Simultaneously Solving Mixed Model Assembly Line Balancing and Sequencing problems with FSS Algorithm

J.B. Monteiro-Filho*, I.M.C. Albuquerque¹, and F.B. Lima Neto¹

¹Computational intelligence research group - Polytechnical School of Pernambuco
Benfica 455, Recife-PE, Brazil
*Corresponding author: jbmf@ecomp.poli.br

July 20, 2017

Abstract

Many assembly lines related optimization problems have been tackled by researchers in the last decades due to its relevance for the decision makers within manufacturing industry. Many of these problems, more specifically Assembly Lines Balancing and Sequencing problems, are known to be NP-Hard. Therefore, Computational Intelligence solution approaches have been conceived in order to provide practical use decision making tools. In this work, we proposed a simultaneous solution approach in order to tackle both Balancing and Sequencing problems utilizing an effective meta-heuristic algorithm referred as Fish School Search. Three different test instances were solved with the original and two modified versions of this algorithm and the results were compared with Particle Swarm Optimization Algorithm.

1 Introduction

Products requirements and consequently the requirements of production systems have changed dramatically over the years. Back in 1913, when Henry Ford conceived the Assembly Lines (AL), this manufacturing paradigm was developed for the mass production of standardized products. However, nowadays increasing variety of models and options to be selected by customers have changed the market scenario and manufacturers have to handle a variety which frequently exceeds several billions of models [13]. As an example, the German car manufacturer BMW produces $10^{32}$ different car models [12].

The high variety issue typically requires implementation of cost efficient flexible production systems. This is normally addressed by the use of the so called mixed-model assembly, where setup operations are reduced to such an extent that various models of a common base product can be manufactured in intermixed sequences [25].

Mixed-model assembly lines (MMAL) facilitate product variations and diversities on the same line in an intermixed scenario. Hence, optimal AL design, balancing and product sequencing of mixed-models assembly are the major challenges for manufacturers for creating high-variety and low-volume production within the process [35].

A novel Assembly Line Balancing Problem (ALBP) referred as Mixed Model Workplace Time Dependent Assembly Line Balancing Problem (MMWALBP) was recently proposed. Due to its mixed model feature, a Mixed Model Sequencing Problem arises (MMSP), as this is the case for any mixed model ALBP.

Mixed Model Sequencing and Assembly lines Balancing are strongly related problems. The sequencing problem depends on a given balancing solution in order to be solved and the quality of the sequencing depends on the quality of the balancing solution provided [29]. Both problems are known to be NP-Hard [9]. Therefore, many different procedures have been developed in order to tackle the assembly line related problems [30]. Moreover, metaheuristic approaches have been playing an important role in their solutions as well as in many other classes of combinatorial optimization problems [11].

The Fish School Search algorithm (FSS), presented originally in 2008 by Bastos-Filho and Lima Neto [5], is a population based continuous optimization technique inspired in the behavior
of fish schools while looking for food. Each fish in the school represents a solution for a given optimization problem and the algorithm utilizes the information of every fish to regulate between exploration/exploitation behaviors and guide the school to promising regions in the search space as well as avoiding early convergence in local optima.

Variations of FSS were proposed and applied in the solution of ALBP [2] [1]. In the present work, an integrated approach was employed to simultaneously solve MMWALBP and MMSP with the original version of FSS, referred as FSS vanilla (FSS-V), as well as with two variations of the algorithm called FSS Stagnation Avoidance Routine (FSS-SAR) [24] and FSS Not Penalizing Static Success (FSS-NPSS).

The remainder of this article is organized as follows: Section 2 defines FSS-V, FSS-SAR and FSS-NPSS. Section 3 introduces the two problems tackled within this research effort. Section 4 presents the simultaneous solution procedure applied to solve both problems and also the specific approaches utilized to tackle each problem individually. Sections 5 and 6 presents the tests performed and the results found.

2 Fish School Search Algorithm

2.1 FSS - Vanilla

FSS is a population based search algorithm inspired in the behavior of a school of swimming fishes that expands and contracts while looking for food. Each fish n-dimensional location represents a possible solution for the optimization problem. The algorithm makes use of weights for all fishes which represents cumulative account on how successful has been the search for each fish in the school [5].

FSS is composed of four operators which run sequentially within each iteration. They are: Individual movement, feeding, Collective-instinctive movement and Collective-volitive movement.

The general pseudo-code for FSS is:

1: Initialize user parameters
2: Initialize fishes positions randomly
3: while Stopping condition is not met do
4: Calculate fitness for each fish
5: Run Individual operator movement
6: Run Feed operator
7: Run Collective-instinctive movement operator
8: Run Collective-volitive movement operator
9: end while

A description of each operator in further details is provided next:

1. **Individual movement operator:** every fish in the school performs a local search looking for promising regions. The movement only happens if the new position increases the fish fitness. The position update happen according to Eq. 1.

\[
x_i(t + 1) = x_i(t) + \text{rand}(-1, 1) \cdot \text{step}_{ind},
\]

where \(x_i(t)\) and \(x_i(t + 1)\) represent the position of the fish \(i\) before and after the individual moment operator, respectively. \(\text{rand}(-1, 1)\) is a uniformly distributed random number varying from -1 up to 1 and \(\text{step}_{ind}\) is a parameter that defines the maximum displacement for this movement.

