1 Logistics related cost calculation

Transporting biomass from the farm gate to the plant requires transporting to the satellite storage, storing in the satellite storage, loading onto trucks in the satellite storage and transporting to the plant. Each of these processes adds to the cost of the biomass. Furthermore, there will be costs associated with the dry matter loss. Therefore the transportation problem of biomass should reflect all these costs that should be minimized. Most of the formulae we used to calculate these costs were based on the thesis "DESIGN AND SCHEDULING OF AGRICULTURAL BIOMASS SUPPLY CHAIN FOR A CELLULOSIC ETHANOL PLANT" by Mahmood Ebadian.

1.1 Transport cost to Satellite storage from farm

We assume that the stinger stacker which is used to collect bales on the farm are used to transport the bales from the farm to the satellite store. Therefore no additional equipment are needed. But the disadvantage of using this equipment is, it being slow when traveling (as per Table 3-4 the speed on public roads is 25 km/h). The parameters for this calculation are given in Table 1. We calculate the distance traveled by the stinger-stacker using the Manhattan distance or the L1 norm.

parameter	Description	Value
$speed_{stinger}$	Maximum speed	25 km/h
$E_{stinger}$	Moving efficiency	0.7
$C_{stinger}$	Cost per hour of using	77.67 $/h
$LoadMass_{stinger}$	Maximum load	4 tons
$L_{stinger}$	Loading time per bale	0.00417 h/bale
$U_{stinger}$	Unloading time per bale	0.002083 h/bale
$E_{stinger_L}$	Loading efficiency	0.65
$E_{stinger_U}$	Unloading efficiency	0.75

Table 1: Stinger-stacker parameters

Using these parameters we calculate the cost of transporting one *ton* of biomass one *km* using Equation 1.

$$L1_{weight} = \frac{2}{(speed_{stinger} * E_{stinger})} * \frac{C_{stinger}}{LoadMass_{stinger}} \tag{1}$$

1.2 Transport cost to Plant from satellite storage

The transportation from the satellite storage is done using a trailer and a truck. It is assumed that each storage is equipped with a tele-handler which loads biomass onto the trailer. The cost of this tele-handler is considered to be the fixed cost associated with establishing a storage. The parameters for the cost calculation of this transport cost are given in Table 2. These values are taken from tables 3-4 and 3-5 of [1], and [3, 4, 5, 6]

parameter	Description	Value
$speed_{truck}$	Maximum speed	80 km/h
E_{truck}	Moving efficiency	0.75
C_{truck}	Cost per hour of truck	97.90 $/h
$C_{trailor}$	Cost per hour of trailer	23.39 $/h
$LoadMass_{truck}$	Maximum load	15 tons
$LoadMass_{loader}$	Maximum load	1 ton
L_{loader}	Time to load one load	0.017 h/load
U_{loader}	Time to unload one load	0.004167 h/load
E_{loader}	load/unload efficiency	0.75

Table 2: Road transport parameters

Using the parameters above we can calculate the cost of transporting a *ton* of biomass one *km* using the Equation (2).

$$highway_{weight} = 2 * \frac{C_{trailer} + C_{truck}}{speed_{truck} * E_{truck} * LoadMass_{truck}} \tag{2}$$

Further to the transport cost we should also add the cost of the trailer and truck when it is idling at the storage till the loader loads the biomass. That cost per ton of Biomass can be calculated using Equation (3). This will not make a difference when assigning sources to stores if the dry matter loss in the road transportation does not depend on the distance.

$$idling_{loading} = \frac{C_{trailer} + C_{truck}}{E_{loader} * LoadMass_{loader}} * (L_{loader} + U_{loader}) \tag{3}$$

1.3 Cost of dry matter loss (DML)

Although the current results does not include the DML costs, we have calculated them for the logistics operation. The DML cost in transporting and storing can be broken down into four parts:

1. DML when transporting from farms to Storage (L1 transport DML)
2. DML when the biomass is held at the storage (storage DML)
3. DML when the biomass is loaded onto trucks (Loading DML)
4. DML when transporting from satellite storage to the plant (Road transport DML)

