SIMULATION MODELS TO SUPPORT GALB HEURISTIC OPTIMIZATION ALGORITHMS BASED ON RESOURCE BALANCING BASED ON MULTI OBJECTIVE PERFORMANCE INDEX

Sergio Amedeo Gallo[(a)], Giovanni Davoli[(b)], Andrea Govoni[(c)], Riccardo Melloni[(d)]

[(a, b, c, d)]Department of Engineering "Enzo Ferrari" (ex DIMeC),University of Modena and Reggio Emilia, ITALY

[(a)]sgallo@unimore.it, [(b)]giovanni.davoli@unimore.it, [(c)]andrea.govoni@unimore.it, [(d)]riccardo.melloni@unimore.it

ABSTRACT
The following paper face with an approach to analyse a multi model manual assembly line, and the following heuristic algorithm to optimize the scheduling of tasks to the available stations, respecting of a set of restrictions, as task/station obligation, and aiming to optimize a multi objective function, based on time, balancing utilization rates, 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 referable configuration of the considered system is an assembly line with six stations, and with six operators.
In the present step, the solution we experimented, shows a redundant (doubled) number of stations, to be a more flexible solution compared to previous solutions, with a layout too much specific and referred to the particular tasks and constrains profile, based on real data.

Keywords: workstation design, work measurement, ergonomics, decision support system

1. INTRODUCTION

As in previous works, in following lines we describe main models and system features.

We face with a manual assembly lines.

Items advancement on line is done on a accumulating conveyor system, so **line** is **paced**, but **not synchronized**. Just a single accumulation among stations, is allowed, in the original system. This opportunity makes the flexibility level higher, but we have to consider no inter operational buffer.

Assembly line process a very large variety of items, defined in families, 6 in the original Assembly Plan, (**AP**), that differ for size, features, optional, lot size. Spurred by the increasing market competition, and customers' requirements, are commonly accepted very low quantity for single order. Tasks assignment to stations must respect efficiency targets, as the maximization of utilization rate, balancing concerns, assignment constrains restrictions of specific tasks to specific stations where special machines are available. This constrains are commons for all items of the mix.

Assembly plan data, **AP** Data, and system configuration arise from those of a real assembly line. Task times are stochastically distributed based on real observations. The influence of stochasticity is not the focus in the present scheduling problem, because no cost for off line completion can be considered, and because tasks that do not respect line cycle time, cause just the tack time increase of the single item, but not of the mean of performed Tack Time for the whole lot.

No incompletion costs and operators moving cost are considered, because both negligible compared to operating cost, both because of the short distance to cover moreover, no costs related to operators training, or both changes of the tasks to operate, has been considered, because, we are facing with a Multi Model Assembly Line, and, furthermore, operators are supported with displays showing instructions for tasks to operate.

The same tasks of different items, has operating times, that can vary for each item, for the operation declared in the same way. All tasks in the whole annual assembly mix in AP are represented on the precedence diagram, uniform for all items. To assembly a model not all the 34 operations are needed for a specific item, depending on features and optional. Each item has a defined number of operation which ID number increase as the assembly process goes on.

The performance parameters are the production rate, to be maximized, that means to reduce tack time, and, at the same time, optimize the internal balancing of both tasks for stations, both labour level among operators. Based on these criteria, a multi objective function with the aim to minimize the whole lot assembly cost, calculated on the effective tack time and on the current 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.

The heuristic algorithm is based, at each step on a logic trying to improve the previous balancing configuration.

We built a virtual model of the assembly line in a simulation environment, to test and measure performances of the heuristic algorithms, but, 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.

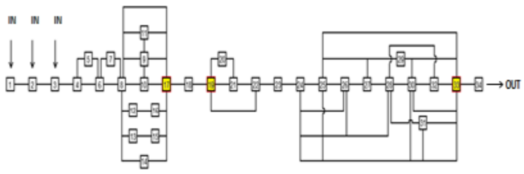

Fig. 1: combinatorial diagram of sequences

The definition of the scheduling in the AP is oriented to the lean production philosophy: first production order has to be produced first, too. Anyway, at least, one week, is the planning time horizon an order can be shifted inside the scheduling window, without affect due dates. So, it is possible over pass the rule to route assembly orders as they were placed in the order list, to get better system performances.

