Balancing lines with CNC machines: A multi-start ant based heuristic

Mohamed Essafi, Xavier Delorme, Alexandre Dolgui

Balancing lines with CNC machines: A multi-start ant based heuristic

Mohamed Essafi, Xavier Delorme, Alexandre Dolgui *

Center for Industrial Engineering and Computer Science, Ecole des Mines de Saint-Etienne, 158, Cours Fauriel, 42023 Saint-Etienne, France

We are working on a machining line balancing problem involving specific constraints. The studied lines are paced and serial, i.e. a part to be machined passes through a sequence of stations. The stations are equipped with CNC (Computer Numerical Control) machines. A CNC machine is a mono-spindle head machine which can use sequentially different tools and can rotate the part in order to perform different tasks. Such a machine is guided by a computer numerical controller system. As usual with machining lines, this problem is subject to precedence constraints as well as exclusion and inclusion constraints. Moreover, the station workload depends on the sequence in which the tasks are assigned because of set-up times related to the change and displacement of tools, rotation of the part, etc. In addition, accessibility constraints have to be considered. Two types of CNC machines with different characteristics can be used. Several tasks require a particular type of machine. The objective is to assign a given set of tasks, required for part machining, as well as a given set of machines to a sequence of stations while minimizing the total cost of the line. In this paper, a multi-start heuristic is proposed and tested on real life industrial problems.

1. Introduction

The optimization of production systems is an important stage for manufacturers to minimize costs and remain competitive. Transfer line, is a widely used production system in machining environment especially in the automotive industry. They are used for mass production thanks to their characteristics. A transfer line is composed by a set of sequentially arranged workstations. A constant flow of parts is ensured by a transport system. Parts move from station to another and in each station, a set of machining operations is performed on each. Each operation is characterized by an operational time; a set of operations which must be assigned before it (precedence constraints); a set of operations which must be executed on the same workstation (inclusion constraints); a set of operations which cannot be executed on the same workstation (exclusion constraints).

However, in the current economic context, with market volatility and fluctuation in demand, industrials manufacturers need more flexible production systems. Thus, new types of lines were created; i.e. flexible and reconfigurable transfer lines. The flexibility or the reconfigurabilty of a line is obtained through the use of special machines, a developed control system for the line, a specific architecture, etc. [30].

The use of Computer Numerical Control (CNC) machines is a common way to add more flexibility or reconfigurability to a machining line. Such machines are highly automated and use computer programs to define the different tools to use for a specific part. Therefore they correspond to standard and interchangeable units in which a new program can be loaded to change the production. However, these advantages have a cost with respect to productivity and they induce particular constraints to deal with when designing the line. In this paper, we consider a machining line using CNC machines working with mono-spindle heads. So in addition to the usual constraints of transfer line balancing problems, we have to consider accessibility constraints, parallel stations and sequence-dependent set-up times.

According to the position of the part on the machines, accessibility constraints are defined. For a specific part position only a subset of operations can be executed. Parallel stations are used in order to obtain a better balancing. Set-up times are due to various causes, namely: the change and displacement of the tool, rotation of the part, etc. They vary according to the sequence in which the operation is assigned. Our main objective is to minimize the total cost of the line for a given cycle time. This work is developed in collaboration with the enterprise PCI-SCEMM and corresponds to a real industrial problem.

A heuristic based on Ant Colony Optimization (ACO) principles is proposed for the resolution of the considered problem. An industrial case is studied and experimentation results are reported.

---
* Corresponding author. Tel.: +33 477 4201 66; fax: +33 477 4266 66.
E-mail addresses: essafi@emse.fr (M. Essafi), delorme@emse.fr (X. Delorme), dolgui@emse.fr (A. Dolgui).
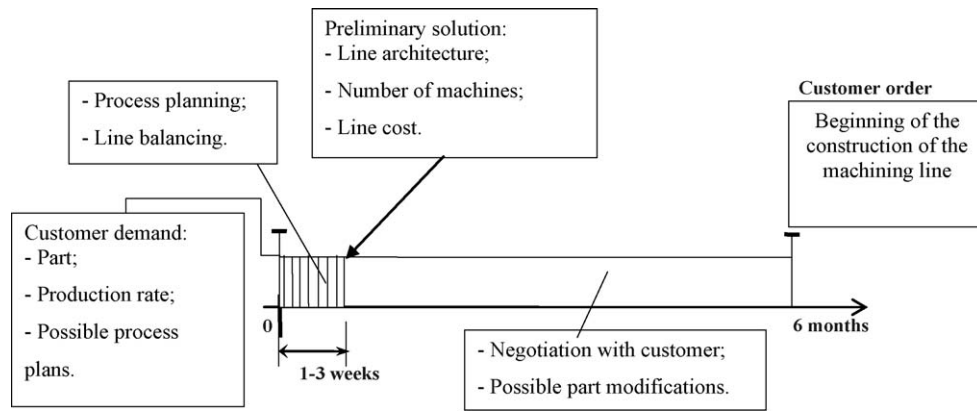
Fig. 1. Negotiation process: the critical phase.

