10-DECOMPOSE

This module is used to simulate the input, decomposition, and storage of carbon in dead and stable pools within cohorts. It also determines the rate charcoal is buried in soil.

All dead **pools** are named after the corresponding live plant parts with the prefix Dead added. Six pools of dead carbon are considered: 1) the **dead foliage** derived from foliage, 2) **dead fine roots** which can be either in the organic or mineral soil 3) **dead branches** (fine woody debris) derived from branches, 4) **dead sapwood** (one form of coarse woody debris), 5) **dead heartwood** (another form of coarse woody debris), and 6) **dead coarse roots**. For computational purposes dead sapwood and dead heartwood are further subdivided into salvageable versus non-salvagable and snag versus log pools (Figure 10-1).

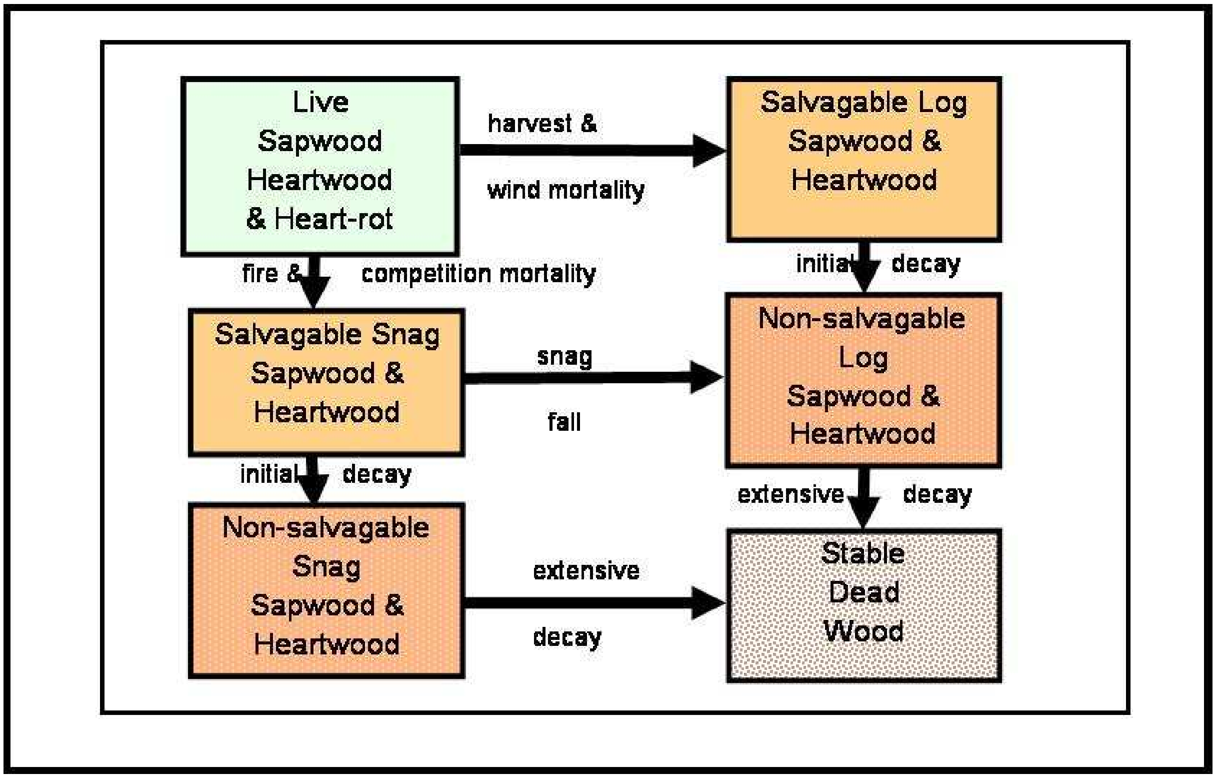

Figure 10-1. Coarse dead wood pools and flows modeled by LANDCARB.

