



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Integração de Ferramenta de Análise de Sensibilidade ao Simulador Ambiental MASE-BDI

Murilo Santos Tavares de Sousa

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientadora
Prof.^a Dr.^a Célia Ghedini Ralha

Brasília
2016

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Henrique Veneziano

Banca examinadora composta por:

Prof.^a Dr.^a Célia Ghedini Ralha (Orientadora) — CIC/UnB
Prof.^a Dr.^a Maria Emilia Machado Telles Walter — CIC/UnB
Carolina Gonçalves Abreu — Mestre CIC/UnB

CIP — Catalogação Internacional na Publicação

Sousa, Murilo Santos Tavares de.

Integração de Ferramenta de Análise de Sensibilidade ao Simulador Ambiental MASE-BDI / Murilo Santos Tavares de Sousa. Brasília : UnB, 2016.

63 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2016.

1. sistemas de informação, 2. análise de sensibilidade, 3. sistema multiagente, 4. simulação ambiental

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedico este trabalho especialmente à minha esposa, que sempre me apoiou desde o início dos meus estudos em Computação e também no meu trabalho, mesmo quando isso significava passar mais um dia em casa. Ela me faz querer mais da vida e por ela sempre vou ter um eterno carinho.

Agradecimentos

Agradeço muito a Profa. Dra. Célia, uma professora que me marcou muito positivamente tanto no âmbito profissional quanto no âmbito pessoal, é uma profissional exemplar na qual me espelho. Agradeço a Carolina Abreu por ter dedicado tanto do seu tempo para me ajudar e orientar nesse meu último esforço na minha graduação. Por fim, agradeço a minha família e meus amigos por terem me apoiado durante todo meu caminho na universidade, desde o primeiro curso que passei na UnB e até o dia de hoje.

Resumo

A integração de sistemas tem suma importância para uma melhoria contínua de processos e ferramentas. Para realizar tal integração, é necessário que se passe por um processo de identificação do problema, para criar uma proposta de solução que atenda as necessidades de todos os sistemas envolvidos. Este trabalho visa integrar uma ferramenta de análise de sensibilidade e quantificação de incertezas, o PSUADE, ao simulador ambiental MASE-BDI. Através da identificação das especificações de cada ferramenta, foi possível obter sucesso na definição arquitetural e implementação de uma ferramenta que realiza tal integração. Este trabalho viabiliza a execução do PSUADE integrado com o MASE-BDI para fornecer resultados de avaliação de análise de sensibilidade.

Palavras-chave: sistemas de informação, análise de sensibilidade, sistema multiagente, simulação ambiental

Abstract

Integration between systems is very important for continuous improvement of processes and tools. To accomplish such integration, it is necessary to first identify an existing problem and then propose a solution that meets all the specifications of the systems involved. This work aims to integrate a sensitivity analysis tool and quantification of uncertainties, PSUADE, with the environmental simulation tool MASE-BDI. By identifying the specifications of each tool, it was possible to successfully define an architecture and to implement a tool that accomplishes the integration. This work enables PSUADE execution integrated with MASE-BDI providing sensitivity analysis results.

Keywords: information systems, sensitivity analysis, multi-agent systems, environmental simulation

Sumário

1	Introdução	1
1.1	O Projeto MASE	2
1.2	Questão de Pesquisa	3
1.3	Objetivos	3
1.4	Estrutura do Documento	3
2	Fundamentação Teórica	4
2.1	Sistemas de Informação	4
2.2	Sistemas Multiagentes	5
2.3	Modelo de Racionalidade BDI	6
2.4	O Projeto MASE	8
2.5	Estudo de Caso: Cerrado DF	10
2.6	Análise de Sensibilidade	11
2.7	Ferramentas de análise de sensibilidade	12
2.7.1	Dakota	12
2.7.2	Spartan	13
2.7.3	PSUADE	13
2.7.4	Comparação entre ferramentas	16
3	Proposta de Solução	18
3.1	Detalhes de Implementação do MASE-Driver	20
3.2	Executando o MASE-Driver com o PSUADE	22
3.3	MASE-Driver-GUI	23
3.3.1	Módulo de Variáveis	23
3.3.2	Módulo de Configuração do PSUADE	25
3.3.3	Módulo de Configuração Geral	27
4	Experimentos e Resultados	28
4.1	Distribuição Uniforme - Amostragem LHS	28
4.1.1	Avaliação de Incerteza	30
4.1.2	Triagem de Parâmetros	30
4.1.3	Análise de Sensibilidade	30
4.2	Avaliação de Incerteza	31
4.3	Triagem de Parâmetros	33
4.4	Análise de Sensibilidade	35
5	Conclusões e Trabalhos Futuros	39

Referências	40
A PSUADE	43
B MASE-Driver	51

Lista de Figuras

2.1	Ciclo básico de um sistema de informação.	5
2.2	Visão geral do modelo de execução do MASE-BDI [1].	9
2.3	Imagens com a classificação da cobertura da terra no DF [10].	11
3.1	Visão geral da interação entre o MASE-BDI, o PSUADE e o usuário.	18
3.2	Diagrama de atividade do fluxo de trabalho com o PSUADE.	19
3.3	Diagrama de atividades da ferramenta de comunicação entre o PSUADE e o MASE.	19
3.4	Diagrama de classes do MASE-Driver.	21
3.5	Diagrama de atividades das ferramentas MASE-Driver-GUI, MASE-Driver, PSUADE e MASE-BDI.	24
3.6	Classes responsáveis pela manipulação de variáveis de entrada e saída na interface gráfica.	24
3.7	Imagem da tela de edição de variáveis da ferramenta MASE-Driver-GUI.	25
3.8	Tela de configuração com valores padrão para alguns campos.	26
3.9	Tela de configuração com o método de geração Hipercubo Latino (LH, <i>Latin Hypercube</i>), cinco amostras e marcado para adicionar perturbações. Os outros valores estão com valores padrão.	26
3.10	Tela contendo as opções de configuração gerais da ferramenta MASE-Driver-GUI.	27
4.1	Configuração da ferramenta MASE-Driver-GUI para o experimento com LHS.	30
4.2	Distribuição de Probabilidades para as variáveis de saída do modelo MASE-BDI	32
4.3	Análise de Correlação - Coeficientes de Pearson e Spearman	36
4.4	Efeito Principal dos parâmetros de saída do modelo MASE-BDI, em relação aos parâmetros de entrada.	37
4.5	Efeito Principal dos parâmetros de saída do modelo MASE-BDI, em relação aos parâmetros de entrada- Parte II	38

Lista de Tabelas

2.1	Funções densidade de probabilidade disponíveis no PSUADE.	14
2.2	Métodos de geração de amostras na ferramenta PSUADE.	15
2.3	Tabela de comparação de funcionalidades entre as ferramentas analisadas. .	16
4.1	Variáveis de entrada da simulação MASE-BDI.	28
4.2	Variáveis de saída.	29
4.3	Avaliação de Incerteza. Entrada: distribuição uniforme e amostragem: LHS.	31
4.4	Coefficientes de correlação de Pearson, saídas X saídas. Entrada: distribuição uniforme e amostragem: LHS.	34

Capítulo 1

Introdução

No âmbito corporativo, a utilização de sistemas de informação isolados, que não interagem de forma eficiente, é um fator limitante ao funcionamento de universidades, empresas multinacionais e até mesmo organizações de pequeno porte. A ausência de integrações automatizadas está associada a diversos prejuízos para a organização, tais como:

- aumento da carga de trabalho sobre as pessoas que operam os sistemas isolados;
- perda de competitividade por incapacidade de redução do tempo de processos da organização;
- dificuldade em identificar eventos que são mais facilmente percebidos com integração da informação entre sistemas;
- aumento da incidência de erros de digitação devido a repetição excessiva de trabalho operacional humano.

A integração entre sistemas de informação cumpre um papel essencial na melhoria dos processos e no aumento do desempenho de empresas e ambientes organizacionais em geral. As organizações já compreendem a necessidade e a importância de se ter um ambiente integrado de sistemas e este tópico têm recebido grande atenção de pesquisadores e profissionais de tecnologia da informação.

Um dos grandes desafios dessa área de interesse é justamente realizar novas integrações de programas. Em geral, para realizar integrações faz-se necessário:

1. implementar novas interfaces nas soluções já existentes;
2. reescrever interfaces já existentes para abarcar necessidades da tecnologia nova e
3. ajustar incompatibilidades entre novas tecnologias e as existentes.

Um exemplo de sistema que atualmente demanda melhorias no que diz respeito a integração é o MASE (*MultiAgent System for Environmental Simulation*) [18]¹ utilizado para realizar simulações ambientais.

¹<http://mase.cic.unb.br>

1.1 O Projeto MASE

Além das transformações geradas pela própria natureza, a sociedade que habita determinado ambiente possui grande influência sobre ele. Nesse sentido, a Terra pode ser considerada um quadro onde humanos e sistemas naturais interagem, sendo a ação humana o maior agente de transformação do meio ambiente. Explorar a forma como esses dois sistemas interagem tem grande valia econômica e ambiental para nossa sociedade e entender todos os fatores que influenciam essa interação tem sido o foco de estudos científicos em vários campos do conhecimento. No entanto, a complexidade dessa interação é tamanha que a verificação direta e não sistemática é insuficiente para se compreender de forma relevante as forças que mudam o ambiente [23].

Ultimamente, desafios ambientais globais e o desenvolvimento de modelos baseados em computação e ferramentas de análises têm expandido muito interesse na aplicação de abordagens computacionais no estudo de sistemas humanos. Essas novas ferramentas computacionais podem ser utilizadas em várias áreas, tais como: geografia; transporte; ou o estudo do uso e cobertura da terra – LUC (do inglês, *Land User and Cover Change*) [8, 16]. No que tange à esta área de interesse, “Cobertura da terra” refere-se aos atributos biofísicos da superfície da terra e “uso da terra” diz respeito ao propósito humano ou intenção aplicados a estes atributos [23]. Essas mudanças são muito importantes para nossa sociedade pois elas impactam em vários aspectos da Terra. LUC tem impacto direto na diversidade biológica do mundo, contribui para a mudança local e regional do clima e também causa impacto no clima global.

Para entender as mudanças de LUC que ocorreram na Terra, cientistas precisam de dados quantitativos referente a como a utilização da terra tem sido alterada pelos humanos nos últimos anos, ou centenas de anos, e como ela será alterada nas próximas décadas. Modelos baseados em agentes de LUC combinam o terreno de interesse com um sistema de agentes que representa entidades de tomadas de decisão. Logo, a utilização do modelo de agentes pode representar uma interação complexa de tomada de decisão com um sistema natural complexo, atendendo a necessidade de modelagem do problema proposto.

O MASE é um sistema computacional que utiliza um modelo configurável pelo usuário para analisar e simular os impactos humanos da mudança de uso e manejo da terra ao longo do tempo e de modo espacialmente explícito [18]. A ferramenta MASE² visa auxiliar a análise da dinâmica do uso e cobertura da terra utilizando informações técnicas para auxiliar o processo de tomada de decisão. O MASE efetivamente ilustra as dinâmicas ambientais, econômicas e sociais por meio de agentes computacionais que simulam a interação do homem no ambiente, ao mesmo tempo em que abrange políticas de gestão do uso do solo. O Projeto MASE foi desenvolvido na Universidade de Brasília e motivado pelo desafio de incorporar a inteligência de agentes computacionais em simulações ambientais. O projeto conta com o apoio de ecólogos, biólogos e analistas ambientais que auxiliam nos requisitos utilizados para a implantação computacional.

Em suma, a ferramenta MASE pressupõe a inclusão de diversas variáveis, que precisam ser submetidas a determinadas análises, para que seja possível compreender a influência de cada variável no resultado da simulação. Existem programas que fazem estas análises, mas a interação com eles normalmente se dá através de uma interface específica e os dados

²<http://mase.cic.unb.br/>

muitas vezes precisam ser tratados, também de forma específica, antes de serem utilizados pelos programas.

O objetivo deste trabalho é contribuir para o aprimoramento do uso do MASE por meio da integração entre sistemas. A implementação utilizará o modelo BDI (Belief-Desire-Intention), integrando a ferramenta MASE-BDI com uma ferramenta de quantificação de incerteza e análise de sensibilidade local e global, sendo proposta uma solução de interface entre os dois sistemas.

1.2 Questão de Pesquisa

O presente trabalho investiga a seguinte questão: Qual a possibilidade de integração do simulador ambiental MASE-BDI com uma ferramenta de quantificação de incerteza e análise de sensibilidade local e global?

1.3 Objetivos

O objetivo geral deste trabalho é integrar uma ferramenta que realiza quantificação de incertezas e análises de sensibilidade ao simulador ambiental MASE-BDI. Tal integração objetiva viabilizar: identificação de incertezas do modelo MASE-BDI, caracterização da forma com que essas incertezas se apresentam e evoluem de acordo com a simulação, quais seus impactos e se é possível reduzir tais incertezas. Pode-se citar como objetivos específicos:

- Identificar e explorar o uso de uma ferramenta de quantificação de incerteza e análise de sensibilidade local e global;
- Desenvolver interface de usuário que facilite a utilização da ferramenta de análise em conjunto com o MASE-BDI;
- Realizar experimentos para viabilizar análise de sensibilidade.

1.4 Estrutura do Documento

Este documento está dividido em cinco capítulos. No Capítulo 2 é apresentado um levantamento teórico a respeito de Sistemas de Informação, Sistemas Multiagentes, análises de probabilidade e estatística, o projeto MASE e o modelo de racionalidade Belief-Desire Intention (BDI). No Capítulo 3 são apresentadas ferramentas existentes de análise de sensibilidade, a proposta de solução e o método utilizado para geração dos resultados e análises estatísticas. No Capítulo 4 são apresentados resultados da execução da ferramenta de análise selecionada e a integração com o MASE-BDI. No Capítulo 5 encerrá-se resumindo o que este trabalho conseguiu alcançar e propõe trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo aborda os principais fundamentos de Sistemas de Informação (SI), agentes inteligentes e Sistemas Multiagentes (SMA), bem como aspectos relevantes de ferramentas de simulação ambiental. Apresenta também o Projeto MASE-BDI, aspectos da quantificação da incerteza, análise de sensibilidade e explora três ferramentas de análise de sensibilidade e quantificação de incerteza.

2.1 Sistemas de Informação

O conceito de informação é muito difícil de ser definido assertivamente. Hoje em dia, se diz que vivemos em uma era da informação ou na sociedade da informação. A informação tem muita importância para nós. Governos e empresas utilizam dela para controlar suas operações, se planejar para o futuro e para atingir suas metas. A informação é muito utilizada para tomada de decisão em várias esferas da nossa sociedade. Essa tomada de decisão vai desde coisas simples, que computadores fazem facilmente, como verificar existência de estoque em uma loja, até problemas estratégicos mais complexos, como decisões de mudança de negócio de uma empresa. Logo, a informação se tornou quase uma mercadoria, trocada por grandes empresas que a veiculam e influenciam milhares à milhões de pessoas no mundo.

Devido a importância que a informação tem na nossa sociedade, vários significados lhe foram atribuídos em contextos diferentes. Estudos linguísticos podem defini-la como o modo que as pessoas interagem entre si usando a fala ou a escrita. Em estudos filosóficos ou religiosos, a informação carrega a verdade ou crenças justificadas. Estatísticos retiram informação significativa para o problema que querem resolver a partir da análise de eventos e inúmeras observações de dados.

Para fins de SI, a informação é constituída por dados que foram formados de tal forma que tem significado útil para seus usuários [14]. Dados, em contrapartida, são fluxos de dados, sem tratamento, de eventos que acontecem em organizações ou no meio físico antes de serem organizados de tal forma que pessoas possam entendê-los e utilizá-los.

Segundo [14], SI pode ser definido como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisão, a coordenação e o controle de uma organização. Além de dar apoio à tomada de decisão, à coordenação e ao controle, esses sistemas também

auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos. A Figura 2.1 apresenta um esquema desse processo.