In this study up to now we have used the same assumptions as per the Thesis of Mahmood Ebadian in terms of the dry matter loss. In this thesis they assumed that the DML can be treated as a percentage loss of the total mass available. Furthermore different types of storage facilities have different costs of establishment and different percentage of DML. As one can anticipate higher the cost of establishment lower will be the percentage DML. We will be incorporating this aspect into the optimization model as the next step. Table 3 contains the parameter values used for this calculation. These values are based on Table 4-1 and 4-2 of [1], and [2].

parameter	Description	Value
$\theta_{stinger}$	DML per ton of biomass transported by stinger	0.0084 tons
θ_{loader}	DML per ton of biomass loaded by loader	0.0091 tons
$\theta_{EncBuild}$	DML in an enclosed building on crushed rock	2%
$\theta_{OpenBuild}$	DML in an open building on crushed rock	4%
$\theta_{tarpRock}$	DML on biomass on crushed rock covered by tarp	7%
θ_{Rock}	DML on biomass on crushed rock	15%
θ_{Ground}	DML on biomass left on ground	25%
θ_{truck}	DML per ton of biomass loaded on truck	0.0089 tons
$C_{EncBuild}$	Cost of an enclosed building on crushed rock	70.39-107.64 $\$/m^2$
$C_{OpenBuild}$	Cost of an open building on crushed rock	53.82 $\$/m^2$
$C_{tarpRock}$	Storage cost of biomass on crushed rock covered by tarp	4.17 $\$/m^2$
C_{Rock}	Storage cost of biomass on crushed rock	2.70$\$/m^2$
C_{Ground}	Storage cost of biomass left on ground	0 $\$/m^2$
C_{loader_v}	Variable cost of loader	47.25 $/h

Table 3: DML parameters

Calculation of the amount of DML in each of the above mentioned states, per ton of biomass collected from the farm are given in equations (4), (5), (6) and (7) respectively.

$$DML_{L1transport} = \theta_{stinger} \tag{4}$$

The DML cost in storage per ton of biomass collected from farms is given in Equation (5), where $storage \in \{EncBuild, OpenBuild, tarpRock, Rock, Ground\}$. As one can see the DML as a percentage of biomass does not depend on the distance traveled. This can be a potential improvement to the model.

$$DML_{storage} = (1 - \theta_{stinger}) * \theta_{storage} \tag{5}$$

$$DML_{loading} = (1 - \theta_{stinger}) * (1 - \theta_{storage}) * \theta_{loader} \tag{6}$$

$$DML_{truck} = (1 - \theta_{stinger}) * (1 - \theta_{storage}) * (1 - \theta_{loader}) * \theta_{truck} \tag{7}$$

2 Optimization using Mixed Integer Linear Programming (MILP) solver

The optimization problem at hand is about the assignment of farms to satellite storage such that the total cost which consists of transportation and fixed storage cost is minimized (in a future extension we are hoping to consider variable storage costs and dry matter loss). In the thesis mentioned above they just solve the optimization problem (8) without considering about the complexity of the problem. They also included multiple years and multiple crop types in the model. Therefore in the optimization problem was about assigning crop type k of farm i to storage j in time period t. In their problem they had bigger farms and they had a crop mix for each farm. In our case we assumed each quarter section is a farm growing a single crop. This assumption was taken to get a better approximation to the actual situation. Therefore in our case we had a lot more farms. Furthermore in the thesis they were trying to find the roadside storage facilities of farms which could be used as satellite stores. In our case we first find suitable locations for satellite storage facilities based on the ease of access for the trucks and then filter them out based on the cost. All the methods introduced below to reduce the complexity of the solution procedure is what is novel about the work carried out by us.

When trying to solve this problem which had many farms and stores the main problem we faced was the time taken to get a solution. In order to make the problem tractable we group the quarter sections according to the nearest potential storage. Therefore all the quarter sections $\mathcal{Q}_i$ has the potential storage i as the storage which minimizes the transport cost to the plant if biomass is sent via i. After grouping we can view the problem as having m sources and n storage units, where n is the number of potential storage locations. The downside to this simplification is not being able to selectively choose quarter sections. Here onward we treat the set $\mathcal{Q}_i$ as source i.

