A tour of the GAMS ecosystem in 2023

Recent developments and some evergreens

André Schnabel
OR 2023 - Hamburg

August 30th, 2023

GAMS Software GmbH

Contents

Short introduction into GAMS

Tour of the GAMS ecosystem and its components

- Studio
- Transfer
- Connect
- MIRO
- Engine

Summary

Short introduction into GAMS

What is GAMS?

- Stands for: **G**eneral **A**lgebraic **M**odeling **S**ystem

What is GAMS?

- Stands for: **G**eneral **A**lgebraic **M**odeling **S**ystem
- Conceived 1976 by World Bank, commercial since 1987

What is GAMS?

- Stands for: **G**eneral **A**lgebraic **M**odeling **S**ystem
- Conceived 1976 by World Bank, commercial since 1987
- Application areas wherever mathematical optimization is useful

What is GAMS?

- Stands for: **G**eneral **A**lgebraic **M**odeling **S**ystem
- Conceived 1976 by World Bank, commercial since 1987
- Application areas wherever mathematical optimization is useful
- Many users from academic and commercial institutions

What is GAMS?

- Stands for: **G**eneral **A**lgebraic **M**odeling **S**ystem
- Conceived 1976 by World Bank, commercial since 1987
- Application areas wherever mathematical optimization is useful
- Many users from academic and commercial institutions
- Extensive software package with mathematical modeling language at its core

The GAMS language

- Focused on declarative programming resembling mathematical notation

The GAMS language

- Focused on declarative programming resembling mathematical notation
- Central elements/keywords are: sets, parameters, variables, equations, and models

The GAMS language

- Focused on declarative programming resembling mathematical notation
- Central elements/keywords are: sets, parameters, variables, equations, and models
- Bridge between human and machine readable/executable model representation

The GAMS language

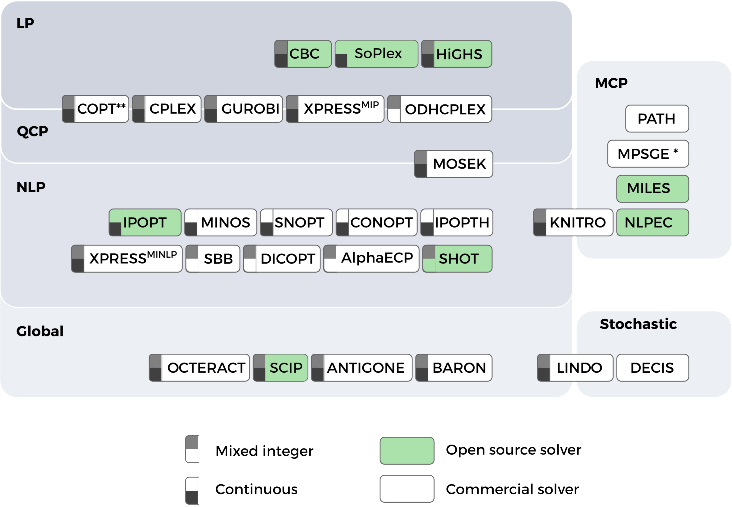
- Focused on declarative programming resembling mathematical notation
- Central elements/keywords are: sets, parameters, variables, equations, and models
- Bridge between human and machine readable/executable model representation
- Imperative language constructs like branching and loops

The GAMS language

- Focused on declarative programming resembling mathematical notation
- Central elements/keywords are: sets, parameters, variables, equations, and models
- Bridge between human and machine readable/executable model representation
- Imperative language constructs like branching and loops

- Flexibility, simplicity, and speed as key strengths of language and ecosystem with
  - 16 expressible model types: linear, quadratic, nonlinear, stochastic, ...
  - 37 solver links: SCIP, Gurobi, HiGHS, KNITRO, ...
  - 3 operating systems: Windows, Linux, macOS (x64 & ARM64)
  - Many data sources and sinks: CSV, GDX, HAR, pandas, Excel, Access, SQL, ...
  - 13 language bindings: C/C++, Python, MATLAB, R, Julia, .NET, Java, ...
  - Efficient model generation for sparse coefficient matrices (widespread "in the wild")
    → see "Performance in Optimization Models" at `https://www.gams.com/blog`

GAMS solver zoo and supported model types



LP

CBC SoPlex HiGHS

COPT** CPLEX GUROBI XPRESS^{MIP} ODHCPLEX

QCP

MOSEK

NLP

IPOPT MINOS SNOPT CONOPT IPOPTH

XPRESS^{MINLP} SBB DICOPT AlphaECP SHOT

Global

OCTERACT SCIP ANTIGONE BARON

MCP

PATH

MPSGE *

MILES

KNITRO NLPEC

Stochastic

LINDO DECIS

Mixed integer

Continuous

🟩 Open source solver

□ Commercial solver

* MPSGE is not a solver, but a GAMS subsystem dedicated to solving economic equillibrium models

** COPT does not handle MIQCPs

Knapsack problem definition and implementation

Backpack can hold weight c.
Items $i \in \mathcal{I}$ with utilities u_i and weights w_i are available to select.

$$x_i = \begin{cases} 1, & \text{iff. item } i \text{ is selected} \\ 0, & \text{otherwise.} \end{cases}$$

$$\max \sum_{i \in \mathcal{I}} u_i \cdot x_i$$

s.t.

$$\sum_{i \in \mathcal{I}} w_i \cdot x_i \leq c$$

$$x_i \in \{0, 1\} \qquad\qquad i \in \mathcal{I}$$

```
set i "available items" / i1*i4 /;

scalar c "capacity of knapsack" / 8 /

parameters
u(i) "utility of item" / i1 10, i2 4, i3 5, i4 8 /
w(i) "weight of item" / i1  8, i2 2, i3 4, i4 5 /;

binary variable
x(i) "1 iff. item i is deposited into knapsack";

free variable
utility "objective value of selection";

equations obj, cap;
obj .. utility =e= sum(i, u(i)*x(i));
cap .. sum(i, w(i)*x(i)) =l= c;

model knapsack /all/;
solve knapsack using mip maximizing utility;
display x.l;
```

Listing 1: knapsack.gms

Tour of the GAMS ecosystem and its components

Tour of the GAMS ecosystem and its components

Studio

GAMS Studio

- Integrated development environment for GAMS code

GAMS Studio

- Integrated development environment for GAMS code
- Actively developed modern replacement of deprecated GAMS IDE

GAMS Studio

- Integrated development environment for GAMS code
- Actively developed modern replacement of deprecated GAMS IDE
- Open source on GitHub: `https://github.com/GAMS-dev/studio`

GAMS Studio

- Integrated development environment for GAMS code
- Actively developed modern replacement of deprecated GAMS IDE
- Open source on GitHub: `https://github.com/GAMS-dev/studio`
- Runs natively on Windows, Linux, and macOS (x64 & ARM64)

GAMS Studio

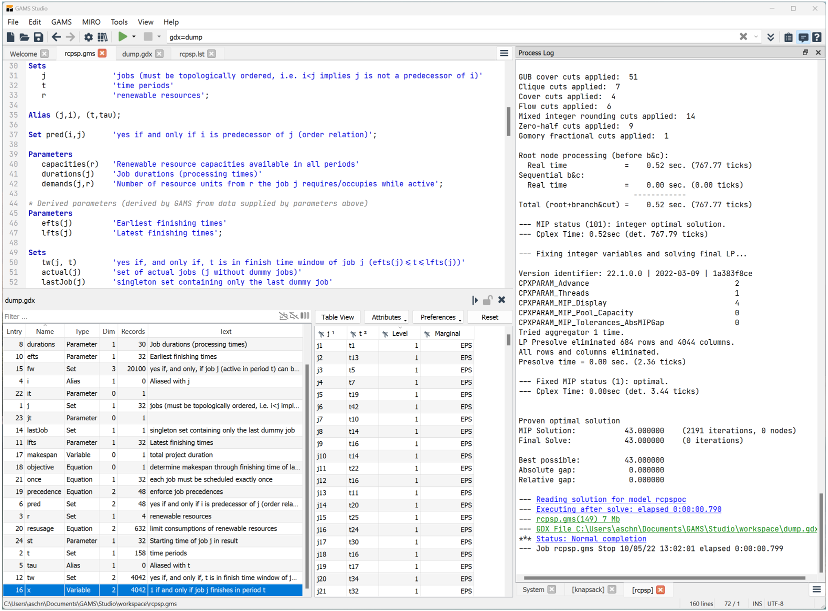
- Integrated development environment for GAMS code
- Actively developed modern replacement of deprecated GAMS IDE
- Open source on GitHub: `https://github.com/GAMS-dev/studio`
- Runs natively on Windows, Linux, and macOS (x64 & ARM64)
- Relatively lightweight in terms of CPU and memory consumption

GAMS Studio

- Integrated development environment for GAMS code
- Actively developed modern replacement of deprecated GAMS IDE
- Open source on GitHub: `https://github.com/GAMS-dev/studio`
- Runs natively on Windows, Linux, and macOS (x64 & ARM64)
- Relatively lightweight in terms of CPU and memory consumption

- Numerous features to help GAMS users
  - Handling of multiple file projects
  - Syntax highlighting, code completion, tooltips, help integration
  - Code folding, dark mode, and distraction free mode
  - Interactive debugger with breakpoints and stepping
  - GDX (**G**AMS **D**ata e**X**change), listing file, and reference file viewers
  - Comfortable editor for GAMS configuration file
  - Integration of Engine, MIRO, and NEOS server