Precedent models have been developed. The first one represents the "as is" configuration of the firm for strategies and configurations, is a basic model to be compared with our improved one. It was also used to verify and validate the virtual representation comparing to available real data. Also any implicit scheduling and assigning rule has been checked and verified.

The following models, last ones before the present one, Gallo and al. (2012), is in many parts similar to the present. So it seems appropriate to recall some feature and logic.

They were based on task assigning rules, attempting to fulfil stations adding tasks to, in sequence, till total station time doesn't overpass Reference Tack Time, **RTT**, calculated for each Order Line, **OL**, and item. Moreover, additional control code was devoted to check if any constrained task is joined, and in this case, provide to verify the station where to assign that task, if not the current one, and to calculate all parameters for intermediate stations.

Additional rules evaluated if all already scheduled operators were idle, and, if not, the algorithm attempted to re allocate the idle ones to more suitable stations. Again, all new values of parameters were calculated and compared to the old ones, as the objective cost function. Just the best allocation survived and recorded.

Moreover, if some station was undercharged, a routine incremented recursively the tack time, till all those stations became empties, so that released operators could be re - assigned to over charged station.

The constraint position of a special chamfer machine, allocated on a defined station, and the assignment constrains of other operations, to stations where other equipment is available, dramatically limited opportunities to gain better performances based on balancing the line with a mixed sequence of selected groups of items extracted from the AP, conveniently defined for quantity and for typology, and seemed, not feasible (balancing on scheduling).

Finally, in the last previous models, to improve performance, based on residual efficiency edges, i.e., on the complementary values of the SUCs, was defined a double line to assemble coupled mixed items, contemporary, each one on one of two lines. This kind of configuration was based on some aspect of constrains position, and on the specific AP data.

The opportunity we found to achieve better performances, grouping single units of different available orders, was of **coupling tasks to be assembled**, on two **parallel lines**, fed in counterflow. A single operator should be assigned to corresponding stations, one on the first line and one on the parallel line, and they should have to complete their operations, alternatively, on both stations of two lines. An improved performance was gained, and was possible operate more assembling strategies.

At this time, a very relevant observation raised in our mind: the parallel coupled line can be a general and versatile solution when applied to more general data sets and different constrains position and configuration?

NO, of course!

In fact, the previous proposed solution to balance the line, was too much based on the specific constrains and available data. Profiles of Stations Utilization Rates, SUCs resulted of similar shape for the large part of available items, differing, often, just for the scale of the RTT. Also, the need to find good matchable items, close enough in the AP, is not sure to face with, also considering that the amount to produce, or the correspondent assembly time, should be quite correspondent. On the opposite, experimenting some heuristic rule to assign to same operators more than one station charge, seems more likely hopeful, especially watching at some item, with residual edges for resources SUCs, no further more improvable because of line configuration, and constrains accomplishment.

So the gauntlet to build a more general, flexible, versatile system configuration, with the associate heuristic logic strategies, was picked up, and challenge started.

1.1. Present models configuration

The strategies to distribute tasks to stations are similar to the previous models.

Also in the present one, task assignment to station has to respect efficiency concerns, as the maximization and the balancing of utilization rate, but, first, specific task assignment restrictions, because of the need of special machines, available at defined station.

The initial heuristic strategy, **config_1,** try to assign tasks in sequence to stations, till the RTT is fulfilled, or till total station time doesn't overpass calculated referable task time. An additional control logic to verify if a specific task is one constrained to be assigned to a specific station, and in case, the consequent logic to point to the correct station, and to calculate all parameters for intermediate stations, is present, also.

In **config_2**, it is present a logic to evaluate if there is some unused resource, and in this case, the heuristics reassign to the more overcharged stations to help the more suitable stations already assigned.