2. Preliminary design of transfer line

Usually for this type of project, the procedure is as follows: a company (client) contacts the transfer line manufacturer. The client gives the parts properties (part plans, characteristics, etc.) and the required output (production rate). Then comes the critical phase: the manufacturer should quickly offer a complete preliminary design solution for the corresponding line: architecture of the line, number of machines, etc. and an approximate line cost. The acceptance of this solution by the client, and consequently the continuation of the negotiation and further development of the project, depends on the quality of this early solution. The temporal progress of the negotiation process and its critical phase are illustrated in Fig. 1.

The manufacturer's objective is to reduce the preliminary design time while minimizing the cost of the potential line. This is decisive, due to high competition among manufacturers in this domain. Moreover, these lines are technically very complex and require huge investments.

If a preliminary solution is more expensive than those of the competitors, then the contract (several hundred million euros) may be lost. If this is cheaper, then the manufacturer increases the chances to obtain the contract. However, if the proposal is not feasible, because some constraints were not considered due to the lack of time, then this can generate additional costs for the manufacturer in correcting the solution. Therefore, this contract may be not profitable. Thus, the manufacturer is under a deadline to produce an initial feasible solution at the lowest possible cost within a very short time period.

In addition, after the preliminary design, almost always the product to be manufactured undergoes some modifications during the stage of detailed design of the line. The line manufacturer must continuously take into account these modifications. Furthermore, modifications of the design solution are difficult and time consuming. Therefore, optimization methodologies and tools are eminently useful for the preliminary design and for taking into account the modifications during the detailed design. We will now present the methodology as applied for one such decision making process.

3. General methodology

No matter what the type of transfer line is (dedicated, flexible or reconfigurable), its design demands an overall approach requiring the resolution of several interconnected problems [3]. Ideally, decisions relating to all these problems must be considered simultaneously. However, the total problem is very complex. Therefore, it is necessary to decompose this problem into several sub-problems, each engendering less complex decisions [24].

Note that only the preliminary design stage is considered in this paper, i.e. when all principal decisions are made concerning line architecture and its elements. Usually, this is followed by a detailed design (specifications for mechanical elements, tools, spindle heads, etc.) which is out of scope of this paper (see [12]).

The following general steps can summarize the preliminary design process. Note that the importance of each step depends on the type of transfer line considered. Some steps can be omitted.

- Product analysis: this gives a complete description of the operations that have to be executed for the future products.
- Process planning: covers the selection of processes required to transform raw parts into finished products. Here, technological constraints are defined. For instance, during process planning, partial order between operations, inclusion and exclusion constraints are established. This requires an accurate understanding of the functional specifications for the products and technological conditions for the operations.
- Configuration design and balancing problem: selection of the type of machining line and the resolution of the balancing problem, i.e. the allocation of operations to workstations in order to obtain the necessary production rate meeting demand while achieving the quality required. It is imperative to consider here all the constraints, particularly, those of precedence.
- Dynamic flow analysis and transport system design: simulation is used to study the flow of products taking into account random events as well as variability in production. The objective is to analyze the dynamic flows and choose the material handling system as well as optimize the facilities layout, i.e. placement of machines. The decisions must be coherent with those defined at the previous steps.
- Detailed design and implementation of the line.

As illustrated in Fig. 2, these steps are executed sequentially. Of course, the designer can return to the previous steps as often as necessary (the decision making process is iterative).

This approach is based on a set of engineering procedures, knowledge based constraints, and some optimization techniques for transfer line balancing.

4. Optimization problem

The studied lines are equipped with CNC machines (machining centers). All the machines are identical (line modularity principle), each machine contains one spindle and a magazine for tools. For each machine, to pass from one operation to the next it is necessary to consider an additional time due to tool changes and displace-
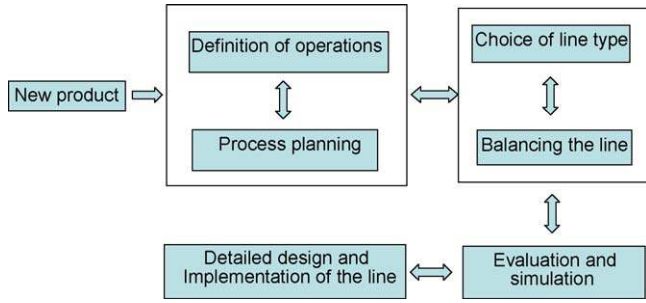
Fig. 2. Design of transfer lines.

ments or/and the rotation of the part (set-up time). Taking into account the fact that a part is held at a machine with some fixtures in a given position (part fixing and clamping), some faces and elements of the part are not accessible for machining even after the part displacement or rotation [13].