2. **Feeding operator:** feeding here means the update of the weight parameter \(W_i\) update for fish \(i\). This is performed based on the improvement experienced by each fish, which is shown in Eq. 2.

\[
W_i(t + 1) = W_i(t) + \frac{\Delta f_i}{\max(|\Delta f_i|)},
\]

where \(\Delta f_i\) is the fitness variation between the last and the new position and \(\max(|\Delta f_i|)\) represents the maximum absolute value of the fitness variation among all the fishes in the school. \(W\) is only allowed to vary from 1 up to \(W_{scale}\), which is a user defined attribute. The weights of all fishes are initialized with the value \(W_{scale}/2\).
3. **Collective-instinctive movement operator:** in this operator, fishes move following the most increasing fitness movements that occurred in the last individual movement run. An average of the individual movements is calculated based on eq. 3.

\[
I = \frac{\sum_{i=1}^{N} \Delta x_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}.
\]

(3)

and then every fish will be moved according to:

\[
x_i(t+1) = x_i(t) + I.
\]

(4)

4. **Collective-volitive movement operator:** this operator is employed in order to automatically regulate the exploration/exploitation ability of the school during the search process. The school’s barycenter \(B\) must be first calculated, which is performed according to Eq. 5, taking into account the position \(x_i\) and the weight \(W_i\) of each fish:

\[
B(t) = \frac{\sum_{i=1}^{N} x_i(t) W_i(t)}{\sum_{i=1}^{N} W_i(t)}.
\]

(5)

And finally, if the total school weight \(\sum_{i=1}^{N} W_i\) has increased from the last to the current iteration, the fishes are attracted to the barycenter according to Eq. 6. If the total school weight has not improved, fishes are spread away from the barycenter according to Eq. 7.

\[
x_i(t+1) = x_i(t) - \text{step}_{\text{ind}} \text{rand}(0,1) \frac{x_i(t) - B(t)}{\text{distance}(x_i(t), B(t))},
\]

(6)

\[
x_i(t+1) = x_i(t) + \text{step}_{\text{vol}} \text{rand}(0,1) \frac{x_i(t) - B(t)}{\text{distance}(x_i(t), B(t))},
\]

(7)

where \(\text{step}_{\text{ind}}\) defines the size of the maximum displacement performed with the use of this operator. \(\text{distance}(x_i(t), B(t))\) is the euclidean distance between the fish \(i\) position and the school barycenter. \(\text{rand}(0,1)\) is a uniformly distributed random number varying from 0 up to 1.

The parameters \(\text{step}_{\text{ind}}\) and \(\text{step}_{\text{vol}}\) decay linearly according to:

\[
\text{step}_{\text{ind}}(t+1) = \text{step}_{\text{ind}}(t) - \frac{\text{step}_{\text{ind}}(\text{initial})}{I_{\text{max}}},
\]

(8)

and similarly:

\[
\text{step}_{\text{vol}}(t+1) = \text{step}_{\text{vol}}(t) - \frac{\text{step}_{\text{vol}}(\text{initial})}{I_{\text{max}}},
\]

(9)

where \(\text{step}_{\text{ind}}(\text{initial})\) and \(\text{step}_{\text{vol}}(\text{initial})\) are user defined initial values for \(\text{step}_{\text{ind}}\) and \(\text{step}_{\text{vol}}\), respectively. \(I_{\text{max}}\) is the maximum number of iterations allowed in the search process.

2.2 **FSS - Stagnation Avoidance Routine**

As mentioned before, a modification was proposed in the original FSS in order to make it improve its exploration ability [24]. In the original version of the algorithm, the Individual movement component is only allowed to move a fish if it improves the fitness. However, in a very smooth search space, there would be many moving trials with no success and the algorithm could fail to converge.

Further, the Collective-volitive movement operator was designed to automatically regulate the exploration/exploitation ability of the algorithm along the search process. However, in order to do so, this behavior depends on the possibility of the total weight of the school to reduce. If it does not happen, only Eq. 6 will be utilized in this operator. This means that the ability of attracting the fishes to the school barycenter in order to exploit the search space will always predominate with relation to the ability of spreading the school away from the barycenter in order to allow exploration.
To solve these issues, we introduced a parameter $\alpha$ for which $0 \leq \alpha \leq 1$ in the individual movement operator. $\alpha$ decays exponentially along with the iterations and measures a probability of a worsening allowance for each fish. This means that, every time a fish tries to move to a position that does not improve its fitness, a random number is chosen and if it is smaller than $\alpha$ the movement is allowed. Therefore, only the fishes which presented improvement in their fitnesses within the individual component of the movement can contribute to the $I$ vector calculation used in the collective instinctive movement. In this case, $I$ will be calculated according to:

$$I = \frac{\sum_{i \in N} \Delta x_i \Delta f_i}{\sum_{i \in N} \Delta f_i},$$

(10)

where $N$ is the set of all the fishes which improved their fitness in the last Individual movement performed.