User interface of GAMS Studio



```
Sets
   j               'jobs (must be topologically ordered, i.e. i<j implies j is not a predecessor of i)'
   t               'time periods'
   r               'renewable resources';

Alias (j,i), (t,tau);

Set pred(i,j)      'yes if and only if i is predecessor of j (order relation)';

Parameters
   capacities(r)   'Renewable resource capacities available in all periods'
   durations(j)    'Job durations (processing times)'
   demands(j,r)    'Number of resource units from r the job j requires/occupies while active';

* Derived parameters (derived by GAMS from data supplied by parameters above)
Parameters
   efts(j)         'Earliest finishing times'
   lfts(j)         'Latest finishing times';

Sets
   tw(j, t)        'yes if, and only if, t is in finish time window of job j (efts(j)≤t≤lfts(j))'
   actual(j)       'set of actual jobs (j without dummy jobs)'
   lastJob(j)      'singleton set containing only the last dummy job'
```

dump.gdx

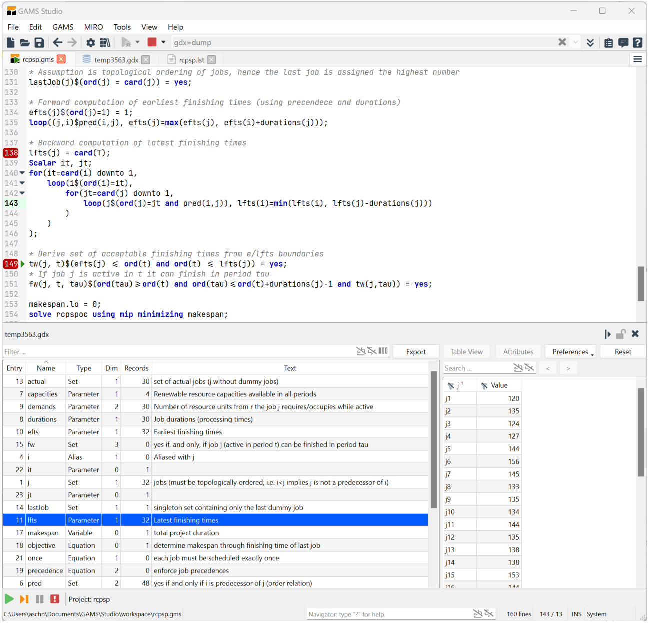
Entry	Name	Type	Dim	Records	Text
8	durations	Parameter	1	30	Job durations (processing times)
10	efts	Parameter	1	32	Earliest finishing times
15	fw	Set	3	20100	yes if, and only, if job j (active in period t) can b...
4	i	Alias	1	0	Aliased with j
22	it	Parameter	0	1	
1	j	Set	1	32	jobs (must be topologically ordered, i.e. i<j impl...
23	jt	Parameter	0	1	
14	lastJob	Set	1	1	singleton set containing only the last dummy job
11	lfts	Parameter	1	32	Latest finishing times
17	makespan	Variable	0	1	total project duration
18	objective	Equation	0	1	determine makespan through finishing time of la...
21	once	Equation	1	32	each job must be scheduled exactly once
19	precedence	Equation	2	48	enforce job precedences
6	pred	Set	2	48	yes if and only if i is predecessor of j (order rela...
3	r	Set	1	4	renewable resources
20	resusage	Equation	2	632	limit consumptions of renewable resources
24	st	Parameter	1	32	Starting time of job j in result
2	t	Set	1	158	time periods
5	tau	Alias	1	0	Aliased with t
12	tw	Set	2	4042	yes if, and only if, t is in finish time window of j...
16	x	Variable	2	4042	1 if and only if job j finishes in period t

j 1	t 2	Level	Marginal
j1	t1	1	EPS
j2	t13	1	EPS
j3	t5	1	EPS
j4	t7	1	EPS
j5	t19	1	EPS
j6	t42	1	EPS
j7	t10	1	EPS
j8	t14	1	EPS
j9	t16	1	EPS
j10	t14	1	EPS
j11	t22	1	EPS
j12	t16	1	EPS
j13	t11	1	EPS
j14	t20	1	EPS
j15	t25	1	EPS
j16	t24	1	EPS
j17	t30	1	EPS
j18	t16	1	EPS
j19	t17	1	EPS
j20	t34	1	EPS
j21	t32	1	EPS

C:\Users\aschn\Documents\GAMS\Studio\workspace\rcpsp.gms

```
GUB cover cuts applied:  51
Clique cuts applied:  7
Cover cuts applied:  4
Flow cuts applied:  6
Mixed integer rounding cuts applied:  14
Zero-half cuts applied:  9
Gomory fractional cuts applied:  1

Root node processing (before b&c):
  Real time             =    0.52 sec. (767.77 ticks)
Sequential b&c:
  Real time             =    0.00 sec. (0.00 ticks)
                          ------------
Total (root+branch&cut) =    0.52 sec. (767.77 ticks)

--- MIP status (101): integer optimal solution.
--- Cplex Time: 0.52sec (det. 767.79 ticks)

--- Fixing integer variables and solving final LP...

Version identifier: 22.1.0.0 | 2022-03-09 | 1a383f8ce
CPXPARAM_Advance                                 2
CPXPARAM_Threads                                 1
CPXPARAM_MIP_Display                             4
CPXPARAM_MIP_Pool_Capacity                       0
CPXPARAM_MIP_Tolerances_AbsMIPGap                0
Tried aggregator 1 time.
LP Presolve eliminated 684 rows and 4044 columns.
All rows and columns eliminated.
Presolve time = 0.00 sec. (2.36 ticks)

--- Fixed MIP status (1): optimal.
--- Cplex Time: 0.00sec (det. 3.44 ticks)


Proven optimal solution
MIP Solution:           43.000000    (2191 iterations, 0 nodes)
Final Solve:            43.000000    (0 iterations)

Best possible:          43.000000
Absolute gap:            0.000000
Relative gap:            0.000000

--- Reading solution for model rcpspoc
--- Executing after solve: elapsed 0:00:00.790
--- rcpsp.gms(149) 7 Mb
--- GDX File C:\Users\aschn\Documents\GAMS\Studio\workspace\dump.gd>
*** Status: Normal completion
--- Job rcpsp.gms Stop 10/05/22 13:02:01 elapsed 0:00:00.799
```

Interactive debugging with GAMS Studio



```
* Assumption is topological ordering of jobs, hence the last job is assigned the highest number
lastJob(j)$(ord(j) = card(j)) = yes;

* Forward computation of earliest finishing times (using precendece and durations)
efts(j)$(ord(j)=1) = 1;
loop((j,i)$pred(i,j), efts(j)=max(efts(j), efts(i)+durations(j)));

* Backward computation of latest finishing times
lfts(j) = card(T);
Scalar it, jt;
for(it=card(i) downto 1,
    loop(i$(ord(i)=it),
        for(jt=card(j) downto 1,
            loop(j$(ord(j)=jt and pred(i,j)), lfts(i)=min(lfts(i), lfts(j)-durations(j)))
        )
    )
);

* Derive set of acceptable finishing times from e/lfts boundaries
tw(j, t)$(efts(j) ≤ ord(t) and ord(t) ≤ lfts(j)) = yes;
* If job j is active in t it can finish in period tau
fw(j, t, tau)$(ord(tau)≥ord(t) and ord(tau)≤ord(t)+durations(j)-1 and tw(j,tau)) = yes;

makespan.lo = 0;
solve rcpspoc using mip minimizing makespan;
```

temp3563.gdx

Entry	Name	Type	Dim	Records	Text
13	actual	Set	1	30	set of actual jobs (j without dummy jobs)
7	capacities	Parameter	1	4	Renewable resource capacities available in all periods
9	demands	Parameter	2	30	Number of resource units from r the job j requires/occupies while active
8	durations	Parameter	1	30	Job durations (processing times)
10	efts	Parameter	1	32	Earliest finishing times
15	fw	Set	3	0	yes if, and only, if job j (active in period t) can be finished in period tau
4	i	Alias	1	0	Aliased with j
22	it	Parameter	0	1	
1	j	Set	1	32	jobs (must be topologically ordered, i.e. i<j implies j is not a predecessor of i)
23	jt	Parameter	0	1	
14	lastJob	Set	1	1	singleton set containing only the last dummy job
11	lfts	Parameter	1	32	Latest finishing times
17	makespan	Variable	0	1	total project duration
18	objective	Equation	0	1	determine makespan through finishing time of last job
21	once	Equation	1	0	each job must be scheduled exactly once
19	precedence	Equation	2	0	enforce job precedences
6	pred	Set	2	48	yes if and only if i is predecessor of j (order relation)

j	Value
j1	120
j2	135
j3	124
j4	127
j5	144
j6	156
j7	145
j8	133
j9	135
j10	134
j11	144
j12	135
j13	138
j14	138
j15	153

Project: rcpsp

C:\Users\aschn\Documents\GAMS\Studio\workspace\rcpsp.gms

Tour of the GAMS ecosystem and its components

Transfer

GAMS Transfer

- An optimization model arises inside a machine learning or data science project?

→ API for data exchange with scripting language (Python, MATLAB, R)

- *Container*-oriented: Holds (linked) symbols with attributes and data (records)
- Symbol records have standardized format:
  - Python: Pandas DataFrame, numpy arrays
  - MATLAB: struct, table, dense matrix, sparse matrix
  - R: vector, data frame
- Batch read/writes through container (fast C++ backend under the hood)

```python
import gamstransfer as gt
import numpy as np
m = gt.Container()
items = [f'i{i+1}' for i in range(4)]
us = np.array([10, 4, 5, 8])
ws = np.array([8, 2, 4, 5])
i = m.addSet('i', records=items, description='items')
c = m.addParameter('c', records=8, description='capacity')
u = m.addParameter('u', domain=i, records=us, description='utility')
w = m.addParameter('w', domain=i, records=ws, description='weight')
m.write('knapsack_instance.gdx')
```

Listing 2: knapsack_transfer.gms

Tour of the GAMS ecosystem and its components

Connect

GAMS Connect: extract, transform, load (ETL) for GAMS

- Processing data in arbitrary formats in a more descriptive way?

GAMS Connect: extract, transform, load (ETL) for GAMS

- Processing data in arbitrary formats in a more descriptive way?

→ Flexible system to read, transform and write data from various formats

GAMS Connect: extract, transform, load (ETL) for GAMS

- Processing data in arbitrary formats in a more descriptive way?

→ Flexible system to read, transform and write data from various formats

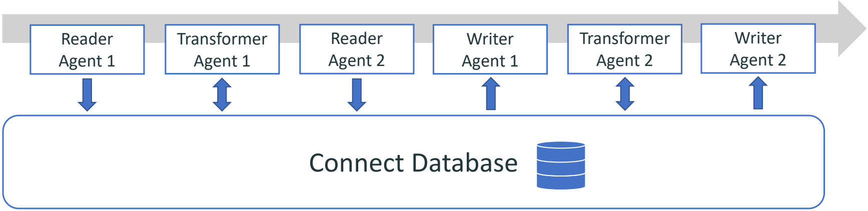
- Available source/sink formats: CSV, SQL, Excel, and more to come...

GAMS Connect: extract, transform, load (ETL) for GAMS

- Processing data in arbitrary formats in a more descriptive way?