Whatever positioning and clamping chosen some areas on the part will be hidden or covered. Therefore, the choice of a part position for part fixing should be also considered in the optimization procedure. We consider the optimization problem of this type of lines and we develop a model for line balancing. The input data used were:

- Cycle time (takt time) imposed by the objective production rate: one part is produced at each cycle;
- Precedence constraints: partial order relation between operations. These relations define feasible sequences of operations;
- Inclusion constraints: the need to carrying out fixed groups of operations on the same workstation due to tolerance technical constraints;
- Exclusion constraints: the impossibility of carrying out certain subsets of operations at the same workstation;
- Accessibility constraints: they are related to the positioning of the part; indeed, for a position some part sides are not accessible, and thus, the operations concerning these sides cannot be carried out without repositioning. In the considered machining line, only one part fixing position is defined for each workstation (the part repositioning is made between two stations).
- Sequence-dependent set-up times: the time required for the execution of two sequential operations is not equal to the sum of their times but depends also on the order in which they are done, because the time needed for the displacement/change of tool and part rotation are not negligible;
- Parallel machines: at each workstation several identical CNC machines can be installed. Thus, the local cycle time of the workstation is equal to the number of parallel machines multiplied by the line cycle time (takt time). The machines of the same workstation execute the same operations (in parallel on different product units).

Hence, here, we have a special case of line balancing with a sequential execution of operations, set-up times, parallel machines, as well as accessibility, exclusion and inclusions constraints.

We will introduce the following notations:

- N set of operations to be assigned;
- n_0 maximum number of machines on a workstation;
- t_i operational time for operation i ($i = 1, 2, \ldots, |N|$);
- t_{ij} set-up time when operation j is processed directly after operation i on the same workstation;
- T_0 objective line cycle time (takt time);

- P_i^* set of all predecessors of operation i (direct and indirect predecessors);
- F_i^* set of all successors of operation i (direct and indirect successors);
- ES a collection of subsets e ($e \in N$) of operations which must be imperatively assigned to the same workstation;
- $\overline{ES}$ a set of pairs of operations (i, j) which cannot be assigned to the same workstation;
- A set of possible part positions for part fixing in a machining center; only one of these positions is chosen for each workstation; a part fixing position defines the accessibility constraints for the part;
- $A(i) \subseteq A$ the subset of positions with which operation (i) can be processed;
- $Type$ set of possible machine types (ordered in increasing cost);
- $NT(\tau) \subseteq N$ subset of operations that can be processed by a machine of type $\tau \in Type$.

Solutions are characterized by:

- N_{ws}: the total number of workstations on the line;
- $N(k)$ ordered set $\{N_1(k), N_2(k), \ldots, N_n(k)\}$ of operations assigned to the station k;
- $M(k)$ the number of machines of the station k;
- $Ts(k)$ the available workload time on the station k; $(T_0 \times M(k) - Ts(k)$ corresponds to the workload time of the station k);
- $t_{ut}(k)$ the proportion of the time available on the station k (i.e., $T_0 \times M(k)$) that is needed to complete the sequence of operations $N(k)$ including operational and set-up times; $\tilde{t}_{ut}$ corresponds to the average value among all the workstations;

$$t_{ut}(k) = \frac{T_0 \times M(k) - Ts(k)}{T_0 \times M(k)};$$

- $type(k)$ the type of machines installed on the station k;
- $As(k)$ set of possible position for the station k;

The solution cost is calculated taken into account the total number of machines, the type of machines used, the number of workstations, and the balancing quality;

- $Cost(\tau)$ the cost of machine of type τ;
- QS_m cost of machines:

$$QS_m = \sum_{k=1}^{N_{ws}} Cost(type(k)) * M(k);$$

- $Cost_{ws}$ cost of installation of one workstation;
- QS_{ws} cost of installation of workstations on the line:

$$QS_{ws} = Cost_{ws} * N_{ws};$$

The objective function to minimize is calculated as follows:

$$QS = QS_m + QS_{ws} + \varepsilon \sum_{k=1}^{N_{ws}} |t_{ut}(k) - \tilde{t}_{ut}|;$$

This function corresponds to the total cost of the line, with ε being a sufficiently small value in order to favor a better balancing among solutions with the same cost.