Again, in **config_3**, it is present a logic to evaluate if, after the first assignment, some undercharged station/operator results. The charging level is defined as a the percentage ratio of the current RTT. A routine defines a RTT recursive increment till, all the undercharged station become empties, so freed operators could be reassigned to over charged stations, as in the previous mentioned case, **config_4**, and **config_5**.

Simulation models can read AP data form a CSV file, with any useful attribute to be used to characterize the specific OL, and the configuration: time distribution parameters, item definition, order quantity, order date, etc.. In this way, is very easy to change configuration.

Any time a new strategy is applied, all performance parameters, as station/operators utilization coefficients, UCs, Direct Assembly Cost, are calculated, stored and compared to best performing configuration emerged at the previous step. Just the better, for each item, is the one considered for the final solution.

But, considering that in the previous model, in the case of the single line, at the end of the cascade of logic steps, nothing more to achieve a better performance was possible, based on work assignment balancing strategy, the current new strategy was considered. The present idea is to increase the number of the stations, with the aim of enforce balancing opportunities based on the resource/work balance.

We decide to create two stations where before one, to have a more relaxed assigning opportunity. Also the constrains position at specific stations has been doubled, to keep proportion with the original system:

- line configuration, the number of available stations, the equal number of assigned operators has been doubled to 12.
- Any equipment available at a constrained i-th station, has been located at 2 * i-th station.
- The tacks assignment logics in the heuristic is the same.
- At the end of the logic cascade, when no further opportunity to achieve a better balancing based on works contents of stations, an adding logic starts to calculates the maximum value of the station time, in other words, the line tack time, and searches to find stations which work load have a sum equal to the current line tack time for each order.

After any assignment strategy was tried, after the reallocation of idle or freed operators to the currently overcharged stations, in the respect of the constrains, it is possible assign operators more than one stations.

To support this opportunity, a **U shape line** has been considered. A display on the top of any station shows which is the current item, which is the mounting cycle and parts, and if and where to go. Not all of initial 12 stations and operators, will be scheduled as final optimized configuration. For each item or OL just one configuration is the best, with specific number of stations, and operators, too.

Models can be applied to a wide variety of systems, with different number of stations, and different constrains positions, just integrating new additional constrains rules based on new configuration values. Data have been those arising from the real system.

Achieved results showed good improvements compared with initial solutions, and any time a new strategy was applied.

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.

Their configuration and planning is relevant both as a optimization problem both because they are systems at medium intensive capital.

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 covers both all tasks allocation and both decisions 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. The high level of automation of assembly systems and the fixed movement system make the (re)-configuration of an assembly line critical.

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

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 simplifying assumptions of this basic problem, this problem was labelled simple assembly line balancing (**SALB**) in the universally 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) defined 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, relevant incompletion costs often occur in stochastic assembly lines.

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, 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

We though for long time to define the correct number of stations in the new systems.

For an assembly line the station number can range between one - a line degenerates in one only station, perfectly balanced - to **N,** where **N** is the total number of tasks the assembly process has been divided in. A low number don't offer a large opportunity to have a sufficient space for a resource balancing, instead, a too large number enforce the indetermination of the configuration to test.

Then, we considered to define a line with a cycle time quite equal to the half of the Ideal Tack Time in the six station configuration. The new system keeps the proportion with the previous. We just tried to "dilate" the previous system and achieve an "**homothetic**" increased system, with an opportunity chances to rebalance the line based on resources.

We have the following constrains:

- At 8th station we have a chamfer machine, the strongest constrain, and task 17 (chamfering).
- At 9th station there is a pneumatic screw driver, and task 19 (screwing).
- At 12th station we find the equipment to apply the air test, and task 33 (air testing machine).

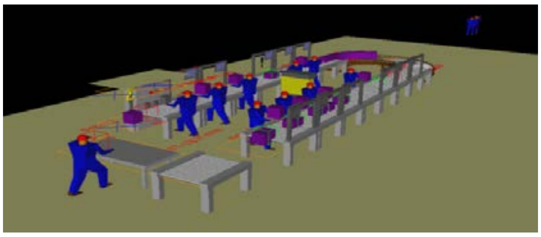

Fig. 2: screenshot of the doubled assembly line.

