SIMULATION MODELS TO SUPPORT GALB HEURISTIC ALGORITHMS AND TO EVALUATE MULTI OBJECTIVE PERFORMANCE INDEX

Sergio Amedeo Gallo [a], Giovanni Davoli[b], Andrea Govoni [c], Riccardo Melloni[d], Gabriele Pattarozzi[e]

[a, b, c, d, e]Department of Mechanical and Civil Engineering (DIMeC),University of Modena and Reggio Emilia, ITALY

Corresponding Author: Sergio A. Gallo e-mail: sgallo@unimore.it, Tel +39-059-2056113, Fax.+39-059-2056126

[a]sgallo@unimore.it, [b]giovanni.davoli@unimore.it, [c]andrea.govoni@unimore.it, [d]riccardo.melloni@unimore.it, [e]gabriele.pattarozzi@gmail.com

ABSTRACT

The following paper deal with an approach to analyze a multi model manual assembly line for oleo dynamic products, and the following **heuristic algorithm** to optimize the scheduling of tasks to the available stations, under the respect of a set of restrictions, as task/station obligation, and with the aim of optimize a multi objective function based on time and line balancing costs elements.

This problem can be considered as belonging to the wide area of **GALB** Problems.

Some strategy about resource scheduling opportunities has been considered.

The original system was an assembly line with six stations, and, in the original and referable configuration, with six operators, but, under the spur of residual margins of the utilization ratios, as a improving evolutive strategy, an idea of evaluate opportunities of improving performances, by assembling coupled mixed items, feeding together the assembly line, have been evaluated, based on a twin mirror counterflowed line.

1. INTRODUCTION

Tasks assignment to stations has to respect efficiency concerns, as the maximization and the balancing of utilization rate, but, furthermore, specific task assignment restrictions to specific station because of the needing of special machines, available just at defined station, for all the items of the production mix.

The line moving is based on a conveyor system, of accumulating type, so that the line is paced, but not synchronized. No inter operational buffer is allowed in the original system, but short accumulation among stations of five pumps is allowed.

Production plan data and configuration derive from a real assembly line. Task times are stochastic, and, based on real observations, and can be considered as triangular or lognormal distributed.

This aspect, has not had opportunity to be considered in the present scheduling problem, because no cost for off line completion can be considered, because task that do not respect line cycle time, cause just the line to increase tack time. No incompletion costs, and no operators moving cost, or costs related to operators training, or changes of the task to operate has been considered, because we are considering a multi model assembly line, moreover supported with displays showing instructions for tasks to operate. Operators moving costs are supposed be negligible compared to operating cost, because, the short distanced to cover, or the parallel shape of the double line.

A precedence diagram supports technological constrains, very similar among items.

The same tasks require item/size dependent times; times can vary for each item, for the same operation.

Not all operations are performed for all items, depending on item features, and optional.

The assembly line can process a very large variety of items, defined in families, six, but that differ for size, features, optional, under the increasing market competition.

Lot size can varies largely, with a strong tendency to the reducing of assembly quantity for single order.

The performance parameters are tack time, to be reduced, equal to the production rate maximization, and, on an opposite way, optimize the internal balancing of tasks for stations, and labour level among operators.

Under these criteria, a multi objective function based on the whole lot assembly cost, calculated on the effective tack time and on the scheduled resources and their balancing level, has been defined.

The results demonstrate the capability of the proposed algorithm of dealing with the multi objective nature of the re-balancing problem. Solutions with advantages both in tack time reduction, and both on balancing improvement are obtained in any case.

Stations quantity, or, in other words, the assembly line configuration, has been defined to replay with efficiency to all needs for all types belonging to the general production mix, and to respect the assignment of specific tasks to specifically equipped stations.

A virtual model of the assembly line has been built in a simulation environment, to test and measure performances of the heuristic algorithms, but, moreover, also all the algorithm code has been implemented in the same software platform, so simulation has been used not only as a verifying tool, but also as a solving or solution finder, and as task and resource scheduler.

Logics in the heuristic model have been wrote to be as more general and flexible as possible, under the dimension level, but also to consider the more general problem possible, with easy data configurations.

In the future modular evolution of the model, it should be able to select the configuration asset and the problem type, and choose the algorithm path to apply the best strategy.

The definition of the sequencing in the assembly plan is oriented to the lean production philosophy, so that, first production order has to be produced first. Any way, at least one week, is the planning time horizon an order can be shifted inside the production plan without affect due dates. So, is possible over pass the rule to route assembly orders as they were placed in the order list, to get better system performances.

An early model has been developed. This precedent model, much more simple, represents the "as is" configuration of the firm for strategies and configurations, and is a basic model to be compared with our improved one.

In the first part of the present model, assigning rules have been defined, attempting to fulfil and add tasks to stations in sequence, till total station time doesn't overpass calculated referable tack time. Additional control code check if any constrained task is joined, and in this case, provide to verify the station where to assign that task, if not the actual one, and to calculate all parameters for intermediate stations.

Additional rules evaluate if some operator is idle, and not charged, and the opportunity to re allocate to the more suitable stations, and the recovered time, and all new parameters values.

Just the best allocation will survive, and configuration will be recorded.

More over, if some station is undercharged, a routine defines a recursive increment of the tack time, till all those stations become empties, so freed operators could be re assigned to over charged station.

Also in this case, global efficiency is calculated and compared to the best previous.

Nothing more is possible under this configuration.

After the algorithm implementation to the whole assembly mix plan, output performances has been analyzed, finding, for some item, residual margins for resources utilization rates, no more improvable, because of line configuration, and constrains accomplishment.

As a improving evolutive strategy, based on residual efficiency values, i.e., on the complementary values of the utilization rates, the idea of evaluate opportunities of improving performances, by assembling coupled mixed items, feeding together the assembly line, have been evaluated.

But, because of the constraint position of a special chamfer machine, allocated on a defined station, and because of other operations, just available at other stations, common to the whole items set, opportunities to gain better performances sequencing repeatedly profiled groups of items extracted by the production or assembly plan, PP or AP, conveniently defined for quantity and for typology, has seemed, not so feasible.

In fact, station time profiles, resulted similar for the large part of available items, differing, often, just for the time scale.

Under this observation, the only opportunity to achieve better performances, grouping single units of different available orders, was to couple tasks to be assembled, on two parallel lines, feeding each in counterflow; operators, in this configuration, should be assigned to stations, one for each line, and they should have to complete their operations, alternatively, on both stations of two lines.

This extension of the original system, make possible, with a minimum impact of investment costs and space consuming, to operate more strategies: some items can be processed alone, on a single line, under the tasks assignment rules, or, if it was better, the lot can be processed, half and half, on both the counter flow lines, coupling with itself; finally, some item can be processed coupled with just one other item, in order to improve the performance index.

The choice to test matches of not more than two items only, is based on the consideration that, we are not sure to find, in a specific new assembly/production plan, right combinations, both under the type issue, and both, under the right matching quantity aspect, with the uncertainty level growing up as the matching size grows. But is much more alike finding sets of just two different items that can be coupled and assembled till the quantity of one of the twos ends, and then the remaining quantity can assembled with another further item, available in the AP, or, in the worse case, can be assembled alone with a lower efficiency.

To do this, special matching list have been compiled in a preliminary phase, launching simulation runs to output performance indexes, for each item considered to test all matches with any possible other.

An algorithm, similar to the one used to efficiently assign tasks to station, now adapted to assign, at the same time, tasks of matching items to symmetrical stations, has been wrote and used.

