Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 17 (2014) 82 – 87

www.elsevier.com/locate/procedia

Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems

Efficient Multi-Objective Optimization Method for the Mixed-Model-Line Assembly Line Design Problem

Jonathan Oesterle[a,]*, Lionel Amodeo[b]

[a]*Fraunhofer Institute for Manufacturing Engineering and Automation, 70569 Stuttgart, Germany*
[b]*University of Technology of Troyes, 10000 Troyes, France*

* Corresponding author. Tel.: +49(0)711/970-1199; fax: +49(0)711/970-1009. *E-mail address:* jonathan.oesterle@ipa.fraunhofer.de

Abstract

This paper presents a mathematical model and an adaptation of the Strength Pareto Evolutionary Algorithm II (SPEA2) for the Mixed-Model Assembly Line balancing and equipment selection problem. The SPEA2 was enriched with a task and equipment reassignment procedure and aims at supporting the planners to find better solutions in the earliest phases of a production system planning project.

Selection and peer-review under responsibility of the International Scientific Committee of "The 47th CIRP Conference on Manufacturing Systems" in the person of the Conference Chair Professor Hoda ElMaraghy"

Keywords: Assembly Line Design problem; balancing problem; equipment selection; Mixed Model Line; Multi-objective; SPEA2

1. Introduction

Nowadays, due to the current levels of globalization, competition and deregulation that have engendered a changeable, dynamic and uncertain global market with greater need for flexibility and responsiveness [1], the ability of a company to compete effectively is influenced to a large extent by its capacity to produce an increased number of customer based products in a timely manner [2]. Shorter product life cycles; high flexible, dynamic and efficient production systems are required, engendering an increased complexity in all factory domains. To handle this complexity, methods of Operations Research are often used to support the decision maker to plan flexible and optimal assembly lines. Assembly lines that allow a low cost production, reduced cycle times and accurate quality levels, can be classified into three variants: (i) the Single Model Line, designed to carry out a single product, (ii) the Mixed Model Line, designed to produce similar models of a product in sequence or batch and (iii) the Multi Model Line, designed to produce various similar or different models in large batches. Several standard scientific problems relating to these three variants have been formulated in the literature, such as the optimal process planning, facility layout, line balancing, buffer allocation, equipment selection, etc. [3]. While the Single and Multi Model Line are the least suited production systems for high variety demand scenarios, the Mixed Model Line is better appropriated to respond to these requirements of flexibility and efficiency. This paper deals with the resolution of a multi-objective problem, namely with the line balancing problem and equipment selection problem, also called Assembly Line Design Problem, for a Mixed-Model-Line. While the line balancing problem is related to the decision problem of optimally partitioning or balancing the assembly tasks among stations, the equipment selection problem is associated to the decision problem of optimally selecting the equipment for each assembly task.

In the next section, the basic concepts of multi-objective problems will be presented, followed by a state of the art in the field of the Assembly Line Design Problem, in which the weaknesses of the current available methods will be presented. Our efficient multi-objective optimization method will be presented in the last sections.

Selection and peer-review under responsibility of the International Scientific Committee of "The 47th CIRP Conference on Manufacturing Systems" in the person of the Conference Chair Professor Hoda ElMaraghy"
doi:10.1016/j.procir.2014.01.038

2. State of the Art

2.1. Multi-Objective Optimization

2.1.1. Basic Concepts and Terminology

A multi-objective optimization problem (MOP) is a problem in which at least two objectives need to be simultaneously optimized. In mathematical terms, a MOP can be formulated as follows:

$$min\,\mathrm{f(x)} = [f_1(x),\dots,f_k(x)]^T \quad (1)$$

$$\text{s.t. } g_j(x) \le 0 \qquad j = 1,\dots,m \quad (2)$$

$$h_l(x) = 0 \qquad l = 1,\dots,e \quad (3)$$

Where $k > 1$ denotes the number of objective functions, m is the number of inequality constraints, and e the number of equality constraints.