Execution times vary strongly, and can be described with lognormal o triangular distributions; in our case are described by triangular density functions with a large extension.

Model parameters (times in hundredth of minute):

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

Task Number $n \in [1, 34]$ (2)

N° of tasks assigned to a single station

$$i \in [1, h] \quad (3)$$

Task Time Top (4)

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

$$T_{Stat} = SUC * RTT \quad (5)$$

Operation Unbalancing Coefficient

$$UC_{Op} = \frac{T_{Op}}{Tm} \quad (6)$$

Station Unbalancing Coefficient

$$SUC = \frac{T_{Stat}}{TT_{Line}} \%$$

$$SUC = \sum_i UC_{Op} = \frac{\sum_i T_{Op}}{TT_{Line}} \quad (7)$$

Line Lead Time $LLT = \sum_{i=1}^{k} T_{Stat}$ (8)

Line Tack Time or Cycle Time

$$LTT = Max(\sum_{1=i}^{h} T_{Op}) = Max(T_{Stat}) \quad (9)$$

Mean/Ideal Tack Time

$$ITT = \frac{\sum_{1=1}^{n} T_{Op}}{k} \quad (10)$$

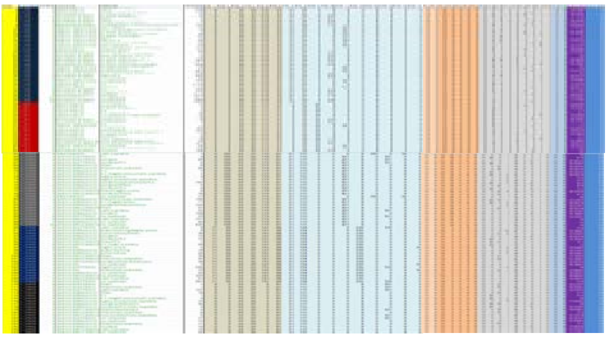

Figure 3: whole mix production plan with parameters values and task times.

3.1. MALB algorithm

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

We define a cost function, Direct Assembly Cost, DAC, as the product of the manpower direct cost, multiplied by the station number (or operators when more than one is assigned to a station), multiplied by the volume for the OL, multiplied by RTT, to be representative of both the RTT of the line for each row, and for each model, both the number of resources used:

$$Direct\_Assembly\_Cost$$
$$DAC = (RTT \bullet ResNum \bullet Lot\ Quantity \bullet Man\ Work\ Cost) \quad (11)$$

Our heuristic algorithm is a mix of Work Content and Resource Balancing, that, with the objective function, takes their role and weight, very freely.

We will configure our situation as a **MALB-E** problem, given number of K stations, the aim is to maximize the efficiency E_{line}, i.e. minimizing the direct cost of assembling the lot.

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

First heuristic logic, called **config_1**, try to assign and redistribute tasks to stations in dynamic and balanced way, under the respect of all constrains, as the sequence diagram, with the aim of minimize a whole cost function, an objective function, and both to increase the Efficiency and the Balancing Level.

Line will result better balanced when maximized

$$MAX(E_{linea})\ where\ E_{linea} = \frac{\sum_i UC_{stat}}{\sum_i S} \quad (11)$$

Efficiency is calculated as sum of all SUC divided by current number of stations defined for each item, and, at the starting time, equal to resource number, then as we maximize efficiency, is maximized utilization rate and is minimized UC's for each stations.

