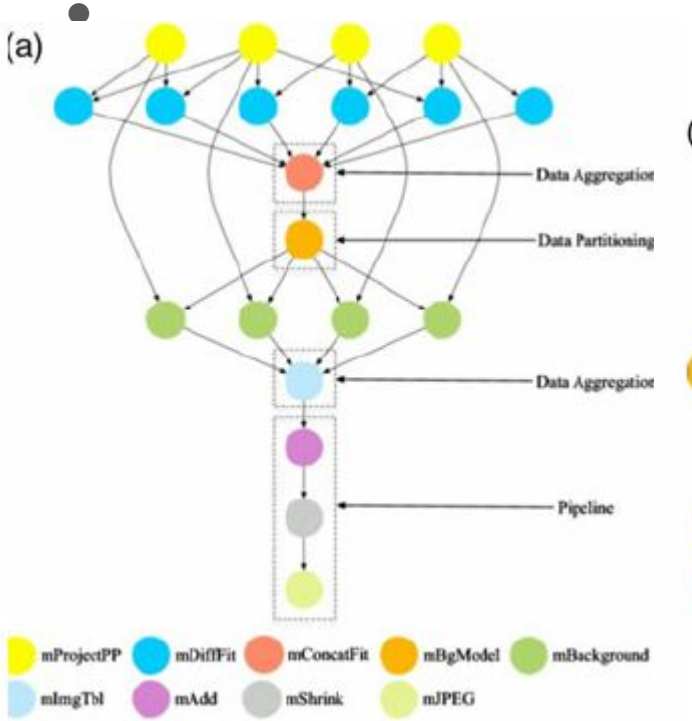
Performability Evaluation and Optimization of Workflow Applications in Cloud Environments

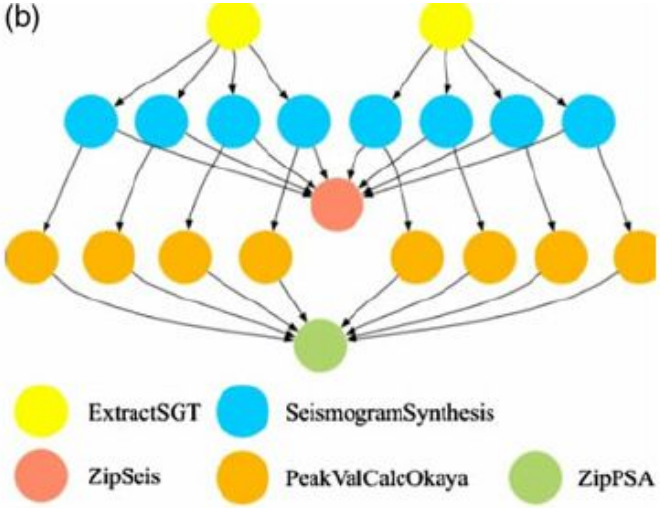
Danilo Oliveira
Orientador: Paulo Maciel
Co-Orientador: Nelson Rosa

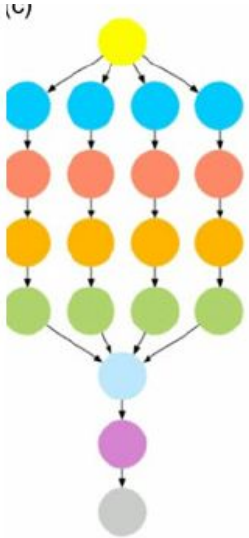
Roteiro

- Introdução
- Metodo de otimização de performabilidade para escalonamento de workflow
- Modelo de performabilidade
- Avaliação e resultados
- Conclusões

