Mixed-model assembly line balancing problem with tasks assignment

K Dziki and D Krenczyk
Silesian University of Technology, Faculty of Mechanical Engineering, Konarskiego 18A, 44-100 Gliwice, Poland
E-mail: karol.dziki@polsl.pl

Abstract. Modern production systems are focused on the objectives dictated by customer needs oriented market. The best results are obtained by production systems combining the advantages of mass production with the possibility of implementing product variants corresponding to the needs of an individual customer. An example of the production system that meets those conditions is mixed-model assembly line. The main problem of this solution is to plan the order of operations and assign the operators’ tasks in such a way to obtain the best balance of a given line. In this study, several metaheuristic methods that can be used to solve line balancing and tasks assignment problems are presented. The purpose of this study is to show the advantage of simulated annealing over other methods in solving the described problem. This paper is introduction to future research in scope of mixed-model assembly line optimization focused on employees' effectiveness.

1. Introduction

The main goal of contemporary production systems is to produce as many products as possible at the lowest production cost, with quality level and features demanded by customers. Growing complexity of products is forcing manufacturers to design more complex production lines. For reducing costs generated by overgrown infrastructure, there is a need to design production lines that are able to manufacture more than one type of products with minimal amount of crew. There are two ways to accomplish this task: First, is to prepare line for one variant of product and make changeovers for other variants, and the second one, mostly used in modern production systems, is designed in that way that it is possible to manufacture many variants of products; however, two conditions must be met: products must have the same base (technological processes must be similar at initial stage) and similar processing times. Assembly lines for those types of production are called accordingly multi- and mixed-model assembly lines. This paper is focused on Mixed-Model Assembly Line Balancing Problem (MMALBP) taking into account tasks assignment optimization. First part is presenting the problem origin and difficulties in solving it and main balancing assembly line indicators are presented. Later in this paper, chosen metaheuristics are shortly described. Three of them represent a group of methods based on evolutionary algorithms and the fourth is Simulated Annealing. The last method is based on the phenomenon described in metallurgy. The culmination of this paper are conclusions made from the comparison of described methods in scope of assembly line balancing and task assignment problems in literature and directions of future research are presented, [1, 2].
2. Balancing assembly line with task assignment

Main problems with the optimization of production are most noticeable in the production of large products, such as automotive, where many versions of the product are often associated with the occurrence of the irregularity of operations duration. Despite the use of mixed-models on one line, line optimization is still a challenge. An additional factor is the determination of the minimum number of operators, high volatility affects the unused time available to operators. In the literature [3, 4], very often the optimization of the line is presented in form of simplification and averaging of the task times for all variants and treating the product group as one model with averaged time values similar to the ones used in SALBP. Balancing a simple assembly line focuses on one of two problems: SALBP-1 is used to define number of workstations with fixed number of workers, SALBP-2 determines minimal number of workers necessary to accomplish products on defined number of workstations. The other side of the problem is optimization of models sequencing in a way that allows you to make a production plan at a strictly specified period of time. For MMALBP some specific constrains are considered and their general assumptions are given as follows:

- the precedence relationships among the tasks are known;
- a single product is assembled on a single workstation;
- assembly task times are deterministic;
- an assembly task is performed by a given number of workers;
- only one task on one workstation performed by one worker at the given time;
- the transportation times are ignored;
- the goal is to minimize the total number of station and number of workers and simultaneous efficiency maximization of the line.

Objective functions that are commonly used to evaluate MMALBP solution are the minimization of smoothness index of line balance and maximization of line efficiency index. In case of tasks assignment aspect, additional objective function is used – minimizing number of assigned workers [5-7].

To solve these types of problems, multi-stage processes using decision algorithms are used.

3. Metaheuristics in MMALBP solving

One of heuristic algorithms categories are methods based on phenomena that can be observed in nature, literature refers them as evolutionary algorithms. The evolution of biological structures causes the need to adapt organisms to the prevailing environmental conditions. Such adaptation is necessary for a given species to survive and not be repulsed by others. Patterns used by nature can be successfully translated into methods of solving problems. An optimal solution must be found, it can be the creation of an organism that will survive thanks to the best adaptation to the prevailing conditions or the configuration of the induction system, which will achieve the assumed goals or a structure that will withstand the set loads.

