12-PRESCRIBED FIRE

This module determines the effect of prescribed fire on plant layers, dead pools, stable pools, and charcoal pools (Figure 12-1). It may be invoked after a timber harvest or independent of timber harvest. For live vegetation it determines the amount that is killed and consumed by a fire and modifies the amount of live carbon in the GROWTH module and transfers some of this material to the DECOMPOSE module as fire-killed dead pool inputs. Not all the live vegetation killed by fire is necessarily transferred to detritus; some is consumed by the fire itself. For dead pools this module reduces the amount of dead material in the DECOMPOSE module to reflect the losses caused by fire. The changes in live parts, dead and stable pools is described in the Prescribedfire.prm file. The PrescribedFire.prm file is set up so that as fire severity increases from light to hot, the fraction each live part or dead pool killed or removed by fire increases. This module also controls the amount of charcoal this formed from live and dead pools and amount of old charcoal consumed by fires. The rate of charcoal formation is controlled by the CharcoalForm.prm file.

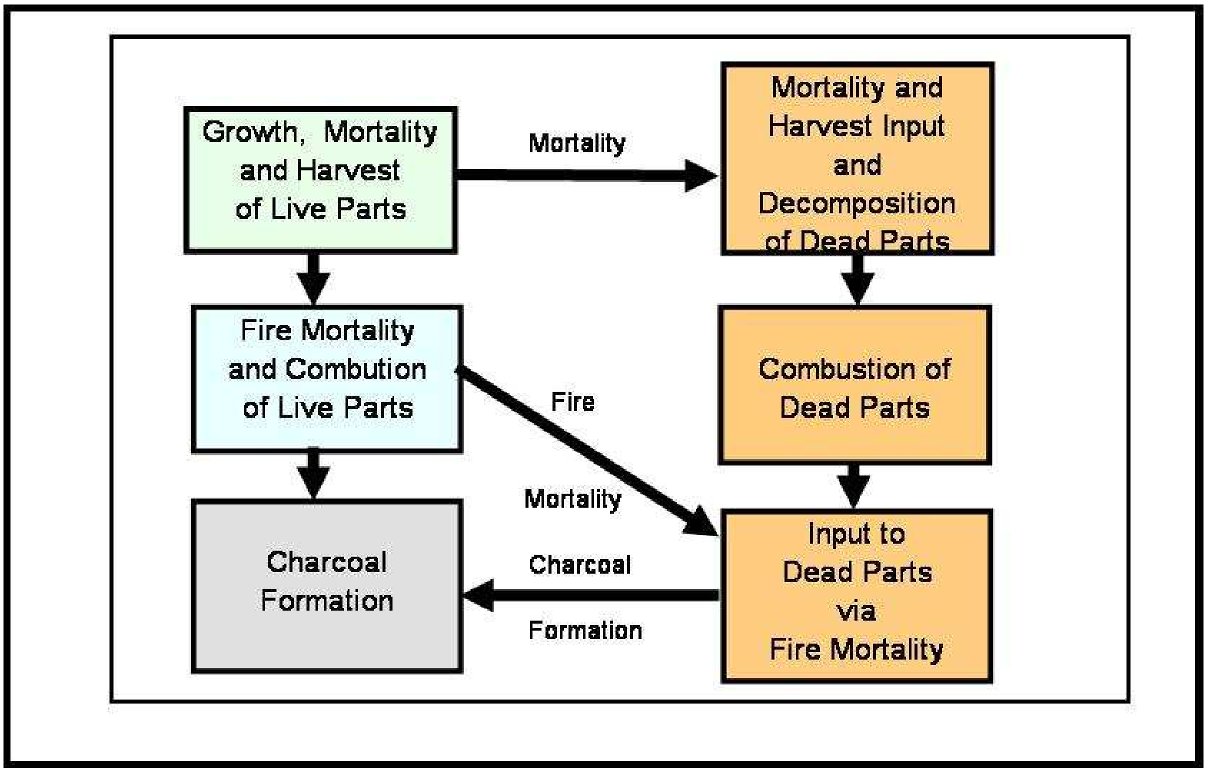

Figure 12-1. Processes occurring in the LANDCARB model when a prescribed fire occurs.

