COMPARISON OF MULTI ITEM CAPACITATED LOT SIZING HEURISTICS

Project submitted in partial fulfillment of the requirements for the degree of Master of Science in Industrial Engineering

Under the guidance of:

Dr. Chase Murray

Date: 05/19/2017

Aditya Patankar 50204875

Ramandeep Singh Makhija 50204307

1. Introduction.

The multi-item single level capacitated dynamic lot sizing problem consists of scheduling a variety of N different items over the planning horizon of T time periods; S.Parveen, A.F.M.Haque, 2007. According to B. Karimia, S.M.T. Fatemi Ghomia, J.M. Wilsonb; 2003, lot sizing is considered to be one of the most important and also one of the most difficult problems in production planning. The complexity of this problem depends on the number of features taken into account by this model. The features which are taken into account are the planning horizon, number of levels, number of products, capacity or resource constraints, deterioration of the items, demand, inventory shortage and the setup structure.

The main objective when considering the multi-item single level lot sizing problem is to minimize the combined inventory holding and setup costs over the planning horizon subject to the capacity constraints in each planning period. Hence the amount and the timing of the production of products during the planning horizon needs to be determined to achieve this objective. According to research presented by Chen and Thizy, the multi-item CLSP is strongly NP-hard. According to Maes, even finding a feasible solution to these problems is considered NP-hard. Hence it is unlikely that any effective optimal algorithm can be developed to solve this problem. Hence based on the literature, developing effective heuristics has been a profitable area of research. According to B. Karimia, S.M.T. Fatemi Ghomia, J.M. Wilsonb; 2003, the heuristics are classified according to the Table 1 shown below. As shown in the Table 1, provided by B. Karimia, S.M.T. Fatemi Ghomia, J.M. Wilsonb; 2003 the heuristics considered in our paper i.e the Lambrecht and Vanderveken; and Dixon and Silver are categorized under the Common-sense or specialized heuristics.

Table 1: Characteristics of Specialized Heuristics

Lot Shifting	Feasibility Check	Priority Indices	Initial Solution	Algorithm
Left Shifting	Feedback	Silver- Meal	NA	Lambrecht and Vanderveken
NA	NA	PPB	NA	EisenHut
Left Shifting	Look- ahead	Silver-Meal	NA	Dixon and Silver

2. Summary:

The main objective of this project is to analyze the research work done with respect to the capacitated lot sizing problem and code the heuristics using Python. The heuristics which are considered for further study are:

1. Extended Dixon- Silver Heuristic as presented in Parveen, Sultana, and AFM Anwarul Haque. "A heuristic solution of multi-item single level capacitated dynamic lot-sizing problem." *Journal of Mechanical Engineering* 38 (2007): 1-7.
2. Extended Eisenhut Heuristic as presented in Lambrecht MR, Vanderveken H. “Heuristic procedures for the single operation, multi-item loading problem”. AIIE Transactions 1979;15(4):319–25.

A common data set has been selected for the evaluation of both the heuristics. Analyzing the results obtained from both the heuristics using a common data set helps us identify the features of heuristic and also identify the possible advantages or flaws in each heuristic. A detailed comparison of the findings and a concluding remarks are provided at the end of the report.

Deliverables:

- Python code: The python code for Extended Dixon-Silver heuristic and the Extended Eisenhut heuristic has been attached along with this report. The readme.txt file attached along with this code gives detailed about how to run the code. A pseudo-code explaining the heuristics has been described in the report.
- Input Data: The input data selected to run the two python scripts has been obtained from Dixon PS, Silver EA. “A heuristic solution procedure for the multi-item, single level, limited capacity, lot sizing problem”. Journal of Operations Management 1981;21(1):23–40. Also in order to further analyze the heuristics, a random manually generated data set has been used. These data sets are attached along with this report in separate csv files. A detailed description of the data set has been provided in later sections of the report.