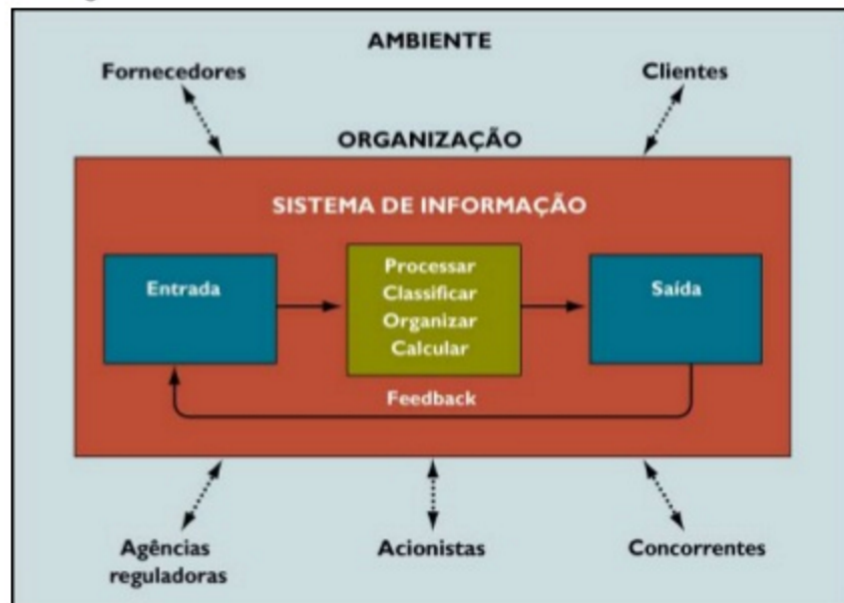


Figura 2.1: Ciclo básico de um sistema de informação.

O estudo de SI utiliza a Tecnologia da Informação (TI) dentro de organizações e sociedades. Especificamente, o estudo envolve a aplicação de TI nesses ambientes visando melhorar a forma que essas funcionam, ajudando as pessoas com suas tarefas. Isso pode ser alcançado, de forma inicial, coletando, guardando, processando e compartilhando dados e informação.

Essa descrição sugere que o estudo de SI envolve pelo menos quatro pontos:

1. as tecnologias digitais que permeiam o gerenciamento de informações baseados em computadores;
2. as pessoas que interagem de alguma forma com o SI;
3. as tarefas que as pessoas querem executar, quais são suas necessidades e requisitos;
4. a estrutura organizacional onde um SI está ou será inserido.

Esses pontos são relacionados de tal forma que se torna difícil considerar um deles de forma desassociada, sendo necessário avaliar todos os pontos na implementação de um sistema de informação. A Seção 2.2 apresenta o tema de Sistemas Multiagentes, necessário para entendimento do projeto MASE.

2.2 Sistemas Multiagentes

Sistemas Multiagentes (SMA) são um paradigma da ciência da computação que buscam resolver problemas por meio da interação de agentes computacionais. Essa área do

conhecimento emerge da pesquisa em inteligência artificial distribuída, que trata de soluções colaborativas produzidas por meio de um grupo de entidades distribuídas [24]. SMA buscam reproduzir o conhecimento e o raciocínio de vários agentes heterogêneos, os quais necessitam de coordenação para a resolução de problemas [6]. SMA é utilizado em uma crescente e ampla variedade de aplicações, que vão desde sistemas comparativamente pequenos e simples de assistência pessoal até sistemas complexos, abertos de missão crítica para aplicações industriais [11].

A unidade conceitual no estudo de sistema multiagente é a de agente inteligente. Um agente é um sistema computacional que se situa em um ambiente e é capaz de agir de maneira autônoma para cumprir os seus objetivos definidos no projeto [25]. Para isso é necessário apresentar características como autonomia, reatividade, proatividade e sociabilidade. Os SMAs são sistemas compostos por diversos agentes, onde cada um deve possuir conhecimento e habilidades para executar uma determinada tarefa, podendo cooperar ou competir, para atingir um objetivo global. A estrutura interna de um agente determina as suas capacidades e comportamento. Essa estrutura é flexível de modo a permitir atender os mais variáveis requisitos de um sistema. Quanto ao tipo de estrutura, é possível classificar os agentes inteligentes em quatro tipos básicos [20]:

- Agentes reativos simples - tipo mais simples de agente, sua ação depende unicamente da percepção atual. Podem ser descritos com regras de produção, tendo por algoritmo geral o ciclo *Interpretar entrada, Verificar regra correspondente e Agir conforme a regra*;
- Agentes reativos baseados em modelos - também conhecidos como agentes reativos com registro de estado, essa classe de agentes mantém um estado interno com base no seu histórico de percepções e no estado do ambiente;
- Agentes baseados em objetivos - além de manterem um estado interno, possuem informações de objetivos que descrevem situações desejáveis, e as utilizam para tomar suas decisões;
- Agentes baseados em utilidade - além de considerarem seus objetivos, possuem uma função de utilidade que embasa suas decisões. O agente sempre escolherá a ação que leva a uma maior utilidade.

Para [25], SMA contém um número de agentes que interagem por comunicação, são capazes de agir em um ambiente, possuem diferentes esferas de influência, e são ligados uns aos outros por relações organizacionais. Cada agente possui uma visão limitada e diferentes controles sobre partes distintas do ambiente, podendo ou não existir sobreposição de nicho entre eles. Trabalham com dados descentralizados e podem interagir por meio de cooperações, competições ou relações hierárquicas, ao compartilhar informações e recursos. Alguns modelos mentalísticos podem aumentar a complexidade do raciocínio de cada um desses agentes. Um desses modelos será apresentado na seção a seguir.

2.3 Modelo de Racionalidade BDI

O agente pode possuir percepções, comportamentos, registros de estados, objetivos e algum nível cognitivo. A base para implementar agentes com capacidades cognitivas é

descrever o comportamento dos indivíduos. Uma das teorias mais influentes no que diz respeito à tecnologia de agentes é o modelo de raciocínio prático baseado em crenças, desejos e intenções. De acordo com [9], o sucesso do modelo é baseado na sua simplicidade em reduzir o comportamento humano complexo para uma postura motivacional. Isto significa que as causas para as ações estão sempre relacionadas às crenças e aos desejos humanos, ignorando facetas como as emoções. O modelo BDI foi concebido por [7] como uma teoria de raciocínio prático [19]. O raciocínio prático consiste de uma fase de deliberação (decidir o que se quer alcançar), seguido por uma fase de raciocínio meio-fim (decidir como alcançar).

Os principais aspectos de um modelo BDI estão resumidos no trio crença-desejo-intenção de [7]. Crenças representam a informação que um agente tem sobre o mundo que habita, e sobre seu próprio estado interno. Crenças fornecem uma abstração dependente de domínio e afetam a maneira pela qual um agente percebe e pensa sobre o mundo. As crenças do agente não são apenas fruto de suas percepções. Elas também podem ser inferências a respeito de outros fatos ou outros valores. As crenças de um agente em determinado instante não são necessariamente verdadeiras para outros agentes ou verdadeiras em geral.

Os desejos representam as atitudes motivacionais de agentes, ao capturar as vontades do agente e ao dirigir o curso de suas ações. Em suma, um desejo é uma abstração que leva o agente a querer alcançar uma meta. Um agente pode ter um conjunto conflitante de desejos. O processo de deliberação de um objetivo tem a tarefa de selecionar um subconjunto consistente de desejos e representar quais estados devem ser alcançados ou mantidos. Assim, define-se a razão por que um conjunto de ações é executado. Atingir um objetivo permite que um agente apresente um comportamento proativo e não apenas reativo. Assim, por exemplo, um desejo pode ser acionado depois da ocorrência de um conjunto de eventos.

As intenções são as possibilidades de ação de um agente. Elas são guiadas pelos objetivos e consideram as crenças para poderem ser executadas. Planos são os meios pelos quais os agentes organizam a sua linha de intenções para atingirem seus objetivos. Quando um agente decide atingir um objetivo a partir de um determinado plano, ele se compromete (momentaneamente) com a realização do objetivo e, portanto, estabelece uma intenção em executar a sequência de ações do plano. Um mecanismo de raciocínio bem estabelecido é necessário para decidir qual conjunto de planos são necessários para satisfazer um determinado objetivo e prever as consequências caso um plano não seja bem sucedido. É assim que o conjunto crença-desejo-intenção permite que um agente estabeleça o seu plano de ação. Essa é a etapa de deliberação do raciocínio.

A execução do plano pode trazer o alcance do objetivo ou o seu fracasso. Isso acontece pois o ambiente dos agentes é dinâmico e os recursos planejados podem não estar mais disponíveis, por exemplo. O fracasso de um objetivo gera no agente uma nova etapa de deliberação, em que novas intenções podem ser consideradas para uma nova tentativa de realização dos desejos, ou a desistência do objetivo.

A arquitetura BDI foi aplicada pela primeira vez em um domínio ambiental e ecológico em [5]. No entanto, a maioria das aplicações para os problemas ambientais usam agentes simples e puramente reativos, e concentram seus esforços na coordenação da relação do agente com o meio ambiente. Embora vários autores reconheçam a importância de estruturas cognitivas na representação de agentes em modelos ambientais, a regra geral é a

ausência da dimensão de cognição nas aplicações ecológicas [17]. O contexto de aplicação ambiental bem como os fundamentos da ferramenta MASE serão apresentados a seguir.

2.4 O Projeto MASE

O MASE, principal software de trabalho do grupo de pesquisa, é ferramenta computacional de simulação, baseada na abordagem SMA, para analisar a dinâmica de uso e cobertura da terra, utilizando como estudo de caso o bioma Cerrado. A finalidade da abordagem de SMA é flexibilizar a interação dos agentes para integrar níveis de complexidade das decisões tipicamente humanas à modelos, em um contexto de uso e cobertura da terra. Além disso, a utilização de agentes é bastante apropriada na resolução de uma tarefa que é processada em níveis, com ações executadas paralelamente ou sequencialmente. Em linhas gerais, a ferramenta de simulação busca fornecer ao usuário a possibilidade de testar hipóteses e de refletir acerca do domínio do modelo utilizado. Em um contexto de tomada de decisão, a ferramenta pode apoiar o processo decisório a partir do momento que o tomador de decisão envolve-se na sua definição, possibilitando a sintetização de informações para gestão.

Dada uma configuração espacial inicial, a ferramenta é capaz de simular a dinâmica de uso e cobertura da terra, com base no comportamento de agentes que podem colaborar ou competir entre si. Os cenários padrão envolvem a disputa do espaço geográfico entre agricultores e pecuaristas, cada qual com seus comportamentos individuais e em grupo. A alteração de parâmetros de entrada configura cenários alternativos que avaliam a relevância das variáveis de simulação, exploram os limites da sustentabilidade dos padrões atuais frente à legislação ambiental existente, ou testam novos conjuntos de parâmetros para simular a dinâmica do uso da terra. A caracterização empírica do modelo conceitual de uso do solo está detalhado em [18] e [2].

Embora o MASE tenha sido comprovado como uma solução válida, ecólogos ainda se esforçam para adotar novas metodologias que representam a complexidade da ação humana. A maioria das arquiteturas SMA incluem agentes reativos, no entanto, isso acaba por não incorporar os processos de deliberação envolvidos no mundo real. Portanto, acrescentar uma dimensão cognitiva é primordial para explorar todas as possibilidades de interação entre o homem e os sistemas ecológicos [6]. O MASE-BDI é uma versão estendida do MASE, incorporando os aspectos do BDI no motor de raciocínio dos agentes computacionais.

Assim, o MASE-BDI apresenta uma arquitetura de agentes de alto nível que implementa o modelo BDI. A Figura 2.2 apresenta o modelo de execução. Embora o uso de uma arquitetura BDI seja prevista na literatura como um método promissor, ainda não há uma revisão crítica amplamente adotado na comunidade científica. Assim, o intuito de implementação do MASE-BDI se expande além das simulações ambientais, tendo também como objetivo investigar as vantagens, limitações e desvantagens da aplicação de agentes cognitivos em SMAs aplicados na área ambiental.

A implementação do MASE-BDI alterou de forma significativa a arquitetura do sistema multiagente e como os agentes tomam as suas decisões. Com esse aperfeiçoamento, torna-se necessário refinar as variáveis de entrada, realizar novas simulações a analisar a indecidibilidade do modelo [8]. Para facilitar o refinamento e melhoria contínua do MASE-BDI, é necessário selecionar uma ferramenta de análise, capaz de executar múltiplas aná-

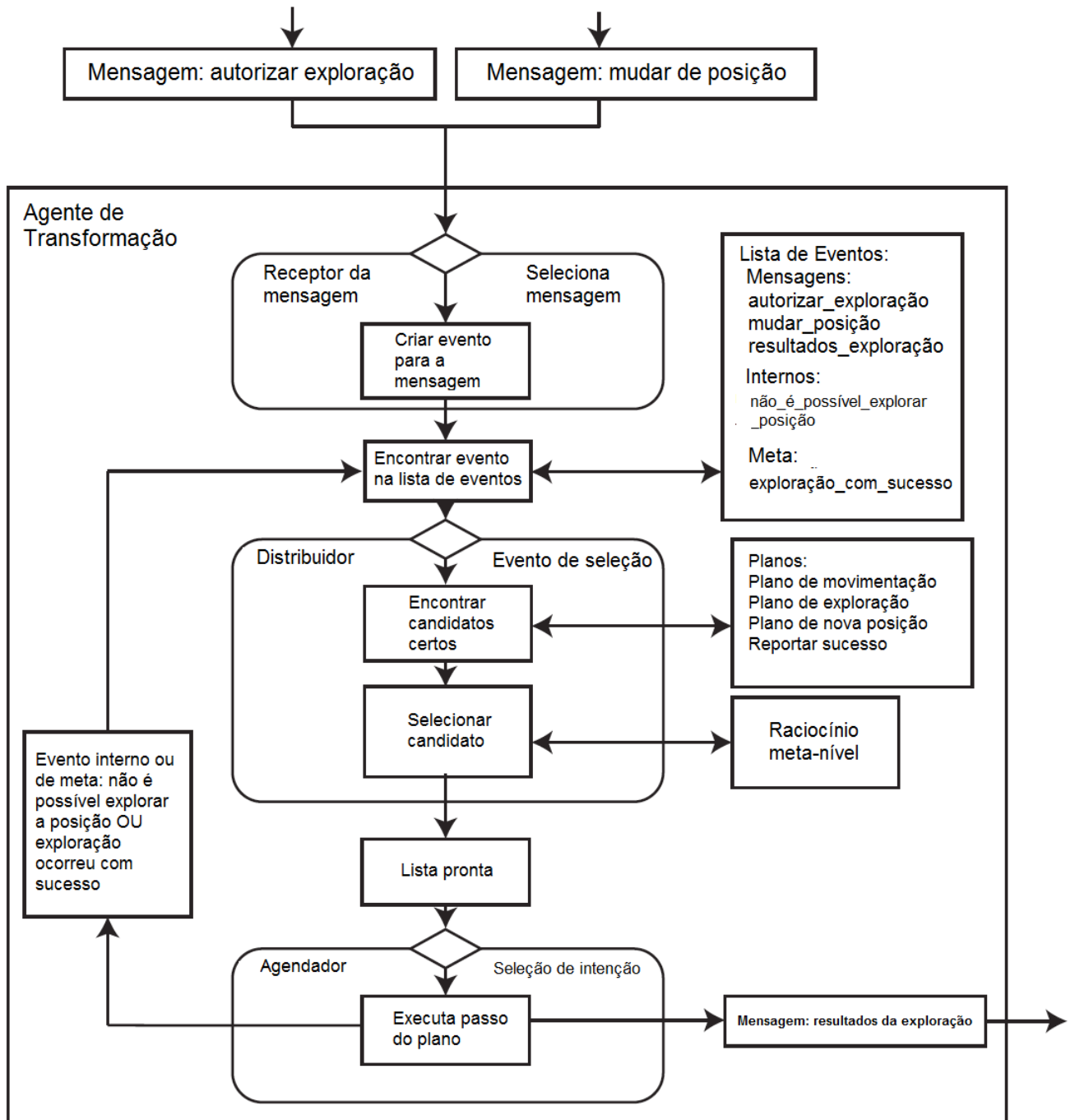


Figura 2.2: Visão geral do modelo de execução do MASE-BDI [1].

lises diferentes a partir dos resultados gerados pelas simulações do modelo MASE-BDI. Como cada ferramenta possui uma interface específica de entrada de dados e também de saída de resultados, é necessário avaliar a possibilidade de integração do MASE-BDI com alguma ferramenta e, caso seja possível tal integração, propor uma solução que automatize o processo de execução do modelo junto com uma ferramenta de análise.