5. Related work

Historically, the line balancing problem was first stated for assembly lines. As far as we know, the earliest publication on assembly line balancing problem (ALBP) was presented by [28]. Furthermore, exhaustive studies were made by several researchers

in the last 50 years with many interesting applications covered. Several articles provide broad surveys of this problem (see e.g. [7,26]). To summarize, the ALBP is NP-hard (see e.g. [9]). Much research has been generated to solve the problem by developing approximate or exact methods (see e.g. [1,2,10,23]).

Several research papers were devoted to the assembly line balancing problem with parallel machines [4,5,25]. Indeed, to duplicate the machines in workstations has several advantages [11], such as: reduction of non-productive time (idle time), respect of cycle time (when there are operations with an operational time exceeding the cycle time), improvement of the production output imposed by the longest operation and, finally, reduction of transport time and matter flows on the line. The SDALBP (sequence-dependent assembly line balancing problem) has been defined by [29]. The SDALBP is a new type of the ALBP. It takes into account set-up times between the operations. Set-up times vary according to the sequence in which the operation is processed. Therefore, to compute the total workstation time, a set-up time must be considered whenever a task is assigned next to another at the same workstation. The problem is not dealt with sufficiently in literature and there are few works devoted to its study [1,31].

The problem of machining line balancing is rather recent. It was defined in [14] for dedicated transfer lines and first called transfer line balancing problem (TLBP). Industry favors solving TLBP because the machining lines become too expensive otherwise. The TLBP consists of answering the following questions:

- Which machining units are to be chosen to execute the required operations?
- How many workstations are necessary?
- How should the machining units be assigned to the stations?

These questions can be answered by an intelligent assignment of operations and machining units to workstations minimizing the line cost while satisfying the objective production rate as well as respecting all other constraints.

Several exact and approximate (or heuristic) methods for TLBP have been proposed. Exact methods are useful to better understand the problem, however for large-scale problems they often require an excessive computing time. Contrarily, approximate methods can provide quicker results but do not guarantee the optimality of solutions. Additionally, a heuristic algorithm is often easier to develop than optimal procedures, notably when complex constraints have to be considered.

The most significant methods for an exact resolution of the TLBP are:

- Linear programming in mixed variables: the problem is modeled as a mixed integer program and solved with an optimization tool such as ILOG Cplex [15,21].
- Dynamic programming: a recursive method used for the resolution of problems having an additive objective function. The examples of this approach for TLBP are given in [14], where the initial problems were transformed into constrained shortest path problems and solved with appropriate algorithms.
- Branch and Bound: an implicit enumerative procedure which avoids verifying all solutions. Several works use this approach for the resolution of the TLBP (see e.g. [16]).

By the way, the Column Generation method can be also used for TLBP. Indeed, it was already successfully used for assembly line balancing (see e.g. [31]). For large scale problems, or when the allocated computing time is severely limited (e.g. for flexible transfer lines), several approximate methods have been designed. We classify these methods into two categories:

1. Heuristics based on priority rules derived from the methods for ALBP. There are several heuristic algorithms, they differ in which rule(s) used:
   - RPW [Ranked Positioned Weight, [23]]: based on the weights of the operations calculated from their execution time and the operational times of their successors [22];
   - COMSOAL (COmputer Method of Sequencing Operations for Assembly Lines, [2]): solutions are generated by assigning operations randomly to the stations (see e.g. [20]).
2. Meta-heuristics, i.e. solving strategies applicable to a wide range of combinatorial optimization problems:
   - A heuristic multi-start decomposition approach was suggested in [22].

In this section, we presented a non-exhaustive survey of methods dealing with line balancing problems. Note that most of the above methods are developed for dedicated assembly or transfer lines. Outside [20,21], no paper in the literature presents an approach which could be applied in the industrial problem considered in this paper. For the machining line balancing problem, we consider a new type of line composed by mono-spindle head machines, and then consider set-up times. To fill this gap, we explore a novel problem in which we treat both paralleling of stations, sequence-dependent set-up times and accessibility constraints.

6. Optimization approach

The resolution of the studied problem is NP-hard as shown in Section 5. An exact method has been based on a mixed linear programming model for this problem [21]. However, experimentation has shown that only small sized instances could be solved. For this, we focused on approximate methods. In this paper, we propose an ant colony algorithm to solve large instances.

