A comparative analysis of sequencing heuristics for solving the Toyota Goal Chasing problem

Article *in* Robotics and Computer-Integrated Manufacturing · December 2004

DOI: 10.1016/j.rcim.2004.07.009

CITATIONS
10

READS
281

3 authors:

G. Celano
University of Catania
100 PUBLICATIONS **1,806** CITATIONS
SEE PROFILE

Antonio Costa
University of Catania
54 PUBLICATIONS **595** CITATIONS
SEE PROFILE

S. Fichera
University of Catania
76 PUBLICATIONS **1,051** CITATIONS
SEE PROFILE

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Computer-Integrated Manufacturing 20 (2004) 573–581
Robotics and Computer-Integrated Manufacturing
www.elsevier.com/locate/rcim

A comparative analysis of sequencing heuristics for solving the Toyota Goal Chasing problem

Giovanni Celano, Antonio Costa, Sergio Fichera*

Dipartimento di Ingegneria Industriale e Meccanica, Università degli Studi di Catania, Viale Andrea Doria 6, 95125 Catania, Italy

Accepted 24 May 2004

Abstract

In this paper, the sequencing of a mixed model paced assembly line is investigated assuming the component parts usage smoothing as the goal of the sequence selection. This sequencing problem, commonly known as Toyota Goal Chasing method, is studied here taking into account not only the traditional Goal Chasing approaches, which assume zero-length assembly lines, but also models which consider the effective length of the assembly line. This means that the number of workstations and their extensions become critical parameters in the selection of the optimal sequence of models to be assembled: in fact, the epochs corresponding to the requirement of different components vary in accordance to the values of the line parameters. The sequencing of the parts is carried out here through a set of heuristic procedures, the commonly adopted Goal Chasing algorithms and a simulated annealing, whose performances are compared with respect to different line scenarios. In particular, the numbers of workstations, parts to be worked and components to be assembled are varied to statistically test their influence on the efficiency of the optimizing procedures and on the differences between zero and finite length approaches.

Keywords: Assembly line; Sequencing; Heuristic algorithms

1. Introduction

Mixed model assembly lines MMAL play a key role in manufacturing industry, due to their flexibility, which allows massive production and several models to be contemporarily produced. Several company's goals have been investigated in literature to optimize the efficiency of MMAL. These goals can represent the objective of the sequencing activities of MMAL, i.e. the decision rules which allow a schedule of models to be selected. In literature the following objectives have primarily been considered:

- to maintain a constant usage rate of the components used in the assembly line in order to obtain a good "just-in-time" management policy: several versions of the Toyota Goal Chasing method were developed to solve this problem, see [1–5];
- to maintain a constant feeding rate of every model introduced into the line: an interesting survey about research dealing with this problem is proposed in [6];
- to level the load at each workstation in order to minimise the risk of stopping the conveyor [7,8];
- to minimize the total conveyor stoppage time, as required by the concept of "autonomation" developed in the Toyota production system: the impact of this objective on assembly line sequencing has been investigated in [9,10].

In this paper, the attention is focused on the smoothing of components parts usage. In particular, assemblies characterized by different components and different quantities of components common among models are considered. An example of such a production can be represented by automotive industry, where a car

*Corresponding author. Tel: +39-095-7382411; fax: +39-095-337994.
E-mail address: sfichera@diim.unict.it (S. Fichera).

0736-5845/$ © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2004.07.009

model can require a sliding roof while others may not, or where different numbers of airbags must be installed. Furthermore, to simplify the problem, a one-level bill of materials is considered for each model: this means that the single model is assembled considering only the subassemblies. As stated before, this problem is commonly known as Toyota Goal Chasing, because the researchers of Toyota Motor Company proposed the earlier solution procedures to this problem. A presentation of these procedures is given in [1]. In this paper one of them, denoted as Goal Chasing (GC), will be mentioned and discussed. Three heuristics were proposed in [2] for products characterized by a multi-level production system; however, they can be easily adapted to the single level configuration, as reported in [4]. Here, one of them, denoted as extended Goal Chasing EGC, is discussed and analyzed: its improvement with respect to the previous Toyota heuristics depends on the way adopted by this procedure to select a model to be sequenced.