The mathematical formulation of the problem is shown below:

$$\text{minimize } Z = \sum_{i=1}^{m}\sum_{j=1}^{n} S_i C_{ij} x_{ij} + \sum_{j=1}^{n} F_j y_j \tag{8}$$

subject to

$$x_{ij} \leq y_j \qquad 1 \leq i \leq m, 1 \leq j \leq n \tag{9}$$

$$\sum_{j=1}^{n} x_{ij} \leq 1 \qquad 1 \leq i \leq m \tag{10}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} S_i R_{ij} x_{ij} \geq D \tag{11}$$

$$x_{ij} \in \{0,1\},\ y_j \in \{0,1\} \qquad 1 \leq i \leq m, 1 \leq j \leq n$$

In this formulation S_i is the available biomass at storage i. As mentioned above storage i is a collection of quarter sections having the same biomass type k which has storage l as the best satellite storage in terms of transport cost. Therefore if we have q satellite storage points and r biomass types we model the system as having $m = q * r$ source nodes. In a similar line of thinking the storage index j represents both storage position and storage type. Therefore if we have q storage locations and t storage types we have $n = q * t$ sink nodes. C_{ij} in the above model represents the cost per km-tonne of biomass and R_{ij} represents the percentage remainder when transporting biomass from source i to the plant via sink j. Thus, $R_{ij} = 1 - DML_{L1transport} - DML_{storage} - DML_{loading} - DML_{truck}$.

We use a python based linear programming (LP) modeler called PuLP to formulate the problem in *.lp format so that it can be solved with most available open source solvers. We use the MILP solver called SYMPHONY to solve the formulated problem. The advantage of using SYMPHONY is the ability of it to utilize multiple processor cores in running the Branch and cut algorithm in generating the binary solutions. In order to use the SYMPHONY utilizing multiple cores we need to configure and build it from the source with the option ”–enable-openmp”. We call this function in our code as a sub-process giving the *.lp file as the argument.

When the original problem has a large number (> 50) of source and sink nodes we first use a variable elimination algorithm to eliminate a number of sources and storage points which should not be considered in the solution for a given demand and fixed cost. Using this elimination we can speed up solving the optimization problem significantly.

3 Optimization using a heuristic approach making use of uni-modular structure

In the heuristic approach we decompose the solving of (8) into two parts. In the first part we solve the LP for relaxed x_{ij} ($0 \leq x_{ij} \leq 1$) for a given y_js. The LP problem formulation for this modified problem is shown in Equation (12). Here we first set the y_j to be all ones, which means that all the potential storage places are used. Since this is the initial guess we call it $\mathbf{Y}^0$ ($k = 0$). All the superscripts in what is to follow are just indices. Let the solution of this problem be $\mathbf{x}^0$. Then we solve the LP given in Equation (13) fixing the supply at source i at X_i, for x_{ij} and y_j. The supply X_i is calculated based on the solution $\mathbf{x}^k$ for problem (12) at the k^{th} iteration as $X_i = \sum_{j=1}^{n} S_i * x_{ij}^k$. This problem with fixed supply and no demand constraints has a special structure where the constraint matrix becomes totally unimodular. The advantage of having this property is that the LP solution to the problem will be integer. Therefore, the computational effort will be far less than solving an MILP.

$$
\begin{array}{rr}
\min\limits_{\mathbf{x}} & \sum_{i=1}^{m}\sum_{j=1}^{n} S_i C_{ij} x_{ij} \\
\text{s.t.} x_{ij} \leq Y_j^k & \forall 1 \leq i \leq m, 1 \leq j \leq n \\
\sum_{j=1}^{n} x_{ij} \leq 1 & \forall 1 \leq i \leq m \\
\sum_{i=1}^{n}\sum_{j=1}^{n} S_i R_{ij} x_{ij} \geq D & \forall 1 \leq j \leq n \\
\\
0 \leq x_{ij} \leq 1 & \forall 1 \leq i \leq m, 1 \leq j \leq n
\end{array}
\tag{12}
$$