In addition to these dead pools, the model simulates the dynamics of three stable pools (**stable foliage**, **stable wood**, and **stable soil)** that represent extremely decomposed organic matter. Despite the prefix stable, these pools do lose organic matter via decomposition, albeit at a very slow rate.

The burial of charcoal is also tracked by this module, although its formation is computed in PRESCRIBED FIRE and WILDFIRE.

All of the plant layers can input dead parts into the DECOMPOSE module, however, some life forms do not have certain woody parts and therefore do not contribute to the woody dead pools. Herbs are assumed to contribute to the dead foliage and dead fine roots only. Shrubs contribute to the dead foliage, dead fine roots, dead branches, dead sapwood, and dead coarse roots. Finally, trees contribute to all the dead pools including dead foliage, dead fine roots, dead branches, dead sapwood, dead heartwood, and dead coarse roots.

The inputs of material to the dead pools comes from three potential sources: 1) normal litterfall and mortality, 2) harvesting, and 3) fire killed plants. The first input is calculated by MORTALITY, the second by HARVEST, and the third by either PRESCRIBED FIRE or WILDFIRE. Although pools receive inputs from the four plant layers, the input mass is aggregated and the substrate quality is averaged for each pool. The input of material into the stable pools is influenced by the lag required to form this material.

The rate that each dead pool decomposes and the time lag assocatiated with transfers to other pools is determined by the species (as parameterized by the Decomp.prm file), and the climate as calculated by the CLIMATE module. Decomposition rates of the stable pools are determined by the values stored in the DecayPool.prm file as modified by climatic factors calculated by the CLIMATE module.

The stores of dead carbon calculated in the DECOMPOSE module are used by the CLIMATE module to calculate the interception of water by the dead foliage, dead wood, and stable pools. Dead fine root, dead coarse root, and stable soil pools are assumed to not intercept water for water balance purposes.

The files directly used by this module are Decomp.prm and DecayPool.prm.

Dead Pool Input Function.

This function is used to calculate the total input into each of the dead pools. The inputs of material to the dead pools comes from three potential sources: 1) normal litterfall and mortality, 2) harvest and 3) fire. For any given year, the input can come from several of these sources. Each year the inputs from normal litterfall and mortality are calculated first, and then additional inputs from harvest or fire are added. The inputs from harvest and fire are calculated in separate time steps that represent a fraction of a year. This method is used to avoid possible conflicts in calculating the mass of pools.

Inputs to each of the dead pools are calculated as follows where *Pool* represents a specific dead pool, *Layer* represents a plant layer, and *Part* represents a plant part:

For the parts foliage, fine roots, branches, and coarse roots in cells not harvested or burned, the input is :

*LayerPool*Input=*Pool*TurnoverRate**Part* + MortRate**Part* where *LayerPool*Input is the input mass from normal leaf fall, fine root turnover, and pruning for a particular layer and part, *Part* is the mass of the live part being considered, *Pool*TurnoverRate is the fraction of the part for a layer that is pruned or replaced in a given year, and MortRate is the mortality rate of trees. *Pool*TurnoverRate has different names depending upon the plant part considered; FoliageTurnoverRate, FRootTurnoverRate, BranchPruneRate, and CRootPruneRate are used for foliage, fine roots, branches and coarse roots, respectively. All these variables are calculated by the MORTALITY module. MortRate it is also calculated in the MORTALITY module and accounts for the input of non-bole parts associated with the mortality of entire trees.

For the parts sapwood, heartwood, and heartrot, in cohorts not harvested or burned, the input is:

*LayerPool*Input=MortRate**Part*

where *LayerPool*Input in this case is the input mass from sapwood, heartwood, or heartrot of dying trees, *Part* is either sapwood, heartwood, or heartrot mass, and MortRate is the mortality rate of trees calculated in the MORTALITY module.

Plant parts from layers can also be added to dead pools when trees are harvested. These inputs are calculated after normal mortality inputs are calculated in a time step representing a fraction of a year (e.g., 12.2). If the trees in a cohort are harvested then the inputs from foliage, fine roots, branches, sapwood, heartwood, and coarse roots are:

*LayerPool*Input=*Pool*Harv

where *Pool*Harv is the amount of material generated from a layer as input to a dead pool by harvest activities from the HARVEST module.