After this, out put results, namely efficiency indexes, and, much more, lot assembly cost reduction, have been listed, for each item, in descending way, to define matching tables.

In a following phase, assembling/production plans have been defined, extracted from the previous one, just items that in the precedent balancing phases showed a significant recovery edges for the efficiency index.

Models can be applied to a wide variety of systems, with a different number of stations, and different constrains positions, just integrating new additional constrains rules. An extension of the actual model to consider other features or restriction is under evaluation, as statistical testing are in progress at this moment to evaluate sensitivity and consistence of the model.

Achieved results show enthusiastic improvements compared with initial solutions, and any time a new strategy is applied.

Keywords: Manual Assembly Lines Balancing, GALB, Modeling and Optimization Heuristic Algorithm.

2. LITERATURE REVIEW

An assembly line is a flow-oriented production system, where the operative location units performing work, referred to as stations, are sequentially aligned. Work pieces move on transportation systems as a conveyor.

Assembly lines are very common in the final phase of industrial production of high quantity of small lots of standardized but customized commodities (mass-customization). They are medium intensive capital systems, so, their configuration planning is relevant.

The Assembly Line Balancing Problem (**ALBP**) means the assignment of tasks to stations and operators on a line, whereas the items are produced at pre-specified production rate. Configuration planning comprises all tasks allocation and decisions which are related to equipping and aligning the productive units for a given production process, including setting the system capacity (cycle time, number of stations, station equipment) as well as assigning the work content to productive units (task assignment, sequence of operations).

Since the times of Henry Ford and the model-T, customer requirements, and consequently, production systems, have changed in a way to increase dramatically customization of their products. Multi-purpose machines with automated tool change, allow, at very low set up cost, for distinct production sequences of varying models. The high level of automation of assembly systems and the fixed movement system make the (re)-configuration of an assembly line critical.

It is easy to find a wide variety of algorithms to solve ALBP in literature, any one facing a partial part of the problem, or oriented to a particular system or configuration.

In spite of the huge academic effort in assembly line balancing, a large distance between requirements of real problems and the status of research, remains.

Many of them consider the problem too much statically, just under a one point of view.

But the increasing need to face continuous changes in customer's requirements, as product design, restyling and lot quantity needed, enforced with high customization and reduction of time-to-market, push to test dynamic versions of ALBP solution procedures.

Those modifications imply a very high flexibility level for the line.

ALBP consists of assigning tasks to stations in such a way that (Salveson, 1955):

- each task is assigned to one and only one station;
- the sum of performance task times assigned to each station does not exceed the cycle time;
- the precedence relationships among the tasks are satisfied;
- some performance measures are optimized.

Most procedures consider the types **I and II ALBP**, based on minimization of the number of stations, given a desired cycle time or minimization of the cycle time, given a desired number of stations, respectively.

Because of the numerous simplifying assumptions of this basic problem, this problem was labelled simple assembly line balancing (**SALB**) in the widely accepted review of Baybars (1986). Subsequent works attempted to extend the problem by integrating practice relevant aspects, like U-shaped lines, parallel stations or processing alternatives (Becker and Scholl, 2006), referred to as general assembly line balancing (GALB).

Scholl (1995), and Pierreval et al. (2003) proposed a very large and comprehensive reviews of the approaches developed to solve the problem.

Ghosh and Gagnon (1989) proposed a taxonomy to classify ALBP solution procedures under two key aspects, mix or variety of items produced on a single line and the nature of performance task times: single model lines or multi/mixed model lines manufacturing more items in batches or simultaneously; deterministic ALBPs, in with performance task times constant, or stochastic ALBPs, with stochastic task times distributed according to a specific distribution function.

ALBP can be solved to optimize both time - and cost, as reported in Amen (2000, 2001) and Erel and Sarin (1998), which concern the deterministic and stochastic versions of the problem, respectively.

Moodie and Young (1965), Raouf and Tsui (1982), Suresh and Sahu (1994), Suresh et al. (1996) have proposed time-oriented algorithms, improving procedures developed for the single-model deterministic problem, with the aim of minimize stations number and the over time to complete the work off the cycle time.

In any case, in stochastic assembly lines relevant incompletion costs often occur, and a multi objective cost function often is needed.

Two cases, both described in literature:

- the whole line is stopped till the over work is completed (Silverman and Carter, 1986);
- incomplete products get completed off-line.

Kottas and Lau (1973, 1976, 1981) proposed heuristic procedures to minimize both the total labour cost and the expected incompletion cost. Extensions of the Kottas and Lau's (1973) method were developed by Vrat and Virani (1976), Shtub (1984)

Sarin et al. (1999) proposed, not so general as Kottas and Lau's (1973), a branch and bound heuristic to minimize the total labour cost and the total expected incompletion cost with good results.

Erel and Sarin (1998) noticed the difficulty of methods in literature to model real conditions, and suggested that newer works should be oriented at useful studies, with impact on real-life assembly lines.

Rekiek (2000) observed that differences among ALBP and real-life statements were the multi-objective nature of the problem, no so considered in literature.

Some studies deal with the re-balancing problem of an existing line, as Sculli (1979, 1984) and, Van Oyen et al. (2001) considered the re-balancing of an existing line, under fluctuations of operator output rates or equipment failures, in short-term problem. The proposed solution to avoid temporary imbalance on the line has been the dynamic work sharing.

Rekiek et al. (2002) demonstrated that the integration between heuristic approaches and multi-attribute decision making techniques is a proven and efficient way for solving assembly lines problems.

3. SYSTEM AND CONFIGURATION DESCRIPTION

The original assembly mix presents six product families, moreover divided in 127 different versions of items to assembly, depending on optional, features, displacement, etc. The original system based on a conveyor system, shows six work stations, based on original firm evaluations.

Some constrains in the system is based on the needing to have some equipment available at a defined station; also related tasks have to be operated at those stations.

Some further constrains arise from the lean production philosophy, and the FIFO logic: the first order is placed in the system, the first get on the line.

In our model we will overpass the layout constrain, and a model with a coupled double line will be tested; all stations, and all equipments will doubled, except for the chamfer machine, the most relevant, that will be shared to achieve a better balancing, cost reduction and productivity improvement.

Line shall be capable to perform assembly orders very different each from others; just a manual assembly line can face this with negligible set up time.

Task time are triangularly distributed.

Some model parameters (times in hundredth of minute):

Station Number $k \in [1, n]$

Task Number $n \in [1, 34]$

N° of tasks assigned to a single station $i \in [1, h]$

Task Time T_{op}

Station Time $T_{Stat} = \sum_{i=1}^{h} T_{Op}$

$T_{Stat} = UC_S * Tm$

Operation Unbalancing Coefficient $UC_{Op} = \frac{T_{Op}}{Tm}$

Station Unbalancing Coefficient $UC_S = \frac{T_{Stat}}{TT_{Line}} \%$

$$UCs = \sum_i UC_{Op} = \frac{\sum_i T_{Op}}{TT_{Line}}$$

Line Lead Time $LLT = \sum_{i=1}^{k} T_{Stat}$

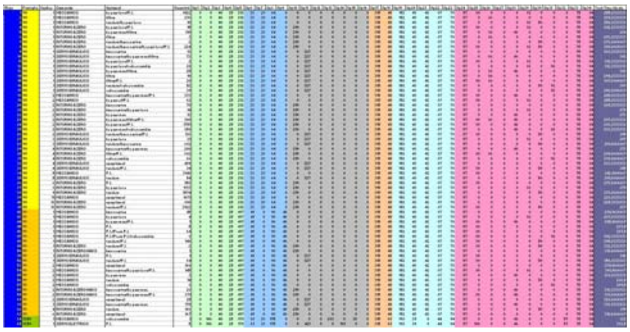
Line Tack Time or Cycle Time

$$LTT = Max(\sum_{1=i}^{h} T_{Op}) = Max(T_{Stat})$$

Mean or Ideal Tack Time $TT_{mean} = \frac{\sum_{1=1}^{n} T_{Op}}{k}$

In table 3 original production plan with all descriptions and task times and ideal Tack Time.