Due to the multi-objective nature of most real-life problems (e.g. in finance, scheduling, engineering design and medical treatment [4]), MOPs have been a rapidly growing area of research and application. Generally, these objectives are in conflict, implying that by improving one objective, another objective will become worse. MOPs with such conflicting objective will provide many optimal solutions, instead of only one. The reason for the optimality of more than one solution is that no one can be considered to be better than any other with respect to all objectives [5]. These optimal solutions are known as the Pareto-optimal solutions [6]. A solution $x^* \in X$, where X is the feasible region, is defined as either a Pareto-optimal solution or a non-dominated solution, if it does not exist another point, $x \in X$, such as $f(x) \le f(x^*)$ and $f_i(x) \le f_i(x^*)$ for at least one function, where $X = \{x | g_j(x) \le 0 \; and \; h_l(x)\}$ $\forall j = 1,\dots,m$ and $i = 1,\dots,e$. A solution $x^* \in X$ is defined as Weakly Pareto-optimal, if there does not exist another point, $x \in X$, such as $f(x) < f(x^*)$.

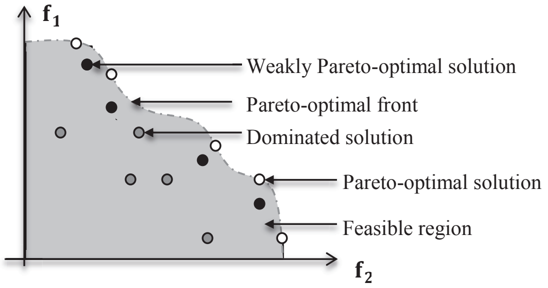


Fig. 1. Illustrative example of Pareto optimality

2.1.2. Approaches to Solve Multi-objective Optimization Problems

There exist many methods and algorithms for solving MOPs. These methods and algorithms can be divided in two categories: (i) classical methods which use direct or gradient-based methods following some mathematical principles and (ii) non-classical methods which follow some natural or physical principles [7,8]. Classical methods mostly attempt to scalarize multiple objectives and perform repeated applications to find a set of Pareto-optimal solutions. In this first category, methods such as the weighted-sum method or scalarization method, ε-Constraints method, Goal-programming, Goal-attainment method and min-max optimization can be found. What has made these methods attractive and why they have been so popular can be attributed to the fact that a wide range of well-studied algorithms for single-objective optimization problem (SOP) can be used. The main criticism of most of these methods is that although they may converge to one Pareto-optimal solution, these methods have to be applied many times in order to get more than one solution. This implies a systematic variation of weight vectors or ε parameters that does not guarantee a good diversity in the set of solutions and thus an inefficient search. In this iterative process, the systematic variation of parameters may also lead to an important CPU time. Moreover, some of these techniques may be sensitive to the shape of the Pareto-optimal front. Indeed, non-convex parts of the Pareto set cannot be reached by optimizing convex combinations of the objective functions [9]. Furthermore, as the solutions mainly depend on parameters such as, weights and upper/lower bounds, these methods also require certain knowledge in order to find Pareto-optimal solutions. Mainly due to these reasons the Multi-Objective Evolutionary Algorithms (MOEA), that stand for a class of stochastic optimization methods, have risen up. Schaffer [10] published the earliest work in the field of MOEA. He proposed a Vector Evaluated Genetic Algorithm (VEGA) based on the traditional Genetic Algorithm by using a modified selection. Since this first publication, the development of MOEA has successfully evolved, producing better and more efficient algorithms, due to in some way the incorporation of the elitism concept, which ensures that the number of non-dominating individuals in the population increases. According to their performances and characteristics, the MOEA can be classified in the following two groups: (i) First Generation, where the Multi-Objective Genetic Algorithm (MOGA), the Niched-Pareto Genetic Algorithm (NPGA) and the Non-dominated Sorting Genetic Algorithm (NSGA) can be found, and (ii) Second Generation, where the Strength Pareto Evolutionary Algorithm (SPEA), SPEA2 [11], the Memetic Pareto Achieved Evolution Strategy (M-PAES), the Pareto Envelope-based Selection Algorithm (PESA), PESA-II and the NSGA-II can be found.