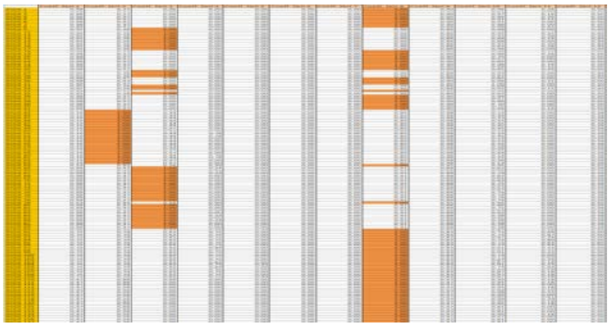

Figure 4: Unbalancing Coefficient for stations at the initial assignment configuration config_1.

After first dynamic task assignment phase, we can outline following considerations about task times.

One among stations 2, 3 or 8 is the one responsible for the line tack time, that is equal to RTT for each OL, because in this configuration, with 12 initial stations, there is more often a task time that overpass the ideal tack time for the line. Quite always, stations 6 and 7 results as empty, then needless. With a lower frequency, the same happens to stations 4 and 5, instead stations 8 and 9, that usually show high unbalancing coefficients, that is clear because those stations are constrained both.

We remember that, in the 6 stations configuration, usually, was the station number 5 to affect LTT.

Furthermore, station number 11 is charged but with a very lower SUC, ad in the following application of strategies, that station become discharged. Station number 12, also, has very low unbalancing coefficients, but unlike of the 11 ones, usually is charged with assigned tasks because there is the air test equipment.

Initially, each operator has been considered bounded to his station, and tasks allocation was made balancing on the content of work. The primary action the algorithm was designed for, is to allocate tasks dynamically to stations, with the aim of minimizing the DAC.

An important parameter has been RTT:

Reference Tack Time

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

Minimum time, in hundreds of minute, used as a limitation roof in the allocation of tasks, maximum threshold the Station Time cannot overcome. It is the higher value between the ideal tack time (**ITT**), and the maximum values of the specific various tasks (T_{OP}) for each line of the production plan. RTT 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.

Other parameter, already declared, is SUC, that is the percentage value calculated as the sum of the Unbalancing Coefficient of any operation assigned to the station. It's value is lower or equal of the RTT one, that is the 100%.

Another specific problem is the highly variable size of the tasks. Some of them, in fact, affect dramatically the Line Tack Time, and, when compared to the value of ITT, they are often even larger.

The increase of the original RTT, that is a read on a CSV file is of 5%. Any station value of the precedent step is put to zero, and again, is tried to reassign tasks to stations with the new RTT, until all tasks are assigned.

The simulation code will be used first to apply the heuristic rules and logic cascade, and later as a validation tool by testing any winning configuration with the emulation of the line. At any step all relevant parameters have been calculated and compared. Chutima P. et alter, (2004), Jolai F., et alter, (2008).

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.

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 SUCs and the parameters of cost and inefficiency. The logic routine dynamically assigns tasks to the stations respecting the allocation of joined tasks to the stations of belonging.

The file "***SpeadSheetWorkDataCSV_line_U***" contains information about a large number of parameters reported in array variables. In another reading file, "***ConfigurationCSV***", are the values of the variables to configure the system, such as the percentage increase value for RTT, the threshold value to divide tasks between two operators when more than one is assigned to a station, the percentage of tack time that defines when a station is under used, the limit value to accept an UC for the resources, and so on.

Moreover, other code portions manage the initialization of operators, of their disposal on the line, and to define the daily and weekly shifts.

As the reading process ends, the assignment process, "***P_allocation***", of tasks to stations starts. A control regards the constraints, in fact check if 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.

Another control checks whether or not, the addition of the task you are trying to give, does not lead to a Station Tack Time exceeding the reference limit. In this case, the station is closed and the allocation try 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 an OL is already processed, because all tasks are finished, the following row is considered till the last one in the AP. When you have no more tasks to be allocated, 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 RTT, as well as, the number of operations assigned to each station for each line, etc. are saved to array variables.

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

Now, we can observe yielded data and 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 uncharged, "**spontaneously**", are reassigned to the station with the highest tack time.

This last configuration (**config_2**) will be compared with the previous scenario (config_1), without considering resources associated with empty stations. In same case, is not possible allocate all uncharged operators because in some station there is just a long task: it doesn't make sense schedule two operators for just one task.