Table 1: whole mix production plan with parameters values and task times



Task times ranges from 5% to 80% of the Referable Line Tack Time.

Out of 34 tasks, three are to be operated with constrained equipment, always at fixed stations:

- task 17 (chamferring) at 4 station;
- task 19 (screwing) at 5 station;
- task 33 (air testing machine) at 6 station.

In the basic configuration station 5 is quite always the most charged one: it affects Line Cycle time.

Quite always station 3 is idle, whereas station 2 is often charged, but with a very lower unbalancing Coefficient.

Station 6, depending on the optional presence, can largely varies its station time.

More over, station time shape looks very similar for all station under different items allocation, and stations 1, 2, 4 e 5 seem similarly charged for same family items, also for the station time.

3.1. MALB algorithm

We propose a heuristic method defined by an algorithm logic to achieve feasible and optimized solutions, not necessary the optimal one.

First, we will allocate tasks to stations trying to fulfil the referable cycle time, more over respecting task constrains, to achieve cost minimization, and a better charge balancing.

Our assembly line is a multi mixed model, and then we will face with a **MALBP** (Mixed-Model Assembly Line Balancing Problem).

In particular, will configure our situation as a **MALB-E** problem, given number of K stations, the aim is to maximize the efficiency E_{line}, ie minimizing the direct cost of assembling the lot, given by the product of number of stations K and the cycle time LTT, the production volume of the lot and the cost of labor.

The balance algorithm we will define, trough all steps of the model, will be a mixture of Balancing on the work content, through a dynamic allocation of tasks to be performed at each station. In the “as is” situation, the station times were very inhomogeneous, and each row of the production plan wanted peculiar allocation. Resources Balancing, which dynamically assigns operators to stations rather than operations. Balancing on scheduling, by analyzing and optimizing the production plan, or by defining an ideal sequence of orders to be sent into production, or mixing them.

Initially, each operator has been considered bounded to his station, and tasks allocation was made balancing on the content of work; the primary goal the algorithm was designed for is to allocate tasks dynamically to stations, with the ultimate goal of maximize efficiency, but also of minimizing the direct cost of assembly:

$$MAX\left(E_{line}\right) \qquad where \quad E_{line} = \frac{\sum_i UC_S}{n} \quad [\%] \qquad (1)$$

Efficiency is calculated as the sum of the UC_S divided by the number of charged stations, initially fixed and corresponding to that of resources, thus as you maximize the efficiency as you maximize utilization, or will reduce the unbalance for each individual station (UC_S).

$$MIN\left(UC_L\right) \qquad where \quad UC_L = \sum_i UC_S \qquad (2)$$

Modified COMSOAL Heuristic

In order to achieve his goals, we will apply a balancing method inspired to the Chrysler heuristic **COMSOAL** (Computerized Method for Sequencing Operations on Assembly Lines), that we adapted to our system.

We can obtain a set of feasible solutions, randomly selecting at each step, from the available tasks, and then going to schedule the best according to a certain criterion.

At each iteration of the algorithm is determined by the set of all tasks that may be assigned to a station k without violating the constraint, ie the tack time LTT, of this. Any task, at any step, is chosen at random, with the same probability of selection.

In our case, the selection of the task to be assigned, can be detected according to a logic which respects the sequence of the assembly operations, and not in a random manner.

Normally, the heuristic is repeated a certain number of times, and each solution found is compared with those found previously. The number of repetitions to be performed depends on the analyst. Note that the same solution can be obtained more than once.

In the modified version of the choice of operations is not random.

In this regard, even the random selection of tasks to be assigned to the first operation is outdated, because in this case the only task for which all of the precedent are either absent or assigned, is the first, and then go on. Following the precedence diagram, is not possible face with multiplicity of possible situations, but it will lead directly to the case that identifies the optimal balance.

To assembly a model not all the 34 operations are needed for a specific item.

Another parameter used in the preparation of the model is described below.

Reference Tack Time

$$Tm = MAX\left(T_{mean} = \frac{\sum_i T_{Op}}{n};\ MAX\left(T_{Op_n}\right)\right) \qquad (3)$$

Minimum time, used as a limitation roof in the allocation of tasks, maximum threshold for the station Time can not overcome. It is the higher value between the tack time average or ideal (**LTT**) on the six stations, and the maximum values of the specific various tasks (T_{OP}) for each line of the production plan. The tack time reference, represents 100% of work time that can be assigned to each station.

In our case, a further control has been added to assure task allocation to the correct stations.

Another specific problem is the highly variable size of the tasks. Some of them, in fact, affect dramatically the ideal tack time, and, where compared to the starting value of T_m, they are often even larger.

As a consequence, in the recursive routine, before T_m was increased enough, for many times the assignment to the current station fails, the current station is closed, and allocation goes tried to the next. In the extreme case, the task can present a particularly high UC_S that causes in the first attempts, with low T_m, its failure to allocate whatever the station is. In this case, a control amplifies the T_m of a defined percentage, and, after resetting all assignments, try to reassign tasks to stations with the new reference T_m, until it reachs.

Our software performs calculations and allocations and defines the values of key parameters (Station tack time, coefficients of imbalance, etc...) of the specific logic block, analysed to define any new further proposals for balancing the line.

3.1.1. Model Description

AP and configuration data, are read from external files.

The code process part that takes care of reading data is called "P_read" program in this part of the main variables are loaded with the values we impose on us from outside.

The conveyor speed is set to a very high value, transferring times are included in the average time of execution of the task, and result very lower when compared to operational times, with no statistical significance.

A first piece of code initializes the model and its parts, to load the variables with the values of the external file and configures the same in accordance with the structured algorithm for assigning tasks to stations.

In this phase, there will be defined the values of the Line tack times, of the UC_S and the parameters of cost and inefficiency.

The logic routine dynamically assigns tasks to the stations whatever model you're sending in production, also respecting the allocation of joined tasks to the stations of belonging.

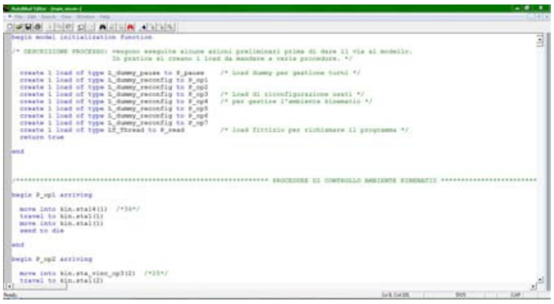


Figure 1: Initialization model code screenshot

The file "***SpeadSheetWorkDataCSV***" contains information about a large number of parameters that will be reported in array variables:

- The tasks time for each line;
- The ideal tack time;
- The reference tack time;
- The lot amount of pumps in the order line;
- The pointers to constrained tasks;
- The pointers to the constrained stations;
- The unit cost of labor;
- The number and type of tasks needed to assembly a specific item order.