→ Flexible system to read, transform and write data from various formats

- Available source/sink formats: CSV, SQL, Excel, and more to come...
- Will replace most data exchange command line utilities (e.g. GDXXRW for Excel)



Knapsack example with Connect

```
i,   u,  w
i1, 10,  8
i2,  4,  2
i3,  5,  4
i4,  8,  5
```

Listing 3: item_data.csv

```
$onEmbeddedCode Connect:
- CSVReader:
    file: item_data.csv
    name: u
    indexColumns: 1
    valueColumns: 2
- CSVReader:
    file: item_data.csv
    name: w
    indexColumns: 1
    valueColumns: 3
- GAMSWriter:
    writeAll: True
$offEmbeddedCode
parameter c /8/;
```

Listing 4: knapsack_connect.gms

Harzer Wandernadel - excourse

- Over 200 stamp stations scattered across the Harz mountains in Germany

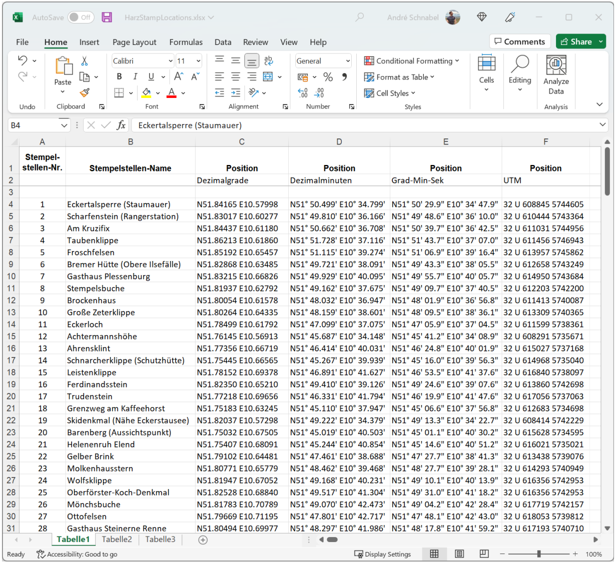
Harzer Wandernadel - excourse

- Over 200 stamp stations scattered across the Harz mountains in Germany
- Put stamp into your pass to “prove” visit during hike

Harzer Wandernadel - excourse

- Over 200 stamp stations scattered across the Harz mountains in Germany
- Put stamp into your pass to “prove” visit during hike
- More information available online:
  - `https://www.harzer-wandernadel.de/`
  - `https://www.oberharzinfo.de/en/hiking-nature/harzer-wandernadel/harzer-wandernadel-up-in-the-harz`

Harzer Wandernadel - Excel input data set



B4 | Eckertalsperre (Staumauer)

	A	B	C	D	E	F
1	**Stempel-stellen-Nr.**	**Stempelstellen-Name**	**Position**	**Position**	**Position**	**Position**
2			Dezimalgrade	Dezimalminuten	Grad-Min-Sek	UTM
3						
4	1	Eckertalsperre (Staumauer)	N51.84165 E10.57998	N51° 50.499' E10° 34.799'	N51° 50' 29.9" E10° 34' 47.9"	32 U 608845 5744605
5	2	Scharfenstein (Rangerstation)	N51.83017 E10.60277	N51° 49.810' E10° 36.166'	N51° 49' 48.6" E10° 36' 10.0"	32 U 610444 5743364
6	3	Am Kruzifix	N51.84437 E10.61180	N51° 50.662' E10° 36.708'	N51° 50' 39.7" E10° 36' 42.5"	32 U 611031 5744956
7	4	Taubenklippe	N51.86213 E10.61860	N51° 51.728' E10° 37.116'	N51° 51' 43.7" E10° 37' 07.0"	32 U 611456 5746943
8	5	Froschfelsen	N51.85192 E10.65457	N51° 51.115' E10° 39.274'	N51° 51' 06.9" E10° 39' 16.4"	32 U 613957 5745862
9	6	Bremer Hütte (Obere Ilsefälle)	N51.82868 E10.63485	N51° 49.721' E10° 38.091'	N51° 49' 43.3" E10° 38' 05.5"	32 U 612658 5743249
10	7	Gasthaus Plessenburg	N51.83215 E10.66826	N51° 49.929' E10° 40.095'	N51° 49' 55.7" E10° 40' 05.7"	32 U 614950 5743684
11	8	Stempelsbuche	N51.81937 E10.62792	N51° 49.162' E10° 37.675'	N51° 49' 09.7" E10° 37' 40.5"	32 U 612203 5742200
12	9	Brockenhaus	N51.80054 E10.61578	N51° 48.032' E10° 36.947'	N51° 48' 01.9" E10° 36' 56.8"	32 U 611413 5740087
13	10	Große Zeterklippe	N51.80264 E10.64335	N51° 48.159' E10° 38.601'	N51° 48' 09.5" E10° 38' 36.1"	32 U 613309 5740365
14	11	Eckerloch	N51.78499 E10.61792	N51° 47.099' E10° 37.075'	N51° 47' 05.9" E10° 37' 04.5"	32 U 611599 5738361
15	12	Achtermannshöhe	N51.76145 E10.56913	N51° 45.687' E10° 34.148'	N51° 45' 41.2" E10° 34' 08.9"	32 U 608291 5735671
16	13	Ahrensklint	N51.77356 E10.66719	N51° 46.414' E10° 40.031'	N51° 46' 24.8" E10° 40' 01.9"	32 U 615027 5737168
17	14	Schnarcherklippe (Schutzhütte)	N51.75445 E10.66565	N51° 45.267' E10° 39.939'	N51° 45' 16.0" E10° 39' 56.3"	32 U 614968 5735040
18	15	Leistenklippe	N51.78152 E10.69378	N51° 46.891' E10° 41.627'	N51° 46' 53.5" E10° 41' 37.6"	32 U 616840 5738097
19	16	Ferdinandsstein	N51.82350 E10.65210	N51° 49.410' E10° 39.126'	N51° 49' 24.6" E10° 39' 07.6"	32 U 613860 5742698
20	17	Trudenstein	N51.77218 E10.69656	N51° 46.331' E10° 41.794'	N51° 46' 19.9" E10° 41' 47.6"	32 U 617056 5737063
21	18	Grenzweg am Kaffeehorst	N51.75183 E10.63245	N51° 45.110' E10° 37.947'	N51° 45' 06.6" E10° 37' 56.8"	32 U 612683 5734698
22	19	Skidenkmal (Nähe Eckerstausee)	N51.82037 E10.57298	N51° 49.222' E10° 34.379'	N51° 49' 13.3" E10° 34' 22.7"	32 U 608414 5742229
23	20	Barenberg (Aussichtspunkt)	N51.75032 E10.67505	N51° 45.019' E10° 40.503'	N51° 45' 01.1" E10° 40' 30.2"	32 U 615628 5734595
24	21	Helenenruh Elend	N51.75407 E10.68091	N51° 45.244' E10° 40.854'	N51° 45' 14.6" E10° 40' 51.2"	32 U 616021 5735021
25	22	Gelber Brink	N51.79102 E10.64481	N51° 47.461' E10° 38.688'	N51° 47' 27.7" E10° 38' 41.3"	32 U 613438 5739076
26	23	Molkenhausstern	N51.80771 E10.65779	N51° 48.462' E10° 39.468'	N51° 48' 27.7" E10° 39' 28.1"	32 U 614293 5740949
27	24	Wolfsklippe	N51.81947 E10.67052	N51° 49.168' E10° 40.231'	N51° 49' 10.1" E10° 40' 13.9"	32 U 616356 5742953
28	25	Oberförster-Koch-Denkmal	N51.82528 E10.68840	N51° 49.517' E10° 41.304'	N51° 49' 31.0" E10° 41' 18.2"	32 U 616356 5742953
29	26	Mönchsbuche	N51.81783 E10.70789	N51° 49.070' E10° 42.473'	N51° 49' 04.2" E10° 42' 28.4"	32 U 617719 5742157
30	27	Ottofelsen	N51.79669 E10.71195	N51° 47.801' E10° 42.717'	N51° 47' 48.1" E10° 42' 43.0"	32 U 618053 5739812
31	28	Gasthaus Steinerne Renne	N51.80494 E10.69977	N51° 48.297' E10° 41.986'	N51° 48' 17.8" E10° 41' 59.2"	32 U 617193 5740710

Source: `https://www.harzer-wandernadel.de/stempelstellen/gps-download/`

Harzer Wandernadel - Specify table layout for Connect

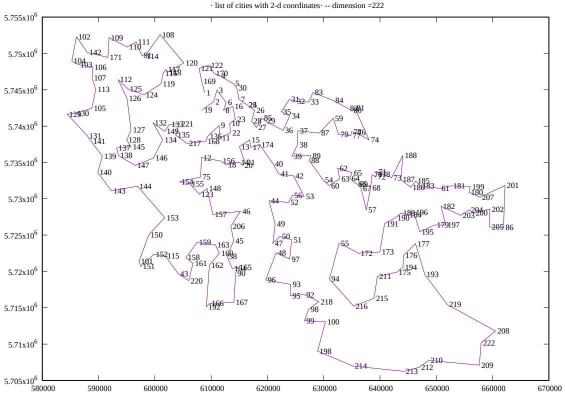
```
$onEmbeddedCode Connect:
- PandasExcelReader:
     file: HarzStampLocations.xlsx
     symbols:
         - name: stamplocations
           range: "Tabelle1!A3"
           type: set
           rowDimension: 6
           columnDimension: 0
- Projection:
     name: stamplocations(nr,name,position1,position2,position3,position4)
     newName: i(nr)
     text: "{name}"
- Projection:
     name: stamplocations(nr,name,position1,position2,position3,position4)
     newName: utmraw(nr)
     text: "{position4}"
- PythonCode:
     code: |
       # Split UTM of format "32 U eeeee nnnnn" into parameter utm
       utm_records = []
       for r in connect.container["utmraw"].records.values:
          s = r[1].split()
          utm_records.append( (r[0], "east", float(s[-2])) )
          utm_records.append( (r[0], "north", float(s[-1])) )
       connect.container.addParameter("utm", domain=["*","*"], records=utm_records)
- GAMSWriter:
     symbols:
        - name: i
        - name: utm
$offEmbeddedCode
```

