9th International Conference on Industrial Engineering and Industrial Management
XXI International Conference on Industrial Engineering and Operations Management
International IIE Conference 2015
Aveiro, Portugal. July 6-8, 2015

Application of Hybrid Symbiotic Organism Search on Flow Shop Scheduling with a New Learning Effect

Amirian H, Sahraeian R[1]

Abstract The present article proposes a hybrid learning effect model which takes into account the previous experience of the operator, separates the machine/manual times and considers truncation. The developed model is fitted to experimental data to investigate its accuracy. The fits are compared with those of four other well-known position based learning models. Next, each learning model is applied to the large scale flow shop problems which makes them strongly *NP*-hard. Hence, the problems are tackled by a hybrid meta-heuristic named Symbiotic Organism Search Simulated Annealing (*SOS-SA*). The algorithm combines the fast and easy implementation of *SOS* with the powerful local search of *SA*. The proposed algorithm is tested on flowshop benchmark problems and the results show its validation.

Keywords: Scheduling; Learning Effect; Flow shop; Symbiotic Organism Search; Simulated Annealing

1 Introduction

Classic scheduling treats the processing times of jobs on different machines as constant input data. Empirical analysis of processing times, however, indicates contradictory results (Biskup, 1999). In other words, the workers are likely to demand less time to perform tasks as repetitions take place due to increasing familiarity with the operation, the tools and the workplace, and because shortcuts to the task execution are found (Jaber, 2011). This phenomena is known as "learning effect" in scheduling. The importance of the effect lies in the fact that under the effect of learning, the optimum schedule might change. Thus ignoring the effect

[1]Rashed Sahraeian (e-mail: sahraeian@shahed.ac.ir)
Department of Industrial Engineering, College of Engineering, Shahed University, Tehran, Iran

leads to considerable loss in profit, time and energy. Learning effects are formulated according to empirically developed learning curves (*LC*s). A *LC* is a mathematical description of workers' performance in repetitive tasks. With each passing year new models for position based learning effect are proposed but most are based on four commonly adopted learning curves. These are classic log-linear, Dejong's, *S*-curve and Plateau *LC*s. In the present study, the models based on these *LC*s are compared with a newly proposed hybrid learning effect. Understanding the differences between *LC*s with this comparative analysis leads to the selection of an appropriate learning model in respect to the work environment. This, in turn, leads to a good estimation of processing times, a proper scheduling and finally, a considerable reduction in production cost. Moreover, to emphasize the applicability of learning effects, the models are carried out on large scale classic flow shop problems using a hybrid version of a recently introduced meta-heuristic named Symbiotic Organisms Search (Cheng & Prayogo, 2014) with Simulated Annealing (Kirkpatrick et al., 1983). The rest of this paper is organized as follows. Section 2, shows the available learning curves formulations and examines our proposed model. In section 3, we discuss the pros and cons of each curve. Section 4, gives a brief introduction on *SOS-SA* and shows the results achieved by adding learning effects to classic flow shop systems. The paper is then concluded in section 5.

2 Position Based Learning Curve Models

Let p_{ijr} be the processing time of j^{th} job in r^{th} position of i^{th} machine, $a(a \le 0)$ the learning rate, $M(0 \le M \le 1)$ the machine time ratio, B the prior experience and C the steady-state performance of an operator. Now, we summarize the most popular position based *LC*s previously used in the scheduling literature (Table 1).