Another file "***FeaturesCSV***" supports parameters that identify a defined type of pump, the family code, the command and options fitted, respectively, stored in variables of type string.

In the third reading file, "***ConfigurationCSV***", are the values of the variables to configure the system, such as the percentage increase value for reference tack time, and others which will be discussed later.

As the reading process ends, the assignment process of tasks to stations starts. This process, "***P_allocation***", consists of several "while" cycles and conditions such as "if ... then ... else" that allow you to assign tasks properly and considering various alternatives.

A first outer iterative cycle, will take care of all the operations of the process on each of the 127 rows and with each new line initialize to zero the variables that contain the values of stations tack time, increased, time by time with the task addition. Also initial values for other control variables or support are reset.

At this point starts the allocation of the first task to the first station through a further "while" loop wrote to check that the provisional station tack time, do not overpass the reference Tack Time T_m.

While this condition keeps true, assigning the next task to the station will stand, and some controls work.

The first control regards the constraints, in fact occurs that the task is not constrained so that could be freely assigned, or, on the other hand, if we are in the station it belongs to, so that it could be attributed.

Figure 2: Counters and main variables Initialization model code screenshot

An "else" condition, closes the current station in case the task bound should be assigned to a different station. In this last case, all the stations among the current one, just closed, and the constrained one, are declared empty (tack time and assigned operations are zero) and tasks allocation start again from the constrained station joined to the constrained operation.

Another control checks whether or not, the addition of the task you are trying to give, does not lead to a tack-time exceeding the reference limit. In this case, the station is closed and the allocation starts again to assign the current task to the next available station.

In case last station gets overcharged, over passing reference task time, before last task should be assigned, the code logic increase the reference by a defined percentage and set all row array values, containing stations tack time, and all others parameters to zero, and again, goes to try allocation again, till it reaches.

When you have no more tasks to be allocated, the assignment process for that line ends and the next one in the production plan is considered, just after saving the data for each station of the completed order row within the appropriate variables: the reference tack time, as well as, the number of operations assigned to each station for each line, etc. are saved to array variables.

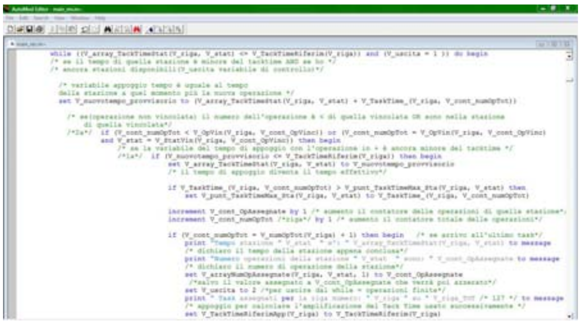


Fig. 3: task allocation code snapshot

3.1.2. Reallocation of “spontaneous” idle operator and strategy of under-used stations emptying

Now, we can observe data and yield first conclusions and analysis: we can see many stations showing markedly under – charged station time or even empty.

The first improving strategy provides that the operators that are already no charged, as their stations are empty, are reassigned to the station with the highest tack time so to lower it.

This last configuration (config_2) will be compared with the previous scenario (config_1), but when we calculate the whole direct assembly cost we will care to not consider the resources associated with the empty stations.

The comparison between these two scenarios is based on a cost parameter calculated for each station in each row in both cases. This parameter is calculated by multiplying the number of resources for the reference tack time, for the unit cost of labour and the number of pumps produced. The config_2 is calculated by two procedures that control the stations without assigned operations, storing them on array of pointers, to define the amount of the resulting uncharged operators to free. The second process chooses the most charged station to assign the operator and calculates the decremented tack.

The tack time reduction follows a procedure that takes care of dividing tasks by two operators in the best balanced manner. The procedure, performed for all rows with new assigned operators, is repeated until there are more operators to be assigned.

The better configuration will test the next improvement. Subsequent values of tack time, tasks assigned, imbalance coefficients, efficiency and direct cost for each configuration are stored in distinct array variables.

At this point, the further forced reallocation is attempted.

In the procedure "**P_uncharged_Op_forced_assignment**" all the stations for each line are checked, to trace the presence of under - utilized stations. Once that has been identified a station of this type, it is emptied, and its operator, released by force.

T_m is increased by a defined percentage, all assignment values are placed to zero, a new tasks reallocation starts till under charged station is empty.

In this logical block, in the first phase, operators resulting already uncharged from the previous step, are not reassigned to overcharged stations, otherwise direct cost would be greater, given the increasing number of T_m. In this way, there is a further situation, even at reduced resources, which is compared with the winning one between the previous and the subsequent re-allocations, which will be a phase next configuration.

At this point the situation is photographed by saving the various parameters in appropriate variables.

As mentioned earlier, the reallocation of uncharged workers will follow.

We will attempt to reassign available operators to the remaining stations, with the objective to reduce tack time and improve efficiency and costs.

Once again all the characteristic parameters of each situation will be saved for later comparison with those from previous situations.

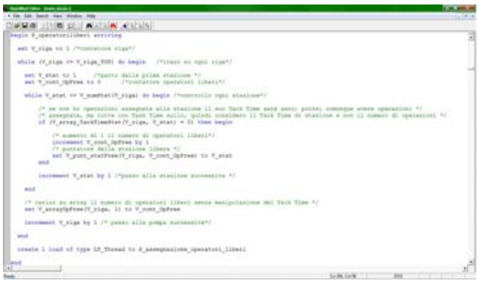


Fig. 4: uncharged operators search code snapshot

All configuration parameters values, are read from an external file "**configuration.csv**", making possible to change its value file.

The highest tack time among all various stations, is saved by line.

The position number of the station with two scheduled workers is saved in a pointer array variable.

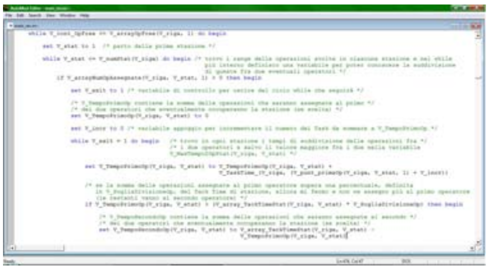


Fig. 5: new reference tack time code snapshot

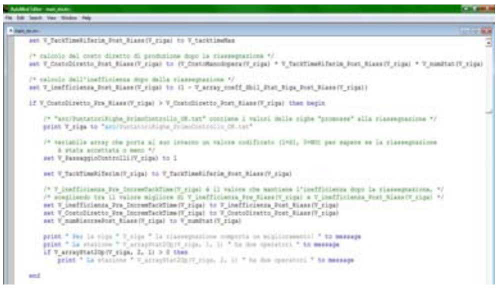


Fig. 6: code snapshot to compare direct assembly cost

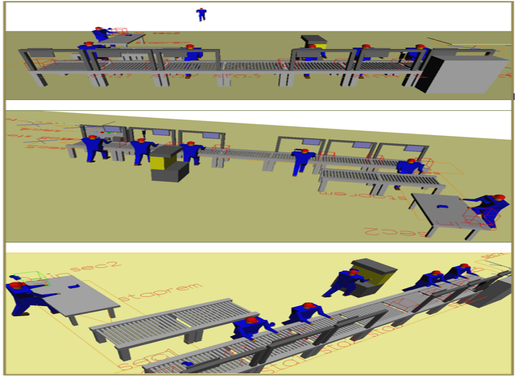


Fig. 7: snapshot of the single assembly line model

3.1.3. Assembly process simulation code

As all possible configurations have been defined, is possible describe the direct cost objective function calculation and their comparison.