Listing 5: stampstations.gms

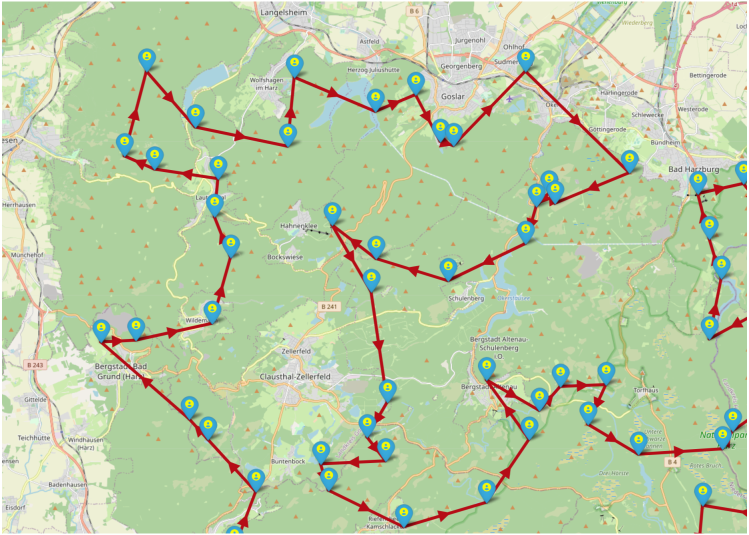
Harzer Wandernadel - full optimal solution

- Local newspaper: Athlete visited all in 3 weeks
- Data fed into TSP model from GAMS Model Library
  - 2 minutes with CPLEX
  - 2.4 seconds with Concorde (coordinates as text)
- Optimal tour length: 557 km (Euclidean distance between nodes)

→ ≥ 26 km per day for Athlete



Harzer Wandernadel - MIRO app and solution map



Try interactively: `https://miro.gams.com/gallery/app_direct/tsp/`

Tour of the GAMS ecosystem and its components

MIRO

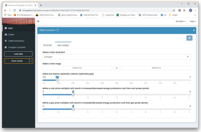
Model Interface with Rapid Orchestration (MIRO)

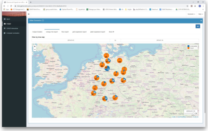
Configuration vs coding

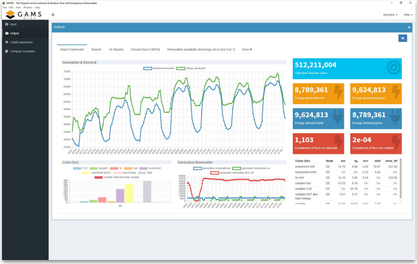
- A few model annotations make GAMS model MIRO ready
- Widgets and graphs can be configured, but do not need to be programmed

Benefits

- Very quick results
- Extendable with custom R code






DIETER - The Dispatch and Investment Evaluation Tool with Endogenous Renewables

512,211,004 Objective function value

8,789,361 Energy generated net

9,624,813 Energy generated gross

9,624,813 Energy demand total

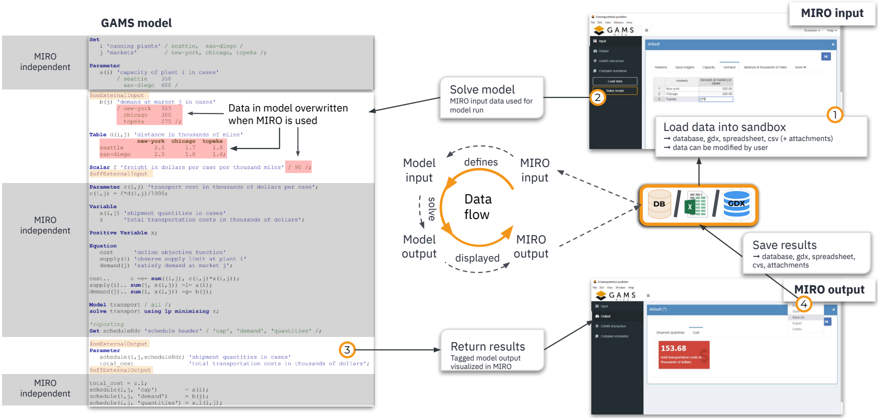
8,789,361 Energy demand gross

1,103 Curtailment of fluct res absolute

2e-04 Curtailment of fluct res relative

Costs ($m)	Node	bio	lig	rsvr	sto5	wind_off
investment MW	DE	13.70	9.88	3.25	19.57	257.80
investment MWh	DE	NA	NA	0.72	5.45	NA
fix MW	DE	12.15	3.69	5.24	NA	102.36
variable fuel	DE	21.03	9.16	NA	NA	NA
variable CO2	DE	NA	30.76	NA	NA	NA
variable WAT aka load change	DE	0.01	0.11	NA	NA	NA

MIRO data concept



GAMS model

MIRO independent

```
Set
   i 'canning plants' / seattle,  san-diego /
   j 'markets'        / new-york, chicago, topeka /;

Parameter
   a(i) 'capacity of plant i in cases'
        / seattle    350
          san-diego  600 /
```

```
$onExternalInput
   b(j) 'demand at market j in cases'
        / new-york   325
          chicago    300
          topeka     275 /;

Table d(i,j) 'distance in thousands of miles'
                 new-york  chicago  topeka
   seattle          2.5      1.7     1.8
   san-diego        2.5      1.8     1.4;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;
$offExternalInput
```

Data in model overwritten when MIRO is used

MIRO independent

```
Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;

Variable
   x(i,j) 'shipment quantities in cases'
   z      'total transportation costs in thousands of dollars';

Positive Variable x;

Equation
   cost      'define objective function'
   supply(i) 'observe supply limit at plant i'
   demand(j) 'satisfy demand at market j';

cost..      z =e= sum((i,j), c(i,j)*x(i,j));
supply(i).. sum(j, x(i,j)) =l= a(i);
demand(j).. sum(i, x(i,j)) =g= b(j);

Model transport / all /;
solve transport using lp minimizing z;

*reporting
Set scheduleHdr 'schedule header' / 'cap', 'demand', 'quantities' /;
```

```
$onExternalOutput
Parameter
   schedule(i,j,scheduleHdr) 'shipment quantities in cases'
   total_cost                'total transportation costs in thousands of dollars';
$offExternalOutput
```

3

MIRO independent

```
total_cost = z.l;
schedule(i,j, 'cap')        = a(i);
schedule(i,j, 'demand')     = b(j);
schedule(i,j, 'quantities') = x.l(i,j);
```

MIRO input

1 Load data into sandbox
- → database, gdx, spreadsheet, csv (+ attachments)
- → data can be modified by user

2 Solve model
MIRO input data used for model run

3 Return results
Tagged model output visualized in MIRO

4 Save results
→ database, gdx, spreadsheet, cvs, attachments

MIRO output

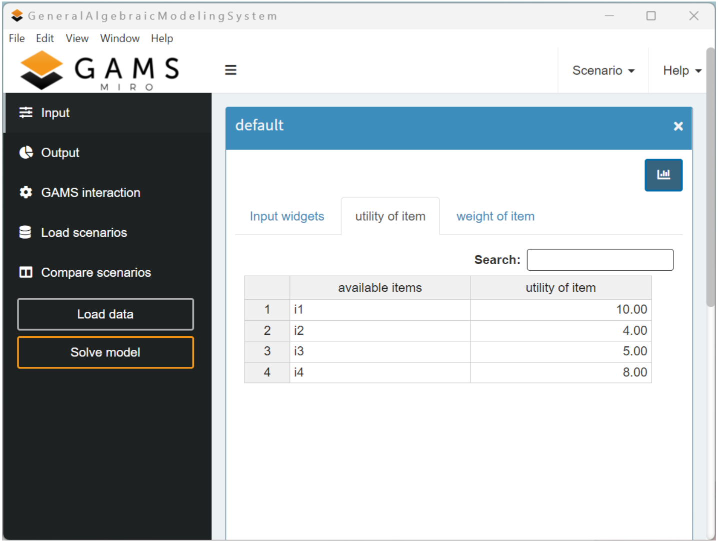
Knapsack example with MIRO

- Parameters (incl. scalars) are “external input”
- Variables are “external output”
- Remaining information related to MIRO are in its configuration (not model!)
- Detailed look (interesting application!) in Robin Schuchmann’s talk (FA-16)

```
parameters
$onExternalInput
c     capacity of knapsack / 8 /
u(i<) utility of item      / i1 10, i2 4, i3 5, i4 8 /
w(i)  weight of item       / i1  8, i2 2, i3 4, i4 5/;
$offExternalInput
$onExternalOutput
binary variable
x(i) 1 iff. item i is put into knapsack;
free variable
utility objective value of selection;
$offExternalOutput
```

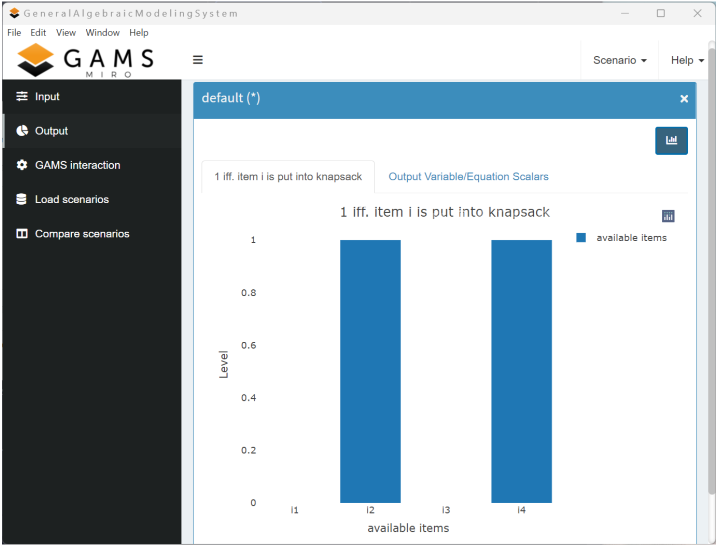
Listing 6: knapsack_miro.gms

Knapsack input UI in MIRO



	available items	utility of item
1	i1	10.00
2	i2	4.00
3	i3	5.00
4	i4	8.00

Knapsack result visualization in MIRO

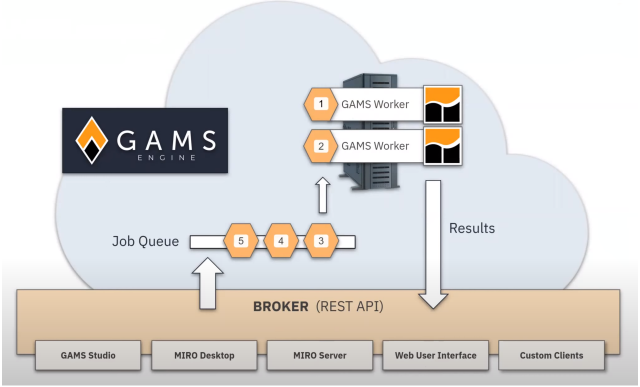


Tour of the GAMS ecosystem and its components

Engine

GAMS Engine

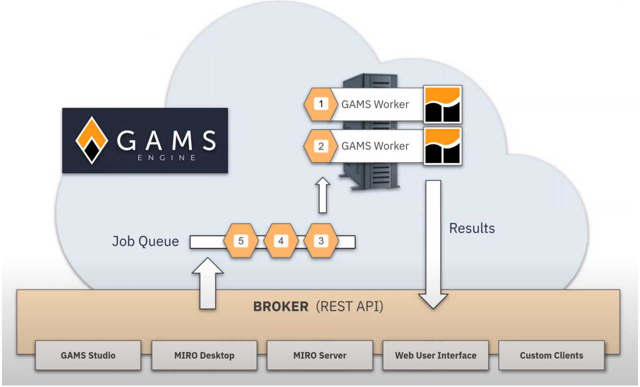
- Running into local machine restrictions with runtimes and parallelism?