3.1. Genetic algorithms

Basic form of above described approach is optimization strictly based on natural process of selection, heritage and mutation of species. Besides original form presented by John Holland [8], many other variations in literature [9, 10] are presented. Changes result from attempts to apply the algorithm for individual problems and attempts to improve efficiency. In general, the build of the algorithm is made along 7 elements:

**Step 1: Determination of the genetic representation**

Mostly common genetic representation for solution is illustrated by binary vector called in this case “chromosome”. Each "feature" of a given solution has been coded and described in binary system.

**Step 2: Generate of initial population**

Mostly populations are generated randomly, based on input base of features and constrains. If problem for optimization have small regions of feasible set of features for optimal solution, there it is possible to find optimal in first population iteration.

**Step 3: Define objective function for determinate scoring method of solution**
Each sample is representing potential solution that needs to be evaluated. Fitness of sample is evaluated by objective function describing problem.

**Step 4: Create and apply selecting rules**

When fitness index is set, the best one should be chosen. Most popular for this are methods of roulette wheel and tournament. In the first one, the chance of selection is based on sample fitness ratio to population relative fitness value and second depends on comparing randomly chosen samples. One of these rules should be used to determine which samples will pass to the next generation.

**Step 5: Altering “genetic composition” of predecessors**

From all possible solutions to choose from, those with the highest fitness index are chosen. In this step there are two possible operations to proceed with. First one is the crossover of the fittest samples that exchange features (“genes”) between samples and create offspring that creates new area that is closer to optimum. The second one is mutation that randomly change one or more features in the best fitted samples. Mutations are a chance that solution will not be misguided towards the local optimum and the algorithm will point the optimal solution.

**Step 6: Creating new generation**

Based on new features, that give the best fitness, new samples are created. Most common is that “child” replaces “parent” but many variations in this step are presented in literature. Some use tournament method to leave only the best samples, other use each offspring as the starting point of local search.

**Step 7: Algorithm cancelation criteria**

This kind of algorithm is considered complete if one of the following outcomes occurs:

- fixed number of generations are set;
- time intended for task is over;
- in last n-th iterations population fitness is not better than maximal previous one;
- in n-th iteration population gets satisfying fitness.

In GA Method every attempt to change the algorithm in order to avoid blocking in the local optimum causes an increase in the complexity of calculations and the amount of iterations, which increases the computation time and resources.

### 3.2. Imperial Competitive Algorithm (ICA)

The ICA Method is similar to GA considering that, in general concept of the fitness of sample. Instead of “chromosomes” describing organisms, population of possible solutions is represented by countries that compete for domination in specific area. Initial, set populations are divided into smaller areas, where from the nearest “countries”, based on objective function, the best one is chosen and is granted the title of the imperialist. Other countries are “colonies” of the imperialistic country and together create an “empire”. The total strength of the empire is dependent on the strength of single colonies. When all the colonies are incorporated into one empire, the development of the algorithm is terminated. In general form [11], procedure of algorithm is formed by the following steps:

**Step 1: Generate initial population**

Countries base values are random, but considering the core of the problem, there can be constrains that limit possible area of solutions.

**Step 2: Accommodation of the colonies near the imperialist**

In this step, the strongest country in the area provides the possibility of the occurrence of changes affecting the features of the neighbouring countries (colonies), and moving them closer to the imperialist to create new space of possible solutions. The main goal of this step is to find the local optimum in the whole spectrum of solutions.

**Step 3: Updating positions of countries in the empire**

After creating new area of possible solutions in previous step, there is a possibility that altered colony has now better score then the imperialist, so now this country is considered to be an imperialist in the area and it will be altering colonies in the next iteration.

**Step 4: Computing strength of the empire**
Strength of the empire is calculated based on score values of each country. The empires, whose score is mainly comprised of the value of the imperialist, rather than its colonies, are considered to be stronger than other empires whose score is the result of the total value of its colonies.

Step 5: Imperialistic competition
All empires try to take over the colonies of other ones. The outcome of that is the fact that the weakest empires’ colonies are divided between other empires.

Step 6: Randomizing weakest colonies
In each iteration, a “revolution” occurs. The number of the weakest colonies is randomly altered and re-assigned to the appropriate empire once again.

