A Featured Tuning of the Simulated Annealing Applied to the Open Shop Scheduling

Chaouqi Mohsine
OSIL Team LRI, ENSEM,
KM7, BP 8118 Route El Jadida
Casablanca, Morocco

Benhra Jamal
OSIL Team LRI, ENSEM,
KM7, BP 8118 Route El Jadida
Casablanca, Morocco

My Ali El Oualidi
OSIL Team LRI, ENSEM,
KM7, BP 8118 Route El Jadida
Casablanca, Morocco

ABSTRACT
The present paper discusses the open shop scheduling problem using a manual tuning of a simulated annealing algorithm's parameters. A comparison has been done between Taillard's Benchmarks for 60 instances, 2 dispatching rules and 1296 variants of the SA algorithm obtained by changing the initial solution, the epoch length, and the steps' number, the initial temperature, the neighborhood and the cooling scheme.

The gotten results lead to some interesting conclusions for the best choice of the parameters.

General Terms
Algorithm, parameters, tuning.

Keywords
Scheduling, Simulated annealing, epoch length, neighborhood, cooling scheme, tuning, and open shop.

1. INTRODUCTION
Shop scheduling involves the processing of a set of jobs on a set of machines by defining the time intervals in which the operations have to be processed. [1]

There are three basic types of shops: a flow-shop (each job is characterized by the same technological route), a job-shop (each job has a specific route) and an open shop (no technological route is imposed on the jobs). [2]

However there exist different objectives in scheduling optimization. The most known objective is the makespan Cmax minimization which is the time's span required to process all the jobs, i.e. the time from the beginning of the first operation until the end of the last operation. The second one is to minimize the flowtime, denoted by $\sum Cj$, which is the completion times' sum of all the jobs. Other objectives are the tardiness's minimization, the number of tardy jobs, etc.

In this article a manual tuning of the simulated annealing's parameters is made and the given results were compared with Taillard's benchmarks for 60 instances where $n=m \in \{4, 5, 7, 10, 15, 20\}$.

The present paper is organized as follow: Section 2 contains a short reminder of the different works found on the open shop scheduling optimization. Section 3 sets out the algorithm used and the tuning made. In section 4 the gotten results are presented and their interpretation. Finally this article ends with a conclusion and some perspectives.

2. LITERATURE REVIEW
As mentioned before the open shop is one type of the three basic shops where a set of n jobs $J_1, J_2, \ldots, J_n$ has to be processed on a set of m machines $M_1, M_2, \ldots, M_m$ with an arbitrary order. The operation (i, j) means the processing of job J_i on machine M_j.

One can find this kind of shop in a large aircraft garage with specialized work centers, in automobile repair, quality control centers, semiconductor manufacturing, teacher-class assignments, examination scheduling, and satellite communications as described in [3], [4] and [5].

There exist many works involving the open shop scheduling problem. One of them is that of Liaw [6] who used simulated annealing in case of minimizing makespan on a nonpremptive open shop. He proposed a neighborhood search. His algorithm was tested on randomly generated problems, benchmark problems. He got a good results but spent a lot of time to reach them (up to 3.5 hours per single run for an instance with n=m=30).

The same author, proposed in [7] a hybrid genetic algorithm (HGA) to resolove the open shop scheduling problem. The hybrid algorithm incorporates a local improvement procedure based on tabu search (TS) into a basic GA. The algorithm developed has been tested on randomly generated instances and on the benchmarks sets by Taillard [8] and Brucker et al. [9]. It has been found that this HGA outperformed other existing algorithms from the literature, and some benchmark instances have been solved to optimality for the first time.

Michael Andresen et al.[3] considered the problem of scheduling n jobs on m machines in an Open Shop environment with the minimization of total weighted tardiness as a goal. The main goal of their study was to find out which parameters have a strong influence and which have a smaller influence on the selection of an appropriate simulated annealing algorithm.

Fang et al. [10] suggested an algorithm which combines a GA with heuristic rules for the schedule construction. The algorithm has been tested on the benchmark instances from [8] using ten runs for each instance. By their tests, they discovered one new best known solution for a problem with 7 machines and 7 jobs and a problem with 10 machines and 10 jobs instance.