A common hypothesis to all of these heuristics is that the epoch when a component is required coincides with the model entrance within the assembly line. In few words, a zero-length one-workstation assembly line is considered. This assumption is not always true in the real production environment: in fact, very often the time elapsed between the entrance of the model within the line and its exit can be very large, i.e. the model flow time can cover a time period of several hours. Considering a paced assembly line, whose workstations are connected through a conveyor, which moves with constant speed, the needed components to assemble the model are required by each workstation at different epochs, in accordance to the model bill of materials and to the task assignment to each workstation. In this case, the traditional Goal Chasing approaches could provide an inefficient sequencing, due to their formulation, which assumes that the length of the unique workstation is equal to zero. In a recent paper [5], the Toyota Goal Chasing problem was solved for a multiple workstation assembly line, where the effective line length is taken into account. This improved model is called modified Goal Chasing (MGC). The introduction of the effective station length requires a re-formulation of the function to be optimized to complete the sequencing of parts: the Euclidean distance between the ideal component usage and the effective use not only is evaluated with respect to each component but must be also extended to each workstation. The optimal sequencing is determined through a heuristic which works step-by-step similarly to the previous mentioned heuristics: the sequence is constructed selecting among the available models the one which minimizes the Euclidean distance with respect to each component and workstation.

In this paper, the Toyota Goal Chasing problem for both the approaches with zero-length and multiple workstations is solved through the GC procedures, which select step-by-step the parts to be sequenced by minimizing a function representing the Euclidean distance from an ideal components usage, and a simulated annealing, denoted as SA, which heuristically investigates the space of allowable solutions. A similar study was performed in [4], but no finite length configuration was considered and only step-by-step procedures were implemented. The SA works on feasible sequences of models to be assembled and assumes as objective function the expression corresponding to each one of the reported approaches. An extensive set of line scenarios has been organized to take into account different numbers of workstations, models and components. In this way, a statistical analysis was carried out to test the influence of workstations, parts and components on the efficiency of the selected optimizing procedures and on the assumption of line finite length. The rest of the paper is organized as follows: in the next paragraph the GC, EGC and MGC approaches are reported; then, the simulated annealing is presented; finally, the computational results and the statistical analysis are discussed.

2. The Toyota Goal Chasing heuristics

Three heuristics have been taken into account to solve the problem of component parts usage smoothing; the heuristic denoted as GC, proposed in [1], and the EGC [2], which represents an improvement of GC: these two heuristics assume that the line coincides with a unique workstation having length equal to zero, Fig. 1a; then, the MGC [5] is introduced, which considers the effective length of the assembly line, Fig. 1b.

In the rest of the paper, the following notation is adopted:

K: set of workstations within the line, $k = 1, \ldots, K$
L_k: Length of workstation k, $L_k \langle 0$
N: total number of parts to be assembled, (each part belongs to a specific model m), $n = 1, \ldots, N$
M: set of models, $m = 1, \ldots, M$

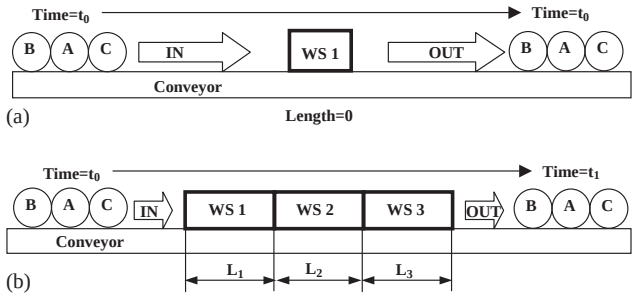

Fig. 1. The scheme of the mixed model assembly line.

d_m: demand for model m, $d_m \langle 0$

C: set of components (subassemblies) to be assembled

c_j: total number of components j needed to assemble the mix of N parts, $j = 1, \ldots, C$

c_{jm}: quantity of component j needed to assemble a unit of model m (valid for models GC and EGC, where $L_k = 0$)

$\boldsymbol{P}_m^k$: three-dimensional array whose generic element $(p^j m)^k$ represents the quantity of component j needed to assemble a unit of model m on workstation k (valid for model MGC, where $L_k > 0$)

The total number N of parts to be assembled can be evaluated as

$$N = \sum_{m=1}^{M} d_m. \tag{1}$$

2.1. The Goal Chasing

The GC proposed in [1] achieves the component parts usage smoothing through the step-by-step minimization of a function which represents the overall measure of deviation, extended to the whole set of C components, from an ideal consumption condition:

$$D_{m,n} = \sum_{j=1}^{C} \left[n\frac{c_j}{N} - \left(X_{j,n-1} + c_{jm}\right) \right]^2, \tag{2}$$

where the first term under summation is the expected (ideal) usage of component j needed to assemble the first n parts and $X_{j,n-1}$ represents the quantity of component j required to assemble the first n-1 parts. Therefore, the model m* which allows the minimization of the $D_{m,n}$ is selected step-by-step in the sequence. In this way, no overall optimization is guaranteed in the selection of the sequencing. In Fig. 2a a graphical representation of the difference between ideal and real usage of a generic component j is reported, ($c_j = 6$ units of component j are required to assemble $N = 10$ parts)