GAMS Engine

- Running into local machine restrictions with runtimes and parallelism?

→ Run and solve GAMS models as jobs in the cloud

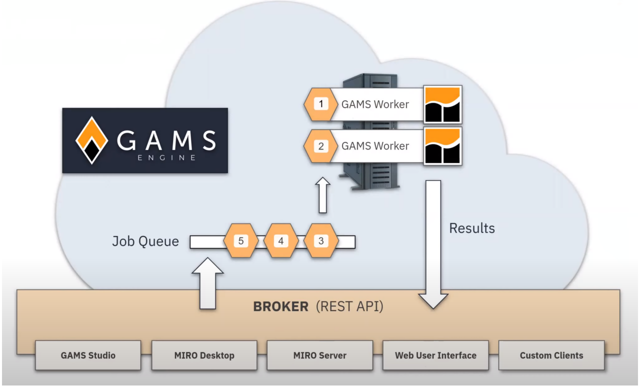


GAMS Engine

- Running into local machine restrictions with runtimes and parallelism?

→ Run and solve GAMS models as jobs in the cloud

- Ideal for sensitivity analysis (what-if) and long-running jobs

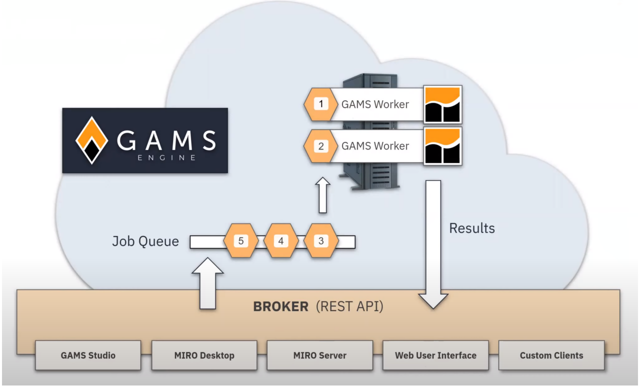


GAMS Engine

- Running into local machine restrictions with runtimes and parallelism?

→ Run and solve GAMS models as jobs in the cloud

- Ideal for sensitivity analysis (what-if) and long-running jobs
- Tight integration with Studio and MIRO

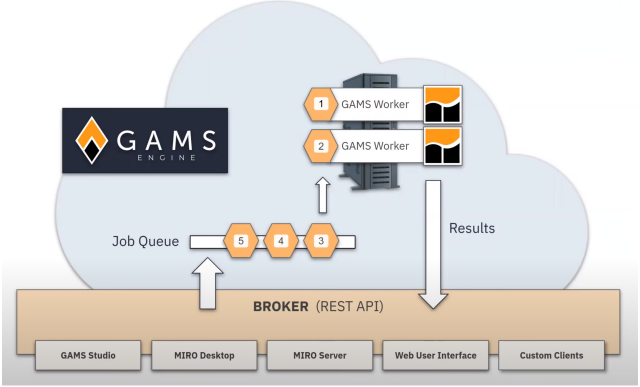


GAMS Engine

- Running into local machine restrictions with runtimes and parallelism?

→ Run and solve GAMS models as jobs in the cloud

- Ideal for sensitivity analysis (what-if) and long-running jobs
- Tight integration with Studio and MIRO
- More details in talk from Stefan Mann (TE-16)



Summary

Summary

What we covered: GAMS is more than just a modeling language

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**
- Offload computations to the cloud (or local cluster) with **Engine**

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**
- Offload computations to the cloud (or local cluster) with **Engine**

Further viewing and reading:

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**
- Offload computations to the cloud (or local cluster) with **Engine**

Further viewing and reading:

- Video-tutorials and -workshops on many topics/components on YouTube: `https://www.youtube.com/user/GAMSLessons`

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**
- Offload computations to the cloud (or local cluster) with **Engine**

Further viewing and reading:

- Video-tutorials and -workshops on many topics/components on YouTube: `https://www.youtube.com/user/GAMSLessons`
- Articles related to GAMS: `https://www.gams.com/blog/`

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**
- Offload computations to the cloud (or local cluster) with **Engine**

Further viewing and reading:

- Video-tutorials and -workshops on many topics/components on YouTube: `https://www.youtube.com/user/GAMSLessons`
- Articles related to GAMS: `https://www.gams.com/blog/`
- Extensive documentation and reference: `https://www.gams.com/latest/docs/`
  - For MIRO: `https://gams.com/miro/`
  - For Engine: `https://gams.com/engine/`

GAMS

Summary

What we covered: GAMS is more than just a modeling language

- Working in a modern integrated development environment with **Studio**
- Data processing for MATLAB/Python/R data scientists with **Transfer**
- Versatile input/output formats and transformations with **Connect**
- Rapidly create a UI for any model with **MIRO**
- Offload computations to the cloud (or local cluster) with **Engine**

Further viewing and reading:

- Video-tutorials and -workshops on many topics/components on YouTube: `https://www.youtube.com/user/GAMSLessons`
- Articles related to GAMS: `https://www.gams.com/blog/`
- Extensive documentation and reference: `https://www.gams.com/latest/docs/`
  - For MIRO: `https://gams.com/miro/`
  - For Engine: `https://gams.com/engine/`
- Open source projects (MIRO, Studio, Engine UI, C++-API, ...) on GitHub: `https://github.com/orgs/GAMS-dev/repositories`

Thank you for your attention!

10 year OR anniversary 🥳

OR 2013 IMPACT

100 JAAR YEARS 1913 2013 IMPACT

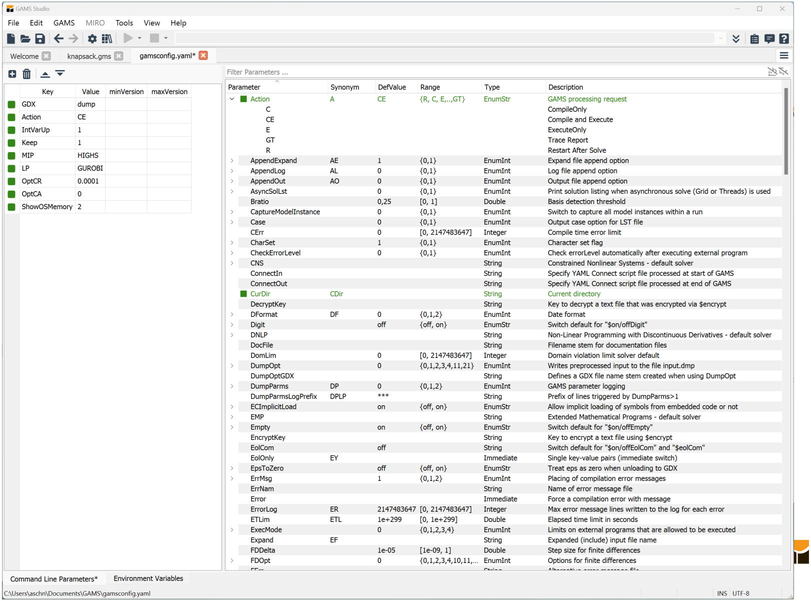
André

Schnabel

Leibniz University Hannover
Germany

Backup slides

GAMS Studio configuration editor

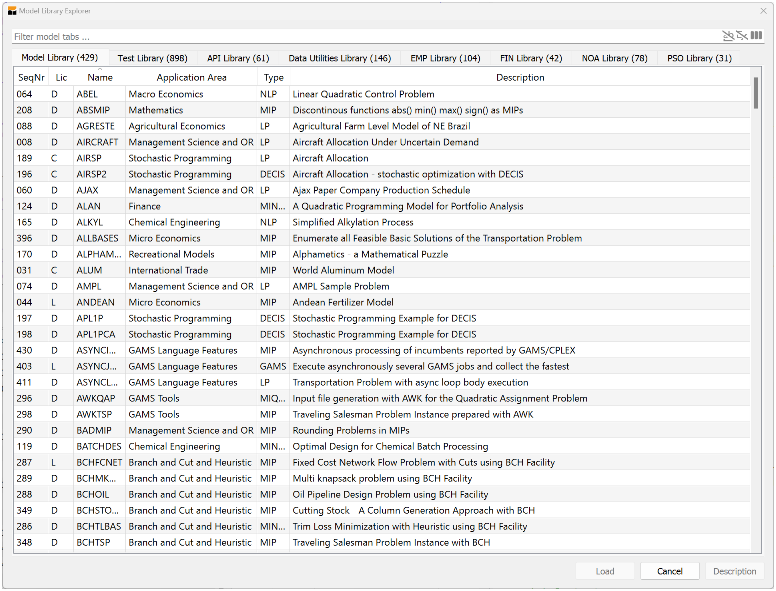


	Key	Value	minVersion	maxVersion
🟩	GDX	dump		
🟩	Action	CE		
🟩	IntVarUp	1		
🟩	Keep	1		
🟩	MIP	HIGHS		
🟩	LP	GUROBI		
🟩	OptCR	0.0001		
🟩	OptCA	0		
🟩	ShowOSMemory	2		