Two major problems must be addressed when an evolutionary algorithm is applied to solve MOP: (i) minimizing the distance to the optimal front and (ii) maximizing the diversity of the generated solutions. In this context, two fundamental issues have to be taken into consideration: (i) the mating selection and (ii) the environmental selection. The first issue is related to the question of how to guide the search towards the Pareto-optimal front, while the second deals with the question of which individuals should be kept in the evolution process. The general concept, common to all these algorithms, is shown in Fig. 2.

First an initial population, representing the starting point of the evolution process, is created at random (or according to a predefined scheme such as heuristics). In the fitness evaluation step, the fitness - reflecting the quality of a solution - is attributed. Afterwards, a binary tournament is normally used for the mating selection process. Here, the mating pool is filled up by individuals that have the best fitness values during the binary tournament selection. In the fourth step, a certain number of parents are selected and a predefined number of children are created by combining parts of the parents. In opposition, the mutation process modifies individuals by changing small parts. Finally, in the environmental selection, individuals of the population and the modified matting pool form the new population [12].

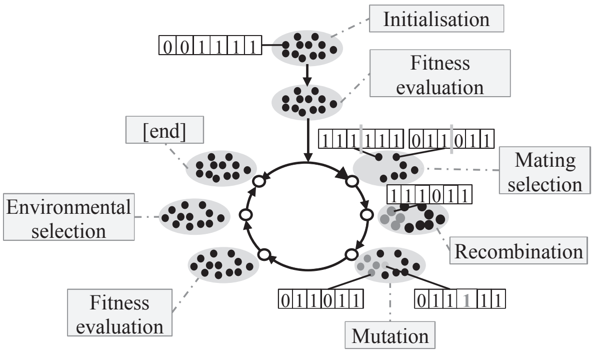


Fig. 2. Outline of a general evolutionary algorithm

2.2. Line balancing and Equipment Selection

While some multi-objective approaches for assembly line balancing can be found in the literature [13–16], there only exist few studies that address both the assembly line balancing problem and the equipment selection problem. The comprehensive reviews of the assembly line balancing problem are provided by Becker and Scholl [17], Boysen et al. [18], Gosh and Gagnon [19]. The combination of these two problems is also known as Assembly Line Design Problem (ALDP).

Graves and Withney [20] are one of the first who addressed the problem of line balancing and equipment selection together. The aim was to select the equipment and assign the tasks to stations in order to minimize the system costs and the annual fixed costs of workstations and operating. Pinto et al. [21] considered a model in which they combined the line balancing problem with the decision of parallel processes. Bukchin and Tzur [22] addressed a model in which the total equipment costs are minimized for a given cycle time. In their model, each equipment type has individual costs and has influence on the task processing times. Nicosia et al. [23] presented a similar model and solved it with a dynamic programming algorithm. Kim and Park [24] associated the problem previously defined with robotic lines. They assigned tasks, parts and tools to robotic cells, defined as stations, in order to minimize the total number of cells activated.

The methods developed in the previous cited works (branch-and-bounds, dynamic-programming, a strong cutting plane algorithm, etc.) belong all to the classical methods for solving MOP. As depicted in the previous chapter, these methods may only be able to find an optimal solution and not a set of optimal ones. As the various objectives involved in such a MOP may be in conflict, a set of optimal solutions is necessary. Furthermore, in mathematical complexity, the line balancing and equipment assignment problem is NP-hard, since it is easy to reduce them to a bin packing problem which is also NP-hard in the strong sense [25]. This means that an optimal solution is not guaranteed for problems of significant size and thus for most of the real-world problems.