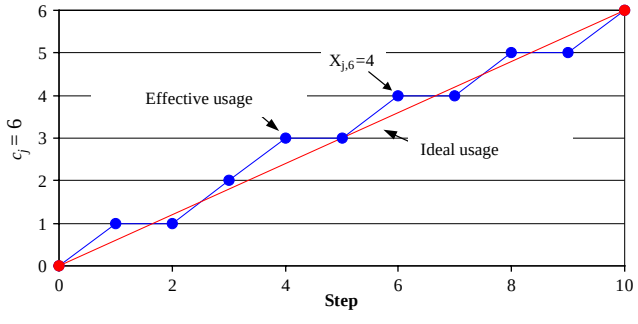

Fig. 2. The difference between real and ideal consumption of component j.

2.2. The extended Goal Chasing

The problem of a greedy optimization due to the step-by-step procedure was partially overcome in [2]. A one step procedure cannot guarantee an optimal solution because a decision about the selection of a model at position n does not consider what can happen in the future positions. Therefore, a possible solution to this problem can be represented by an approach, which allows some considerations about successive step to be taken into account. In [2], the selection of nth product within the sequence depends on the evaluation of the expression [2] for positions n and $n+1$. In few words, at step n a product m^* is selected, which corresponds to

$$D_{m^*,n} + \min_m \left\{ D_{m,n+1} \right\}. \tag{3}$$

2.3. The modified Goal Chasing

When the line length must be considered because of a high flow time of the parts within the manufacturing line, the GC and EGC procedures could provide misleading sequencing solutions. As stated before, these two procedures approximate the line length to zero and consider the component request as soon as each part enters within the line. This problem has been solved in [5], where a new GC model, denoted as MGC, is developed. The MGC takes into account the effective line length and evaluates the epochs when each component is effectively required to perform a task on a selected model. As a consequence, the implementation of this model requires the determination of the times when each part enters within the line workstations. As an example, in Fig. 3 the part flow and worker movement diagram of a paced mixed model assembly line is depicted: the light oblique lines represent the part flow, while the heavy and dotted lines with arrows correspond to the operations on the parts and worker movement respectively; v_c is the conveyor speed and t_c is the cycle time. This diagram allows an easy to be interpreted graphic representation of the line dynamics. With reference to Fig. 3, it is possible to introduce the terms that allow the MGC model to be formalized. The component quantities needed to assemble a part are considered through a three-dimensionsional array

$$\mathrm{P}_m^k = \left[p_m^i\right]^k, \tag{4}$$

where p_m^i represents the quantity of component i needed to assemble a unit of component m. The third dimension of the array is related to the number k denoting the position of the generic workstation within the line. The number of parts $n^k(t)$ entered within each generic workstation k at time t must be evaluated in order to determine the time epoch when a set of components

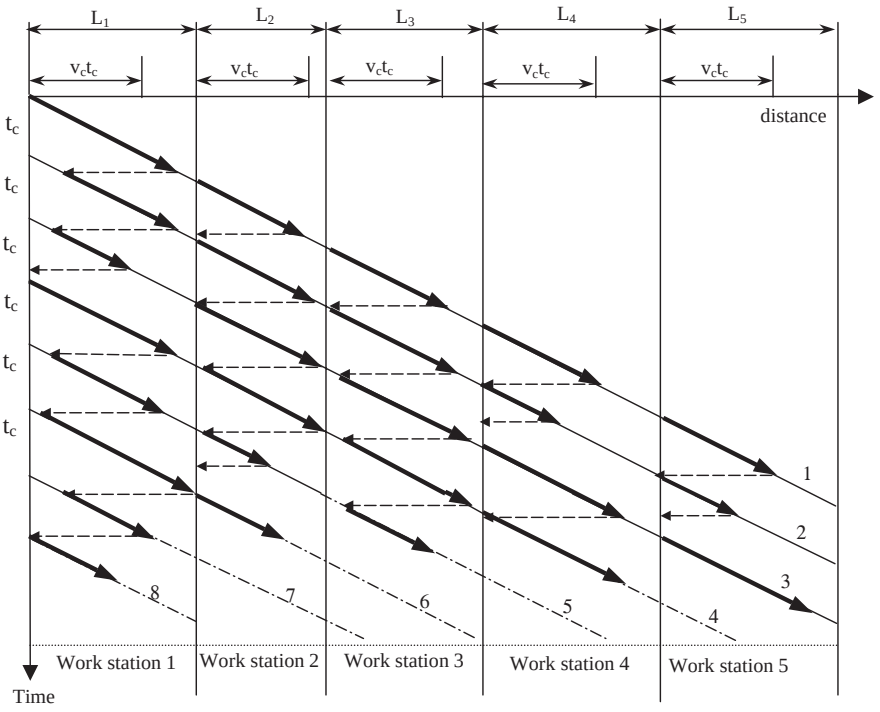

Fig. 3. Part flow and worker movement diagram.