Prescribed fires can either create a new cohort (all the plant layers are killed) or if a new cohort is not created, then how existing cohorts are impacted. Specifically, to mimic a prescribed fire of a stand grid cell it is possible to use one of two options:

1) if plant layers are not completely killed by the prescribed fire, the new cohort area is set to 0, and the fire severity level indicates which plant layers, dead pools, stable pools, and

charcoal pools are influenced and what degree. All cohorts would be effected by this kind of prescribed fire;

2) if all the plant layers are killed by the prescribed fire, then area for a new cohort is created and plant layers are recolonized. Depending on the area of new cohort space created, some of the cohorts will be effected and other will not.

In the latter case the user needs to specify the number of patches the prescribed fire creates. There is a minimum number of patches that a prescribed fire creates. If half the stand is killed, then one needs to have at least two patches. If one-quarter is killed, then one needs at least four patches. The number of patches can be any multiple of the minimum number of patches. The number of patches determines the mean distance between cohorts and the light levels recieved, but does not influence the number of cohorts that are modeled.

LANDCARB uses the concept of a virtual gridwork of substand cells within stand grid cells (Figure 12-2). In STANDCARB this substand gridwok of cells is explicit. The number of cells in the LANDCARB substand gridwork is set by the minimum number of patches implied by the area of new cohort space created by the prescribed fire. If this disturbance pattern is repeated, then in time the number of cohorts in a stand grid cell will be the same as the minimum number of patches. This prevents a situation where the number of cohorts increases without limit with each disturbance cycle.

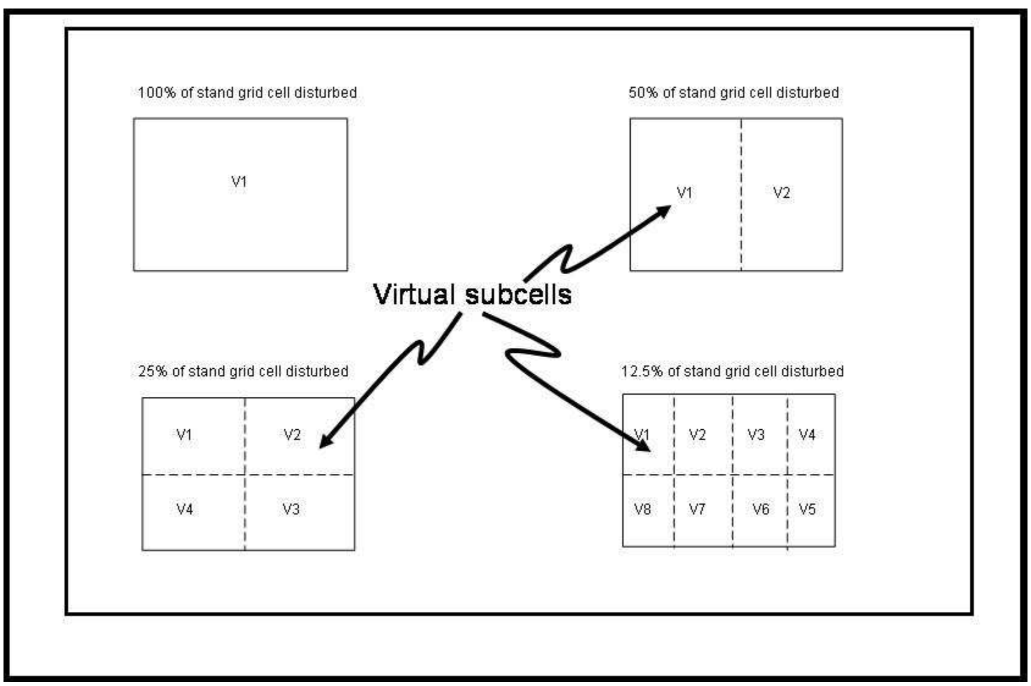

Figure 12-2. Examples of virtual subcells used by the LANDCARB model when a prescribed fire occurs in a stant grid cell.

