SCHEDULING FOR A TTCAN NETWORK WITH A STOCHASTIC OPTIMIZATION ALGORITHM

Fernanda Coutinho*, Jorge Barreiros*, José A. Fonseca**

* *Instituto Politécnico de Coimbra*

** *DET-IEETA University of Aveiro P-3810-193 Aveiro, Portugal*

Presented by José A. Fonseca at FET2001, Nancy-France, 15-16 Nov 2001

Presentation outline:

1 – Objectives

2 – TT-CAN : a brief presentation

3 – Building a schedule

4 – Simulations and results

5 – Conclusions

1 - Objectives:

- **To develop an automatic tool to create valid scheduling tables for the Time Triggered Controller Area Network (TTCAN) protocol.**
- **To exploit the optimisation features of stochastic algorithms (similar to genetic algorithms).**
- **To achieve low message jitter.**

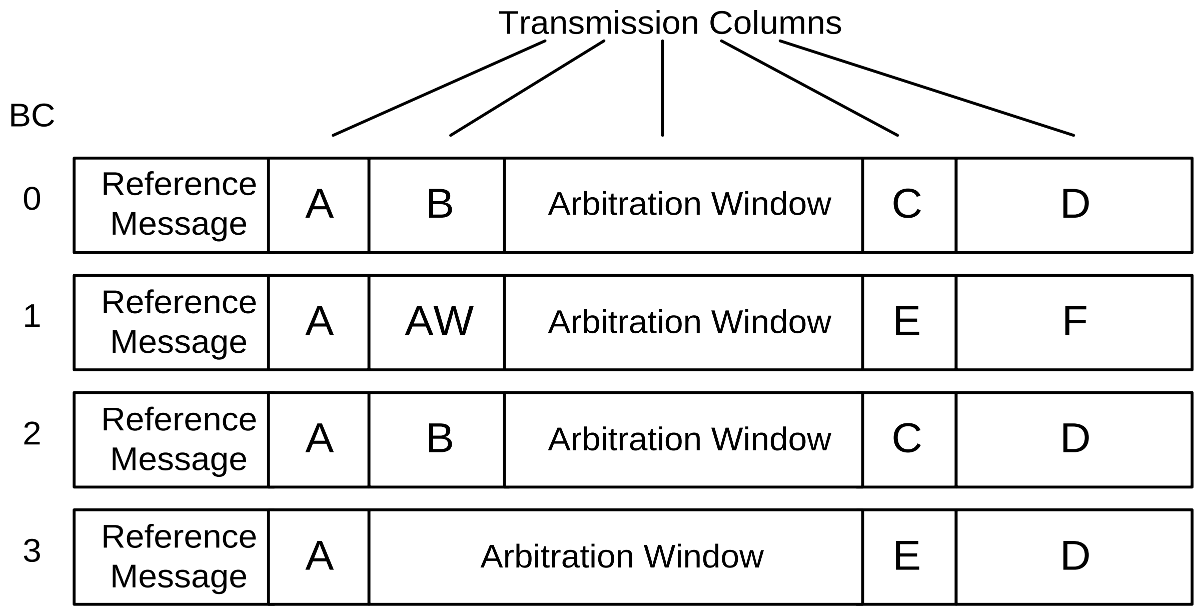
2–TT-CAN: a brief presentation

Protocol based on CAN (Controller Area Network)

- Draft standard produced by task force (ISO/TC 22/SC 3/WG 1/TF 6) in 2000.
- If approved will become ISO IS11898-4.
- Defines a Session Layer for CAN.
- A special node, the time master, transmits a systolic reference message used to achieve synchronisation between nodes.
- The reference message triggers the basic cycles (BC).
- The final schedule includes a fixed number of BCs and is called the System Matrix or Matrix Cycle (MC).

2–TT-CAN: a brief presentation

A Matrix Cycle in TTCAN:



AW - Arbitration Window

BC - Basic Cycles

A, B, C, D, E, F - Periodic Messages

2–TT-CAN: a brief presentation

Some additional issues on generating the schedule:

- The system matrix must be defined off-line.
- The number of Basic Cycles (BCs) must be an integer power of two.