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 time.

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.

In the second one, all stations suitable to have a assigned another operator are defined.

In "**P_undercharged_Op_forced_assignment**" procedure, all the stations for each line are checked, to trace the presence of under used stations. Once that has been identified a station of this type, it is emptied, and its operator, released by force: RTT is increased by a defined percentage, all assignment values are placed to zero, a new tasks reallocation starts till under charged station is empty.

The tack time reduction follows a procedure that care of dividing tasks operators in the best balanced manner. The procedure is repeated until there are more operators to be assigned. In previous versions we divided station tasks time by two, and half was charged to one operator and half to another. In the present model, 12 stations instead of 6, there is an increased difficulty to share tasks between two operators, because of the lower number of them that could affect the error approximation, much more than before. Four sharing strategies are been introduced, that tray to assign tasks to each of operators, looking for the best one.

Successively, through conditional cycles "if…then", all results are compared and just the best sharing is chosen for each case and the maximum time of the operators become the station time, i.e. the closest to the 50% of the station tack time.

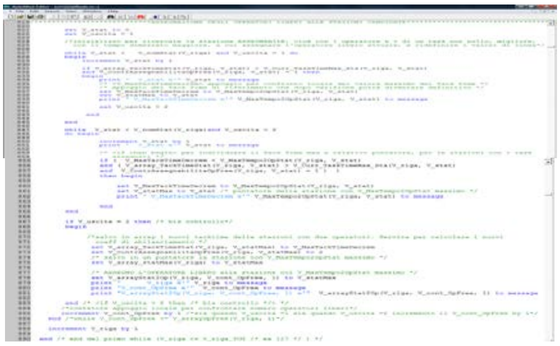

Fig. 5: logic to assign freed operators and to calculate new RTT snapshot.

A while loop choose the most decremented tack time among all stations that tried the assignment of another operator as the one we confirm, and the logic is applied till any freed operator be not assigned.

The assignment of operators to overcharged stations is done in three distinct steps: first one, called **config_3**, where in case of undercharged stations, RTT is increased recursively, and undercharged operators are just freed but not reassigned, **config_4** where just operators "spontaneously" uncharged are assigned, and then, in the **config_5**, also freed operators will be assigned again. In both strategies all logics to calculate new tack time, to define the station to help are similar, and, once again all the characteristic parameters of each situation will be saved for later comparison with those from previous situations in appropriate variables, with the same name distinct just for the suffix.

The highest tack time among all various stations, is saved for each OL as LTT

3.1.3. Resource balancing and the multi station assigning process

Just after first simulation runs, when the winning configurations for each OL, when no other opportunity to improve the balancing performance seemed possible, we tried to achieve an adding opportunity, the resource balancing.

On the same layout, for each OL we consider just the already final assignment configuration, and we tried to assign operators more than one station, in two distinct way.

In the first one, we apply a logic that calculate the RTT in the winning final configuration, **config_final**, and then, recursively, look for the station with the minimum value of the station time, not already considered. The algorithm try to match this station with the one that at this step shows the higher value for the station time with exception of the one that define the RTT. In case the station time sum doesn't overpass the RTT, the two stations are coupled and assigned to the first scheduled operator. If not, the station with the current minimum station time is tried to be coupled with the station with the second maximum station time, and so on. Then, next two station are evaluated.

In the first approach, this rule is applied without increase the RTT, **config_6**. In the second approach, **config_7**, instead, to favorite the coupling opportunity, a recursive increase of the RTT is allowed till the 200% of the initial value, or if RUCs, considering all tasks for any station assigned them, result higher than a defined percentage of RTT. At the end of all step, any performance parameter is recorded and compared.

3.2. ANALYSIS AND COMPARISON OF EXPERIMENTAL RESULTS

Single Line

First, in next tables it is possible to observe the low presence, in the best configuration case, of stations under-utilized. Second, we can see that many of the stations, under charged at the first instance, have been 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 shows as the config_final results in an overall improvement in the efficiency and cost, without worsening, in fact, but at least, values remain the same. Infact, it means that the best configuration is the first, the one obtained after the first dynamic allocation. Any way, our new dynamic assignment results a large improvement for cost and efficiency values, if we should compare these to those corresponding to the first situation, or that provided by the company.

