POSL: A Parallel-Oriented Solver Language

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le sceau de l'Université Bretagne Loire

Alejandro Reyes Amaro

École doctorale : Sciences et technologies de l'information, et mathématiques
Discipline : Informatique et applications, section CNU 27
Unité de recherche : Laboratoire d'Informatique de Nantes-Atlantique (LINA)

Directeur de thèse : M. Eric Monfroy, Professeur, Université de Nantes
Co-encadrant : M. Florian Richoux, Maître de Conferences, Université de Nantes

UNIVERSITÉ DE NANTES

Submitted: dd/mm/2016

Jury:

Président :	**M. Salvador ABREU**, *Professeur étranger*, Université d'Évora
Rapporteurs :	**M. Frédéric LARDEUX**, *Maître de conférences*, Université d'Angers
	M. Christophe LECOUTRE, *Professeur*, Université d'Artois
Examinateur :	**M. Arnaud LALLOUET**, *Chercheur industriel*, Huawei Technologies Ltd.

POSL: A Parallel-Oriented Solver Language

Short abstract:

The multi-core technology and massive parallel architectures are nowadays more accessible for a broad public through hardware like the Xeon Phi or GPU cards. This architecture strategy has been commonly adopted by processor manufacturers to stick with Moore's law. However, this new architecture implies new ways of designing and implementing algorithms to exploit their full potential. This is in particular true for constraint-based solvers dealing with combinatorial optimization problems.

Furthermore, the developing time needed to code parallel solvers is often underestimated. In fact, conceiving efficient algorithms to solve certain problems takes a considerable amount of time. In this thesis we present POSL, a Parallel-Oriented Solver Language for building solvers based on meta-heuristic, in order to solve Constraint Satisfaction Problems (CSP) in parallel. The main goal of this thesis is to obtain a system with which solvers can be easily built, reducing therefore their development effort, by proposing a mechanism of code reusing between solvers. It provides a mechanism to implement solver-independent communication strategies. We also present a detailed analysis of the results obtained when solving some CSPs. The goal is not to outperform the state of the art in terms of efficiency, but showing that it is possible to rapidly prototyping with POSL in order to experiment different communication strategies.

Keywords: Constraint satisfaction, meta-heuristics, parallel, inter-process communication, language.

Contents

1

Bibliography

[1] Daniel Diaz, Florian Richoux, Philippe Codognet, Yves Caniou, and Salvador Abreu. Constraint-Based Local Search for the Costas Array Problem. In *Learning and Intelligent Optimization*, pages 378–383. Springer, 2012.

[2] Danny Munera, Daniel Diaz, Salvador Abreu, and Philippe Codognet. A Parametric Framework for Cooperative Parallel Local Search. In *Evolutionary Computation in Combinatorial Optimisation*, volume 8600 of *LNCS*, pages 13–24. Springer, 2014.

[3] Stephan Frank, Petra Hofstedt, and Pierre R. Mai. Meta-S: A Strategy-Oriented Meta-Solver Framework. In *Florida AI Research Society (FLAIRS) Conference*, pages 177–181, 2003.

[4] Alexander E.I. Brownlee, Jerry Swan, Ender Özcan, and Andrew J. Parkes. Hyperion 2. A toolkit for {meta- , hyper-} heuristic research. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO Comp '14, pages 1133–1140, Vancouver, BC, 2014. ACM.

[5] Alex S Fukunaga. Automated discovery of local search heuristics for satisfiability testing. *Evolutionary computation*, 16(1):31–61, 2008.

[6] El-Ghazali Talbi. Combining metaheuristics with mathematical programming, constraint programming and machine learning. *4or*, 11(2):101–150, 2013.

[7] Jerry Swan and Nathan Burles. Templar - a framework for template-method hyper-heuristics. In *Genetic Programming*, volume 9025 of *LNCS*, pages 205–216. Springer International Publishing, 2015.

[8] Renaud De Landtsheer, Yoann Guyot, Gustavo Ospina, and Christophe Ponsard. Combining Neighborhoods into Local Search Strategies. In *11th MetaHeuristics International Conference*, Agadir, 2015. Springer.

[9] Simon Martin, Djamila Ouelhadj, Patrick Beullens, Ender Ozcan, Angel A Juan, and Edmund K Burke. A Multi-Agent Based Cooperative Approach To Scheduling and Routing. *European Journal of Operational Research*, 2016.

[10] Jordan Bell and Brett Stevens. A survey of known results and research areas for n-queens. *Discrete Mathematics*, 309(1):1–31, 2009.

[11] Rok Sosic and Jun Gu. Effcient Local Search with Conflict Minimization: A Case Study of the N-Queens Problem. *IEEE Transactions on Knowledge and Data Engineering*, 6:661–668, 1994.

Part I

APPENDIX

A

Results of experiments with *Social Golfers Problem*

This Appendix presents graphically a summary of individuals runs using Social Golfers Problem. Figures show a box-plot representation for different strategies and a bar representation for the percentage of winner solvers types.

A.1 Comparison between sequential and parallel runs

Figures A.1, A.2 and A.3 show the huge difference between the spread in results of sequential and parallel runs. Although both parallel strategies (best and first improvement) are equally stable, the one using the selection computation module of first improvement shows better results: all of their runs are under the mean value of sequential runs (except a suspected outlier).

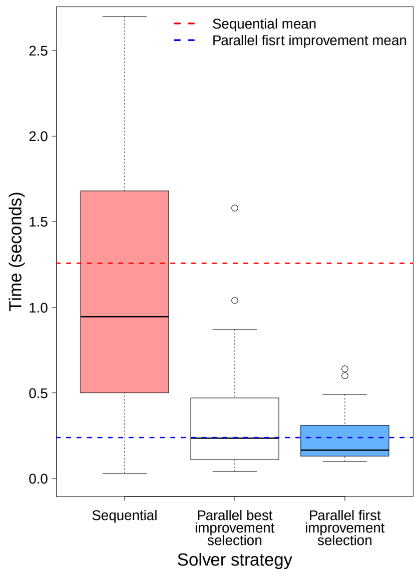


Figure A.1: Comparison between sequential and parallel (best improvement and first improvement selections) runs to solve *SGP* 5-3-7 using POSL