Finally, plant parts from layers can be added to dead pools when plants are killed by prescribed fire or wildfires. These inputs are calculated after normal mortality inputs and harvest inputs are calculated in a separate time step that represents a fraction of a year (e.g., 12.3). If plants in a cell are killed by fire, then the inputs from foliage, fine roots, branches, sapwood, heartwood, and coarse roots are:

*LayerPool*Input=BurnInput*Pool*

where BurnInput*Pool* is the amount of plant parts killed by fire but not consumed as calculated by either PRESCRIBED FIRE or WILDFIRE.

The total input to a pool (*Pool*Input) at any time step is the sum of all the inputs from the layers in a cell.

*Pool*Input=Σ (*LayerPool*Input)

Dead Substrate Effect Function.

This function calculates the effect of the substrate quality of the various inputs on the overall decomposition rate of a dead pool. This function is invoked each time inputs are added to a cohort. Thus is possible to invoke this function three times in one year if normal growth, harvest, and fire occurs in a year.

The decomposition rate of each pool is dependent on the substrate quality of the inputs to that pool and the current substrate quality of the pool. The overall decomposition rate is a weighted average of the input and current stores. This has the dual effect of building in a system memory but allowing the decomposition rate to gradually change if the substrate quality of the inputs change.

The first step is to calculate the weighted average decomposition rates of the inputs of each pool from the herb, shrub, lower tree, and upper tree layers so that the layers with the largest inputs have the greatest impact on the decomposition rate:

InputDecayRate*Pool*=Σ (*LayerPool*Input*DecayRate*LayerPart*)/*Pool*Input

where InputDecayRate*Pool* is the weighted average decomposition rate of the inputs to a dead pool, *LayerPool*Input is the mass input of each plant part from a specific plant layer (e.g., herbs) to a dead pool, *Pool*Input is the total input of all layers to a pool, and DecayRate*LayerPar*t is the decomposition rate of a part for a layer at 10 C and when moisture is not limiting. For herb and shrub layers the latter parameter is fixed for the entire layer. For trees, however, this parameter is a function of the tree species occupying the particular tree layer. The values of DecayRate*LayerPar*t are the values stored in the Decomp.prm file.

The second step is to calculate the weighted average decomposition rate from the average substrate quality of the inputs and the current material within the dead pool. This step builds in a system memory and allows the decomposition rate of a dead pool to change gradually when the substrate quality of the inputs change. Therefore one can not change the decomposition rate of a dead pool unless the change in substrate quality of the inputs is continued. The weighted average decomposition rate of each dead pool is:

*Pool*DecayRateAvg=(InputDecayRate*Pool***Pool*Input
+Old*Pool*DecayRateAvg**Pool*)/(*Pool*Input+*Pool*)

where *Pool*InputDecayRate and *Pool*Input are as above, Old*Pool*DecayRateAvg is the weighted average decomposition of each dead pool from the past year, and *Pool* is the last year's mass of a particular dead pool. An array of the values from the previous year containing Old*Pool*DecayRateAvg is stored and used to calculate the current value of *Pool*DecayRateAvg.

Position Function.

This function determines if dead wood pool inputs from sapwood, heartwood, and heartrot are added in the "position" of a snag or log. This dichotomy is important in that it determines the water balance of the dead wood and hence the decomposition rate and the time it remains in a salvagable condition.

The proportion of sapwood, heartwood, and heartrot that is added to the snag versus log pool depends on the location of the site, cause of mortality, the age of the trees in a cell, and the tree layer considered:

1). In addition, the proportion of snags dying from normal mortality causes is determined from the location as defined in the Ecoregion.prm file. Most trees die as snags early in stand development, however, the proportion of snags decreases as stands approach the old-growth phase. The proportion of snags also varies with location in the Pacific Northwest, with the the lowest proportion of snags along the wind-prone coastal zone, and the highest in the insect-prone interior zone (Franklin et al. 1987).