Based on this, all configuration values are tested to be verified on the code part of the assembly line emulation.

A single load actives all the logic and all the code lines that can emulate the assembly line.

The active load acts a process cascade, one for each station, where the presence or needing of the operators is evaluated, and in case, the correct tack time is calculated and charged in a parameters set, respecting the time probability distribution.

A counter takes memory of produced pumps.

3.1.4. The double line

As soon as the best configuration has been outlined for each pump model, we observed that a large margin for the UC_S remains.

Under the last configuration it's very hard to image gaining further improvements. The idea was to evaluate a new balancing opportunity based on a double coupled line, a solution that goes to break the layout constrain, but looks as the only opportunity to start a mixed model balancing policy.

In fact, because of our specific constrains position, all station time profile look very similar and the idea to test any balancing strategy based on scheduling distinct model groups to achieve balancing benefits, since the preliminary observation seems to fail.

On the opposite, it seems possible have a good match, fitting station profile in a counterflowed way, or just considering a single item matched with its self.

Any way, in more general case of distinct constrained stations and tasks, no limitation arises to improve a single or parallel direct flow mixing model balancing strategy.

The possibility of obtaining a clear improvement raises up, under a profile of efficiency and cost, as well as from a point of view of the balance of the loads assigned to resources.

The idea is to abstract from the system under examination, while keeping it as a reference for the data for the model. No investment costs are available for an investment analysis, but if the double line should be proved to be an optimal solution, there are no serious limits to its realization, as the unavoidable costs will be offset by savings in production.

In the new layout, a new twin line will be realized, disposed parallel to the original one, but with the stations arranged in a totally symmetrical way, for example, the station 1 on line 1, it will match to 6 on the second line, the station 2 in the first 5 in line with the second and so on, keeping the number of fixed resources.

In the analysis of the resulting output from the model with the solutions previously applied, some of the stations, in fact, had still space for further optimization because of the presence of stations not well filled, due to the presence of constraints and to the number and features of tasks in which it was divided assembly. To define best configurations and matches to set for the new algorithm implementation, further tests to try to needed to be conducted. In fact, this configuration requires that distinct pump models could be produced simultaneously.

To determine the best matches between pumps we used simulation software.

By observing results of optimization phase for the single line case, it pops up that there were lines of production plan proposed for the double line. We defined multiple sheets to be read in the program with the following additional information with respect to single - line:

- Production line value: indicates the line the item will be assembled in, first or second.
- Task time: The times are the same as single line, but for item that will result to be assembled on second line, tasks are renumbered in a symmetrical manner, ie, the task number 34, becomes the number 1, number 2, 33, and so on.
- Number of matches: number of rows, subsequent to the current one, who will attempt the match with it.
- New Reference Tack time: average of all coupled items: being twos at a time, the actual value of this will be the sum of the 68 tasks of the two lines, divided by the number of resources (6).

The parameters just defined for each row are then placed into the new files to read in order to verify combinations with all others. Each files show in the first line the data of product to be assembled on the first line, and than those for all others to score the match performances.

The aim is thus to compare each row with the others and define, based on the goodness of each match with the other lines, define for each order line a priority list of coupling with the other in order to improve the joint efficiency.

In order to achieve the best balancing of the line, obtaining a better efficiency and the minimization of costs, a new heuristics has been structured, then checked for the distribution of tasks to the stations of two lines in a dynamic manner, the structure of which will be verified later.

We can observe, first, that it was decided to couple the line 2 to the first in a antisymmetric way. So, corresponding to the entrance of the pumps on line 1, is the exit of the assembled pumps in line 2. The number of operators is six, and each operator works on both lines turning him self of 180° and advancing few steps, the possibility of movement is then guaranteed.

In the tasks assigning process, stations will be considered coupled. We will, therefore, face with a pair of stations, and tasks will be assigned to a numbered operator, rather than to station.

The chamfer machine has now been moved in the middle of lines space corresponding to the station 3 and 4, available for the two operators who have to use this resource: the operator 4 for the first line and the operator 3 for the second.

The task assignment to stations will be based on the operator scheduled to, as tasks of the two order lines are available on the two opposing lines to be allocated to the two stations under operator jurisdiction, respecting each precedence diagram.

The reference total tack time is calculated as average for all tasks of coupled items, or as the sum of the ideal tack times of the lines that try the match.

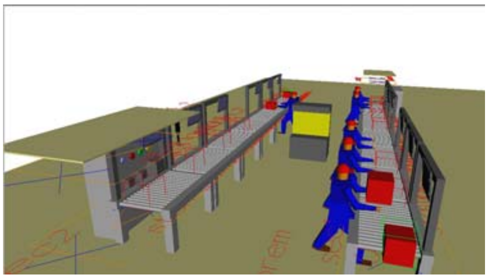


Fig. 7: snapshot of the double assembly line layout

3.1.5. Definition and implementation of heuristic for the double line

In the first part or preliminary phase of the double line algorithm, a pair of rows at a time will be read at the time, considering the first, as the leading one, to be produced in line 1, and the second one, instead, to be tested as mate item on the second line.

At this point, we start to assign tasks to the first operator, or rather the first pair of stations (in our case St_1.1 – St_6.2 where the second index refers to the line on which you are).

The algorithm logic for the allocation, considers that, for each coupled stations/operator, allocation starts with the station, if exists, with constrained tasks till constrained task has been allocated; since that, tasks are allocated alternately to both coupled stations from those of matched order rows, till reference times has got, in respect of priorities.

If the assignment of the task can not be made because of the tack time reference level is exceeded, or if is not possible to assign all tasks within the available stations, an increment of reference Tack Time starts, and the assignment has be done again, as previously.

As the assignment is completed, all the values for tack times, efficiency and cost for subsequent comparison are memorized, and the dominant line with his next match is tested, till the end of the file.

At this point, obtained all the possible configurations for the match with the dominant line, is possible to compare the parameters of best matches with the winning ones in the single line case.

Direct cost and inefficiency, are calculated as in the single line case, just considering the cumulative coefficients of imbalance and a tack time, meaning the added value for the two models matched in the double line.

The improvement combinations are sorted in descending order of saving costs, and this defines the priority matching lists, available to be used for any production plan. Next matching criterion is the production volume.

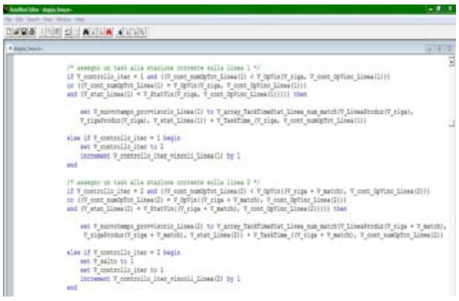


Fig. 8: logic for the allocation of alternate operations

For example, if there is an order which requires the production of a different number of pumps belonging to different families and with various optional and drives, by consulting the list of priority can be known, fixing a rule to choose the model to be produced in line 1, which of the remaining ones, mates better produced on line 2.

At this point, one ore more criteria to define the order to assembly in line 1, was requested to combine these results at our disposal, with a specific PP, to obtain a suitable situation to our case to test.