6.1. Ant Colony System

Ant Colony Optimization algorithms have been introduced by Dorigo [17] and were formalized as a new meta-heuristic in [18]. ACO has been firstly applied to Traveling Salesman Problem [19], but it has also been successfully applied to a wide range of combinatorial optimization problems and notably to scheduling and balancing problems (see e.g. [6,8,27–30]). ACO algorithms correspond to a multi-start constructive procedure. They are based on the movement of ants; solutions are generated by the combined effects of their movements and pheromones trails. The procedure is iterative; solutions are constructed item by item. Items are selected using a probabilistic function and a heuristic evaluation. Ants probabilistically prefer item with a lot of pheromone trail.

Initially, m ants are placed on randomly selected items. Each ant constructs its solution with the probabilistic selection policy and updates the pheromone level according to their solutions. A global collective memory is used to focus good solutions. This procedure is repeated a lot of time until a stop condition. Following the progress of the procedure, the items corresponding to the best solutions have a pheromone trails more important than others, which will converge towards local optima.

6.2. Optimization procedure

The proposed procedure is a constructive algorithm that assigns operations sequentially to stations. We call "current station" the station on which we assign operations on the current step of the algorithm. As long as the current station is not entirely filed, we cannot open a new station. If a station is opened, it becomes

current station. A partial sequence of operations is assigned to the current station; this partial sequence is completed step by step until the closure of the station.

We introduce some notations:

- N_a set of operation to be assigned;
- For any $N' \subseteq N$, $Type(N')$ is the subset of machine types capable to perform all the operations of N';
- L_m subset of operations of L_p have not predecessors not assigned and which can be assigned to the current station; i.e.: (i) the needed time to the execution of the operation (operational time plus the corresponding set-up time) is less than the available workload time of the current station, (ii) at least one of the possible positions of the operation is available on the current station, (iii) at least one possible type for the current station is compatible with the operation assignment, (iv) there is no exclusion constraints between the operation and the already assigned operations of the station;
- $LI(i)$ for all operation i of L_m, this set includes all the operations which must be assigned on the same station with operation i (inclusion constraints) and their predecessors of the set N_a;
- $t_l(i)$ the required time for the execution of all operations of the set $LI(i)$ (the sum of operational times and set-up times);

The algorithm allows us to build a feasible solution by the use of a greedy function namely g_i. g_i is calculated considering different parameters such as, (i) the operational time of the operations of $LI(i)$, (ii) the operational time of the successors of operations of $LI(i)$, (iii) the weight of operations which have not a common machining position with operations of $LI(i)$.

To calculate this function, we define the set $N_{pos}(i)$, the set operations which have not a common machining position with operations of $LI(i)$:

$$N_{pos}(i) = \{ j \in N_a | \exists l \in LI(i), A(j) \cap A(l) = \varnothing \};$$

Then, the greedy function expression is as follows:

$$g_i = \left(\left(\sum_{j \in LI(i)} \left(t_j + \sum_{f \in F_j^*} t_f \right) \right) + \left(\frac{|N_{pos}(i)|}{|N_a|} \right) * (T_0 - t_l(i)) \right).$$

This algorithm builds the solution step by step. On every step a set of operation candidates L_m, which can be added to the solution, is defined. After that, the elements are classified according the values of the corresponding greedy function.

We introduce the following notations for Ant Colony parameters:

- m: the total number of ants;
- τ_{ik}: the quantity of pheromones on the item when operation i is assigned to the station k. The update of τ is performed through the following formula:

$$\tau_{ik} \leftarrow \rho * \tau_{ik} + \Delta \tau_{ik};$$

where ρ is the memory conservation parameter; $(1 - \rho)$ corresponds to the evaporation factor of pheromones;
- $\forall i, k, i \in N(k), \Delta \tau_{ik} \leftarrow \sum_{Ant=1}^{m} Q/QS_{Ant}$: quantity of pheromones tabled at the current iteration;
- Q: average cost of previous solutions.

We define a discrete probability distribution function for each operation of L_m according to the following expression:

$$Pr_{ik} = \frac{[g_i]^\alpha * [\tau_{ik}]^\beta}{\sum_{j \in L_m} ([g_j]^\alpha * [\tau_{jk}]^\beta)};$$

- α: importance parameter of the greedy function;
- β: importance parameter of the pheromones trail.