For these reasons, our contribution in this paper will be focused on the development of a resolution method for a ALDP that is independent from the size of the problem that needs to be resolved and that aims at delivering a set of Pareto-solutions that minimizes the total costs and the idle time of the line.

2.3. Problem Description

There are P models of a product and a set of tasks S. For each task, there is a set of available equipment with different task processing times and costs. The task processing times may vary between the models. The problem is to allocate the tasks to workstation in such a way that the idle time for each model and the equipment costs are minimized and the various precedence constraints respected.

The decisions that have to be taken address two related issues: (i) the equipment selection problem, where the equipment for a given task has to be selected and (ii) the balancing problem, where the tasks have to be assigned to the stations.

2.4. Problem Assumptions

The following assumptions are stated to clarify the setting of the addressed problem:

- There is a given set of equipment types for each task. Each type is associated with a specific deterministic processing time and a specific cost. The equipment cost includes the purchasing and operational costs.
- The precedence diagram for each model are known
- Parallel stations and buffer are not allowed
- Material handling, loading and unloading times, such as setup-times are negligible or included in the task processing times
- Common tasks exist between the models and have to be assigned to the same stations

2.5. Notations

The notations of the mathematical model are listed below:

- N denotes the total number of tasks
- P denotes the total number of products
- M denotes the total number of workstations
- E denotes the total number of equipment available pro task
- ST_k denotes the set of tasks in the workstation k
- S denotes the set of tasks in the whole problem
- C_m denotes the cycle time of product m
- $c_{i,l}$ denotes the costs of the equipment l of the task i
- $t_{i,l,m}$ denotes the processing time of the task i of the product m of the equipment l

- $x_{i,l,m,k} = \left\{\begin{matrix} 1\ if\ the\ task\ i\ of\ the\ product\ m\ is \\ assigned\ to\ the\ station\ k\ with \\ the\ equipment\ l \\ 0\ otherwise \end{matrix}\right\}$

2.6. Mathematical Model

Our approach aims at minimizing the idle time for each product over the assembly line and the total equipment costs. Thus, it remains to find the set of Pareto-solutions of $f(x)$, where:

$$\mathrm{f(x)} = [f_1(x), \dots, f_P(x), f_{P+1}(x)\ ]^T \qquad (4)$$

The first P objective functions are related to the minimization of the idle time of a model $m\ \forall m = 1, \dots, P$, with $C = \frac{1}{P}\sum_{m=1}^{P} C_m$. The reader will note that the cycle time of the models should be identical for an efficient mixed model line.

$$min\,\mathrm{f}_m(\mathrm{x}) = \sum_{k=1}^{M}\{C \ - \sum_{l=1}^{E}\sum_{i=1}^{N} x_{i,l,m,k}\cdot t_{i,l,m\}} \qquad (5)$$

If $P + 1 > 3$, a graphical representation of the Pareto-solutions becomes more difficult. In this case, the P objective functions can be replaced by the following objective function, where α_m represents a weight of importance for a given product m ($\sum_{m=1}^{P} \alpha_m$=1):

$$min\,\mathrm{f}_{\prime}(\mathrm{x}) = \sum_{m=1}^{P} \alpha_m f_m \qquad (6)$$

The last objective function is related to the costs of the set of equipment assigned to the workstation.

$$min\,\mathrm{f}_{P+1}(\mathrm{x}) = \sum_{i=1}^{N}\sum_{l=1}^{E}\sum_{m=1}^{P}\sum_{k=1}^{M} x_{i,l,m,k}\cdot c_{i,l} \qquad (7)$$

