Proposing a New Promising Constraint to the Single Stage Capacitated Warehouse Location Problem (SSCWLP)

RRK Sharma
Department of IME, IIT Kanpur 208016 India
Email: rrks@iitk.ac.in

Ajay jha
Department of Operations Management
Jaipuria Institute of Management, Lucknow, 226010 India
Email: ajayjha.knp@gmail.com

Priyank Dubey
Kanpur Divisional Office
Indian Oil Corporation Ltd, Kanpur 208001 India
Email: rehanr94@gmail.com

KK Lai
President, CYUT Taiwan
laikk.tw@gmail.com

Abstract

In this work, we advance the solution approach to the single stage capacitated warehouse location problem (SSCWLP) problem. We took six formulations given by Sharma and Berry (2007) for SSCWLP with various constraints combinations and added a new promising constraint ($\sum capj * yj \geq 1$). The computations done after additional constraint results in statistically better results reduction in number of nodes searched as well as in CPU time taken to solve the problem. We created random 50 samples of problems of size 15x15x15, 30x30x30 and 50x50x50 size problems and solved using GAMS software to show the computational advantage even in such large magnitude problem. We performed paired t-test on all the six pairs of results to observe statistical significance of improvement with the new constraint. Results with new constraint were observed statistically significant and better at two-tailed t-test with 95% confidence interval.

Keywords
Warehouse location problem, Single stage capacitated warehouse location problem, plant location problem, supply chain design and capacitated plant location problem.

1. Introduction
Warehouse Location problem involves identifying the specific locations, which can serve as temporary storage and redistribution point to optimize the total cost of maintenance and transportation in meeting the varying demands of markets, than serving directly from supply points. All the major companies (like Nestle, Unilever, etc.) catering to multiregional, multiproduct and multi-plant management use the warehouse distribution model (also referred to as hub and spoke) model to address the vast global market demand. It also helps in deconsolidating the large shipment to less container loads (LCL) through other smaller transportation modes and serving the purpose of consolidation by aggregating smaller shipments from various plants to large as required by large demand point.

A single stage capacitated warehouse location problem (SSCWLP) is a class of linear programming problems that involves selection of cluster centers or redistribution points that gives the most economical shipping routes for transfer of a uniform commodity from a number of sources to a number of destinations. The set of potential warehouse locations are known and fixed cost associated with them is also known. The objective is to choose the sufficient no of warehouses such that sum total of the fixed location cost and the transportation cost of shipping goods from plants to warehouse and warehouses to market is minimized while satisfying the demand at each point. It also decides upon the quantity of goods transported from various supply points to the warehouse point and further from warehouse to the demand centers.

Warehouse location problems are NP hard problem. If the size of the problem increases i.e. no of plants, warehouses and market location increases, the computational complexity increases exponentially requiring large amount of time and memory space. Researchers have used various approximate algorithms to address such issues. In this paper, the authors add a new constraint in the successful model of SSCWLP given by Sharma and Berry (2007) to achieve significantly better computational time and total cost. The simulation is used to compare the results obtained from this formulation with established standard solutions. A larger generalization of the standard transshipment model consisting of 50 supply points, 50 warehouse locations and 50 demand points is also taken to show the computational superiority of the new formulation. The results obtained are highly encouraging in optimizing the transshipment problem of such large scale. The solution method can serve as an effective tool to the global managers looking for international transshipment points and also the regional plant manager handling production allocation problems.

Next section 2 gives the standard formulations and procedures for solving SSCWLP problem and its variants as given by Sharma and Berry (2007). Section 3 modifies these formulations with addition of a new constraint and discusses the role of constraint addition to the standard solutions. Finally, computational results and comparison with the standard solutions of SSCWLP is done in section 4 and conclusions in section 5 establishing the utility of the new formulation. Reader is also referred to Sharma (2019, 2019, 2020, 2020, 2021 and 2022) for latest details.