Table 1 Summary of commonly used *LCs*

Model	Formulation	*Parameters*	*Feature*	*Reference*
Log-linear	$p_{ijr} = p_{ij} r^a$	a	Added the learning rate	Biskup (1999)
Dejong	$p_{ijr} = p_{ij}.(M + (1-M).r^a)$	a, M	Separation of manual and machine times	Okolowski & Gawiejnowicz (2010)
S-Curve	$p_{ijr} = p_{ij}.(M + (1-M).(r+B)^a)$	a, M, B	Adding Operator's experience	Jaber (2011)
Plateau	$p_{ijr} = C + p_{ij}.r^a$	a, C	Added Truncation effect using steady-state performance	Baloff (1971)

Proposed Hybrid LC: In our paper (Amirian & Sahraeian, 2015), we introduced a model based on the previous learning curves as (2.1). However, the model used *maximum* operator for truncation. This led to higher errors in comparison to the

models such as Plateau that uses *addition* operator as a means of truncation. Hence, imitating Plateau *LC*, the model is modified as (2.2). This modification decreases the mean square error while maintaining the generalization of the model.

$$p_{ijr} = p_{ij}.(M_{ij} + (1 - M_{ij}).max\{(B_j + r)^{a_i}, C_{ij}\}). \quad i = 1,...,m, j, r = 1,\ldots,n \qquad (2.1)$$

$$p_{ijr} = p_{ij}.(M_{ij} + (1 - M_{ij}).(C_{ij} + (B_j + r)^{a_i})). \quad i = 1,...,m, j, r = 1,\ldots,n \qquad (2.2)$$

3 Which Learning Curve Should Be Used?

It's a constant challenge for managers to find the correct *LC* model for a work environment. The best way to make sure that the learning model is in tune with a particular production is by gathering the empirical data on the site. Then we can estimate the parameters by fitting the models to the data using mean square method (*MSE*). The *LC* with the lowest *MSE* is usually considered to be the most precise option. However, other factors such as the prediction ability of the model and the number of required parameters should be considered as well before choosing a learning model. An example is suggested by Zhang et al. (2014) for a construction project in China. The data for the accumulative average time needed to finish 40 levels of a high-rise project can be found at their paper. To test our hybrid model and compare it to the others, we fitted each model to the first 20 real data of the construction project and calculated the mean square error of each model in table 2. Then we predicted the 40 points using the fitted model in figure 1. As can be seen in table 2, the Dejong's and Plateau have the lowest *MSE*, closely followed by the proposed hybrid and log-linear model. *S*-curve, however, showed the worst performance. Now, let's examine figure 1, where the variance of achieved results from the real data for each model are plotted. Note that values close to zero are more desirable. As can be seen in figure 1, the proposed hybrid has the lowest variance in all 40 points which basically means it has found the real data with very little error. Once again Dejong and Plateau show similar performance. Next comes log-linear, and finally *S*-curve shows the worst performance. However, in evaluating a *LC*, most researchers believe that the model with the lowest *MSE* is the best option. This is only partially true, since a mean square error which is calculated on limited data (e.g. 20 data here), does not guarantee a good prediction ability of that model for all the points (e.g. all 40 points). Now, in relation to human learning, there is always a point where the operator stops learning since s/he has fully grasped the workings of the process and from that point on there is no reduction in the processing times. Imitating the literature, we will call this phenomena the "truncation effect" and emphasize that only a model that takes this effect into account can be a good option. Among the available models, both the Plateau *LC* and the proposed hybrid model have this feature. However, our model takes into account the operator's former experience while Plateau *LC* forgoes this feature.

Table 2 Fitted function of different learning curves

Learning model	Fitted function	*MSE*

Learning model	Fitted function	*MSE*
Log-linear	$p_r = 10.97(r)^{-0.1665}$	0.0034
Dejong	$p_r = 2 + (11.035 - 2).(r)^{-0.2185}$	0.0028
S-Curve	$p_r = 2 + (14.158 - 2).(r + 2)^{-0.3184}$	0.0265
Plateau	$p_r = 1.73 + 9.29(r)^{-0.2097}$	0.0030
Proposed hybrid	$p_r = 2 + (14.54 - 2).\ (0.289 + (2 + r)^{-0.7822}).$	0.0034

