Local Search Algorithms For Solving A Function With Five-Objectives And Release Dates on One-Machine

Hussein Abdullah Jaafar[1,*], Hanan Ali Chachan[2]
[1]Department of Mathematics, Open Educational College, Samawah, 66001 , Iraq.
[2]Department of Mathematics, College of Sciences, University of Mustansiriyah , Baghdad, 10001 , Iraq.
Emails: alhassen237@gmail.com; Hanan-altaai@yahoo.com

Abstract

In this research, the issue of scheduling n-jobs on one-machine is represented to minimize Five-Objectives-Function (FOF), for finding approximation solutions for the sum of completion time, total tardiness, total earliness, number of late jobs and late work with release date, this issue denoted by: $1/r_j/\sum_{j=1}^{n}(C_j + T_j + E_j + U_j + V_j)$. Hanan and Hussein used a branch and bound technique (B-a-B) to discovery an optimal solution path. Computational results showed the (B-a-B) technique was efficient in solving issues with up to (16- jobs). Because our issue is of a very difficult type (NP-hard), we suggest local search algorithms to discovery near optimal solution. The execution of local search techniques can be tested on large group of test issues. Computational results showed with up to (30000 jobs) in acceptable time.

Keywords: Branch and Bound (B-a-B)**;** Local Search (LS); Simulated Annealing (SA); Genetic algorithm (GA**).**

1.Introduction

A scheduling theory "focuses mainly on the optimal or efficient distribution of one or more resources to tasks over time", for finding approximation or near optimal solutions [1].

The theory of scheduling has been developed rapidly and significantly at the hands of researchers in this field. Many theories, solution techniques, graphs, timelines, and solution algorithms have been used to bring this theory to where it is today. For example, the complexity theory that has played a key in the classification of scheduling problems, were classified into categories according to the degree of difficulty of the solved or computational complexity [2].

Scientific research operations have primarily sought to ensure optimal solutions for scheduling problems by devising simple rules for solving as rule [Earliest-Due-Date] or rule [EDD], and [Shortest-Processing-Time] or rule [SPT] of Jackson and Smith respectively. Also, to obtain the exact solutions of hard issues, is usually used the main kinds from are enumerative techniques, of which, dynamic programming (DP) and (B-a-B) technique [1].

Due to the difficulties of many problems, which cannot be solved using enumerative methods. Many the approximate methods or heuristic method specifically local search methods, from prominent examples, Simulated annealing (SA), Particle Swarms optimization (PSO) and Genetic algorithm (GA) Tabu search (TS) and other, have been utilized to get high quality solutions [3].

For more than a decade, scheduling studies focused on single or (1-objective) performance measures, but in practical world, most an application, with more than (1-objective) [4].

In this research , we discuss the machine scheduling problems MSP with five objectives to measure performance and release dates on (1- machine), this issue is denoted by: $1/r_j/\sum_{j=1}^{n}(C_j + T_j + E_j + U_j + V_j)$ …(HH).

Hanan and Hussein [5] used (B-a-B) technique to find an optimal solution. Computational results showed (B-a-B) technique was good or effective in solving issues with up to 16-jobs.

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

Clearly this issue is (NP- hard), since as include sub issues are (NP- hard) in the strong sense, as $1/r_j/\sum_{j=1}^{n} C_j$. Therefore, we use a local search (LS) method for obtaining approximate solutions close to the optimal solution in less time. In this research, we used Annealing (SA) method and genetic algorithm (GA).

2- Coding:s

The following coding's are used in this research:

j=Job index.

$\bar{N}$=The set of all n-jobs .

n=Number of jobs .

p_j=Processing time for job-j ,i.e. which means that it has to be processed for a period of length p_j .

d_j=Due date for job j ,i.e. the date when the a tasks or jobs should ideally be completed.

r_j=a release date of job j ,i.e. the earliness time at which the processing of job can begin.

C_j=Completion time of (job-j) .

T_j=The Tardiness of (job-j).

E_j=The Earliness of (job-j) .

U_j=The Unit penalty of (job-j) .

V_j= The Late work of (job-j) , finally $\sum$=The Sum

3- Mathematical Model:

In issue or problem (HH):

- Schedule a group of tasks $\bar{N}$ or (n-jobs), $\bar{N}=\{1,\dots,n\}$ on (1-machine).
- Integer processed time (p_j) for all job (j), (r_j), and (d_j).
- $C_1=r_1+p_1$, and when (j=2,…,n) then $C_j=Max(r_j , C_{j-1})+p_j$.
- $T_j=Max(C_j – d_j, 0)$.
- $E_j=Max(d_j –C_j , 0)$.
- $U_j= 1$, if $C_j> d_j$; o.w, $U_j=0$.
- $V_j=Min(T_j ,p_j)$.
- Let δ be a set of all feasible solutions, and σ ∈ δ , σ =(1,…,n). The mathematical model of issue (HH) can be formulated as follows:

$$(HH)=\text{Min } \delta(\sigma)=\text{Min}_{\sigma\epsilon\delta}\{ \sum_{j=1}^{n}(C_{\sigma(j)} + T_{\sigma(j)} + E_{\sigma(j)} + U_{\sigma(j)} + V_{\sigma(j)}) \}$$