Step 7: Erasing “empty” empires
If an empire loses all its colonies, that means it is the weakest from all the population of empires, which in turn, translates into the fact that it can be erased from the area of possible solutions.

The algorithm is terminated when only one empire is left in the area. The accuracy of this method depends on initial population’s layout in the area of possible solutions, in case of complex problems there is a concern that global optimum may be missed in the process of finding the best possible solution.

3.3. Particle Swarm Optimization (PSO)
Optimization method based on particle swarm behaviour was first presented in 1995 by Kennedy and Eberhart [10]. The method was created on the basis of previous developments of metaheuristics, specifically: artificial life and evolutionary computing. During the studies, some features of those methods were removed, and conclusions were made: more accurate solutions could be found if population of solutions will behave like swarm, and for simplifying calculations and predictions a single individual should be treated as a point (devoid of size and mass). Following features of elements of this method were determined accordingly to Millonas [13] work concerning intelligent swarm:

- Information about swarm is contained in simple structures;
- Precisely defined spatial coordinates and velocity value;
- Quality of particle depends on objective function;
- Objective function for current location and the best score should be remembered within particle;
- Population should not fall into similar answers area;
- Population should keep its behaviour regardless of small environmental changes;
- Population should show opportunistic behaviour;
- The nearest particles should share their memory for cooperative reasons.

To avoid “explosion” effect, the constrain of maximal velocity is often used and then it is added to model of PSO. In PSO model, each particle in subsequent iterations is recalculating its new velocity value and location. New movement vector is calculated based on three vectors:

- Vector aimed at the best score for an individual particle;
- Vector aimed at the best score of the nearest neighbours of a particle;
- Previous velocity vector.

Simplified algorithm path is implemented in the following steps:

Step 1: Population generation
Set number of particles is placed randomly with the random initial velocity.

Step 2: Computation of objective function value on each particle
According to objective function formula, values are calculated and “memorised” by particles.

Step 3: Validation of new values
If calculated values are better than memorised, than particle changes the vector according to PSO rules.

Algorithm is looped until all generated particles meet in the same spot, where coordinates describe optimal solution for a specific problem. In PSO, like in other population based evolutionary metaheuristics, main problem is that algorithm could find false-positive solution – local optimum. In literature
[14] some modifiers to PSO are presented but lead to longer calculations, where increased usage of resources is needed.

4. Simulated Annealing (SA)
In 1983, Scott Kirkpatrick published a paper [15] that describes the progress of the work that resulted in developing optimization that uses annealing phenomenon as method to find solution for defined multi-objective problems [16]. The working principle of Simulated Annealing (SA) operation is based on the representation of the scope of solutions by temperature. Annealing is a metallurgical process that is used to achieve thermodynamic equilibrium to modified material properties to achieve intended values. A simulated annealing algorithm is a well known [17] neighbourhood search approach with probabilistic module for escaping from local optima. Parameter that controls the flow of process is analoical to the temperature that represents the inner state of annealed material. In SA, new solutions are generated by the nearest points and generation mechanism. The process of optimization search with SA could be divided into the following steps:

- **Step 1: Initializing starting state**
  Whole algorithm starts with random initial placement of first points for the validation and determination of the highest temperature.

- **Step 2: Checking neighbourhood solutions**
  For initial point, objective function value is calculated. Score is compared with the score of the nearest point.

- **Step 3: Change of initial points**
  If score of the nearest points is better than the previous one, than the local optimum finding area is moving to the new best score location.

- **Step 4: Equilibrium criterion**
  Checking number of generated points, if it is large enough then algorithm can pass to the next step, if not then a loop on step 2 is made until population will be satisfying.

- **Step 5: Schedule of cooling**
  Initial temperature is decreasing in order to narrow down the area of possible solutions. In every iteration determined in the algorithm linear dependence specifies rate of temperature value drop.

Algorithm is looped until the number of solutions is acceptable small and configuration reaches the freezing point. In multi-objective, real-life, nonlinear problems, that area of possible solutions is described as n-dimensional space. There must be a possibility of temperature change indicator sensitivity alteration. At any stage of SA it is reasonable to spread the range of the search scope in order to ensure whether the area where algorithm finds its optimum is not only the local best value but also one needs to verify it with the use of worse values for further iterations. It is made feasible with the use of simple linear rescaling of acceptable temperature.