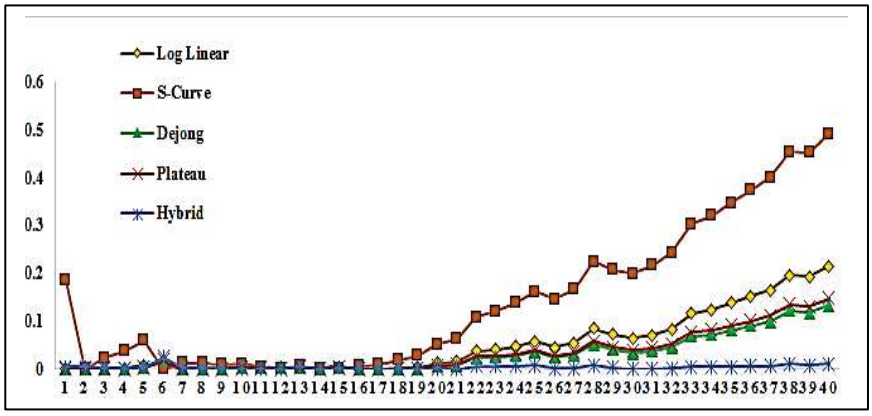

Fig. 1 Variations of different learning curves from the actual data

4 Application of Learning in Large Scale Flow shops

Among the different environments of scheduling, flow shops are of the most applicable and well-known systems, and have been proved to be strongly *NP*-hard. In a classic flowshop system, *n* jobs in sequence go through *m* machines one by one so that no machine can process more than one job at one point in time and vice versa. If the sequence of the jobs on all machines is the same, then the system is called permutation flowshop. In such systems, the goal is often to find the maximum completion time among the jobs on the last machine (i.e. the makespan). Due to the *NP*-hardness of the flowshop systems, meta-heuristics are usually proposed as solution techniques. Similarly, in this section, first a hybrid meta-heuristic is proposed and is used to calculate the makespan of *classic flowshop* benchmarks (Taillard, 1993). Once the algorithm is verified, we use it to solve the *learning effect flowshop* problems with each of the *LC*s previously presented.

4.1 Hybrid SOS-SA for Flowshop Scheduling

SOS (Cheng and Prayogo, 2014) is a newly introduced population based meta-heuristic. It begins with a randomly-generated population called ecosystem. Each individual (organism) in the population represents a candidate solution to the corresponding problem. The phases in *SOS* are governed by imitating the biological interaction between two random organisms in the ecosystem. These phases are mutualism, commensalism and parasitism. Mutualism is based on mutual benefit from a relation. Here, two organisms are selected randomly from the ecosystem (i.e. X_i, X_j). Then *Mutual Vector* is calculated as $MV = (X_i + X_j)/2$. Since in a relation, one party can gain more/less than the other, a benefit factor (*BF*) is introduced for each of the selected organisms. The new solutions in (4.1), (4.2) are calculated using X_{best} which is the best solution in the current population. The new solutions replace the older ones as long as they are better, otherwise they are discarded:

$$X_{iNew} = X_i + rand(0,1).(X_{best} - MV.BF_1) \tag{4.1}$$

$$X_{jNew} = X_{ij} + rand(0,1).(X_{best} - MV.BF_2) \tag{4.2}$$

In commensalism phase, one individual benefits from the interaction while the other is unaffected. Here, organism X_i is selected randomly to be improved by another randomly chosen organism X_j as follows in (4.3). Once again the new solution is accepted only if it is better than the old one.

$$X_{iNew} = X_i + rand(-1,1).(X_{best} - X_j) \tag{4.3}$$

Parasitism: Here, the relation between the selected organisms is harmful. First a vector X_i is selected and modified in some dimensions randomly. The resultant vector is called parasite vector which acts as a disease, ready to infect another organism X_j ; known as host. If the parasite vector is better than the host X_j then it replaces it.