All these objective functions are subject to the following constraints:

$$\bigcup_{i=1}^{M} ST_i = S \qquad (8)$$

$$ST_i \cap ST_j = \emptyset \qquad \forall (i,j)_{i\neq j} \in S \qquad (9)$$

$$\sum_{l=1}^{E}\sum_{m=1}^{P}\sum_{k=1}^{M} k.\,x_{i,l,m,k} \leq \sum_{l=1}^{E}\sum_{m=1}^{P}\sum_{k'=1}^{M} k'.\,x_{j,l,m,k'} \quad \forall (i,j) \in S \quad (10)$$

$$\sum_{i=1}^{N}\sum_{l=1}^{E} x_{i,l,m,k}\cdot t_{i,l,m} \leq C \qquad \begin{matrix} \forall k = 1,..,M \\ \forall m = 1,..,P \end{matrix} \qquad (11)$$

$$\sum_{l=1}^{E}\sum_{m=1}^{P} x_{i,l,m,k} = 1 \qquad \begin{matrix} \forall k = 1,..,M \\ \forall i = 1,\dots,N \end{matrix} \qquad (12)$$

$$x_{i,l,m,k} \in \{0,1\} \qquad (13)$$

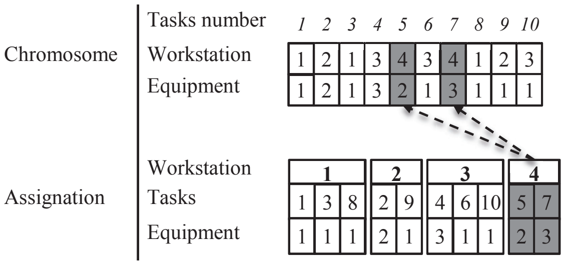
The constraints (8) will assure that all tasks have been assigned to one station, while the constraint (9) will assure that tasks cannot be assigned to more than one station. The constraint (10) will only affect a task $j \in S$ if all predecessors i have already been assigned to previous stations. The constraint (11) prevent to assign tasks to station if the workload of this given station does not exceed the given cycle time of the model. The constraint (12) will guarantee that the same equipment will be allocated to each task i over all the model m. The constraint (12) indicates that the decision variable $x_{i,l,m,k}$ can only take the values 0 or 1.

2.7. Optimization Method

The choice of the SPEA2 was based on the study of Zitzler et. al [11], in which they compared the behavior of various evolutionary approaches based on test functions. The study shows that SPEA2 and NSGA-II displayed the best performance among the algorithms.

As all MOEA, our SPEA2 consists of two main phases, namely: (i) Initialization of the population and (ii) Genetic procedures aiming at the evolution of the population, comprising the selection, crossover, mutation).

In our method, we used the following solution encoding, based on a double chromosome. To each task will be assigned two numbers that respectively represent the workstation to which it will be assigned and the number of the chosen equipment in the available set of equipment. This attribution is done by taking the previously listed constraints into account. The following figure shows an example where the tasks 1,3,8 are affected to station 1, the tasks 2,9 to the station 2, the tasks 4,6,10 to the station 3 and the tasks 5 and 7 to the station 4. In this example, each task of the first station will be performed with the first equipment of the set of task specific equipment.



Tasks number	1	2	3	4	5	6	7	8	9	10
Workstation	1	2	1	3	4	3	4	1	2	3
Equipment	1	2	1	3	2	1	3	1	1	1

Workstation	1			2		3			4	
Tasks	1	3	8	2	9	4	6	10	5	7
Equipment	1	1	1	2	1	3	1	1	2	3

Fig. 3. Chromosome encoding

2.7.1. Initialization of the Population

The initial population is created by using a set of 20 well-known tasks assignment rules, often used in scheduling and bin-packing problems and the equipment are selected randomly. The rest of the initial population P_0 is generated randomly.