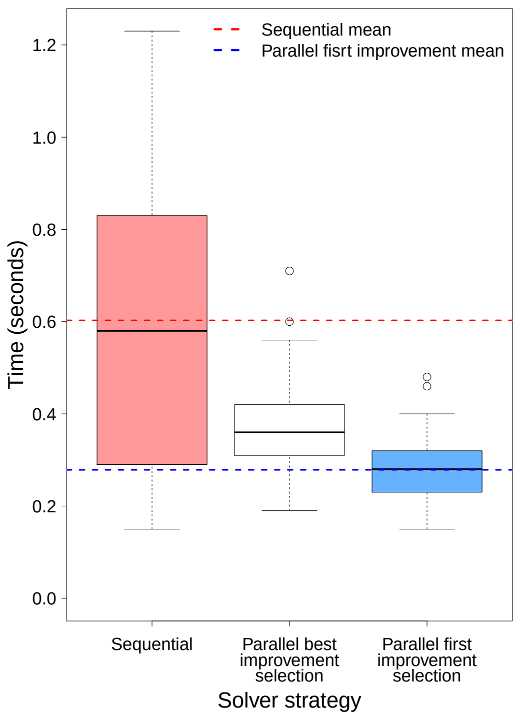


Figure A.2: Comparison between sequential and parallel (best improvement and first improvement selections) runs to solve *SGP* 8-4-7 using POSL

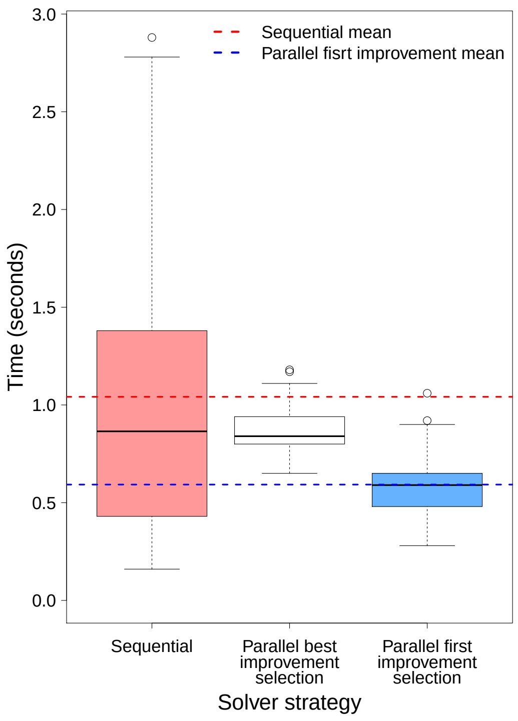


Figure A.3: Comparison between sequential and parallel (best improvement and first improvement selections) runs to solve *SGP* 9-4-8 using POSL

A.2 Comparison between communication strategies

In Figures A.4, A.5 and A.6, labels of the x-axis correspond to the following strategies:

NC:	Non communication strategy
100SC1-1:	100% of communicating solvers performing simple communication one to one
50SC1-1:	50% of communicating solvers performing simple communication one to one
25SC1-1:	25% of communicating solvers performing simple communication one to one
100SC1-n:	100% of communicating solvers performing simple communication one to N
50SC1-n:	50% of communicating solvers performing simple communication one to N
25SC1-n:	25% of communicating solvers performing simple communication one to N
CC1-n:	One set of solvers performing dynamic exchange communication one to N
CC1-n/2:	Two sets of solvers performing dynamic exchange communication one to N
CC1-n/4:	Four sets of solvers performing dynamic exchange communication one to N
100CC1-1:	100% of communicating solvers performing dynamic exchange communication one to one
50CC1-1:	50% of communicating solvers performing dynamic exchange communication one to one
25CC1-1:	25% of communicating solvers performing dynamic exchange communication one to one

Figures A.4, A.5 and A.6 show results summaries of different communication strategies. They are presented together with the box-plot graph of parallel results without communication (`NC`). In all cases we can observe that strategies `100%CC1-1` and `50%CC1-1` show excellent results in comparison to the one without communication. This figure shows also that simple communication strategies do not contribute enough to the improvement of the search time.