This modification is intended to improve the algorithm exploration ability by allowing stochastic worsening movements. However, as the parameter $\alpha$ decays exponentially along the iterations, this effect is intense only in the beginning of the search process and gradually becomes irrelevant in the final of the search.

2.3 FSS - Not Penalizing Static Success

FSS-Not Penalizing Static Success (FSS-NPSS) was proposed in the work of Monteiro et al. [23]. The main modifications are related to the Feeding operator an with the inclusion of elitism within the Collective-instinctive movement. The new feeding operator is now defined according to Eq. 11.

$$W_i(t + 1) = 1 + (W_{scale} - 1) \times \frac{(F_i - F_{min})}{(F_{max} - F_{min})},$$

(11)

where $W_i$ and $F_i$ represent fish $i$ weight and fitness, respectively. $F_{min}$ and $F_{max}$ are the maximum and minimum fitness values found during all the search process.

For the elitism inclusion, fishes which were not able to move within the Individual movement due to the fact that they have already reached a good fitness region can still be able to contribute in the Collective-instinctive movement operator.

In FSS-NPSS, fishes that have not improved will have their displacement $\Delta x_i$ and weight variation $\Delta W_i$ replaced by:

$$\Delta x_i^* = step_{vol} \frac{\Delta x_i^t}{step_{ind}},$$

(12)

and:

$$\Delta W_i^* = max(\Delta W_i) \times \frac{W_i - 1}{W_{scale} - 1},$$

(13)

where $\Delta x_i^*$ and $\Delta W_i^*$ are fish $i$ “fake” displacement and weight increase to be used within the Collective-instinctive movement. $\Delta x_i^t$ is the last displacement of fish $i$ which occurred in iteration $t$ and $step_{ind}$ is the step of the Individual movement operator in iteration $t$.

Considering that after the Feeding operator the weight of each fish is a reliable representation of how good is the performance of it during the search, we also propose a modification in the Collective-instinctive movement operator. Instead of using the fitness variation to weight the displacements, this new operator considers weights variation values. Thus, for a school with $N$ fishes, vector $I$ is computed as follows:

$$I = \frac{\sum_{i=1}^{N} \Delta x_i \Delta W_i}{\sum_{i=1}^{N} \Delta W_i}.$$  

(14)

FSS-NPSS can be combined with FSS-SAR raising other variation known as FSS-NPSS-SAR. This version was chosen to be used in this article in order to provide a comparison on the solution quality provided by the different solution approaches.
3 Problem Definition

3.1 Mixed Model Workplace Time Dependent Assembly Line Balancing Problem

The Mixed Model Workplace Time Dependent Assembly Line Balancing Problem (MMWALBP) was proposed taking simultaneously into account the features of Mixed Model Two Sided and Multi Manned Assembly Line Balancing Problem [32] [16]. Those features are: Mixed models and many workplaces per workstations.

The authors applied the mean model approach, first introduced in the work of [34], in order to tackle the mixed model issue. This means that tasks times are adjusted based on the production level of each model and a joint precedence graph is created. The mean model is utilized to convert a mixed model problem in a single model one.

The main goal of using multi-manned workstations, i.e. workstations containing more than one active workplace, is to minimize the number of workstations of the line while its total effectiveness (in terms of number of workers) remains optimal [28]. Multi-manned assembly lines have substantial advantage over a simple assembly line such as reducing: the length of the assembly line and consequently the Work in Progress number of products, the total throughput time, the cost of tools and fixtures, the material handling, workers movement and setup times [17].

MMWALBP has a similar approach when compared to Variable Workplace Assembly Line Balancing Problem (VWALBP) [10]. The workstation was divided in different zones as it is shown in Figure 1. The arrow indicates the flow sense of the line. It can be seen that each number identify a work zone within the workstation that will be used to define workplaces. Number 4 represents the zone in the interior of the product to be assembled. Each task contains an attribute telling the decision maker in which zone it should be performed. A maximum number of opened workplaces (operators) per workstation is defined. However, in VWALBP employs a list of prohibited workplaces combinations containing the pairs of workplaces for which a displacement from one to the other would take a relevant amount of time. This was changed within MMWALBP tasks assignment. A time correction procedure was utilized in a way that every displacement between different zones within the workstation generates an additional time in the task that required the displacement.

In order to address the times correction issue, a times matrix must be defined containing the times necessary to move from one workplace to another. The displacement times matrix is specific for each assembly line, hence it should be specifically defined for a real line assembly balancing problem. Many different approaches exist in order to define human operations standard times. One of most common and used approaches is the Method-Times Measurement (MTM) [19][21].