2) All upper and lower trees dying from fire, associated with either wildfire or site preparation are added to the snag pools.

3) All woody parts of upper and lower trees that result from cutting operations are added to the log pools.

Lag Emulation Function.

For many decomposition processes there are lags before a transfer can occur. Specifically, the rate dead mass is transfered from salvagable to non-salvagable pools, from snag to log pools, and to stable pools all involves lags. That is, snags do not break into log pieces until a period of decomposition occurs. Likewise, dead wood is merchantable until a large proportion of the material is decayed. In STANDCARB a dead pool cohort data structure is used to introduce these lag time effects. This is a memory intensive approach, and is not used in LANDCARB. Rather than directly track cohorts to create lags, LANDCARB emulates the lag process.

The decomposition lag emulator is based on observations of how these lag effects interact with the amount of input to create deviations in the perceived rate-constant of the transfer of one pool to another. For example, as long as the input is constant, the perceived transfer rate-constant appears to be a constant over time. In contrast, when there is a large pulse of input the transfer rate-constant appears to change. Initially it decreases, because the pool is dominated by new material that has not decomposed sufficiently to be transferred. Eventually as the pulse of input is decomposed, the transfer rate-constant appears to increase and then decrease back to a lower value reflecting a more even input of dead mass. These behaviors are emulated by the following:

1. The transfer rate-constant is initially be set to the average that occurs when inputs to a dead pool are constant. This is specified in the DecayPool.prm.

2. Whenever there is a disturbance in a cohort, a counter is started called the preceeding lags counter. The value of this counter is equal to the lag time (adjusted for the AbioticDecay Index) of the preceeding pools that may have received the disturbance related pulse. If the disturbance adds directly to the pool of interest, then this counter is set to zero. Each subsequent year the counter is decreased by 1. This counter is used to account for the fact that lag for a transfer to occur is dependent on how many steps occur between the time of the disturbance and the transfer of interest.
3. When the preceeding lags counter has reached zero another counter called the disturbance input counter is initiated. The value of this counter is equal to the lag time (adjusted for the AbioticDecay Index) of the transfer of interest. Each subsequent year the counter is decreased by 1. During this time the perceived transfer-rate constant will be set to a level lower than the normal average value.
4. When the disturbance input reaches zero, then increased rates of transfer from the disturbance related input can occur. When the disturbance input counter reaches zero two things happen: a) The disturbance input counter is set to zero until the next disturbance and b) a new counter called the disturbance transfer counter is set to a value that is determined from the rate that cohorts are cleared out as defined in the DecayPool.prm file:

Disturbance Transfer Counter=3/CohortClearingRate

Cohort in this case refers to the decomposition related cohorts used in STANDCARB. In each subsequent year the disturbance transfer counter is reduced by 1. When the disturbance transfer counter is zero, then the transfer rate-constant is returned to the normal average value (see 1 above). Moreover, the disturbance transfer counter is also set to zero until the next time the disturbance input counter becomes positive and reaches zero. When the disturbance transfer counter is greater than zero, the transfer rate-constant will be above that of the normal average value.

The degree that the transfer rate-constant decreased by input from a disturbance depends on the size of the input from disturbance relative to the existing pool size:

TransferRateDuringLag=
transfer rate-constant average *(1-exp(-0.69**Pool*/(*Pool*Input)))

This will mean that if the input is equal to the pool, then the transfer rate-constant will be halved. As the imbalance between inputs and stores increases, the transfer rate-constant approaches zero. If the pool is much greater than the input, then the transfer rate-constant is equal to the average. The inputs and pool sizes are specific to the transfer in question and its related pool. *The input and pool mass considered is just for the year the disturbance input counter is initiated.*

The transfer rate-constant during the period that the disturbance-related input is being transferred is calculated in a series of steps:

MaxTransferRateDuringClearing= CohortClearingRate

This assures that the pulse is cleared out in a time compatible with the disturbance transfer counter. This maximum is adjusted so that the losses from decomposition and transfers does not exceed 100% of the pool size. The total losses from the pool are calculated as:

LossTotal= MaxTransferRateDuringClearing+ *Pool*Decay

Where *Pool*Decay is calculated below and represents the proportion decomposing based on the substrate quality and the climate. If LossTotal exceeds 1 then the MaxTransferRateDuringClearing is adjusted downward:

TransferRateDuringClearingAdjusted= 1- *Pool*Decay

If the LossTotal is equal to or less than 1 then

TransferRateDuringClearingAdjusted= MaxTransferRateDuringClearing

The transfer rate is then adjusted to account for the balance of new inputs versus the mass remaining in the pulse from the disturbance:

TransferReduction=exp [- 0.69*(*Pool*Input)/*Pool*]

When the ratio of inputs to the pool size becomes equal, then the transfer is reduced from that indicated by the cohort clearing rate by 50%. As the input for a period equal to the lag time becomes greater than the pool, then the transfer rate-constant will decrease. The inputs and pool sizes are specific to the transfer in question and its related pool. The input and pool mass is specific to each year that the Disturbance Transfer Counter is working.

The actual transfer rate when the pulse from the disturbance is clearing is:

TransferRateDuringClearing= TransferReduction*MaxTransferRateDuringClearingAdjusted

Lags are averaged based on the lag time for a process for each plant layer present in a cohort. This is a weighted average, weighted by the mass of input versus the existing mass of a dead pool. The input lag time is also a weighted average, but based on the inputs from the different plant layers. This is similar to the weighted average procedure used to calcuate the decomposition rate. The lag times are also adjusted for the effects of climate. The lag time in the Decompose.prm file is an optimum. This optimum is divided by the AbioticDecompositionIndex. Therefore as the AbioticDecompositionIndex decreases, the lag time increases. Lags are rounded up to the nearest year. Lags are all adjusted by the AbioticDecayIndex of the relevant pool except for the lags associated with snags. In the case of snags, the lag is adjusted by either the snag or the coarse root AbioticDecayIndex depending upon which is larger. The reason is that in cases where snag decomposition is low, snags fall to the ground because the roots break after decomposing. Conversely, if root decomposition is slow, then snags break along the stem.

To account for the possibility that disturbances may input a new pulse into dead pools before the decomposition-related lag are exceeded, a lag emulator is created each time a disturbance occurs. The effects of multiple disturbances on apparent lags is simulated by averaging the lag emulators for each disturbance. This average is weighted by the mass remaining in each pulse. When the series of three counters in a lag emulator reach their endpoint, the lag emulator is removed and does not impact the transfer rate calculations.

Dead Decomposition Function.

This function calculates the decomposition rate and mass of dead pools lost from decomposition. Decomposition rate is calculated from the substrate effect (see Dead Substrate Effect function above) and the effects of abiotic factors, temperature, solar radiation warming, and moisture as calculated in the CLIMATE module. The rate of decomposition losses from all the dead pools is:

*Pool*Decay=*Pool*DecayRateAvg**Pool*AnnualAbioticDecayIndex

where *Pool*Decay is the realized decomposition rate of a dead pool, *Pool*DecayRateAvg is the substrate quality determined rate when temperature is 10 C and moisture is not limiting, and *Pool*AnnualAbioticDecayIndex is the combined effects of temperature and moisture on decomposition as calculated in the AbioticDecayIndex function of the CLIMATE module.

The mass of dead lost via decomposition in a year from a pool is :

*Pool*DecayLoss= *Pool*Decay*Pool

where *Pool*DecayLoss is the mass lost via decomposition, *Pool*Decay is the realized decomposition rate of a pool, and *Pool* is the mass of a dead pool cohort.

Salvagable Transfer Function.

This function transfers mass from dead salvageable wood pools to dead non-salvageable wood pools:

SalvagableTransferPool=*Pool*SalvageTransfer Rate**Pool*

where SalvagableTransferPool is the mass transferred to the non-salvagable wood pools and *Pool* is the mass of a contributing salvageable dead wood pool.

Snag Transfer Function.