S.t :

$C_{\sigma(1)}=\{r_{\sigma(1)} + p_{\sigma(1)}\}$

$C_{\sigma(j)}=Max\{r_j, C_{j-1}\}+p_j$, $j = 2,\dots, n$

$T_{\sigma(j)}=Max\{C_{\sigma(j)} – d_{\sigma(j)}, 0\}$, $j= 1,\dots,n$

$E_{\sigma(j)}=Max\{d_{\sigma(j)} – C_{\sigma(j)}, 0\}$, $j= 1,\dots, n$

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

$$U_{\sigma(j)}=\begin{cases}1 & \text{if } C_{\sigma(j)} > d_{\sigma(j)} \\ 0 & \text{o.w}\end{cases}, \qquad j = 1, \dots, n$$

$$V_{\sigma(j)}=\text{Min}\{ T_{\sigma(j)}, p_{\sigma(j)}\}, \qquad j = 1,\dots, n$$

The purpose is to Getting a processing order $\sigma=(\sigma(1),\dots,\sigma(n))$ for minimize the issue (HH). In order to simplify the issue (HH), can be Partition into (3) sub issues (hh_1), (hh_2) and (hh_3), as below:

$$(HH)=\text{Min}\ \delta(\sigma)=\text{Min}_{\sigma\epsilon\delta}\{ \textstyle\sum_{j=1}^{n}(C_{\sigma(j)} + T_{\sigma(j)} + E_{\sigma(j)} + U_{\sigma(j)} + V_{\sigma(j)}) \}$$

$$hh_1= \text{Min}_{\sigma\epsilon\delta} \{ \textstyle\sum_{j=1}^{n}(C_{\sigma(j)} + T_{\sigma(j)} + E_{\sigma(j)}) \}$$

$$hh_2 =\text{Min}_{\sigma\epsilon\delta} (\textstyle\sum_{j=1}^{n} U_{\sigma(j)})$$

$$hh_3 =\text{Min}_{\sigma\epsilon\delta} (\textstyle\sum_{j=1}^{n} V_{\sigma(j)})$$

Then we can find values of (HH) since $hh_1+ hh_2+ hh_3 \le HH$ see [6].

4- Six-heuristics techniques to obtain an initial feasible solution:

To obtain an initial feasible solution (σ), in this paper (6) heuristics methods are used for arranging the jobs and calculation the cost issues (HH).

- The first an initial feasible solution is (1-heuristic)= Arrange using the rule [SRT] i.e:($r_1\le\dots\le r_n$), then compute value $\delta(\sigma_1)$.
- (2-heuristic)= Set j_1=Min{$r_j + p_j$}, where (j=2,…,n) order by rule [SRT], then calculate value $\delta(\sigma_2)$.
- (3-heuristic)= By ($r_1+ p_1\le\dots\le r_n+ p_n$), then calculate value $\delta(\sigma_3)$.
- (4-heuristic)= set j_1=Min{$r_j + p_j$}, then Arrange all jobs by rule [SPT], [SPT], i.e. ($p_2\le\dots\le p_n$), then calculate value $\delta(\sigma_4)$.
- (5-heuristic)= By rule [SPT], then find value $\delta(\sigma_5)$.
- (6-heuristic) =By rule [EDD], i.e.($d_1\le\dots\le d_n$), then find value $\delta(\sigma_6)$. Among these (6-heuristic), we select one $\delta(\sigma)$ from: $\delta(\sigma)$ =Min{ $\delta(\sigma_1)$,..., $\delta(\sigma_6)$}).

5- Local Search Algorithms:

The local search algorithms (methods or techniques) are a set of general-uses techniques for optimization problems, each technique determine a different strategy for dealing with the problems. The use of (LS) techniques to optimization problems dates to early 1960s [7]. Since that history the interest in this subject has considerably grown

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

in the fields of computer and operations research sciences, and artificial intelligence. Some of the prominent heuristics' method emulate some of processes that occur in nature (like genetic evolution, metal annealing, and so on), from the common examples SA, GA, Swarm- Intelligence algorithms SIA, and Tabu search TS [8]. These strategies are usually called as "Meta-Heuristics". In this paper we propose two (LS) (meta-heuristics) techniques, to solve our problem (HH), are simulated annealing (SA) and Genetic Algorithm (GA), in sections next, we describe the two algorithms for our problem as follows:

5-1- Simulated Annealing Technique (SA):

The (SA) was displayed by (Kirkpatrick et.al [9]), in the early 1950s. This technique simulates the annealing process in which a material or (metal), this process boils down to raising the temperature of the metals above their melting point and then gradually cooling them to finding an optimal structure. The instrument to crystal formation is carefully controlling the rate of change of temperature [10].