2.5 Estudo de Caso: Cerrado DF

A área definida para a aplicação do modelo e teste das versões da ferramenta MASE discutidas nesse trabalho foi o Distrito Federal, Unidade Federativa (UF) que tem 100% do seu território em região de Cerrado. A escolha pelo Cerrado justifica-se por atualmente ser o bioma mais pressionado pela intensificação do uso da terra, por ser considerado um *hotspot* de biodiversidade, por ter esforço de conservação ínfimo, quando comparado a outros biomas (principalmente com a Amazônia) e pelo desconhecimento da população e dos tomadores de decisão quanto à sua importância.

Pesquisas mundiais têm demonstrado que o Cerrado é um das mais ricas savanas tropicais com alto nível de endemismo [12]. Mesmo que seja um ecossistema de elevado valor ecológico, o Cerrado é atualmente um dos biomas mais ameaçados do planeta, devido à rápida expansão da agricultura [15]. Dados do Instituto Brasileiro de Meio Ambiente e dos Recursos Naturais Renováveis (IBAMA) mostram uma perda acumulada de 47,8% da cobertura vegetal natural Cerrado nas últimas três décadas. Especialistas apontam um esforço de conservação muito abaixo das necessidades reais do bioma. Apenas 2,2 % do território ocupado pelo Cerrado é legalmente protegido [13]. A intensa alteração na cobertura da terra se dá pela substituição da vegetação nativa por monoculturas, incentivadas pelo fato de que o Cerrado é considerado uma das últimas fronteiras de produção agrícola no mundo [22], responsável por quase 70% da produção de carne bovina do Brasil e por quase a totalidade da produção de soja nacional e para exportação. A expansão destas atividades exploratórias é impulsionada por uma série de fatores socioeconômicos interligados, muitos deles incentivados pela política governamental.

Com área de $5.787,784Km^2$, o bioma do Cerrado do Distrito Federal foi escolhido para estudo de caso pois estima-se que 68,11% da vegetação nativa tenha sido desmatada até 2008 [10]. Esta informação contrasta com o fato de que aproximadamente cerca de 90% da área total do DF seja protegida por alguma categoria de unidades de conservação ou áreas protegidas (pertencentes ou não ao Sistema Nacional SNUC), de proteção integral e uso sustentável. Além desses fatores, a facilidade de obtenção de informação facilitou o processo de seleção da escolha. Entretanto, a possibilidade de investigar as consequências do Plano Diretor de Planejamento Territorial do DF, o PDOT, muito discutido e envolvido em polêmicas sobre as destinações da terra, supostamente influenciadas por interesses políticos e econômicos, consolidou a escolha da região para explorar as capacidades da ferramenta.

O cerrado do Distrito Federal (DF) foi escolhido como estudo de caso no MASE e mantido por razões de comparação no MASE-BDI. Para as simulações foram utilizadas as imagens do Satélite LANDSAT ETM+, classificadas pelo Projeto de Monitoramento do Desmatamento dos Biomas Brasileiros (PROBIO) por satélite (MMA/Ibama) e o Mapa de Cobertura Vegetal dos Biomas Brasileiros (MMA), escala 1:250.000, resolução de 30m, ano base 2002 e 2008. Os dados são de domínio público e atualmente usados pelo governo

brasileiro no controle do desmatamento e o ordenamento do território. O estado físico das células na grade de simulação corresponde ao conjunto de entrada real utilizando mapas de ocupação do solo, fornecidos pelo IBAMA, em um momento inicial (2002 - T0) e um tempo final (2008 - T6), conforme ilustrado na Figura 2.3. Pode-se observar a existência de três categorias básicas de classificação da terra: uso antrópico (área amarela), vegetação nativa (área verde) e cursos de água (área azul).

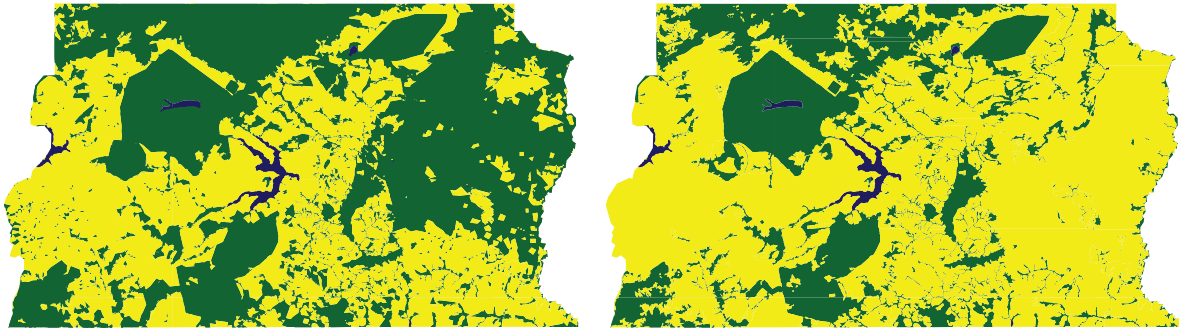


Figura 2.3: Imagens com a classificação da cobertura da terra no DF [10].

2.6 Análise de Sensibilidade

A análise de sensibilidade é utilizada para determinar quão “sensível” é uma variável de saída relativamente a variações individuais nos parâmetros de entrada. Em outras palavras, a análise de sensibilidade é utilizada para verificar como variáveis de saída podem ser afetadas por mudanças em variáveis de entrada. Uma das formas de realizar essa análise é por meio da monitoração de mudanças na variável de saída enquanto cada variável de entrada é alterada por vez, ou seja, para identificar qual variável mais influencia a variável de saída, altera-se o valor de uma das variáveis de entrada, mantendo-se os valores de base das outras. Depois, retorna-se para o valor de base a variável alterada, repetindo-se o procedimento para cada uma das demais variáveis de entrada. A partir dessas alterações de valores nas variáveis de entrada, monitora-se a variável de saída.

Quando traduzimos uma situação real em um modelo, algumas incertezas estão envolvidas e devem ser consideradas. Exemplos de incertezas incluem aquelas decorrentes de ruídos e erros de medição, de ausência de dados, de variabilidade dos parâmetros e dos fatores ambientais, bem como as incertezas nas condições iniciais e de fronteira, nos submodelos derivados de dados físicos (principalmente por causa das necessárias simplificações) e aquelas que surgem da implementação do modelo, de discretização de erros, de erros de algoritmo etc.

Em estudos dessa natureza, é importante que as incertezas sejam analisadas, de forma a permitir declarações mais precisas e um aumento no grau de confiança de predições baseadas em uma simulação, ou seja, a estimação da incerteza torna-se essencial para validar e verificar modelos computacionais. Uma vez identificadas e caracterizadas as incertezas, é possível estudar como elas se propagam, quais são os seus impactos e, finalmente, como minimizá-las. Uma análise de sensibilidade pode contribuir para uma análise de incerteza

por meio da identificação de quais parâmetros são relevantes para prever a imprecisão [21].

Após essa introdução sobre análise de sensibilidade, precisamos identificar algumas ferramentas existentes atualmente e analisá-las para escolher uma ferramenta que satisfaça os requisitos deste trabalho.

2.7 Ferramentas de análise de sensibilidade

Para que seja possível integração MASE-BDI, faz-se necessário escolher uma ferramenta adequada para geração de amostras, execução do modelo e análise dos resultados. Foram avaliadas três ferramentas e os principais critérios de avaliação foram:

1. capacidade multiplataforma;
2. facilidade de integração com outras ferramentas;
3. disponibilidade de métodos de geração de amostras;
4. disponibilidade de métodos de análise;
5. rapidez na execução da geração de amostras e análises.

Três ferramentas foram identificadas e analisadas para atender às necessidades apresentadas por este trabalho: PSUADE, Dakota e Spartan. As próximas seções são dedicadas à análise de cada uma delas.

2.7.1 Dakota

Dakota¹ (*The Dakota Design Analysis Kit for Optimization and Terascale Applications*) é uma ferramenta que agrupa algoritmos para a exploração experimental e aumento da credibilidade das simulações. O framework Dakota foi desenvolvido e mantido pelo Laboratório Nacional de Sandia e fornece soluções para:

- Otimização com métodos baseados em gradiente e não gradiente;
- Quantificação da incerteza com diferentes métodos de amostragem, confiabilidade, expansão estocástica e métodos epistêmicos;
- Estimativa de parâmetros não lineares com métodos de mínimos quadrados;
- Análise de Sensibilidade ou Variância com o projeto de experimentos e o estudo de parâmetros.

Cada um desses recursos pode ser usados sozinhos ou como componente dentro de estratégias avançadas, tais como a otimização híbrida, otimização com modelos substitutos à base, programação não-linear, ou otimização sob incerteza [3].

O kit de ferramentas Dakota utiliza projetos orientados a objeto para implementar abstrações dos componentes principais necessários para a análise de sistemas iterativos. Fornece um ambiente flexível e extensível de resolução de problemas para análise de

¹<https://dakota.sandia.gov/>

projeto e desempenho de modelos computacionais em computadores de alto desempenho. A quantidade de funcionalidades disponíveis e o número de possibilidades de métodos e algoritmos exige do usuário uma noção muito mais apurada das técnicas estatísticas e um conhecimento mais profundo sobre os algoritmos.

2.7.2 Spartan

A ferramenta Spartan (*Simulation Parameter Analysis R Toolkit Application*) [4] é um pacote que disponibiliza técnicas de análise estatística em R para tratar aspectos de incerteza, inerentes aos sistemas estocásticos, e permitir que pesquisadores possam entender melhor as relações emergentes em uma simulação computacional. Apesar de ter sido originalmente desenvolvido para análise em sistemas de simulação biológicas, pode ser aplicada na análise de resultados empíricos em outros contextos. A ferramenta Spartan possui 3 principais técnicas, que utilizam métodos de amostragem distintos: Análise da Incerteza Aleatória (AIA), Análise da Robustez (AR) e Análise da Sensibilidade Global (ASG).

A técnica AIA é utilizada a fim de definir o número de execuções necessárias para diminuir a incerteza associada à aleatoriedade do algoritmo estocástico empregado no modelo em análise, a atingir-se algum nível pré-definido de acurácia. Essa técnica é relevante principalmente por permitir equilibrar o *trade-off* entre requisitos de qualidade e recursos computacionais.

A técnica de Análise da Robustez é necessária pois em modelos ecológicos/biológicos há grande demanda pela calibração dos parâmetros, o que impacta o desempenho dos algoritmos. A técnica determina a robustez do algoritmo implementado, ou seja, analisa sob quais parâmetros o algoritmo produz resultados mais previsíveis, com maior grau de confiança.

A análise de sensibilidade Global, terceira técnica disponível da ferramenta Spartan, apresenta como a relação entre diferentes parâmetros indicam quais deles têm maior influência nos resultados do algoritmo. A técnica ASG perturba todos os parâmetros simultaneamente, simulando a dinâmica entre os parâmetros. Essa técnica utiliza a amostragem por hipercubo latino (*Latin Hypercube Sampling* - LHS) para criar conjuntos de valores de parâmetros dentro dos limites informados de modo que minimiza a correlação entre os parâmetros.

O pacote Spartan está disponível² para todos os usuários com um mínimo conhecimento em R, embora ainda não existam aplicações diferentes da área biológica.

2.7.3 PSUADE

O programa PSUADE (*Problem Solving environment for Uncertainty Analysis and Design Exploration*) é uma ferramenta com uma licença de software livre (*Lesser General Public License*³, LGPL). Tem um grande número de algoritmos que realizam análise de incerteza e análise de sensibilidade local e global⁴. Possui três principais componentes:

1. métodos de amostragem;

²<https://cran.r-project.org/web/packages/spartan/index.html>

³<https://www.gnu.org/copyleft/lesser.html>

⁴http://computation.llnl.gov/casc/uncertainty_quantification/

2. ambiente de execução e simulação;
3. métodos de análise.

O ambiente de execução criado pelo PSUADE, permite a execução automática de um programa em sequência ou paralelamente. A partir da entrada de um arquivo de configuração é possível controlar:

- A quantidade de variáveis de entrada e quais suas funções densidade de probabilidade (algumas das funções disponíveis na Tabela 2.1).
- Selecionar o método de geração de amostras (métodos disponíveis na Tabela 2.2).
- Configurar o programa que o PSUADE chamará em suas execuções (chamado de *drive*).
- Selecionar o método de análise (entre os mais de 20 métodos disponíveis).

O arquivo de entrada tem 5 seções: *INPUT*, *OUTPUT*, *METHOD*, *ANALYSIS* e *APPLICATION*. Na seção *INPUT* são definidas as variáveis de entrada e suas funções densidade de probabilidade. Nessa seção é necessário informar a quantidade de variáveis que a entrada vai ter e seus intervalos. A seção *OUTPUT* define a quantidade e ordem de saídas no arquivo informado pelo usuário na seção *APPLICATION*. A seção *METHOD* define várias características a respeito da geração de amostras, o método que será utilizado, bem como se a amostra vai sofrer perturbações aleatórias e a semente que gera uma amostragem (para que seja possível reproduzir os experimentos). A seção *APPLICATION* define qual programa o PSUADE irá chamar para ler seu arquivo de entrada e produzir um arquivo de saída, além de informar se as tarefas vão ser executadas paralelamente ou não. Na seção *ANALYSIS* o usuário escolhe se deseja já fazer algum tipo de análise quando a ferramenta terminar sua execução. Um exemplo do arquivo de configuração do *PSUADE*, por padrão chamado de *psuade.in*, pode ser encontrado no Apêndice A.

Tabela 2.1: Funções densidade de probabilidade disponíveis no PSUADE.

Funções densidade de probabilidade
Uniforme (padrão)
Normal
Log-normal
Triangular
Beta
Exponencial
Gamma
Weibull
Amostras definidas pelo Usuário

O *driver* é o programa responsável por processar o arquivo contendo as variáveis geradas por cada iteração do PSUADE e produzir uma saída como a definida na seção

OUTPUT do arquivo de entrada. Após o processamento da entrada e produção do arquivo de saída, o PSUADE detecta que houve mudança no diretório de execução da ferramenta e encerra a execução atual, coleta os resultados escritos no arquivo de saída e envia mais uma vez um arquivo de entrada contendo novas amostras.

Tabela 2.2: Métodos de geração de amostras na ferramenta PSUADE.

Métodos de geração de amostra
MC - Monte Carlo
LPTAU - <i>A quasi-random Sequence</i>
FACT - <i>Full Factorial Design</i>
MOAT - <i>Morris one-at-a-time Screening</i>
LH - <i>Latin Hypercube</i>
OA - <i>Orthogonal Array</i>
OALH - <i>Orthogonal Array-based Latin Hypercube</i>
FAST - <i>Fourier Amplitude Sampling Test (FAST)</i>
BBD - <i>Box Behnken design</i>
PBD - <i>Plackett Burman design</i>
FF4 - <i>Fractional factorial of resolution IV</i>
FF5 - <i>Fractional factorial of resolution V</i>
CCI4 - <i>Central composite (circumscribed) of resolution IV</i>
CCI5 - <i>Central composite (circumscribed) of resolution V</i>
METIS - <i>full space-filling based on domain decomposition</i>
SPARSEGRID - <i>a sparse grid method</i>

Logo que o PSUADE chega a quantidade de repetições definidas pelo usuário, a ferramenta termina a execução e consolida todos os resultados em um documento. Este documento, chamado por padrão de *psuadeData* (Apêndice A), pode ser carregado na memória através de uma ferramenta de comando de linha que o PSUADE possui e várias análises podem ser feitas utilizando esse documento. Todas análises tem duas saídas, a primeira, uma saída de texto no terminal de execução e a segunda um arquivo MATLAB⁵, contendo os dados da análise plotados em um gráfico.

Limitações da ferramenta

Apesar da ferramenta PSUADE atender às necessidades levantadas para integração com a ferramenta MASE-BDI, ela ainda possui alguns problemas na sua utilização.

Em ambientes Unix, onde foi testada, a ferramenta requer um pouco de conhecimento prévio de como o sistema operacional funciona. Apesar do tutorial abranger todos os passos necessários para instalação da ferramenta, ainda é necessário que o usuário consiga abrir uma janela de comando de linha, seja capaz de compilar a ferramenta, indicando diretórios de compiladores, instalação do Java e outros requisitos. Logo, para que seja possível executar a ferramenta, o usuário antes vai ter que passar por uma etapa de configuração do ambiente, que pode não ser trivial para o usuário comum.