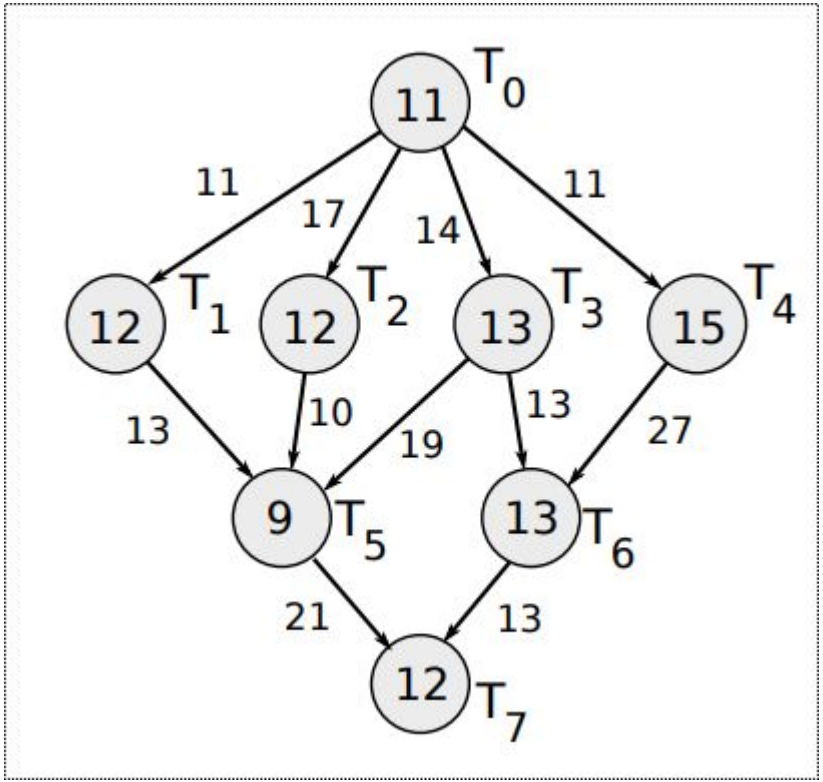
Scientific Workflow Applications

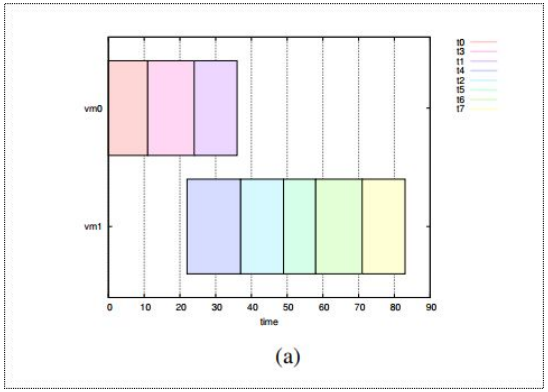




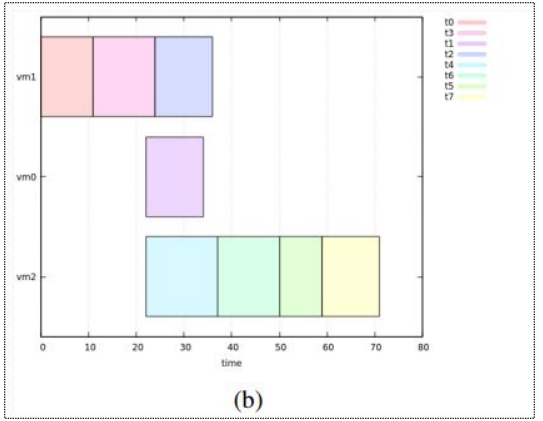


Scientific Workflow Applications - scheduling





(a)



(b)