should be available on the shop floor to assemble a part

$$n^k(t) = \left\lfloor \frac{t - \frac{1}{v_c}\sum_{j=1}^{k-1} L^j}{t_c} \right\rfloor + 1. \tag{5}$$

Therefore, the effective component usage $\mathbf{U}^k(t)$ at time t on each workstation k is equal to

$$\mathrm{U}^k(t) = \sum_{n=1}^{n^k(t)} \mathrm{P}_n^k. \tag{6}$$

The ideal component usage $\mathbf{r}^k$ within each workstation is equal to the total component usage corresponding to the mix of models to be assembled and divided for the total completion time of the mix T_N

$$T_N = (N-1)\, t_c + \frac{1}{v_c} \sum_{k \in K} L_k, \tag{7}$$

$$\mathbf{r}^k = \frac{1}{T_N} \cdot \mathrm{U}^k(T_N). \tag{8}$$

Finally, the objective function of the MGC model can be formalized

$$G(\pi) = \sum_{j=1}^{I_N} \sum_{k \in K} \left| \mathrm{U}^k((j-1)t_c) - j \cdot t_c \cdot \mathbf{r}^k \right|, \tag{9}$$

where $I_N = \lfloor T_N / t_c \rfloor$ allows the number of steps of the procedure to be determined. $\boldsymbol{\pi}$ corresponds to a feasible sequencing of the models to be assembled. Each part is sequenced by evaluating the function step-by-step

$$Z(n) = \sum_{j=1}^{I_n} \sum_{k \in K} \left| \mathrm{U}^k[(j-1)t_c] - j \cdot t_c \cdot \mathbf{r}^k \right| \tag{10}$$

which represents the $G(\pi)$ when a partial schedule of n elements is considered.

3. The sequencing algorithms

The GC, EGC and MGC procedures select the optimal sequence of models through a step-by-step procedure, which allow a model to be scheduled on the basis of the difference between the actual and the ideal accumulation of every component part. Furthermore, the EGC procedure considers two successive steps to minimize this difference. The solving procedure can be resumed in the following steps:

Step 1: $D \leftarrow \{d_1, \ldots, d_m\}$, $\pi = \{0\}$, $n \leftarrow 0$;
Step 2: $n \leftarrow n + 1$;
Step 3: Select a model m from $D, d_m > 0$, which minimizes the objective function OBJ: Eqs. (2), (3), (10);
Step 4: If $D \neq 0$, then recalculate **D** and $\pi = \pi + \{m\}$;
Step 5: $OBJ \leftarrow \min OBJ(n)$;
Step 6: Stop if $n = N$; otherwise, go to step 2.

An alternative to the reported procedure is the application of a simulated annealing algorithm (SA),

i.e., a heuristic algorithm which allows an entire sequence to be selected and evaluated. SA algorithm is an efficient tool for solving NP hard problems like the sequencing of a mixed model assembly line. These algorithms represent an enhanced version of the traditional techniques of local optimization or iterative improvement. In fact SA combines to the structure of the traditional steepest descent method the possibility to probabilistically accept "retrogressive" movements towards worst solutions. This feature allows new possible optimal solutions to be found out and reduces the probability of accepting poor local optimal solutions. SA operates through an analogy with statistical mechanics of condensed matter physics. Each stage of the algorithm represents a cooling "temperature" level, which is maintained till an "equilibrium condition" is reached; then the temperature is cooled and the local search goes on.

The first step is the formulation of the objective function, which is here represented by the expressions (2), (3) and (10), which consider zero-length or finite length assembly lines, respectively. The generic sequencing $\boldsymbol{\pi}$ represents the vector to be optimized by the simulated annealing. The algorithm starts at a specified initial temperature T_0 by randomly selecting an initial seed vector; then it evolves through a selected number of iterations, the "temperature levels", trying to improve the value of the selected objective function. The "temperature" is cooled in accordance with a selected cooling scheme. In this paper, a simple cooling schedule based on the variation of the objective function *OBJ* and the actual temperature level has been chosen. Once a new feasible vector $\boldsymbol{\pi}'$ has been determined as a perturbation of the original one, if $\boldsymbol{\pi}'$ permits an objective function value lower than the one corresponding to original vector $\boldsymbol{\pi}$, it is accepted and becomes the new reference sequence. On the other hand, if the vector π' has a worse value of *OBJ* it is accepted with a probability lower than 1. The algorithm is frozen when the temperature reaches its final value T_f.