⁵<http://www.mathworks.com/products/matlab/>

No que diz respeito a sua utilização, desde a criação do arquivo de entrada até a geração do arquivo de saída é muito simples, como já explicado, basta executar o PSUADE com o arquivo de entrada. Agora, a utilização do PSUADE para realização de análises é um pouco mais complexa. Primeiro, a única forma de interação com o PSUADE é através da execução do seu próprio ambiente por comando de linha. Outra vez, a falta de uma interface gráfica do usuário para este tipo de análise é algo que hoje em dia complica a utilização dessa ferramenta.

A utilização do PSUADE por comando de linha também é um pouco difícil, mesmo que o manual informe todos os comandos existentes desse ambiente. Para carregar um arquivo de saída do PSUADE (“psuadeData”, já mencionado), é necessário executar um comando *load* passando o caminho absoluto para o arquivo de saída. Depois que o PSUADE carrega esse arquivo na memória você pode passar a fazer análises dos dados. Cada análise normalmente resume-se a entrada de comandos de duas ou três letras, normalmente acrônimos dos nomes das análises, e a escolha de qual variável de saída que se deseja fazer a análise escolhida. Algumas dessas análises, como a análise de incerteza (*Uncertainty Analysis*, UA), gera como saída além de dados como média, desvio padrão, etc., um arquivo do programa MATLAB. O problema é que o PSUADE não emprega um nome específico para cada arquivo do MATLAB, somente identifica qual foi a análise escolhida (e.g. *matlabua.m*, UA). Logo, caso se queira realizar análises em várias variáveis é necessário ir até o sistema de arquivos e renomear cada arquivo de saída para que não se perca nenhuma informação.

Apesar dessas dificuldades, principalmente para usuários que não tem facilidade com sistemas computacionais, a ferramenta mostrou-se muito eficaz e capaz de executar tarefas complexas relacionadas a geração de amostras e análises de dados.

2.7.4 Comparação entre ferramentas

A Tabela 2.3 apresenta as ferramenta consideradas e os aspectos avaliados. Seguindo os critérios de avaliação constantes desta tabela, todas as ferramentas são ótimas candidatas para serem integradas ao MASE-BDI. No entanto, a ferramenta PSUADE apresenta uma grande facilidade em sua utilização e também no que diz respeito a integração com o MASE-BDI.

Tabela 2.3: Tabela de comparação de funcionalidades entre as ferramentas analisadas.

	PSUADE	Spartan	Dakota
Multiplataforma	Sim	Sim	Sim
Complexidade de Integração	Baixa	Alta	Alta
Disponibilidade de Métodos de Amostragem	Alta	Alta	Alta
Disponibilidade de Métodos de Análise	Alta	Alta	Alta
Performance	Alta	Alta	Alta

O PSUADE conta com 17 métodos de geração de amostras, além de possuir os métodos mais utilizados como Monte Carlo (MC) e *Fourier Amplitude Sampling Test* (FAST). Além disso, possui mais de 20 métodos de análises, desde métodos básicos de estatística,

contendo média, desvio padrão, variância, assimetria, curtose, etc., até métodos mais complexos como reduções e análise superficial de resposta. Utilizando os exemplos contidos na ferramenta PSUADE foi possível verificar também que a ferramenta gera de forma rápida as amostras para execução das instâncias dos programas, sendo o maior problema a necessidade de realização de leituras e escritas na memória física do computador.

Após essa análise, pode-se perceber que o software PSUADE tem muito a contribuir com este trabalho, sobretudo por ter facilidade de integração, além de atender a todas as especificações deste trabalho.

Através da análise da ferramenta PSUADE, foi identificada a necessidade da implementação de uma nova funcionalidade dentro do MASE-BDI, ou um módulo de interface que fizesse a comunicação entre o MASE-BDI e o PSUADE. Pois é necessário que o MASE-BDI processe o arquivo com as variáveis de entrada geradas pelo PSUADE e que produza o arquivo de saída no formato que o PSUADE necessita. A proposta desse módulo será apresentado no Capítulo 3.

Capítulo 3

Proposta de Solução

Para integração entre o PSUADE e o MASE-BDI faz-se necessário dividir a integração em duas etapas. A primeira trata da integração entre o MASE-BDI e o PSUADE, que precisa seguir um protocolo específico para que ocorra com sucesso. Na segunda etapa, é necessário implementar uma interface gráfica de usuário que retire a necessidade de interação com arquivos de configuração, tanto do MASE-BDI quanto do PSUADE, visando diminuir ao máximo erros devido a interação do usuário com arquivos sensíveis a formatação dos dados e estrutura do documento. A Figura 3.1 apresenta uma visão geral da interação entre os sistemas e o usuário.

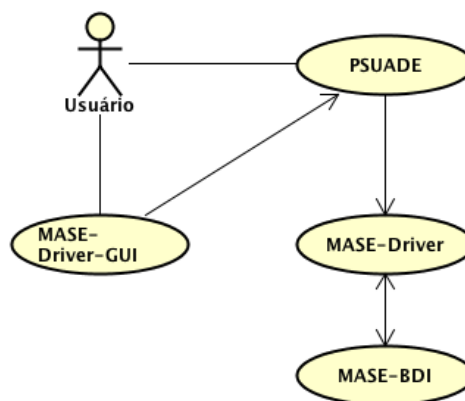


Figura 3.1: Visão geral da interação entre o MASE-BDI, o PSUADE e o usuário.

Para a integração entre o MASE-BDI e o PSUADE, é necessário definir quais são as entradas que o modelo MASE-BDI aceita e qual configuração é necessária para utilizar o PSUADE. A Figura 3.2 apresenta o fluxo de atividades para trabalhar com a ferramenta PSUADE. A seguir, apresentamos um resumo das atividades que o MASE-Driver, ferramenta integrante da proposta, tem que realizar para integrar o MASE-BDI ao PSUADE:

1. Receber o nome do arquivo de entrada criado pelo PSUADE
2. Abrir o arquivo e interpretar as entradas
3. Compilar as entradas da forma que o MASE-BDI precisa

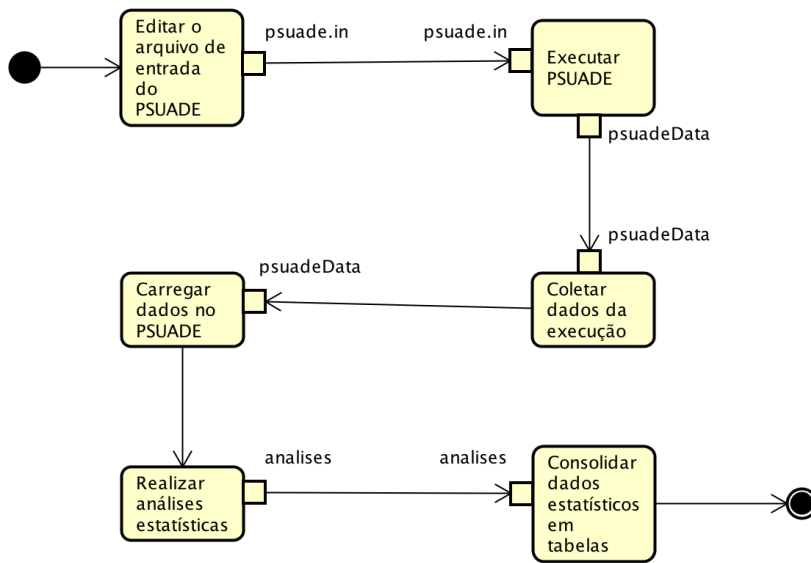


Figura 3.2: Diagrama de atividade do fluxo de trabalho com o PSUADE.

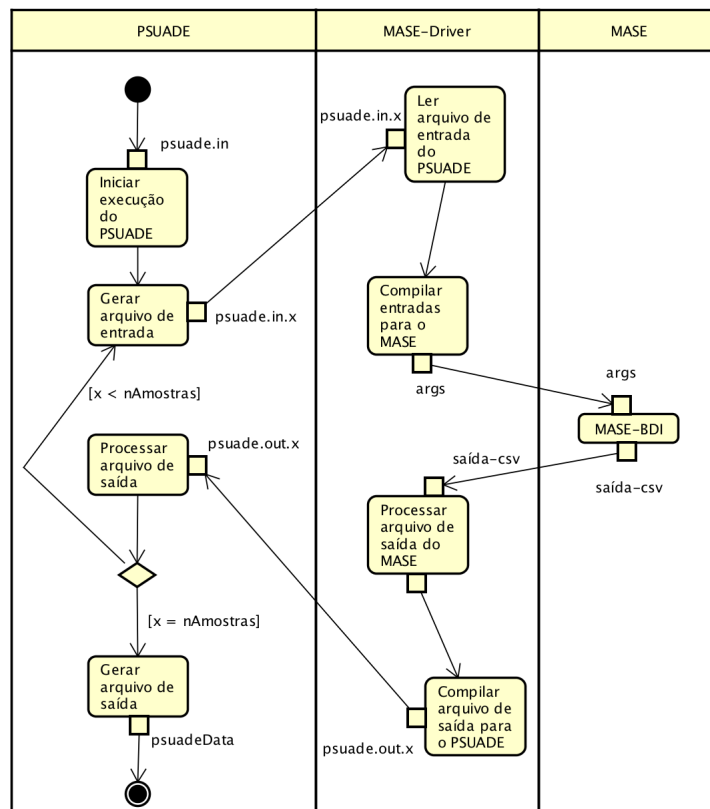


Figura 3.3: Diagrama de atividades da ferramenta de comunicação entre o PSUADE e o MASE.

4. Iniciar a execução do MASE-BDI com as entradas do PSUADE
5. Abrir o arquivo de saída do MASE
6. Criar e popular o arquivo de saída para o PSUADE com as saídas do MASE.

Para viabilizar a integração, a solução precisa receber do PSUADE o documento de entrada *psuade.in.x* (exemplo no Apêndice A), processar este arquivo e criar um arquivo de saída chamado *psuade.out.x* (Apêndice B). A letra *x* nos nomes dos arquivos representa qual o número da entrada que o PSUADE gerou. A próxima seção (Seção 3.1) vai tratar dos aspectos de implementação do MASE-Driver. A Figura 3.3 apresenta o fluxograma da interação entre o PSUADE, o MASE-Driver e o MASE-BDI.

Em um segundo momento, trata-se da interação entre o usuário, o MASE-BDI e o PSUADE, sendo necessário criar uma interface gráfica que seja capaz de:

1. Possibilitar que o usuário crie uma configuração de entrada para o PSUADE, que inclui, edição de variáveis de entrada e saídas, edição da seção de método do arquivo *psuade.in*;
2. Permite gravar um arquivo de configuração contendo os dados informados pelo usuário;
3. Iniciar a execução PSUADE a partir da interface gráfica com o arquivo de configuração gerado.

3.1 Detalhes de Implementação do MASE-Driver

A ferramenta proposta, denominada MASE-Driver, foi implementada em Python¹, uma linguagem interpretada, interativa e orientada a objetos. Foi escolhida para implementação da solução por permitir grande rapidez no trabalho, pois possui bibliotecas que permitem facilmente ler, escrever e manipular arquivos, o que tem grande importância neste caso. Contendo ampla documentação na internet, Python é muito recomendada quando se quer integrar sistemas mais eficientemente. Além disso, Python é multiplataforma, o que a torna perfeita para integrar dois sistemas que também são multiplataforma.

Para este trabalho o MASE-Driver funciona somente com o PSUADE mas é facilmente extensível para ser o principal meio de interação com a ferramenta MASE-BDI. Novos módulos podem ser implementados, alcançando um impacto ainda maior no desenvolvimento do MASE-BDI, mesmo módulos que dão suporte a protocolos de rede podem ser implementados e acoplados ao MASE-Driver para viabilizar esse tipo de comunicação com o MASE-BDI.

A implementação de um módulo novo se dá de forma simples, bastando herdar a interface “MaseDriver” e implementar os métodos abstratos *Read* e *Write*. Neste momento, somente uma classe implementa essa interface, dando suporte ao PSUADE, chamada *MasePsuadeDriver*. A Figura 3.4 apresenta o diagrama de classes da ferramenta MASE-Driver. Foi necessário limitar o MASE-Driver para execução do PSUADE. Essa limitação diz respeito a necessidade de ser obrigatória a criação de um *script* Python para cada

¹<https://www.python.org/>

novo módulo. Logo, se a ferramenta for estendida para suportar outros programas, será necessário criar um novo *script* que herda da classe *MaseDriver* e implementa os métodos necessários.

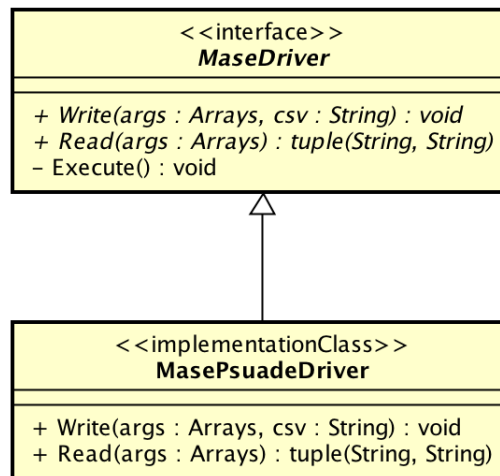


Figura 3.4: Diagrama de classes do MASE-Driver.

A implementação da interface *MaseDriver* requer definição de dois métodos: *Read* e *Write*. O método *Read* recebe como parâmetro os argumentos que são passados quando da execução do MASE-Driver e retorna uma lista de tuplas contendo, nessa ordem, o nome de uma variável do modelo MASE-BDI e o seu valor. No caso do PSUADE, o método *Read* recebe duas cadeias de caracteres: o nome do arquivo de entrada *psuade.in.x*; e o nome de arquivo de saída *psuade.out.x* onde *x* é o número atual da amostra gerada pelo PSUADE. O método *Write* recebe como argumentos a saída do MASE-BDI, um arquivo CSV (*Comma Separated Values*). Este método deve tratar a escrita desses valores da forma que for conveniente para o software que está aguardando estes dados. No caso do PSUADE, um arquivo chamado *psuade.out.x* é criado com os dados das 12 variáveis presentes no arquivo CSV gerado pelo MASE. Um exemplo de um arquivo de saída criado pelo MASE-Driver para o PSUADE pode ser observado no Apêndice B.

Para executar o MASE-Driver não é preciso ter o PSUADE instalado, mas é preciso seguir as regras de entrada e saída do PSUADE, pois essas são as únicas implementadas neste momento, além de editar o arquivo de configuração “mase-driver-config.json”, um arquivo no formato JSON ² que deve estar na mesma pasta do MASE-Driver. Antes de executar o MASE-Driver é preciso:

1. Ter a ambiente Python instalado na máquina e configurado na variável PATH do sistema;
2. Ter a ambiente Java instalado na máquina e configurado na variável PATH do sistema;
3. Efetuar a configuração do MASE-Driver utilizando a interface gráfica implementada neste trabalho (introduzida na Seção 3.3).

²<http://json.org/>

É possível executar o MASE-Driver sem auxílio da interface gráfica, no entanto, é um processo muito mais custoso. Para executar o MASE-Driver basta editar o arquivo de configuração (exemplo do arquivo no Apêndice B) e executar, por exemplo, o comando: `./MASE-Driver.py psuade.in.1 psuade.out.1`. Neste caso, o arquivo `psuade.in.1` existe no diretório atual e está no formato indicado pelo PSUADE (Apêndice A). Os elementos necessários do arquivo de configuração são:

- Um objeto JSON chamado “lastConfiguration” contendo as chaves: “maseLocation”, “input_variables” e “output_variables”
- A chave “maseLocation” aponta para o caminho absoluto do executável da ferramenta MASE-BDI
- A chave “input_variables” contém uma lista ordenada (de acordo com o arquivo de entrada do PSUADE) das variáveis de entrada do MASE-BDI
- A chave “output_variables” contém uma lista ordenada (de acordo com o arquivo de entrada do PSUADE) das variáveis de saída do MASE-BDI
- Cada elemento da lista de variáveis é um objeto contendo, no mínimo, o nome da variável no atributo “name”.

Após execução do MASE-Driver no exemplo acima, será criado um arquivo chamado “psuade.out.1” com as saídas do MASE-BDI no formato esperado pelo PSUADE. Como muitas vezes se tratam de várias execuções do MASE-BDI quando se utiliza o PSUADE, optou-se por sempre remover todas as saídas do MASE-BDI após utilizá-las.