Rui Zhang et al.[11] executed a simulated annealing algorithm based on bottleneck jobs on an Open Shop scheduling problem. Their study attempt to minimize the total weighted tardiness.

Recently Bai et al [12] used a heuristic called general dense scheduling to solve the static and dynamic versions of the flexible open shop scheduling problem. The heuristic proposed brings forth some interesting results.

Naderi et al [13] studied the scheduling open shop problem with no intermediate buffer, called no-wait open shops under makespan minimization. They develop three mathematical models and propose metaheuristics based on genetic and variable neighborhood search algorithms. The results they got show that the models and metaheuristics are effective to deal with the no-wait open shop problems.

The simulated annealing algorithm proposed in this article was used by Chaouqi et al [14] in hybridization within the intuitive heuristic to perform a joint scheduling of production and maintenance in the job shop problem. On the same topic the authors used the Johnson's algorithm combined with a genetic algorithm and the intuitive heuristic to optimize three objectives of the flow shop problem. [15]

3. SIMULATED ANNEALING AND PARAMETERS TUNING

In the years 80 three researchers from IBM company – S. Kirkpatrick, C. D. Gelatt and M.P. Vecchi have proposed and published a new iterative method called Simulated annealing Kirkpatrick et al.[16], which avoid the local minimum. Since that discovery, this method has been tested in different fields like the design of the electronic circuits, the image processing etc.

Simulated annealing is inspired from a metallurgic process called annealing. The annealing process is a heat treatment that alters the physical and sometimes chemical properties of a material to increase its ductility and reduce its hardness, making it more workable. It involves heating a material to above its recrystallization temperature, maintaining a suitable temperature, and then cooling [17]. This strategy leads to a crystallized solid state, which is a stable state, corresponding to an absolute minimum of energy.

Similarly to the real process simulated annealing involves two steps; heating and cooling. In the first step a control parameter called temperature is introduced which must lead the system to an optimal state. In the second one this temperature is reduced during certain epochs within a cooling parameter till the end of the period then it will be reinitialized.

Instead of Hill Climbing method, the simulated annealing accepts a worse solution with a certain probability. This one depends on the decreasing temperature so a local optimum is avoided.

In this paper where the objective is the minimization of the makespan, the algorithm is as follow [18] :

BEGIN

Generate an initial feasible solution R AND calculate Cmax(R)

BestSol= R; BestCmax=Cmax(R); T=Initial temperature;

While (Stopping criterea not met) Do

R′ = best neighbor in between the generated neighbors of R;

ΔC = Cmax(R')-Cmax(R);

Prob=Rand(0,1);

If ((ΔC≤0) or (prob < exp(-ΔC/T))) Then

R= R'; Cmax(R):=Cmax(R');

If (Cmax(R)<BestCmax) then BestCmax=Cmax(R); BestSol= R;

End if

Update T according to the chosen cooling scheme;

End if

End while

END

Figure 1: Simulated algorithm applied to open shop problem with makespan minimization as an objective

Where R is the rank matrix, Cmax is the makespan and neighbor is the new solution generated according to the chosen neighborhood.

3.1 Rank matrix

The rank matrix R= (r_{ij}) describes a sequence graph G(MO, JO) which is a feasible combination of machine orders and job orders. The rank r_{ij} is the maximal number of operations on a longest path ending in operation (i, j) [2].

Let the case n=m=3 where the job orders are described by the matrix JO and the machine orders are described by the matrix MO.

$$MO = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \qquad JO = \begin{pmatrix} 3 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 2 & 3 \end{pmatrix}$$

The graphs related to these two matrices are shown in figure 2:

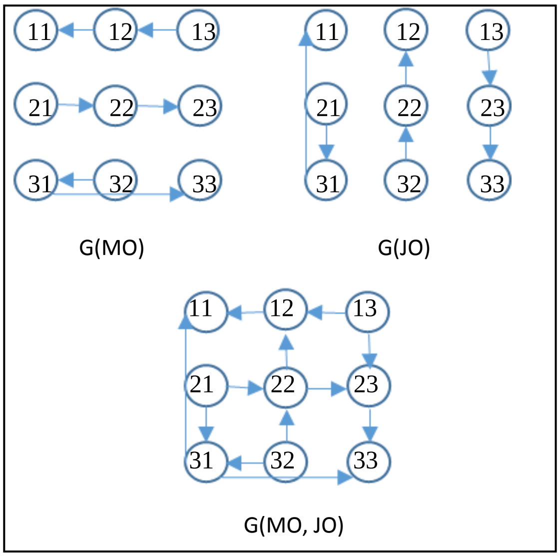

Figure 2: G(MO), G(JO) and G(MO, JO)

In the same figure the graph G(MO, JO) is given which is the combination of the two graphs.

3.2 Neighborhoods

In this paper nine neighborhoods are used. Here a brief description of them is given.

The first one is the PI neighborhood which stands for pairwise interchange where two arbitrary operations are interchanged. The second one is the API neighborhood. This neighborhood consists on interchanging two adjacent operations on a rank matrix to get a new neighbor, i.e an operation (i,j) is randomly selected and then it is interchanged with the predecessor or successor operation on machine M_j or of job J_i. If the pairwise interchange leads to a feasible schedule, it is accepted as a neighbor, otherwise another second operation is chosen to perform a new adjacent pairwise interchange. The neighborhood k_API is the same as the API with the exception of generating k neighbors. When generating a neighbor, the interchanges' number of two adjacent operations or on a machine of a job is randomly chosen. However in CR_API (resp. BL_API) neighborhood a Cmax critical operation (resp. block-end-operation) is interchanged with a directly adjacent operation.

In the neighborhood SHIFT one operation is changed in the relative order of operations, i.e. one operation is shifted left or right in the job order on one machine or in the machine order of one job. The operation is chosen randomly. Then another operation belonging to the same job or to be processed on the same machine is selected, e.g. let (i, k) be the other chosen operation. If the rank aik is smaller than a_{ij}, the rank a_{ij} is modified such that operation (i, j) appears immediately before (i, k) (it corresponds to a right shift of machine M_j in the machine order of job J_i). If the generated solution is infeasible, two other operations will be chosen randomly for performing a shift. In the BL_SHIFT a Cmax critical block-end-operation is shifted in the sequence. However the CR_SHIFT neighborhood which is a sub-neighborhood of the SHIFT neighborhood generates new neighbors from this one where the old ones satisfy a necessary condition for an improvement of the makespan, namely a critical path in the starting solution is destroyed, and there does not exist a path in the graph describing the generated neighbor which contains the same vertices as this critical path of the current starting solution.

Finally in this paper two other neighborhoods are used: the 3_CR where a Cmax critical operation is interchanged with a directly adjacent operation.[3] [19]

4. RESULTS AND DISCUSSION

In this section, computational results are presented for the simulated annealing. The parameters used in this study are:

Steps ϵ {5000, 10000, 15000, 20000}, Epoch_length ϵ {100, 200, 300}, initial_temperature ϵ {10, 20, 30}, cool_scheme ϵ {GEOMETRIC, LUNDYANDMEES}, initial_solution ϵ {SPT, LPT}, neighborhood ϵ {PI, API, k_API, CR_API, BL_API, SHIFT, BL_SHIFT, CR_SHIFT, 3_CR}, k=3, number of neighbors = 1, cool_paramter = 0.0005. Thus the number of variants is 4*3*3*2*2*9=1296 variants.

The time limit is 10 seconds fixed as a stopping criterion.

4.1 Case n° 1 : Cmax= Optimum

Here we describe the frequencies of variants which solved a certain number of problems out of the sixty instances given by Taillard, i.e. the final makespan calculated for these variants has reached the optimums of the corresponding problem. The table n°1 presents the gotten results in this first case.

Table 1. Number of variants by the number of solutions where Cmax=Opt

Variants' number	*Solutions' number*
831	0
196	1
97	2
55	3
50	4
26	5
18	6
13	7
6	8
3	9
1	10

One can see in the figure n° 3, most of variants do not generate an optimal solution for each instance from the sixty instances which is undesirable, i.e. the tuning suggested is not as good as expected. Only one configuration reached optimum solution for 10 out of 60. In this one the epoch length is 100, the number of steps is equal to 20000, the initial temperature is 10, the cooling parameter is 0.0005, the neighborhood is CR_SHIFT the initial solution is LPT and the cooling scheme is GEOMETRIC. This variant will be denoted by SA1.