During the algorithm evolution, the perturbed vectors π' are determined through two proper neighborhood search schemes having an equal probability to be selected: the adjacent exchange and the shift insertion. The adjacent exchange works by selecting two adjacent elements and mutually changing them within the sequence. The shift insertion selects two generic positions s and r ($s<r$), within the sequence; then it extracts the element in the sth position and moves the elements between $s+1$ and r positions on the left; finally, it positions the extracted element in position r. The following parameters must be settled before starting the algorithm:

T_0	initial temperature
T_f	final temperature
N_{max}	maximum number of iterations at each temperature
α	temperature reduction factor.

The procedure of the developed SA algorithm is presented below

Step 1: Generate the seed sequence $\boldsymbol{\pi}$;
Step 2: Parameter initialization, $T_i = T_0$, T_f, N_{max}, α;
Step 3: $N_{rip} = 0$;
Step 4: Generate a perturbed sequence π' through adjacent exchange or shift insertion neighborhood scheme;
$N_{rip} \leftarrow N_{rip} + 1$;
Step 5: Evaluation of $\Delta OBJ = OBJ(\pi') - OBJ(\pi)$;
if $\exp(-\Delta OBJ/T_i) > \xi$, where ξ is a rectangular random number within [0;1]
Then $\pi \leftarrow \pi'$;
else $N_{rip} = N_{max}$
Step 6: If $N_{rip} < N_{max}$ go to *Step* 4;
Step 7: $T_i \leftarrow \alpha T_i$
Step 8: If $T_i > T_f$
then go to *Step* 3
else STOP the program.

4. Computational results

The analysis of the heuristics was carried out by developing a set of problems which include several assembly line scenarios characterized by different numbers of workstations K, parts to be assembled N and components C needed to assemble the mix; 9 classes of problems were considered:

Class I:	$N = 30$,	$K = 6$,	$C = 10$;
Class II:	$N = 50$,	$K = 6$,	$C = 10$;
Class III:	$N = 100$,	$K = 6$,	$C = 10$;
Class IV:	$N = 30$,	$K = 10$,	$C = 15$;
Class V:	$N = 50$,	$K = 10$,	$C = 15$;
Class VI:	$N = 100$,	$K = 10$,	$C = 15$;
Class VII:	$N = 30$,	$K = 20$,	$C = 20$;
Class VIII:	$N = 50$,	$K = 20$,	$C = 20$;
Class IX:	$N = 100$,	$K = 20$,	$C = 20$.

Totally, 90 scenarios (10 for each class), were developed. The N parts to be sequenced belong to $M = 5$ models which require different combinations of the C component parts. Furthermore, a one-level bill of materials is assumed for each model and the line is assumed perfectly balanced: this results in the absence of uncompleted operations during the working activities and no conveyor stoppage occurs during the mix completion. The three-dimensional array $\mathbf{P}_m^k$ representing the component quantities needed to assemble a part in correspondence to each workstation has been

determined assuming a certain probability for component usage $\{p_m^i\}^k$, $(m = 1, \ldots, M;\ i = 1, \ldots C;\ k = 1, \ldots, K)$ (Table 1).

For example, a probability equal to 0.2 corresponds to the requirement of two units of component i to assemble the model m within workstation k. The approach is similar for the other quantities. The cycle time t_c was computed considering the following expression:

$$t_c = 1.025 \cdot \frac{\sum_{n=1}^{N}\sum_{k=1}^{K} t_n^k}{NK}, \tag{11}$$

where the generic working time t_n^k of nth scheduled part on workstation k is computed assigning to each component assembly a specific task time and considering the component demand at each workstation. The conveyor speed v_c was set equal to 1 speed unit and the length L_k of the workstations was evaluated as

$$[1.3 v_c t_c,\ 1.6 v_c t_c]. \tag{12}$$

In this way, all the input data are available to perform the analysis. A massive run for the SA was designed and performed, that is the number of investigated sequences was enlarged to test if the SA efficiency previously determined in [11] with a lower number of iterations has an acceptable level. A preliminary tuning of the SA parameters was conducted to determine their optimal configuration; in this way, the whole set of examples was sequenced by approximately fixing a maximum number of selected sequences respectively equal to:

- 1.0×10^5 investigated sequences when $N = 30$ (SA parameters: $T_0 = 350\,^{\circ}C$; $T_f = 0.1\,^{\circ}C$; $\alpha = 0.9$; $N_{max} = 2030$);
- 1.5×10^5 investigated sequences when $N = 50$ (SA parameters: $T_0 = 350\,^{\circ}C$; $T_f = 0.1\,^{\circ}C$; $\alpha = 0.9$; $N_{max} = 3430$);
- 2.5×10^5 investigated sequences when $N = 100$ (SA parameters: $T_0 = 350\,^{\circ}C$; $T_f = 0.1\,^{\circ}C$; $\alpha = 0.9$; $N_{max} = 6930$);

Table 2 shows the average values of the Euclidean distance (corresponding to expressions (2), (3) and (10), in accordance to the adopted approach), and the standard deviation for each class of 10 problems; furthermore, the corresponding average percentage improvements obtained through the SA algorithm versus the other heuristic procedures are reported.