3.2 Executando o MASE-Driver com o PSUADE

O PSUADE precisa de um arquivo de configuração como entrada para realizar sua execução. Para executar o PSUADE utilizando o MASE-Driver, é necessário editar o arquivo de entrada, explicado com maiores detalhes na Seção 2.7.3, especificamente as seções APPLICATION que especifica a aplicação que o PSUADE vai executar e METHOD que especifica parâmetros de execução do PSUADE. Basta informar na variável “driver” da seção APPLICATION o caminho para o MASE-Driver e o PSUADE, já pode ser executado (após feita configuração do MASE-Driver, explicado na seção anterior).

Como exemplo, em plataformas UNIX, a indicação do “driver” no arquivo de entrada para o PSUADE é feita da seguinte forma: “driver = ./MASE-Driver.py”, no caso da execução do PSUADE acontecer na mesma pasta que contém o arquivo de entrada *psuade.in*. Após este passo, basta configurar o arquivo “mase-driver-config.json” nas seções necessárias, apresentadas na seção anterior, e o PSUADE passando o arquivo de entrada, configurado de acordo com a necessidade da geração de amostras, que o PSUADE vai começar a gerar as entradas, processar as saídas do MASE-Driver e depois executar a quantidade de vezes definida no arquivo de entrada, gerando um arquivo de saída com o nome padrão de “psuadeData” (vide Apêndice A).

Para que o MASE-Driver seja mais fácil de ser utilizado, foi incluída a implementação de uma interface gráfica que facilita a operação do MASE-Driver. A próxima seção discute detalhes de implementação da interface gráfica.

3.3 MASE-Driver-GUI

A ferramenta que permite execução do PSUADE junto com o MASE-Driver, chamada de MASE-Driver-GUI, foi implementada em Python, assim como o MASE-Driver. Para criação da interface gráfica foi utilizado um pacote que faz parte da própria instalação do Python, chamado Tkinter ³. Tkinter é a implementação mais comum de pacotes gráficos para a linguagem, provendo várias ferramentas que permitem criação facilitada de elementos gráficos como: botões, caixas de diálogo, campos de texto e etc. Antes da implementação da ferramenta, foram levantados os seguintes requisitos:

- Ser capaz de adicionar variáveis de entrada e de saída, escolher seus limites inferiores e superiores;
- Esconder detalhes do arquivo de entrada do PSUADE do usuário, permitindo que o usuário edite o mínimo necessário para execução do PSUADE;
- Ser capaz de salvar configurações para que o usuário possa carregá-las no futuro, facilitando a reprodução de experimentos em qualquer ambiente;
- Ter facilidade de edição dos dados presentes na ferramenta, visando facilitar, por exemplo, a inclusão ou remoção de métodos de geração de amostra no futuro;
- Realizar validações mínimas nos dados para que estes não gerem erros na execução do PSUADE;
- Permitir que o usuário execute o PSUADE (com o MASE-Driver) a partir da própria interface, sem necessidade de interação com comando de linha.

Definido os requisitos mínimos necessários para implementação da interface gráfica, passamos para implementação. A Figura 3.5 apresenta o diagrama de atividades da ferramenta.

A ferramenta MASE-Driver-GUI foi desenvolvida utilizando o paradigma de orientação a objetos para organização de responsabilidades. Ela pode ser dividida em três módulos distintos, o primeiro lida com as variáveis necessárias para execução correta do *driver* definido no arquivo de entrada do PSUADE, no caso deste trabalho, o MASE-Driver. Esse módulo cuida especificamente da manipulação de variáveis de entrada e saída, ou seja, este é o módulo responsável por alcançar o primeiro requisito levantado para a ferramenta. O segundo módulo cuida da configuração do PSUADE em si. O terceiro módulo cuida de configurações gerais da ferramenta. A Figura 3.6 apresenta as classes responsáveis pelo módulo de controle de variáveis.

3.3.1 Módulo de Variáveis

A classe *VariableEntryManager* foi criada para servir como gerenciadora da classe *VariableEntry*, responsável por adicionar o nome da variável, campos de texto para o limite inferior e superior da variável, opção de variável de entrada ou saída e remoção da variável da lista de variáveis. As operações de inserção e remoção de *VariableEntry* é responsabilidade da classe *VariableEntryManager*, também responsável por passar todas

³<https://docs.python.org/2/library/tkinter.html>

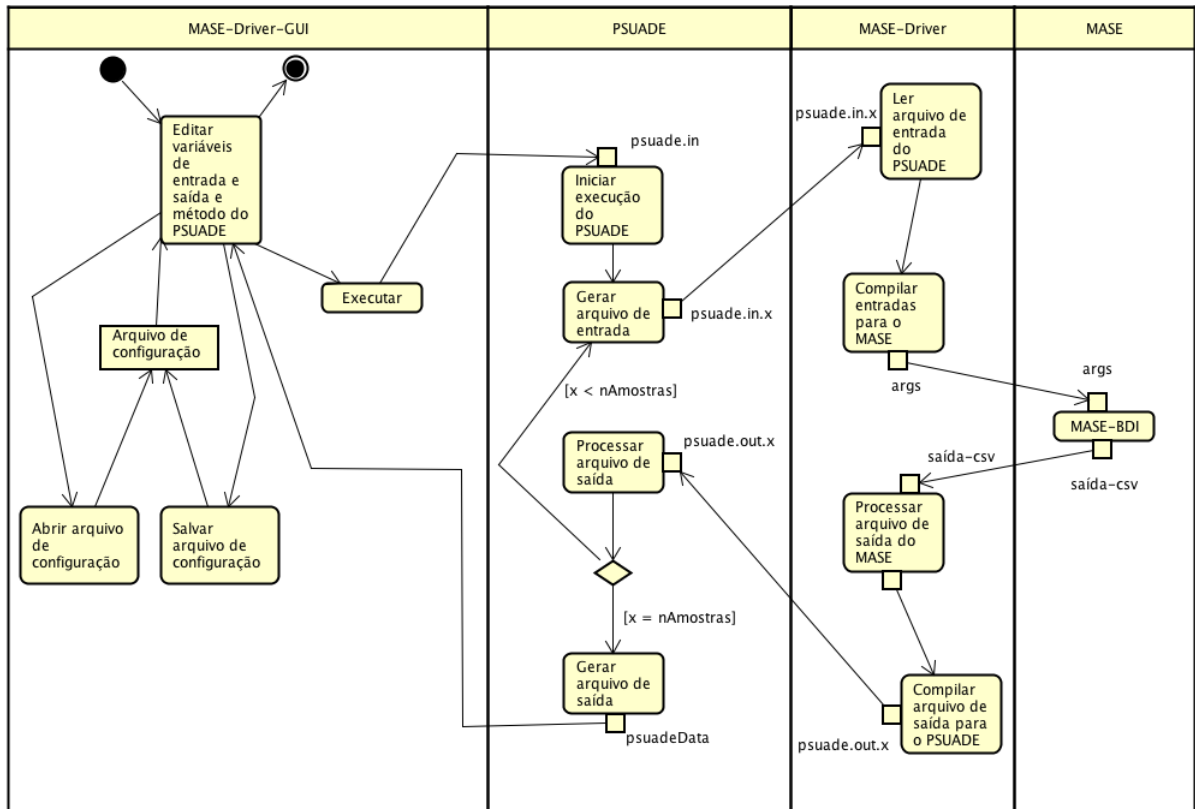


Figura 3.5: Diagrama de atividades das ferramentas MASE-Driver-GUI, MASE-Driver, PSUADE e MASE-BDI.

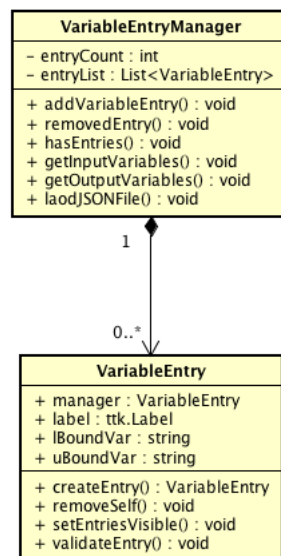


Figura 3.6: Classes responsáveis pela manipulação de variáveis de entrada e saída na interface gráfica.

as informações providas pelo usuário para o módulo que configura o arquivo de entrada do PSUADE, *psuade.in*. A validação dos campos de limite inferior e superior é feita pela própria classe *VariableEntry*, validando somente se o usuário está inserindo números nesses campos, pois somente números são aceitos pelo PSUADE como limite inferior e superior. Portanto, quando pronto para salvar ou executar o PSUADE, basta invocar os métodos *getInputVariables* e *getOutputVariables* da classe *VariableEntryManager* que serão compilados as variáveis de entrada e saída. A Figura 3.7 apresenta as variáveis de entrada e saída utilizadas para rodar o PSUADE com o MASE-Driver.

Nome	Lim. Inferior	Lim. Superior	Entrada	Saída	
transformationAgentQty	<input type="text" value="1"/>	<input type="text" value="100"/>	<input checked="" type="radio"/>	<input type="radio"/>	-
transformationAgentGroupPercentage	<input type="text" value="10"/>	<input type="text" value="100"/>	<input checked="" type="radio"/>	<input type="radio"/>	-
individualExploration	<input type="text" value="1"/>	<input type="text" value="500"/>	<input checked="" type="radio"/>	<input type="radio"/>	-
groupExploration	<input type="text" value="1"/>	<input type="text" value="1500"/>	<input checked="" type="radio"/>	<input type="radio"/>	-
time			<input type="radio"/>	<input checked="" type="radio"/>	-
qtyAgents			<input type="radio"/>	<input checked="" type="radio"/>	-
percentageAgents			<input type="radio"/>	<input checked="" type="radio"/>	-
figureOfMerit			<input type="radio"/>	<input checked="" type="radio"/>	-
producersaccuracy			<input type="radio"/>	<input checked="" type="radio"/>	-
usersaccuracy			<input type="radio"/>	<input checked="" type="radio"/>	-
wrongchange			<input type="radio"/>	<input checked="" type="radio"/>	-
rightchange			<input type="radio"/>	<input checked="" type="radio"/>	-
wrongpersistence			<input type="radio"/>	<input checked="" type="radio"/>	-
nullModel			<input type="radio"/>	<input checked="" type="radio"/>	-
simulatedNullModel			<input type="radio"/>	<input checked="" type="radio"/>	-
steps			<input type="radio"/>	<input checked="" type="radio"/>	-

Figura 3.7: Imagem da tela de edição de variáveis da ferramenta MASE-Driver-GUI.

3.3.2 Módulo de Configuração do PSUADE

Para implementação do segundo módulo que permite a edição dos parâmetros de configuração do PSUADE (seções METHOD, APPLICATION e ANALYSIS do arquivo *psuade.in*) foi necessário utilizar o arquivo de configuração *mase – driver – config – file.json*, criado na seção anterior para configurar o MASE-Driver, pois este módulo precisava ter certeza que não haveriam dados incorretos sendo enviados para o PSUADE. No Apêndice B é possível verificar o objeto *sampling_methods*, que nada mais é uma lista de valores que o PSUADE aceita como método de amostragem. Esse arquivo de configuração foi atualizado de acordo com a documentação do PSUADE na data desse trabalho de conclusão de curso. Na janela de configuração do PSUADE é possível escolher:

- método de geração;
- quantidade de amostras;
- número de replicações;
- número de refinamentos;

- tamanho do refinamento;
- número de referência de refinamento;
- tipo de refinamento;
- Semente aleatória;
- adicionar perturbações nas amostras;
- adicionar mais perturbações nas amostras;

Cada campo disponível na tela de configuração do PSUADE passa por validação mas os únicos campos necessários são: método de geração e quantidade de amostras. Por padrão, o arquivo de entrada do PSUADE tem os valores variáveis: número de replicações, número de refinamentos, tamanho de refinamento, número de referência de refinamento; já preenchidas por padrão, logo, somente é obrigatório o preenchimento do método de geração e da quantidade de amostras. As Figuras 3.8 e 3.9 mostra a janela de configuração do PSUADE da forma padrão e preenchida com um exemplo de silumação.

PSUADE - Configuração

Método de Geração*: [dropdown menu]

Quantidade de Amostras*: 300

Número de Replicações: 1

Número de Refinamentos: 0

Tamanho Refinamento: 10000000

Número de Referência de Refinamento: 0

Tipo de Refinamento: [dropdown menu]

Semente Aleatória: [text field]

Adicionar Perturbações Mais Perturbações

Aplicar Cancelar

Figura 3.8: Tela de configuração com valores padrão para alguns campos.

PSUADE - Configuração

Método de Geração*: LH

Quantidade de Amostras*: 5

Número de Replicações: 1

Número de Refinamentos: 0

Tamanho Refinamento: 10000000

Número de Referência de Refinamento: 0

Tipo de Refinamento: [dropdown menu]

Semente Aleatória: 12504321

Adicionar Perturbações Mais Perturbações

Aplicar Cancelar

Figura 3.9: Tela de configuração com o método de geração Hipercubo Latino (LH, *Latin Hypercube*), cinco amostras e marcado para adicionar perturbações. Os outros valores estão com valores padrão.

3.3.3 Módulo de Configuração Geral

Finalmente, o terceiro módulo lida com aspectos gerais da aplicação. Esses aspectos são:

- salvar configuração atual;
- carregar configuração;
- encontrar executável do PSUADE;
- encontrar executável do MASE-BDI.

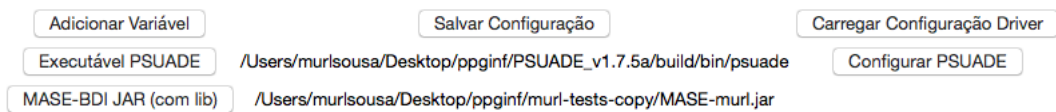


Figura 3.10: Tela contendo as opções de configuração gerais da ferramenta MASE-Driver-GUI.

A primeira função, salvar configuração atual, cria um arquivo JSON na máquina do usuário no local e nome da sua escolha. Essa configuração poderá ser carregada posteriormente através da opção carregar configuração, na qual o usuário irá escolher um arquivo JSON, previamente salvo pelo MASE-Driver-GUI, e a ferramenta carregará nos campos de variáveis e de configuração do PSUADE os valores salvos anteriormente. A opção “Executável do PSUADE” serve para que o usuário escolha o local onde está salvo o software PSUADE, dependendo do seu sistema operacional o programa estará em locais diferentes do computador. A opção “MASE-BDI Jar (com lib)” serve para apontar onde está no computador o *JAR* da ferramenta MASE-BDI e se assume que os arquivos de configuração necessários para o MASE-BDI se encontram na mesma pasta do arquivo *JAR*. A Figura 3.10 apresenta a tela com as opções e também um exemplo de como os caminhos da ferramenta PSUADE e do MASE-BDI são apresentados.

Por fim, após configurar as variáveis de entrada e saída e também configurar PSUADE basta clicar no botão “Executar” para que o PSUADE seja executado. Para executar a interface gráfica basta ter o Python instalado no computador e executar o comando *pythonMASE-Driver-GUI.py* (a depender do sistema operacional) em sistemas UNIX. No próximo Capítulo 4 serão apresentados os experimentos e resultados encontrados após utilização das ferramentas propostas.

Capítulo 4

Experimentos e Resultados

A partir da implementação da ferramenta MASE-Driver, foi possível integrar a ferramenta de soluções de análise de sensibilidade, PSUADE, e o ambiente de simulação ambiental, MASE-BDI, permitindo a realização de várias abordagens de geração de amostras bem como abrindo um leque muito grande de possibilidades de análise de incerteza e sensibilidade.

Alguns experimentos foram propostos para testar a integração MASE-BDI vs PSUADE, via MASE-Driver. O objetivo era verificar as interfaces e comunicação entre as diferentes ferramentas, bem como acompanhar a execução das simulações e análises, de modo a garantir que o MASE-Driver cumpriria o seu papel na integração. Foram utilizados diferentes métodos de amostragem, configurações de entrada e saída, e métodos de análise, cada um exigindo diferentes configurações da simulação do MASE-BDI, gerando diferentes resultados. As configurações de alguns desses experimentos são detalhadas a seguir.

4.1 Distribuição Uniforme - Amostragem LHS

Neste experimento, a configuração dos parâmetros de entrada da simulação é semelhante. Quatro variáveis são argumentos para uma simulação MASE-BDI. A Tabela 4.1 apresenta a extensão dos valores possíveis e qual é a função de distribuição de probabilidade utilizada para esse conjunto de experimentos. A entrada específica do arquivo *psuade.in* pode ser observada no Apêndice A.

Tabela 4.1: Variáveis de entrada da simulação MASE-BDI.