3.2 Mixed Model Assembly Line Sequencing Problem

In Mixed Model Sequencing Problem (MMSP), attention is focused on the shop floor where a decision must be made as to the specific order in which the different models should be launched onto the line. The demand rate of each model may vary. Hence, this short term decision problem must be solved periodically for the appropriate demand rates and the predetermined line balance [4].

According to the literature review performed within the work of Boysen et al. [13], MMSP can be grouped according to its optimization target:
• **Work Overload**: If several models follow each other at the same station, work overloads might occur, which need to be compensated, e.g. by additional utility workers. Work overloads can be avoided if a sequence of models is found, where those models which cause high station times alternate with less work-intensive ones.

• **Just-in-time-objectives**: An important prerequisite for JIT-supply is a steady demand rate of material over time, as otherwise the advantages of JIT are sapped by enlarged safety stocks that become necessary to avoid stock-outs during demand peaks. Accordingly, JIT-centric sequencing approaches aim at distributing the material requirements evenly over the planning horizon.

Furthermore, three general sequencing approaches were proposed in literature:

• Mixed-model sequencing [37];
• Car sequencing [33];
• Level scheduling [20];

In this work, we proposed a FSS based simultaneous solution approach. The balancing version is the aforementioned MMWALBP and the sequencing problem is the Mixed-model sequencing for work overload minimization (MMSP-W). This approach aims at avoiding/minimizing sequence-dependent work overload based on a detailed scheduling which explicitly takes operation times, worker movements, station borders and other operational characteristics of the line into account [13].

In this work, paced straight assembly lines with no parallel workstations and no setup times are taken into account. The workstations are closed. The uncompleted work is performed by utility workers with no concurrency considered.

4 FSS Solution Procedure

4.1 Balancing and Sequencing Simultaneous Solution Approach

As mentioned before, a simultaneous approach was proposed in order to simultaneously solve both the balancing and sequencing problems of MMALs.

The procedure, detailed in the flow shown in Figure 2, is performed according to the following steps:

1. To input balancing data;
2. To solve MMWALBP;
3. Balancing procedures outputs the \( n \) (user defined) best solution found;
4. Input the sequencing data;
5. To solve the sequencing problem individually for each of the \( n \) balancing solutions;
6. To return the best balancing/sequencing combination found;

The solutions representation as well as the objective functions utilized will be described in the following sections.

4.2 Solution representation

Simple procedures were applied in order to map continuous solutions candidates (fishes positions) in discrete valued arrays representing solutions in the problem domain. FSS algorithm was then applied in the continuous space. Once a fitness calculation is required, the fish position is converted into discrete sequences.

Specifically in the case of MMWALBP, the task-oriented approach was applied [29]. This means that a balancing solution is an permutation of the tasks indexes which is utilized within the assignment procedure for providing fitness calculation. The mapping was performed with a
variation of the random-keys [18] procedure which converts the smallest value in the fish position array into number 1, the second smallest value into number 2, and this is repeated up to when all the array values are mapped into a task index. Figure 3 shows an example of an application of the random-keys procedure on a real valued array with five tasks.

Figure 2: Simultaneous solution procedure

Figure 3: Random-keys variation mapping procedure

For the sequencing portion of the solution, the mapping was proposed through another variation of the aforementioned procedure and it is referred as multiple random-keys. For a given set \( i = 1, 2, 3...I \) of \( I \) different models with production levels given by \( P_i \), the total production \( P_m \) will be given by \( \sum_{i=1}^{I} P_i \). The mapping is performed by assigning \( i \) to the \( P_i \) smallest values.

The proposed mapping procedure for a given position vector \( F \) with dimensions \((P_k, 1)\) is:

1: \( A = F \) {Copy \( F \)}
2: \textbf{for} \( i = 1 \) to \( I \) \textbf{do}
3: \hspace{1em} \textbf{for} \( j = 1 \) to \( P_i \) \textbf{do}
4: \hspace{2em} \( t = \text{Index}(\text{Min}(A)) \) \{t is the index of the minimum value of \( A \)\}
5: \hspace{2em} \( C(t, 1) = i \)
6: \hspace{2em} \( A(t, 1) = \beta \) \{\( \beta \) is a very large number\}
7: \hspace{1em} \textbf{end for}
8: \textbf{end for}
9: \textbf{return} \( C \)

4.3 Objective function

In the balancing portion of the problem solution, it was applied the objective function shown in Eq. 15.

\[
\text{minimize} \left( K \times \sqrt{\frac{1}{K} \sum_{k=1}^{K} (C - t_k)^2} \right), \quad (15)
\]
where $K$ is the number of workplaces (number of operators), $C$ is the cycle time and $t_k$ is the workload at workplace $k$.