Método de otimização

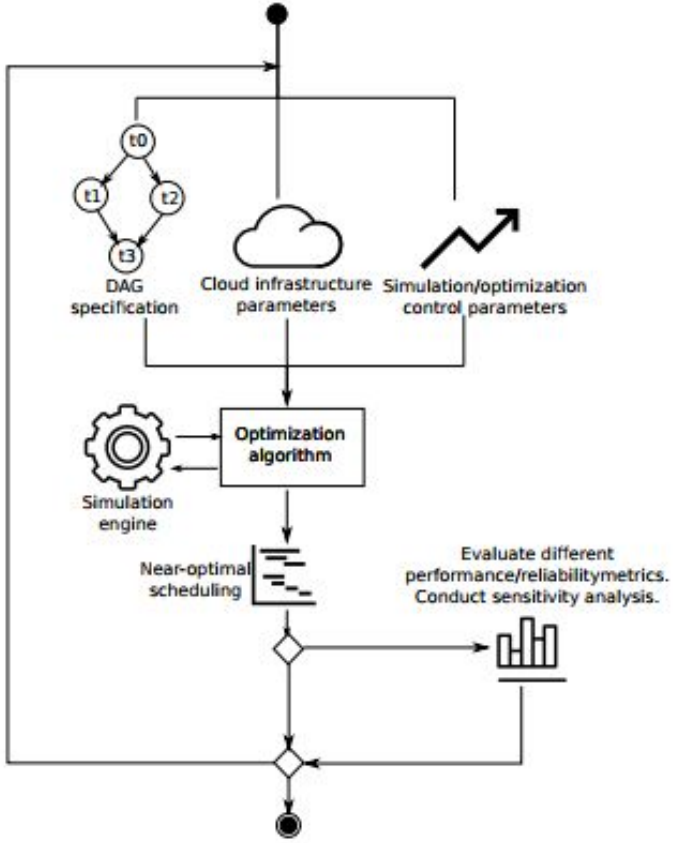
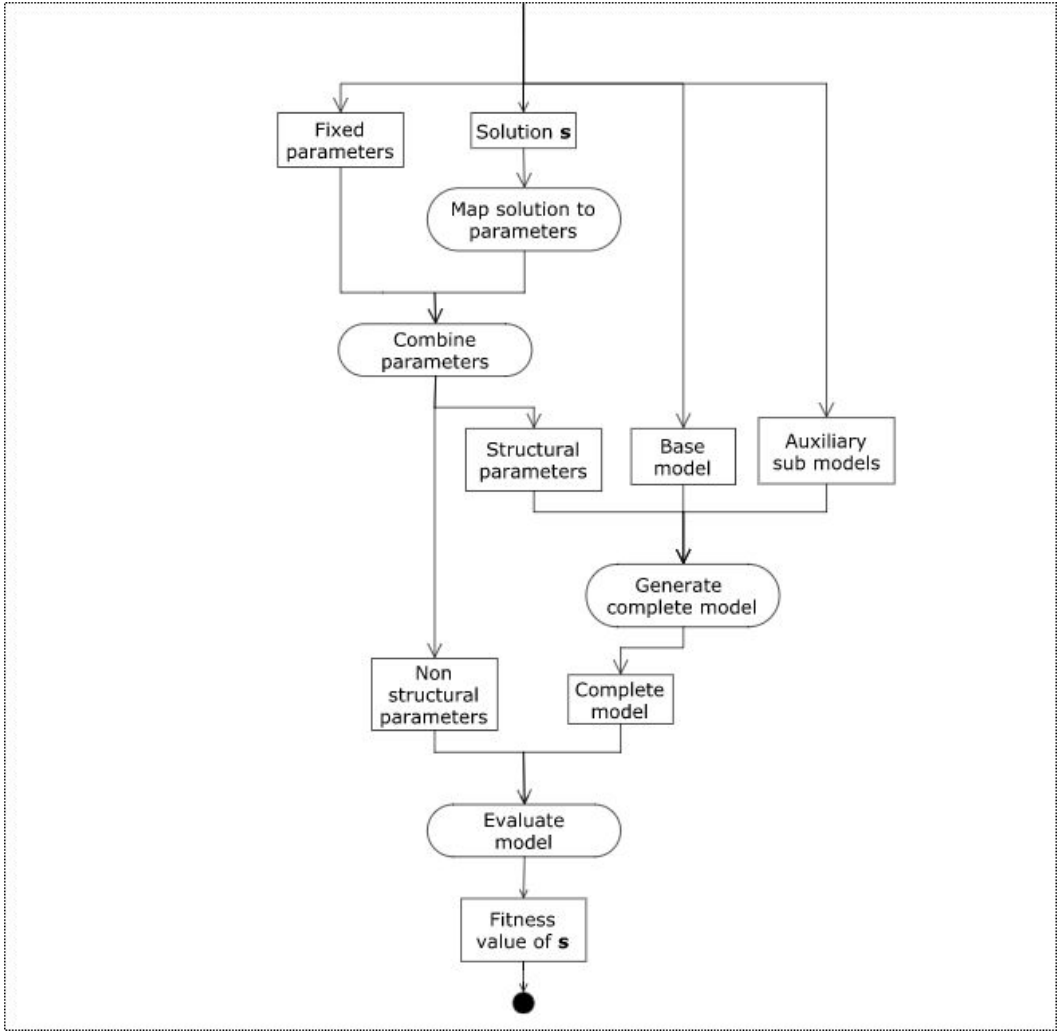


