**Budget Functions**

**I. Introduction**

Given limitation on resources, resource allocation mechanism has to be enforced. In OpenNARS resource allocation mechanism is implemented using Budget Value and Budget Functions. Budget Value is a triplet consisting of *Priority*, *Durability* and *Quality* while Budget Functions are some sort of utility functions that allow keeping the most relevant knowledge the system resources allow.

Main role of Budget Functions is to update Budget Value depending on a specific situation and current state of the system. Budget Value and Budget Functions are defined in *BudgetFunctions.java* and *BudgetValue.java* respectively. It is better to emphasize two states of Budget Value: *Initialization* and *Update*. Initialization happens when new entity is created while update happens during inference cycle.

**II. Initialization**

Budget Value is initialized using predefined set of parameters from *defaultConfig.xml* as well as some other types of initialization

**Task in Global Buffer:**

**Priority:** Default value from Config file depending on type of statement

**Durability:** Default value from Config file depending on type of statement

**Quality:**  where is Truth

Expectation. The intuition is to give higher quality of bigger evidence

**Concept in Main Memory:**

Whenever Task is being selected from Global Buffer, for the first time concept

inherent Budget Value of the Task, that is, Concept Budget Value is being initialized

to Task’s *Priority, Durability* and *Quality*

**TaskLink within concept:**

Budget Value of a TaskLink is initialized with Budget Value of a Concept. That

is after initialization is done, Task, Concept and TaskLink have same values for

*Priority, Durability* and *Quality*

**TermLink within concept:**

Budget Value is partially inherited from Task’s Budget Value, with Priority being

re-adjusted by *distributeAmongLinks()* method in *BudgetFunctions.java*

**Priority:**

**Durability:** Same as Task’s Durability

**Quality:** Same as Task’s Quality

**III. Budget Update**

Budget Value is being updated during inference cycle with application of Budget Functions. Budget Functions are extension of Utility Functions defined in *UtilityFunctions.java*. Budget Functions extensively use *w2c(),* *or(), aveAri(), aveGeo()* and *and()* defined in *UtilityFunctions.java*. Below is the outline of budget functions as they appear in *BudgetFunctions.java*

* **truthToQuality(**Truth Value**)**

Given truth value, get its Truth Expectation and return the following expression:

return

* **revise*(***Truth Value *t,* Truth Value *b,* Truth Value *r****) #*** *NEED show dec of task and belief ?*

Evaluate the quality of a revision rule, then de-prioritize the premises

t – Truth Value of a Task

b – Truth Value of a belief

r – Truth Value after revision rule has been applied

**Priority:** *or(confDiff, t.priority)*

**Durability:** *aveAri (confDiff , t.durability)*

**Quality:** *truthToQuality(r)*

*\** *confDiff is difference between confidences defined as*

* **update(**Current Task, Belief Truth Value**) # NEVER USED**

Given a Task and a Truth Value of a Belief updates Task’s Budget Value

**Priority:** *or(expDiff, Current Tasks Priority)*

**Durability:** *aveAri (expDiff, Current Task Durability)*

**Quality:** *truthToQuality(Belief Truth Value)*

*\*expDiff is difference between truth expectations of a Task and a Belief*

* **activate(**Current Concept, Provided BudgetValue**)**

Activates Current Concept’s Budget Value by an incoming TaskLink’s budget value. In other words adjusts Current Concept’s Budget Value by provided Budget Value.

**Priority:** *or(Current Concept’s Priority, provided Priority)*

**Durability:** *aveAri (Current Concept’s Durability, provided Durability)*

**Quality:** *Current* *Concept’s quality i.e. stays the same*

*\* aveAri() is arithmetic average function defined in UtilityFunctions.java*

* **merge(**BudgetValue b, BudgetValue a**)**

Merging identical entities together, get the max of all Budget Value components

**Priority:** *max(priority of a, priority of b)*

**Durability:** *max(durability of a, durability of b)*

**Quality:** *max(quality of a, quality of b)*

* **applyForgetting(**BudgetValue *bv*, ForgetCycles *c*, relativeThreshold *t***)**

After an item is being used its priority needs to be adjusted, concretely it needs to be decreased to allow fair competition for resources for other items. *applyForgetting()* reduces the priority and allow the item to be forgotten, i.e. removed once priority is less than a certain threshold.

**Priority:**

**Durability:** *Stays the same*

**Quality:** *Stays the same*

* **budgetInference (**Quality *q*, Complexity *c,* Task *t***,** BeliefLink *b***)**

*budgetInference()* function is not a budget function itself, but rather a common processing function that is being called from the following budget functions that are used in backward/forward inferences: *forward(),*

*backward(),*

*backwardWeak(),*

*compoundForward(),*

*compoundBackward(),*

*compoundBackwardWeak()*.

*budgetInference()* function assigns Budget Value to a derivation (conclusion) task, given quality of the inference (Quality *q*), syntactic complexity of the conclusion (Complexity *c*) task (Task *t*) and belief link (BeliefLink *b*):

**Priority:** *or(t.priority, b.priority)*

**Durability:** *and(t.durability / c, b.durability)*

**Quality:** *q/c*

In addition *budgetInference()* function increases Priority and Durability of *b:*

**Priority:** *min(1.0, or(b.priority,or(q/c,b.target.priority ))*

where b.target.priority is priority value of a concept *b* points to

**Durability:** *or(b.durability, q/c)*

* **forward (**TruthValue *truth***,** Task *t***,** BeliefLink *b***)**

Calls*budgetInference()* function with the following args:

***budgetInference (truthToQuality*(*truth*), 1, *t, b*)**

* **backward (**TruthValue *truth***,** Task *t***,** BeliefLink *b***)**

Calls*budgetInference()* function with the following args:

***budgetInference (truthToQuality*(*truth*), 1, *t, b*)**

* **backwardWeak (**TruthValue *truth***,** Task *t***,** BeliefLink *b***)**

Calls*budgetInference()* function with the following args:

***budgetInference (0.5 \* truthToQuality*(*truth*), 1, *t, b*)**

where 0.5 is a value computed by w2c(1, narParameters.HORIZON) and HORIZON is pre-defined parameter to 1

* **compoundForward (**TruthValue *truth***,** Complexity *c*, Task *t***,** BeliefLink *b***)**

Calls*budgetInference()* function with the following args:

***budgetInference ( truthToQuality*(*truth*), *c*, *t, b*)**

* **compoundBackward (**Complexity *c***,** Task *t***,** BeliefLink *b***)**

Calls*budgetInference()* function with the following args:

***budgetInference (*1, *c*, *t, b*)**

* **compoundBackwardWeak (**Complexity *c***,** Task *t***,** BeliefLink *b***)**

Calls*budgetInference()* function with the following args:

***budgetInference (0.5*, *c*, *t, b*)**

where 0.5 is a value computed by w2c(1, narParameters.HORIZON) and HORIZON is pre-defined parameter to 1