1. ABSTRACT

Various algorithms have been used for solving combinatorial optimization problems. Variable Neighborhood Search (VNS) and Particle Swarm Optimization (PSO) are two algorithms that have been able to effectively solve these problems, weren't able to produce optimum results [3, 4]. This failure was an account of not considering the effect of some constraints on optimality.

In this study, the effects of make-span and flow time constraints on optimality have been taken into account to solve various sized combinatorial optimization problems. The results show that VNS produces optimum results and gives 0.75% average improvement over PSO.

2. INTRODUCTION

2.1 Variable Neighborhood Search Algorithm

Variable Neighborhood Search (VNS) was discovered by Mladenovic, Hansen in 1972. It is based on an iterative process of finding the solutions using the local search method for combinatorial optimization problems.It is a meta-heuristic framework based model. Local Search Method, a feature in VNS overcomes the drawback of heuristic problems, where optimum results are not quarantined.

Decisions and Variables

A solution $s^* \in S$ can be considered as optimal if

$$P(x^*) \leq P(x), \forall x \in X$$

Where, min f(x) is a global or combinatorial problem

Initialized terms and associated values

For a group containing of ‘size’ number of neighborhood structures in a given problem:

I.e., N_k (k= 1 … k_{size})

The initialization process is carried out by giving positions to all the neighborhood particles in a random manner (either choosing arbitrarily or using functions to generate a randomized order within in a specific range).

Procedure for optimization

The following iterations after the initialization are carried out in a similar fashion unless the stopping condition is achieved. The stopping condition depends on the type of the problem and also on the requirement of the problem.

2.2 Particle Swarm Optimization

Fascinated by the social behavior of a flock of birds and fishes, that was studied by Craig Reynolds (a Biologist) in the late 80’s and 90’s, Russel Eberhart (Electrical Engineer) and James Kennedy (Social Psychologist) developed this theory in 1995 (both U. Indiana, Purdue). PSO is an iterative, evolutionary algorithm, where the solution is optimized gradually based on the movement and values associated with the swarms.

Description of Swarms

After the initialization of the attributes related to swarms based on the problem requirement, the evolution of these attributes begins. This evolution is carried out on the basis of two solutions of every particle and iteration.

- P-best (personal best)
- G-best (global best)

Initialization of particles

Depending upon the size and complexity of the problem, the number of swarms is to be decided and they are instantiated with a position and velocity within a solution space in a calculative randomized manner.

I.e., X_{ij} and V_{ij} for the swarms are initialized for iteration 0

Where, 'i' denotes iteration number

'j' denotes particle/swarm number in i^{th} iteration

Formulae used during the optimization

After finding the two best values: personal best and global best, the particle updates its positions with the following equations (1) and (2).

I.e., $X_{ij}^{t} = V_{ij}^{t} + X_{ij}^{t-1}$

Where, V_{ij}^{t} represents the velocity of particle i at iteration t with respect to j^{th} dimension (j=1,2 …n)

X_{ij}^{t} is the position value of the i^{th} particle with respect to j^{th} dimension.

X_{ij}^{t-1} is the previous position value

3. BODY

3.1 Analysis of different constraints involved and extent of usage

There are several constraints on jobs and machines. One of the constraints is to operate on various jobs simultaneously. Here, simultaneous refers to performing operations on various jobs at a time and the job itself. Other constraints that the machine should always be available until the task is completed, plays a vital role in determining the usage of the machines by the jobs. Constraints for prioritizing different jobs are required for giving preferences for certain jobs over the others. For any kind of algorithm being used for performing operations on the jobs, it is necessary that no operation can be disrupted before completing the task.

3.2Application of algorithms on same problems

Effect of Optimization based on constraints

The following constraints are the two key points to optimality:

1. The maximum completion time (make span): C_{max}
2. The sum of the completion times (flow time): C_{sum}

Minimizing C_{sum} asks the average job finishes quickly, at the expense of the largest job taking a long time, whereas minimizing C_{max}, asks that no job takes too long, at the expense of most jobs taking a long time. Minimization of C_{max} would result in maximization of C_{sum}[2].

Number of iterations

In PSO, the number of parameters to be adjusted plays a key role in attaining the optimum solution within a minimum number of iterations. While comparing PSO and VNS, number of iterations is not a criteria for attaining the optimum values: because there wouldn't be any significant difference between them. Hence, PSO loses the edge over VNS.

Procedure for Application of VNS

Step 1: All the neighborhood particles are given positions in a random manner (either choosing arbitrarily or using functions to generate a randomized order within in a specific range)

I.e., $x_1 \in N_k(x)$ is generated randomly

Step 2: Local optimal values are computed using the local search method staring from the first particle to the 'size' values.

Step 3: If the values of x of a neighbor are better than that of its previous iteration, then it remains the same. On the contrary, if the value of the incumbent is better, then the value of the incumbent is taken into consideration.