Fig. 3. Overview of the proposed method

Método de otimização



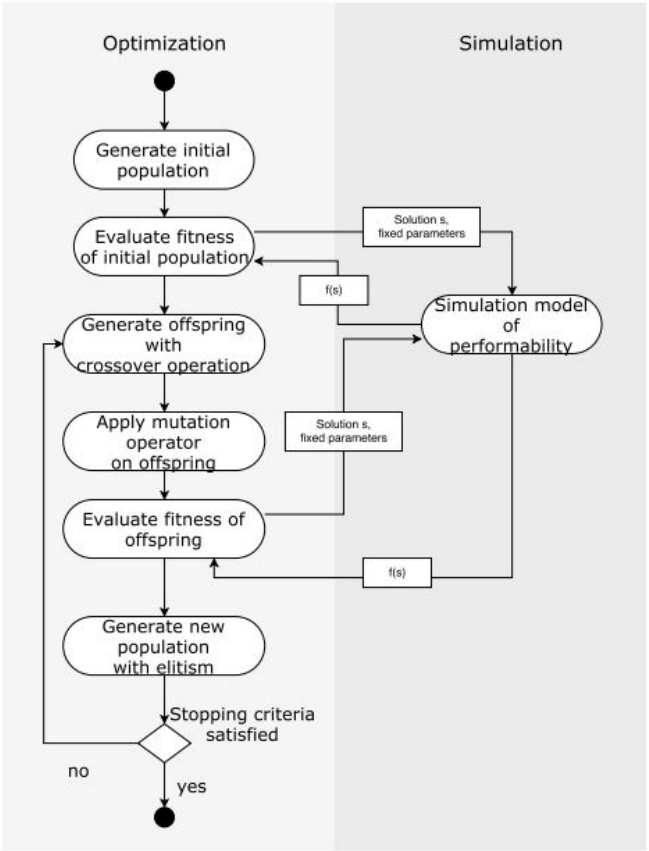


Fig. 4. Genetic algorithm with a stochastic fitness function

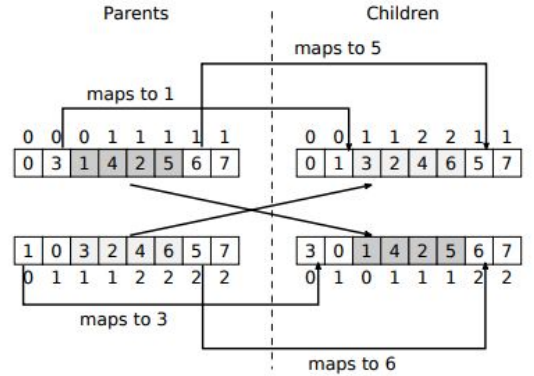