- Results: The results obtained from the heuristic i.e the lot sizes for each item over each planning horizon are saved in separate csv files. A detailed analysis and interpretation of the results has been provided in the later sections of the report.
- Animation: An animation file showing the findings of the heuristics has been attached and details about it explained in later sections of the report.
- YouTube video: A video summarizing the key findings and addressing all the sections of the project has been created and uploaded on YouTube. The link to the video has been attached. The link to the YouTube videos is as follows:

  https://www.youtube.com/watch?v=Td4PQ7GCJqI&lc=z13dg5vaxo2hy1exs04cclmojpvpidnw1s40k

3. Deliverables:

3.1 Pseudo Code for Extended Dixon-Silver heuristic:

```
1.  INPUT the holding and setup cost, maximum lot size and production rate for all items
2.  INPUT the forecasted demand.
3.  CREATE a pandas DataFrame from Forecasted demand
4.  INITIALIZE period = 1 : j ε {1,12}
    INITIALIZE (Remaining inventory) = (Initial inventory) – (Safety Stock)
    IF (Remaining Inventory)_i > (Demand)_ij
        SET (Equivalent Demand)_ij = 0
    ELSE
        SET (Equivalent Demand)_ij = (Demand)_ij – (Remaining Inventory)_i
    COMPUTE (Remaining Inventory)_I = (Remaining Inventory)_I - (Demand)_ij
    INCREMENT J BY ONE
    RECYCLE till (Remaining Inventory)_I >0
5.  FOR EVERY VALUE IN PERIOD:
        SUM (TOTAL DEMAND FOR ALL ITEMS) <= SUM (TOTAL CAPACITY FOR ALL ITEMS)
6.  SET (Maximum Demand)_i=   Maximum demand for all items
7.  CALCULATE n_i = (Maximum demand)_I / (Maximum lot-size)_I -1
8.  CALCULATE N` = N + SUM(n_i) : item i is split into n_i + 1 items.
9.  INITIALIZE (Remaining demand)_ij = (Demand)_ij and SET l = 0
10. IF (Remaining demand)_ij   <= (Maximum lot-size)_i
        SET (Demand)_ij = (Remaining demand)_ij
        COMPUTE (Remaining Demand)_ij = 0
   ELSE
        SET (Demand)_ij = (Maximum lot-size)_i
        COMPUTE (Remaining Demand)_ij = (Remaining Demand)_ij - (Maximum lot-size)_I
    INCREMENT  l by one.
    RECYCLE till l = n_i
    INITIALIZE Updated Setup and holding costs.
11. INITIALIZE PERIOD R =1
12. INITIALIZE (Lot-size)_ij = (Equivalent Demand)_ij
   CALCULATE (Remaining Lot-size)_ij = (Maximum Lot-size)_i- (Lot-size)_ij
13. INITIALIZE T_i = 1
14. CALCULATE (Remaining Capacity)_R = (Capacity)_R – SUM ((Production rate)_i* (Equivalent demand)_IR)
15. INITIALIZE I'_ij = 0.
16. CALCULATE AP_j = SUM(K_i * (I'_{i,j-1} - I'_ij )
17. CALCULATE (Total demand)_j = SUM(K_i * (Equivalent demand_ij)
18. IF  SUM(AP)_j < SUM((Total demand)_j - (capacity)_j)
        SET t= Period j
        SET t_c = Minimum (t)
   ELSE
        SET t_c = H+1
```

19. CALCULATE $x_{can} = \min((Equivalent\ demand)_{i,R+T_i} - (x_{rem})_{i'R})$

IF $T_{i'} < t_c$ and $x_{can} > 0$ $and\ ((Remaining\ capacity)_R - x_{can}) > 0$

CALCULATE $U_i = \frac{AC(T_i) - AC(T_i+1)}{k_i(equivalent\ demand)_i}$

CALCULATE $(AC)_{T_i} = \frac{S_i + h_i\,Sum(j-R)*(equivalent\ demand)_{ij}}{T_i}$

