User Guide for Habitat\_Opt.gms

Author/Programmer: Austin Milt ([milt@wisc.edu](mailto:milt@wisc.edu))

Last Updated: March 23, 2016

# Intro

Habitat\_Opt.gms is the current GAMS habitat optimization model. At the time of writing, it was not part of the Fishwerks tool, but instead the newest research model. The purpose of this guide is to simplify data formatting for make\_gdx.py. To do so, this guide is a brief summary of the structure of the GAMS model, as well as a glossary of terms.

# Installation

As a general user you shouldn’t need to install anything (because you wont be running Habitat\_Opt.gms). If you wish to run the model, you’ll need to install the latest version of GAMS on your computer. For help, contact the author.

# Model Summary and Structure

The current version of Habitat\_Opt.gms is meant to accept many variations on possible types of models. To accomplish this, the naming conventions and dimensions of model parts are probably beyond your usual needs and as a result some additional work will have to be done up front to format your data correctly.

Habitat\_Opt.gms performs a linear integer programming optimization, where the objective is to maximize a combined benefit over multiple beneficiary and/or control targets by doing projects at or above barriers with a limited budget(s). Barriers included in the optimization may be candidates for one or more projects, or may just be up or downstream of such barriers. Projects may affect either the passability of a barrier or the amount of potential benefit above a barrier for a target.

The schematic below shows how each of the parameters and/or sets specified in your data fit together in the model. Names on the first line in each rounded box are the parameter/set name and on the second line in italics are the sets over which the parameter/set is indexed in the order that they are indexed. These indices need to be reflected in your data definitions. Arrows show the direction data flow, e.g. from one set to another if one is a super set of the other or from data into the model. More information on each parameter/set is given in the glossary after the schematic.

**`**

**Targets**

**Barriers**

**Projects**

**BudgetNames**

**Downstream**

*Barriers, Barriers*

**TargetsBeneficiary**

*Targets*

**TargetsControl**

*Targets*

**ProjectsBenefit**

*Projects*

**ProjectsPassability**

*Projects*

**ProjectToBudget**

*Projects, BudgetNames*

**isCandidate**

*Barriers, Projects*

**isRoot**

*Barriers*

**passBase**

*Barriers, Targets*

**passChange**

*Barriers, Projects, Targets*

**benefitMaxBase**

*Barriers, Targets*

**benefitMaxChange**

*Barriers, Projects, Targets*

**cost**

*Barriers, Projects*

**cap**

*Targets*

**weight**

*Targets*

**budget**

*BudgetNames*

**obj2Weight**

**cumulative passability for each target at each barrier**

**total potential benefit for each target at each barrier**

**total spent on all projects**

*subject to budget constraint*

*subject to benefit max/min*

*maximize weighted benefit for beneficiary targets*

*(secondarily) maximize* weighted *benefit for control targets*

**Input Data Intermediate Calculations Objective & Constraints**

# Glossary

**Barriers** dams and road stream crossing culverts at/above which projects are completed; barriers are always organized in a branching network with each barrier having one downstream barrier and any number of upstream barriers

**benefitMaxBase** equivalent of **passBase** for the baseline potential benefit above the **Barriers** for each of the **Targets**; used to calculate the total potential benefit above a barrier given any **ProjectsBenefit** completed at/above a barrier; at the time of writing, this would be e.g. the HAB\_UP field, i.e. length of river segment above a barrier, in the Fishworks database

**benefitMaxChange** change in potential benefit (**benefitMaxBase**) for each of the **Targets** above each of the **Barriers** when each of the **ProjectsBenefit** are completed; at the time of writing, this would be e.g. quality-weighted habitat adjustments for sea lamprey after lampricide application

**BudgetNames** names of budgets from which projects may draw funding; projects may draw money from only one budget (**ProjectToBudget**), so you should have no more budgets than projects

**budget** designated amount of money available to complete each of the budgets (**BudgetNames**); these are floating points indexed by **BudgetNames**