The SA outperforms the GC heuristic in 82 examples out from 90 (the remaining have the same result); the EGC is outperformed in 74 examples out from 90 (12 scenarios are equally optimised); the MGC is outperformed for the entire subset of examples. If compared with the mean results presented in [11], Table 2 shows that the performances of the SA have a slight to moderate improvement: in particular, the differences are more evident for the comparison with the GC method than with the others, even if the number

Table 1
Component parts usage on each workstation

$\{p_m^i\}^k$	% Probability
0	20
1	50
2	20
3	10

Table 2
Heuristics comparison: average values of the Euclidean distance and corresponding percentage improvement

Class	*GC*	*SA*	Δ_{GC_SA} (%)	*EGC*	*SA*	Δ_{EGC_SA} (%)	*MGC*	*SA*	Δ_{MGC_SA} (%)
I	990.4 (96.3)*	813.2 (40.8)	17.90	1633.4 (105.1)	1532.8 (61.0)	6.16	5714.7 (294.5)	5456.6 (260.6)	4.52
II	1545.7 (162.4)	1306.7 (51.8)	15.47	2727.2 (252.7)	2488.1 (82.2)	8.77	9760.8 (491)	9281.6 (426)	4.91
III	2855.2 (303.5)	2532.5 (131.2)	11.30	5137.7 (557.9)	4833.0 (203.7)	5.93	20 099.6 (1424.8)	19 125.1 (1335.6)	4.85
IV	1883.9 (195.2)	1766.7 (84.9)	6.22	3540.7 (164.1)	3490.7 (137.9)	1.41	24 170.9 (389.0)	24 057.7 (394.0)	0.47
V	3009.1 (163.5)	2837.9 (100.7)	5.69	5656.3 (186.0)	5603.0 (148.7)	0.94	39881 (732.0)	39 654.7 (746.7)	0.57
VI	5978.6 (431.1)	5779.7 (276.8)	3.33	11 533.5 (707.0)	11 357.6 (419.2)	1.53	79 976.6 (2391.0)	79 441.7 (2303.6)	0.67
VII	4257.8 (433.6)	4041.8 (172.1)	5.07	8165.6 (415.4)	7942.9 (304.4)	2.73	13 7382.9 (1319.3)	13 6923.4 (1324.3)	0.33
VIII	6705.5 (506.4)	6394.8 (200.9)	4.63	12 760.1 (395.3)	12 717.4 (346.8)	0.33	229 852.3 (2415.6)	228 674.5 (2300.3)	0.51
IX	13501.6 (794.2)	12726.1 (402.1)	5.74	25 435.6 (1069.7)	25 113.3 (613.1)	1.27	45 7430.6 (2867.8)	45 4255.3 (2833.6)	0.69

*Standard deviation of the Euclidean distance for each class of 10 problems.

Table 3
Comparison between MGC and SA: two way ANOVA on the effects of number of workstations and parts

Factors	Df	Sum of *S*q	*Mean S*q	F_0	*P*-value
Workstations	2	39 434.18	19 717.09	941.56	0.000
Parts	2	159.92	79.96	3.81	0.026
Interaction	4	35.98	8.99	0.43	0.78
Residuals	81	1696.21	20.94		

Two way analysis of variance model with replications

of outperforming solutions is the same as found in the previous study (equal to 82); on the contrary, the efficiency variation with respect to both EGC and MGC heuristics remains substantially unchanged. Different interpretations can be attributed to this result: in fact, the EGC is more performing than the GC and therefore an extended search space for the SA seems not to influence the average percentage variation at all. The anomalous result for class VIII in the comparison between EGC and SA is due to the presence of seven out of 10 equally optimized scenarios: this can depend on random values selected to organize the scenarios; however, the SA outperforms the EGC in the remaining three classes of examples. On the other hand, the great complexity of the $G(\pi)$ adopted by MGC and SA to cope with the effective line length, tends to level the differences between alternative sequences of parts, in particular when the number of machines increases. As a consequence, the SA always outperforms the MGC heuristic, but with slight average improvements. Evaluating the possible influence of the assembly line factors, which characterize the different classes of scenarios, continued the study. A general factorial design was performed, assuming as influencing effects the number of workstations K, (and consequently the number C of components), and the number of parts N, both characterized by three levels; the response variable was coded as the ratio R_{XXX} between the values of the optimal distance functions corresponding to one of the literature heuristics and the SA: $R_{GC} = 10^3 * D_{m,n}^{(GC)}/D_{m,n}^{(SA)}$, $R_{EGC} = 10^3 * D_{m,n}^{(EGC)}/D_{m,n}^{(SA)}$, $R_{MGC} = 10^3 * G(\pi)^{(MGC)}/G(\pi)^{(SA)}$. In Table 3 the ANOVA table concerning the R_{MGC} is presented.