2. Sharma and Berry (2007) formulations of SSCWLP

The warehouse location problem is given due importance from very past and lot of research work has already been done in this area (Baurnol and Wolf, 1958; Khumawal, 1972). Also the literature on capacitated warehouse location problem is rich with most of the earlier researchers attempting to solve by Branch and Bound algorithm (Akinc, & Khumawala, 1977; Kaufman, Eede, & Hansen 1979) and Lagrangian relaxation approach (Christofides & Beasley, 1983; Beasle. 1993). However thee problem complexity increases with the increase in number of supply, intermediary and demand points and researchers resort to some heuristic or solve the relaxed integer problem to have an approximate solution.

Sharma and Berry (2007) developed a variety of constraints that link real and binary integer variables for identifying warehouse locations and thus developed many formulations of single stage capacitated warehouse location problem (SSCWLP). Their formulation is as under:

2.1 Constants definition

D_k → demand for the commodity at market ‘k’

d_k → $D_k / \sum D_k$ (demand at market ‘k’ as a fraction of total market demand)

S_i → supply available at plant ‘i’

s_i → $S_i / \sum D_k$ (supply available at plant ‘i’ as a fraction of the total market demand)

F_j → fixed cost of locating a warehouse at ‘j’

cpw_{ij} → cost of transporting $\sum D_k$ quantity of goods from plant ‘i’ to warehouse ‘j’

cwm_{jk} → cost of transporting $\sum D_k$ quantity of goods from warehouse ‘j’ to market ‘k’.

CAP_j → capacity of a warehouse ‘j’

cap_j → $CAP_j / \sum D_k$ (capacity of a warehouse at ‘j’ as a fraction of the total market demand)

2.2 Variable definition

XPW_{ij} → quantity of commodity transported from plant ‘i’, to warehouse ‘j .

xpw_{ij} → $XPW_{ij}/\sum D_k$ (quantity transported as a fraction of total market demand)

XWM_{jk} → quantity of commodity transported from warehouse 'j' to market 'k'.

xwm_{jk} → $XWM_{jk}/\sum D_k$ (quantity transported as a fraction of total market demand)

y_j → 1 if warehouse is located at 'j', 0 otherwise.

Now here are all set of equations that formulate the SSCWLP problem.

2.3 Formulations of SSCWLP

Objective function:

$$min\sum i\sum j\ (cpw_{ij} * xpw_{ij}) + \sum j \sum k\ (cwm_{jk} * xwm_{jk}) + \sum i(f_j * y_j)\ , \qquad (1a)$$

Subject to:

$$\sum i \sum j\ xpw_{ij} = 1, \qquad (2a)$$

$$\sum j \sum k\ xwm_{jk} = 1, \qquad (2b)$$

$$\sum i\ xpw_{ij} = \sum k\ xwm_{jk} \qquad (2c)$$

$$\sum j xpw_{ij} \leq si, \quad \forall i \qquad (3)$$

$$\sum j\ xwmjk \geq d_k, \quad \forall k \qquad (4)$$

$$\sum i\ xpw_{ij} \leq cap_j, \quad \forall j \qquad (5)$$

$$xpw_{ij} \geq 0, \quad \forall i, j \qquad (6a)$$

$$xwm_{jk} \geq 0, \quad \forall j, k \qquad (6b)$$

$$\sum i\ xpw_{ij} \leq y_j, \quad \forall j \qquad (7)$$

$$\sum j\ xwm_{jk} \leq d_k * y_j, \quad \forall j, k \qquad (8)$$

$$xpw_{ij} \leq s_i * y_j, \quad \forall i, j \qquad (9)$$

$$y_j \in \{0, 1\}, \quad \forall j \qquad (10)$$

In the above formulation equation 1 is the objective function of minimizing the total cost of transportation of goods from supply points to the warehouses and from warehouses to demand points as well as cost of establishing the warehouse at particular location. Equation 2a and 2b give the total demand satisfaction condition. Equation 2c is flow balance equation for transshipment point. Equation 3 and 5 are supply and warehouse capacity constraint. Equation 4 gives meeting the total demand condition. Equation 6a and 6b give nonnegative condition on transported quantities. Equation 7 links the binary variable associated with establishing the warehouse and *is a weak constraint*. Equations 8 and 9 also link the binary variable of locating warehouse with transported quantities and *are strong constraints* as they put tighter bound on the solution. Constrain 10 is a binary variable constraint for establishing a warehouse at a particular location.