The selection of operations within the algorithm will be performed using a roulette wheel based on the defined probability distribution function. The algorithm starts and creates a station. The algorithm stops when all operations are assigned. This iteration is repeated several times on the aim to build a large list of feasible solutions (L_s). The general scheme of the algorithm is presented following:

Algorithm.
1. $t \leftarrow 0; \tau_{ik} \leftarrow c, \forall i, k$;
2. $While(t < stop\_condition)$
3.   **For** $Ant = 1, \ldots, m$ **do**
4. $Initialize\ k \leftarrow 1, N_a \leftarrow N$;
5. $Compute\ L_m$;
6.   **While** $(N_a \neq \varnothing)$
7.   **If** $L_m = \varnothing$ **Then** $Close\ station\ k, k \leftarrow k + 1$,
          Go to $Step\ 5.$;
8.   **For All** $i \in L_m$;
9. $Compute\ LI(i)$;
10.   **For all** $i \in L_m$, calculate g_i and Pr_{ik};
11.   **Select an element** $i \in L_m$ according to the probability distribution function (Pr_{ik}) of operations of L_m;
12.   **If** $Ts(k) \geq t_l(i)$ **Then** add all operations of $LI(i)$ to the station k and update $L_p, Ts(k), N(k), As(k)$;
            $N_a \leftarrow N_a \backslash LI(i)$;
          Else If $M(k) < n_0$ **Then** $M(k) \leftarrow M(k) + 1$
          Else close the station $k; k \leftarrow k + 1$;
13.   **If** $L_p = \varnothing$ **Then Go to** $Step\ 4$
14.   **End While**
15.   **End For Ant**
16. $Update\ \tau, \alpha$ and β; $t \leftarrow t + 1$;
17.   **End While**

This algorithm is applied many times (a fixed number of iterations). Then, the best solution is selected by considering a solution criterion ($Cost(s)$). The function $Cost(s)$ takes into account the total number of machines, the type of machines used, and the number of workstations. If we have two solutions with the same cost, we select the one which have a better balancing quality.

7. Experimental study

In this paper, we proposed an optimization approach for large scale industrial problem. For the studied problem, outside methods proposed on [20,21], there are no other methods to solve it on the literature. Moreover the latter method cannot tackle problems of industrial size. To illustrate the capacity of the approach proposed in this paper, we have considered an industrial case and the results of the optimization by using the developed method are presented. The studied case is composed of 119 operations. We can use two types of machines, the first one with only one rotation axe and the second one with two different rotation axes. Consequently the two main differences between both machines are:

1. machines with two rotation axes can access to five sides of the part instead of only four for the machines with one rotation axe;
2. machines with two rotation axes are more expensive.

Table 1
Cost variations with the cycle time.

Cycle time	Heuristic	Maximum cost	Average cost	Minimum cost	Lower bound	Gap
100	1311.39	1221.14	1214.19	1210.65	840	30.82%
150	992.26	861.63	851.27	840.50	590	30.69%
200	802.71	671.30	657.82	651.00	490	25.51%
250	652.38	600.60	574.11	550.88	440	23.36%
300	582.26	521.70	501.18	490.87	390	22.18%
350	511.65	411.52	393.95	380.32	340	13.69%
400	492.37	401.31	389.09	380.66	340	12.62%
450	441.59	381.40	367.48	360.45	340	7.48%
500	422.05	381.29	367.52	360.65	340	7.49%

Outside of the basic constraints (i.e., precedence, inclusion and exclusion constraints), there is three possible positions for the part fixing on the machine. This is a very strong constraint for this problem since 75 of the 119 operations can be processed with only one part position. Our main objective consists to minimize the total cost of the line. We note that we cannot even find feasible solutions with the method of [21].

We apply the proposed method to this case study. We execute the procedure with 100 iterations. The computational time for 100 iterations is between 3 and 5 min which is fully compatible with a practical use. For each one, we present, in Table 1, the average, maximum and minimum costs for the best solutions generated by the ACO obtained on 6 independent runs .

The parameters of the algorithm are fixed as follows:

- $\alpha = 8$ and decreases with iterations;
- $\beta = 4$ and increases with iterations;

Each iteration α and β are updated as follows: $\beta \leftarrow 1.01 * \beta$ and $\alpha \leftarrow 0.99 * \alpha$;

- $m = 10$.

Several values of cycle time, from 100 to 500 units of time, have been considered. The corresponding results are reported in Table 1, as well as the results obtained with the heuristic proposed in [20]. Also, in order to assess the solution quality obtained by the proposed algorithm, we introduce a lower bound of the problem (LB). This lower bound concerns the solutions cost, then, it takes into account both the number of workstations, machines and types of machines. We use the algorithm proposed in [21] to calculate the minimum number of workstations (LB_{WS}). It is calculated using the problem properties, i.e., operational times, cycle time, precedence constraints, exclusion and accessibility constraints. The minimum number of machines corresponds to LB_m, it is calculated as follows:

$$LB_m = \max\left(LB_{WS}, \lceil \frac{\sum_{i \in N} t_i}{T_0} \rceil\right).$$