In order not to get too far away from the terminology of improvement problems, so here we simply describe the algorithm for (HH) as follows:

Step-1 : Initialization: To obtain an initial feasible solution s we applied six heuristics methods (see section 4) and compute its objective function value δ(s), then we choose minimum value (i.e. min- objective function value =an initial feasible solution s).

Step-2 : Initial value of parameter (T=50000) and reduction method:

Step -3 : Neighborhood solution s': To get neighborhood solution s' of the an initial feasible solution s by switch the location of any two jobs that adjacent or not be adjacent, and compute its objective function value δ(s')

Step-4 : Find $\Delta\delta = \delta(s') - \delta(s)$

Step -5 : Acceptance or rejection test: If $\Delta\delta \leq 0$, we accept s' (i.e. s':=s), but if $\Delta\delta > 0$ then we accept s' with a certain probability p (where $P = \exp^{-\Delta/T}$).

Step-6 : Stop condition: After (50000) iterations the algorithm stops, or reach a runtime (600) second . If no we reduce T by the following relationship: $T^{new} = T^{old} - 1$, then go step (3).

5-2- Genetic Algorithm (GA):

The (GA) technique is based on Darwin's (1859) theory of the "survival-of-the-fittest" that is, it depends it is based on the principles of genetics and natural chose [10]. Therefore, it is very appropriate to use genetics concepts in this technique. The (GA) is a kind of optimization techniques, for finding near optimal solutions to (NP- hard) issues. The basic rules for these methods were developed by Holland and Goldberg see ([11], [12]) (1988) during their investigations on how to develop computing means and Programs that are capable of simulation or learning [3].

The work of the GA involves a set of procedures that we explain for our problem as follows as:

Step (1): Initialization: Here we start with initial population of size 50 solutions, 6 of them generated by applied six heuristics methods (see section 4), otherwise generated at randomly.

Step (2) : Calculate the value for δ(σ) for each solution

Step (3) : Create a new population: Repetition the following processes (genetic operators) :

a) Selection: Select the best individuals from the current population based on the value of the δ(σ). In this work we selected the best five solutions, and ten solutions randomly.

b) Crossover: This process is applied to the five solutions that we get from the previous procedure, by applying the type one-point crossover (see [13]). From this procedure we get 40 solutions, besides the ten random solutions we have got 50 solutions.

c) Mutation: The mutation procedure applies to all solutions obtained from the two procedures above.

Step(4): Stop Criteria: The (GA) Stops when (50) iterations, or reach a runtime (600) second .

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

6- Computational Experience:

The algorithms of this paper (SA) and (GA) are coding in (MATLAB-2018) and Programmed on a computer -dell-Core- i7- with ram-32- (G-B)], CPU= 3.4, hard 1 T.

We created (5) issues randomly, for each issue, nϵ{5,…, 30000} jobs, that were generated as follows:

- The (p_j), (dj) and (rj) are generated for each j from the uniform distribution.
- For (pj): [1,10]
- For (dj): [1, (M)(1-F+R/2)], where M= $\sum_{j=1}^{n} p_j$, where (F) is the [Tardiness factor], and (R) is the [Relative-range of the due-dates]. For the (F) and (R), the values 0.2 , 0.4 , 0.6 ,0.8 and 1are considered.
- For (r_j): [1, 5].

Table (1) shows a comparison of the performance of (SA) and (GA) when the value of $n \le 30000$ jobs. We also have other table (2) that show the average objective function values, and average run time for (SA) and (GA).

Table 1 : Results comparison between SA and GA for n =5, 10, 16, 100, 500, 1000, 5000, 10000, 20000, 25000, 30000

n	**EX**	**SA**	**GA**	**T. (SA)**	**T. (GA)**
5	1	163	163	0.430	0.083
	2	160	160	0.356	0.035
	3	207	207	0.395	0.028
	4	93	93	0.359	0.028
	5	87	87	0.360	0.028
10	1	540	540	0.444	0.032
	2	377	377	0.400	0.033
	3	525	525	0.385	0.031
	4	325	325	0.381	0.031
	5	402	411	0.405	0.030
16	1	1158	1150	0.385	0.032
	2	1241	1193	0.385	0.034
	3	929	918	0.391	0.032
	4	1242	1229	0.383	0.032
	5	817	789	0.384	0.032