SOS is an easy and fast algorithm since it has few steps and given a good initial population it finds the global optimum in a short time. However, this algorithm is only tested in continuous space where *SOS* has been reported to show utmost precision (Cheng and Prayogo, 2014). When tested on a discrete problem such as flowshop system, however, more often than not, the algorithm finds a local optimum solution rather than the global optimum which shows low diversity in *SOS*. The concept of adding a powerful local search to a meta-heuristic has been employed by many researchers. Similarly, to improve *SOS* in diversity, we have added the conditional local search mechanism used in *SA* to the algorithm. This way, the population of individuals found in each iteration is improved greatly in diversity and somewhat in convergence before entering the *SOS* loop. Note that *SA* is

employed on the whole population so that for every individual in the population, a neighbor is created, then the neighbor is evaluated against the original solution. If the neighbor is better, it replaces the original solution, if not, then the neighbor is accepted conditionally. For more information on *SA*, we refer the interested reader to Kirkpatrick et al. (1983). The combination of *SA* and *SOS*, returns good results in flowshop scheduling problems. A brief overview of this algorithm can be given as follows[2]:

Step 1: Problem definition, set SA parameters, maximum iterations, population size
Step 2: Initialization - Create random population and evaluate it
Step 3: Main Loop
For main=1: maximum iterations
Step 3-1: SA loop:
For out=1: outer loop
For in=1: inner loop
For pop=1: population size loop
Create Neighbor
If the neighbor is better
Accept the neighbor
Else
Accept the neighbor conditionally
End if
End of population size loop
End of inner loop
Update the temperature
End of outer loop
End of SA Loop (Step 3-1)
Step 3-2: Sort Population
Step 3-3: SOS loop:
For i=1: population size
Mutualism Phase
Commensalism Phase
Parasitism Phase
End of SOS Loop (Step 3-3)
End of Step 3 (Main Loop)
Step 4: Display the results

The experiments are conducted on a PC with Intel i5 CPU @ 1.70 GHz with 4 GB of RAM with a population size of 100 and maximum iterations of 5 and 50 for the SA loop and the main loop respectively (i.e. 250 iterations overall). The code is tested on the first 10 Taillard's benchmark problems (i.e. *Ta001-Ta010*) to calculate the makespan and the results are shown in Table 3. The error ratio (*ER*) is calculated as $ER = (R - Opt) / Opt$ where R is the achieved result and Opt is the known

[2] The MATLAB implementation of this algorithm is stored at http://le-scheduling.blogfa.com/ for comparison purposes.

optimal value of the objective extracted from the available benchmarks. Note that lower *ER* is more desirable since it shows higher precision.

Table 3 Testing hybrid *SOS-SA* on Taillard's benchmarks[1]

Model	Ta001	Ta002	Ta003	Ta004	Ta005	Ta006	Ta007	Ta008	Ta009	Ta010
Optimum Makespan	1278	1359	1081	1293	1235	1195	**1234**	1206	1230	1108
Makespan by *SOS-SA*	1278	1359	1081	1293	1235	1195	**1251**	1206	1230	1108
ER	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**	0.00	0.00	0.00
Time	290.9	288.0	287.5	266.9	380.5	390.1	905.7	965.2	1147.3	1142.7

[1]The results are the best result of 10 runs and time is recorded in seconds

In table 3, out of 10 problems, *SOS-SA* found the best point for nine tests with the *ER* of zero which indicates the efficiency of the algorithm. For problem Ta007, the result is found by a 1% deviation (i.e. *ER*=0.01). We also test the effectiveness of adding *SA* to *SOS*. Table 4 shows the performance of *SOS* with and without *SA*. As can be seen in table 4, adding *SA* lowers *ER* which indicates higher precision. In table 5, *SOS-SA* is applied to the flow shop scheduling for different learning models and the results are reported. The learning parameters are set as $a = -0.322$, $M = 0.5, B = 0.5$ and C is set to 50% of the minimum processing times of all jobs. Since there is no benchmark on *learning effect flowshop*, the data in table 5 is given simply for comparison purposes.