Fig. 6. PMX crossover operator

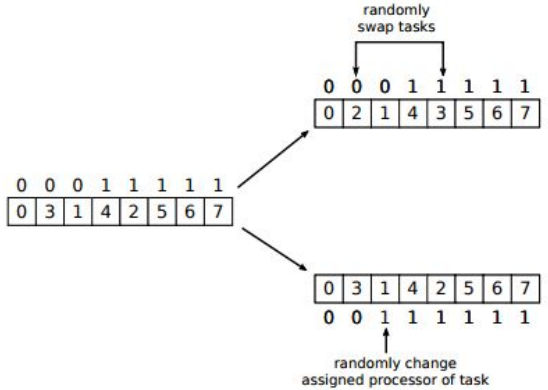


Fig. 7. Mutation operator

Modelo de simulação

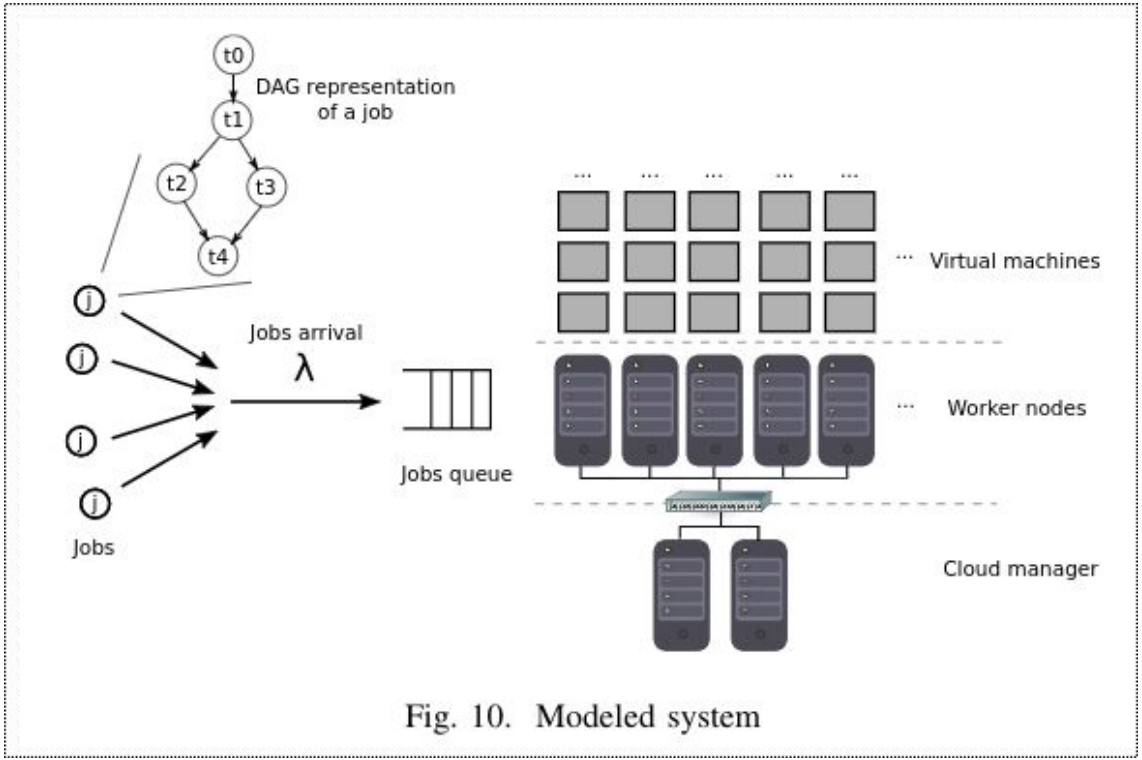
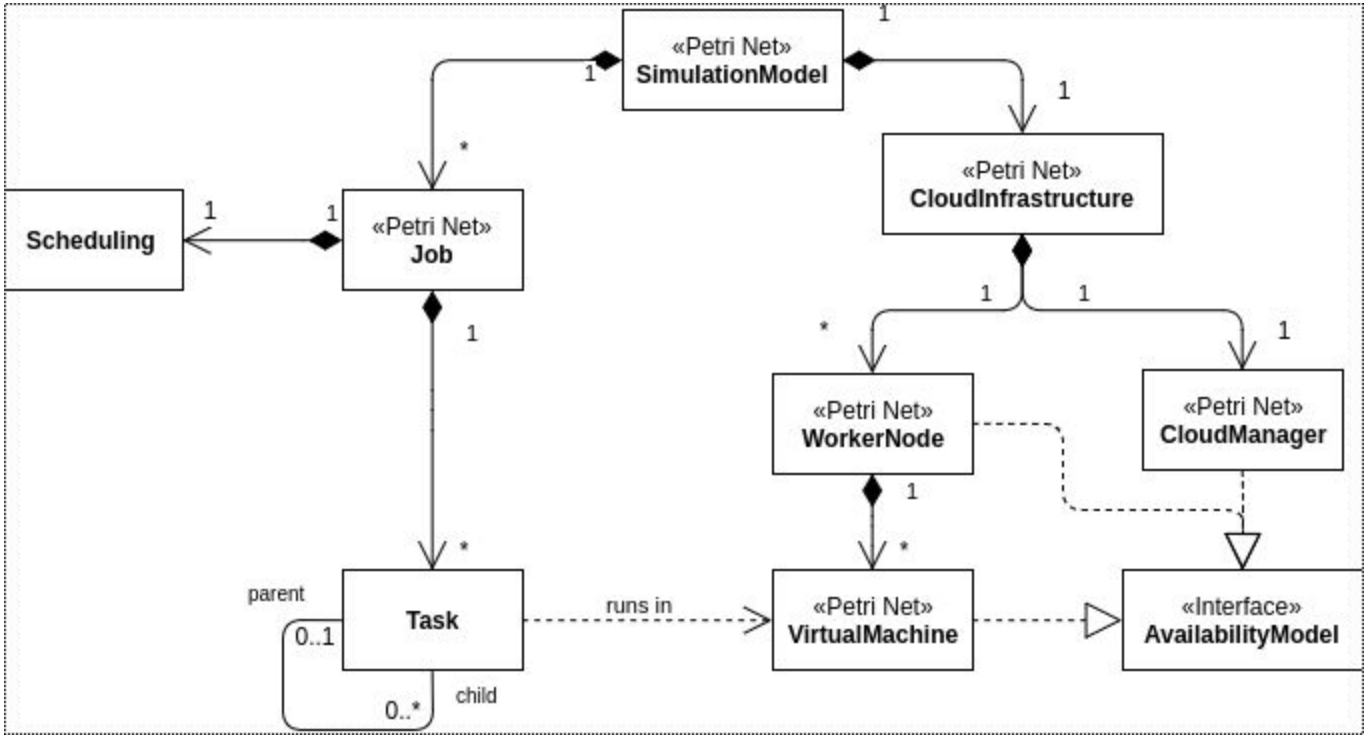


