *International Journal of Computer Integrated Manufacturing*, 24 (1), 87–93.

Moslehi, G., and Mahnam, M., 2011. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129 (1), 14–22.

Stecke, K.E., *et al*., 2012. *Seru*: the organizational extension of JIT for a super-talent factory. *International Journal of Strategic Decision Sciences*, 3 (1), 105–118.

Tan, K.C., Lee, T.H., and Khor, E.F., 2001. Evolutionary algorithm with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 5 (6), 565–588.

Ulungu, E.L., Teghem, J., and Ost, C., 1998. Efficiency of interactive multi-objective simulated annealing through a case study. *Journal of the Operational Research Society*, 49 (10), 1044–1050.

Yin, Y., Stecke, K.E., and Kaku, I., 2008. The evolution of *seru* production systems throughout Canon. *Operations Management Education Review*, 2, 27–40.

Yin, Y., Liu, C., and Kaku, I., 2011. Some underlying mathematical definitions and principles for cellular manufacturing. Working paper submitted to *Asia–Pacific Journal of Operational Research*.

Yin, Y., *et al*., 2012. *Integrating lean and agile production paradigms in a highly volatile environment with seru production systems*: *Sony and Canon case studies*. Working paper, Yamagata University.

Yu, Y., *et al*., 2011. Mathematical analysis and solutions for multi-objective line-cell conversion problem. Working paper submitted to *European Journal of Operational Research*.