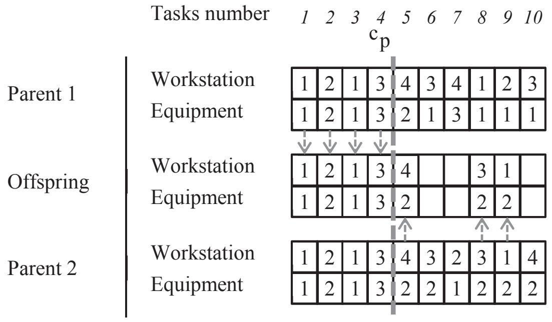
2.7.2. Genetic Procedures

Starting with this initial population and an empty archive, the following steps (see Fig. 4) are performed per iteration until reaching a stopping criterion (predefined computed time, iterations, or when the archive is only composed of non-dominated solutions). The fitness value of each individual in the population is the sum of the strengths of all individual by which it is dominated. The strength $s \in [0,1)$ of an individual is proportional to the number of individuals it dominates. Further information about the fitness assignment can be found here [26]. According to this fitness values, the new archive is composed of the non-dominated solution of the union of the old archive and the population. Here, if the size of the archive exceeds its predefined size, the truncation procedure, procedure whereby the individuals with the minimum distance to other ones are truncated until reaching the predefined size, is computed. In the other case, the archive is filled with dominated solution. In the mating selection, only individuals from the archive are selected via a tournament selection. In the last step, according to their probabilities, the crossover and mutation operations are performed to generate the new population.

```
Input:Population_size, Archive_size, P_crossover, P_mutation
Output:Archive
Population←InitializationPopulation(Population_size)
Archive←{}
While(StopCondition())
FitnessPopulation←FitnessAssignment(Population)
FitnessArchive←FitnessAssignment(Archive)
Archive←ExtractNonDominatedSolution(Population,Archive)
If(size(Archive)>Archive_size)
    FunctionTrucationOperator
ElseIf
    FunctionFillOutWithDominatedSolution(Population,Archive)
End
Selected←BinaryTournament(Archive,Population_size)
Population←Crossover&Mutation(Selected,P_crossover,P_mutation)
Population←truckationprocedure(Population)
End
```

Fig. 4. Main Loop of the SPEA2

Fig. 5 shows an example of the crossover procedure, which has the role to combine pieces of information of different individuals in the population. Two parents from the tournament selection are chosen, and a crossover point c_P is randomly generated. The assignment to workstations and the equipment selection information until c_P is copied from the Parent 1 to the Offspring and the remaining position are copied from the Parent 2. As also represented in Fig. 5, due to the verification of the preservation of the various constraints (cycle time, precedence constraints), some tasks have, after the crossover process, no workstation and equipment assignation. In order to produce feasible individuals, these tasks must be reassigned. The reassignment procedure aims to allocate the tasks to workstations in such a way that all the previous defined constraints are respected. For each task i to be reassigned, the procedure computes the earliest M_E and the latest M_L workstation to which it can be assigned, by taking into account the equipment that has the greatest processing time ($E = E_i$). Each task i is assigned to the first workstation that meets the constraints and the equipment is randomly selected (by taking into account the new size of E). When it is not possible to find a feasible workstation within $[M_E; M_L]$, a new workstation is opened for the task i. The mutation procedure, which randomly disturbs the genetic information, performs small changes in a single parent. Here, a set of tasks will be reassigned and new equipment will be selected. For this mutation procedure, the reassignment procedure described above is also used. The reader will note that the size of this set of tasks should not exceed 10% of the number of tasks N in order to avoid a random search.



	Tasks number	1	2	3	4	5	6	7	8	9	10
Parent 1	Workstation	1	2	1	3	4	3	4	1	2	3
	Equipment	1	2	1	3	2	1	3	1	1	1
Offspring	Workstation	1	2	1	3	4			3	1	
	Equipment	1	2	1	3	2			2	2	
Parent 2	Workstation	1	2	1	3	4	3	2	3	1	4
	Equipment	1	2	1	3	2	2	1	2	2	2

Fig. 5. Example of the crossover procedure

3. Numerical Experiment

In order to illustrate our approach, the following combined precedence diagram and equipment characteristics were used. The cycle time was fixed to 50 for both models. The evolution of the solutions through the iterations is shown Fig. 7. An example of non-dominated solutions for this problem is shown in Table 2. Here, each solution is different and cannot be considered to be better than any other with respect to the three objectives.