20. IF $U_i > 0$

CALCULATE $x_{iR} = x_{iR} + x_{can}$

CALCULATE $I'_{ij} = I'_{ij} + x_{can}$

CALCULATE $xrem_{i,R+T_i} = xrem_{i,R+T_i} + x_{can}$

CALCULATE $xrem_{i,R} = xrem_{i,R} - x_{can}$

CALCULATE $x_{iR+T_i} = x_{iR+T_i} - x_{can}$

CALCULATE $(Equivalent\ demand)_{iR+T_i} = (Equivalent\ demand)_{iR+T_i} - x_{can}$

CALCULATE $Remaining\ capacity)_R = (Remaining\ capacity\ )_R - x_{can}$

SET $T_i = T_i + 1$

ELSE

IF (t_c > H)

GOTO STEP 27

ELSE

GOTO STEP 21

21. CALCULATE amount of production needed $Q = MAX\ (SUM((Total\ demand)_j - (capacity)_j - (AP)_j))$

22. IF $T_{i'} < t_c$ and $x_{can} > 0$ $and\ ((Remaining\ capacity)_R - x_{can}) > 0$

CALCULATE $\Delta_{i'} = \frac{AC(T_i+1) - AC(T_{i'})}{k_{i'}(equivalent\ demand)_{i'T_i+1}}$

23. COMPUTE MIN $\Delta_{i'}$

24. SET W = $k_i * x_{can}$

IF Q> W

CALCULATE $x_{iR} = x_{iR} + x_{can}$

CALCULATE $I'_{ij} = I'_{ij} + x_{can}$

CALCULATE $xrem_{i,R+T_i} = xrem_{i,R+T_i} + x_{can}$

CALCULATE $xrem_{i,R} = xrem_{i,R} - x_{can}$

CALCULATE $x_{iR+T_i} = x_{iR+T_i} - x_{can}$

CALCULATE $(Equivalent\ demand)_{iR+T_i} = (Equivalent\ demand)_{iR+T_i} - x_{can}$

CALCULATE $Remaining\ capacity)_R = (Remaining\ capacity\ )_R - x_{can}$

INCREMENT Q BY ONE

SET $T_i = T_i + 1$

ELSE

SET IQ = $\frac{Q}{K_i}$

FOR PERIOD = R +1 TO R + T_i

$x_{iR} = x_{iR} + IQ$

$I'_{ij} = I'_{ij} + IQ$

$xrem_{i,R+T_i} = xrem_{i,R+T_i} + IQ$

$x_{iR+T_i} = x_{iR+T_i} - IQ$

$(Equivalent\ demand)_{iR+T_i} = (Equivalent\ demand)_{iR+T_i} - IQ$

$xrem_{i,R} = xrem_{i,R} - IQ$

25. INCREMENT R BY ONE.

26. IF R < H

GOTO STEP 13

ELSEIF R> H

PRINT “HEURISTIC COMPLETE”