Parameter	Synonym	DefValue	Range	Type	Description
🟩 Action	A	CE	{R, C, E,..,GT}	EnumStr	GAMS processing request
C					CompileOnly
CE					Compile and Execute
E					ExecuteOnly
GT					Trace Report
R					Restart After Solve
AppendExpand	AE	1	{0,1}	EnumInt	Expand file append option
AppendLog	AL	0	{0,1}	EnumInt	Log file append option
AppendOut	AO	0	{0,1}	EnumInt	Output file append option
AsyncSolLst		0	{0,1}	EnumInt	Print solution listing when asynchronous solve (Grid or Threads) is used
Bratio		0,25	[0, 1]	Double	Basis detection threshold
CaptureModelInstance		0	{0,1}	EnumInt	Switch to capture all model instances within a run
Case		0	{0,1}	EnumInt	Output case option for LST file
CErr		0	[0, 2147483647]	Integer	Compile time error limit
CharSet		1	{0,1}	EnumInt	Character set flag
CheckErrorLevel		0	{0,1}	EnumInt	Check errorLevel automatically after executing external program
CNS				String	Constrained Nonlinear Systems - default solver
ConnectIn				String	Specify YAML Connect script file processed at start of GAMS
ConnectOut				String	Specify YAML Connect script file processed at end of GAMS
🟩 CurDir	CDir			String	Current directory
DecryptKey				String	Key to decrypt a text file that was encrypted via $encrypt
DFormat	DF	0	{0,1,2}	EnumInt	Date format
Digit		off	{off, on}	EnumStr	Switch default for "$on/offDigit"
DNLP				String	Non-Linear Programming with Discontinuous Derivatives - default solver
DocFile				String	Filename stem for documentation files
DomLim		0	[0, 2147483647]	Integer	Domain violation limit solver default
DumpOpt		0	{0,1,2,3,4,11,21}	EnumInt	Writes preprocessed input to the file input.dmp
DumpOptGDX				String	Defines a GDX file name stem created when using DumpOpt
DumpParms	DP	0	{0,1,2}	EnumInt	GAMS parameter logging
DumpParmsLogPrefix	DPLP	***		String	Prefix of lines triggered by DumpParms>1
ECImplicitLoad		on	{off, on}	EnumStr	Allow implicit loading of symbols from embedded code or not
EMP				String	Extended Mathematical Programs - default solver
Empty		on	{off, on}	EnumStr	Switch default for "$on/offEmpty"
EncryptKey				String	Key to encrypt a text file using $encrypt
EolCom		off		String	Switch default for "$on/offEolCom" and "$eolCom"
EolOnly	EY			Immediate	Single key-value pairs (immediate switch)
EpsToZero		off	{off, on}	EnumStr	Treat eps as zero when unloading to GDX
ErrMsg		1	{0,1,2}	EnumInt	Placing of compilation error messages
ErrNam				String	Name of error message file
Error				Immediate	Force a compilation error with message
ErrorLog	ER	2147483647	[0, 2147483647]	Integer	Max error message lines written to the log for each error
ETLim	ETL	1e+299	[0, 1e+299]	Double	Elapsed time limit in seconds
ExecMode		0	{0,1,2,3,4}	EnumInt	Limits on external programs that are allowed to be executed
Expand	EF			String	Expanded (include) input file name
FDDelta		1e-05	[1e-09, 1]	Double	Step size for finite differences
FDOpt		0	{0,1,2,3,4,10,11,...	EnumInt	Options for finite differences

C:\Users\aschn\Documents\GAMS\gamsconfig.yaml

Extensive model library



SeqNr	Lic	Name	Application Area	Type	Description
064	D	ABEL	Macro Economics	NLP	Linear Quadratic Control Problem
208	D	ABSMIP	Mathematics	MIP	Discontinous functions abs() min() max() sign() as MIPs
088	D	AGRESTE	Agricultural Economics	LP	Agricultural Farm Level Model of NE Brazil
008	D	AIRCRAFT	Management Science and OR	LP	Aircraft Allocation Under Uncertain Demand
189	C	AIRSP	Stochastic Programming	LP	Aircraft Allocation
196	C	AIRSP2	Stochastic Programming	DECIS	Aircraft Allocation - stochastic optimization with DECIS
060	D	AJAX	Management Science and OR	LP	Ajax Paper Company Production Schedule
124	D	ALAN	Finance	MIN...	A Quadratic Programming Model for Portfolio Analysis
165	D	ALKYL	Chemical Engineering	NLP	Simplified Alkylation Process
396	D	ALLBASES	Micro Economics	MIP	Enumerate all Feasible Basic Solutions of the Transportation Problem
170	D	ALPHAM...	Recreational Models	MIP	Alphametics - a Mathematical Puzzle
031	C	ALUM	International Trade	MIP	World Aluminum Model
074	D	AMPL	Management Science and OR	LP	AMPL Sample Problem
044	L	ANDEAN	Micro Economics	MIP	Andean Fertilizer Model
197	D	APL1P	Stochastic Programming	DECIS	Stochastic Programming Example for DECIS
198	D	APL1PCA	Stochastic Programming	DECIS	Stochastic Programming Example for DECIS
430	D	ASYNCI...	GAMS Language Features	MIP	Asynchronous processing of incumbents reported by GAMS/CPLEX
403	L	ASYNCJ...	GAMS Language Features	GAMS	Execute asynchronously several GAMS jobs and collect the fastest
411	D	ASYNCL...	GAMS Language Features	LP	Transportation Problem with async loop body execution
296	D	AWKQAP	GAMS Tools	MIQ...	Input file generation with AWK for the Quadratic Assignment Problem
298	D	AWKTSP	GAMS Tools	MIP	Traveling Salesman Problem Instance prepared with AWK
290	D	BADMIP	Management Science and OR	MIP	Rounding Problems in MIPs
119	D	BATCHDES	Chemical Engineering	MIN...	Optimal Design for Chemical Batch Processing
287	L	BCHFCNET	Branch and Cut and Heuristic	MIP	Fixed Cost Network Flow Problem with Cuts using BCH Facility
289	D	BCHMK...	Branch and Cut and Heuristic	MIP	Multi knapsack problem using BCH Facility
288	D	BCHOIL	Branch and Cut and Heuristic	MIP	Oil Pipeline Design Problem using BCH Facility
349	D	BCHSTO...	Branch and Cut and Heuristic	MIP	Cutting Stock - A Column Generation Approach with BCH
286	D	BCHTLBAS	Branch and Cut and Heuristic	MIN...	Trim Loss Minimization with Heuristic using BCH Facility
348	D	BCHTSP	Branch and Cut and Heuristic	MIP	Traveling Salesman Problem Instance with BCH

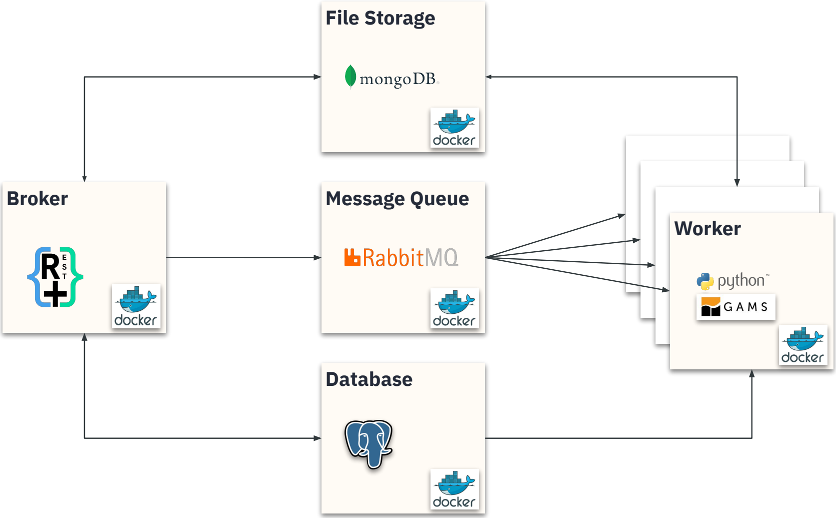
Library explorer dialog from GAMS Studio

Python API: GDX file construction

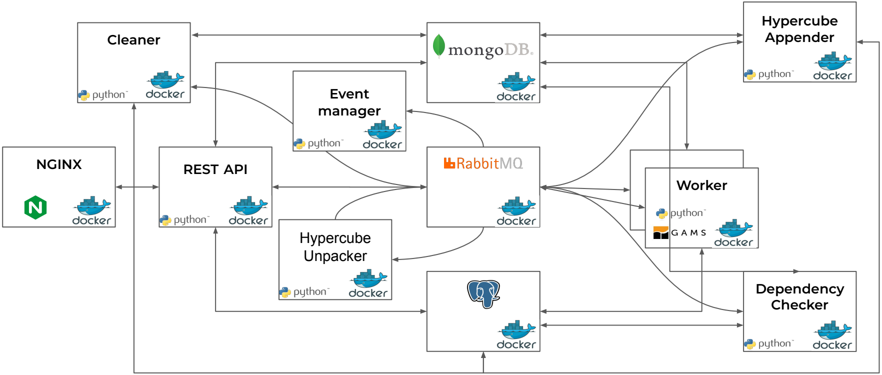
```python
ws = GamsWorkspace()
db = ws.add_database()
items = [f'i{i + 1}' for i in range(4)]
utilities, weights = [10, 4, 5, 8], [8, 2, 4, 5]
i = db.add_set('i', 1, 'available items')
for item in items: i.add_record(item)
c = db.add_parameter('c', 0, 'capacity of knapsack')
c.add_record().value = 8
u = db.add_parameter('u', 1, 'utility of item')
for item, utility in zip(items, utilities):
    u.add_record(item).value = utility
w = db.add_parameter('w', 1, 'weight of item')
for item, weight in zip(items, weights):
    w.add_record(item).value = weight
db.export('knapsack_instance.gdx')
...
```

Listing 7: knapsack_pyapi.gms

GAMS Engine architecture internals



GAMS Engine architecture internals



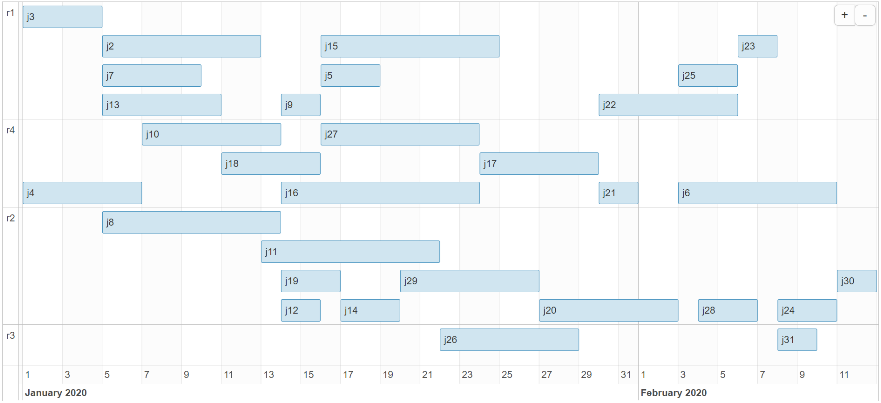
GANTT chart for a RCPSP instance in MIRO