Finally, Sharma and Berry (2007) considered the alternate ways of linking the fixed cost of the location to positive out flow of goods in SSCWLP. These are categorized as the strong constraints (7 & 8). Let us denote F1 as the weak formulation and F3 and F5 as their strong formulations.

F1. Weak Formulation: Minimize (1a) subject to constraints (2a)–(6b), (7), (10)
F3. Strong Formulation: Minimize (1a) subject to constraints (2a)–(6b), (8), (10)
F5. Strong Formulation: Minimize (1a) subject to constraints. (2a)– (6b), (9), (10)

The definition of strong and weak formulations can be stated as suppose that there are two formulations A and B for the same problem. By excluding the integrality constraints (which restricted variables to take only integer value), we obtain the linear optimization relaxation. Let the feasible region of relaxed formulations be PA and PB. When the region PB contains PA, i.e., PA⊂ PB, formulation A is stronger than formulation B (analogously, B is weaker than A). As PA is narrower than PB, the lower bound obtained by the relaxation A is more close to the optimum of the integer problem. It can be shown by an example in which we can say that $d_k * y_i - x_{ik} \geq 0$, ∀ i ,k is stronger than using only constraint $\sum k\ x_{ik} \leq$ yi , ∀ i. To verify this we assume that PA is the feasible region using the former constraint and PB is the feasible region of latter. It has been observed that the latter constraint were obtained by adding the former, hence PA⊂ PB.

It can be observed that strong formulation though gives better bound, it require many more constraint or variable than weaker formulation and hence the time taken to solve the problem is longer for strong formulation as compare to weak formulation. Therefore, there is a tradeoff between shorter time taken by the weak formulation and better bound given by the strong formulation. However, when the size of the problem increases stronger formulation is considered more desirable than weaker formulation.

Sharma & Berry (20007) relaxed the integer constraints to obtain the relaxations of all the above three formulations. They conducted an empirical investigation to find which formulation gives better bounds than the 'strong' relaxation of SSCWLP. The strong formulations always gave better result in terms of total cost whereas the weak formulation was better in terms of computational effort.

We used Branch and Bound Algorithm to solve the existing formulations and later proposed additional formulations. The basic concept behind Branch and Bound method is to solve linear programming problem by adding newer constraints to the relaxed problem that gives progressively improved lower bounds in case of minimization (upper bound in case of maximization type problem). First, the problem is solved as a simple linear program (without integer requirements of the y's) and gives a value Z0 (lower bound). That is y ε {0,1} (integer) is replaced by 0 <= y <= 1; (real). If all y's (which are initially 0, i.e., the facilities are closed,) are found to be an integer, the problem is solved if not then it is first fixed at zero, and the linear program is solved. However, if say some y1 = 0.7; then we create two child nodes with y1 = 1 and y1 = 0; & resolve the LP relaxation and hence in subsequent steps get two objective functions for y1 producing Z1 and Z2 respectively. Whichever is lower gives the bounding criterion. Min. (z1, z2) is the new lower bound. As more constraints get added, then objective function value goes up. In this way, a tree is being constructed with nodes represented by Z's and the corresponding values of fixed y's. In this way, branching continues and improved lower bound is noted. We need to keep track of only terminal nodes, and if a node is infeasible, then no branches can emanate from it. The process gets terminated when a node with all y's integer is reached, and value is less than the value of any other terminal node.

The major difficulty with branch and bound is computation. For a large number of linear programs, the computational time for each linear program is high and the method could become expensive. Adding more strong constraints allows the Branch and Bound takes lesser time as lesser no of nodes are processed. We attempt to provide a formulation that addresses both the issues of computational effort and the lower bound.