$$
\begin{array}{llllr}
\min\limits_{\mathbf{x},\mathbf{y}} & \sum_{i=1}^{n}\sum_{j=1}^{n} X_i C_{ij} x_{ij} + F\sum_{j=1}^{n} y_j & & & \\
\text{s.t.} & x_{ij} \quad \leq \quad y_j & \forall 1 \leq i \leq m, 1 \leq j \leq n & & \\
& \sum_{j=1}^{n} x_{ij} \quad = \quad 1 & \forall 1 \leq i \leq m & & (13) \\
& x_{ij} \in \{0,1\} \quad y_j \in \{0,1\} & \forall 1 \leq i \leq m, 1 \leq j \leq n & &
\end{array}
$$

Let the solution of this problem be $\mathbf{x}^{k+1}$ and $\mathbf{y}^{k+1}$. Then we fix $\mathbf{Y}^k$ in problem (12) to be $\mathbf{y}^{k+1}$ and solve for relaxed x_{ij} $(0 \leq x_{ij} \leq 1)$ followed by solving problem (13). If the cost for the allocation $\mathbf{x}^{k+1}$ and $\mathbf{y}^{k+1}$ is less than $\mathbf{x}^k$ and $\mathbf{y}^k$ we continue the process. If it is more, then we assume $\mathbf{x}^k$ and $\mathbf{y}^k$ to be our solution and stop the process.

Algorithm 1: Optimization using unimodular heuristic

$k \leftarrow 0$;
$\mathbf{Y}_k \leftarrow [1, 1, ..., 1]$;
$z^{-1} \leftarrow \infty$;
while *1* **do**
    Solve (12) $\rightarrow \mathbf{x}$;
    $X_i = S_i * \sum_{j=1}^{n} x_{ij} \forall i = \{1, 2, ..., m\}$;
    Solve (13) $\rightarrow \mathbf{x}, \mathbf{y}$;
    Objective function value of (13) $\rightarrow z^k$;
    if $z^{k-1} < z^k$ **then**
        break;
    else
        $k \leftarrow k + 1$;
        $Y_k \leftarrow \mathbf{y}$;
    end
end

4 Optimization using a heuristic approach making use of VEB trees

This method is very similar to the unimodular solution method, but in here relaxed x_{ij} $(0 \leq x_{ij} \leq 1)$ for a given $\mathbf{Y}^k$ is found without solving an LP. Here also we initialize the process setting $Y_j^0 = 1 \qquad \forall j \in \{1, ..., n\}$, which means that all the potential storage locations are being used. Let the cost of transporting a ton of biomass from source i to plant via storage j be C_{ij}. Then we draw biomass from the ascending order of C_{ij} till the demand is satisfied. Then we keep note of the total cost Z^0 and create a set of n storage allocations $\mathcal{Y}^1$ by flipping each bit of $\mathbf{Y}^0$ and draw biomass to full fill the demand as earlier and find the allocation $\mathbf{Y}^1$ which gives the least cost Z^1 from the set $\mathcal{Y}^1$. The difference in this case is that you cannot use the stores which are not used. In other words if $\mathbf{Y}_j^1 = 0$ then $C_{ij} = \infty \quad \forall i$. Then if $Z^1 < Z^0$ we flip each non-zero bit in $\mathbf{Y}^1$ to create a set of $n-1$ storage allocations $\mathcal{Y}^2$. Then the same procedure outlined above is carried out. This process goes on until $Z^{k+1} >= Z^k$ and we take $\mathbf{Y}^k$ to be the best storage allocation.

As mentioned above we have to find the source to storage assignment for a given storage allocation to minimize the cost. To make this more efficient we make use of a tree structure called the VEB trees.