27. $(Lotsize)_{ij} = SUM\ ((Lotsize)_{i_i,j}$

28. CALCULATE Inventory, setup and total cost.

3.2 Pseudo Code for Extended Eisenhut Heuristic:

```
1.  GENERATE Equivalent Requirements Matrix d_it
2.  B_t = SUM(SUM(demand_ik − capacity_k))
3.  IF B_k > 0
        Order upto period: T* = k
    ELSE
        T* = H + 1
4.  COMPUTE Inventory_i(t) = h_i * SUM(( k − 1) * demand_ik))
5.  IF SUM (capacity_k) ≥ SUM (demand_ik)
        COMPUTE Coefficient U_i(t) = (s_i − Inventory_i(t)) / (t^2 (demand)_it)
    ELSE
        GOTO STEP 8
6.  SELECT i with U = MAX(U_i(t))
7.  IF U_i(t)> 0 and demand_it ≤ remaining capacity_t
        INCREMENT x_it−1 by d_it
        DECREMENT remaining capacity_t and d_it by d_it
    ELSE
        DELETE MAX((U_i(t))
        REPEAT STEP 6
8.  CALCULATE INF_t = SUM(demand_it − capacity_t)
    SET a = 1
    IF d_it−a != 0
        APPEND i to M*
9.  COMPUTE C_i = min(demand_it, INF_t)
10. COMPUTE MIN (a * C_i * h_i)
11. SELECT q corresponding to minimum STEP 10
12. COMPUTE E = MIN(remaining capacity_t−a, C_q)
13. INCREMENT x_qt−a with E units.
    DECREMENT d_qt, remaining capacity_t−a , INF_t by E units
14. IF INF_t > 0
        IF E = remaining capacity_t−a
                SET a = a + 1
                GOTO STEP 1
14.1    ELSE
                PROCESS All conditions given in Table 2
                SELECT x_it with minimum additional cost.
                UPDATE Transfer units for period t-a
                IF t = feasible
                        GOTO SETEP 14.1
                ELSE
                        SET a =a + 1
                        GOTO STEP 8
15. CALCULATE Inventory_i(t) = h_i * SUM(( k − 1) * demand_it))
16. CALCULATE Inventory cost, setup cost, and total cost.
```

Table 2: Table for Step 14.1 of Extended Eisenhut heuristic

	DETERMINE	IF	TRANSFER	ADDITIONAL COSTS
GROUP 1 $d_{it} < INF_t$	$E_i = \min\{demand_{it}, remaining\ capacity_{\ t-a}\}$	• $E_i = demand_{it}$ • E_i =remaining capacity$_{t-a}$	$demand_{it}$ remaining capacity$_{t-a}$	$a * demand_{it} * h_i$ $a *$ remaining capacity$_{t-a} * h_i + s_i$
GROUP 2 $d_{it} > INF_t$	$B_i = \min\{INF_t, remainingcapacity_{\ t-a}\}$	• $B_i = remainingcapacity_{\ t-a}$ • $B_i = INF_t$ • $demand_{it} < remaining\ capacity_{\ t-a}$	remaining capacity$_{t-a}$ INF_t $demand_{it}$	$a *$ remaining capacity$_{t-a} * h_i + s_i$ $a * INF_t * h_i + s_i$ $a * demand_{it} * h_i$

3.3 Input Data:

- The input data selected for this project has been obtained from Dixon, Paul S., and Edward A. Silver. "A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem." *Journal of operations management* 2.1 (1981): 23-39.
- The tables 2 and 3 from this paper have been selected as Data 1 and tables 6 and 7 as Data 2.
- These data sources have been shown in our report in the Tables 3,4,5 and 6 respectively.
- These data tables have been saved in a csv file which has been uploaded along with this report.
- The data from Table 3, provides information about various factors like standard cost, marginal factor, carriage cost, setup cost, production rate, lead time, safety stock, free balance inventory, ending inventory and the inventory plan.
- This information has been provided for all the 'n' items which in case of Table 3 is 12.
- The Table 4 provides information about the forecasted demand for n items over m time periods.
- Tables 5 and 6 provide the same information as tables 3 and 4 respectively and act as Data 2.

Table 3: Relevant product Data for hypothetical machine 1

Item No (i)	**Holding cost (hi)**	**Setup Cost (Si)**	**Maximum lot size (Xmaxi)**	**Production Rate (1/ki)**	**Safety Stock (SSi)**	**Initial Inventory (Iin)**	**Ending Inventory (Iend)**
1	0.0167	322	6000	524	0	19320	18893
2	0.0167	81	60000	349	10602	200180	124225
3	0.0167	124	68000	245	4577	24460	43294
4	0.0167	124	29000	172	1974	23260	21757
5	0.0167	81	49000	349	7581	55489	92168
6	0.0167	124	68000	245	4861	-2727	44394
7	0.0167	124	44000	172	2026	9659	8466
8	0.0167	105	41000	847	11117	29705	40273
9	0.0167	105	32000	464	9533	11362	84717
10	0.0167	106	185000	575	20417	242944	227344
11	0.0167	105	150000	1261	16634	324215	271627
12	0.0167	105	97000	663	9794	45439	69068