- Convert time periods $1 \ldots T$ to dates for interoperability with 'timevis'
- Model available via 'gamslib rcpsp' from GAMS model library

```
set ds /2020-01-01/;
$onMulti
$onEmbeddedCode Python:
from datetime import date
def to_date(t):
    refdate = date.toordinal(date(2020, 1, 1))
    return date.fromordinal(refdate + t).strftime("%Y-%m-%d")
ds=[to_date(int(t.replace(’t’, ’’))) for t in gams.get(’t’)]
gams.set("ds", ds)
$offEmbeddedCode ds
alias(ds, start, end);
$onExternalOutput
parameter gantt(j, start, end, r);
$offExternalOutput
gantt(j, start, end, r)$(st(j)=ord(start) and st(j)+durations(j)=
    ord(end) and demands(j,r) > 0) = demands(j,r);
```

Listing 8: rcpsp_miro.gms

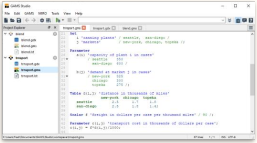
Resulting schedule for the RCPSP in MIRO



The modern GAMS ecosystem

GAMS -
Modeling Platform

- **Algebraic Modeling Language** (platform independent)
- **Commercial and Academic Solvers** (packaged)
- **APIs for major programming languages** (C++, Java, Python, Matlab and more)

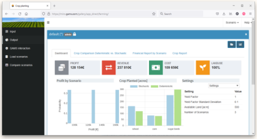


```
Set
   i 'canning plants' / seattle,  san-diego /
   j 'markets'        / new-york, chicago, topeka /;

Parameter
   a(i) 'capacity of plant i in cases'
        / seattle    350
          san-diego  600 /

   b(j) 'demand at market j in cases'
        / new-york   325
          chicago    300
          topeka     275 /;

Table d(i,j) 'distance in thousands of miles'
              new-york  chicago  topeka
   seattle       2.5       1.7     1.8
   san-diego     2.5       1.8     1.4;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;
```

→ Domain Experts
→ Modellers/Developers

MIRO -
Graphical UI Generator

- **Turns models into web applications**
- **Interactive graphical output**
- **Extendable with custom code**
- **Local or Server Installation**

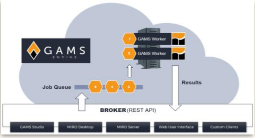


Setting	Value
Yield Factor	1
Yield Factor Standard Deviation	0.1
Available Land [acre]	500
Number of Scenarios	3

→ End Users
→ Analysts

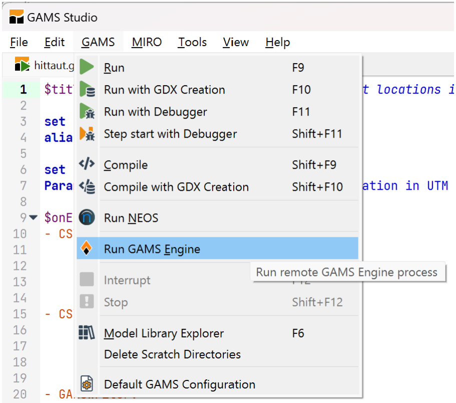
Engine -
Deployment Solution

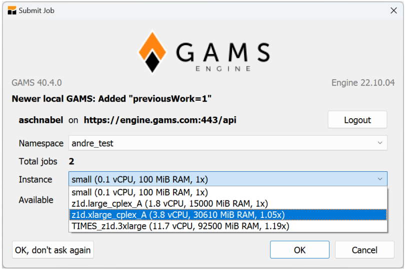
- **Solves GAMS models on centralized resources** (on-prem or cloud)
- **REST API** (user & job management)
- **GAMS job scheduling & Load balancing**



→ IT Admins
→ Transparent to End Users

GAMS Studio integration of remote execution


GAMS Studio
File
Edit
GAMS
MIRO
Tools
View
Help
hittaut.g
Run F9
Run with GDX Creation F10
Run with Debugger F11
Step start with Debugger Shift+F11
Compile Shift+F9
Compile with GDX Creation Shift+F10
Run NEOS
Run GAMS Engine
Run remote GAMS Engine process
Interrupt F12
Stop Shift+F12
Model Library Explorer F6
Delete Scratch Directories
Default GAMS Configuration
$tit
set
alia
set
Para
$onE
- CS
- CS
- GA
t locations i
ation in UTM



GAMS 40.4.0

Engine 22.10.04

Newer local GAMS: Added "previousWork=1"

aschnabel on **https://engine.gams.com:443/api**

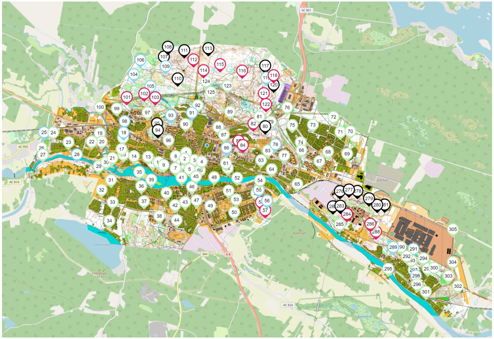
Namespace: andre_test

Total jobs: **2**

Instance: small (0.1 vCPU, 100 MiB RAM, 1x)

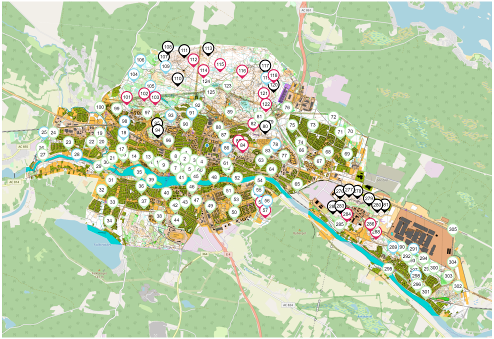
Available:
- small (0.1 vCPU, 100 MiB RAM, 1x)
- z1d.large_cplex_A (1.8 vCPU, 15000 MiB RAM, 1x)
- z1d.xlarge_cplex_A (3.8 vCPU, 30610 MiB RAM, 1.05x)
- TIMES_z1d.3xlarge (11.7 vCPU, 92500 MiB RAM, 1.19x)

HITTAUT in Sweden (e.g. Skellefteå)



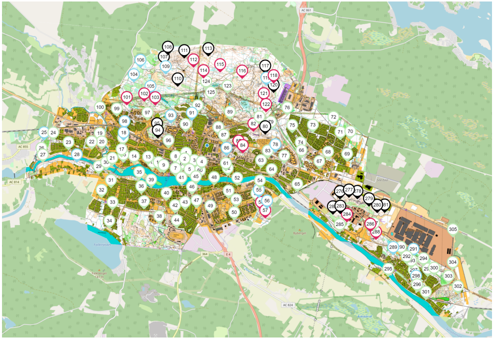
- Similar to Wandernadel but some tags intentionally difficult to find

HITTAUT in Sweden (e.g. Skellefteå)



- Similar to Wandernadel but some tags intentionally difficult to find
- Prove visit by putting secret code into smartphone app with online highscore

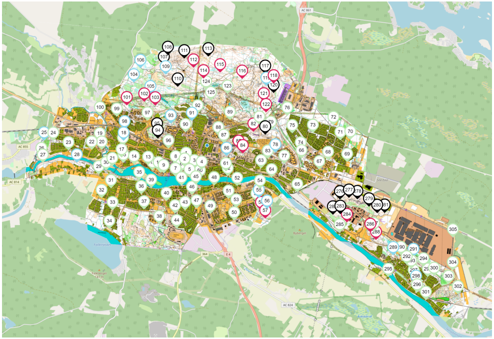
HITTAUT in Sweden (e.g. Skellefteå)



- Similar to Wandernadel but some tags intentionally difficult to find
- Prove visit by putting secret code into smartphone app with online highscore

- Determine an optimal hiking tour with GAMS
  1. Fetch locations as JSON via REST-API (Embedded Code)

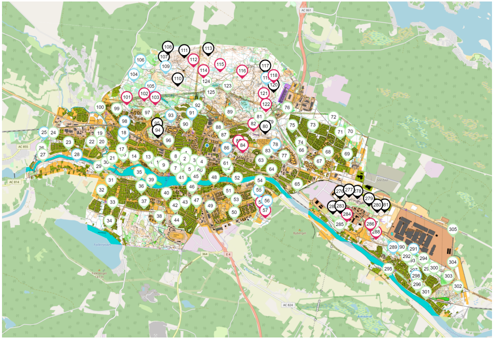
HITTAUT in Sweden (e.g. Skellefteå)



- Similar to Wandernadel but some tags intentionally difficult to find
- Prove visit by putting secret code into smartphone app with online highscore

- Determine an optimal hiking tour with GAMS
  1. Fetch locations as JSON via REST-API (Embedded Code)
  2. Fill location set and UTM coordinates into GAMS symbols (Embedded Code)

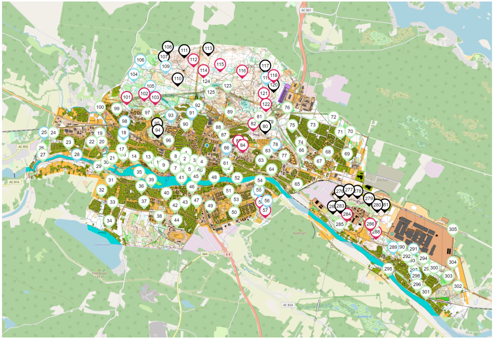
HITTAUT in Sweden (e.g. Skellefteå)



- Similar to Wandernadel but some tags intentionally difficult to find
- Prove visit by putting secret code into smartphone app with online highscore

- Determine an optimal hiking tour with GAMS
  1. Fetch locations as JSON via REST-API (Embedded Code)
  2. Fill location set and UTM coordinates into GAMS symbols (Embedded Code)
  3. Solve TSP model with CPLEX