3. A New Formulation (Proposing a constraint in SSCWLP Models)

A new promising constraint ($\sum cap_j * y_j \geq 1$) is added to the different formulations used by Sharma and Berry (2007) and tested empirically establish the strength of this new promising constraint in terms of lesser numbers of node searched and less computing time taken in solving the problem through Branch and Bound algorithm .

$$\sum cap_j * y_j \geq 1 \qquad (11)$$

Therefore, the final mathematical formulations of our research interest are as following F1 to F12:

Formulation 1(F1): min (1a) s. t. (2a)–(6b), (7), (10).

Formulation 2(F2): min (1a) s. t. (2a)–(6b), (7), (10), (11).

Formulation 3(F3): min (1a) s. t. (2a)–(6b), (8), (10).

Formulation 4(F4): min (1a) s. t. (2a)–(6b), (8), (10), (11).

Formulation 5(F5): min (1a) s. t. (2a)–(6b), (9), (10).

Formulation 6(F6): min (1a) s. t. (2a)–(6b), (9), (10), (11).

Formulation 7(F7): min (1a) s. t. (2a)–(6b), (7), (8).

Formulation 8(F8): min (1a) s. t. (2a)–(6b), (7), (8), (11).

Formulation 9(F9): min (1a) s. t. (2a)–(6b), (7), (9).

Formulation 10(F10): min (1a) s. t. (2a)–(6b), (7), (19), (11).

Formulation 11(F11): min (1a) s. t. (2a)–(6b), (8), (19).

Formulation 12(F12): min (1a) s. t. (2a)–(6b), (8), (19), (11).

4. Computational Results and Discussion

Random samples have been generated for the small as well as large sized problems. The solution technique used to solve the formulations was Branch and Bound Algorithm. We modelled all the above 12 sets of formulation on the GAMS. Further, we created 50 different random instances of all the formulations in sets of 15X15X15, 30X30X30, and 50X50X50 of each and solved through Branch and Bound Algorithm on GAMS. We show below only 10 computations of each to reduce space. After obtaining the results, we performed one tailed t-test for number of nodes searched and CPU time taken between the formulations, with and without our proposed constraint ($\sum cap_j * y_j \geq 1$) on MS Excel and then statistically established the strength of this new constraint after finding out if there is a significant difference in number of node searched and CPU time taken (Table 1-Table 18).

4.1 Computational results for problem size 15x15x15

Table 1. Computational results with weak formulation F1 and new constraint added weak formulation F2) for problem size 15x15x15

	F2 (with $\sum cap_j.y_j \geq 1$)				F1 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	4670263.2	0	83.42	6.18794	4670263.2	0	284.12	6.5141

Table 2. Computational results with strong formulation F3 and new constraint added strong formulation F4 for problem size 15x15x15

	F4 (with $\sum cap_j.y_j \geq 1$)				F 3 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	4670263.2	0	93.54	6.0925	4670263.2	0	281.9	6.92722

Table 3. Computational results with strong formulation F5 and new constraint added strong formulation F6 for problem size 15x15x15

	F6 (with $\sum cap_j.y_j \geq 1$)				F 5 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	4670263.2	0	86.42857143	4.14532	4670263.2	0	297.72	4.51528

Table 4. Computational results with strong formulation F7 and new constraint added strong formulation F8) for problem size 15x15x15

	F8 (with $\sum cap_j.y_j \geq 1$)				F 7 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	4670263.2	0	86.96	5.56732	4670263.2	0	309	5.93626

Table 5. Computational results with strong formulation F9 and new constraint added strong formulation F10 for problem size 15x15x15

	F10(with $\sum cap_j.y_j \geq 1$)				F 9 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	4670263.2	0	84.24	6.15964	4670263.2	0	285.92	6.62068

Table 6. Computational results with strong formulation F10 and new constraint added strong formulation F12 for problem size 15x15x15

	F12 (with $\sum cap_j.y_j \geq 1$)				F 11 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	4670263.2	0	86.96	5.56732	4670263.2	0	309	5.93626