Table 4: Forecasted demand and available machine hours for hypothetical machine 1

Item No	**Period**											
(i)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	11456	11456	10501	13365	13365	11456	8592	1909	1909	1909	4773	4773
2	53124	53124	48697	61977	61977	53124	39842	8854	8854	8854	22135	22135
3	18099	18099	16591	21116	21116	18099	13574	3016	3016	3016	7541	7541
4	9250	9250	8480	10792	10792	9250	6938	1542	1542	1542	3854	3854
5	39546	39546	36250	46137	46137	39546	29659	6591	6591	6591	16478	16478
6	18363	18363	16833	21423	21423	18363	13772	3060	3060	3060	7651	7651
7	4976	4976	4562	5806	5806	4976	3732	829	829	829	2074	2074
8	41690	41690	38216	48638	48638	41690	31267	6948	6948	6948	17371	17371
9	32816	32816	30081	38285	38285	32816	24612	5469	5469	5469	13673	13673
10	96745	96745	88683	112868	112868	96745	72559	16124	16124	16124	40310	40310
11	119220	119220	109285	139088	139088	119220	89415	19870	19870	19870	49675	49675
12	27715	27715	25405	32333	32333	27715	20786	4619	4619	4619	11548	11548

Table 5: Relevant product Data for hypothetical machine 2

Item No (i)	**Holding cost (hi)**	**Setup Cost (Si)**	**Maximum lot size (Xmaxi)**	**Production Rate (1/ki)**	**Safety Stock (SSi)**	**Initial Inventory (Iin)**	**Ending Inventory (Iend)**
1	0.0167	186	6000	187	9343	43671	98945
2	0.0167	320	60000	111	3106	26280	18551
3	0.0167	640	68000	73	823	13480	7463
4	0.0167	640	29000	50	1156	15980	16309
5	0.0167	186	49000	222	1579	27480	24156
6	0.0167	320	68000	137	417	0	12668
7	0.0167	640	44000	101	0	6000	1311
8	0.0167	640	41000	71	0	9240	3239

Table 6: Forecasted demand and available machine hours for hypothetical machine 1

Item No	**Period**											
(i)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	39600	39600	36300	46200	46200	39600	29700	6600	6600	6600	16500	16500
2	11537	11537	10575	13459	13459	11537	8652	1923	1923	1923	4807	4807
3	7957	7957	7294	9283	9283	7957	5968	1326	1326	1326	3315	3315
4	11663	11663	10691	13607	13607	11663	8747	1944	1944	1944	4860	4860
5	14105	14105	12929	16456	16456	14105	10579	2351	2351	2351	5877	5877
6	5886	5886	5396	6867	6867	5886	4415	981	981	981	2453	2453
7	1177	1177	1079	1373	1373	1177	883	196	196	196	491	491
8	1962	1962	1799	2289	2289	1962	1472	327	327	327	818	818
	Available Machine hours											
	706	729	729	706	729	706	729	729	660	729	706	729

3.4 Results:

- The output obtained from the heuristic is the lot-size or the actual production plan for n items over the m time periods.
- This lot-size is also used to calculate the total expected setup costs and the total expected inventory holding cost. The summation of these quantities provides us the total expected costs for this production plan.
- The production plan and the total expected costs for this production plan are calculated for both the Data Sets using the Extended Eisenhut Heuristic and the Extended Dixon- Silver Heuristic.
- The outputs have been saved in a separate csv file and the results have been represented in the Table 5, Table 6, Table7 and Table 8 respectively.