The possible criterions to use priority lists, were:

to respect lean and FIFO logic, was to consider the list position of the rows of the production plan, considering to match to the first matching available one, as it were the time succession of the arrival of the various orders and then define a logic in which each not yet scheduled row, was matched with available sequent ones, under the respect of its priority list, till the quantity ending.

If, however, there is the possibility of delaying the production start of an order, it is possible decide to wait for the arrival of the one that best mates with it, or it can wait until it reaches a number of orders such that their production volume equals that of the order on hold.

Some control code adjusts the emulation of the assembly process, coupled and phased, of the production plan limited to the order lines suitable to be processed in a double line, on the basis of attributes further introduced to manage the routing.

Control code protect the simulation from the possibility that on one of two lines new orders could entry the system if coupled order have been not completed already. The cadence of work considers the operator to work alternately on both coupled items before start a new item following on one of lines.

This has been realized through cross - inhibitions and unlock.

3.2. ANALYSIS AND COMPARISON OF EXPERIMENTAL RESULTS

In the following lines some algorithm and simulation application have been outlined.

Single Line

First, in table 3 it is possible to observe the total absence, in the best configuration case, of stations under-utilized.

It’s easy understand that their presence affect the best performances of balancing.

Second, we can see that the stations, under charged at the first instance, have been permanently depleted and erased, as those already empty since first tasks allocation. This means that the final winning situations appear to be those which have a lower number of resources, except in those cases in which stations are all well filled.

Table 4 shows as the final configuration results in an overall improvement in the efficiency and cost. There is no worsening in the final configuration, in fact, but at least, values remain the same. For those rows, it means that the best configuration is the first, the one obtained after the first dynamic allocation. Any way, our new dynamic assignment yield a large improvement for cost and efficiency values, if we should compare these to those corresponding to the first situation, namely that provided by the company, thus obtaining a positive result.

Table 2: unbalance coefficients after dynamic allocation process. In green stations with the maximum coefficient, in orange under charged stations, and red empty ones.

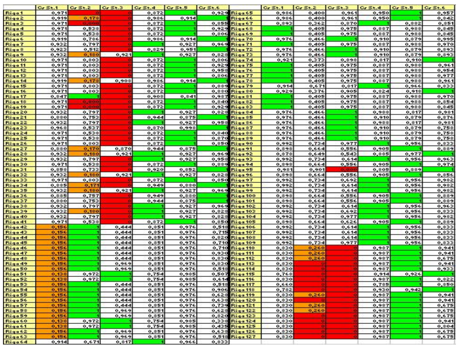


Table 3: UC_L for single line best configuration. In green stations with the maximum coefficient, in orange under charged stations, and red empty ones.

From Table 5 it is evident that the winning configurations are Config_1, corresponding to the initial dynamic distribution, and Config_3, when reference tack time is amplified, and tasks of the under charged station, are reassigned. Before continuing the analysis, it is good to declare that since the Config_2, with operators uncharged since from the beginning, reassigned to overcharged stations, is never successful, it will not be shown in the following charts, as data relating to the configuration number 4.

In other words, since there are no lines which pass configuration 2, there are no rows that will attempt to use the number 4.

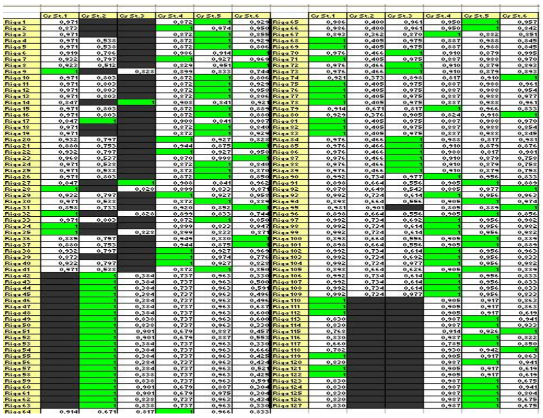


Table 4: Global Efficiency and Whole Direct Assembly Cost, Config_1 vs final best configuration. In green, rows showing an improvement.

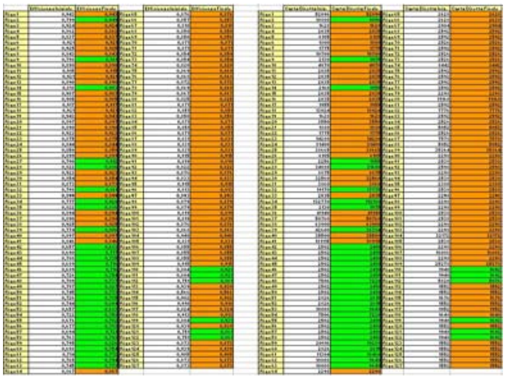


Table 5: number of winning configurations for all lines.

Conf_1	Conf_2	Conf_3	Conf_4	Conf_5
90	0	37	0	0

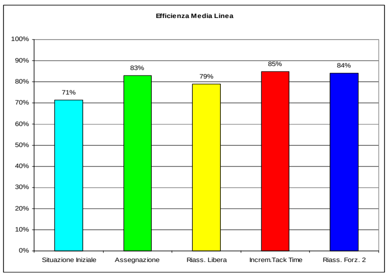


Fig. 9: Average Efficiency for all single line configurations

The efficiency of the line after the first dynamic reassignment grows with a significant increase of 12 percentage points, which leads to 83%.

The following phase, knew as Config_2, with the redistribution of the operators will cause a decrease, while, again, in the Config_3, the efficiency decrease is compensated with the increase of the tack time, that makes possible the elimination of the undercharged - stations which lead to a vertex efficiency of 85%, no more over passed also by Config_4, when redistribution of released workers, then records a decrease, albeit minimal.

But, although the final situation, Config_4, is typically better, from this point of view, if we observe the objective function, the development of costs, it can be observed that it is no more convenient. Since that the objective is to minimize costs, much more than increase efficiency, Config_2 is winning on Config_5.

The cost trend is still fairly close to that one of efficiency, this trend is shown in Figure 9, while in the following figures are the changes in costs and the maximum decrease achieved in various configurations.

The Average Direct assembly Cost undergoes a substantial decrease, both in the case of dynamic allocation, equal to 16.7%, both in the case of depletion of the stations under - utilized, when compared to the initial situation.

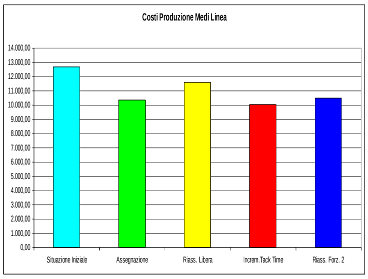


Fig. 10: average Direct assembly Cost for all single line configurations

The reallocation of workers leeds to a fall, but is much less important, being equal to 2.9%.

The cost of production was calculated in such a way as to be representative of both the tack time reference of the line for each row, and then for each model, both the number of resources used, as well as the production volume:

$$
\begin{aligned}
&Direct\_Assembly\_Cost\\
&DAC = (Tm \bullet S \bullet LQ \bullet ManWorkCost)\\
&\text{where}\\
&LQ = \text{LotQuantity}
\end{aligned}
\qquad (3)
$$

Another evidence is that there is a general tendency that are Station_2 and Station_3 to get empty when we are working to kill under charged stations, indicative of the fact that they appear to be the less loaded. Station_3 has a tendency to be empty already in the case of the first dynamic assignment.

Given the trend of the last three stations, which is almost the same in various assignments, it's possible justify the initial drop of operations on the station 2, in fact offset by the significant increase in allocations to station 1, and this trend is also indicative of the constrained station that creates a bonding allocation in station 4.