Table 4 Testing *SOS* with and without *SA* (The results are the *best* of 10 runs)

	Ta001	Ta002	Ta003	Ta004	Ta005	Ta006	Ta007	Ta008	Ta009	Ta010
ER (SOS)	0.014	0.005	0.019	0.019	0.012	0.012	0.013	0.014	0.020	0.013
ER (SOS-SA)	0.000	0.000	0.000	0.000	0.000	0.000	0.013	0.000	0.000	0.000
Best Solution found (SOS)	1297	1366	1102	1318	1250	1210	1251	1224	1255	1123
Best Solution found (SOS-SA)	1278	1359	1081	1293	1235	1195	1251	1206	1230	1108

Table 5 Makespan achieved by *SOS-SA* on different *LC* models[1]

Model	Ta001	Ta002	Ta003	Ta004	Ta005	Ta006	Ta007	Ta008	Ta009	Ta010
Log-linear	632.58	748.56	607.53	704.79	616.22	629.15	648.45	642.05	663.92	599.41
Dejong	940.29	1066.55	851.58	998.32	906.56	908.48	942.40	920.87	948.88	852.52
S-Curve	934.78	1047.33	835.78	982.40	901.14	898.53	933.30	911.95	932.75	841.34
Plateau	644.39	796.57	625.85	740.84	629.66	660.99	661.60	665.34	676.65	610.62
Hybrid	982.13	1079.19	855.58	1007.12	935.22	935.29	963.46	934.64	963.25	857.58

[1]The results are the best of 10 runs

It can be used in evaluating a meta-heuristic on *learning effect flowshop* when the results of a specific *LC* (e.g. log-linear) found by *SOS-SA* is compared to the results of the same *LC* (i.e. log-linear) found by any other meta-heuristic. It is illogical to compare the result of one *LC* to another *LC* since each of them follows different concepts and any of them can be employed in a work environment, it is only the matter of precision needed and the characteristics of the workplace. For instance if in a work place, low precision is sufficient, then log-linear model is the ideal choice since it has only one parameter and relatively acceptable precision. Similarly, if all the job is done manually then there is no need to use Dejong's model.

5 Conclusion and Future Works

In the present article, log linear, Dejong, *S*-curve, Plateau learning models are compared to a proposed hybrid learning effect. The results indicate that Dejong, Plateau and the proposed hybrid have similar precision while the hybrid model has the best prediction ability and generalization. These learning models are added to the classic flow shop problems which are solved by a hybrid meta-heuristic. The meta-heuristic (named *SOS-SA*) combines the robustness and efficiency of symbiotic organism search with the powerful local search and diversification method of simulated annealing. *SOS-SA* is then tested on Taillard's benchmarks and the results show its accuracy. In the future, we hope to develop more applicable learning models and solution techniques in different scheduling environments.

6 References

Amirian H, Sahraeian R (2015) Augmented ε-constraint method in multi-objective flowshop problem with past sequence set-up times and a modified learning effect. Int J Prod Res, DOI: 10.1080/00207543.2015.1033033

Biskup D (1999) Single machine scheduling with learning considerations. Eur J Oper Res 115: 173-178

Baloff, N (1971) Extension of the learning curve: Some empirical results. Oper. Res. Quarterly 22: 329-340

Cheng M, Prayogo D (2014) Symbiotic Organisms Search: A new meta-heuristic optimization algorithm. Comput Struct 139: 98–112

Jaber M-Y (2011) Learning curves: Theory, models, and applications. CRC Press

Kirkpatrick S, Gelatt C-D, Vecchi M-P (1983) Optimization by Simulated Annealing. Sci. 220: 671-680.

Okolowski D, Gawiejnowicz S (2010) Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect. Comput Ind Eng 59: 272-279

Taillard E (1993) Benchmarks for basic scheduling problems. Eur J Oper Res 64: 278–285

Zhang L, Zou X, Kan Z (2014) Improved strategy for resource allocation in repetitive projects considering the learning effect. J Constr Eng Manage, doi: 10.1061/(ASCE)CO.1943-7862.0000896