When a prescribed fires occurs certain of the existing “patches” can be burned. The user sets whether the youngest, oldest, or a random set of patches is burned. If the oldest is selected, then an area equal to the “patch” size is killed by fire creating a new cohort. Suppose one starts with one cohort in a stand grid cell and 25% of the stand grid cell will be burned in 4 patches. The first round of prescribed fires would result in two cohorts; the original cohort would occupy 3 “virtual patches” and the new cohort would occupy 1 “patch”. After the second round of prescribed fires there would be three cohorts, two occupying one “patch” and one occupying two “virtual patches”. With another round of prescribed fires there would be four cohorts, each one occupying one “patch”. Yet another round of prescribed fires and there still would be 4 cohorts each in one “patch”, but the original cohort would have disappeared. In the case where the youngest cohort is selected then there would always be two cohorts, as the youngest is always burned. The first cohort would always occupy 3 “virtual patches” and the youngest one “patch”.

If a harvest has occurred, a site preparation fire is set to occur after the harvest. If the prescribed fire is independent of the harvest, then it occurs the year it is scheduled. Moreover, the calculations occur after normal growth and decomposition calculations have been made for that year. Only one of three types of fires can occur in a given year: light, medium or hot fires.

The parameters controlling this module are found in the PrescribedFire.prm and CharcoalForm.prm files. The user selects the timing and type of prescribed fire in the PrescribedFireScheduler.drv files.

BurnKill function.

This function determines the proportion of each part of each layer remaining after a fire occurs. The first step is to determine when and what type of fire occurs in a stand grid cell. The fraction of the above-ground live mass of a part of a layer surviving (Surv*Part*) after a fire is:

Surv*Part*=(1-AboveKill/100)**Part*

where AboveKill is the percentage of above-ground parts killed by the fire (determined from the BurnKill.prm file), and *Part* is the mass of the above-ground part (i.e., foliage, branches, sapwood, and heartwood) in question. An exception occurs when AboveKill is 100% and it is determined that the tree in the cell sprouts (see Sprout module). In this case *Layer*Foliage is set to the same value as when a layer is planted. This allows the layer to begin growing again in the cell following the fire. These calculations are performed for each of the layers in a cell.

The fraction of the below-ground live mass of a part surviving (Surv*Part*) after a fire is:

Surv*Part*=(1-BelowKill/100)**Part*

where BelowKill is the fraction of below-ground parts killed by the fire (determined from the Prescribed Fire.prm file), and *Part* is the mass of the below-ground part (i.e., fine roots and coarse roots) in question. These calculations are performed for each of the plant layers.

The mass of above- and below-ground parts killed (Kill*Part*) is calculated as:

Kill*Part*=*Part*-Surv*Part*

where *Par*t refers to a specific plant part of a layer in a cell. This quantity is deducted from the live mass of the parts for each layer of each cell in the GROWTH module.

Live Consume Function.

This function calculates the of mass plant parts that is consumed by fire. Above- and below-ground parts have different portions of parts consumed by fire. For above-ground parts the mass consumed (Consum*Part*) is:

Consum*Part*=AboveBurn*Kill*Part*/100

where AboveBurn is the percentage of the above-ground parts that are killed by fire that are combusted (determined from the Prescribed Fire.prm file), and *Part* is the mass of the above-ground part (i.e., foliage, branches, sapwood, and heartwood) in question. These calculations are performed for each of the layers in a cell.

For below-ground parts the mass consumed (Consum*Part*) is:

Consum*Part*=BelowBurn*Kill*Part*/100

where BelowBurn is the percentage of the below-ground parts that are killed by fire that are combusted (determined from the BurnKill.prm file), and *Part* is the mass of the below-ground part (i.e., foliage, branches, sapwood, and heartwood) in question. These calculations are performed for each of the layers in a cell.