Table 5: Final lot sizes for hypothetical machine 1 from Extended Dixon-Silver heuristic

Item No	**Period**											
(i)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	6000	8093	14730	12000	12000	8048	6000	0	0	6000	16500	0
2	0	0	27344	61977	53124	0	48696	39843	0	0	120000	0
3	32906	0	42232	0	0	18099	19606	0	56815	0	0	0
4	0	5694	10792	20042	0	0	6938	3084	29000	0	33	0
5	49000	49000	0	15571	46137	39546	29659	13182	49000	0	72069	0
6	44314	68000	0	0	0	10042	13772	64015	0	0	0	0
7	6881	0	11612	0	0	8708	829	12246	0	0	0	0
8	41000	41000	20318	41000	41000	41000	31267	6948	41000	0	31267	0
9	32816	32000	45386	32000	32000	29068	24612	5469	5469	32000	19142	32000
10	0	0	59646	112868	112868	96745	72559	16124	32248	185000	0	40310
11	0	150000	0	29232	139088	119220	89415	59610	0	150000	49675	150000
12	45190	0	64666	0	0	27715	20786	96227	0	0	0	0

Table 6: Final lot sizes for hypothetical machine 1 from Extended Eisenhut heuristic

Item No	**Period**											
(i)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0	39300	23580	23056	0	11004	8384	48208	1572	1572	28296	46636
2	0	0	0	26873	61773	53048	39786	8725	8725	8725	21987	135412
3	0	15925	16415	21070	21070	17885	13475	2940	2940	49000	7350	0
4	0	0	5504	10664	10664	9116	6880	1376	1376	14448	14276	0
5	0	31061	35947	46068	46068	39437	29316	37692	76082	6282	16403	0
6	64925	18130	16660	21315	0	0	13720	2940	19845	33320	7595	0
7	0	7740	10148	0	0	4816	3612	688	688	688	10664	0
8	125356	116039	60137	0	28798	31339	30492	6776	6776	6776	62678	0
9	94656	77488	29696	0	0	0	24592	5104	5104	93728	13456	0
10	0	0	59225	112700	112700	96600	72450	16100	16100	16100	40250	247250
11	0	0	39091	138710	138710	118534	88270	18915	322816	18915	49179	0
12	0	51051	57018	0	0	27183	20553	3978	3978	3978	82212	0

Table 7: Final lot sizes for hypothetical machine 2 from Extended Dixon-Silver heuristic

Item No (i)	Period 1	2	3	4	5	6	7	8	9	10	11	12
1	42000	36000	48000	44400	37500	29700	12000	6000	6000	18000	96300	5272
2	0	37393	0	0	11537	14421	0	0	0	25059	0	0
3	10551	0	18566	0	0	7957	23216	0	0	0	0	0
4	8502	0	10691	13607	13607	11663	8747	29000	0	0	1705	0
5	15238	0	0	16456	30561	0	17632	0	0	34331	0	0
6	12189	5396	0	13734	0	5886	7358	0	0	17157	0	0
7	0	0	0	2827	0	0	0	0	0	2293	0	0
8	0	0	0	0	1061	1962	3271	0	0	0	4057	0

Table 8: Final lot sizes for hypothetical machine 2 from Extended Eisenhut heuristic

Item No (i)	Period 1	2	3	4	5	6	7	8	9	10	11	12
1	28798	50864	36278	34782	31977	30107	29546	6545	6545	51238	16456	61523
2	0	0	10323	13431	13431	11433	8547	1887	1887	22200	4773	0
3	0	3285	7227	9271	9271	7957	5913	1314	1314	6132	8395	0
4	0	8500	10650	13600	13600	11650	8700	1900	1900	1900	4850	20000
5	0	29526	12876	0	5550	13986	10434	2220	2220	2220	33966	0
6	11919	14933	9864	0	0	0	4384	959	959	959	16851	0
7	0	0	0	0	0	1010	808	101	101	101	2020	0
8	0	0	0	0	923	1917	1420	284	284	284	4757	0

3.5 Interpretation of The Results:

Table 9: Comparison of Inventory, Setup and Total cost calculated by two heuristics considering two data sets