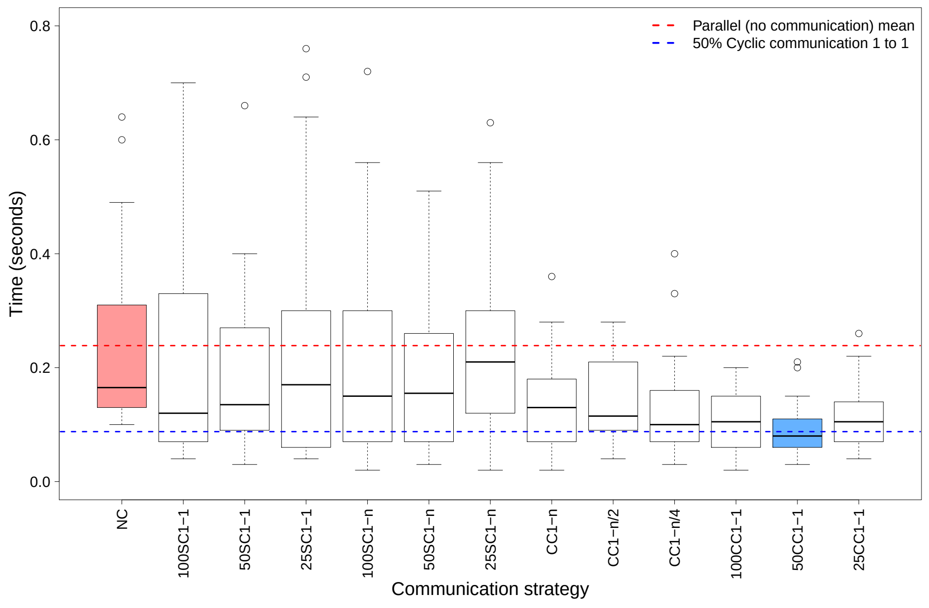


Figure A.4: Different communication strategies to solve *SGP* 5-3-7 using POSL

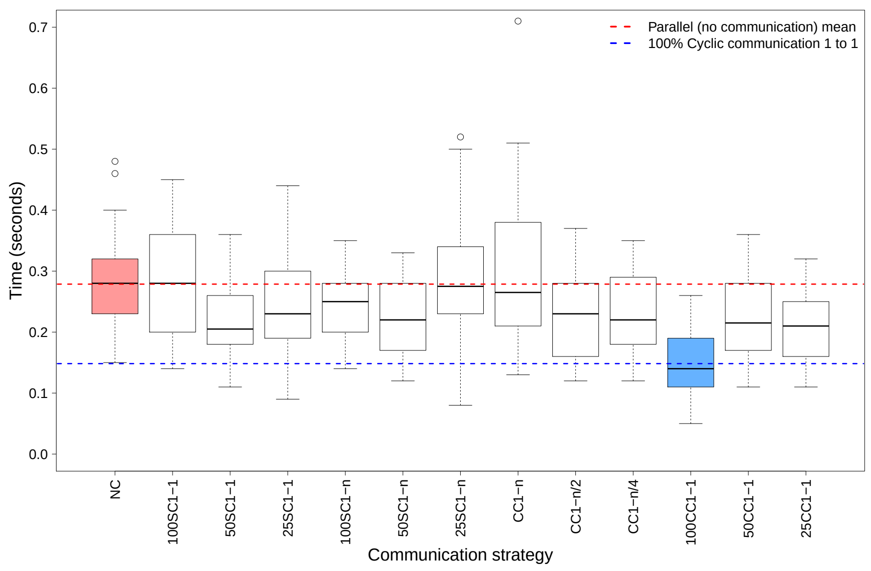


Figure A.5: Different communication strategies to solve *SGP* 8-4-7 using POSL

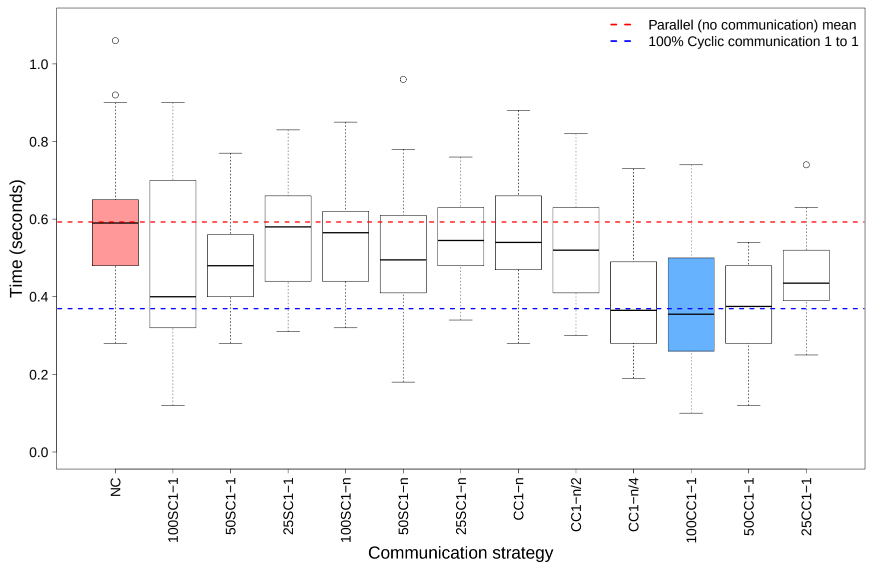


Figure A.6: Different communication strategies to solve *SGP* 9-4-8 using POSL

A.3 Winner solver type representation

Figures A.7, A.8 and A.9, represent the percentage of winner solvers for each communication strategy, according to four different types:

Receiver:	Receiver solver wining thanks to the received information
Sender:	Sender solver
Passive receiver:	Receiver solver wining without using the received information
Non communicating:	Non communicating solver

In these bar graphs is evident to see how the communication has played an important role in the solution process. The number of receiver solvers which have won the search process is high, as expected, performing dynamic exchange strategies with communication one to N. However, due to the huge traffic of information, the communication overhead make the resulting runtimes not competitive.

The other interesting detail showed in these graphs is that this number of winner receiver solvers remains high on winner communication strategies (`100%CC1-1` and `50%CC1-1`), achieving an important trade-off between information traffic and search speed.

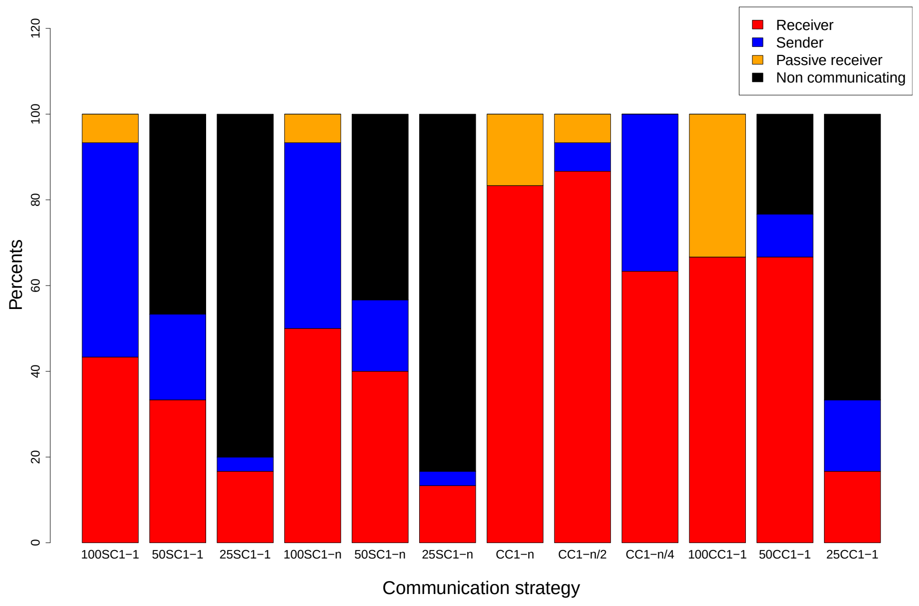


Figure A.7: Solver proportion for each communication strategy to solve *SGP* 5-3-7 using POSL

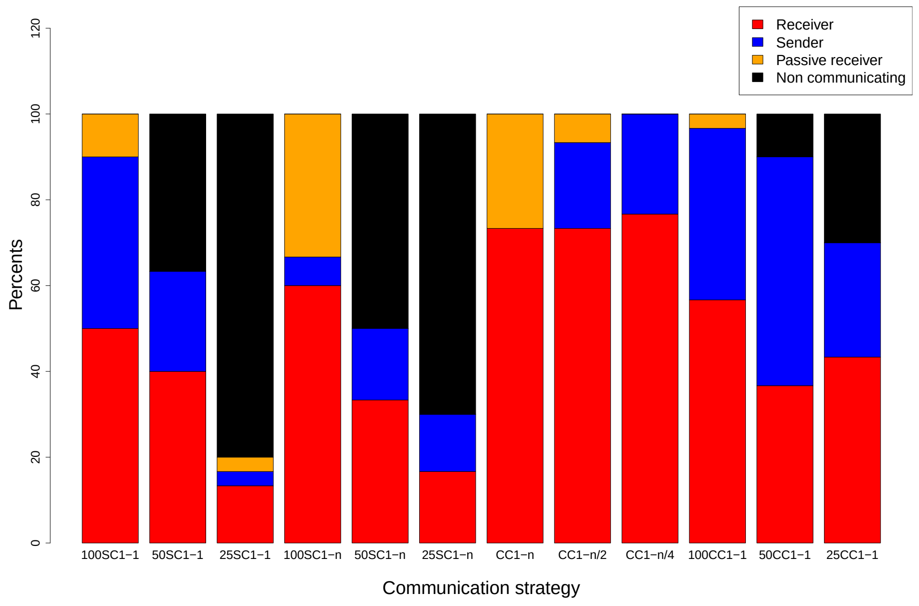


Figure A.8: Solver proportion for each communication strategy to solve *SGP* 8-4-7 using POSL