Observing data on tables 3, it's possible note many differences.

First, in the final case there are few undercharged stations that means a better balancing level, so that config_final is with a general lower number of resources.

Figure 6: snapshot of UCS in config_1 vs config_final configuration. In blue, station with maximum UC, in green, undercharged stations, in red, empties stations.

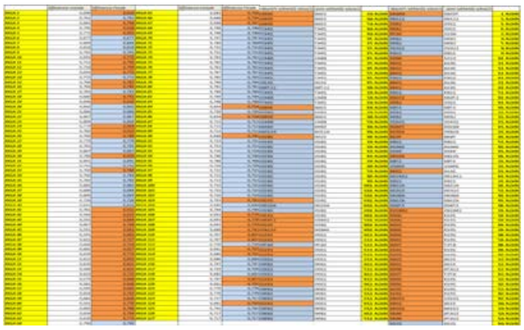

Figure 7: Efficiency and Direct Cost in the config_1 vs config_final. In orange, improved values, in blue, unchanged values, in red, worsened values.

Generally, stations that change from the config_1 to the final one, are those with one or more than undercharged stations in the initial situation.

In the table 4 UCs for the config_1 vs config_final show the gained improvement: at least, in the config_final, for a single OL, there is the same situation as in config_1, but never a worse one.

Table 1: number of winning configurations for all lines.

Config.1	Config.2	Config.3	Config.4	Config.5
69	0	58	0	0

From Table 5 it is evident that the winning configurations are config_1, corresponding to the initial dynamic distribution, and config_3, when RTT is amplified, before tasks of the under charged station are reassigned.

In the following figure 5 is showed the whole OLs Efficiency value for all considered configurations.

The whole mean efficiency value of the line after the first dynamic reassignment is of 72%.

The following phase, knew as Config_2, with the redistribution of the operators will cause a decrease, while to 52%, where, again, in the Config_3, the increase of the tack time makes possible the elimination of the undercharged stations which lead to a vertex efficiency of 78%, no more over passed also by Config_4, with redistribution of released workers, and config_5.

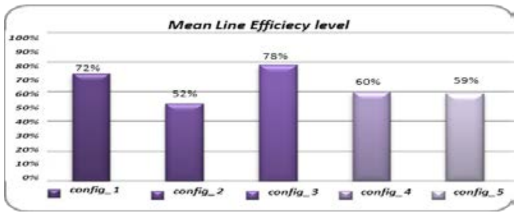

Fig. 8: Mean Line Efficiency level for all configuration.

The cost trend is still fairly close to that one of efficiency, this trend is shown in Figure 6. 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.

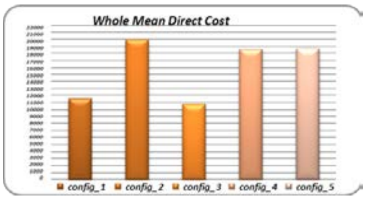

Fig. 9: Average Direct Assembly Cost for all single line configurations

Many further evidences, much more strong, arise from the observation of the UCs, when grouped just for OL with the same number of scheduled stations/resources. We will show just some situation.

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.

Now it is clearer the balancing advantages that we can achieve with our proposed heuristic.

Moreover, we propose just means values, just aggregated in some way for shortness needs, but if we observe the single OL better results can be noted.

In fact, in the above figures never is reached the 100% value, since they are averaged.