The mass of above- and below-ground parts added to the appropriate detrital pool in DECOMPOSE (BurnInput*Pool*) is calculated as:

BurnInput*Pool*=Kill*Part*-Consum*Part*

where *Pool* refers to a specific detrital pool in a cell. This quantity is added to the appropriate detrital pool of each cell in the DECOMPOSE module.

Dead/Stable/Charcoal Consume Function.

This function determines the proportion of each dead, stable, and charcoal pool remaining after a prescribed fire occurs. The first step is to determine when and what type of prescribed fire occurs in a stand grid cell. To calculate the amount removed in each of the pools after fire (*Pool*FireLoss), the fraction remaining is multiplied by the mass of the pool:

*Pool*FireLoss=(1-*Burn*Remaining/100)**Pool*

where *Pool* is the mass of a given detrital pool (i.e., dead foliage, dead branches, dead sapwood, dead heartwood, dead fine roots, dead coarse roots, or stable soil), *Burn*Remaining is the percent remaining after a fire of type *Burn*. The latter parameter is determined from the PrescribedFire.prm file.

If a fire does not occur in a given stand grid cell on a given year then

*Pool*FireLoss=0.

The pool mass is then reduced to account for these fire losses:

*Pool*New=*Pool*Old-*Pool*FireLoss

where *Pool*New and *Pool*Old are the pool mass after and before the fire, respectively.

Charcoal Formation Function.

This function computes the amount of charcoal that is formed by prescribed fire. Live parts, dead pools, and stable pools are all potentially capable of forming charcoal, although it should be noted that non-woody materials are unlikely to form charcoal. The amount of charcoal formed from different parts and pools is varied by fire severity and is defined in the CharcoalFormation.prm file.

The mass of charcoal created from live parts is:

*Part*SurfaceCharcoalInput=CharcoalFormationRate* Kill*Part*/100

The mass of charcoal created from dead and stable pools is:

*Pool*SurfaceCharcoalInput=(*Burn*Remaining/100)**Pool*

The total mass of charcoal formed is:

SurfaceCharcoalInput= Σ*Part*SurfaceCharcoalInput+ Σ*Pool*SurfaceCharcoalInput

Fuel Load

This module creates an overall fuel loading that includes live and dead carbon pools. Each pool is given a weighting based on the relative contribution of a given pool to killing vegetation and/or causing other pools to burn. The store of each pool is multiplied by the weighting factor (see BurnDead.prm) to calculate the pool fuel load:

PoolFuelLoad=FuelWeightingFactor*Pool

The total fuel load is calculated by summing all the pool fuel loads:

FuelLoad= ∑ PoolFuelLoad

The fuel load is a relative index and depends on the weighting factors used. The fuel loads calculated need to be scaled appropriately to the fuel laod limits associated with given levels of fire severity.

Fire Severity-Fuel Load Feedback

Prescribed fire severity may be altered from the target value if the fuel load is lower or higher than the target range appropriate for the target severity (Figure 12-3). When the severity-fuel feedback is switched on in the Simul.drv file, then the fuel load is compared to the range appropriate for the target fire severity. If the fuel load is lower than expected for a given severity, then there is a chance that fire severity will decrease. Likewise, if the fuel load is higher than expected for a given fire severity, then there is a chance the fire severity will increase. The underlying probability of change is set in the BurnDead.prm file. The farther the fuel loading is from the target range, the higher the chance fire severity will change. A similar process is followed for wildfires. However, to account for the fact that weather can be controlled to a higher degree in prescribed fires than wildfires, the probability of changing fire severity should be substantially lower for prescribed fires than wildfires.