5. Comparison of methods
Using above described methods to solve line balancing and task assignment problems, features of the line are represented by elements described in table 1.

| Method                | Problem elements’ representations                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------|
| Genetic algorithms    | Features of assembly line are encoded as “chromosome” parts. First sequence represents possible second sequencing for each model that needs to be produced, second represents possible orders of tasks, next one describes how single tasks should be assigned to workstation to reach the production goal [18]. |
Imperial Competitive Algorithm

Similarly, to the previous method, in this case “features of chromosome” are replaced with “features of nation”. Similarity in problem encoding is that first sequence is describing sequencing, next one possible orders of tasks and the last one is the representation of task assignment.

Particle Swarm Optimization

In PSO components of the coordinates of particle represents line features. Each axis of possible solutions area is representing one feature, like for example models sequencing or orders of assigning tasks to workstations.

Simulated Annealing

The encoding of possible values of solution for each feature in SA is similar to PSO. There is n-dimensional area of possible solutions that is limited by boundaries resulting from the assumptions of the line where each axis is representing one feature [19].

For each of the above-mentioned methods there is a different way to evaluate which conditions give optimal results. In the table 2 evaluation methods for each described method are presented.

| Method                        | Problem goals evaluation method                                                                                                                                                                                                 |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Genetic algorithms            | In each iteration (generation) fitness indicator is calculated. Depending on accepted rules, one or more best samples are chosen. New best scores comprise the base for the narrowing down of the scope of optimum search. |
| Imperial Competitive Algorithm| Same as in GA, in ICA index values are calculated and the best one from area is chosen. Moving worst solutions towards better leads to the narrowing of the scope of optimum search area.                                      |
| Particle Swarm Optimization   | Objective goal is calculated from particle coordinates values. If score is better than previous and/or better then best score of population than velocity vector is changing towards area with possibly better solutions.                        |
| Simulated Annealing           | Objective function is calculated similarly to PSO. Coordinates’ values of examined points in area of possible solutions are used to determine the score of solution. Depending on rules of algorithm, decision about direction of search scope is made. |

Common feature of metaheuristics presented in previous part of this paper is implemented in algorithms attempt to avoid getting a local optimum. Initial population and basic conditions may lead to the unintentional removal of the global optimum from the search area. Table 3 compares tools that are used in the discussed methods to neutralize that effect.

| Method                        | Local minimum avoiding method                                                                                                                                                                                                 |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Genetic algorithms            | Phenomenon of mutation is a tool to make random change on one or more features of offspring.                                                                                                                                       |
| Imperial Competitive Algorithm| Phenomenon of “revolution” is used to insert the new variables to current set for changing course of moving to the best possible solutions area.                                                                                     |
Particle Swarm Optimization
Vectors of particles movement are dependent on resultant vector of:
the best particle score, the best overall score for population and pre-
vious velocity vector. Depending on accepted rules, random changes
on those values can be done, that impact the next calculations.

Simulated Annealing
In basic form of this method there is no random moderator of search
scope driving values.

Basic forms of described methods try to avoid stopping at the local optimum by randomizing one
or several values of computational input conditions of the algorithm. In reviewed literature, there is no
information about possibility to check the rejected possible solutions areas for global optimum. Rand-
dom changes cannot be reversed to return to the worse results for checking other areas. In SA method,
it is possible to make linear change in control parameter value, which allows to choose worse possible
solution in considered scope in order to check the other area [20].

Summarizing comparison of described metaheuristics with SA optimization, it should be specified
that the results of evolutionary methods are very dependent on initial conditions such as the initial
density of the population in the area of possible solutions and random spread of areas in the initial
stages. The way to minimize the effects of rejecting the global minimum from possible solutions area
is to use algorithm mechanisms to stimulate the return to rejected areas through random changes in the
values of the algorithm elements. In real life problems, which are related to assembly lines’ optimiza-
tion methods that cannot achieve the optimal results should not be used. Based on the above-
mentioned observations, adaptation of Simulated Annealing Optimization for this kind of optimization
problems is a good choice, because SA characteristic feature is the ability to search for the global op-
timum in the previously rejected areas of possible solutions if there is a suspicion that the indicated
solution is in local optimum.