4.1 Construction and usage of VEB trees

VEB trees can only accommodate unique integers. In our case these unique integers are the positions of each source storage pair in a sequence when the source storage pairs are put in the ascending order of the cost of transporting a ton of biomass from a given source i to the plant via the given storage j, $C_{ij}\ \forall i, j \in \{1, 2, 3, ..., n\}$. First we augment the entry C_{ij} by adding the row and column indices to the entry as $[C_{ij}, i, j]$. Then these augmented $C_{ij} \qquad \forall i, j \in \{1, 2, ..., n\}$ are arranged in the ascending order of C_{ij} and the position number of each C_{ij} is stored in a matrix $\mathbf{B}$, where the position of C_{ij} in the list is stored in the ij^{th} element of B_{ij}. Then each row of matrix $\mathbf{B}$, B_i, is used to create a VEB tree T_i. In these trees the insert, search and delete operations all can be done with a time complexity of $\mathcal{O}(\log(\log(M)))$ where M is the maximum integer you are storing in the tree. In other words all the above mentioned operations could be done very quickly when VEB trees are used. Then using the $T_i^{min} \quad \forall i \in \{1, 2, ..., n\}$, which are the minimum position indices of each of the trees T_i, we build another VEB tree T_{min}. After building the trees for the complete set of source storage pairs, we have to modify the trees to take account of the stores which are not used. In the k^{th} iteration if $Y_j^q = 0$ where $\mathbf{Y}^q \in \mathcal{Y}^k$, we take all the indices in the j^{th} column of $\mathbf{B}$ and remove the index in the i^{th} row from T_i and this is continued for all $i \in \{1, 2, ..., n\}$. After doing this if any of the $T_i^{min} \quad \forall i \in \{1, 2, ..., n\}$ is changed then this should be updated in the T_{min}. This change is temporary till we find the cost for all the allocations in $\mathcal{Y}^k$. Once we find the allocation with the minimum cost in $\mathcal{Y}^k$ we make it permanent.

In these trees there is a built-in function "successor" which returns the next smallest index given an index. Using the tree T_{min} and the function successor we can find the indices of the C_{ij} to be used in the ascending order of cost. We start with the minimum index of T_{min}, which is a stored value in the tree. Let this minimum index in the T_{min} be T_{min}^{min}. Then we take entry number T_{min}^{min} on the ascending sequence. Let this entry be $[C_{lm}, l, m]$. Then if after pulling S_l amount of biomass from storage l via m, does not fully satisfy the demand or if it exactly satisfies, we assign value 1 to x_{lm}. If it goes above the demand then we find x_{lm} such that $S_l * x_{lm}$ strictly satisfies the demand. If the demand is not still satisfied we find the next smallest index after T_{min}^{min} using the function successor on the tree T_{min} and repeat the procedure. This repetition is done till the demand is satisfied. Let the total cost be Z_q^k after the demand is satisfied in the k^{th} iteration for a storage allocation $\mathbf{Y}^q \in \mathcal{Y}^k$. Then we modify the trees as explained above for $\mathbf{Y}^{q+1} \in \mathcal{Y}^k$. Then the same procedure is followed till we have evaluated all the storage allocations in $\mathcal{Y}^k$. Then the allocation in $\mathcal{Y}^k$ which gives the lowest cost is considered the best allocation in iteration k. Let this allocation be $\mathbf{Y}^{k*}$ and the optimal cost be Z^k. Then if the cost $Z^k < Z^{k-1}$ we generate another set of allocations $\mathcal{Y}^{k+1}$ flipping each non-zero bit one at a time and continue the procedure until $Z^k => Z^{k-1}$ for some k. After the termination we consider Z^{k-1} to be the optimum cost and $\mathbf{Y}^{k-1*}$ to be the optimal storage allocation and the respective x_{ij} to be the optimal source to storage mapping.

4.2 Using LP to find $\mathbf{Y}^0$

To further speedup the algorithm we relax the integer constraints in (8) and solve it. Then we assign the value one to all the y_js which are strictly greater than zero. This modified $\mathbf{y}$ is used as the $\mathbf{Y}^0$.

5 Comparison of the VEB and MILP in-terms of time required to solve after variables are eliminated

The time required to get a solution depends on the demand requirement and the fixed cost because they determine the number of sources and stores left after the variable elimination is done. Therefore, we plot the time required against the demand in Figure 1. The time required is averaged over 100 iterations for each demand level to average out the random effects. This timings are for a single crop type.