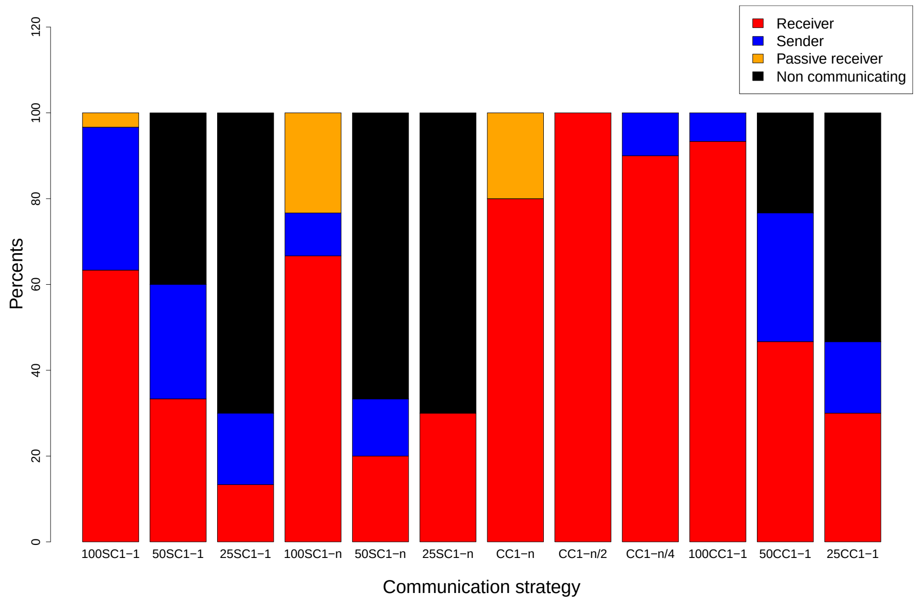


Figure A.9: Solver proportion for each communication strategy to solve *SGP* 9-4-8 using POSL

B

Results of experiments with *N-Queens Problem*

This Appendix, presents graphically a summary of individuals runs using N-Queens Problem. Figures show a box-plot representation for different strategies and a bar representation for the percentage of winner solvers types.

B.1 Comparison between sequential and parallel runs

In Figures B.1, B.2, B.3, B.4 and B.5, labels of the x-axis correspond to the following strategies:

Seq:	Sequential strategy (1 core)
NC:	Parallel non communicative strategy
Cyc1-1:	Cyclic communicating strategy with communication one to one
Cyc1-n:	Cyclic communicating strategy with communication one to N
S1-1:	Simple communicating strategy with communication one to one
S1-n:	One set of solvers performing a simple communicating strategy with communication one to one
S1-n/2:	Two sets of solvers performing a simple communicating strategy with communication one to one
S1-n/4:	Four sets of solvers performing a simple communicating strategy with communication one to one

Figures B.1, B.2, B.3, B.4 and B.5 show summaries of runs from the experimentation with *N-Queens Problem* using box-plot representation. They show sequential runs (`Seq`), parallel without communication runs (`NC`) and runs from the different communication strategies proposed.

As explained before in Chapter **??**, in these graphs can be observed from another point of view how the best found communication strategy contribute less in terms of runtime when the problem order (N: the number of queens) increases. They also show that simple communication strategies do not work for this problem: their results are better (lower runtimes) but not competitive.

For the smaller instance, the cyclic communication strategy work very well. The graph in Figure B.1 shows how for both strategies (`Cyc1-1` and `Cyc1-n`) all runtimes are under the mean of parallel runs without communication (`NC`).