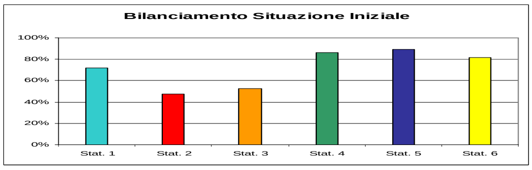


Fig. 10: Initial configuration, Average Station unbalance coefficients, UC_S.

The initial situation presents a quite low balancing level, especially as regards to stations 2 and 3.

All subsequent situations lead to an improvement in percentage for the station, but do not entail a better balance, ie a situation in here there is balance between the various coefficients that should settle to values similar to each other and quite high.

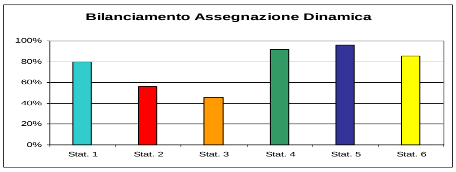


Fig. 11: Config_1, Average UC_S.

We have seen earlier that the resignation of the operators, with subsequent lowering of the tack time, never leads to an advantage compared to the situation where not schedule one or more resources, keeping the same tack time.

Whole UCs representation for each of all available station as in the initial configuration, any way, is not really well representative, in fact, we are in presence of situations with a variable number of /stations/operators, and therefore, it will be better outline results based on the number of remaining stations/operators.

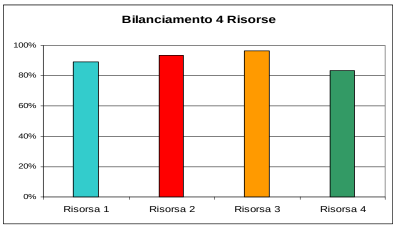


Fig. 12: Final configuration, 4 resources Average Station unbalance coefficients, UC_S.

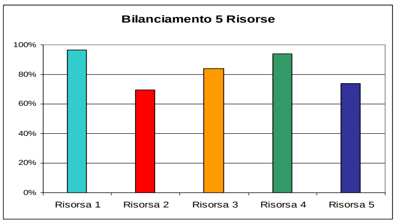


Fig. 13: Final configuration, 5 resources UC_S.

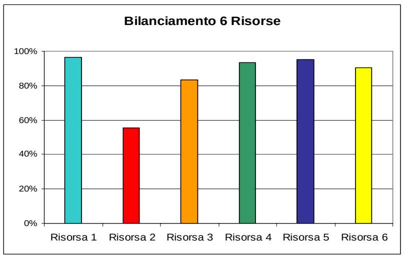


Fig. 14: Final configuration, 6 resources UC$_S$.

It is now more evident the achieved advantage with the proposed configurations. In cases with a greater number of resources, however, it is noted still the possibility to have an improvement, especially for what concerns the second resource, as the case corresponding to the stations 2 or 3.

In general, however, UC$_S$ coefficients range within a fairly narrow range and values placed around between 70% and 80%.

3.2.1. The analysis of the double line

The double line configurations are all 6 resources.

Two main ranges:

- 0.304 - 0.568: light green, more unbalanced stations;
- 0.568 - 0.7: dark green, less unbalanced stations.

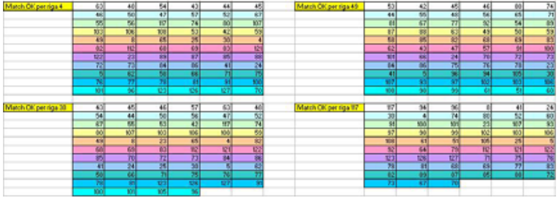
To obtain these ranges, an interval that would go from the minimum value of the coefficients of unbalance (0.304) to a maximum value, chosen by us, equal to 0.7, is considered.

Several possible scenarios were examined to define matches, tested all using the simulator. This solution proved to be the best under speed and efficiency of calculation point of view.

The scheduled lines to be assembled in the double line are 79.

The priorities lists so defined are presented in Table 6.

Table 6: an example of priority list


Match OK per riga 4

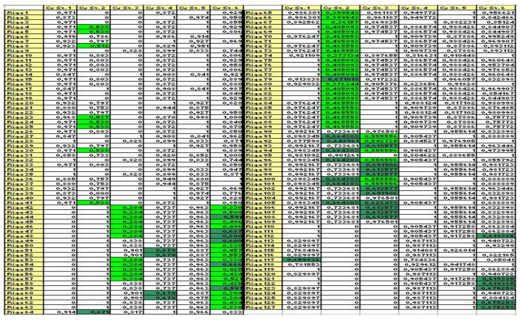
Match OK per riga 49

Match OK per riga 38

Match OK per riga 117

As the priority list were defined, two defined production plans, extract from the original one just for matching lines, one defined to match the higher-volume lines to all other lines, and one defined to give the leading role to test matches with all other lines, to the rows in the topmost position, as indicative of the sequence of arrival of orders, were tested.

Table 7: final UC$_S$; in different shades of green worse values.



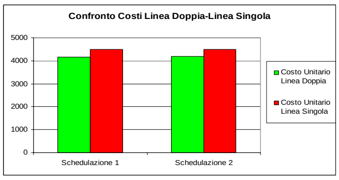


Fig. 14: Comparison of costs in a single line and double line for the two production plans.

As shown by the graph, with the double line is achieved a reduction of costs to produce the same plane vs the single line. Is also observed a higher gain (7.2%) with the first type of scheduling compared to the second (6.8%) even if the difference between the two is minimum, equal to 0.4%.

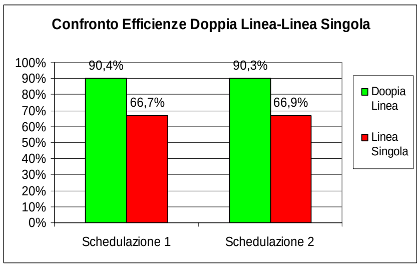


Fig. 15: Comparison of the efficiencies double line vs. single line under two different schedules.

The efficiency implies a remarkable increase, also. The values for both types of scheduling are very good, and the increase compared to the single line best configurations, is 35.5% for the first schedule, while 35% for the second.

The improvement is made even more visible from the observation of the graphs for the UC$_S$.

With the double line, in both schedules, it is possible observe a very good balance and the range within which falls the various coefficients are very narrow and with excellent values, being included between 85% and 97%.

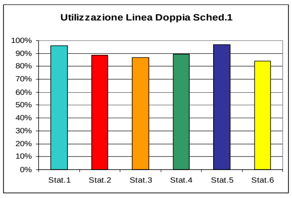


Fig. 15: UC_S for the double line in first scheduling.

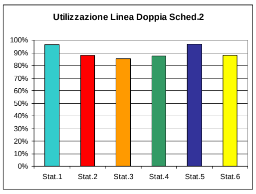


Fig. 15: UC_S for the double line in second scheduling.

3.3. CONCLUSIONS

In this paper a thoroughly research was conducted regarding possible improvements to be applied to the case of an manual assembly line for hydraulic devices.

In this work were identified and then addressed the most critical issues, through an multi phase algorithm definition and consequent simulation of the process.

Preliminarily, we have conducted analyzes that could lead to the definition of the necessary elements to optimize the load balancing of the line, as well as to reduce costs and increase efficiency.