There are several types of machines used (with different costs). For every type of machine, we calculate a lower bound $LB_m(\tau)$:

$$LB_m(\tau) = \lceil \frac{\sum_{i \in NT(\tau) \text{ and } i \notin \bigcup_{\tau' < \tau} NT(\tau')} t_i}{T_0} \rceil, \ \forall \tau \in 2 \ldots |Type|;$$

$$LB_m(1) = LB_m - \sum_{\tau=2}^{|Type|} LB_m(\tau).$$

Then, the lower bound calculation corresponds to:

$$LB = LB_{WS} * Cost_{WS} + \sum_{\tau \in Type} LB_m(\tau) * Cost(\tau).$$

In Table 1, the last column reports the gap between the average solution obtained with ACO and the lower bound. Note that the gap

variation can come from the poor quality of the lower bound for problem with a low cycle time.

Fig. 3 represents the costs obtained for each cycle time considered. It is interesting to note that the variation of the cycle times between 350 and 550 units of time only affect slightly the cost of the line.

One other interesting way to analyze these results is to consider the notion of efficiency which is used for assembly line balancing problems. For the problem considered in this paper, the efficiency would correspond to the product of the cost by the cycle time (see Table 2). Thus minimizing this value is equivalent to minimizing the cost per unit of product. For a decision-maker, this information can be particularly valuable when it is possible to duplicate the line to take advantages of a configuration with a larger cycle time but a better efficiency, or when the future demand is uncertain. In Table 2, the best efficiency corresponds to a line cycle time equal to 100 units of time. Such a cycle time would correspond to a very high production rate. However if the decision-maker seeks for a cycle time around 300 or 400 units of time, the solution with a cycle time equal to 350 appears to provide a good efficiency as well as a reasonable production rate.
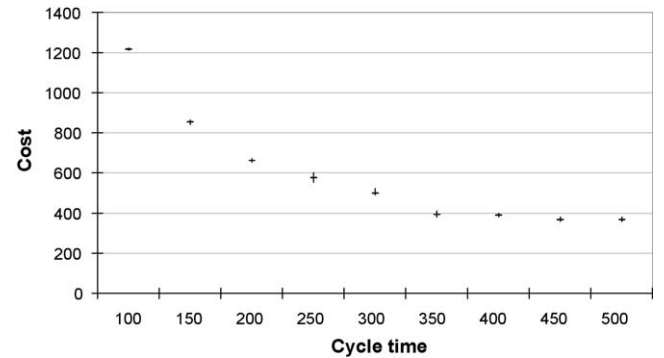
Fig. 3. Cost variation with the cycle time.

Table 2
Efficiency of the lines obtained with each cycle time.

Cycle time	Efficiency
100	121,419.50
150	127,690.75
200	131,565.66
250	143,527.50
300	150,354.00
350	137,884.25
400	155,637.33
450	165,366.00
500	183,760.83

8. Summary

A real industrial transfer line balancing problem with specific characteristics is considered and a multi-start ant based heuristic method is presented. The proposed heuristic is applied to resolve a real industrial problem. The processed experimentations show that we can use the proposed method to help designers to explore different scenarios and possibilities. Future research will focus to improve the developed method for example by considering a phase of local search applied on the solutions provided.

Acknowledgment

References

[1] Andrés, C., Miralles, C., Pastor, R., 2008, Balancing and Scheduling Tasks in Assembly Lines with Sequence-Dependent Setup Times, European Journal of Operational Research, 187:1212–1223.

[2] Arcus, A.L., 1966, COMSOAL: A COmputer Method of Sequencing Operations for Assembly Lines, International Journal of Production Research, 4:259–277.

[3] Askin, R.G., Standridge, C.R., 1993, Modeling and Analysis of Manufacturing Systems, Wiley, New York.

[4] Askin, R.G., Zhou, M., 1997, A Parallel Station Heuristic for the Mixed-Model Production Line Balancing Problem, International Journal of Production Research, 35/11: 3095–3105.

[5] Bard, J.F., 1989, Assembly Line Balancing with Parallel Workstations and Dead Time, International Journal of Production Research, 27/6: 1005–1018.

[6] Bautista, J., Pereira, J., 2007, Ant Algorithms for a Time and Space Constrained Assembly Line Balancing Problem, European Journal of Operational Research, 177/3: 2016–2032.

[7] Baybars, I., 1986, A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem, Management Science, 32/8: 909–932.

[8] Berrichi, A., Yalaoui, F., Amodeo, L., Mezghiche, M., 2010, Bi-Objective Ant Colony Optimization Approach to Optimize Production and Maintenance Scheduling, Computers & Operations Research, 37/9: 1584–1596.