3 – Building a Schedule

Building the system matrix (SM):

- Number of columns
- Duration of each column
- Number of rows
- Messages' period and duration must be respected

First approach:

- The average period of each message is kept identical to the respective instantaneous period (duration of SM / # of message instances).

3 – Building a Schedule

Assessing the quality of the schedule in the SM

Cost function based on the sum of message jitter:

$$Jitter = \frac{1}{M} \sum_{p} \sum_{i} \left| e_i^p - a_i^p \right|$$

e_i^p Expected beginning time of transmission of instance ***i*** of message ***p***

a_i^p Actual beginning time

M Duration of the system matrix

3 – Building a Schedule

Required Steps

- **Scheduling:**
  - Generation of a feasible set of distinct System Matrixes (SMs).
- **Optimisation:**
  - Selection of the best SMs according to the cost function.

3 – Building a Schedule

Generation of a feasible set of SMs

- Determination of the maximum number of lines of the System Matrix.
- Message allocation.
  - Longer messages are taken into consideration first.
- Free time redistribution.
  - The random factor in the first set of system matrixes.

3 – Building a Schedule

Optimisation Process

• Generate a set of system matrices with a random number of lines.

• Run an initial transient phase, in order to increase the diversity of the matrices in the set.

• Randomly select one of the matrices and apply a random transformation.

• If this new matrix is better (has less jitter) than the worst matrix included in the set, then the later is replaced by the former.

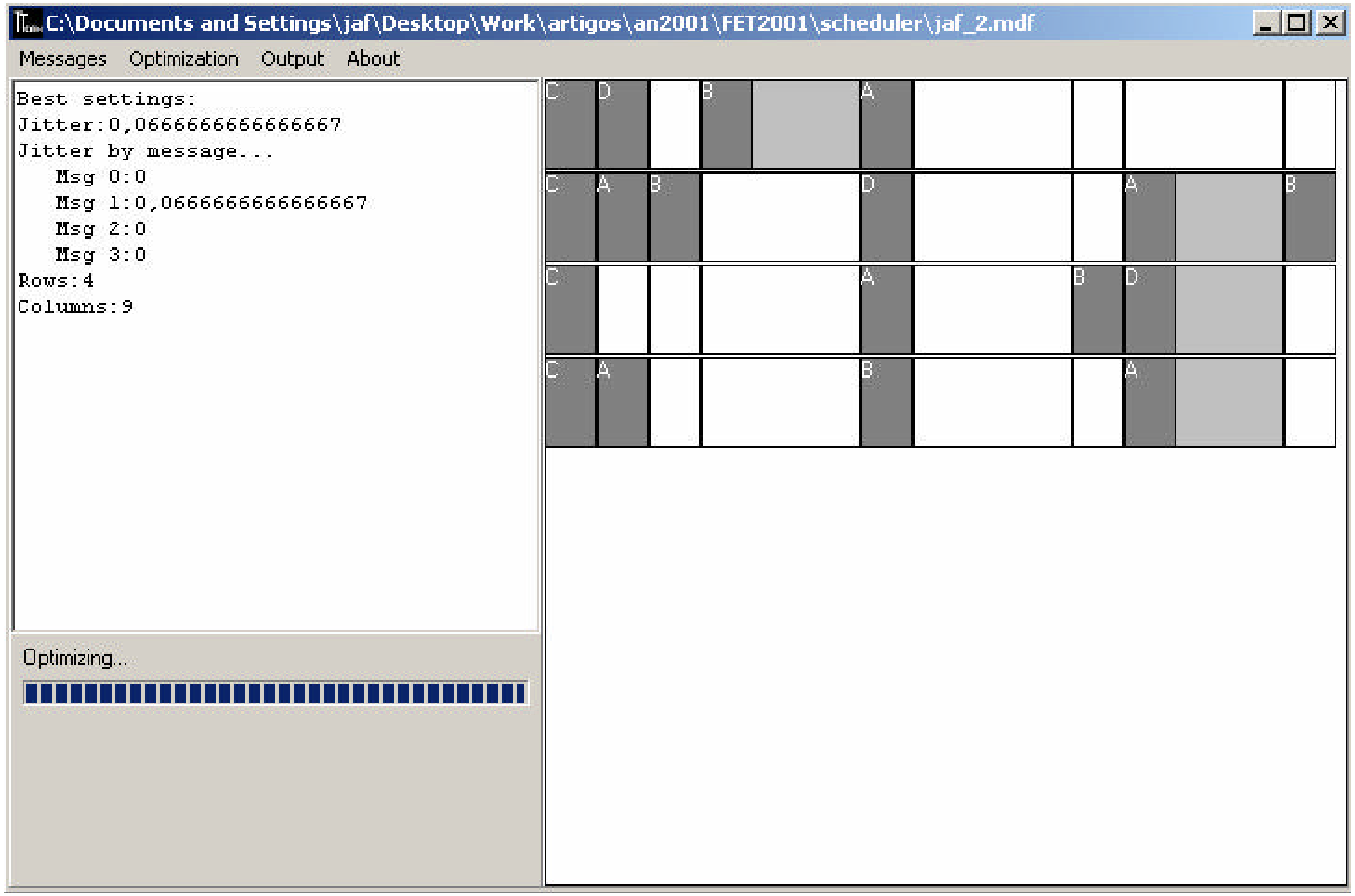
• Repeat until the maximum number of iterations.