Variável	Valor Mínimo	Valor Máximo	Função de Densidade de Probabilidade
Qtde de Agentes Transformadores	1	100	Uniforme
Qtde de Agentes de Grupo	10	100	Uniforme
Potencial de Exploração Individual	1	500	Uniforme
Potencial de Exploração Grupo	1	1500	Uniforme

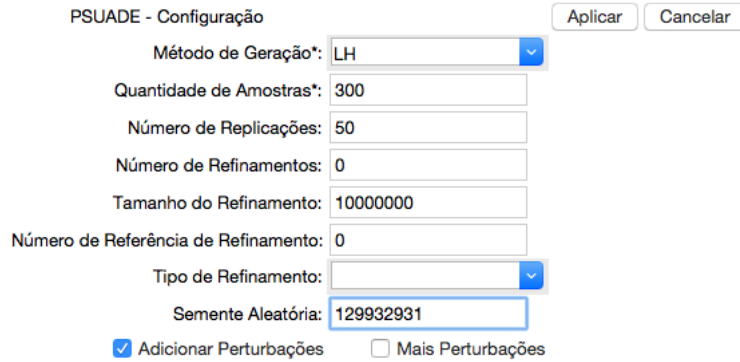
O conjunto de variáveis de saída de uma simulação MASE-BDI são apresentadas na Tabela 4.2. O PSUADE ordena essas variáveis de saída e todas as análises consideram essa referência, independente do método aplicado. Os gráficos no formato *.m* gerados pelo PSUADE também são exportados considerando essa posição, definida no arquivo *psuade.in*.

Tabela 4.2: Variáveis de saída.

Posição	Variável de Saída	SIGLA
1	Tempo	TE
2	Quantidade de Agentes	QA
3	Porcentagem de Agentes	PA
4	Figura de mérito	FM
5	Acurácia dos Produtores	AP
6	Acurácia dos Usuários	AU
7	Wrong Change	WC
8	Right Change	RC
9	Wrong Persistence	WP
10	Modelo Nulo	MN
11	Modelo Nulo Simulado	MS
11	Passos	PS

O método de amostragem selecionado no PSUADE foi a amostragem LHS com replicação. Nesse experimento, o tamanho da amostra foi determinado como 300 e nenhum refinamento é utilizado (algumas opções de refinamento estão disponíveis no PSUADE). É possível utilizar a amostragem LHS sem replicação, entretanto a replicação é útil quando se deseja fazer algumas análises globais de sensibilidade, um dos propósitos desse experimento. Nesses exemplos, com 50 replicações, o número de níveis é 6 o cálculo de quantidade de amostras/número de replicações. Além disso, a *flag* de randomização é adicionada à amostra, para informar ao método de amostragem que certa variação aleatória é permitida na amostra. De modo opcional, a semente de geração do número randômico também é fornecida. Isso é especialmente útil quando se deseja que os métodos de amostragem sejam reproduzíveis. Essas configurações também são definidas no arquivo de entrada do *psuade*, e parte da especificação é ilustrada abaixo.

O MASE-Driver-GUI foi utilizado para integrar a ferramenta PSUADE e o simulador MASE-BDI. Todas as amostras geradas a partir das definições mencionadas anteriormente são comunicadas ao MASE-BDI via o driver implementado nesse trabalho. O MASE-BDI simula o uso e a cobertura da terra com os parâmetros gerados pela amostra e gera os mapas e as métricas de saída. O resultado de cada uma dessas simulações é retornado ao PSUADE, via driver, de modo que as análises possam ser feitas. A seção de análise especifica o tipo de análise a ser executada e os parâmetros a serem utilizados. Para esse experimento, três métodos de análises foram executados para verificar o funcionamento do MASE-Driver. Cada análise é explicitada a seguir.



PSUADE - Configuração Aplicar Cancelar

Método de Geração*: LH

Quantidade de Amostras*: 300

Número de Replicações*: 50

Número de Refinamentos*: 0

Tamanho do Refinamento*: 10000000

Número de Referência de Refinamento*: 0

Tipo de Refinamento*:

Semente Aleatória*: 129932931

Adicionar Perturbações Mais Perturbações

Figura 4.1: Configuração da ferramenta MASE-Driver-GUI para o experimento com LHS.

4.1.1 Avaliação de Incerteza

O método de Avaliação da Incerteza é identificado pelo PSUADE sob a sigla *ua* (*Uncertainty Analysis*). Esse comando é usado após uma amostra ter sido carregada na memória local. Essa análise apresenta informações estatísticas básicas como média, desvio padrão, assimetria (*skewness*) e curtose. Esse comando também cria gráficos de distribuição dos dados aferidos. Esse método é recomendado para avaliar o impacto das incertezas dos parâmetros na saída do modelo de interesse.

4.1.2 Triagem de Parâmetros

O método de Triagem de Parâmetros é identificado pelo PSUADE sob a sigla *ca*. Esse comando é usado após uma amostra ter sido carregada na memória local. Essa análise exibe informações de correlação, como Pearson, Spearman e coeficientes de Kendall. O propósito desse tipo de análise é identificar um pequeno subconjunto de parâmetros que representam a maioria da variabilidades de saída de uma simulação.

Existem vários métodos de triagem de parâmetros que podem ser úteis em cenários diferentes. Os coeficientes de Pearson são indicadores de sensibilidade úteis quando a relação de entrada-saída é aproximadamente linear. Para as funções não-lineares, mas monótonas, o coeficiente de Spearman é o mais indicado. Para relacionamentos não-lineares gerais, esta análise não é recomendada.

4.1.3 Análise de Sensibilidade

A análise quantitativa de sensibilidade procura quantificar o impacto de um subconjunto particular de parâmetros sobre as incertezas de saída. Esse tipo de análise pode identificar o efeito principal, o efeito de interação em pares, o efeito principal de grupo e até uma análise total de sensibilidade. Geralmente, a análise de sensibilidade quantitativa exige uma grande amostra e é, muitas vezes, realizada em superfícies de resposta validadas, a menos que as simulações sejam computacionalmente baratas. A análise de Efeito Principal foi a que teve seus índices calculados no experimento. Essa análise estuda a sensibilidade de primeira ordem dos parâmetros individuais de entrada da simulação, com base na decomposição da variância. Os índices de Sensibilidade (Sobol) foram utilizados com o comando *eu* na amostra LHS já disponível. A decomposição baseada em variância

(*Variance-based decomposition* - VBD) é um método de sensibilidade global que resume como a incerteza no resultado do modelo pode ser imputado à incerteza nas variáveis de entrada individuais. O Método PSUADE Efeito Principal é a técnica VBD que será exemplificada nesse experimento. VBD utiliza duas medidas preliminares, o índice de sensibilidade do efeito principal S_i e o índice de efeito total T_i . Estes índices são também chamados índices Sobol. O índice de sensibilidade do efeito principal corresponde à fração da incerteza na saída, Y , que pode ser atribuída à entrada x_i sozinha.

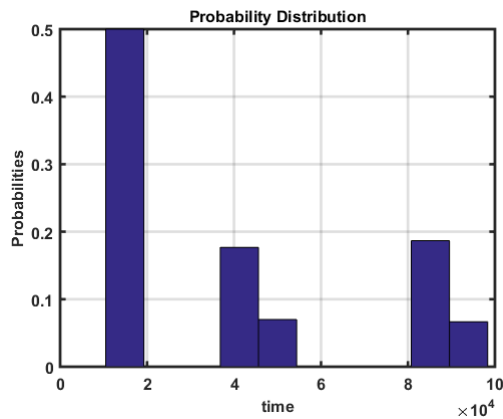
O MASE-Driver foi capaz de integrar de modo eficiente a execução da ferramenta de simulação MASE-BDI e da ferramenta de análises quantitativas PSUADE, ao definir um método de comunicação e coordenação sobre as interfaces desses diferentes softwares. Um extenso conjunto de amostras foram geradas, sob diferentes métodos, e a execução do MASE-BDI foi requisitada para cada uma dessas amostras, exigindo a organização das simulações e coordenação das métricas de saída. Esses resultados foram então retroalimentados via MASE-Driver ao PSUADE, e uma sequência de experimentos foi feita.

4.2 Avaliação de Incerteza

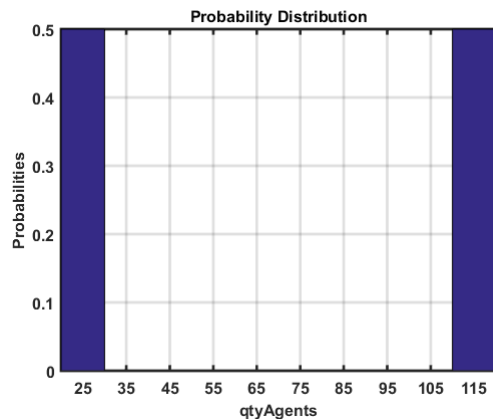
Os resultados gerados pelo algoritmo PSUADE de avaliação de incerteza foram resumidos na Tabela 4.3. Algumas das métricas calculadas pelo PSUADE foram suprimidas. Optou-se por apresentar um panorama geral, embora superficial, da quantificação da incerteza. É importante frisar que todos os dados da tabela referem-se à amostra, portanto, onde lê-se média, compreende-se a média da amostra. Algumas variáveis foram suprimidas por já estarem contidas nas premissas desse experimento. Por exemplo, a métrica assimetria (*skewness*) foi igual a zero e a curtose foi igual a 0.5 para todos as variáveis, medidas previsíveis, uma vez que o ponto de partida tinha as variáveis de entrada definidas com uma distribuição uniforme.

Tabela 4.3: Avaliação de Incerteza. Entrada: distribuição uniforme e amostragem: LHS.

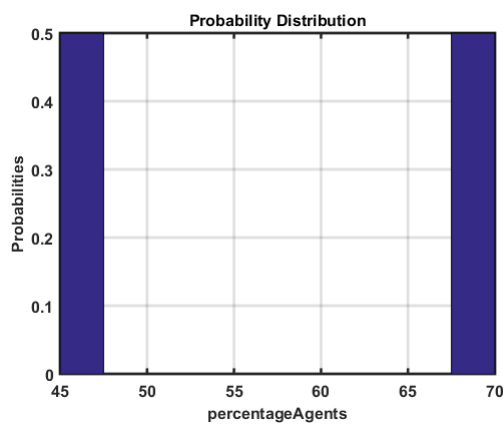
Variável	Média	Desvio Padrão	SE Média	SE Variância
TE	55.506	60.580	107.255.264	1.703.902.850.000.000.000
QA	70	70,71	0	0
PA	57,5	17,68	0	0
FM	32,4	26,46	19,13	59.681,75
AP	44,39	42,54	4,26	434.558,74
UA	68,97	9,91	21,12	289,40
WC	6,6	7,65	1,57	549,85
RC	11,66	11,18	0,29	2.071,03
WP	14,61	11,18	0,29	2.071,03
MN	0,26	0	0	0
MS	0,21	0,04	0	0
PS	295,50	98,29	1198,94	23.005.097,30



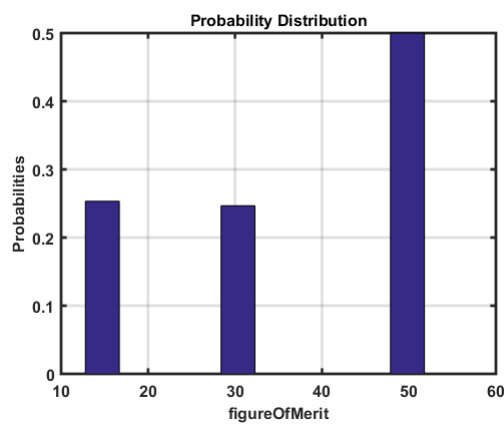
(a) Tempo



(b) Quantidade de Agentes



(c) Porcentagem de Agentes



(d) Figura de Mérito

Figura 4.2: Distribuição de Probabilidades para as variáveis de saída do modelo MASE-BDI

Os valores encontrados para a análise do MASE-BDI fornecida pelo PSUADE são consistentes com os resultados quantitativos decorrentes de cerca de 85 mil simulações documentadas do MASE-BDI, em um experimento de caracterização de carga realizado pelo grupo de pesquisa desse projeto. Para a variável tempo, por exemplo, é possível observar a enorme variação do erro padrão da média e da variância (*Standard Error-SE*). O erro padrão da média estima a variabilidade entre as médias amostrais enquanto o desvio padrão mede a variabilidade em uma única amostra.

De fato, observa-se uma intensa variação entre o tempo de execução das simulações MASE-BDI. Observe a distribuição de probabilidade da variável tempo, ilustrada na Figura 4.2. É possível notar que grande parte das simulações ocorre dentro uma faixa determinada de tempo, e que outra parte, em menor quantidade, chega a demorar 5 vezes mais. Observando o modelo de execução do MASE-BDI, é possível prever que em alguns casos em que os agentes passam mais tempo resolvendo os seus conflitos e se movimentando pelo mapa, menos tempo eles passam explorando e mais tempo leva a simulação. Esse comportamento também foi notado em experimentos anteriores e cabem estudos mais aprofundados para identificar a causa dessa possível lentidão.

A média da Quantidade de Agentes (QA) de 70 por simulação também é um dado

corroborado por experimentos anteriores da equipe. O valor de qualidade da Figura de Mérito (FM) em 32,4 também estão consoantes aos testes iniciais feitos com o MASE-BDI e descritos em [8]. Além disso, o pequeno desvio padrão da variável Modelo Nulo Simulado indica que, em geral, independente da amostra, as simulações apresentam resultados semelhantes, todos eles passando no teste do modelo nulo.

A Figura 4.2 mostra algumas das distribuições de probabilidade para as variáveis de saída do MASE-BDI. Observa-se, de modo geral, que são variáveis aleatórias discretas com funções de probabilidade bem definidas.

4.3 Triagem de Parâmetros

A triagem de parâmetros foi realizada por meio da análise de correlação. Nem todas as estatísticas foram aqui replicadas. Apenas alguns dados mais relevantes serão apresentados. A correlação de Pearson, ou coeficiente de correlação produto-momento, das variáveis de saída foi calculada e apresentada na Tabela 4.4. Os dados da tabela mostram o grau da correlação e a direção da correlação (positiva ou negativa) entre duas variáveis. Esse coeficiente é adequado quando as variáveis apresentam uma relação linear, o que já havia sido detectado pela caracterização de carga que investigou os parâmetros do MASE-BDI. O coeficiente de Pearson varia de -1 a 1. Uma correlação perfeita (-1 ou 1) indica que o escore de uma variável pode ser determinado exatamente ao se saber o escore da outra. Esse é o caso da relação entre o parâmetro *Right Change* (RC) e o Acurácia do Produtor (AP), que é um comportamento esperado do MASE-BDI, uma vez que a acurácia do produtor apresenta a probabilidade de que uma amostra de referência (uma classe de cobertura da terra simulada pelo MASE-BDI) será corretamente mapeada, considerando os erros de omissão. A variável RC é a quantificação de quantos *pixels* observados como mudanças na cobertura da terra foram de fato preditos corretamente. Por essa relação, fica claro que esse coeficiente não diferencia entre variáveis independentes e variáveis dependentes, que deve ser investigada por outros meios estatísticos não contemplados nesse experimento. Algumas correlações confirmam o comportamento esperado do MASE-BDI. É possível verificar que os parâmetros Tempo (TE) e QA tem uma forte correlação positiva de mais de 0,85, o que implica que se QA está acima da média, o TE também estará acima da média. Como o coeficiente é adimensional, não pode ser considerado como uma proporção (considerar que a correlação acontece em 85% dos casos, por exemplo). Do mesmo modo, os dados mostram que a QA possui uma correlação positiva forte com a métrica de qualidade FM. É possível observar também que o número de Passos (PS) de uma simulação apresenta uma correlação negativa com a FM, o que poderia indicar que simulações mais longas tendem a ter uma qualidade reduzida.

Tabela 4.4: Coeficientes de correlação de Pearson, saídas X saídas. Entrada: distribuição uniforme e amostragem: LHS.