Table 1. Equipment properties

	Model 1			**Model 2**					
	Processing times						Costs		
Task	E_1	E_2	E_3	E_1	E_2	E_3	E_1	E_2	E_3
1	30	21	22	23	21	11	101	199	142
2	18	27	16	28	18	16	131	191	148
3	16	26	29	22	15	15	157	189	166
4	13	20	18	29	11	21	164	166	173
5	16	29	21	19	27	24	186	189	126
6	20	28	22	23	26	19	165	145	108
7	29	14	14	12	29	28	106	195	132
8	25	15	30	28	10	23	183	149	101

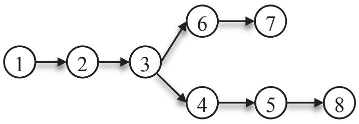


Fig. 6. Combined precedence diagram

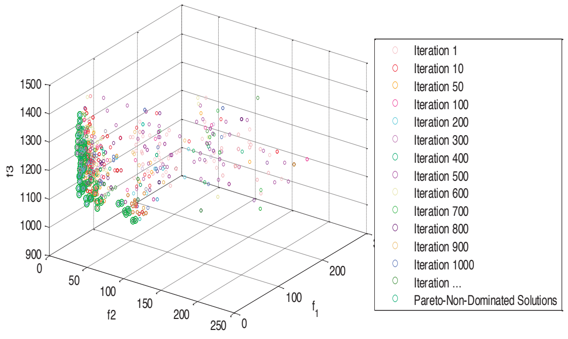


Fig. 7. Evolution of the population through the iterations

Table 2. Example of non-dominated solutions

	Station	Tasks	Equipment	Idle time	Idle Time	Costs
Solution 1	1	{1,2}	{2,2}	13	48	1251
	2	{3,6}	{1,2}			
	3	{4,7}	{2,1}			
	4	{5,8}	{1,3}			
Solution 2	1	{1}	{1}	45	88	1042
	2	{2,3}	{1,3}			
	3	{4,6}	{2,2}			
	4	{5,7}	{3,1}			
	5	{8}	{3}			
Solution 3	1	{1,2}	{2,1}	44	6	1369
	2	{3,6}	{1,2}			
	3	{4,7}	{3,2}			
	4	{5,8}	{1,1}			

4. Conclusion

In this paper, we formulated and solved the Mixed Model Assembly Line Design Problem, composed of a line balancing and equipment selection problem. While the line balancing problem is related to the decision problem of optimally partitioning or balancing the assembly tasks among the stations with respect to some objective(s), the equipment solution is associated to the decision problem of optimally selecting the equipment for each assembly task.

While the research in this area has so far been focused on the resolution of this problem through the utilization of single-objective optimization methods and classical-methods, we proposed a new non-classical method based on an adaptation of the SPEA2 algorithm. This optimization method aims at finding a set of non-dominated solutions that minimize the idle time of various models among an assembly line and minimize the equipment costs. This approach, which is independent from the size of the problem, was enriched with a task and an equipment reassignment procedure. Taking into account the perspectives, it could be interesting to test other genetic procedures (crossover, mutation and also the reassignment procedure) and add some operating conditions of real-world assembly lines, such as tasks assignment restrictions, parallel stations, etc. In the future, we will compare various evolutionary approaches to solve the addressed problem and compare their results, such as their strengths and weaknesses.

References

[1] Youssef, Ayman M. A., ElMaraghy, H., 2006. Assessment of manufacturing systems reconfiguration smoothness. The International Journal of Advanced Manufacturing Technology 30 (1-2), 174-193.

[2] Samy, S.N., ElMaraghy, H., 2010. A model for measuring products assembly complexity. International Journal of Computer Integrated Manufacturing 23 (11), 1015–1027.