On the basis of data provided by the company, of the imposed constraints and the criticalities, in general, are then created simulation models, which showed the system behaviour depending on the logic strategy applied in each phase.

The aim of this study was to define a global strategy to apply o a wide range of assembling systems, to optimize production.

Therefore, all the relevant parameters have been identified, in order to define all the possible achieved improvements, ie the best balance of stations time distributed on the line, as a certain balance among between the workloads assigned to the operators and avoiding the lack of homogeneity among the various resources, the minimization of production costs of products mix, ie. the maximization of the efficiency.

All strategies have been defined respecting any of the main constraints and considering an appropriate volume production, which could give validity to the model.

The execution of this work has begun with an analysis of manual assembly line specific question, for what regards its characteristics and peculiarities.

Then, a cascade of ameliorative approaches were evaluated, structured as algorithms and heuristics so that they could then translate into a programming language for the implementation and verification of their actual goodness, to the computer.

In conclusion, as verified, considerable improvements in those that were indicated as the objective functions of this work have been got, ie the balance and the production costs (or efficiency), resulting in an improvement in productivity.

Another goal that can be considered reached, the definition of a model that was as versatile as possible. Thanks to the type of structured algorithm and its consequent translation logic programming software, the model can, with its shape, to adapt to other situations of lines of manual assembly with similar characteristics.

A good improvement is definitely the tasks dynamic assignment, useful to assign operations in a flexible way and fully accomplishing with the constraints, structural ones or otherwise, imposed by production statements.

The layout changes for the double line, while going to change the physical structure of the system, and leading to an investment of time and money, can still be considered valid since the benefit obtained.

Thanks to the creation of lists of priorities you have in fact a useful and powerful tool that allows you to go to produce combinations of multiple products minimizing down costs and gain efficiency, and just only when needed, being able to choose among multiple strategies to define double line assembly plans configurations, depending on to ways in which the company prefers replay to market requests and orders. As already mentioned, the costs for the extension can still be reduced by acquiring simple systems to share equipments.

The first part of the study had been inherited from a previous research, much more simple, where the improving strategy has been developed on a spreadsheet.

Future subsequent optimization approaches could include a new data collection and the variation of data of the system randomly with logic, to have a greater validation of the algorithm.

It could be possible also refer to an advancement of multiple products simultaneously on the same line, similar to what we saw in the last part of this work, but without another line, but by simply extending the existing one, with U-Shaped layout, and evaluate, a scheduling strategy but on the double of the stations, with possible assignment of stations even at the same operators, in order to obtain a greater opportunity to balance based on the scheduling of resources, but also to be able to feed as a double line, alternately.

This opportunity is under evaluation.

Finally you could structure the analyzes concerning the study of the cost of any delays on deliveries or completion of the off-line products, evaluating solutions for the optimization of these parameters and the creation of configurations can prevent the emergence of such issues.

In general, the use of solution approaches such as algorithms, heuristics, simulation techniques and classical analytical techniques, opens up a panorama of endless possibilities, which allow to deduce a solution or, more often, a set solutions to the problem to improve the present situation. As there is no best technique, the goodness of one technique over another, is a function of the objectives to be pursued, and the results that we are provided, as well as the proper structuring of the preliminary steps in the analytical process.

REFERENCES

Amen, M., 2000. Heuristic methods for cost-oriented assembly line balancing: *A survey. International Journal of Production Economics 68*, 1–14.

Amen, M., 2001. Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time. *International Journal of ProductionEconomics 69*, 255–264.

Baybars, I., 1986. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science 32*, 09–932.

Becker, C., Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research 168*, 694–715.

Chutima, P. Suphapruksapongse, H., 2004. Practical Assembly-Line Balancing in a Monitor Manufacturing Company, *Tharnmasat Int. J. Sc. Tech.*, Vol. 9, No. 2

Ghosh, 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*, 637–670.

Erel, E., Sarin, S.C., 1998. A survey of the assembly line balancing procedures. *Production Planning and Control 9*, 414–434.

Gökçen, H K. Ağpak, R. 2006. "Balancing of Parallel Assembly Lines", International Journal of Production Economics.

Kottas, J.F., Lau, H.S., 1973. A cost oriented approach to stochastic line balancing. *AIIE Transactions 5*, 164–171.

Kottas, J.F., Lau, H.S., 1976. A total operating cost model for paced lines with stochastic task times. *AIIE Transactions 8*, 234–240.

Kottas, J.F., Lau, H.S., 1981. A stochastic line balancing procedure. *International Journal of Production Research 19*, 177–193.

Jolai, F., Jahangoshai REZAEE M., Vazifeh, A. 2008. Multi-Criteria Decision Making for Assembly Line Balancing, *Springer Science Business Media*.

Moodie, C.L., Young, H.H., 1965. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *Journal of Industrial Engineering 16*, 23–29.

Pierreval, H., Caux, C., Paris, J.L., Viguier, F., 2003. Evolutionary approaches to the design and organization of manufacturing systems. *Computers & Industrial Engineering 44*, 339–364.

Raouf, A., Tsui, C.L., 1982. A new method for assembly line balancing having stochastic work elements. *Computers & Industrial Engineering 6*, 131–148.

Rekiek, B., 2000. Design of assembly lines. Memoire presente en vue de l'obtention du grade de docteur en sciences appliquees. *Universite libre de Bruxelles*, Brussels, Belgium.

Rekiek, B., De Lit, P., Delchambre, A., 2002. Hybrid assembly line design and user's preferences. *International Journal of Production Research 40*, 1095–1111.

Salveson, M. E., 1955. The assembly line balancing problem. *Journal of Industrial Engineering* 6, 18–25.

Sarin, S.C., Erel, E., Dar-El, E.M., 1999. A methodology for solving single-model, stochastic assembly line balancing problem. *OMEGA—The International Journal of Management Science 27*, 525–535.

Scholl, A., 1995. Balancing and Sequencing of Assembly Lines. *Physica-Verlag, Heildelberg*.

Scholl, A., Boysen, N., 2009. Designing Parallel Assembly Lines with Split Workplaces: Model and Optimization Procedure. *International Journal of Production Economics*.

Silverman, F.N., Carter, J.C., 1986. A cost-based methodology for stochastic line balancing with intermittent line stoppages. *Management Science 32*, 455–463.

Sculli, D., 1979. Dynamic aspects of line balancing. *OMEGA— The International Journal of Management Science 7*, 557–561.

Sculli, D., 1984. Short term adjustments to production lines. *Computers & Industrial Engineering 8*, 53–63.

Shtub, A., 1984. The effect of incompletion cost on the line balancing with multiple manning of work stations. *International Journal of Production Research 22*, 235–245.

Süer G.A., 1998. Designing Parallel Assembly Lines, *Industrial Engineering Department*, University of Puerto Rico-Mayagüez.

Suresh, G., Sahu, S., 1994. Stochastic assembly line balancing using simulated annealing. *International Journal of Production Research 32*, 1801–1810.

Suresh, G., Vinod, V.V., Sahu, S., 1996. A genetic algorithm for assembly line balancing. *Production Planning & Control 7*, 38–46.

Van Oyen, M.P., Gel, E.S., Hopp, W.J., 2001. Performance opportunity for workforce agility in collaborative and noncollaborative work systems. *IIE Transactions 33*, 761–777.

Vrat, P., Virani, A., 1976. A cost model for optimal mix of balanced stochastic assembly line and the modular assembly system for a customer oriented production system. *International Journal of Production Research 14*, 445–463.