This modified objective function simultaneously improves the number of workstations and smoothness of the line balancing, furthermore it changes the search space allowing more variation in the fitness when fishes move.

For sequencing problems, many different objective functions appear in literature. Boysen et al. [13] cites: minimize work overload [31], minimize line length [4], minimize throughput time [3], minimize maximum displacement of workers from their respective reference point [26] among others. Further, in the works of Bautista et al. [6] [7] the case of workload minimization was considered with the introduction of interruption rules and regularity constraints, respectively.

In this article, we considered the same problem as in the work of Yano and Rachamadugu [38] for minimization of the total unfinished work or maximization of the total completed work which is the same as minimize work overload [8].

In order to compute the objective function, consider:

$L$: workstation length (in cycle time units);
$p_{ik}$: process time of job $i$ at workplace $k$;
$s_{ik}$: start time of job $i$ at workplace $k$;
$f_{ik}$: finish time of job $i$ at workplace $k$;
$v_{ik}$: completed work in job $i$ at workplace $k$.

Assuming $S_{1k} = 0$ for all $k$ workplaces, we have:

$$s_{ik} = \max(i - 1, f_{i-1});$$

(16)

$$f_{ik} = \min(s_{ik} + p_{ik}, i - 1 + L);$$

(17)

$$v_{ik} = \min(p_{ik}, L + i - 1 - s_{ik});$$

(18)

The objective function chosen to be maximized, the total completed work, will be given by Eq. 19.

$$\sum_{i=1}^{P_l} \sum_{k=1}^{K} v_{ik}.$$  (19)

5 Experiments and Tests

Tests were performed in six problems derived from SALBP-1 datasets from www.assembly-line-balancing.de. We have used the sets containing 50 different models with the number of tasks 20, 50 and 100. These problems are referred as $n = 20 - 50$ (small), $n = 50 - 50$ (medium) and $n = 100 - 50$ (large). The first number indicates the quantity of tasks to be allocated and the second stands for the number of different models to be sequenced. In all of these problems, a production plan of 998 models is provided as well as the production level $P_i$ for each model $i$. Cycle time is 1000 in all test cases.

In order to evaluate the performance of FSS variations, more specifically FSS-Vanilla, FSS-SAR and FSS-NPSS-SAR, we compared their outputs against results obtained using PSO. In the PSO version chosen [15], for a particle $x_i$, its position in iteration $t + 1$ is defined in Eq. 20:

$$x_i(t + 1) = x_i(t) + \chi[v_i + c_1 r_1 (p_{ib} - x_i(t)) + c_2 r_2 (g_{ib} - x_i(t))],$$

(20)

where $\chi = \frac{2}{[2-(c_1+c_2)-\sqrt{(c_1+c_2)(c_1+c_2)-4}]}$ is known as constriction factor and $r_1$ and $r_2$ are uniformly distributed random numbers in the interval $[0;1]$. In this version $c_1$ and $c_2$ must satisfy $c_1 + c_2 \geq 4$. For this work we have chosen $c_1 = c_2 = 2.1$. The solution of MMWALP with PSO follows the same flow for FSS, as described in Section 4.

FSS, FSS-SAR, FSS-NPSS-SAR and PSO were used to solve all the three aforementioned test instances. We solved the problem instances using the same parameters shown in Table 1. Each test case was repeated 450 times. In SAR versions of FSS, the value of $\alpha$ parameter in iteration $t$ is $\alpha = 0.8e^{-0.007t}$ in all tests performed.
Table 1: Test Parameters