3 – Building a Schedule

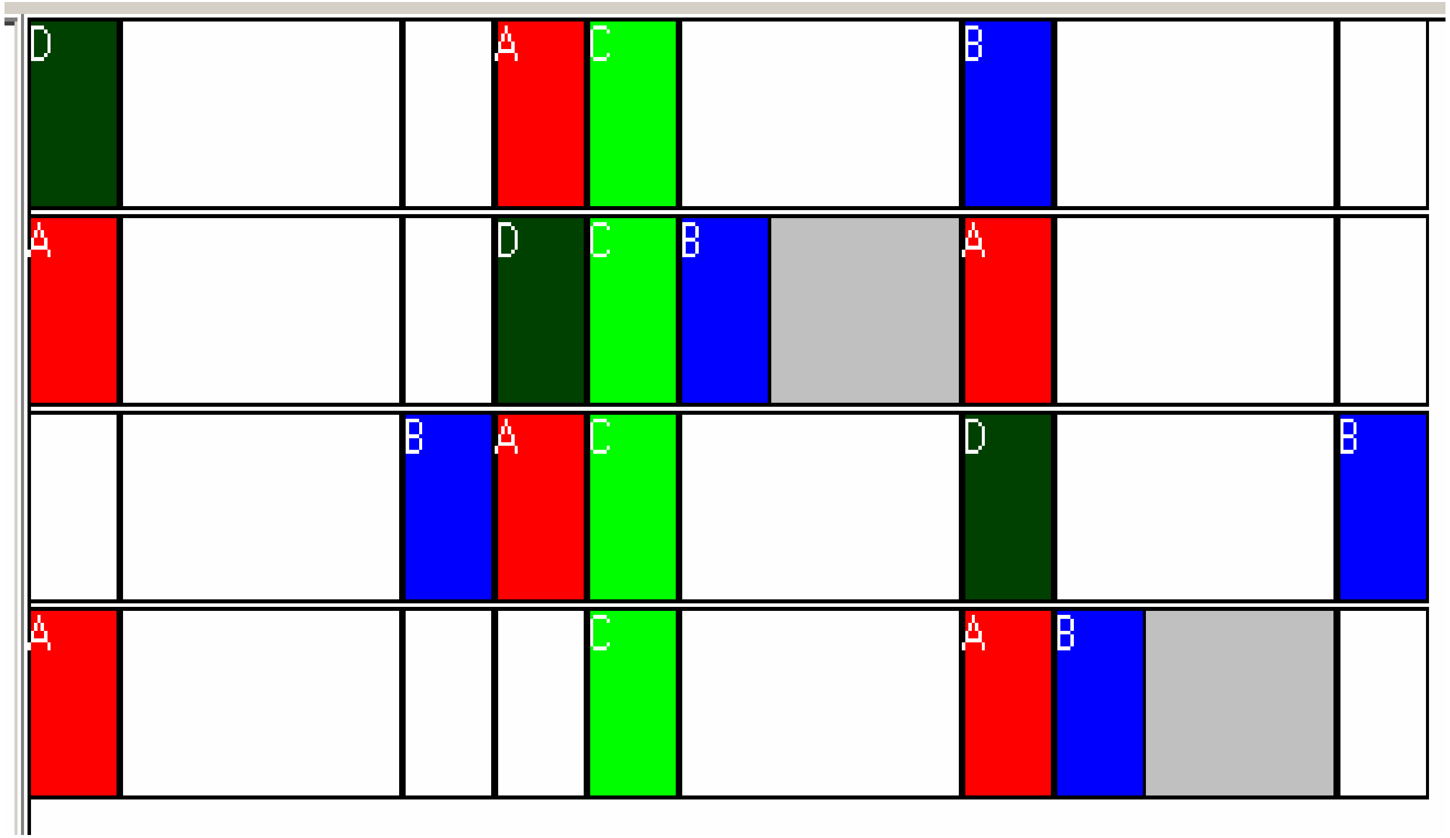
Matrix Transformations

- Cell swap (dependent on cell width).
- Column swap (between two randomly selected columns).
- Free time redistribution (A random percentage of the time slack in a random column is allocated to another).
- Vertical Mirror (A random number of adjacent columns is mirrored across its vertical axis).
- Horizontal Mirror (A random rectangular area inside the matrix is mirrored across its horizontal axis).