Application to a problem

For the analysis of a two algorithms, a job-shop scheduling problem is considered and a modified version of the algorithms are considered in order to produce the best possible results from [4]. The application was evaluated for every instance, until an optimum solution was attained and the results are as follows.

Instance	Size	Opt	Z_{NS}	JSP-PSO$_{VNS}$			JSP-VNS			JSP-PSO		
	m x n	(UB)		Z	Avg	Time	Z	Avg	Time	Z	Avg	Time
la01	5x10	666	666*	666*	670	18	666*	666	0.3	666*	671	1
la02	5x10	655	655*	655*	656	17	655*	655	2	704	734	1
la03	5x10	597	597*	597*	603	81	597*	602	35	630	664	1
la04	5x10	590	590*	590*	597	54	590*	593	23	619	641	2
la05	5x10	593	593*	593*	593	0.4	593*	593	0.3	593*	593	0.2
la06	5x15	926	926*	926*	926	1	926*	926	1	926*	930	2
la07	5x15	890	890*	890*	891	51	890*	890	1	922	957	4
la08	5x15	863	863*	863*	863	1	863*	863	1	884	895	4
la09	5x15	951	951*	951*	951	1	951*	951	1	951*	971	3
la10	5x15	958	958*	958*	958	2	958*	958	1	958*	958	1
la11	5x20	1222	1222*	1222*	1222	4	1222*	1222	4	1222*	1233	8
la12	5x20	1039	1039*	1039*	1039	4	1039*	1039	4	1039*	1050	5
la13	5x20	1150	1150*	1150*	1150	4	1150*	1150	4	1150*	1155	5
la14	5x20	1292	1292*	1292*	1292	6	1292*	1292	4	1292*	1292	1
la15	5x20	1207	1207*	1207*	1207	3	1207*	1207	3	1305	1332	9
la16	10x10	945	945*	945*	945	294	945*	946	461	1047	1065	7
la17	10x10	784	784*	784*	784	125	784*	784	35	865	884	6
la18	10x10	848	848*	848*	854	196	848*	854	228	888	947	7
la19	10x10	842	842*	842*	849	563	842*	846	286	958	984	9
la20	10x10	902	902*	902*	905	421	902*	905	320	995	1053	7

Table 1: Results from first 20 instances [4]

Instance	Size	Opt	Z_{NS}	JSP-PSO_{VNS}			JSP-VNS			JSP-PSO		
	$m \times n$	(UB)		Z	Avg	Time	Z	Avg	Time	Z	Avg	Time
la21	10x15	1046	1047	1046*	1052	3253	1047	1057	2749	1293	1308	27
la22	10x15	927	927*	927*	941	3068	927*	928	2037	1102	1169	26
la23	10x15	1032	1032*	1032*	1032	24	1032*	1032	10	1210	1232	28
la24	10x15	935	939	935*	942	3047	937	941	3558	1129	1156	26
la25	10x15	977	977*	984	988	2592	977*	981	2717	1190	1225	32
la26	10x20	1218	1218*	1218*	1218	109	1218*	1218	63	1453	1517	88
la27	10x20	1235	1236	1240	1259	7695	1235*	1252	5179	1556	1623	75
la28	10x20	1216	1216*	1216*	1216	4791	1216*	1216	1471	1443	1518	97
la29	10x20	1152	1160	1163	1174	11730	1163	1171	11100	1465	1520	70
la30	10x20	1355	1355*	1355*	1355	28	1355*	1355	39	1566	1630	77
la31	10x30	1784	1784*	1784*	1784	153	1784*	1784	125	1987	2089	264
la32	10x30	1850	1850*	1850*	1850	142	1850*	1850	117	2143	2174	304
la33	10x30	1719	1719*	1719*	1719	146	1719*	1719	121	1919	2017	309
la34	10x30	1721	1721*	1721*	1721	139	1721*	1721	115	2022	2066	260
la35	10x30	1888	1888*	1888*	1888	160	1888*	1888	124	2152	2258	237
la36	15x15	1268	1268*	1268*	1271	13928	1268*	1269	8520	1560	1612	86
la37	15x15	1397	1407	1397*	1410	8785	1397*	1407	7084	1670	1742	108
la38	15x15	1196	1196*	1201	1205	14155	1201	1207	9599	1495	1551	99
la39	15x15	1233	1233*	1233*	1236	9737	1233*	1237	7802	1543	1619	83
la40	15x15	1222	1229	1224	1226	11678	1224	1227	12398	1540	1576	119
ta11	15x20	(1364)	—	1386	1396	41628	1380	1394	28701	1826	1863	223
ta12	15x20	(1367)	1377	1377	1379	32317	1377	1385	31288	1814	1899	196

Table 2: Results from iterations 21 [4]

A star mark indicates that the value was attained as an optimum solution.

Contradictory case

When the size of the problem is small, PSO can reach the optimal value in lesser number of iterations when compared to VNS. On the other hand, VNS attains a better optimal solution in more number of iterations. An example of such case is evident from the below example.

