

The implementation is divided into three separate c files:

- 1) **rtsched_gen.c** [Generates the required files]
- 2) **rtsched_sa.c** [Scheduler Implementation]
- 3) **read_online.c** [Result Analyser]

Compiling :

GSL library is a requirement for the simulator. Well known library. Install it from <http://www.gnu.org/software/gsl/>.

```
gcc -std=c99 rtsched_gen.c -o rtsched_gen -lgsl -lgslcblas -lm
gcc -std=c99 rtsched_sa.c -o rtsched_sa -lgsl -lgslcblas -lm
gcc -std=c99 read_online.c -o read_online
```

A typical Run of the simulator:

- 1) Set the required TPG Parameters in rtsched_gen.h and rtsched_gen.c
- 2) Compile rtsched_gen.c
- 3) run rtsched_gen
- 4) Set scheduling specific options in rtsched_sa.c and rtsched_sa.h
- 5) Compile rtsched_sa.c
- 6) run rtsched_sa
- 7) Modify result analyzer read_online.c as required
- 8) Compile read_online.c
- 9) Run read_online

rtsched_gen.[ch]

Main TPG Parameters (change rtsched_gen.c):

```
#define NUMBER_OF_TASKS      ( Number of tasks )
#define NUMBER_OF_PROCESSORS ( Number of Processors )
#define NUMBER_OF_GENERATIONS ( Total number of TPG's )
#define MAX_OUT_DEGREE      ( Maximum Outdegree )
#define EDGE_PROBABILITY    ( Edge Probability )
#define MIN_RUN_TIME        ( Minimum run time fraction )
#define DEVIATION_FACTOR    ( Deviation factor )
#define ALLOC_TIME          ( Allocation Granularity )
```

Important functions , structures and notes:

main() - global function controls generation of TPG's
initialize() - initialize metadata for the TPG
deadline_fd - file contains the deadlines for each of the TPG's
generate_adj_matrix() - generates a valid DAG as per specification
struct task_type TaskPtr - Task structure

Adj_Mat , Comm_Mat , Sen_Mat - Adjacency , Communication and Sensitivity Matrix

rtsched_sa.[ch]

Important Scheduling parameters

```
#define N_TRIES      - Number of tries
#define ITERS_FIXED_T  - Iterations at each temperature
#define K           - Boltzmann constant
#define T_INITIAL    - initial temperature
#define MU_T 1.1     - damping factor for temperature
#define T_MIN 0.01   - Final Temperature
```

```
E1, M1, S1, P1      - Simulated Annealing related functions ( Refer Gsl Reference manual )
greedy_allocator     - Offline Greedy Allocator
offline_final_error_gen[] - holds offline final error of TPG
online_final_error_gen[] - holds online error of the TPG
pre_online()         - Prepares the TPG for online schedule
online_schedule()    - simulates the online stage of a task

online_error.txt     - Contains online error of the TPGs
```

read_online.c

Simple file to analyse the results produced by the scheduler.