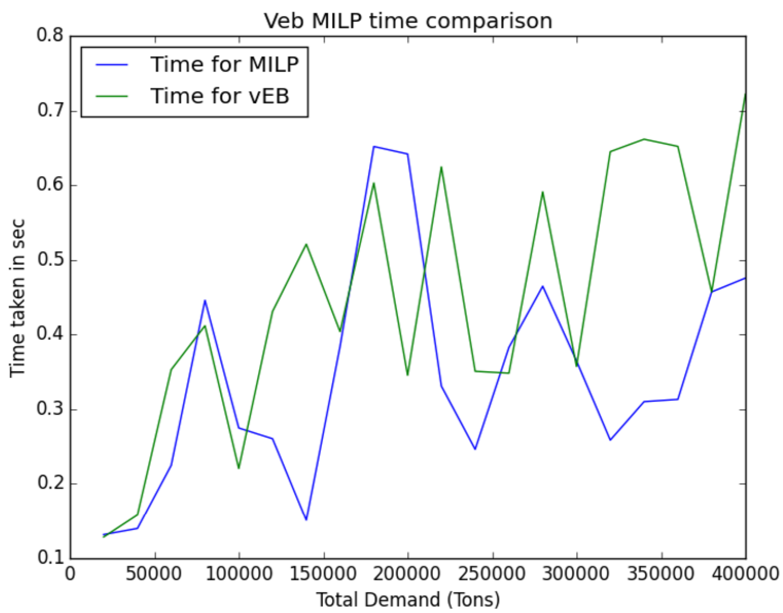

Figure 1: Execution time for VEB and MILP

If the variable elimination is not used the time required will be more than 10 fold high. The execution time without the variable elimination is shown in Figure 2.

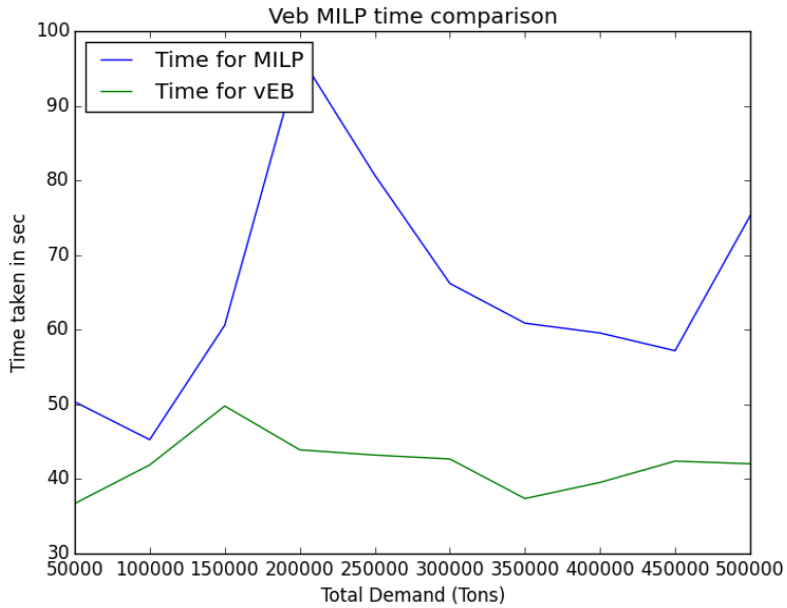

Figure 2: Execution time for VEB and MILP without variable elimination

6 Cost estimates for different biomass types

In the optimization problem the plan was to treat the tuple biomass type and source storage as a single source and find the best combination of biomass types and storage locations to minimize the cost. The cost per ton of biomass mix for this allocation is given in Figure 3. But when doing that the variable elimination algorithm could not filter out much of the sources because to satisfy the demand you need too many sources since each source only has one type of biomass and the quantity is small. In the filtering of sinks too the problem was there since in the filtering algorithm a sink can be eliminated only when the cost of establishing a sink j is larger than the cost difference of routing the biomass through sink k instead of sink j. Since there were a lot of sources and sinks the optimization algorithm takes a longer time to converge. As a temporary solution to the problem, we grouped the biomass types in a particular storage. Therefore, this can now be viewed as a source having a single biomass having the cost to be the weighted average of all the biomass types in that source.