Fig. 10. Modeled system

Modelo de simulação



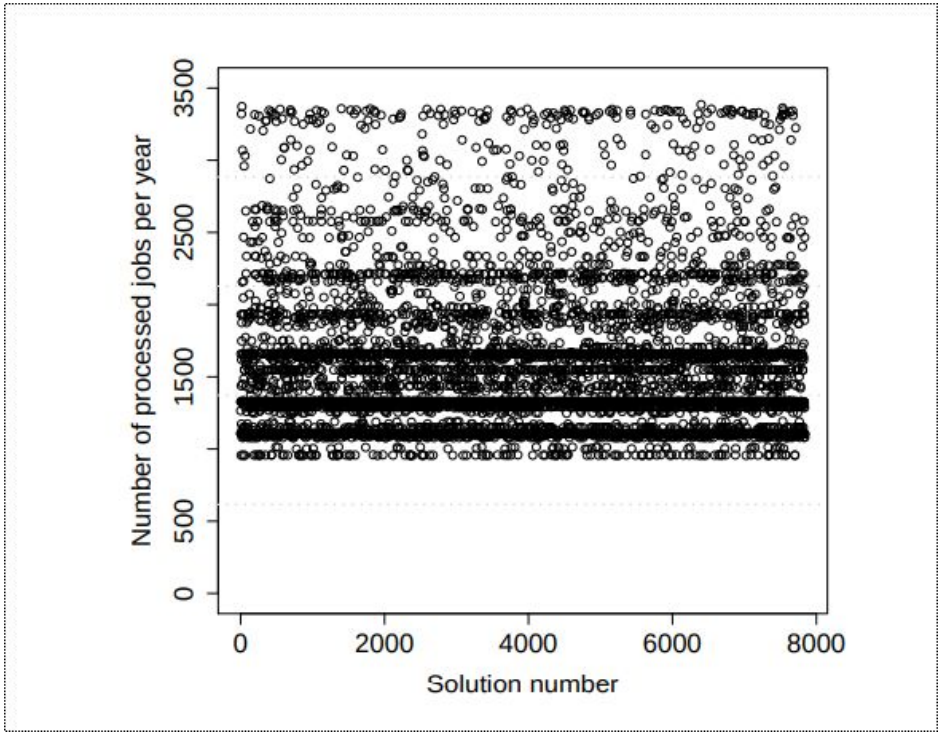
```
OPN JobNet extends opn.JobNet {

    place pResults extends ResultsPlace;
    place pCompletedTasks extends CompletedPlace
    place pFailedTasks extends FailedPlace;

    for i in range( 0, this.processors - 1 ){

        place pRunningTask#($i) ( capacity = 1
        place pSendingResults#($i);
        place pTasks#($i);

        immediateTransition scheduleTask#($i)(
            filters = [ t in pTasks#($i) | ? t.
            inputs = [pTasks#($i)(t)],
            outputs = [pRunningTask#($i)(t)]
        );

        timedTransition executeTask#($i) extends
            inputs = [pRunningTask#($i)(t)],
            outputs = [pSendingResults#($i)(), p
        );

        timedTransition sendResults#($i) extends
            inputs = [pSendingResults#($i)(t)],
            outputs = [pResults(t)],
            serverType = InfiniteServer
        );
```

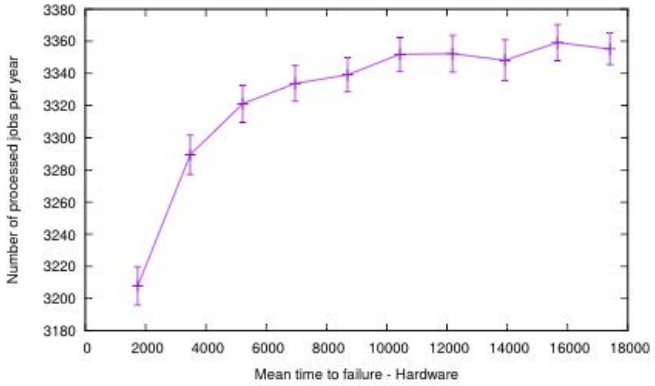
```
public class JobNet extends OPNNet {

    private DAG dag;
    private Scheduling scheduling;
    private ResultsPlace pResults;
    private List<Set<String>> allocatedTasks;
    private boolean success = false;
    private boolean failure = false;
    private long id;
    private List<List<String>> executedTasks;

    @Override
    public void preCreate(Dictionary parameters) {
        setAttribute("processors", parameters.getDoubleValue(

        initTransitionsFactories();
        initPlaceFactories();
    }

    @Override
    public void posCreate() {
        pResults = (ResultsPlace) getPlace("pResults");
    }

    @Override
    public boolean checkProperty(String property, Dictionary

        boolean terminated = isTerminated();

        if (null != property) {
            switch (property) {
                case "success":
                    return terminated && success;
                case "failure":
                    return terminated && failure;
                case "isUp":
                    return isProcessorUp(dict);
```