Variável	TE	QA	PA	FM	AP	UA	WC	RC	WC	MN	MS	PS
QA	0,8595	0	-0,0267	0,9261	0,9526	-0,7989	0,8956	0,9526	-0,9526	0	-0,7637	-0,5825
PA	-0,0578	-0,0267	0	-0,0158	-0,0195	-0,0623	-0,0269	-0,0195	0,0195	0	0,23	0,0174
FM	0,8075	0,9261	-0,0158	0	0,9812	-0,5949	0,8396	0,9812	-0,9812	0	-0,9145	-0,5311
AP	0,9033	0,9526	-0,0195	0,9812	0	-0,7309	0,9286	1	-1	0	-0,8195	-0,6742
UA	-0,9024	-0,7989	0,0623	-0,5949	-0,7309	0	-0,9091	-0,7039	0,7309	0	-0,7637	0,8462
WC	0,9913	0,8956	-0,0269	0,8396	0,9286	-0,9091	0	0,9286	-0,9286	0	0,2431	-0,8818
RC	0,9033	0,9526	-0,0195	0,9812	1	-0,7309	0,9286	0	-1	0	-0,5483	-0,6742
WP	-0,9033	-0,9526	0,0195	-0,9812	1	0,7309	-0,9286	-1	0	0	-0,8195	0,6742
MN	0	0	0	0	0	0	0	0	0	0	0	0
MS	-0,5046	-0,7637	0,23	-0,9145	0,8195	0,2431	-0,5483	-0,8195	-0,8195	0	0	0,1574
PS	-0,9045	-0,5825	0,0174	-0,5311	0,6742	0,8462	-0,8818	-0,6742	-0,6742	0	0,1574	0

As estatísticas geradas pelo PSUADE também foram compiladas em gráficos que apresentam a correlação. Os gráficos de algumas das variáveis foram selecionados para representar os resultados e estão ilustrados na Figura 4.3. Em cada gráfico, o eixo y apresenta o coeficiente de correlação de Pearson (no gráfico à esquerda) ou o coeficiente de correlação de Spearman (gráfico à direita), enquanto o eixo x apresenta as quatro variáveis discretas de entrada da simulação MASE-BDI em quatro colunas: Coluna 1 - Quantidade de Agentes Transformadores (*transformationAgentQty*), Coluna 2 - Quantidade de Agentes de Grupo (*transformationAgentGroupPercentage*), Coluna 3 - Potencial de Exploração Individual (*individualExploration*) e Coluna 4 - Potencial de Exploração de Grupo (*groupExploration*).

Os gráficos apresentam correlações que reafirmam algumas das conclusões observadas após a análise de incerteza. O tempo de simulação tem correlação positiva perfeita com a quantidade de agentes transformadores. De modo oposto, a quantidade de passos apresenta correlação negativa com o potencial de exploração do grupo e a quantidade de agentes transformadores.

4.4 Análise de Sensibilidade

A análise de sensibilidade foi realizada por meio do componente Efeito Principal. As estatísticas geradas pelo PSUADE foram compiladas em gráficos apresentados nas Figuras 4.4 e 4.5. Em cada gráfico, o eixo y apresenta o índice de Sobol enquanto o eixo x apresenta as quatro variáveis discretas de entrada da simulação MASE-BDI em quatro colunas: Coluna 1 - Quantidade de Agentes Transformadores (*transformationAgentQty*), Coluna 2 - Quantidade de Agentes de Grupo (*transformationAgentGroupPercentage*), Coluna 3 - Potencial de Exploração Individual (*individualExploration*) e Coluna 4 - Potencial de Exploração Grupo (*groupExploration*).

Esse tipo de análise quantifica o impacto dos parâmetros de entrada da simulação sobre as incertezas de cada variável de saída. Inicialmente, o gráfico que apresenta o efeito principal acerca da saída modelo nulo destoa dos demais, por aparecer em branco. Esse comportamento é esperado, uma vez que o cálculo do modelo nulo depende do cálculo de alteração da terra baseada nas imagens de satélite, no caso do DF, nos mapas de 2002 e 2008. Como essa variável de saída não possui nenhuma relação com as variáveis de entrada, os índices de Sobol são iguais a zero.

É possível perceber que o parâmetro de entrada Quantidade de Agentes Transformadores é totalmente determinante em todas as variáveis de saída (já excluindo a exceção da variável de saída modelo nulo), com índices de Sobol atingindo o teto da estatística. Ver tamanho efeito de somente uma variável leva à reflexão de que a simulação em si seja extremamente dependente da quantidade de agentes transformadores agindo sobre a terra, alterando a cobertura dos mapas simulados. De um modo é uma constatação racional: quanto mais agentes, maior o nível de exploração e intensidade da simulação ambiental. Entretanto, uma análise mais profunda deve ser conduzida para que se elimine a possibilidade de que o resultado da simulação seja uma resposta quase determinística com relação ao número de agentes.

Somente uma segunda variável de entrada se destaca, mesmo que de modo periférico, à influência da Quantidade de Agentes Transformadores. O parâmetro De Potencial de Exploração do Grupo possui influência secundária na maioria das variáveis de saída. Para

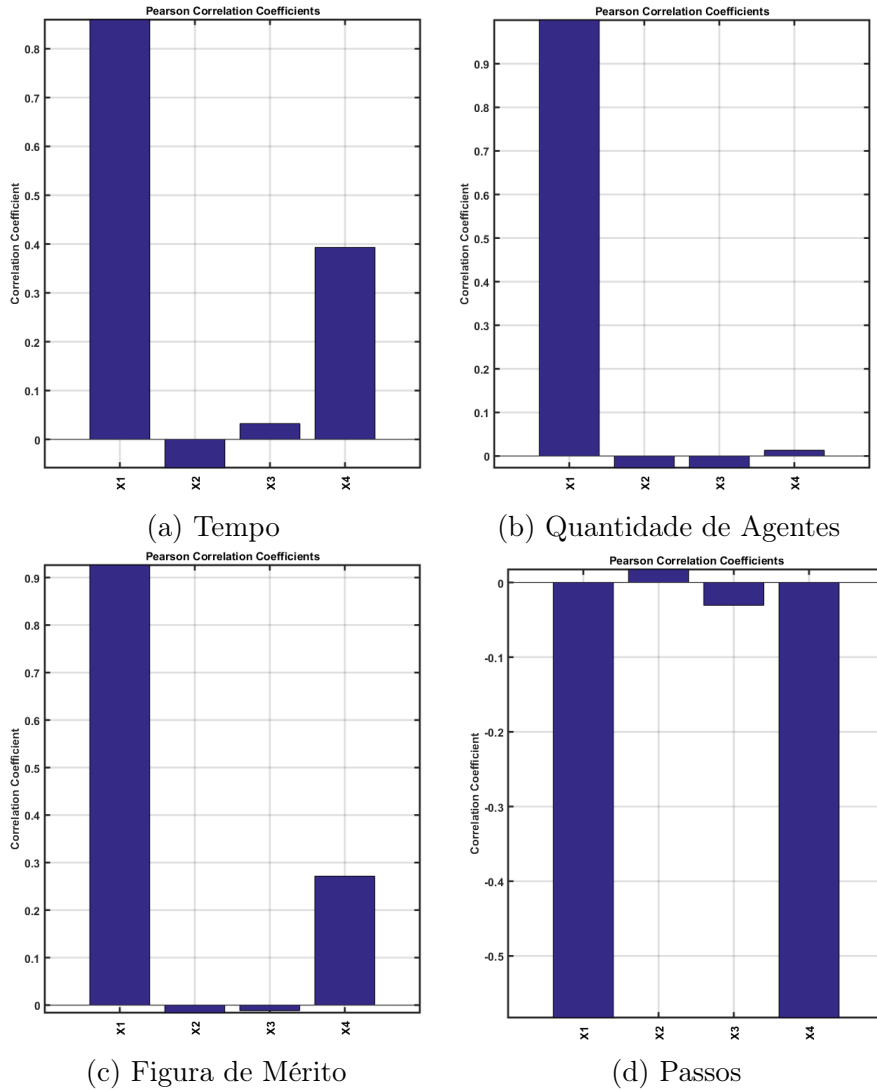
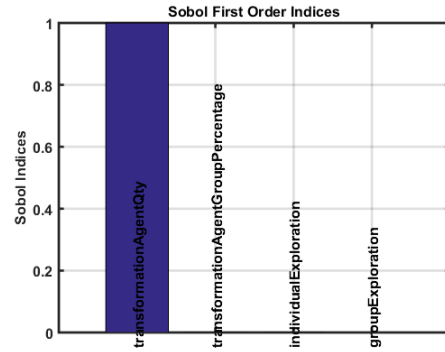


Figura 4.3: Análise de Correlação - Coeficientes de Pearson e Spearman

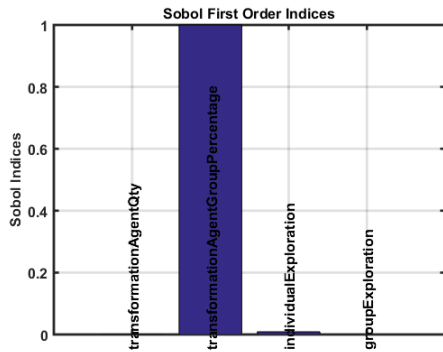
a variável de saída PS, que representa a quantidade de passos na simulação, a variável potencial de exploração do grupo tem igual relevância à quantidade de agentes transformadores. Mais uma vez, a relação causal parece evidente: Quanto maior o potencial de exploração, maior o impacto no desmatamento e alteração do uso da terra na simulação. Como essa variável pode ser até três vezes maior que o potencial de exploração individual, tem-se clara a consequência na simulação. No Capítulo 5 serão apresentados as conclusões e trabalhos futuros.



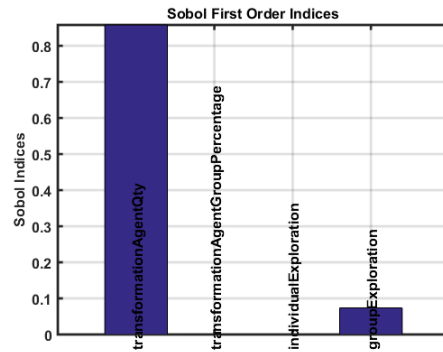
(a) Tempo



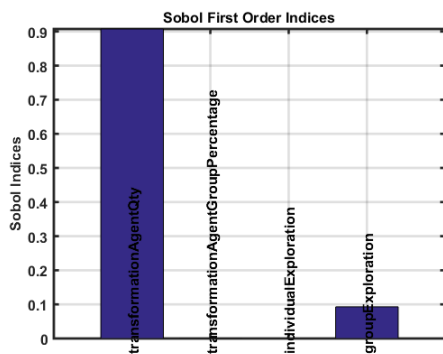
(b) Quantidade de Agentes



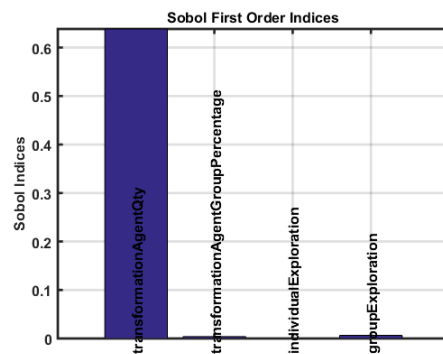
(c) Porcentagem de Agentes



(d) Figura de Mérito



(e) Acurácia dos Produtores

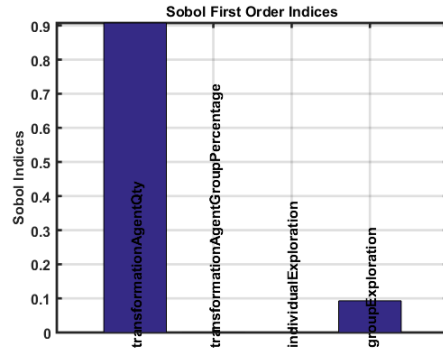


(f) Acurácia dos Usuários

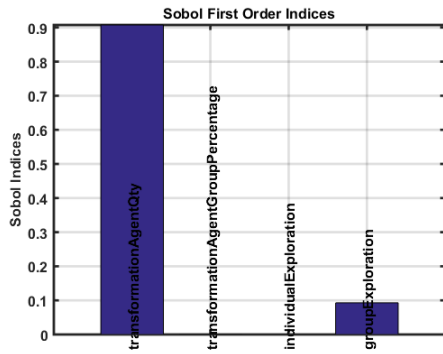
Figura 4.4: Efeito Principal dos parâmetros de saída do modelo MASE-BDI, em relação aos parâmetros de entrada.



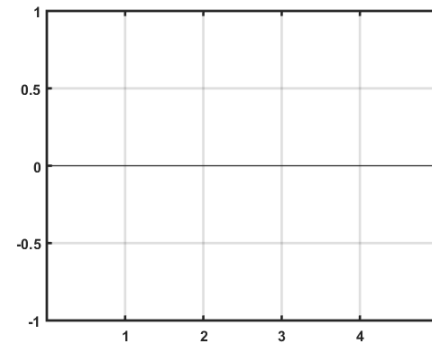
(a) *Wrong Change*



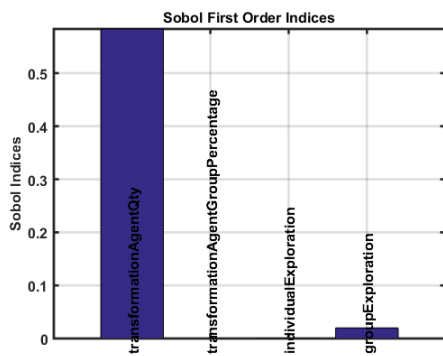
(b) *Right Change*



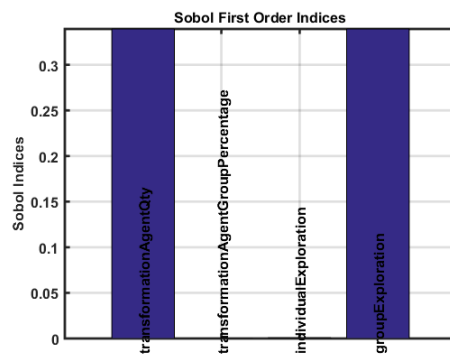
(c) *Wrong Persistence*



(d) *Modelo Nulo*



(e) *Modelo Nulo Simulado*



(f) *Passos*

Figura 4.5: Efeito Principal dos parâmetros de saída do modelo MASE-BDI, em relação aos parâmetros de entrada- Parte II

Capítulo 5

Conclusões e Trabalhos Futuros

Neste trabalho foi feita a integração do simulador ambiental MASE-BDI com uma ferramenta de quantificação de incerteza e análise de sensibilidade local e global, denominada PSUADE. Essa integração resultou na implementação do MASE-Driver, o qual viabiliza diversas análises relacionadas a aspectos de incerteza incluindo sua caracterização, evolução e impactos. Para que essas análises fossem realizadas de forma mais rápida e transparente ao usuário foi desenvolvido o MASE-Driver-GUI, o qual abstrai detalhes da entrada de dados no PSUADE ou saída de dados do MASE-BDI, tornando a interação com o MASE-Driver mais amigável.

A implementação do MASE-BDI e MASE-Driver-GUI adotou práticas de desenvolvimento de software que permite a evolução dessas ferramentas de forma adequada e podendo se tornar o principal meio de interação com o MASE-BDI. Foi utilizado a linguagem Python por ser uma linguagem robusta, que permite, por exemplo, estender a entrada de dados para o MASE-Driver através de protocolos de redes de uma forma simples por meio da interface de programação de aplicação (*Application Programming Interfaces*, API). Foi possível alcançar com simplicidade uma interface robusta, que impede inserção de dados errôneos e que ao mesmo tempo comunica tudo que o usuário precisa para realizar a utilização do MASE-BDI e PSUADE.

Como trabalhos futuros pode-se citar:

- Aprimorar a interface gráfica para um modelo mais genérico, possibilitando utilizar a interface para qualquer driver e não somente o MASE-Driver;
- Abranger outras possibilidades de configurações do PSUADE via interface gráfica, uma vez que atualmente o usuário deve interagir diretamente com os arquivos de configuração do PSUADE para utilizar todas as suas configurações;
- Fazer com que a ferramenta seja parte integrante do MASE-BDI ou melhorar a interface com o MASE-BDI, automatizando, por exemplo, quais variáveis estão disponíveis para edição pelo usuário.