4.2 Computational results for problem size 30x30x30

Table 7. Computational results with weak formulation F1 and new constraint added weak formulation F2 for problem size 30x30x30

	F2 (with $\sum cap_j.y_j \geq 1$)				F1 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	8940681.32	0	30368.2	41.92094	8940681.32	0	48159.16	73.3865

Table 8. Computational results with strong formulation F3 and new constraint added weak formulation F4 for problem size 30x30x30

	F4 (with $\sum cap_j.y_j \geq 1$)				F 3 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	8940681.32	0	33311.66	60.90464	8940681.32	0	49665.24	121.64182

Table 9. Computational results with strong formulation F5 and new constraint added weak formulation F6 for problem size 30x30x30

	F6 (with $\sum cap_j.y_j \geq 1$)				F 5 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	8940681.32	0	28176.32	35.58556	8940681.32	0	42154.16	74.96668

Table 10. Computational results with strong formulation F7 and new constraint added weak formulation F8 for problem size 30x30x30

	F8 (with $\sum cap_j.y_j \geq 1$)				F 7 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	8940681.32	0	28561.88	47.86346	8940681.32	0	54380.76	106.01606

Table 11. Computational results with strong formulation F9 and new constraint added weak formulation F10 for problem size 30x30x30

	F10(with $\sum cap_j.y_j \geq 1$)				F 9 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	8940681.32	0	29079.52	37.97924	8940681.32	0	47762.9	73.58074

Table 12. Computational results with strong formulation F11 and new constraint added weak formulation F12 for problem size 30x30x30

	F12 (with $\sum cap_j.y_j \geq 1$)				F 11 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	8940681.32	0	28807.46	48.5684	8940681.32	0	51746.42	100.6726

4.3 Computational results for problem size 50x50x50

Table 13. Computational results with weak formulation F1 and new constraint added weak formulation F2 for problem size 50x50x50

	F2 (with $\sum cap_j.y_j \geq 1$)				F1 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	14975588.8	0	1012148.52	8187.07112	14975588.8	0	1472048.32	11943.415

Table 14. Computational results with strong formulation F3 and new constraint added strong formulation F4 for problem size 50x50x50

	F4 (with $\sum cap_j.y_j \geq 1$)				F 3 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	14975588.8	0	1011545.32	8144.76738	14975588.8	0	1471873.74	11915.33052

Table 15. Computational results with strong formulation F5 and new constraint added strong formulation F6 for

problem size 50x50x50

F6 (with $\sum cap_j.y_j \geq 1$)					F 5 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	14975588.8	0	974216.68	7590.67218	14975588.8	0	1426529.9	11325.53404

Table 16. Computational results with strong formulation F7 and new constraint added strong formulation F8 for problem size 50x50x50

F8 (with $\sum cap_j.y_j \geq 1$)					F 7 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	14975588.8	0	1013783.8	8152.17188	14975588.8	0	1475913.02	11903.00742

Table 17. Computational results with strong formulation F9 and new constraint added strong formulation F10 for problem size 50x50x50

F10(with $\sum cap_j.y_j \geq 1$)					F 9 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	14975588.8	0	1017062.62	8282.67344	14975588.8	0	1480544.72	12016.30368

Table 18. Computational results with strong formulation F11 and new constraint added strong formulation F12 for problem size 50x50x50

F12 (with $\sum cap_j.y_j \geq 1$)					F 11 (without $\sum cap_j.y_j \geq 1$)			
	Objective value	Gap	Node Checked	Time Elapsed	Objective value	Gap	Node Checked	Time Elapsed
Mean	14975588.8	0	1070655.26	8617.09	14975588.8	0	1572972.94	12667.27586

4.4 T-Test Comparison of Results

We performed paired t-test on the computational results mentioned above. The paired t-test is conducted on no. of nodes searched as well as on CPU time taken. This helped us to reach on some conclusive results (Table 19-20).