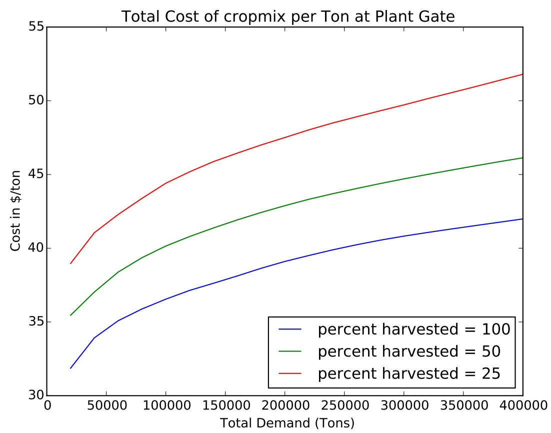

Figure 3: Total cost at plant for biomass mix

6.1 Total cost at plant for a mix of biomass types

Here we find the weighted average cost of a ton of biomass taking into account the supply at each source and the cost per ton of each biomass type. Then we draw biomass such that the total cost is minimized. The difference here is treating the storage to have only one type of biomass having the weighted cost. This weighted cost per ton for different demands are shown in Figure 4.

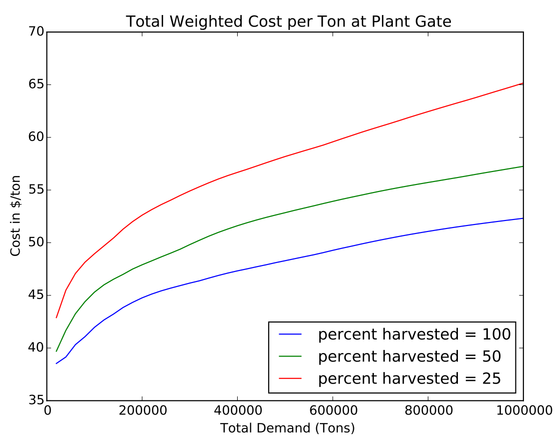

Figure 4: Total weighted cost at plant for biomass mix

We further include the costs for cattail and wheat for comparison purposes in figures 5 and 6.