3 – Building a Schedule



3 – Building a Schedule



4 – Simulations and Results

Message sets and simulations

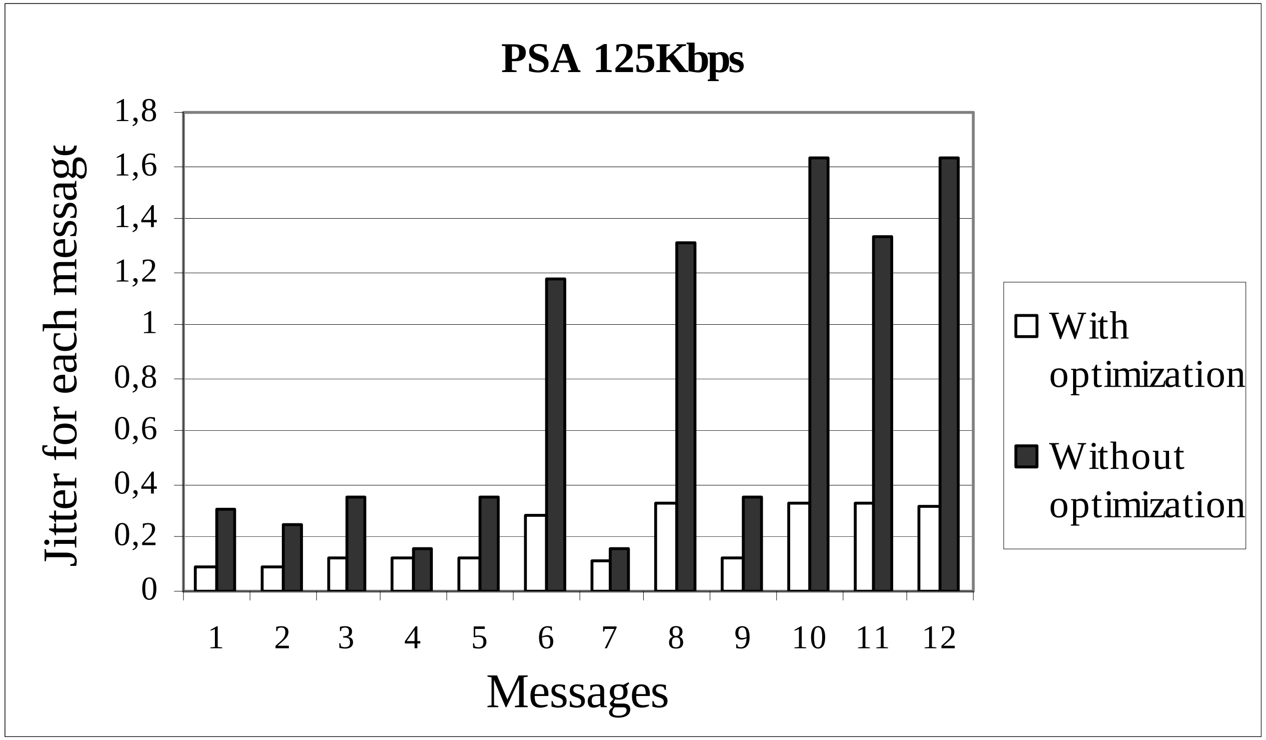
- PSA (12 messages, periods between 10 and 100 ms)
  - 125, 250 and 500 Kbps.
- SAE (53 messages, periods between 5 and 1000 ms)
  - 250 and 500 Kbps.
- 20 optimising runs with 5.000 iterations for SAE and 10.000 for PSA.
- Results compared with a set of 30 solutions obtained without running the optimisation phase.