Hypothesis testing for no. of nodes searched:

Null Hypothesis: There is no significant difference in the no. of nodes searched.
Alternate Hypothesis: There a significant difference in the no. of nodes searched.

Hypothesis testing For CPU time taken:

Null Hypothesis: There is no significant difference in the CPU time taken
Alternate Hypothesis: There a significant difference in the CPU time taken.

Table 19. T-test Analysis for no. of nodes searched

t-test performed on no. Of nodes searched at 95% confidence interval, n=50, DOF =49					
S.N.	Formulations	Problem Size	t Calculated	T Critical for 2-tail	T Critical for 1-tail
1	F2 vs F1	15X15	-7.692993132	2.009575237	1.676550893
		30X30	-3.024122412	2.009575237	1.676550893

		50X50	-2.569415813	2.009575237	1.676550893
2	F4 vs F3	15X15	-6.444099851	2.009575237	1.676550893
		30X30	-3.618425389	2.009575237	1.676550893
		50X50	-2.563978168	2.009575237	1.676550893
3	F6 vs F5	15X15	-7.236870957	2.009575237	1.676550893
		30X30	-2.542055165	2.009575237	1.676550893
		50X50	-2.530554174	2.009575237	1.676550893
4	F8 vs F7	15X15	-7.696802494	2.009575237	1.676550893
		30X30	-3.131948886	2.009575237	1.676550893
		50X50	-2.565370218	2.009575237	1.676550893
5	F10 vs F9	15X15	-7.445227613	2.009575237	1.676550893
		30X30	-3.160285985	2.009575237	1.676550893
		50X50	-2.569018998	2.009575237	1.676550893
6	F12 vs F11	15X15	-6.712797479	2.009575237	1.676550893
		30X30	-3.416984904	2.009575237	1.676550893
		50X50	-2.758405711	2.009575237	1.676550893

Table 20. T-test Analysis for CPU time taken

t-test performed on CPU time taken at 95% confidence interval, n=50, DOF =49					
S.N.	Formulations	Problem Size	t Calculated	T Critical for 2-tail	T Critical for 1-tail
1	F2 vs F1	15X15	-3.56664601	2.009575237	1.676550893
		30X30	-3.85284471	2.009575237	1.676550893
		50X50	-3.4134918	2.009575237	1.676550893
2	F4 vs F3	15X15	-5.93428404	2.009575237	1.676550893
		30X30	-3.5388374	2.009575237	1.676550893
		50X50	-3.35554872	2.009575237	1.676550893
3	F6 vs F5	15X15	-7.87920636	2.009575237	1.676550893
		30X30	-2.72357594	2.009575237	1.676550893
		50X50	-3.34375483	2.009575237	1.676550893
4	F8 vs F7	15X15	-3.31788949	2.009575237	1.676550893
		30X30	-3.86586115	2.009575237	1.676550893
		50X50	-3.30261383	2.009575237	1.676550893
5	F10 vs F9	15X15	-2.79497084	2.009575237	1.676550893
		30X30	-3.72011927	2.009575237	1.676550893
		50X50	-3.31631458	2.009575237	1.676550893
6	F12 vs F11	15X15	-6.23370872	2.009575237	1.676550893
		30X30	-3.89993784	2.009575237	1.676550893
		50X50	-3.40879172	2.009575237	1.676550893

4.4.1 Comparison F2 vs F1

It can be clearly observed from t values coming out of two tailed t-test in both the tables that the difference between both the formulations in no. of nodes searched and CPU time taken is statistically significant in small as well as in large size problems .so we can reject the null hypothesis .also from one tailed t-test in both the tables it is worth to see that F2(with $\sum cap_j.y_j \geq 1$) is better as compare to F1(without $\sum cap_j.y_j \geq 1$)in terms of no of nodes searched and CPU time taken.

4.4.2 Comparison F4 vs F3

It can be clearly seen from t values coming out of two tailed t-test in both the tables, that the difference between both the formulations in no. of nodes searched and CPU time taken is statistically significant in small as well as in large size problems, so we can reject the null hypothesis .Also from one tailed t-test in both the tables it is worth to see that F4(with $\sum cap_j.y_j \geq 1$) is better as compare to F3(without $\sum cap_j.y_j \geq 1$) in terms of no of nodes searched and CPU time taken.