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

100	1	35828	33778	0.578	0.062
	2	38367	36005	0.539	0.061
	3	34963	30548	0.553	0.061
	4	32192	31514	0.560	0.060
	5	35217	31724	0.544	0.062

n	EX	SA	GA	T. (SA)	T. (GA)
5000	1	85754746	85607954	8.368	20.602
	2	95144242	95125826	8.226	16.733
	3	90706288	90686246	8.234	16.742
	4	84949249	84736126	8.404	16.593
	5	87663692	87570412	8.310	16.757
10000	1	345580067	345281307	16.397	33.209
	2	345661911	336233346	16.896	33.165
	3	352995188	293414015	17.236	33.280
	4	340350267	335216852	16.603	33.764
	5	348832073	348567431	16.500	32.973
20000	1	1413470856	1413205015	31.819	65.282
	2	1372944309	1353026554	32.948	65.834
	3	1364641922	1353716972	32.563	65.892
	4	1365073694	1346601732	32.795	65.871
	5	1455644779	1455580324	31.044	65.576
30000	1	2990252315	2783039410	66.616	132.482
	2	3176092043	3175597477	62.579	144.147
	3	2971959154	2759356015	69.002	142.914
	4	3266049658	3265912043	59.152	109.222
	5	3191408496	2992449292	51.532	109.340

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

n: n-jobs.

EX: Example number.

SA: The Simulated Annealing (SA) method.

GA: The Genetic algorithm (GA) method.

T. (SA): The time which is required for (SA) (in seconds).

T. (GA): The time which is required for (GA) (in seconds).

Table 2 : Compare the average value and time for SA and GA.

n	Av.SA	Av.GA	Av.T. (SA)	Av.T. (GA)
5	142	142	0.38	0.0404
10	434	436	0.403	0.0314
16	1077	1056	0.3856	0.0324
100	35313	32714	0.5548	0.0612
5000	88843643	88745313	8.3084	17.4854
10000	346683901	331742590	16.7264	33.2782
20000	1394355112	1384426119	32.2338	65.691
30000	3119152333	2995270847	61.7762	127.621

In the tables (2) we have:

n: n-jobs.

Av.SA: The average objective function values of (SA) method.

Av.GA: The average objective function values of (GA) method.

Av. T.(SA): The average run time for (SA).

Av. T.(GA): The average run time for (GA).

7. Conclusions and suggestions:

1. From the results obtained for using of two methods (SA and GA) on issue (HH), The performance of two methods can be tested on large class of test issues. The computational results showed a superior ability to solve problems, as up to (30,000) functions were solved in a record period.

2. GA is better in n=5 to less than 5000 jobs, while SA best in n=5000 ,..,30000 jobs.

3. To make the performance of the (SA and GA) methods better, we propose making a hybrid between the two methods search or between them and another (LS) methods,

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024

References

[1] Chen, B., Chris N. P, and Gerhard, J. W. "A review of machine scheduling: Complexity, algorithms and Approximability," Handbook of combinatorial optimization. Springer, Boston, MA, (1998). 1493-1641.

[2] Michael, L. P. " Scheduling: theory, algorithms, and systems". Fifth Edition, (2016).

[3] Walser, J. P. "Integer optimization by local search: a domain-independent approach". Springer-Verlag, (1999).

[4] Nagar A., Haddock J. and Heragu S.,(1995). Multiple and Bi-Criteria Scheduling a Literature Survey, Eur. J.OPI. Res. 1:88-104.

[5] HA Chachan, HA Jaafar. "Exact Solutions for Minimizing cost Function with Five Criteria and Release Dates on Single Machine". Ibn AL-Haitham Journal For Pure and Applied Science, 2020. Sci. 33 (3).

[6] Chu, C, and Portmannm M.C. "Some new efficient methods to solve the n/1/ri/∑Ti scheduling problem", European journal of operational research 58.3 (1992): 404-413.

[7] Dunham, B., Fridshal, D., Fridshal, R., & North, J. H. Design by natural selection. In Form and Strategy in Science. (1964). 306-311.

[8] Sörensen, K., and Fred, W. G. "Metaheuristics." Encyclopedia of operations research and management science (2013): 960-970.

[9] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M.P. "Optimization by simulated annealing." science 220.4598 (1983): 671-680.

[10] Haupt, R. L., and Sue, E. H. "Practical genetic algorithms." .SECOND EDITION, (2004).

[11] Holland, J. "Adaptation in natural and artificial systems: an introductory analysis with application to biology." Control and artificial intelligence (1975).

[12] Goldberg, D. E., and Holland, J.H. "Genetic algorithms and machine learning." Machine learning 3.2 (1988): 95-99.

[13] Baúto, J., Neves,R., and Horta, N. "Parallel Genetic Algorithms for Financial Pattern Discovery Using GPUs". Springer International Publishing, (2018).

Doi: https://doi.org/10.54216/GJMSA.0100204
Received: October 13, 2023 Revised: January 19, 2024 Accepted: March 24, 2024