| $W_{scale}$ | 10000 |
| $Step_{ind}$ | 20% |
| $Step_{vol}$ | 20% |

Search Space $[-1000; 1000]$

Table 2: Simultaneous Balancing Sequencing results

| Output | Dataset | Measure  | SAR      | NPSS-SAR  | PSO       | Vanilla  |
|--------|---------|----------|----------|-----------|-----------|----------|
|        |         |          |          |           |           |          |
|        | Small   | Mean     | 4496.5119| 4491.2129| 4474.8687| 4493.6283|
|        |         | SD       | 28.0606  | 33.1336   | 51.1579   | 31.2959  |
| CW     | Medium  | Mean     | 3350.5474| 3356.0287| 3395.3011| 3352.2192|
|        |         | SD       | 19.9296  | 21.0870   | 32.2869   | 20.5765  |
|        | Large   | Mean     | 7254.7956| 7276.9778| 7148.5438| 7253.4780|
|        |         | SD       | 117.3200 | 141.9520  | 67.1200   | 120.0066 |
| WP     | Small   | Mean     | 6.0000   | 6.0000    | 6.0111    | 6.0000   |
|        |         | SD       | 0.0000   | 0.0000    | 0.1049    | 0.0000   |
|        | Medium  | Mean     | 6.0000   | 6.0000    | 6.0000    | 6.0000   |
|        |         | SD       | 0.0000   | 0.0000    | 0.0000    | 0.0000   |
|        | Large   | Mean     | 10.0289  | 10.0844   | 9.1956    | 10.0067  |
|        |         | SD       | 0.5435   | 0.6345    | 0.4189    | 0.5643   |
| WL     | Small   | Mean     | 5575.7109| 5577.9851| 5598.0277| 5574.0965|
|        |         | SD       | 20.7721  | 21.1984   | 22.3673   | 22.5772  |
|        | Medium  | Mean     | 3404.2130| 3406.0236| 3442.2676| 3406.2830|
|        |         | SD       | 24.9794  | 25.9916   | 34.7513   | 25.1982  |
|        | Large   | Mean     | 8024.2664| 8025.5771| 7985.7387| 8024.2224|
|        |         | SD       | 73.1592  | 74.3165   | 79.2672   | 73.5636  |
| IUC    | Small   | Mean     | 200.9244 | 231.9178 | 219.2689 | 200.9844 |
|        |         | SD       | 123.3553 | 126.2045 | 137.5588 | 127.9451 |
|        | Medium  | Mean     | 267.6311 | 325.3444 | 360.3489 | 274.6689 |
|        |         | SD       | 145.3887 | 144.2873 | 106.0932 | 147.3617 |
|        | Large   | Mean     | 245.1978 | 225.3756 | 292.2111 | 241.6711 |
|        |         | SD       | 0.5435   | 0.6345    | 0.4189    | 0.5643   |

We have applied the widely used Analysis of Variance (ANOVA) technique [36] to establish whether the results obtained are significantly different and, if so, which algorithm is better for each one of the criteria considered.

To apply ANOVA, normality of data should be guaranteed. Thus, 450 outputs for each test case were grouped in samples of size 15 and the means of those were considered as input data for the ANOVA, which results in 30 samples per algorithm for each instance. We applied Shapiro-Wilk test [27] and then concluded that normality is guaranteed for the results obtained in all test cases.

5.1 Results

First of all, a comparison of the four different solution approaches was performed in the three selected test instances in order to evaluate their efficiencies when trying to maximize Completed Work.

Table 2 presents the obtained results for different outputs: Completed Work (CW), Workload (WL), number of workplaces (WP) and number of iterations until convergence (IUC), i.e. the last iteration in which an improvement higher than $10^{-4}$ occurred. For these tests, the workstation length was set to $L = 0.95$ and the maximum number of workplaces per workstation was set to 3. The FSS versions are presented only by: Vanilla, SAR and NPSS-SAR.

An one-way 95% confidence ANOVA was performed in order to evaluate the performance differences among the approaches employed. Calculated degrees of freedom were $v_1 = 2$ and $v_2 = 87$, thus $F_{ref} = 4.89$ in all cases.
Table 3 presents the $F$ statistic provided by ANOVA for Small and Medium data sets. The varying factor was the solution procedure.

|                | Small | Medium |
|----------------|-------|--------|
| **CW**         | 27.35 | 373.44 |
| **IUC**        | 7.82  | 41.60  |

$F$ calculated in all test cases considered was greater than $F_{ref}$ which means that we are able to reject the hypothesis that outputs provided by the different approaches do not differ. This can be better illustrated in Figure 4 where the pooled confidence interval for Completed Work and IUC are presented for the Small and Medium instances.

Figure 4: Confidence intervals based on the pooled standard deviation for the Small and Medium data sets

In the cases of the Completed Work, the overlaps on pooled confidence intervals in either Small and Medium instances suggest the data compared is likely to come from the same probability distribution and do not differ significantly. However, PSO algorithm presented a lower output in Small case and a best result in the Medium problem. In terms of convergence speed, IUC for FSS-NPSS-SAR was always higher than the other approaches, which means that this version usually takes longer to converge with no relevant output improvement.

Specifically for the Large data set, a more detailed analysis was carried out and ANOVA was applied for all the outputs available. Table 4 presents the $F$ statistic values for: Completed Work (CW), number of workplaces (WP), the ratio Completed Work/Workload (CW/WL) and the number of iterations until convergence (IUC). Once more, the test performed suggest that at least one of the approaches employed differs of the others for all the outputs considered.

The aforementioned can be better analyzed with basis in Figure 5. The pooled confidence intervals for WP indicate that PSO was able to find balancing outputs better than those returned by FSS variations. However, the results of the ratio Completed Work/Workload make clear that a higher fraction of the workload is completed in the outputs provided by FSS-NPSS-SAR. This
means that the balancing/sequencing solution provided by PSO seems more efficient once it utilizes less workplaces, but in the other hand this solution requires more utility work (or line stoppages) in order to finish the production plan. The higher number of workplaces of FSS-NPSS-SAR increases the workload but also provides a higher proportion of completed work requiring less utility work or reducing the risk of line stoppages. For IUC, PSO took longer to converge in the Large instance.