Table 3 shows how the numbers of workstations (components) and parts are influencing factors on the performances of the SA with respect to MGC with a level of significance $\alpha < 0.03$; on the other hand, no interaction is present. An analysis of the residuals show an improvement of their dispersion for classes I–III, characterized by a lower number of workstations: this means that the lower size problems are probably affected by the variability due to the randomization in the formulation of the scenarios. However, more stable results are obtained for the larger problems. Fig. 4 gives

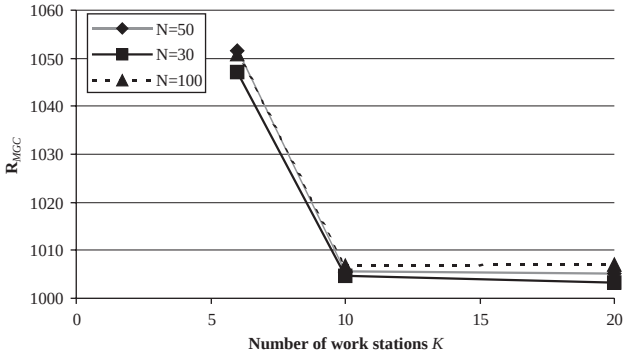

Fig. 4. Comparison between EGC and SA heuristics: influence of workstations (components) and parts on R_{MGC}.

a graphical representation of the R_{MGC} dependence versus the number K of workstations.

The factorial experiments conducted on the other ratios R_{GC} and R_{EGC} confirm only the influence of workstations. Considering the mean differences in performances reported in Table 2, small-sized problems show large variations; on the other hand, in the more complex scenarios and with the $G(\pi)$ function SA outperforms the other heuristics, but with smaller improvements. This means that the functions to be optimized are particularly flat, i.e., it is very hard to find a sequence that satisfies the goal of components usage smoothing. These conclusions suggested to further investigate the differences existing between a SA sequencing obtained assuming $D_{m,n}$ as objective function and another achieved considering $G(\pi)$: in this way the robustness of $D_{m,n}$ can be studied with respect to line length. Therefore, a comparison between the optimal $G(\pi)^{GC}$ corresponding to the sequencing obtained by assuming $D_{m,n}$ as objective function, i.e. neglecting the line extension, and the $G(\pi)^{MGC}$ corresponding to the sequencing obtained considering the effective line length was performed on all of the scenarios. A similar investigation was performed in [5], but on a very small example with $K = 5$ workstations, $N = 5$ parts and $C = 3$ models. Here, a more structured study was conducted on the 9 classes. In particular a factorial design was organized to understand if workstations (and

Table 4
Comparison between zero and finite line length approaches: two way ANOVA on the effects of number of workstations and parts

Factors	*Df*	Sum of *S*q	*Mean S*q	F_0	*P*-value
Workstations	2	49 718.03	24 859.02	2388.798	0.000
Parts	2	86.91	43.46	4.176	0.019
Interaction	4	118.80	29.70	2.854	0.029
Residuals	81	1696.21	20.94		
Mean values					
	Part 1	Part 2	Part 3		
$K = 6$	1055.3	1060.4	1055.2		
$K = 10$	1011.8	1013.4	1014.1		
$K = 20$	1002.5	1002.9	1002.9		

Two way Analysis of Variance Model with replications

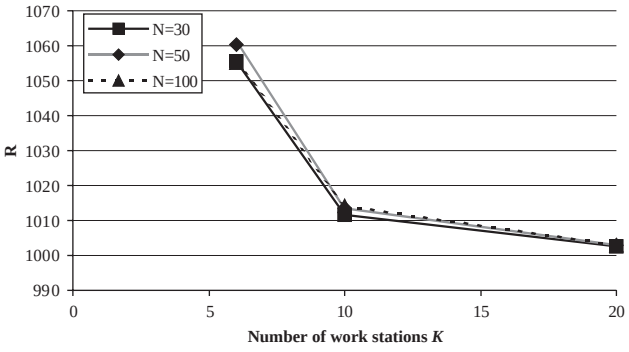

Fig. 5. Comparison between zero and finite line length approaches: influence of workstations (components) and parts.