Heuristic -->	Extended Dixon-Silver			Extended Eisenhut		
	Inventory Cost	**Setup Cost**	**Total cost**	**Inventory Cost**	**Setup Cost**	**Total cost**
Data 1	82046	11575	93621	86898	13958	100856
Data 2	19261	17882	37143	11297	31106	42403

- As we can see from the Table 9 for Data 1 the Inventory, Setup and the Total cost for Extended Dixon Silver Heuristic is better than the costs of Extended Eisenhut heuristic.
- We can also see that for Data 2 Extended Eisenhut heuristic is performing better to lower down the inventory cost but on the expense of Setup cost. Therefore, Extended Dixon-Silver heuristic is again better in terms of Total cost.
- So, for a given Dataset with limited lot size per setup we can say that Extended Dixon-Silver heuristic will give better lot sizes than the Extended Eisenhut heuristic.

3.6 Animation

- In–order to better visualize the results, an animation plot has been provided. This animation plot has been generated using the 'matplotlib.animation' library. The animation provides information about the comparison between the total costs obtained from both the heuristics for Data 1 and Data 2.
- The total costs have been represented on y –axis and the data set on the x –axis. The image generated from this animation is as shown in the Figure 1.

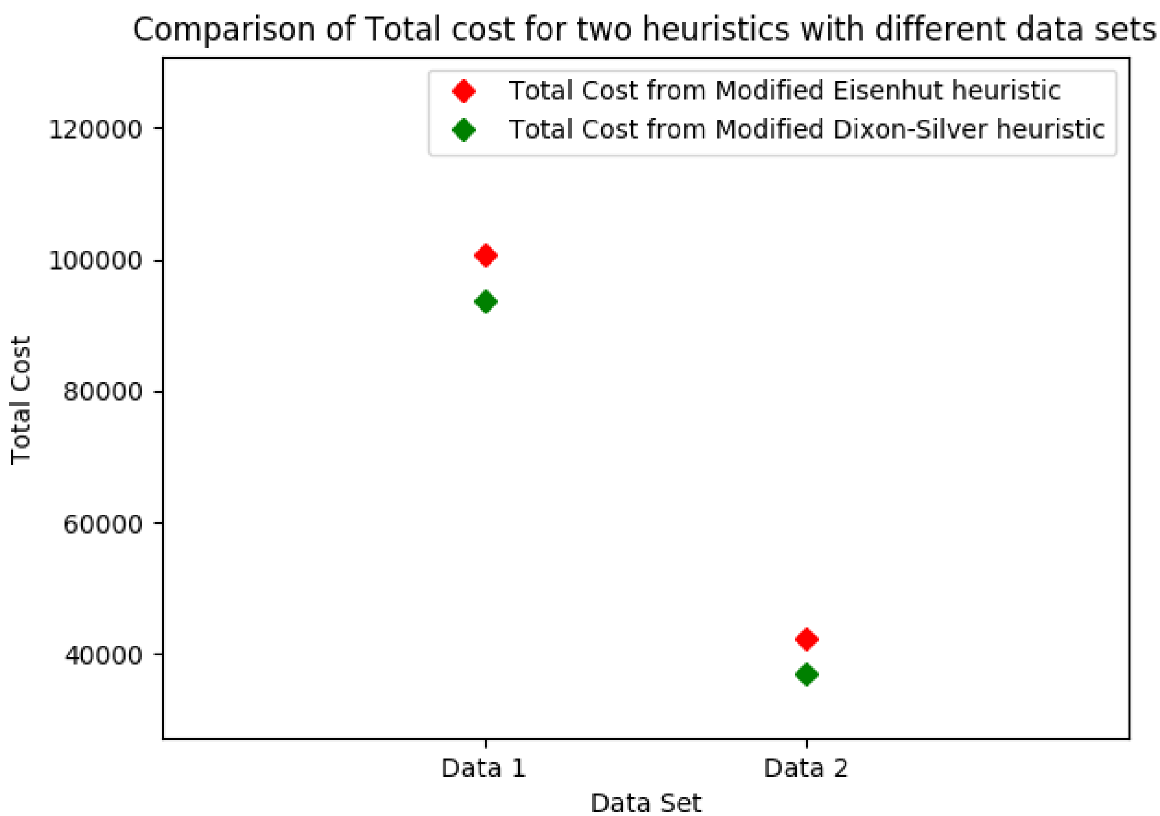

Figure 1: Comparison of Total cost for two heuristics with two different data sets