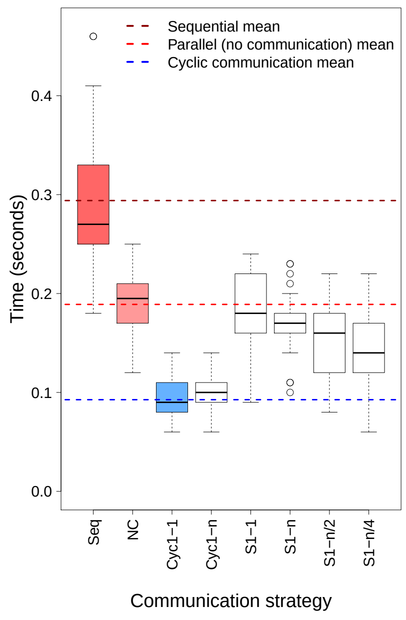


Figure B.1: Different communication strategies to solve 250-Queens using POSL

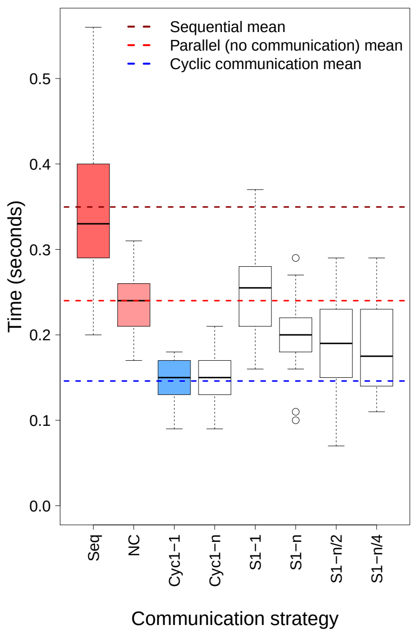


Figure B.2: Different communication strategies to solve 500-Queens using POSL

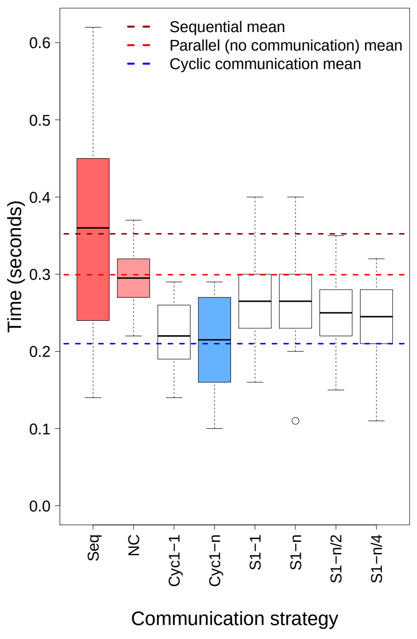


Figure B.3: Different communication strategies to solve 1000-Queens using POSL

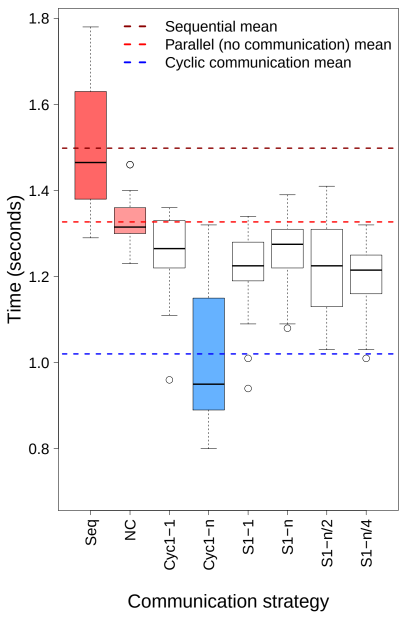


Figure B.4: Different communication strategies to solve 3000-Queens using POSL

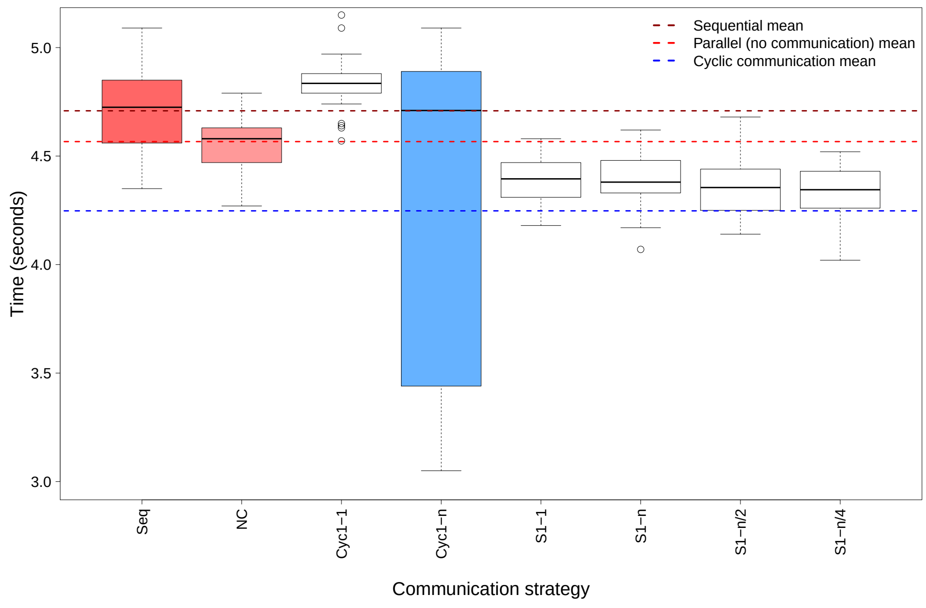


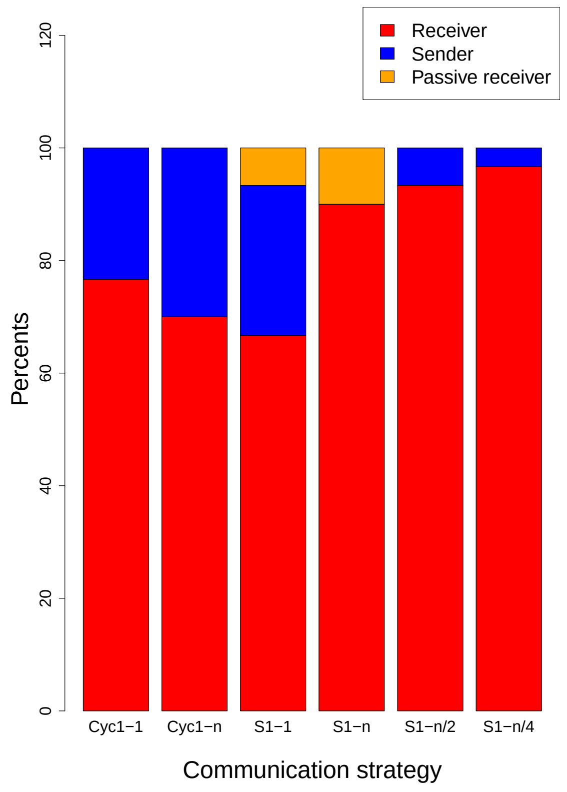
Figure B.5: Different communication strategies to solve 6000-Queens using POSL

B.2 Winner solver type representation

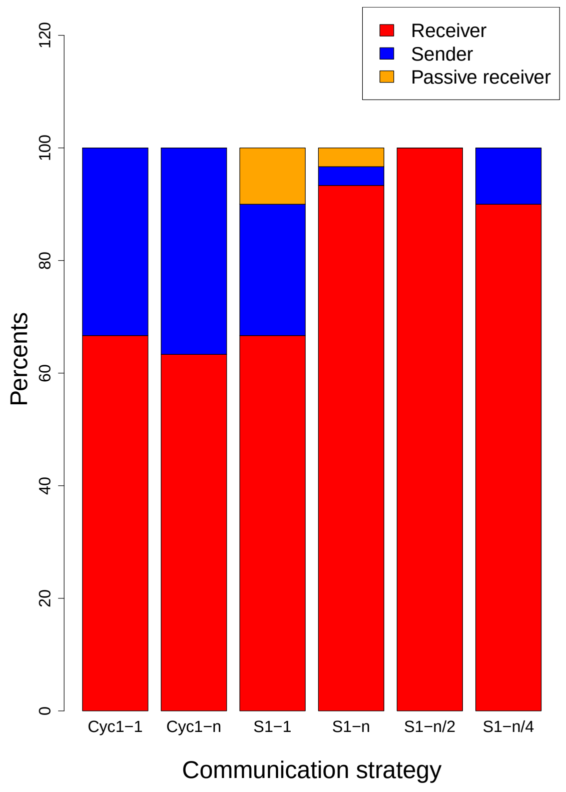
Figures B.6a, B.6b, B.6c, B.6d and B.7, represent the percentage of winner solvers for each communication strategy, according to three different types:

Receiver:	Receiver solver wining thanks to the received information
Sender:	Sender solver
Passive receiver:	Receiver solver wining without using the received information

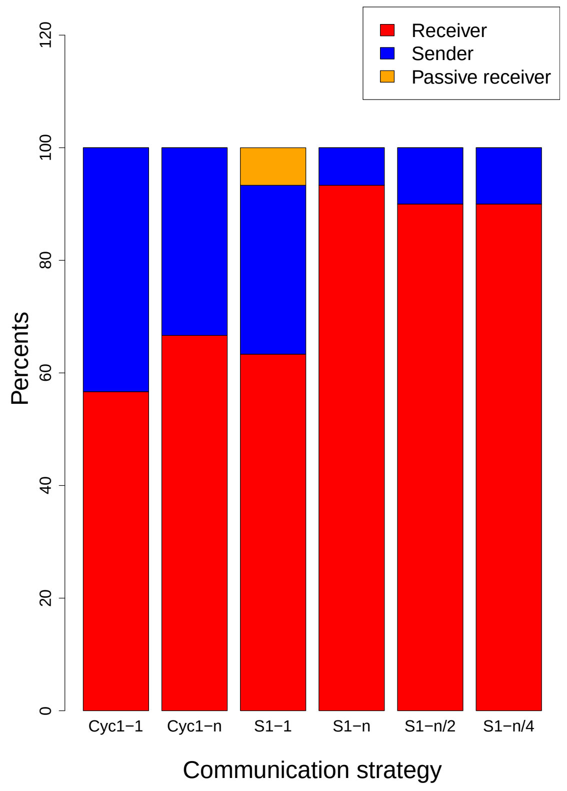
Bar graphs in this section provide a clear evidence that the communication has played an important role during the solution process. They are also a different point of view to observe how the number of winner receiver solvers decrease along with the order of the problem, i.e. the communication is less effective.