**cap** for **TargetsControl**, this is a maximum allowed accessibility-weighted benefit; for **TargetsBeneficiary**, this is a minimum allowed accessibility-weighted benefit

**cost** cost of completing each of the **Projects** at/above each of the **Barriers**; used to enforce budget constraints

**Downstream** next downstream barrier from each of the **Barriers**; first index of this set refers to the source barrier; second index of this set refers to the downstream barrier; at the time of writing, convention for designating a root (**isRoot**) barrier as the downstream barrier is ‘-1’

**isCandidate** binary (0/1) designation that a barrier (**Barriers**) is a candidate for a project (**Projects**); barriers can be candidates for any number of **Projects**

**isRoot** binary (0/1) designation that a barrier (**Barriers**) is a root barrier, i.e. has no **Downstream** barriers; this must be specified in addition to the convention described in **Downstream**; root barriers get special treatment in the model since the calculation of cumulative passability for other barriers relies on the presence of one or more **Downstream** barriers

**obj2Weight** weight of the secondary objective in the objective function, which maximizes the weighted sum of accessibility-weighted benefits for **TargetsControl**; the weight of the objective for **TargetsBeneficiary** is always implied to be 1; this is multiplied by the target-specific weights (**weight**); this should generally be positive and small while the **weight** for each of the **TargetsControl** should be negative

**parameter** one kind of data in a GAMS model; these are always treated as numbers, but could be binary; parameters may have anywhere from zero (i.e. a single value) to any number of dimensions; at the time of writing, the parameters with the largest number of dimensions are **benefitMaxChange** and **passChange**, each of which have 3 dimensions

**passBase** barrier-specific passability for each of the **Targets** at each of the **Barriers**; used to calculate the cumulative passability of a barrier given its **Downstream** barriers and any **ProjectsPassability** completed at those barriers; at the time of writing, this would be e.g. the Strong, Moderate, or Weak swimming passability scores in the Fishworks database

**passChange** change in barrier-specific passability (**passBase**) for each of the **Targets** at each of the **Barriers** when each of the **ProjectsPassability** are completed; at the time of writing, this would be e.g. the default assumed change in passability of a barrier after remove (i.e. 1-**passBase**)

**Projects** projects that can be complete at/above a barrier to affect either the passability (**passChange**) of a barrier or the potential benefit (**benefitMaxChange**) above a barrier; projects are broken into two types: **ProjectsPassability** which affect barrier passability and **ProjectsBenefit** which affect the potential benefit above a barrier

**ProjectsBenefit**  one of two types of **Projects**; those projects that affect the total potential benefit above a barrier; these **Projects** modify the baseline benefit values (**benefitMaxChange**) at barriers

**ProjectsPassability** one of two types of **Projects**; those projects that affect the barrier-specific passability for each of the **Targets** at each of the **Barriers**; these **Projects** modify the baseline passability (**passBase**) at barriers

**ProjectToBudget** designated budget name (**BudgetNames**) from which each of the **Projects** can draw funding; first index is the name of the project; second index is the name of the budget to which it belongs

**set** one kind of data in a GAMS model; all sets are collections of names with a common theme, e.g. **Barriers**, **Projects**, **Targets**, etc.; these are always treated as text and never numbers

**scalar** one kind of data in a GAMS model; this is just another name for a zero-dimensional **parameter**

**Targets** target metrics/objects of the optimization; generally these are species, but could be any conservation target; targets are broken into two types: **TargetsBeneficiary** and **TargetsControl**, which are handled differently in the objective function

**TargetsBeneficiary** one of two types of **Targets**; those targets which are the focus of the optimization; generally you should specify a positive **weight** for these targets

**TargetsControl** one of two types of **Targets**; those targets which are optimized secondarily to the **TargetsBeneficiary**; generally you should specify a negative **weight** for these targets

**weight** relative weight of a target in the objective function; weights for **TargetsBeneficiary** should generally be positive and will enter the objective as the value specified; weights for **TargetsContrrol** should generally be negative and are first multiplied by **obj2Weight** as part of the secondary objective