Iteration	New Sequence	System Unbalance	Through put	Objective function value	Unassigned jobs
1	4,6,7,3,1,2,5,8	141	41	1.439	1,2,5,8
2	4,7,2,3,5,6,1,8	14	48	1.5927	2,6,8
3	4,7,2,5,3,6,1,8	14	48	1.5927	2,6,8
4	2,5,4,7,8,1,3,6	0	36	1.45	8,3,6
5	8,5,1,2,6,3,7,4	185	40	1.403	2,3,7,4
6	8,1,6,3,5,2,7,4	127	44	1.483	5,2,7,4
7	1,6,8,3,5,7,2,4	127	44	1.483	5,7,2,4
8	6,3,4,1,7,8,2,5	18	46	1.565	7,8,5
9	4,7,2,3,6,5,1,8	14	48	1.592	2,6,8
10	4,7,2,5,3,6,1,8	14	48	1.592	2,6,8
11	4,2,5,7,8,3,1,6	0	36	1.45	8,3,1,6
12	8,5,2,1,3,6,7,4	144	43	1.4625	2,6,7,4
13	8,1,6,5,3,2,7,4	185	40	1.403	3,2,7,4
14	8,1,6,3,5,2,7,4	127	44	1.483	5,2,7,4
15	6,1,3,8,7,5,4,2	127	44	1.483	7,5,4,2
16	4,7,6,3,1,2,5,8	14	48	1.5927	6,2,8
17	4,7,2,3,5,6,1,8	14	48	1.5927	2,6,8
18	4,7,2,5,3,6,1,8	14	48	1.5927	2,6,8
19	4,2,5,7,8,3,1,6	0	36	1.45	8,3,1,6
20	5,2,8,1,7,3,4,6	130	31	1.319	1,7,3,4,6

Table 3: Example to a contrary case

Evidence of Optimization

Flexible Manufacturing Systems consists of a number of pre-release and post-release decisions. Machine loading problem is one of the pre-release decisions, which is a combinatorial optimization problem. Particle Swarm Optimization based on the previous constraints and analysis is applied to it and optimum results are produced [5].

4.RESULTS

Problem type	Δ_{NS} (%)	JSP-PSO$_{VNS}$		JSP-VNS		JSP-PSO	
		Δ (%)	Time	Δ (%)	Time	Δ (%)	Time
5 × 10	0.00	0.00	34	0.00	12	3.58	1
5 × 15	0.00	0.00	11	0.00	1	1.21	3
5 × 20	0.00	0.00	4	0.00	4	1.62	6
10 × 10	0.00	0.00	320	0.00	266	9.99	7
10 × 15	0.10	0.14	2397	0.06	2214	20.46	28
10 × 20	0.16	0.27	4871	0.19	3571	21.34	81
10 × 30	0.00	0.00	148	0.00	120	14.06	275
15 × 15	0.26	0.12	11657	0.12	9081	23.75	99
15 × 20	—	1.17	36973	0.95	29995	33.29	210

Table 4: Summary from results of Table 2

In the above illustration, percentage (%) of values are computed for average deviation between the upper bound and the best solution value found by the algorithms.After considering the constraints, it is observed that VNS and PSO have been able to produce efficient results. Also, the results produced by VNS are better than PSO.

5. CONCLUSION

According to my study, when the size of the problem is big (problem having 10 or more number of jobs), VNS produces better results than PSO in a similar number of iterations.

On the other hand, when the size of the problem is small (problem having less than 10 number of jobs), PSO reaches its optimum value in lesser number of iterations. But, the optimum value reached by PSO is not better than that of VNS. So, even when the problem size is small, VNS produces better results.

It is observed that VNS on an average produces 0.75% improvement over PSO.

6. REFERENCES

[1] Thomas Stutzle (2003), Iterated Local Search Variable Neighborhood Search Thomas [online]. Available: http://www.sls-book.net/Slides/sls-ils+vns.pdf.

[2] Hongbo Liu et al (2006), Variable Neighborhood Particle Swarm Optimization. 6th International Conference, SEAL 2006, Hefei, China, October 15-18, 2006. Proceedings. pp 197-204. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4444189

[3] Kennedy, J., Eberhart, R. (1995),"Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968

[4] Pisut Pongchairerks and Voratas Kachitvichyanukul (2007), "A Comparison between Algorithms VNS with PSO and VNS without PSO for Job-Shop Scheduling Problems", International Journal of Computational Science, Vol. 1, No. 2, 179-191.

[5] A. Somaiah, M. Indira Rani and Ch. Varun Kumar (2014), "Particle Swarm Optimization for Machine Loading Problem in F.M.S", Indian Journal of Applied Research, Vol. 4, Issue 8, ISSN 2249-555X.
Available:http://www.worldwidejournals.com/ijar/file.php?val=August_2014_1408353536__66.pdf