Furthermore, a comparison was performed in order to study the effect of the variation of the workstation maximum permitted number of workplaces per workstation. Values 2, 4 and 6 were employed as independent variables in an ANOVA using large data set and FSS-SAR as solution approach. Results are presented in Table 5 and $F$ values are shown in Table 6.

The pooled confidence intervals are presented in Figure 6. The case in which the maximum number of workplaces is 4 is the one in which the best ration CW/WL occurred due to the excess of displacements in cases where few workplaces are allowed and the excess of precedence relation conflicts in the case which several workplaces are allowed.
Table 5: Simultaneous Balancing and Sequencing results with the variation of the maximum number of workplaces per workstation

|       | 2     | 4     | 6     |
|-------|-------|-------|-------|
| WP    | Mean  | 10.0000 | 11.9844 | 12.0000 |
|       | SD    | 0.0000  | 0.1239  | 0.0000  |
| WL    | Mean  | 8273.4059 | 8121.8921 | 7964.2821 |
|       | SD    | 50.5872  | 78.2892  | 59.3590  |
| CW/WL | Mean  | 0.9194  | 0.9612  | 0.9577  |
|       | SD    | 0.0055  | 0.0062  | 0.0053  |
| CW    | Mean  | 7606.7499 | 7806.5702 | 7627.5828 |
|       | SD    | 29.5040  | 61.8566  | 52.6124  |

Table 6: $F$ values for the variation of the maximum number of workplaces per workstation

| Output  | $F$  |
|---------|------|
| CW      | 2561.24 |
| CW/WP   | 6289.95 |

Figure 6: Confidence intervals based on the pooled standard deviation for the variation of the maximum number of workplaces per workstation

(a) Completed Work  
(b) CW/WL ratio

Figure 6: Confidence intervals based on the pooled standard deviation for the variation of the maximum number of workplaces per workstation
6 Conclusion

In this work we have applied an nature-inspired metaheuristic called Fish School Search algorithm in order to solve two relevant decision problems: Assembly Line Balancing Problem and Mixed Model Assembly Line Sequencing Problem. The relevance of the tackled problems relies on their NP-Hard nature as well as in practitioners interest within Industry.

FSS is a continuous optimization technique and both of aforementioned problems have combinatorial nature. Hence, mapping procedures were applied in order to convert fishes positions into sequences of tasks for the balancing portion of the solution approach and a sequence of models for the sequencing part.

Furthermore, both problems are known to be strongly interdependent. Thus, a simultaneous approach was applied in a sense that first the balancing problem is solved and a list of the best solutions found within the search are used as input for the sequencing problem solution. The best combination of balancing/sequencing are returned as solution of the problem.

Tests were performed in three different data sets and a comparison between the original version of FSS and two variations referred as FSS-SAR and FSS-NPSS-SAR as well as Particle Swarm Optimization algorithm was carried out. Results show that, in general, FSS versions generate solution requiring more workplaces than PSO. However, this fact ends up resulting in more efficient results for the sequencing portion once the outputs of FSS require less use of utility work. FSS-NPSS-SAR was the variation of FSS which returned the best results.

As future work, some niching able metaheuristic [14, 22] may be applied in the balancing portion of the solution approach. Hence, the list containing balancing best found solutions would be directly the output of the multi-solution procedure. Moreover, different combinations of the approaches applied in each portion of the solution (balancing/sequencing) can be applied for performance evaluation.
References

[1] I. M. C. Albuquerque, J. B. Monteiro, F. B. L. Neto, and A. M. Oliveira. Fish school search variations and other metaheuristics in the solution of assembly line balancing problems. Submitted to The 16th International Conference on Intelligent Systems Design and Applications, 2016.

[2] I. M. C. Albuquerque, J. B. Monteiro, F. B. L. Neto, and A. M. Oliveira. Solving assembly line balancing problems with fish school search algorithm. Submitted to 2016 IEE-Symposium Series on Computational Intelligence, 2016.

[3] J. F. Bard, E. Dar-Elj, and A. Shtub. An analytic framework for sequencing mixed model assembly lines. International Journal of Production Research, 30(1):35–48, 1992.

[4] J. F. Bard, A. Shtub, and S. B. Joshi. Sequencing mixed-model assembly lines to level parts usage and minimize line length. International Journal of Production Research, 32(10):2431–2454, 1994.

[5] C. J. A. Bastos-Filho, F. B. D. Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima. A novel search algorithm based on fish school behavior. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, pages 2646–2651, 2008.

[6] J. Bautista and A. Cano. Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimisation and interruption rules. European Journal of Operational Research, 210(3):495–513, 2011.

[7] J. Bautista, A. Cano, and R. Alfaro. Modeling and solving a variant of the mixed-model sequencing problem with work overload minimisation and regularity constraints. An application in Nissan’s Barcelona Plant. Expert Systems with Applications, 39(12):11001–11010, 2012.

[8] J. Bautista and J. Cano. Minimizing work overload in mixed-model assembly lines. International Journal of Production Economics, 112(1):177–191, 2008.