The second good result, which is 9 optimums over 60, is given by three configurations: SA2= (epoch_length = 200, steps = 20000, t_start=10, cool_parameter = 0.0005, neighborhood = CR_SHIFT, cool_scheme = GEOMETRIC, init_sol = LPT), SA3= (100, 15000, 10, 0.0005, CR_SHIFT, GEOMETRIC, LPT) and SA4= (200, 20000, 10, 0.0005, CR_SHIFT, LUNDYANDMEES).

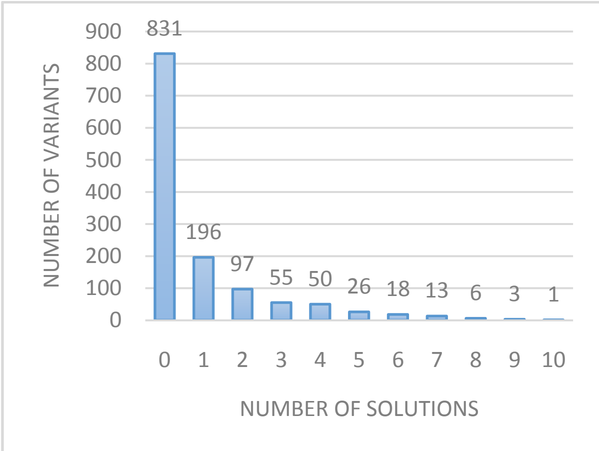

Figure 3: Number of variants by the number of solutions where Cmax=Opt

To get more information from the generated data a deep analyze of them is made by studying another case; where the ratio Optimum/Cmax is greater than or equal to 0.99.

4.2 Case n° 2: Optimum/Cmax>=0.99

In this second case a new distribution is given which differs from the previous one. One can observe in figure 4 a decreasing graph from 0 to 4 solutions then a new pic and again a decreasing graph from 5 to 18.

The maximal number of solutions with Opt/Cmax>=0.99 is 18 out of 60 reached by one variant: SA1, followed by 16 out of 60 obtained for these two configurations SA2 and SA3.

Here one can observe that the three variants SA1, SA2 and SA3 performed well in either cases.

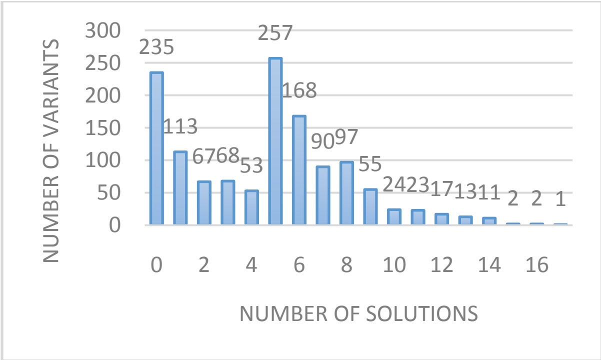

Figure 4: Number of variants by the number of solutions where Optimum/Cmax>=0.99

We grouped all the results of SA1, SA2 and SA3 applied to the Taillard's problem, with those given by SPT and LPT rules in table n° 2. The abbreviation LB means the lower bound and Opt stands for the optimum found in the literature.