Avaliação - força bruta

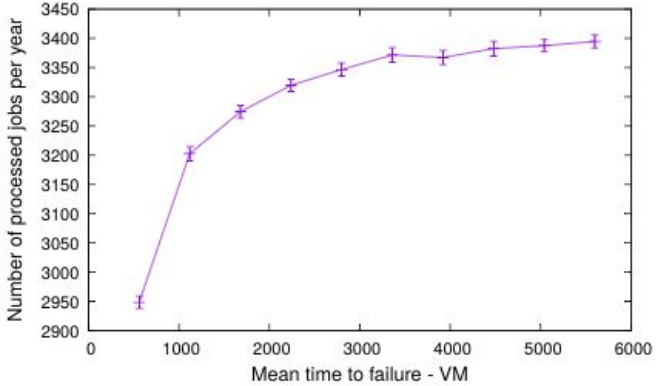
TABLE II
MODEL PARAMETERS

Parameter	Value
Mean time to failure - physical machine	8760 h
Mean time to failure - idle p. machine	13140 h
Mean time to failure - virtual machine	2880 h
Mean time to repair - physical machine	1 h
Mean time to repair - virtual machine	1 h
Arrival rate of jobs	1/2.5 (1/h)
Number of workers	20
VMs per worker	3
Activation time of standby server	0.004 (h)
Number of replications for the simulation	30

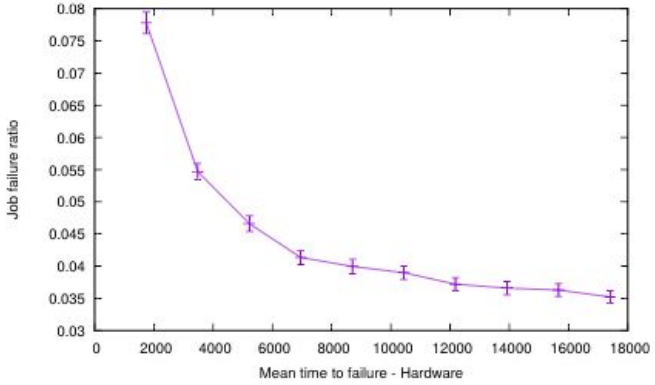




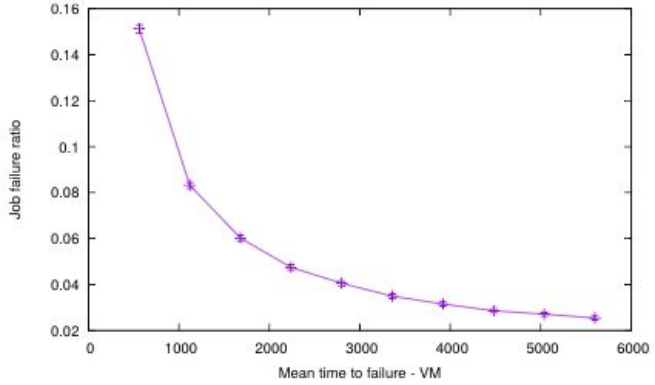
(a) MTTF Hardware × Number of processed jobs