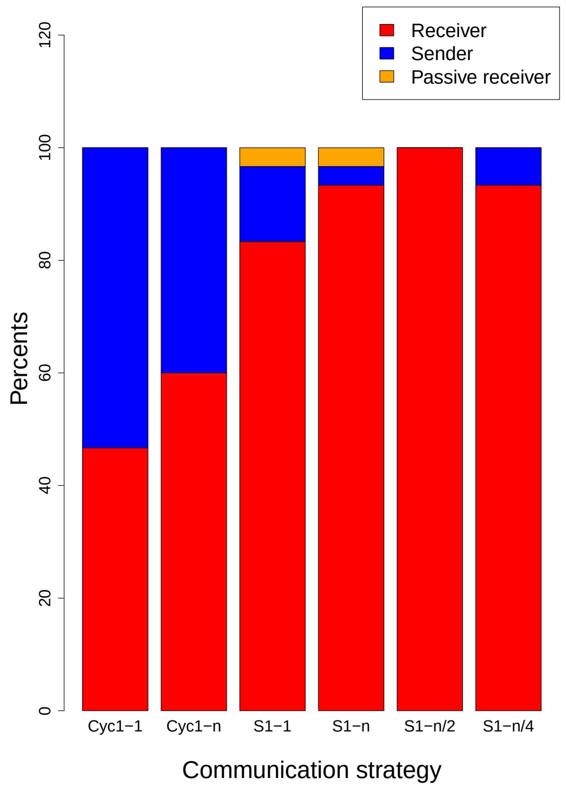
(a) *NQP* 250



(b) *NQP* 500



(c) *NQP* 1000



(d) *NQP* 3000

Figure B.6: Solver proportion for each communication strategy to solve *NQP* using POSL

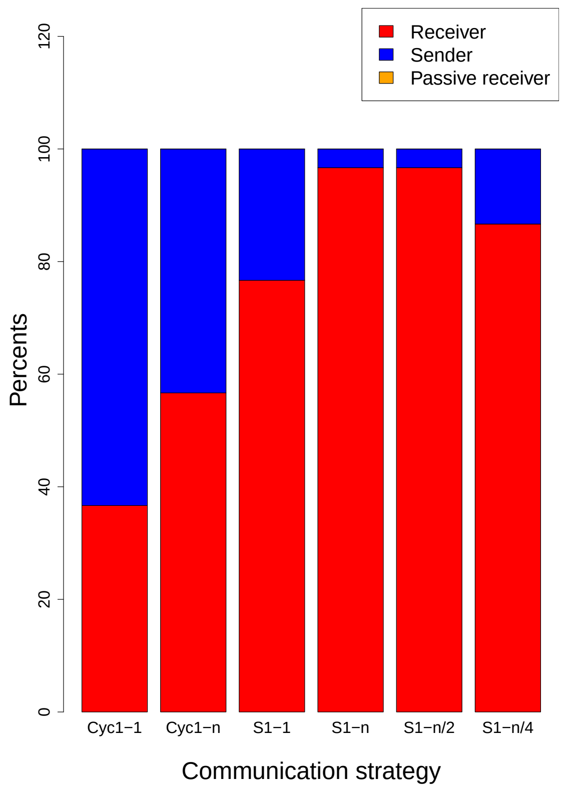


Figure B.7: Solver proportion for each communication strategy to solve *NQP* using POSL

C

Results of experiments with *Costas Array Problem*

This Appendix presents graphically a summary of individuals runs using Costas Array Problem. Figures show a box-plot representation for different strategies and a bar representation for the percentage of winner solvers types.

C.1 Comparison between sequential and parallel runs

Figure C.1 shows that the approach in parallel largely outperforms the sequential one. this is because this problem is very sensitive to the starting point. In the graph we can observe that all parallel runs are far below the mean of sequential runs. In addition, the minimum sequential runtime is higher that the 50% of the parallel runs.

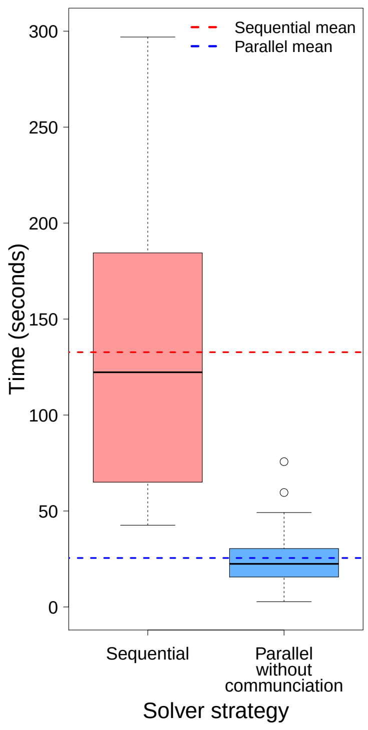


Figure C.1: Comparison between sequential and parallel runs to solve *CAP* 19 using POSL

C.2 Comparison between communication strategies

In Figure C.2, labels of the x-axis correspond to the following strategies:

NC:	Non communication strategy
A1-1:	100% of communicating solvers performing the communication strategy A one to one
B1-1:	100% of communicating solvers performing the communication strategy B one to one
A1-n:	100% of communicating solvers performing the communication strategy A one to N
B1-n:	100% of communicating solvers performing the communication strategy B one to N
50A1-1:	50% of communicating solvers performing the communication strategy A one to one
50B1-1:	50% of communicating solvers performing the communication strategy B one to one
50A1-n:	50% of communicating solvers performing the communication strategy A one to N
50B1-n:	50% of communicating solvers performing the communication strategy B one to N

Figure C.2 shows that communication strategies of type `A` provide better and more stable results: the lower quartile value of parallel runs (`NC`) is higher that the upper quartile value of both communication strategies (`A1-1` an `A1-n`)