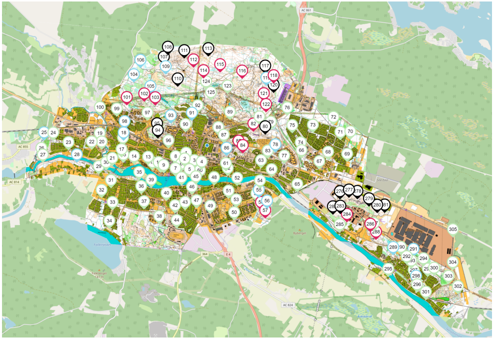
HITTAUT in Sweden (e.g. Skellefteå)



- Similar to Wandernadel but some tags intentionally difficult to find
- Prove visit by putting secret code into smartphone app with online highscore

- Determine an optimal hiking tour with GAMS
  1. Fetch locations as JSON via REST-API (Embedded Code)
  2. Fill location set and UTM coordinates into GAMS symbols (Embedded Code)
  3. Solve TSP model with CPLEX
  4. Write solution to CSV (Connect)

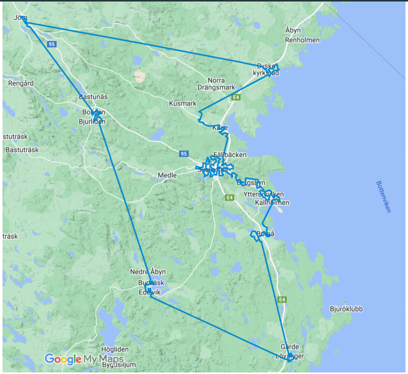
HITTAUT in Sweden (e.g. Skellefteå)

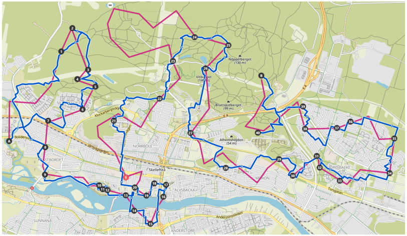


- Similar to Wandernadel but some tags intentionally difficult to find
- Prove visit by putting secret code into smartphone app with online highscore

- Determine an optimal hiking tour with GAMS
  1. Fetch locations as JSON via REST-API (Embedded Code)
  2. Fill location set and UTM coordinates into GAMS symbols (Embedded Code)
  3. Solve TSP model with CPLEX
  4. Write solution to CSV (Connect)
  5. Convert into GPX file (Python script)

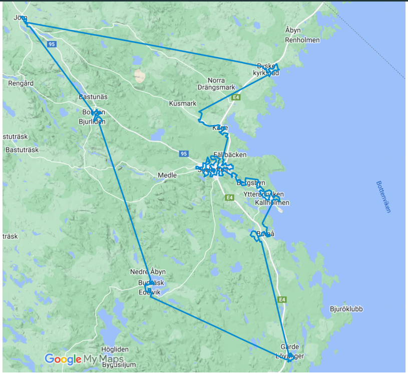
Optimal tour for HITTAUT in Skellefteå

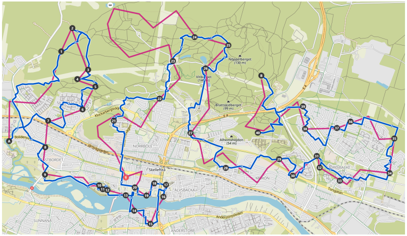




- GAMS/CPLEX needs 4min 20s and Concorde/CPLEX just 1.7s

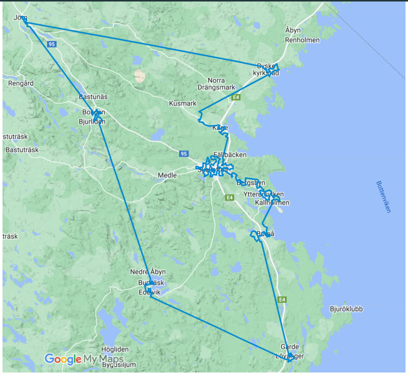
Optimal tour for HITTAUT in Skellefteå

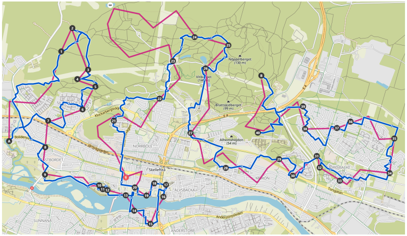




- GAMS/CPLEX needs 4min 20s and Concorde/CPLEX just 1.7s
- Optimal tour for all 419 tags is ≥ 371 km (as the crow flies)

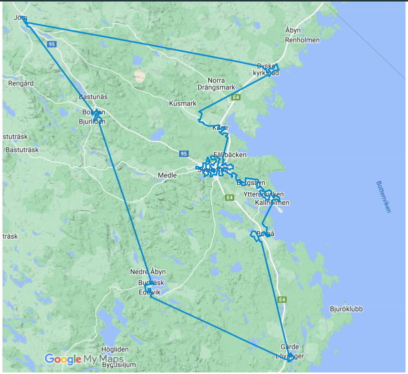
Optimal tour for HITTAUT in Skellefteå

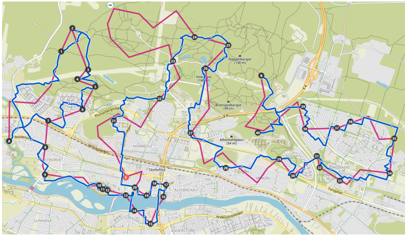




- GAMS/CPLEX needs 4min 20s and Concorde/CPLEX just 1.7s
- Optimal tour for all 419 tags is ≥ 371 km (as the crow flies)
- Many tag locations too far off paths

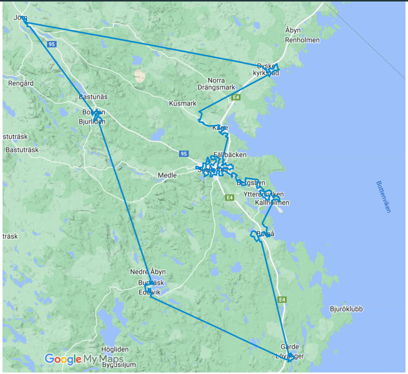
Optimal tour for HITTAUT in Skellefteå

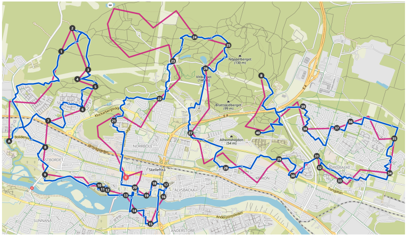




- GAMS/CPLEX needs 4min 20s and Concorde/CPLEX just 1.7s
- Optimal tour for all 419 tags is ≥ 371 km (as the crow flies)
- Many tag locations too far off paths
- Komoot routing struggles to cover first 80 waypoints with a hiking route

Optimal tour for HITTAUT in Skellefteå





- GAMS/CPLEX needs 4min 20s and Concorde/CPLEX just 1.7s
- Optimal tour for all 419 tags is ≥ 371 km (as the crow flies)
- Many tag locations too far off paths
- Komoot routing struggles to cover first 80 waypoints with a hiking route
- Shows limitations of using euclidean distances (vs. walkable connections)

Python API

- GAMS is well-suited for model definition but not algorithmic programming

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records
- Create and read GAMS Data eXchange (GDX) files

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records
- Create and read GAMS Data eXchange (GDX) files
- Trigger GAMS jobs with custom options from code

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records
- Create and read GAMS Data eXchange (GDX) files
- Trigger GAMS jobs with custom options from code
- Re-solve slightly modified instances without full model re-generation

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records
- Create and read GAMS Data eXchange (GDX) files
- Trigger GAMS jobs with custom options from code
- Re-solve slightly modified instances without full model re-generation
- Retrieve result data back to native data structures

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records
- Create and read GAMS Data eXchange (GDX) files
- Trigger GAMS jobs with custom options from code
- Re-solve slightly modified instances without full model re-generation
- Retrieve result data back to native data structures
- Similar APIs also available for: .NET, Java, MATLAB, C++

Python API

- GAMS is well-suited for model definition but not algorithmic programming

→ Solution: Use API to interact with GAMS from Python

Control and data API

- Object-oriented: Database, symbols (sets, parameters, variables), and records
- Create and read GAMS Data eXchange (GDX) files
- Trigger GAMS jobs with custom options from code
- Re-solve slightly modified instances without full model re-generation
- Retrieve result data back to native data structures
- Similar APIs also available for: .NET, Java, MATLAB, C++
- More details in Justine's talk coming up next in this session

Python API: solve with varying knapsack capacities

```python
...
job = ws.add_job_from_file('knapsack_mi.gms')
cp = ws.add_checkpoint()
job.run(checkpoint=cp)
mi = cp.add_modelinstance()
mi_capacity = mi.sync_db.add_parameter("c", 0, "varying capacity")
mi.instantiate('knapsack use mip max utility',GamsModifier(mi_capacity))
mi_capacity.add_record().value = 1
for cap_vary in range(6, 12):
    mi_capacity.first_record().value = cap_vary
    mi.solve()
    print(f'Maximum utility for capacity {cap_vary}: {mi.sync_db["utility"].find_record().level}')
```

Listing 9: knapsack_pyapi.gms

```
Maximum utility for capacity 6: 9.0
Maximum utility for capacity 7: 12.0
Maximum utility for capacity 8: 12.0
Maximum utility for capacity 9: 13.0
Maximum utility for capacity 10: 14.0
Maximum utility for capacity 11: 17.0
```

Listing 10: output

Embedded code

- Only fragments of Python code required to “decorate” GAMS model?

→ Intersperse snippets inside a GAMS model with easy API for data exchange

```
$include knapsack_core
$onEmbeddedCode Python:
import urllib.request
import zipfile
def ints(coll): return [ int(elem) for elem in coll ]
url='http://artemisa.unicauca.edu.co/~johnyortega/instances_01_KP/instances_01_KP.zip'
urllib.request.urlretrieve(url, 'instances.zip')
with zipfile.ZipFile('instances.zip', 'r') as zip_ref:
    zip_ref.extractall('instances')
with open('instances/low-dimensional/f1_l-d_kp_10_269') as fp:
    lines = fp.readlines()
num_items, capacity = ints(lines[0].split())
items = [f'i{i+1}' for i in range(num_items)]
utilities = ints([ line.split()[0] for line in lines[1:] ])
weights = ints([ line.split()[1] for line in lines[1:] ])
gams.set('i', items)
gams.set('c', [capacity])
gams.set('u', list(zip(items, utilities)))
gams.set('w', list(zip(items, weights)))
$offEmbeddedCode i c u w
model knapsack /all/;
solve knapsack using mip maximizing utility;
```

Listing 11: knapsack_embcode.gms