Table 2. Results for the benchmark problems from Taillard

(nxm)_i	*LB*	*Opt*	*SPT*	*LPT*	*SA1*	*SA2*	*SA3*
(4x4)_1	186	193	228	219	193	193	193
(4x4)_2	229	236	276	256	236	239	236
(4x4)_3	262	271	304	299	271	271	272
(4x4)_4	245	250	307	260	250	250	250
(4x4)_5	287	295	348	317	295	303	295
(4x4)_6	185	189	225	239	189	189	189
(4x4)_7	197	201	247	218	201	201	201
(4x4)_8	212	217	233	248	217	217	217
(4x4)_9	258	261	282	282	261	261	261
(4x4)_10	213	217	235	225	217	217	217
(5x5)_11	295	300	333	344	301	303	305
(5x5)_12	255	262	297	297	263	270	263
(5x5)_13	321	323	404	364	336	338	337
(5x5)_14	306	310	317	369	322	319	316
(5x5)_15	321	326	392	358	339	339	339
(5x5)_16	307	312	353	360	322	327	324
(5x5)_17	298	303	340	357	314	308	307
(5x5)_18	292	300	369	343	312	301	314
(5x5)_19	349	353	372	418	367	362	357
(5x5)_20	321	326	375	371	332	326	332
(7x7)_21	435	435	507	465	459	453	455
(7x7)_22	443	443	485	526	459	470	459
(7x7)_23	468	468	538	527	500	498	500
(7x7)_24	463	463	492	527	494	472	474
(7x7)_25	416	416	461	443	434	429	434
(7x7)_26	451	451	518	494	469	490	469
(7x7)_27	422	422	464	451	445	436	445
(7x7)_28	424	424	482	494	442	447	442
(7x7)_29	458	458	520	488	472	472	472
(7x7)_30	398	398	435	445	416	429	407
(10x10)_31	637	637	685	661	661	661	661
(10x10)_32	588	588	658	643	606	607	633
(10x10)_33	598	598	679	672	650	662	647
(10x10)_34	577	577	632	591	584	591	584
(10x10)_35	640	640	693	701	688	677	686
(10x10)_36	538	538	559	556	554	555	555
(10x10)_37	616	616	672	637	635	637	637
(10x10)_38	595	595	651	686	623	644	630
(10x10)_39	595	595	655	621	621	621	621
(10x10)_40	596	596	633	636	630	621	634
(15x15)_41	937	937	987	972	970	972	972
(15x15)_42	918	918	937	972	971	971	969
(15x15)_43	871	871	891	878	876	878	878
(15x15)_44	934	934	975	965	939	949	964
(15x15)_45	946	946	959	999	990	990	990
(15x15)_46	933	933	981	952	951	950	952
(15x15)_4	891	891	919	955	950	954	954

7							
(15x15)_48	893	893	928	929	920	919	920
(15x15)_49	899	899	990	927	927	927	927
(15x15)_50	902	902	922	943	942	941	943
(20x20)_51	1155	1155	1194	1200	1196	1199	1200
(20x20)_52	1241	1241	1296	1296	1289	1296	1296
(20x20)_53	1257	1257	1304	1258	1258	1258	1258
(20x20)_54	1248	1248	1312	1274	1274	1264	1274
(20x20)_55	1256	1256	1277	1262	1260	1262	1260
(20x20)_56	1204	1204	1219	1215	1215	1215	1213
(20x20)_57	1294	1294	1407	1317	1313	1312	1317
(20x20)_58	1169	1169	1205	1216	1215	1215	1215
(20x20)_59	1289	1289	1306	1293	1292	1292	1292
(20x20)_60	1241	1241	1272	1265	1259	1259	1265

One can notice that the gotten results using the three variants SA1, SA2 and SA3 for the sixty instances are very close to the optimums found in the literature, and better than the two dispatching rules.

The worst value of the makespan given by the SA1 compared to the optimum value was 650 for the 33th instance where n=m=10, and the optimum is equal to 598. For this value the ratio Opt/Cmax is 92% which is acceptable.

Again the worst obtained values for the SA2 and SA3 are respectively 662 and 647 also obtained for the 33th instance. Thus the worst values for the ratio Opt/Cmax are 90% and 92% which still acceptable.

In addition to the ten first instances where Opt=Cmax for the SA1 variant, there are five other instances where the ratio Opt/Cmax is approximatively equal to 100%. Those instances are (5x5)_11, (5x5)_12, (20x20)_53, (20x20)_55 and (20x20)_59.

5. CONCLUSION

In this paper a comparison has been done between Taillard's Benchmarks for 60 instances in the open shop problem, the SPT and the LPT dispatching rules and 1296 variants of the simulated annealing algorithm obtained by changing the initial solution, the epoch length, the number of steps, the initial temperature, the neighborhood and the cooling scheme.