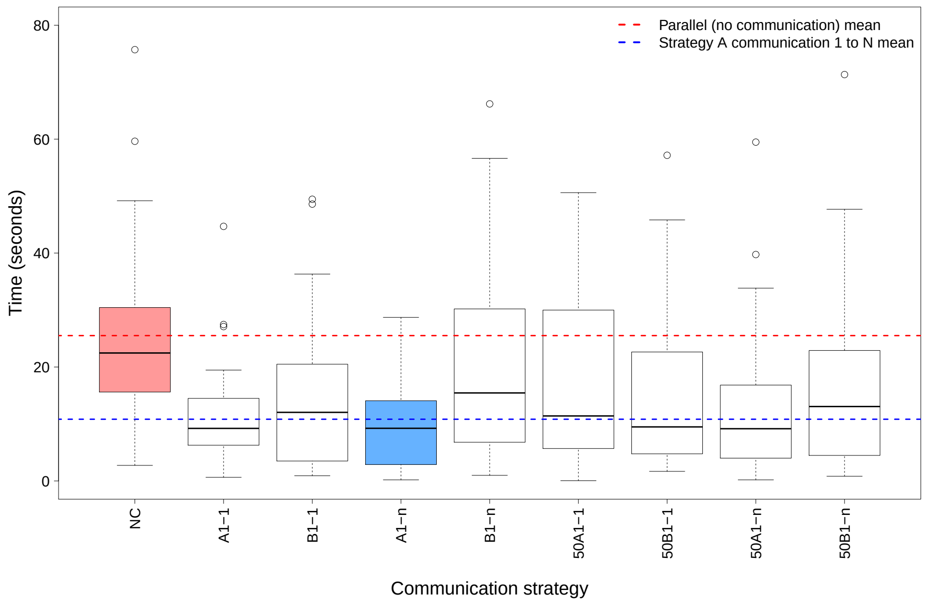


Figure C.2: Different communication strategies to solve *CAP* 19 using POSL

C.3 Winner solver type representation

Figure C.3 represents the percentage of winner solvers for each communication strategy, according to three different types:

Receiver:	Receiver solver wining thanks to the received information
Sender:	Sender solver
Non communicating:	Non communicating solver

The bars graph in Figure C.3 is self-explained: percentages of winner communicating solvers match with percentage of communicating solvers participating in the respective communication strategy.

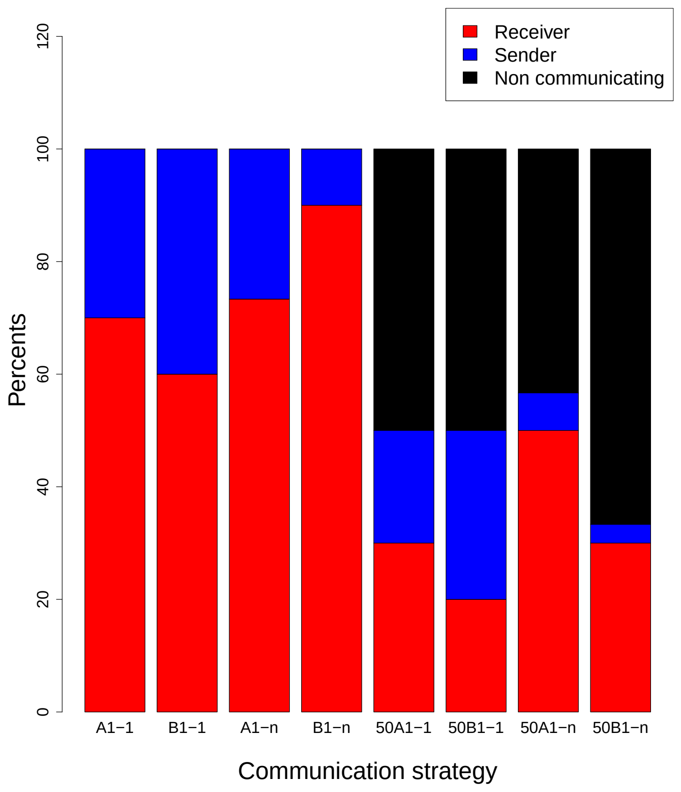


Figure C.3: Solver proportion for each communication strategy to solve *CAP* 19 using POSL

D

Results of experiments with *Golomb Ruler Problem*

This Appendix, presents graphically a summary of individuals runs using Golomb Ruler Problem. Figures show a box-plot representation for different strategies and a bar representation for the percentage of winner solvers types.

D.1 Comparison between sequential and parallel runs

In graphs showed in Figures D.1, D.2 and D.3 it is visible that the parallel approach outperforms the sequential one (in Figures D.1 and D.2 the difference is abysmal). In Figure D.3 this difference is less pronounced because the instance of the problem is more complex.

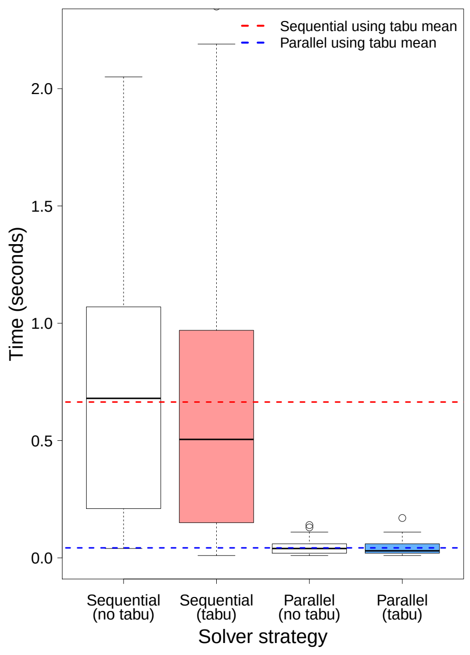


Figure D.1: Comparison between sequential and parallel runs to solve *GRP* 8-34 using POSL

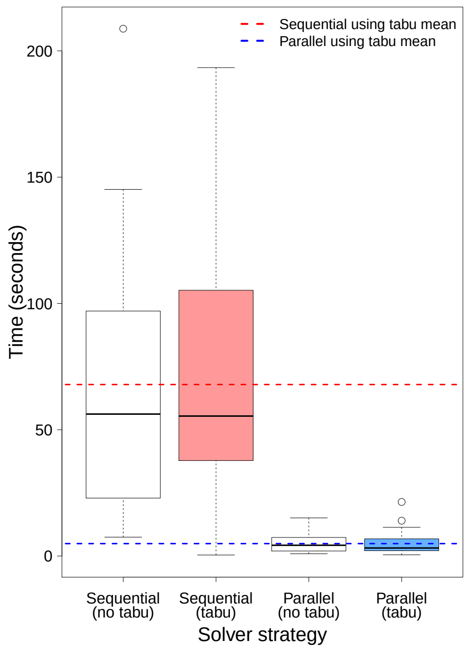


Figure D.2: Comparison between sequential and parallel runs to solve *GRP* 10-55 using POSL