This function transfers mass from dead snag wood pools to dead log wood pools:

SnagTransferPool=*Pool*SnagTransfer Rate**Pool* where SnagTransferPool is the mass transferred to log pools and *Pool* is the mass of a contributing snag dead wood pool.

Stable Transfer Function.

This function transfers mass from dead pools to stable pools:

StableTransferPool=*Pool*StableTransfer Rate**Pool*

where StableTransferPool is the mass transferred to the stable pool and *Pool* is the mass of a contributing dead pool.

Dead Stores function.

This function is used to calculate the change in the mass of dead pools each year. The balance for each dead pool is the inputs minus the losses from decomposition and transfers to the stable pools. Losses from fire are calculated by the PRESCRIBED FIRE or WILDFIE modules. The overall rate of change for a dead pool is:

Δ*Pool*=*Pool*Input -*Pool*DecayLoss-*Process*Transfer*Pool*

where *Pool*Input is calculated in the Dead Input function, *Pool*DecayLoss is calculated in the Dead Decomposition function, and *Process*Transfer*Pool* can represent salvageable pool, snag, or stable pool-related transfers depending on the dead pool.

The mass in a given dead pool for a given year is therefore:

Pool= Old*Pool* + Δ*Pool*

where Pool is the mass in particular dead pool, Old*Pool* is the value for the previous year, and Δ*Poo*l is as above.

Stable Soil Function.

This function controls the input and decomposition of stable organic matter. These represent three very stable pools in the model and should not change greatly over time unless the forest is removed for extensive periods. These include: 1) StableFoliage, which is derived from dead foliage (similar to O2 horizons), 2) StableWood, which is derived from logs and branches (similar to brown-rotted wood), and StableSoil, which is derived from dead fine and coarse roots (similar to mineral soil organic matter). The intent of this function is to mimic the slow changes in stores in these pools.

Because the decomposition parameters of stable pools are difficult to measure, these parameters may have to be changed for each particular site. We recommend that they be estimated by running the model and comparing to the values expected for each ecosystem being examined. If the inputs of dead pools to the stable pools are constant and the estimated stable pool approximates the store expected for particular site, then the decomposition rates for these pools are probably correct. To see if inputs to the stable pools are constant one can examine the Mort.Dgn file. If the inputs are constant and the stable pool is increasing, then the decomposition rates are probably too low. If the inputs are constant and the stable pool is decreasing, then the decomposition rates are probably too high. The user should change the value of StableDecayRate in the DecayPool.prm file until it converges on the target level of the stable pool being calibrated.

The equation describing the inputs to the stable pools is the sum of all the various inputs from dead pools:

Stable*Pool*Input= Σ StableTransfer*Pool*

where StableTransfer*Pool* is the transfer rate from each of the dead pools as calculated in the Stable Transfer function.

As with the dead pools, the decomposition rate-constant of each stable pool (StablePoolDecay) is a function of the substrate and abiotic effects:

Stable*Pool*Decay=Stable*Pool*DecayRate*Stable*Pool*AnnualAbioticDecayIndex*OldStable*Pool*

where Stable*Pool*AnnualAbioticDecayIndex represents the combined effects of temperature and moisture calculated by CLIMATE, Stable*Pool*DecayRate is the rate of decomposition of a pool at 10 C when moisture conditions are not limiting, and OldStable*Pool* is the mass for the previous year.

The overall rate of change for each pool is:

Δ Stable*Pool*=Stable*Pool*Input-Stable*Pool*Decay

The stores for this pool in a given year are:

Stable*Pool*=OldStable*Pool*+ Δ Stable*Pool*

where OldStable*Pool* is the mass for the previous year and Δ Stable*Pool* is calculated as above.

Total Dead Stores Function.

This function calculates the total mass of dead pools and total mass of stable carbon. These values are reported in the Total.out file.

Charcoal Function.

This function computes mass of charcoal and the rate that charcoal is buried, becoming protected from fire losses. Charcoal is followed in two pools: 1) a surface pool that is created and consumed by fire and 2) a buried pool that transferred from the surface pool and is not subject to losses via fire or decomposition (Figure 10-2). The rate the surface charcoal pool is transferred is determined in the DecayPool.prm file.