4 – Simulations and Results

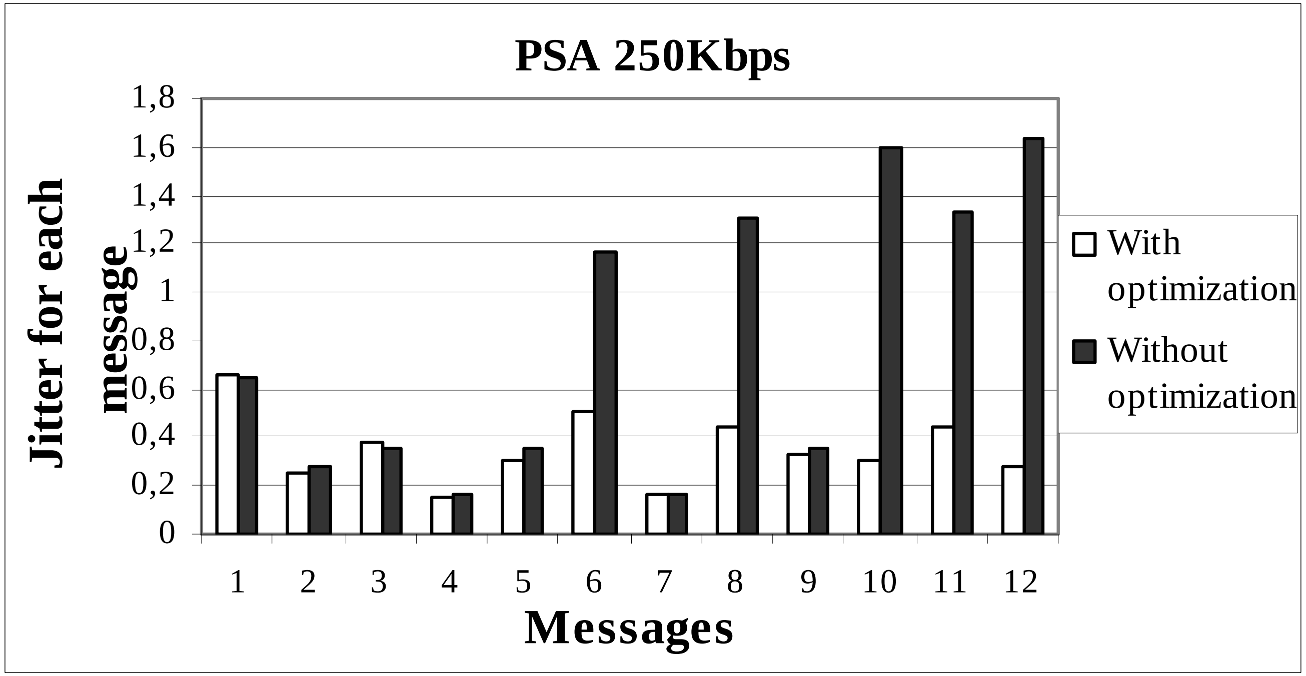
Message Set / Conditions	With Optimisation	Without Optimisation
PSA 125 Kbps	2,838	9,013
PSA 250 Kbps	4,226	9,357
PSA 500 Kbps	4,368	9,401
SAE 250 Kbps	17,225	28,748
SAE 500 Kbps	18,169	29,581