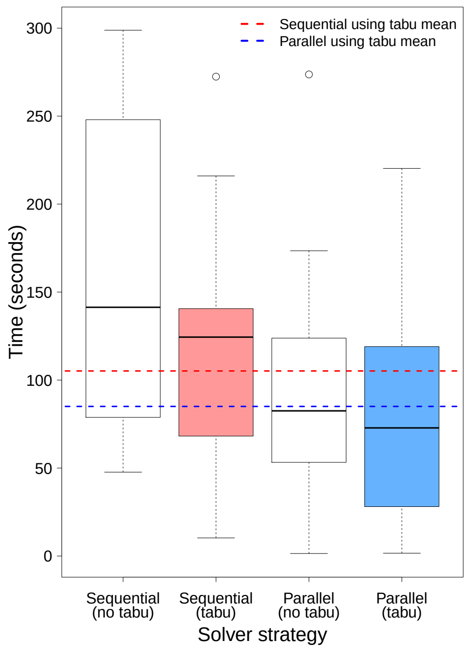


Figure D.3: Comparison between sequential and parallel runs to solve *GRP* 11-72 using POSL

D.2 Comparison between communication strategies

In Figures D.4, D.5 and D.6, labels of the x-axis correspond to the following strategies:

NC(noT):	Non communication strategy without using tabu list
NC(T):	Non communication strategy using tabu list
C1-1:	Communicating solvers performing communication one to one
C1-n:	Communicating solvers performing communication one to N

Figure D.4 does not show any improvement when applying the communicative approach because the instance is very easy to solve. Along with the problem complexity growth, the communication begins to bear its fruits: runtimes are visibly improved with respect to the non co-operative approach (see Figures D.5 and D.6).

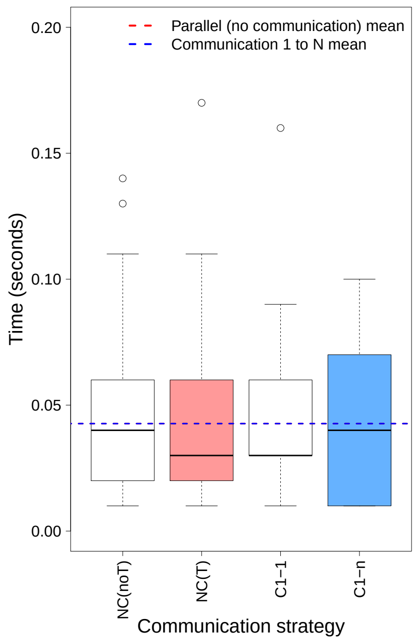


Figure D.4: Different communication strategies to solve *GRP* 8-34 using POSL

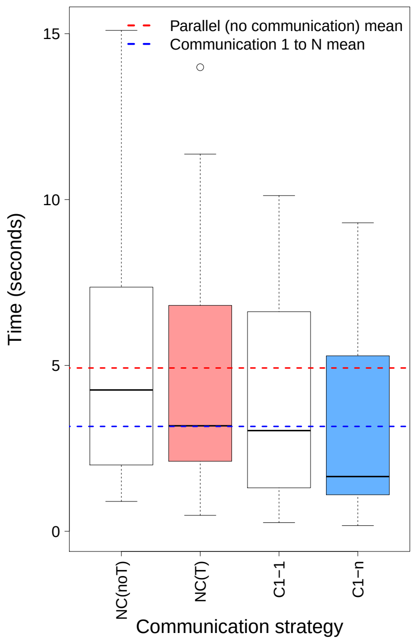


Figure D.5: Different communication strategies to solve *GRP* 10-55 using POSL

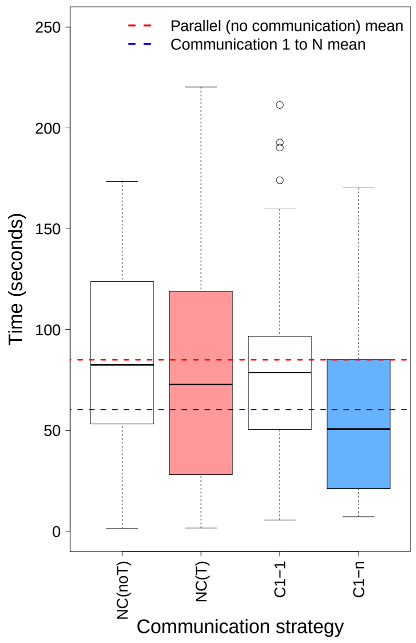


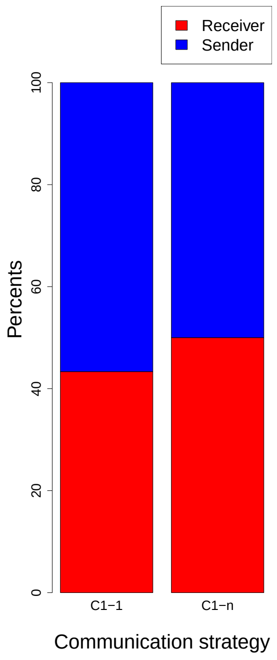
Figure D.6: Different communication strategies to solve *GRP* 11-72 using POSL

D.3 Winner solver type representation

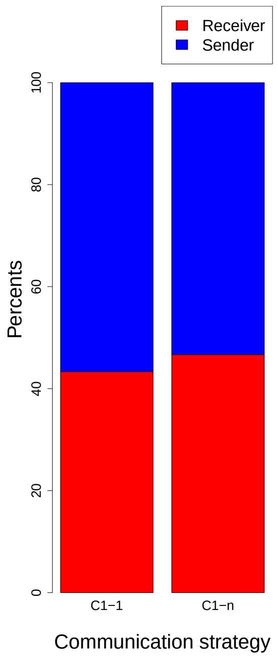
Figures D.7a, D.7b and D.7c, represent the percentage of winner solvers for each communication strategy, according to two different types:

Receiver:	Receiver solver wining thanks to the received information
Sender:	Sender solver

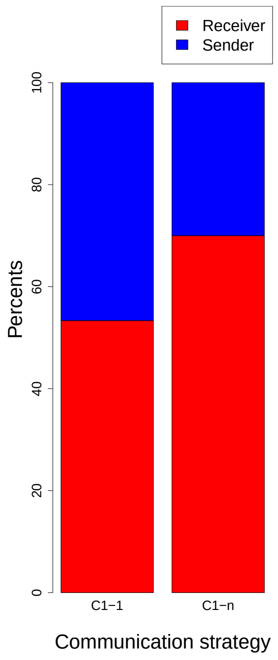
Bar graphs in Figure D.7 are another point of view to show how the communication becomes more important as long as the problem order grows: the percentage of winner receiver solvers increases along with the problem order.



(a) *GRP* 8-34



(b) *GRP* 10-55



(c) *GRP* 11-72

Figure D.7: Solver proportion for each communication strategy to solve *GRP* using POSL