[9] C. Becker and A. Scholl. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research, 168(3):694–715, 2006.

[10] C. Becker and A. Scholl. Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure. European Journal of Operational Research, 199(2):359–374, 2009.

[11] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR), 35(3):268–308, 2003.

[12] N. Boysen, M. Fliedner, and A. Scholl. Assembly line balancing: Which model to use when? International Journal of Production Economics, 111(2):509–528, 2008.

[13] N. Boysen, M. Fliedner, and A. Scholl. Sequencing mixed-model assembly lines: Survey, classification and model critique. European Journal of Operational Research, 192(2):349–373, 2009.

[14] F. Buarque De Lima Neto and M. Gomes Pereira de Lacerda. Weight based fish school search. In Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, pages 270–277. IEEE, 2014.

[15] M. Clerc and J. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. Evolutionary Computation, IEEE Transactions on, 6(1):58–73, 2002.

[16] S. G. Dimitriadis. Assembly line balancing and group working: A heuristic procedure for workers’ groups operating on the same product and workstation. Computers and Operations Research, 33(9):2757–2774, 2006.

[17] P. Fattahi, A. Roshani, and A. Roshani. A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. The International Journal of Advanced Manufacturing Technology, 53(1-4):363–378, 2011.
[18] N. Hamta, S. M. T. Fatemi Ghomi, F. Jolai, and M. Akbarpour Shirazi. A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *International Journal of Production Economics*, 141(1):99–111, 2013.

[19] G. Kanawaty. *Introduction to work study*. International Labour Organization, 1992.

[20] W. Kubiak. Minimizing variation of production rates in just-in-time systems: A survey. *European Journal of Operational Research*, 66(3):259–271, 1993.

[21] F. Longo and G. Mirabelli. Effective design of an assembly line using modelling and simulation. *Journal of Simulation*, 3(1):50–60, 2009.

[22] S. S. Madeiro, F. B. De Lima-Neto, C. J. A. Bastos-Filho, and E. M. Do Nascimento Figueiredo. Density as the segregation mechanism in fish school search for multimodal optimization problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6729 LNCS(PART 2):563–572, 2011.

[23] J. B. Monteiro, I. M. C. Albuquerque, F. B. L. Neto, and F. V. S. Ferreira. Improved search mechanisms for the fish school search algorithm. *Submitted to The 16th International Conference on Intelligent Systems Design and Applications*, 2016.

[24] J. B. Monteiro, I. M. C. Albuquerque, F. B. L. Neto, and F. V. S. Ferreira. Optimizing multi-plateau functions with fss-sar (stagnation avoidance routine). *Submitted to 2016 IEEE-Symposium Series on Computatational Intelligence*, 2016.

[25] H. Moradi and M. Zandieh. An imperialist competitive algorithm for a mixed-model assembly line sequencing problem. *Journal of Manufacturing Systems*, 32(1):46–54, 2012.

[26] K. Okamura and H. Yamashina. A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *International Journal of Production Research*, 17(3):233–247, 1979.

[27] N. M. Razali and Y. B. Wah. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of Statistical Modeling and Analytics*, 2(1):21–33, 2011.

[28] A. Roshani, A. Roshani, A. Roshani, M. Salehi, and A. Esfandyari. A simulated annealing algorithm for multi-manned assembly line balancing problem. *Journal of Manufacturing Systems*, 32(1):238–247, 2013.

[29] A. Scholl. *Balancing and sequencing of assembly lines*. Physica-Verlag Heidelberg, 1999.

[30] A. Scholl and C. Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. In *European Journal of Operational Research*, volume 168, pages 666–693, 2006.

[31] A. Scholl, R. Klein, and W. Domschke. Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. *Journal of Heuristics*, 4(4):359–381, 1998.

[32] A. S. Simaria and P. M. Vilarinho. 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines. *Computers and Industrial Engineering*, 56(2):489–506, 2009.

[33] C. Solnon, V. D. Cung, A. Nguyen, and C. Artigues. The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF’2005 challenge problem. *European Journal of Operational Research*, 191(3):912–927, 2008.

[34] N. T. Thomopoulos. Line Balancing-Sequencing for Mixed-Model Assembly. *Management Science*, 14(2):B–59–B–75, 1967.

[35] M. K. Uddin, M. C. Soto, and J. L. M. Lastra. An integrated approach to mixed-model assembly line balancing and sequencing. *Assembly Automation*, 30(2):164–172, 2010.
[36] B. J. Winer, D. R. Brown, and K. M. Michels. *Statistical principles in experimental design.* McGraw-Hill New York, 1971.

[37] C. A. Yano. *Survey, development, and application of algorithms for sequencing paced assembly lines.* 1989.

[38] C. a. Yano and R. Rachamadugu. Sequencing to Minimize Work Overload in Assembly Lines with Product Options. *Management Science*, 37(5):572–586, 1991.