6. Conclusions
The results of finding optimal solution with evolutionary, population based metaheuristics are dependent
on two main factors: the initial population points allocation and the direction of progression. With com-
plex, multi-objective problems, there is a chance that the algorithm after entering the local optimum zone
will not be able to get a better result and will give the local optimum found as the best result. In case of
balancing problem solutions and workload optimization in mixed-model assembly lines, there is a need
to choose a method to determine optimal line parameters that can be modified if there is a suspicion that
the result found can be a global optimum. Further research on mixed-model assembly line balancing
problem, as described in this paper, will be associated with previous research on methods for balancing
and optimizing multi-manned assembly with location constraints in order to create a coherent, compre-
hensive solution for the optimization of this type of assembly lines.

7. References
[1] Freitas F and Mourao A 2019 Methodological proposal for an indoor industrial AGV implement-
tation International Journal of Modern Manufacturing Technologies XI(1) 28-35
[2] Hozdic E 2015 Smart factory for industry 4.0: A Review International Journal of Modern Manu-
facturing Technologies VII(1) 28-35
[3] Grzechca W 2014 Linia montażowa z wieloma operatorami na pojedynczej stacji roboczej
Innovations in management and production engineering 1 540-551
[4] Scholl A and Becker C 2006 State-of-the-art exact and heuristic solution procedures for simple
assembly line balancing European Journal of Operational Research 168 666–693
[5] Yilmaz H and Yilmaz M 2016 A multi-manned assembly line balancing problem with classified
teams: a new approach Assembly Automation 36 51-59
[6] Dimitriadis S 2005 Assembly line balancing and group working: A heuristic procedure for
workers’ groups operating on the same product and workstation Computers & Operations
Research 33 2754-2774
[7] Holland J 1984 Genetic Algorithms and Adaptation *Adaptive Control of Ill-Defined Systems* 16 317-333
[8] Prasad M, Genesan K and Suresh R K 2013 An Optimal Balancing of Multiple Assembly Line for A Batch Production Unit *International Journal of Lean Thinking* 4 22-32
[9] Mutlu O, Olcay P and Supciller A 2012 An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II *Computers & Operations Research* 40 418-426
[10] Akpinar S and Bayhan GM 2010 A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints *Engineering Applications of Artificial Intelligence* 24 449-457
[11] Gharavi H, Ardehali M M and Ghanbari-Tichi S 2015 Imperial competitive algorithm optimization of fuzzy multi-objective design of a hybrid green power system with considerations for economics, reliability, and environmental emissions *Renewable Energy* 78 427-437
[12] Kennedy J and Eberhart R 1995 Particle Swarm Optimization *Proceedings of the IEEE International Conference on Neural Networks* 3 pp 1942-1948
[13] Millonas M 1994 Swarms, phase transitions, and collective intelligence *Computational Intelligence: A Dynamic System Perspective* Publisher: IEEE Neural Networks Council Editors: Palaniswami M et al. pp 137-151
[14] HongGuang L and Cong L 2010 An assembly sequence planning approach with a discrete particle swarm optimization algorithm *Int J Adv Manuf Technol* 50 761-770
[15] Kirkpatrick S, Gelatt C and Vecchi M 1983 Optimization by Simulated Annealing *Science, New Series* 220(4598) pp 671-680
[16] Baykasoglu A 2006 Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems *J Intell Manuf* 17 217-232
[17] Dekkers A and Aarts E 1991 Global optimization and simulated annealing *Mathematical Programming* 50 pp 367–393
[18] Aytug H, Khouja M and Verara F 2003 Use of genetic algorithms to solve production and operations management problems: a review *Int. J. Prod. Res.* 41(17) 3955-4009
[19] Bandypadhyay S, Saha S, Maulik U and Deb K 2008 A Simulated Annealing-Based Multi-objective Optimization Algorithm: AMOSA *IEEE Transactions on Evolutionary Computation* 12 269–283
[20] Yang M et al. 2019 An improved genetic simulated annealing algorithm for stochastic two-sided assembly line balancing problem *Int simul model* 18 175-186