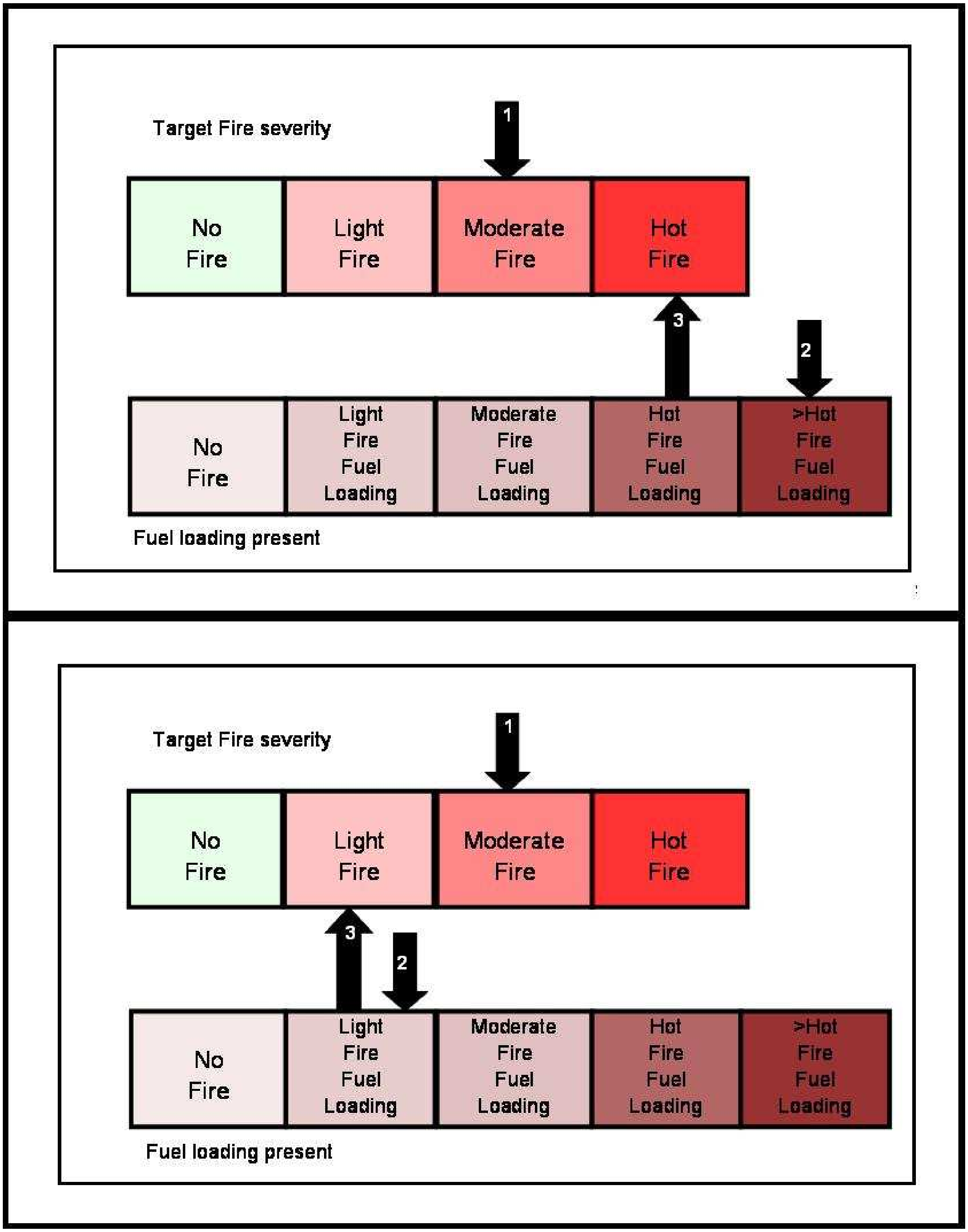

Figure 12-3. Conceptual examples of how the LANDCARB model modifies the realized fire severity from the initial target fire severity. 1- A target fire severity is selected; 2-the fuel loading is compared to the range for the target; and 3- the final fire severity level is adjusted up or down depending on the fuel load.