(b) MTTF VM × Number of processed jobs



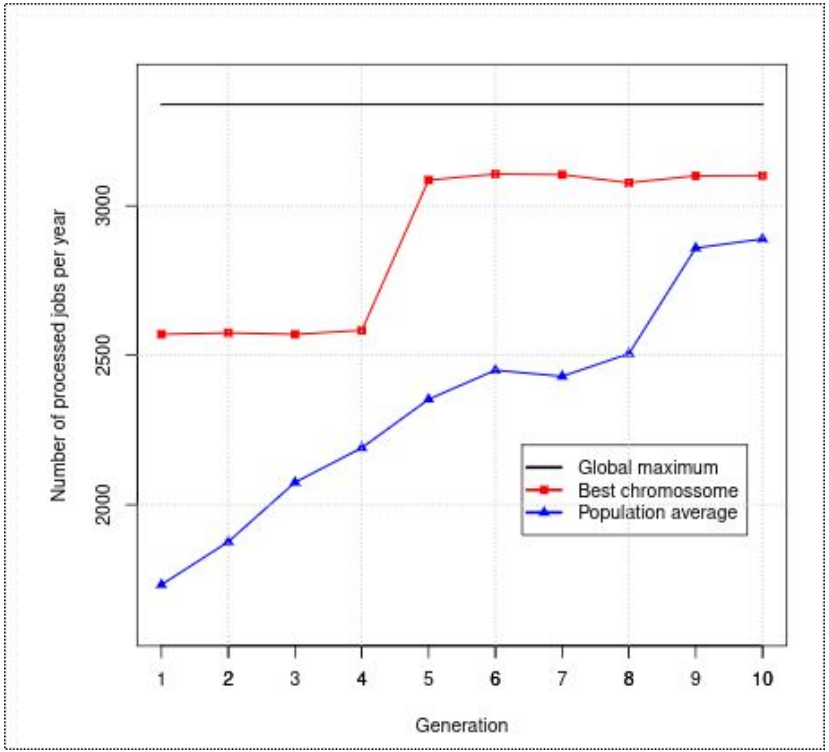
(c) MTTF Hardware × Job failure ratio



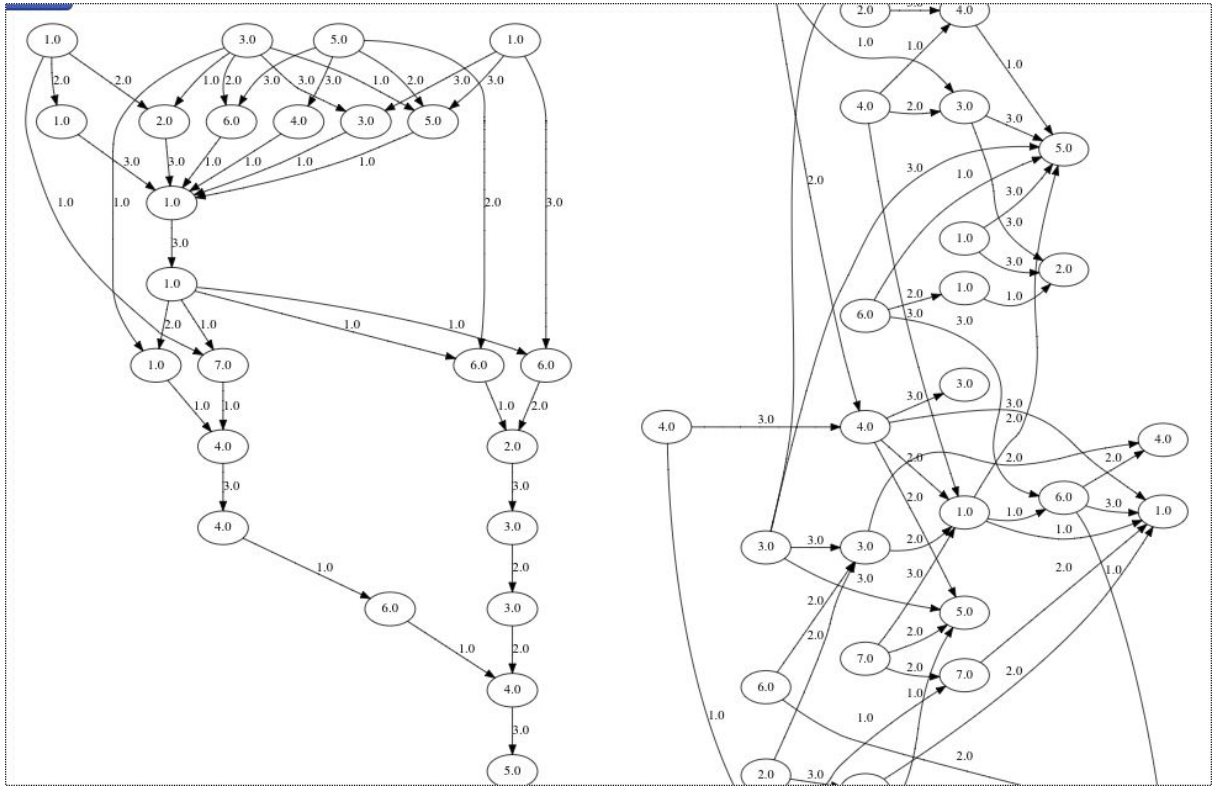
(d) MTTF VM × Job failure ratio

Fig. 13. Sensitivity analysis - one factor at time

Avaliação - Algoritmo genético



Segundo estudo de caso



Considerações finais

- O algoritmo de otimização se mostrou eficiente em otimizar o escalonamento ao mesmo tempo do número de máquinas a fim de maximizar a vazão do sistema
- Stiffness é um problema sério para simulação de performabilidade de sistemas de grande escala. Técnicas de simulação paralela SRIP (Single Replication In Parallel) podem ser empregadas para contornar esse problema.
- Trabalhos futuros incluirão aspectos de heterogeneidade e contenção de recursos, bem como estratégias de tolerância à falhas (checkpointing e redundância espacial)