3.7 Python Libraries

- Pandas:

  As this project demanded working with DataFrames representing the data in a matrix form, Pandas library has been chosen as it provides an easy framework to analyze the data. Pandas has an R- styled DataFrame with column names which helps in keeping track of the data. It also provides good IO capabilities, making it easier to access and output the data from a csv file.

- Math:

  The math library has been imported in order to have access to some of the common math functions which makes it easier to manipulate the data.

- Matplotlib:

  The matplotlib library provides an easy way to generate plots and animations. It is possible to easily include legends for the plots, the axis labels, title of the plot and also provides access to variety of plots.

4. Conclusion and Suggestions:

4.1 Conclusion:

- The extended Eisenhut heuristic is an efficient procedure and can be used to generate feasible production schedules for the multi-item, multi- period capacitated lot sizing problem.
- The extended Dixon- Silver Heuristic contains an important improvement over the original heuristic by including an important parameter which is the maximum lot size from the machine.
- Based on the comparison of these two heuristics it was observed that the Extended Dixon – Silver Heuristic produced lesser total costs for both the data sets. The reason for this may be that, lot –sizing of the current period is done after considering the feasibility from current period to period T_i at the same time. But in the Eisenhut heuristic, the feasibility check is performed for each period at a time.

4.2 Future Suggestions:

- The run-time of the heuristics can be compared to check which heuristic performs better.
- The data sets can be manipulated to check which heuristic performs better for various sorts of data.

5 Appendix:

Instructions to run the codes:

1. Keep the Sample_Data folder and all three codes in the same folder.
2. Open Command line and change the directory to the folder location
3. To get the solution of Extended Dixon-Silver heuristic:
   For Data 1:
   Type in the command line: python Extended_Dixon_Silver.py Data_1
   For Data 2:
   Type in the command line: python Extended_Dixon_Silver.py Data_2
4. To get the solution of Extended Eisenhut Heuristic:
   For Data 1:
   Type in command line: python Extended_Eisenhut.py Data_1
   For Data 2:
   Type in command line: python Extended_Eisenhut.py Data_2
5. To get the animation plot for the heuristics:
   Type in command line: Animation.py

6. References:

Chen WH, Thizy JM. Analysis of relaxations for the multi-item capacitated lot-sizing problem. Annals of Operations Research 1990; 26:29–72.

Dixon PS, Silver EA. A heuristic solution procedure for the multi-item, single level, limited capacity, lot sizing problem. Journal of Operations Management 1981;21(1):23–40.

Eisenhut PS. A dynamic lot sizing algorithm with capacity constraints. AIIE Transactions 1975;7(2):170–6.

Karimi, Behrooz, SMT Fatemi Ghomi, and J. M. Wilson. "The capacitated lot sizing problem: a review of models and algorithms." *Omega* 31.5 (2003): 365-378.

Lambrecht MR, Vanderveken H. Heuristic procedures for the single operation, multi-item loading problem. AIIE Transactions 1979;15(4):319–25.

Maes J, McClain JO, Van Wassenhove LN. Multilevel capacitated lot sizing complexity and LP-based heuristics. European Journal of Operational Research 1991;53(2): 131–48.

https://jakevdp.github.io/blog/2012/08/18/matplotlib-animation-tutorial/

http://stackoverflow.com/questions/18016390/save-matplotlib-animation

http://matplotlib.org/examples/animation/histogram.html

http://stackoverflow.com/questions/28821150/plot-points-with-different-colors-in-a-matplotlib-animation

http://matplotlib.org/api/animation_api.html