consequently components) and parts influence the problem. The investigated response variable was the ratio $R = 10^3 * G(\pi)^{GC}/G(\pi)^{MGC}$. Table 4 and Fig. 5 show the results obtained from the statistical experiment:

As expected, the assumption of considering the effective line length is particularly effective when small-sized line scenarios are considered (Classes *I–III*, $K = 6$); the MGC efficiency progressively deteriorates as the number of workstations and parts increases: this result is due to the high complexity of process dynamics, characterized by high number of components to be assembled, workstations to be passed and parts to be completed; as a consequence the optimal distance from an ideal component consumption becomes more and more larger and difficult to be reduced, i.e. the values of $G(\pi)$ flatten and the sequencing corresponding to GC model becomes robust with respect to the effective line length. Therefore, it can be concluded that the MGC model is useful when the sequencing activity must be performed on small sized assembly lines, ($K_{\max} = 10$); in the other cases the traditional GC model can be adopted and the line extension could be neglected.

5. Conclusions and future research

In this paper, an analysis of heuristics for the solution of the sequencing problem of a mixed model assembly line is presented. The selected company's goal is the smoothing of component parts usage, known as Toyota Goal Chasing problem. Three different heuristics proposed in literature for the solution of this problem are adopted and compared with a simulated annealing algorithm. The three heuristics perform step-by-step procedures and take into account both lines with zero and finite length. The SA works on vectors representing a possible sequencing of the parts to be assembled and assumes as objective functions the same expressions as those used in the other heuristics. A statistical analysis has been performed on a set of 90 problems divided in 9 different classes to evaluate the effects of the line design and workload on algorithms and models performances The obtained results show that in the most cases the SA outperforms the other heuristics; in particular, the differences in the algorithm performances are influenced by workstations and parts number: in fact, the growth of line length and mix to be assembled causes a flattening of the sequencing solutions, being very hard to satisfy the component usage smoothing constraint. This aspect of the problem was confirmed by a further comparison carried out on the performance differences between models considering zero and finite line length, respectively: the effects of considering the effective line length reduces as the numbers of workstations and parts to be worked increase.

All of this research was conducted assuming that the mixed model lines are perfectly balanced, that is assuming no stopping conditions to occur. For paced mixed model assembly lines the perfect balancing condition is a very hard task to be achieved, therefore a more likewise approach requires that the effect of conveyor stoppages must be taken into account. In this sense, future research will be focused into this direction, developing sequencing models which take into account not only the timing of

parts at the entrance of the workstations but also their effective working times and eventual line stoppages. In this way, a multi-objective sequencing considering both components usage smoothing and minimization of mix completion time will be achieved.

References

[1] Monden Y. Toyota Production System: an integrated approach to Just in Time, 3rd ed. Norcross: Engineering and Management Press; 1998.
[2] Miltenburg J, Sinnamon G. Scheduling mixed-model multi-level just-in-time production systems. Int J. Prod Res 1989;27: 1487–509.
[3] Bautista J, Companys R, Corominas A. Heuristics and exact algorithms for solving the Monden problem. Eur J. Oper Res 1996;88:101–13.
[4] Duplaga EA, Bragg DJ. Mixed-model assembly line sequencing heuristics for smoothing component parts usage: a comparative analysis. Int J. Prod Res 1998;36(8):2209–24.
[5] Xiaobo Z, Zhou Z, Asres A. A note on Toyota's goal of sequencing mixed models on an assembly line. Comp Ind Eng 1999;36:57–65.
[6] Kubiak W. Minimizing variation of production rates in just-in-time systems: a survey. Eur J. Oper Res 1993;66:259–71.
[7] Yano CA, Rachamadugu R. Sequencing to minimize work overloading assembly lines with product options. Man Sci 1991;37:572–86.
[8] Sumichrast RT, Russel RS, Taylor B. A comparative analysis of sequencing procedures for mixed-model assembly lines in a just in time production system. Int J. Prod Res 1992; 30:199–214.
[9] Xiaobo Z, Ohno K. Algorithms for sequencing mixed models on an assembly line in a JIT production system. Comp Ind Eng 1997;32(1):47–56.
[10] Xiaobo Z, Ohno K. Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. Eur J. Oper Res 2000;124(3):560–70.
[11] Celano G, Costa A, Fichera S. A comparative analysis of sequencing heuristics for solving the Toyota Goal Chasing problem. Proceedings of the 13th Conference on Flex Aut and Intelligence Manufacture, 2003. p. 307.