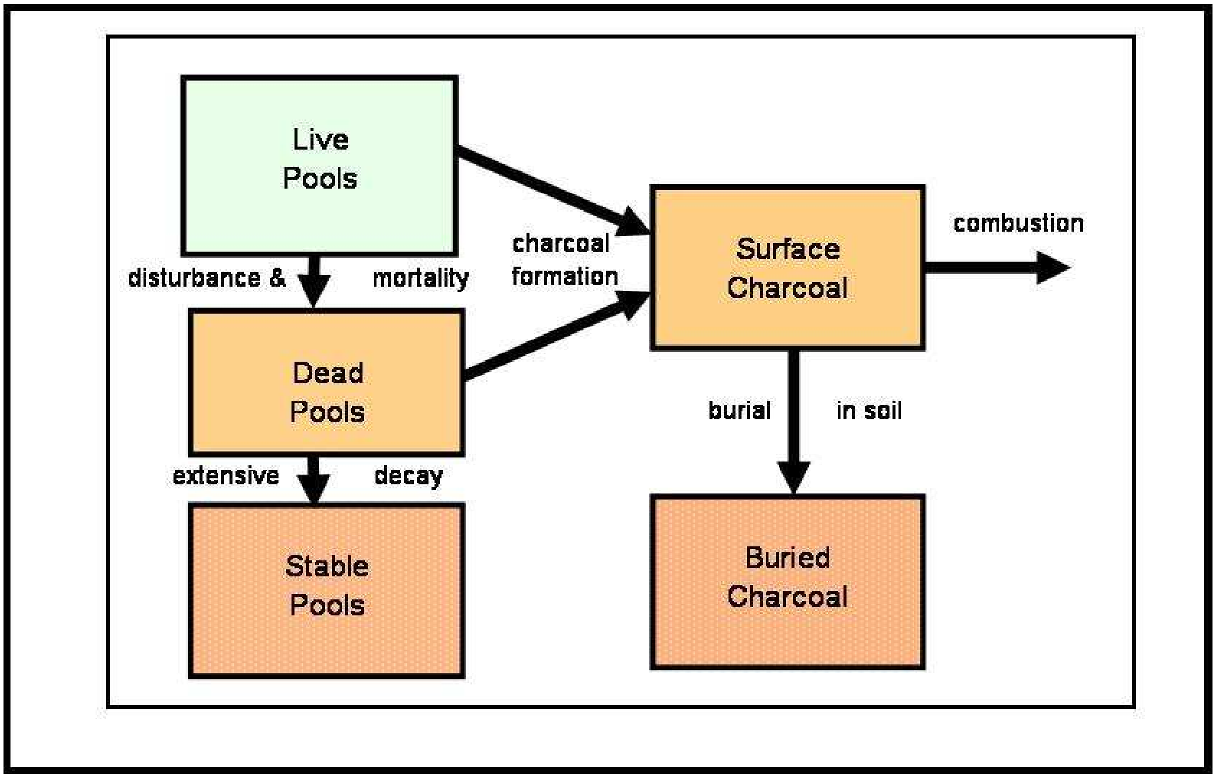

Figure 10-2. Charcoal formation, combustion, and burial used in the LANDCARB model.

The mass of the surface charcoal pool is:

SurfaceCharcoal=SurfaceCharcoalInput+SurfaceCharcoalOld-CharcoalTransfer

Where SurfaceCharcoalInput is the amount of charcoal created from live parts and dead pools in PRESCRIBED FIRE or WILDFIRE, SurfaceCharcoalOld is the pool size the previous year, and CharcoalTransfer is the mass of surface charcoal buried:

CharcoalTransfer= SurfaceCharcoalOld*CharcoalTransferRate

SurfaceCharcoalInput is zero unless a fire occurs, then it is the sum of the charcoal formed from all the live parts and dead pools.

The mass of buried charcoal is:

BuriedCharcoal= CharcoalTransfer+BuriedCharcoalOld

Where BuriedCharcoalOld is the pool size the previous year.