sysPreliminary tests

Test

Sampled

Optimisation

Theory

Relationships matrix

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,N} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N,1} & m_{N,2} & \cdots & m_{N,N} \end{bmatrix}$$

$m_{i,j} \in [-1, 1]$.

Select component A_i

Introduce the set $\{s_{i,j}\}$ of random variables of binomial distribution:

$P\{s_{i,j} = 1\} = |m_{i,j}|$,

$P\{s_{i,j} = 0\} = 1 - |m_{i,j}|$.

Let in the moment $t = t_0$ the system has the state $(A_1(t_0), A_2(t_0), \ldots, A_N(t_0))$

Consider

$$d_i = \sum_{j=1}^{N} sign(m_{i,j}) s_{i,j} \varphi_{i,j}(A_j(t_0))$$

d_i is a random variable taking values $\sum_{j=1}^{N} sign(m_{i,j}) k_j \varphi_{i,j}(A_j(t_0))$, $k_j = \{0, 1\}$, $j = 1, 2, \ldots, N$ with probabilities $\prod_{j=1}^{N} (2|m_{i,j}| - 1) k_j + 1 - |m_{i,j}|$

$\mathrm{D} = (d_1, d_2, \ldots, d_N)$ -- random vector

Let $d_1, d_2, \ldots, d_M$ — the values which d takes ($M \le 2^N$) and $p_1, p_2, \ldots, p_M$ — probabilities:

$P\{d = d_l\} = p_l$.

Let δ — threashold.

Let S -- a set of sets $\boldsymbol{s} = (s_1, s_2, \ldots, s_N)$, where each $s_i \in \{-, 0, +\}$. I.e. there 3^N such $\boldsymbol{s}$ in S.

Let $I^- = (-\infty, -\delta]$, $I^0 = (-\delta, \delta)$, $I^+ = [\delta, \infty)$.

For each $\boldsymbol{s} \in S$ compute $P_s = P\left\{d_1 \in I^{s_1}, d_2 \in I^{s_2}, \ldots, d_N \in I^{s_N}\right\}$

It's clear, that some P_s can be equal 0.

Next define the next state of the system $(A_1(t+1), A_2(t+1), \ldots, A_N(t+1))$ and probability of transition to this state as following:

for each i ($i = 1, 2, \ldots, N$) define:

$$A_i(t+1) = \begin{cases} \begin{bmatrix} Dec(A_i(t)) & \textbf{if } s_i = - \\ A_i(t) & \textbf{if } s_i = 0 \\ Inc(A_i(t)) & \textbf{if } s_i = + \end{bmatrix} \end{cases}$$

with probability s_i.

As there are N components, to each state $A_1(t), A_2(t),\ldots, A_N(t)$ can correspons 3^N states $A_1(t+1), A_2(t+1),\ldots, A_N(t+1)$.

N = 4, **K** = 3

```
> N := parse(DocumentTools[GetProperty](ComboN, 'value')): K:=parse(DocumentTools
  [GetProperty](ComboK, 'value')): KN :=K^N;
  RG := (i,j) -> (RandomTools[MersenneTwister][GenerateFloat64]()-0.5)*2:
```

$KN := 81$

Relation matrix generation

```
> M:= Matrix(N, N, (i,j) -> RG(i,j));
```

M =

	1	2	3	4
1	.62	.81	.74	.82
2	-.26	-.80	-.44	-.97
3	.59	.92	.68	.94
4	.91	.29	.60	.72

["1.1.1.1", "1.2.2.1", "1.2.2.2", "2.1.1.1", "2.2.2.1", "2.2.2.2"]

```
> GraphTheory[Arrivals](G1, "3.1.1.1"); GraphTheory[Departures](G1, "1.2.2.2");
```

["2.1.1.2", "2.1.2.1", "2.1.2.2", "2.2.1.1", "2.2.1.2", "2.2.2.1", "2.2.2.2", "3.1.1.1", "3.1.1.2", "3.1.2.1", "3.1.2.2", "3.2.1.1", "3.2.1.2", "3.2.2.1", "3.2.2.2"]

["1.2.2.2", "1.2.2.3", "1.2.3.2", "1.2.3.3", "1.3.2.2", "1.3.2.3", "1.3.3.2", "1.3.3.3", "2.2.2.2", "2.2.2.3", "2.2.3.2", "2.2.3.3", "2.3.2.2", "2.3.2.3", "2.3.3.2", "2.3.3.3"]

```
> print(M);
```

Presentation of a number bas-bases with N0 length

N0 — width of output, **bas** — base, **nb** — number to be convert (from *0*).

```
> BasePres := proc(N0, bas, nb::integer)::list;
  return [seq(0, i02 =1.. N0-numelems(ListTools[Reverse](convert(nb,
   'base', bas)))), op(ListTools[Reverse](convert(nb, 'base', bas)))]
  end proc;
  Inc := proc(k :: integer)::integer;
  return min(k+1,K);
  end proc:
  Dec := proc(k :: integer)::integer;
  return max(k-1,1);
  end proc:
```

BasePres := **proc**(*N0*, *bas*, *nb*::*integer*)::*list*;

return [*seq*(0, *i02* = 1 ..*N0* − *numelems*(*ListTools*[*Reverse*](*convert*(*nb*, '*base*', *bas*)))),