4.43 Comparison F6 vs F5

It can be clearly seen from t values coming out of two tailed t-test in both the tables, that the difference between both the formulations in no. of nodes searched and CPU time taken is statistically significant in small as well as in large size problems , so we can reject the null hypothesis. Also from one tailed t-test in both the tables it is worth to see that F6(with $\sum cap_j.y_j \geq 1$) is better as compare to F5(without $\sum cap_j.y_j \geq 1$) in terms of no of nodes searched and CPU time taken.

4.4.4 Comparison F8 vs F7

It can be clearly seen from t values coming out of two tailed t-test in both the tables, that the difference between both the formulations in no. of nodes searched and CPU time taken is statistically significant in small as well as in large size problems, so we can reject the null hypothesis. Also from one tailed t-test in both the tables it is worth to see that F8(with $\sum cap_j.y_j \geq 1$) is better as compare to F7(without $\sum cap_j.y_j \geq 1$) in terms of no of nodes searched and CPU time taken.

4.4.5 Comparison F10 vs F9

It can be clearly seen from t values coming out of two tailed t-test in both the tables, that the difference between both the formulations in no. of nodes searched and CPU time taken is statistically significant in small as well as in large size problems, so we can reject the null hypothesis. Also from one tailed t-test in both the tables it is worth to see that F10(with $\sum cap_j.y_j \geq 1$) is better as compare to F9(without $\sum cap_j.y_j \geq 1$) in terms of no of nodes searched and CPU time taken.

4.4.6 Comparison F12 vs F11

It can be clearly seen from t values coming out of two tailed t-test in both the tables, that the difference between both the formulations in no. of nodes searched and CPU time taken is statistically significant in small as well as in large size problems, so we can reject the null hypothesis .also from one tailed t-test in both the tables it is worth to see that F12(with $\sum cap_j.y_j \geq 1$) is better as compare to F11(without $\sum cap_j.y_j \geq 1$) in terms of no of nodes searched and CPU time taken.

5. Conclusion

In this work, we add one new promising constraint in the various formulations proposed in the style of Sharma and Sharma in the quest of finding the strength and computational benefit of the new constraint added. All formulations were solved on GAMS with Branch and Bound algorithm. After performing the statistical analysis it can clearly be established that the formulations having the new added constraint ($\sum cap_j.y_j \geq 1$) take less no of node searched also can be executed in lesser computational time as compare to their counterpart formulations, not having this constraint. This was statistically evident in both small as well as large size of problems taken.

However, we couldn't go beyond 50X50X50 size problem because of the termination of program run on the GAMS due to low processing capacity of the system but the strength of this constraint can further be checked on problems having size of 100X100X100 and beyond. In whole of the analysis, we considered single commodity problem but this work can be extended to multi commodity problems as well.

References

Akinc, U., & Khumawala, B. M., An efficient branch and bound algorithm for the capacitated warehouse location problem. Management Science, 23(6), 585-594, 1977.

Baumol, W. J., & Wolfe, P., A warehouse-location problem. Operations research, 6(2), 252-263, 1958.

Beasley, J. E., Lagrangean heuristics for location problems. European journal of operational research, 65(3), 383-399, 1993.

Christofides, N. B. J. E., & Beasley, J. E., Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. European Journal of Operational Research, 12(1), 19-28, 1983.

Kaufman, L., Eede, M. V., & Hansen, P., A plant and warehouse location problem. Journal of the Operational Research Society, 28(3), 547-554, 1977.

Khumawala, B. M., An efficient branch and bound algorithm for the warehouse location problem. Management science, 18(12), B-718, 1972.

RRK Sharma, "Working Paper Series: Lecture Notes in Management Science: Vol 1, A collection of 148 working papers, (All Authored by Prof. RRK Sharma). EXCEL PUBLISHERS NEW DELHI, p. 149. ISBN: 9-789-388-237116, 2019.