[9] Bhattachajee, T.K., Sahu, S., 1990, Complexity of Single Model Assembly Line Balancing Problems, Engineering Costs and Production Economics, 18:203–214.

[10] Bukchin, J., Rubinovitz, A., 2002, A Weighted Approach for Assembly Line Design with Station Paralleling and Equipment Selection, IIE Transactions, 35:73–85.

[11] Buxey, G.M., 1974, Assembly Line Balancing with Multiple Stations, Management Science, 20:1010–1021.

[12] Dashchenko AI, (Ed.) (2003), Manufacturing Technologies for Machines of the Future: 21st Century Technologies. Springer, Berlin/Heidelberg.

[13] Delorme, X., Dolgui, A., Essafi, M., Linxe, L., Poyard, D., 2009, Machining Lines Automation. Part D, in Nof SY, (Ed.) Springer Handbook of Automation with DVD-ROM. Springer, pp.599–618. (Chapter 35).

[14] Dolgui, A., Guschinsky, N., Levin, G., 1999, On Problem of Optimal Design of Transfer Lines with Parallel and Sequential Operations, in Fuertes JM, (Ed.) in: Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'99), vol. 1pp.329–334.

[15] Dolgui, A., Finel, B., Guschinsky, N., Levin, G., Vernadat, F., 2006, MIP Approach to Balancing Transfer Lines with Blocks of Parallel Operations, IIE Transactions, 38:869–882.

[16] Dolgui, A., Ihnatsenka, I., 2009, Branch and Bound Algorithm for a Transfer Line Design Problem: Stations with Sequentially Activated Multi-Spindle Heads, European Journal of Operational Research, 197/3: 1119–1132.

[17] Dorigo, M., 1992, Optimization, Learning and Natural Algorithms (in Italian), Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano.

[18] Dorigo, M., Di Caro, G., 1999, Ant Colony Optimization: A New Meta-Heuristic, in Angeline PJ, Michalewicz Z, Schoenauer M, Yao X, Zalzala A, (Eds.) Proceeding of the Congress on Evolutionary Computation, vol. 2. IEEE Press, pp.1470–1477.

[19] Dorigo, M., Gambardella, L.M., 1997, Ant Colonies for the Travelling Salesman Problem, BioSystems, 43:73–81.

[20] Essafi, M., Delorme, X., Dolgui, A., 2007, A Heuristic Method for Balancing Machining Lines with Paralleling of Stations and Sequence-Dependent Setup Times, Proceeding of the International Workshop LT2007 (Sousse, Tunisia, 18–20 November), pp.349–354.

[21] Essafi, M., Delorme, X., Dolgui, A., Guschinskaya, O., 2010, A MIP Approach for Balancing Transfer Line with Complex Industrial Constraints, Computers & Industrial Engineering, 58/3: 393–400.

[22] Guschinskaya, O., Dolgui, A., Guschinsky, N., Levin, G., 2008, A Heuristic Multi-Start Decomposition Approach for Optimal Design of Serial Machining Lines, European Journal of Operational Research, 189/3: 902–913.

[23] Helgeson, W.P., Birnie, D.P., 1961, Assembly Line Balancing Using the Ranked Positional Weight Technic, Journal of Industrial Engineering, 12:394–398.

[24] Hitomi, K., 1996, Manufacturing System Engineering, Taylor & Francis, London

[25] Pinto, P.A., Dannenbring, D.G., Khumawala, P.M., 1975, A Branch and Bound Algorithm for Assembly Line Balancing with Paralleling, International Journal of Production Research, 13/2: 183–196.

[26] Rekiek, B., Dolgui, A., Delchambre, A., Bratcu, A., 2002, State of art of optimization methods for assembly line design, Annual Reviews in Control, 26:163–174.

[27] Sabuncuoglu, I., Erel, E., Alp, A., 2009, Ant Colony Optimization for the Single Model U-type Assembly Line Balancing Problem, International Journal of Production Economics, 120:287–300.

[28] Salveson, M.A., 1955, The Assembly Line Balancing Problem, Journal of Industrial Engineering, 6/4: 18–25.

[29] Scholl, A., Boysen, N., Fliedner, M., 2008, The Sequence-Dependent Assembly Line Balancing Problem, Operational Research Spectrum, 30/3: 579–609.

[30] Spicer, P., Koren, Y., Shpitalni, M., Yip-Hoi, D., 2002, Design Principles for Machining System Configurations, CIRP Annals-Manufacturing Technology, 51/1: 275–280.

[31] Wilhelm, W.E., 1999, A Column-Generation Approach for the Assembly System Design Problem with Tool Changes, International Journal of Flexible Manufacturing Systems, 11:177–205.