Cost Function Value

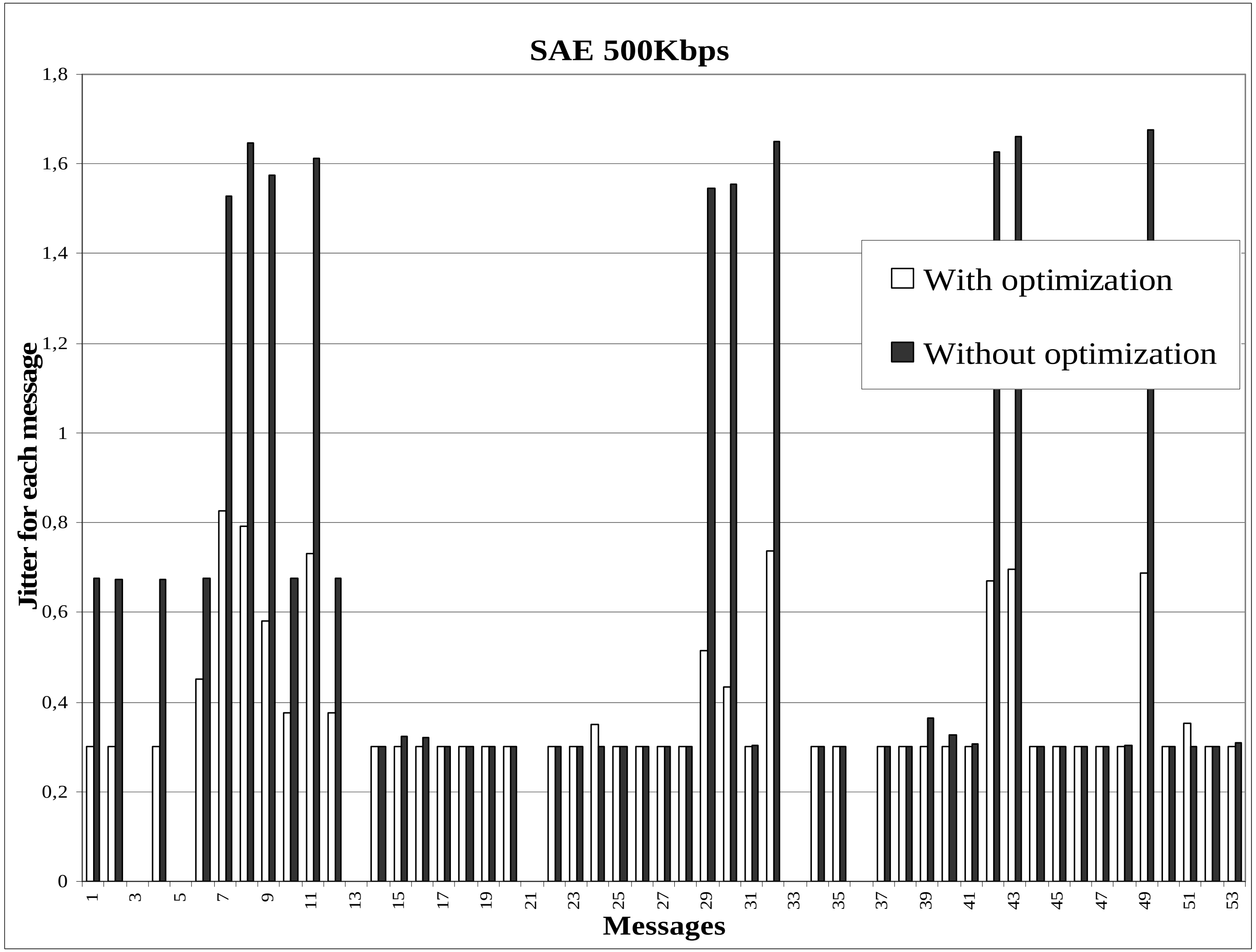
4 – Simulations and Results



4 – Simulations and Results



4 – Simulations and Results



4 – Simulations and Results

- Example with modified PSA benchmark adapted to simplify the building of the System Matrix.

6 – Conclusions

This work led to a first prototype of a scheduling and optimisation tool to build the system matrix for the TTCAN protocol, given a set of periodic messages.

The tool always generates a valid solution with an interesting reduction in the message jitter.

Many issues are still open, such as:

- ✓To compare the results with other optimisation techniques.
- ✓To discover why the optimisation seems to be more effective at lower transmission rates.
- ✓To study the influence of the weight of the matrix transformation techniques in the algorithm performance.
- ✓To investigate more of those techniques.