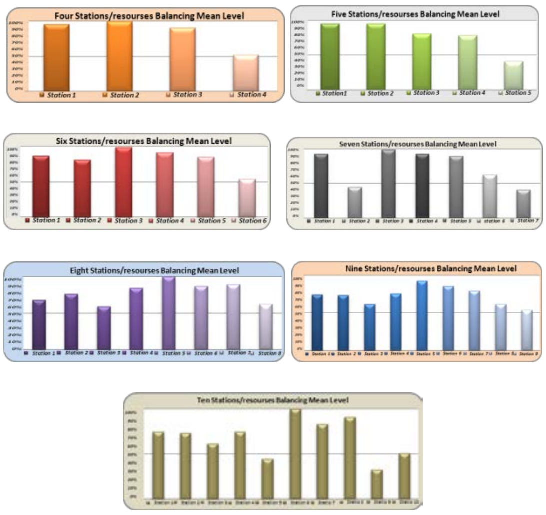

Fig. 10: Mean UCs just for OL with xx stations.

Balancing Strategy based on Resources

In the following figures, we outlines results for the resources balancing approaches.

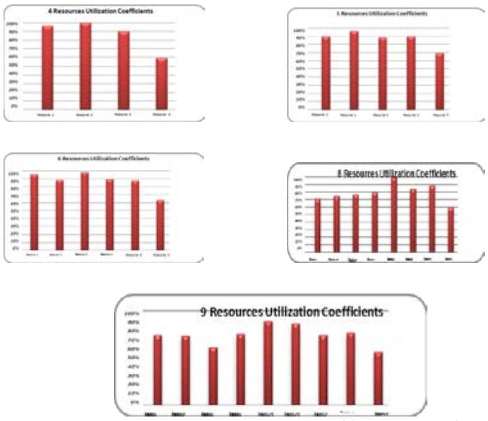

Fig. 11: Mean Utilization Coefficients for line configurations with xxx Operators and coupled stations in the first approach.

Observing UCs, a good improvement can be noted, both for the average efficiency, and both for the internal SUCs, that are much more homogeneous than before.

In the following lines we show a comparison of the average Line Efficiency among the pre coupling approach, in the config_final, in the first coupling approach, and, finally in the second coupling approach.

Fig. 12: Mean RUCs for line configurations.

Line efficiency presents a relevant increase of 7% since the first coupling approach, and one smoother passing to the second.

Similar consideration can raise observing the mean Direct Cost for the whole AP.

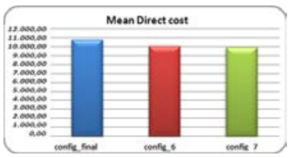

Fig. 13: Mean Direct Cost for line configurations.

In no one OL we can observe a worsening of the performances of efficiency and Direct Cost.

In fact, in config_6, without increase RTT, is just possible the RUCs improve, and in the second approach, config_7, just more opportunities to have a resource balance come, and the increase of RTT is compensated by the eventual resource reduction.

The resource balancing operates on 95 OL on 127.

4. CONCLUSIONS

In this paper a new step of thoroughly research was conducted regarding possible improvements of heuristics logics to be applied to the case of an manual assembly line.

The most critical issues were identified and then addressed the, through an multiphase algorithm definition and consequent simulation of the process.

We based this new step based on previous models and on related outputs. We oriented our attention to a more general solution, in terms of flexibility, variability of mix, number of resources and stations, number and position of some constrains.

Some deeper solutions have been evaluated to define time savings when more than one resource is assigned to the same station, and to decide the station where assign more than one operator to.

The ultimate strategy based on resource balancing seems be much more better to face a large typology of situations, with any values for tasks time and constrains positions.

The opportunity of improvement can be obtained with the layout shape and with a very low investment cost, and with a really general, versatile and flexible algorithm, under the dimension level, but also with easy configuration of data and parameters values.

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

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

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.

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.

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.

Bautista J., Pereira J., 2008. "*A Dynamic Programming Based Heuristic for the Assembly Line Balancing Problem*", International Journal of Production Economics.

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

Gallo, S. A., Davoli G., Govoni A., Melloni R., Pattarozzi G., Simulation models to support GALB heuristic algorithms and to evaluate multi objective performance index, The 24th European Modeling & Simulation Symposium, September, 19-21, 2012, Vienna, Austria.

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., 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.