The gotten results are interesting in few cases. This is maybe due to the time limit set as a stopping criterion or it is due to the chosen simulated annealing algorithm implementation which needs some improvements. However some variants yield great results.

As a perspective of future studies, one can use some other tools like neural networks or Bayesian networks and a hybridization in between different algorithms to get a better tuning of the SA parameters then try new tests to get an optimum makespan for all instances.

6. REFERENCES

[1] M. L. Pinedo, *Scheduling*, vol. 1. Boston, MA: Springer US, 2012.

[2] F. Werner, "Genetic algorithms for shop scheduling problems: A survey," *Preprint*, 2011.

[3] O. Magdeburg, F. Mathematik, M. Andresen, H. Bräsel, M. Plauschin, and F. Werner, "Using Simulated Annealing for Open Shop Scheduling with Sum Criteria," pp. 1–26, 2008.

[4] C. Prins, "An Overview of Scheduling Problems Arising in Satellite Communications," *J. Oper. Res. Soc.*, vol. 45, no. 6, p. 611, Jun. 1994.

[5] C. Y. Liu and R. L. Bulfin, "Scheduling ordered open shops," *Comput. Oper. Res.*, vol. 14, no. 3, pp. 257–264, Jan. 1987.

[6] C. F. Liaw, "Applying simulated annealing to the open shop scheduling problem," *IIE Trans. (Institute Ind. Eng.*, vol. 31, no. 5, pp. 457–465, 1999.

[7] C.-F. Liaw, "A hybrid genetic algorithm for the open shop scheduling problem," *Eur. J. Oper. Res.*, vol. 124, no. 1, pp. 28–42, Jul. 2000.

[8] Taillard E., "Benchmarks for basic scheduling problems," *Eur. J. Oper. Res.*, vol. 64, pp. 1–17, 1993.

[9] P. Brucker, J. Hurink, B. Jurisch, and B. Wöstmann, "A branch & bound algorithm for the open-shop problem," *Discret. Appl. Math.*, vol. 76, no. 1–3, pp. 43–59, Jun. 1997.

[10] H. Fang and P. Ross, "A Promising Hybrid GA/Heuristic Approach for Open-Shop Scheduling Problems In Proceedings of the 11th European Conference on Arti cial Intelligence, John Wiley and Sons, 1994, pages 590{594.," no. 699, 1994.

[11] R. Zhang and C. Wu, "An Immune Genetic Algorithm Based on Bottleneck Jobs for the Job Shop Scheduling," no. 1, pp. 147–157, 2008.

[12] D. Bai, Z.-H. Zhang, and Q. Zhang, "Flexible open shop scheduling problem to minimize makespan," *Comput. Oper. Res.*, vol. 67, pp. 207–215, Mar. 2016.

[13] B. Naderi and M. Zandieh, "Modeling and scheduling no-wait open shop problems," *Int. J. Prod. Econ.*, vol. 158, pp. 256–266, Dec. 2014.

[14] M. Chaouqi and J. Benhra, "Recuit simulé hybride pour un ordonnancement conjoint de la production et de la maintenance dans un atelier job-shop," *Int. Work. Theory Appl. Logist. Transp. TALT15*, 2015.

[15] M. Chaouqi, J. Benhra, and A. Zakari, "Agile Approach for Joint Scheduling of Production and Maintenance in Flow Shop," *Int. J. Comput. Appl.*, vol. 59, no. 11, pp. 29–36, Dec. 2012.

[16] M. P. V. S. Kirkpatrick C. D. Gelatt, "Optimization by Simulated Annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] "Annealing - Wikipedia, the free encyclopedia." [Online]. Available: https://en.wikipedia.org/wiki/Annealing. [Accessed: 19-Mar-2016].

[18] K. Hasani, S. A. Kravchenko, and F. Werner, "Minimizing the makespan for the two-machine scheduling problem with a single server: Two algorithms for very large instances," *Eng. Optim.*, vol. 48, no. 1, pp. 173–183, 2016.

[19] M. Andresen, F. Engelhardt, and F. Werner, "LiSA - A Library of Scheduling Algorithms Handbook for Version 3.0," 2010.