[3] Chehade, H., Dolgui, A., Dugardin, F., Makdessian, L., Yalaoui, F., 2012. Multi-Objective Approach for Production Line Equipment Selection. Management and Production Engineering Review 3 (1), 4–17.

[4] Lindroth, P., Patriksson, M., Strömberg, A.-B., 2010. Approximating the Pareto optimal set using a reduced set of objective functions. European Journal of Operational Research 207 (3), 1519–1534.

[5] Potti, S., Chinnasamy, C., 2011. Strength Pareto Evolutionary Algorithm based Multi-Objective Optimization for Shortest Path Routing Problem in Computer Networks. Journal of Computer Science 7 (1), 17–26.

[6] Pareto, V., 1906. Manuale di economia politica. Societa Editrice.

[7] Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. Struct Multidisc Optim 26 (6), 369-395.

[8] Shukla, P.K., Deb, K., 2007. On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. European Journal of Operational Research 181 (3), 1630–1652.

[9] Caramia, M., Dell'Olmo, P., 2008. Multi-objective management in freight logistics: Increasing capacity, service level and safety with optimization algorithms. Springer, London.

[10] Schaffer, J.D. (Ed.), 1985. Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of the 1st International Conference on Genetic Algorithms, 93-100.

[11] Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. Institut für Technische Informatik und Kommunikationsnetze (TIK).

[12] Zitzler, E., Laumanns, M., Bleuler, S., 2004. A tutorial on evolutionary multiobjective optimization, in: , Metaheuristics for Multiobjective Optimisation. Springer, pp. 3–37.

[13] Chehade, H., Yalaoui, F., Amodeo, L., 2011. New multiobjective optimization algorithms for assembly lines design. International Journal of Advanced Operations Management 5 (2), 94–120.

[14] Kara, Y., Ozcan, U., Peker, A., 2007. Balancing and sequencing mixed-model just-in-time U lines with multiple objectives. Proceedings of Applied Mathematics and Computation 184 (2), 566–588.

[15] McMullen, P.R., Rarasewich, P., 2006. Multi-objective assembly line balancing via a modified ant colony optimization technique. International Journal of Production Research 44 (1), 27–42.

[16] Yang, C., Gao, J., Sun, L., 2011. A Multi-objective Genetic Algorithm for Mixed-model Assembly Line Rebalancing. Computers and Industrial Engineering 65 (1), 109–116.

[17] Becker, C., Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research 168 (3), 694–715.

[18] Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: Which model to use when. International Journal of Production Economics 111 (2), 509–528.

[19] Gosh, S., Gagnon, R.J., 1989. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. International Journal of Production Research 27 (4), 637–670.

[20] Graves, S.C., Redfield, C.H., 1988. Equipment selection and task assignment for multiproduct assembly system design. International Journal of Flexible Manufacturing Systems 1 (1), 31–50.

[21] Pinto, P.A., Dannenbring, D.G., Khumawala, B.M., 1981. Branch and bound and heuristic procedures for assembly line balancing with paralleling of stations. International Journal of Production Research 19 (5), 565–576.

[22] Bukchin, J., Tzur, M., 2000. Design of flexible assembly line to minimize equipment cost. IIE Transactions 32 (7), 585–598.

[23] Nicosia, G., Pacciarelli, D., Pacifici, A., 2002. Optimally balancing assembly lines with different workstations. Special Issue devoted to the ALIO-EURO Workshop on Applied Combinatorial Optimization 118 (1–2), 99–113.

[24] Kim, H., Park, S., 1995. A strong cutting plane algorithm for the robotic assembly line balancing problem. International Journal of Production Research 33 (8), 2311–2323.

[25] Gokcen, H., Erdal Erel, 1998. Binary Integer formulation for Mixed-Model Assembly Line Balancing Problem. Computers & Industrial Engineering 34 (2), 451–461.

[26] Zitzler, E., Thiele, L., 1998. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH).

[27] Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary computation 8 (2), 173–195.