RRK Sharma, Working Paper Series: Lecture Notes in Management Science: Vol 2, A collection of 295 working papers, (All Authored by Prof. RRK Sharma); EXCEL PUBLISHERS NEW DELHI, Aug 2019; p. 234. ISBN: 9-789-388-237796. 2019

RRK Sharma, Working Paper Series: Lecture Notes in Management Science: Vol 3", (150 articles are written: All Authored by Prof. RRK Sharma); FEB 2020. ISBN: 978-93-89947-08-3, p. 156, 2020.

RRK Sharma, Working Paper Series: Lecture Notes in Management Science: Vol 4", (It has 01 article + 4 software: All Authored by Prof. RRK Sharma); p. 192. ISBN: 9789389947212, 2020.

RRK Sharma, "Working Paper Series: Lecture Notes in Management Science: Vol 5", (It has 139 articles are written: All Authored by Prof. RRK Sharma); ISBN: 978-93-89947-31-1; 2021.

RRK Sharma, Working Paper Series: Lecture Notes in Management Science: Vol 6", (It has 048 articles are written so far: All Authored by Prof. RRK Sharma); ISBN: 978-93-91355-65-4; 2022.

Sharma, R. R. K., & Berry, V., Developing new formulations and relaxations of single stage capacitated warehouse location problem (SSCWLP): Empirical investigation for assessing relative strengths and computational effort. European journal of operational research, 177(2), 803-812, 2007.

Biographies

Prof. RRK Sharma: He is B.E. (mechanical engineering) from VNIT Nagpur India, and PhD in management from I.I.M., Ahmedabad, INDIA. He has nearly three years of experience in automotive companies in India (Tata Motors and TVS-Suzuki). He has 32 years of teaching and research experience at the Department of Industrial and Management Engineering, I.I.T., Kanpur, 208016 INDIA. To date he has written 1217 papers (peer-reviewed (401) / under review (35) / working papers 781 (not referred)). He has developed over ten software products. To date, he has guided 66 M TECH and 23 Ph D theses at I.I.T. Kanpur. He has been Sanjay Mittal Chair Professor at IIT KANPUR (15.09.2015 to 14.09.2018) and is currently a H.A.G. scale professor at I.I.T. Kanpur. In 2015, he received "Membership Award" given by IABE USA (International Academy of Business and Economics). In 2016 he received the "Distinguished Educator Award" from IEOM (Industrial Engineering and Operations Management) Society, U.S.A. In 2021, he received IEOM Distinguished Service Award. In 2019 and 2020, he was invited by the Ministry of Human Resources Department, India, to participate in the NIRF rankings survey for management schools in India. In 2019, he was invited to participate in the Q.S. ranking exercise for ranking management schools in South Asia.

Dr. Ajay Jha: He is working as an Associate Professor at Jaipuria Institute of Management, Lucknow. He is a Ph.D. in the area of Strategy Implementation (specifically technological innovation management and supply chain management) from Indian Institute of Technology, Kanpur (IIT Kanpur), India. His other educational qualifications include B. Tech. (Mechanical Engineering) from HBTI, Kanpur and M. Tech. (Industrial and Management Engineering) from IIT Kanpur. He has published in top ranked International journals and Conferences, for example, Supply Chain Management: an International Journal, International Journal of Productivity and Performance Management, Management of Environmental Quality: An International Journal , The TQM Journal and numerous papers in IEOM Conferences.

Mr. Priyank Dubey: He post graduated from IIT Kanpur in 2020 from Industrial and Management Engineering Department. He completed my B. Tech. in Mechanical Engineering from K.I.E.T., Ghaziabad (UP Technical University) in 2014. He is currently working as Engineering Officer, Kanpur Divisional Office in Indian Oil Corporation. Before that he also worked as Data Analyst for 6 months in HSBC.

Prof. KK Lai: He is currently President of CYUT Taiwan. He has numerous publications to his credit.