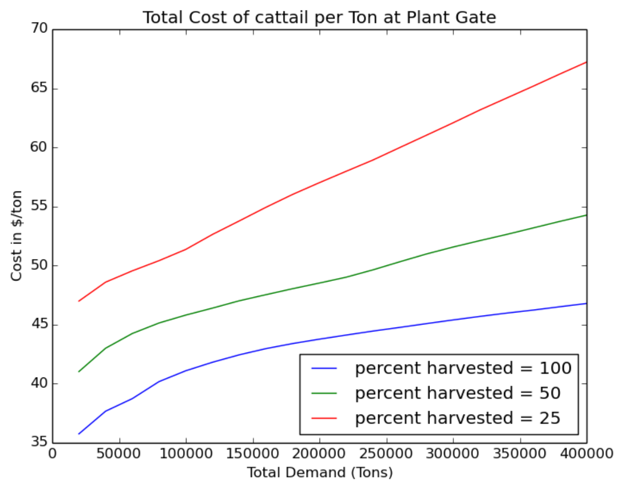

Figure 5: Total cost at plant for Cattail

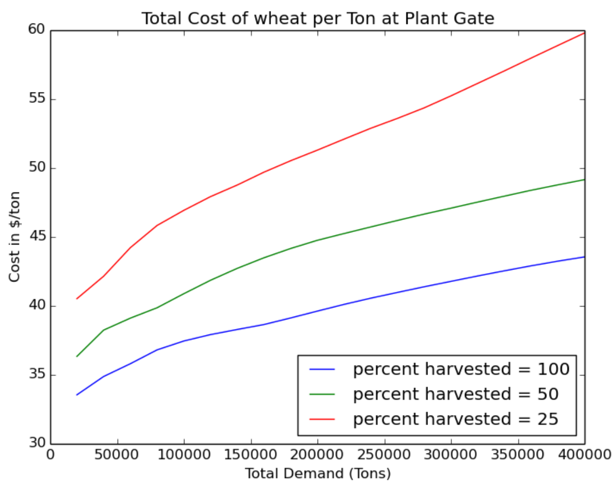

Figure 6: Total cost at plant for Wheat

6.2 Transport cost from farm to storage

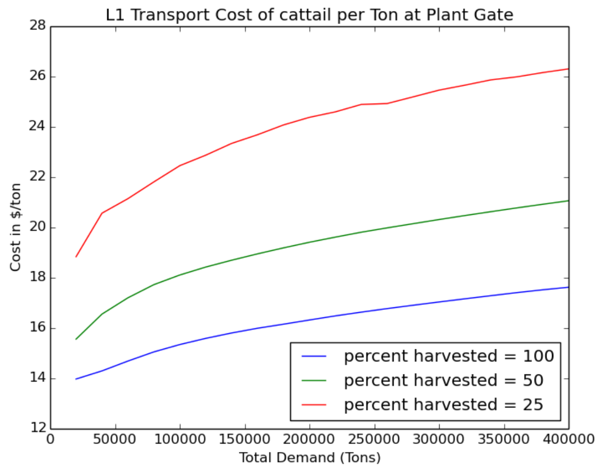

Figure 7: L1 cost for Cattail

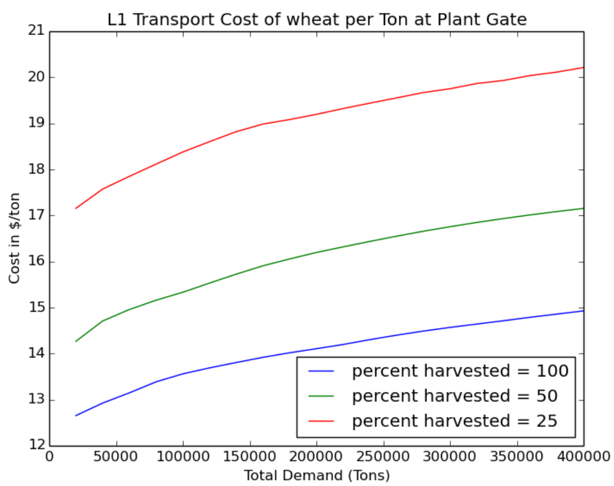

Figure 8: L1 cost for Wheat

References

[1] Mahmood Ebadian (2013). DESIGN AND SCHEDULING OF AGRICULTURAL BIOMASS SUPPLY CHAIN FOR A CELLULOSIC ETHANOL PLANT. THE UNIVERSITY OF BRITISH COLUMBIA, Vancouver, BC.

[2] Brummer, E.C., Burras, C.L., Duffy, M.D., and Moore, K.J. (2000). Switchgrass production in Iowa: economic analysis, soil suitability, and varietal performance. Iowa State University, Ames, Iowa.

[3] Hess, J.R., Kenney, K.., Ovard, L.., Searcy, E.., and Wright, C.. (2009). Commodity-Scale Production of an Infrastructure-Compatible Bulk Solid from Herbaceous Lignocellulosic Biomass. US Department of Energy, Idaho National Lab (INL).

[4] Sokhansanj, S., Turhollow, A.F., and Wilkerson, E. (2008b). Development of the Integrated Biomass Supply Analysis and Logistics Model (IBSAL). Technical Memorandum ORNL/TM-2006/57. Oak Ridge, TN: Oak Ridge National Laboratory.

[5] Sokhansanj, S., and Turhollow, A.F. (2002). Baseline cost for corn stover collection. Applied Engineering in Agriculture 18, 525–532.

[6] Turhollow, A.F., and Sokhansanj, S. (2007). Costs of harvesting, storing in a large pile, and transporting corn stover in a wet form. Applied Engineering in Agriculture 23, 439–448.