Referências

- [1] C. G. Abreu, C. G. C. Coelho, and C. G. Ralha. (mase-bdi): Agents with practical reasoning for land use and cover change simulation. In *Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais (WCAMA)*, pages 1 – 10. Sociedade Brasileira de Computação, 2015. [x](#), [9](#)
- [2] Carolina Gonçalves Abreu. *Ferramenta de Simulação com Abordagem de Sistema Multiagente para a Dinâmica de Uso da terra*. Msc, Departamento de Ciência da Computação, Universidade de Brasília, 2012. [8](#)
- [3] B. M. Adams, M. S. Ebeida, M. S. Eldred, J. D. Jakeman, L. P. Swiler, J. A. Stephens, D. M. Vigil, and T. M. Wildey. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis Version 6.3 User’s Manual SAND2014-4633. Technical Report July 2014, Sandia National Laboratories, 2015. [12](#)
- [4] K. Alden, M. Read, J. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. Coles. Spartan: A comprehensive tool for understanding uncertainty in simulations of biological systems. *PLoS Comput Biol*, 9(2):1–9, 02 2013. [13](#)
- [5] F. Bousquet, C. Cambier, C. Mullon, P. Morand, J. Quensiere, and A. Pavé. Simulating the interaction between a society and a renewable resource. *Journal of Biological Systems*, 01(02):199–214, jun 1993. [7](#)
- [6] F. Bousquet and C. Le Page. Multi-agent simulations and ecosystem management: A review. *Ecological Modelling*, 176(3-4):313–332, September 2004. [6](#), [8](#)
- [7] M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987. [7](#)
- [8] C. G. C. Coelho. *Agentes Racionais Baseados no Modelo Belief-Desire-Intention para o Sistema Multiagente*. Msc, Departamento de Ciência da Computação, Universidade de Brasília, 2014. [2](#), [8](#), [33](#)
- [9] D. C. Dennett. Précis of The Intentional Stance. *Behavioral and Brain Sciences*, 11(03):495–505, 1988. [7](#)
- [10] Ibama. Monitoramento do bioma Cerrado 2002 a 2008. Monitoramento do desmatamento nos biomas brasileiros por satélite: acordo de cooperação técnica MMA/IBAMA. Technical report, Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis, 2009. [x](#), [10](#), [11](#)

- [11] N. R. Jennings, K. Sycara, and M. Wooldridge. A Roadmap of Agent Research and Development. *Int. Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):31, 1998. 6
- [12] W. Jepson. A disappearing biome? Reconsidering land-cover change in the Brazilian savanna. *Geographical Journal*, 171(2):99–111, 2005. 10
- [13] C. A. Klink and R. B. Machado. A conservação do Cerrado brasileiro. *Megadiversidade*, 1(1):147–155, 2005. 10
- [14] K. C. Laudon and J. P. Laudon. *Management Information Systems: Managing the Digital Firm*. Prentice Hall, 13 edition, 2013. 4
- [15] R. J. Marquis and P. S. Oliveira. *The Cerrados of Brazil: Ecology and Natural History of a Neotropical Savana*. Columbia University Press, New York, 1st edition, 2002. 10
- [16] D. C. Parker, S. M. Manson, M. A. Janssen, M. J. Hoffmann, and P. Deadman. Multi-agent systems for the simulation of land-use and land-cover change: A review. *Annals of the Association of American Geographers*, 93(2):314–337, 2003. 2
- [17] S. F. Railsback, S. L. Lytinen, and S. K. Jackson. Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation*, 82(9):609–623, September 2006. 8
- [18] C. G. Ralha, C. G. Abreu, C. G. C. Coelho, A. Zaghetto, B. Macchiavello, and R. Machado. A multi-agent model system for land-use change simulation. *Environ. Model. Softw.*, 42:30–46, April 2013. 1, 2, 8
- [19] A. S. Rao and M. P. Georgeff. Bdi agents: From theory to practice. In *In Proceedings of the First International Conference on Multi-agent systems (ICMAS-95)*, pages 312–319, 1995. 7
- [20] S. J Russel and P. Norvig. *Artificial Intelligence: a modern approach*. Prentice Hall, 2nd edition, 2010. 6
- [21] H. Dowlatabadi S. M. Blower. Sensitivity and uncertainty analysis of complex models of disease transmission: An hiv model, as an example. *International Statistical Review / Revue Internationale de Statistique*, 62(2):229–243, 1994. 12
- [22] E. E. Sano, R. Rosa, J. L. S. Brito, and L. G. Ferreira. Mapeamento semidetalhado do uso da terra do Bioma Cerrado/Semidetailed land use mapping in the Cerrado. *Pesquisa Agropecuária Brasileira*, 43(1):153–156, January 2008. 10
- [23] P. H. Verburg. Simulating feedbacks in land use and land cover change models. *Landscape Ecology*, 21(8):1171–1183, 2003. 2
- [24] G. Weiss, editor. *Multiagent Systems*. MIT Press, Cambridge, MA, USA, 2nd edition, 2013. 6

- [25] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition, 2009. 6

Apêndice A

PSUADE

1. Arquivo de Entrada

Exemplo de arquivo de entrada para o PSUADE que especifica que o método de geração de amostras será *Latin Hypercube*, gerando 300 amostras, quatro variáveis de entrada e 12 de saída:

PSUADE

INPUT

```
dimension = 4
variable 1 transformationAgentQty = 1 100
variable 2 transformationAgentGroupPercentage = 10 100
variable 3 individualExploration = 1 500
variable 4 groupExploration = 1 1500
```

END

OUTPUT

```
dimension = 12
variable 1 time
variable 2 qtyAgents
variable 3 percentageAgents
variable 4 figureOfMerit
variable 5 producersaccuracy
variable 6 usersaccuracy
variable 7 wrongchange
variable 8 rightchange
variable 9 wrongpersistance
variable 10 nullModel
variable 11 simulatedNullModel
variable 12 steps
```

END

METHOD

```
sampling = LH
num_samples = 300
randomize
random_seed = 12504321
```

```

END
APPLICATION
    driver = ./MASE-Driver.py
END
ANALYSIS
    analyzer output_id = 1
    printlevel 1
END
END

```

2. Exemplo de arquivo de saída

Exemplo de arquivo que o PSUADE gera como entrada para o *driver* (configuração definida no arquivo *psuade.in*). Este arquivo de entrada tem quatro valores, um para cada variável definida no arquivo de configuração.

```

4
4.4077203315252991e+00
2.3059819407323289e+01
3.3939929566939020e+02
5.3074622200728061e+02

```

3. Exemplo de saída de uma execução do PSUADE com cinco amostras:

```

PSUADE_IO (Note : inputs not true inputs if pdf ~=U)
4 12 5
1 1
6.5221003675004937e+01
8.3402061597165684e+01
3.0402182771713564e+02
2.6778385731847209e+01
2.0501000000000000e+04
1.3000000000000000e+02
8.3000000000000000e+01
7.2374805353072595e+00
7.4347977280630584e+00
7.3169062286105401e+01
7.1632845121382949e-01
1.9534569243560809e+00
2.4321056462640975e+01
2.6274513386997056e-01
2.5037384913854804e-01
3.6500000000000000e+02
2 1
9.7612277671234821e+01
1.1722795566414854e+01

```

2.8365121975776327e+02
9.1055380339760052e+02
1.0116700000000000e+05
1.9400000000000000e+02
1.1000000000000000e+01
5.1103370947348338e+01
7.4490552915266022e+01
6.1943823257248610e+01
1.2024389278219900e+01
1.9572030297769697e+01
6.7024830892273579e+00
2.6274513386997056e-01
1.8726872367447259e-01
1.7700000000000000e+02

3 1

1.3727632153279908e+01
2.8702295384696821e+01
4.9627162071600168e+02
7.5717581582003083e+02
1.2886000000000000e+04
2.6000000000000000e+01
2.8000000000000000e+01
2.7217953195948308e+01
2.8904601831459370e+01
8.2345948087973056e+01
1.6281853317130663e+00
7.5945434776649892e+00
1.8679969909332065e+01
2.6274513386997056e-01
2.0308155241045134e-01
3.6500000000000000e+02

4 1

2.3804368331378498e+01
5.5540561429010971e+01
1.0049253925601604e+02
4.3074101593370597e+02
1.5580000000000000e+04
4.6000000000000000e+01
5.5000000000000000e+01
2.9921838682825054e+01
3.2082995247478848e+01
8.1624347518357951e+01
1.8977221851672499e+00
8.4296508812484578e+00
1.7844862505748598e+01
2.6274513386997056e-01


```

1.9742584690915846e-01
3.6500000000000000e+02
5 1
5.1023511104482928e+01
6.4046857663452101e+01
1.8091600955804626e+02
1.3565082620715295e+03
9.4924000000000000e+04
1.0200000000000000e+02
6.4000000000000000e+01
5.1044996898411036e+01
7.4400139098180134e+01
6.1920564940139109e+01
1.2021648225472907e+01
1.9548274507295769e+01
6.7262388797012864e+00
2.6274513386997056e-01
1.8747887105174194e-01
2.4600000000000000e+02
PSUADE_IO
PSUADE
INPUT
    dimension = 4
    variable 1 transformationAgentQty = 1 1e+02
    variable 2 transformationAgentGroupPercentage = 1e+01 1e+02
    variable 3 individualExploration = 1 5e+02
    variable 4 groupExploration = 1 1.5e+03
END
OUTPUT
    dimension = 12
    variable 1 time
    variable 2 qtyAgents
    variable 3 percentageAgents
    variable 4 figureOfMerit
    variable 5 producersaccuracy
    variable 6 usersaccuracy
    variable 7 wrongchange
    variable 8 rightchange
    variable 9 wrongpersistance
    variable 10 nullModel
    variable 11 simulatedNullModel
    variable 12 steps
END
METHOD
# sampling = MC
# sampling = FACT

```

```

    sampling = LH
# sampling = OA
# sampling = OALH
# sampling = MOAT
# sampling = SOBOL
# sampling = LPTAU
# sampling = METIS
# sampling = FAST
# sampling = BBD
# sampling = PBD
# sampling = FF4
# sampling = FF5
# sampling = CCI4
# sampling = CCI5
# sampling = CCIF
# sampling = CCF4
# sampling = CCF5
# sampling = CCFF
# sampling = CCC4
# sampling = CCC5
# sampling = CCCF
# sampling = SFAST
# sampling = UMETIS
# sampling = GMOAT
# sampling = GMETIS
# sampling = SPARSEGRID
# sampling = LSA
# sampling = RFF4
# sampling = RFF5
    num_samples = 5
    num_replications = 1
    num_refinements = 0
    refinement_size = 10000000
    reference_num_refinements = 0
# refinement_type = adaptive
    randomize
# randomize_more
    random_seed = 12504321
END
APPLICATION
    driver = ./MASE-Driver.py
    opt_driver = NONE
    aux_opt_driver = NONE
    ensemble_driver = NONE
    ensemble_opt_driver = NONE
# max_parallel_jobs = 1

```

```

# min_job_wait_time = 1
# max_job_wait_time = 1000000
# nondeterministic
# launch_only
# limited_launch_only
# gen_inputfile_only
# ensemble_run_mode
# launch_interval = 1
# save_frequency = 1000000
END
ANALYSIS
# analyzer method = Moment
# analyzer method = MainEffect
# analyzer method = TwoParamEffect
# analyzer method = ANOVA
# analyzer method = GLSA
# analyzer method = RSFA
# analyzer method = MOAT
# analyzer method = Sobol
# analyzer method = Correlation
# analyzer method = Integration
# analyzer method = FAST
# analyzer method = FF
# analyzer method = PCA
# analyzer method = ARSMGP
# analyzer method = FORM
# analyzer method = RSMSobol1
# analyzer method = RSMSobol2
# analyzer method = RSMSobolTSI
# analyzer method = Bootstrap
# analyzer method = RSMSobolG
# analyzer method = ARSMNN
# analyzer method = ARSM
# analyzer method = REL
# analyzer method = AOPT
# analyzer method = GOWER
# analyzer method = DELTA
# analyzer method = ETA
# analyzer method = ARSM
# analyzer method = LSA
# analyzer output_id = 1
# analyzer rstype = MARS
# analyzer rstype = linear
# analyzer rstype = quadratic
# analyzer rstype = cubic
# analyzer rstype = quartic

```

```

# analyzer rstype = selective_regression
# analyzer rstype = GP1
# analyzer rstype = GP2
# analyzer rstype = SVM
# analyzer rstype = PWL
# analyzer rstype = TGP
# analyzer rstype = MARSBag
# analyzer rstype = EARTH
# analyzer rstype = sum_of_trees
# analyzer rstype = Legendre
# analyzer rstype = user_regression
# analyzer rstype = sparse_grid_regression
# analyzer rstype = Kriging
# analyzer rstype = splines
# analyzer rstype = KNN
# analyzer rstype = RBF
# analyzer rstype = Acosso
# analyzer rstype = Bssanova
# analyzer rstype = psuade_regression
# analyzer rstype = RBFBag
# analyzer rs_legendre_order = -1
# analyzer rs_mars_num_bases = -1
# analyzer rs_mars_interaction = -1
# analyzer rs_num_mars = -1
# analyzer rs_kriging_mode = -1
# analyzer rs_kriging_tol = -1
# analyzer opt_save_history
# analyzer opt_use_history
# analyzer regression_wgt_id = -1
# graphics
# sample_graphics
  analyzer threshold = 1.000000e+00
  rs_max_pts = 10000
# analyzer rs_constraint = psData indexFile Lbnd Ubnd
# analyzer moat_constraint = psData indexFile Lbnd Ubnd
# analyzer rs_index_file = indexFile
# optimization method = crude
# optimization method = txmath
# optimization method = appspack
# optimization method = minpack
# optimization method = sm
# optimization method = mm
# optimization method = mm_adaptive
# optimization method = bobyqa
# optimization method = sce
# optimization method = moo

```

```
# optimization method = ouu
# optimization method = ouu1
# optimization method = ouu2
# optimization num_local_minima = 0
# optimization use_response_surface
# optimization print_level = 0
# optimization num_fmin = 0
# optimization output_id = 0
# optimization max_feval = 10000
# optimization deltax = 1.0e-6
# optimization fmin = not defined
# optimization cutoff = not defined
# optimization tolerance = not defined
  printlevel 1
# file_write matlab
# use_config_file = NONE
# use_input_pdfs
# constraint_op_and
END
END
```

Apêndice B

MASE-Driver

1. Exemplo do arquivo de configuração necessário para executar a ferramenta MASE-Driver

```
{
  "sampling_methods": [
    "MC",
    "FACT",
    "LH",
    "OA",
    "OALH",
    "MOAT",
    "SOBOL",
    "LPTAU",
    "METIS",
    "FAST",
    "BBD",
    "PBD",
    "FF4",
    "FF5",
    "CCI4",
    "CCI5",
    "CCIF",
    "CCF4",
    "CCF5",
    "CCFF",
    "CCC4",
    "CCC5",
    "CCCF",
    "SFAST",
    "UMETIS",
    "GMOAT",
    "GMETIS",
    "SPARSEGRID",
    "LSA",
```

```

    "RFF4",
    "RFF5"
  ],
  "refinement_types": [
    "adaptive"
  ],
  "lastConfiguration": {
    "output_variables": [
      {
        "name": "time"
      },
      {
        "name": "qtyAgents"
      },
      {
        "name": "percentageAgents"
      },
      {
        "name": "figureOfMerit"
      },
      {
        "name": "producersaccuracy"
      },
      {
        "name": "usersaccuracy"
      },
      {
        "name": "wrongchange"
      },
      {
        "name": "rightchange"
      },
      {
        "name": "wrongpersistance"
      },
      {
        "name": "nullModel"
      },
      {
        "name": "simulatedNullModel"
      },
      {
        "name": "steps"
      }
    ]
  },
  "input_variables": [

```

```

    {
      "lowerBound": "1",
      "name": "transformationAgentQty",
      "upperBound": "100"
    },
    {
      "lowerBound": "10",
      "name": "transformationAgentGroupPercentage",
      "upperBound": "100"
    },
    {
      "lowerBound": "1",
      "name": "individualExploration",
      "upperBound": "500"
    },
    {
      "lowerBound": "1",
      "name": "groupExploration",
      "upperBound": "1500"
    }
  ]
},
"psuadeLocation": "/pathToPsuadeInstallFolder/psuade",
"maseLocation": "/pathToMaseFolder/MASE-murl.jar"
}

```

2. Exemplo de arquivo gerado como saída para o PSUADE

Exemplo de arquivo que o MASE-Driver gera como saída para o PSUADE (configuração definida no arquivo *psuade.in*, Apêndice A). O arquivo de saída possui um valor em cada linha, sendo que cada valor corresponde à uma variável de saída definida no arquivo de configuração de entrada para o PSUADE.

```

6490
8
23
6.921068396839571
7.09748464124261
73.57606344628695
0.6697305545149708
1.8648295522033496
24.409683834793704
0.26274513386997056
0.25079414389308674
365

```