

# Uma proposta de solução para o problema das n-rainhas usando SMA Híbridos

Vágner de Oliveira Gabriel<sup>1</sup>, Júlia de Avila dos Santos<sup>1</sup>, Raquel Machado Leite<sup>1</sup>, Cleo Zanella Billa<sup>1</sup>

<sup>1</sup>Centro de Ciências Computacionais - Universidade Federal do Rio Grande - FURG  
Av. Itália, Km 8 - Campus Carreiros CEP 96203-900 - Rio Grande/RS

{vdeoliveiragabriel, juliaavila.santos, tpraquel}@gmail.com

cleobilla@furg.br

**Abstract.** *The purpose of this work is to present a solution to the classic n-queens problem, which seeks to find a place for n queens on a chessboard, and they can not attack each other. As a solution, the present work uses a hybrid model for multiagent systems, developed in the Jadex environment. In this paper, we describe the steps carried out for the development of the research, as well as the potential of working with a hybrid model of multiagent systems. Finally, a case study is showed to demonstrate the effectiveness of using MAS to solve the problem.*

**Resumo.** *Este trabalho apresenta uma proposta para a resolução do problema das n-rainhas, o qual se caracteriza por posicionar um número n de rainhas em um tabuleiro sem que ocorram ataques entre elas. Como solução, o presente trabalho utiliza um modelo híbrido para sistemas multiagente, desenvolvido no ambiente Jadex. No trabalho são descritas as etapas decorrentes e as análises realizadas para o desenvolvimento da pesquisa, bem como o potencial de trabalho com um modelo do tipo híbrido de sistemas multiagente, além de apresentar um estudo de caso que valida a eficácia do recurso na resolução do problema.*

## 1. Introdução

O problema das n-rainhas é um problema clássico de satisfação de restrições, que consiste na combinação e disposição de  $n$  rainhas em um tabuleiro de xadrez  $n \times n$ , de modo que elas não se ataquem. No tabuleiro do jogo de xadrez, a rainha pode se movimentar e atacar nas horizontais, verticais e diagonais (principais e menores) [Hu et al. 2003]. É possível encontrar diversas soluções para o problema das n-rainhas, algumas são citadas neste trabalho.

Na inteligência artificial distribuída, a subárea de Sistemas multiagente aborda as atividades de um conjunto de agentes autônomos que trabalham de forma colaborativa para a realização de um problema [Rezende 2003]. Os Sistemas multiagente podem ser utilizados em vários campos de estudos, com potencial para serem empregados na realização de simulações computacionais de diversas inclinações.

Sistemas multiagente podem ser reativos ou deliberativos, no qual, os agentes reativos são aqueles que apresentam um modelo organizacional etológico, ou seja, sua organização é similar a dos animais. Já os agentes cognitivos possuem um modelo de

organização baseado em modelos sociológicos, semelhante às atitudes dos seres humanos [Alvares 1997]. Sistemas multiagente híbridos contém agentes com características dos modelos deliberativos e reativos [Bastos 1998].

O trabalho com sistemas multiagente híbridos se torna uma abordagem eficiente pois permite utilizar nas simulações tanto agentes reativos, quanto deliberativos em um mesmo ambiente. Dessa forma, é possível utilizar uma determinada categoria de agentes onde suas habilidades serão mais úteis para a sociedade como um todo.

Este trabalho tem como objetivo implementar uma solução para resolver o problema clássico das n-rainhas. Para o auxílio no desenvolvimento da solução será utilizado o software de desenvolvimento para sistemas multiagente Jadex, buscando a implementação de uma arquitetura híbrida para sistema multiagente, em que um agente mestre possui características deliberativas, enquanto os demais agentes apresentam características reativas.

O presente trabalho está dividido em oito seções. A primeira é a introdução, a segunda apresenta uma descrição sobre Sistemas Multiagente. Na seção três é abordada a técnica utilizada para modelar o problema. Na quarta seção são apresentados alguns trabalhos relacionados. A quinta traz os materiais e métodos empregados, detalhando a ferramenta Jadex. A sexta explana o estudo de caso do trabalho, com algumas figuras demonstrativas. A sétima seção mostra os resultados alcançados. E na última seção são caracterizadas as discussões e as considerações finais.

## **2. Sistemas Multiagente**

O agente é uma entidade real ou virtual, capaz de ações independentes, atuando em um ambiente e se comunicando com outros agentes, movido por um conjunto de inclinações individuais, buscando atingir um determinado objetivo [Ferber and Gasser 1991] [Wooldridge 2002]. Pode trabalhar de forma isolada ou em comunidades produzindo sistemas multiagente [Rezende 2003].

Sistemas Multiagente (SMA) são uma subárea da inteligência artificial que estuda o comportamento de um grupo de agentes autônomos. Estes trabalham de forma colaborativa, com o objetivo de resolver problemas que são comuns ao grupo [Amandi 1997]. SMA apresentam características distintas, convivem em um mesmo ambiente, cooperando, trocando informações e buscando atingir sua finalidade [Wooldridge 2002] [Wooldridge 2009].

## **3. Técnica Abordada**

A técnica utilizada para a resolução do problema foi o uso de sistemas multiagente híbridos, onde um agente deliberativo realiza o raciocínio lógico e comunica aos agentes reativos as decisões tomadas. Os agentes reativos então executam as ações de acordo com as novas informações do agente deliberativo.

### **3.1. Agente deliberativo**

Agentes deliberativos são caracterizados por possuírem um raciocínio lógico. Estes agentes podem possuir planos, metas, memórias passadas e demais características similares a de seres humanos.

[Ferber and Gasser 1991] aponta como características essenciais de agentes deliberativos conforme citado por [Alvares 1997], as seguintes:

- A comunicação exercida pelos agentes é feita diretamente através de trocas de mensagens.
- O mecanismo de controle dos agentes é deliberativo permitindo que raciocinem e decidam sobre os objetivos, planos e ações que desejam executar.
- Agentes deliberativos mantêm uma representação explícita da sociedade e do ambiente em que está situado.
- Possuem memórias, guardando um histórico de informações passadas.
- O modelo organizacional dos agentes deliberativos é um modelo sociológico, ou seja, sua organização se assemelha com os modelos organizacionais humanos.
- Suas sociedades são compostas geralmente por poucos agentes.

### **3.2. Agentes reativos**

Agentes reativos podem perceber informações do ambiente e reagir em um curto espaço de tempo apenas tendo como base as percepções que seus sensores captam do ambiente [Bastos 1998]. Agentes reativos não guardam registro de suas atividades, não planejam ações e um de seus pontos fortes é o trabalho em grupo, onde se unem para desempenhar tarefas que seriam incapazes de realizar isoladamente [Alvares 1997].

Segundo [Alvares 1997], agentes reativos possuem as seguintes propriedades:

- O conhecimento que os agentes possuem é implícito e se manifesta apenas pelo seu comportamento.
- O comportamento dos agentes é baseado no que é percebido a cada instante no ambiente, porém sem existir uma representação explícita do ambiente.
- Os agentes não armazenam uma memória de suas ações, logo, o resultado de uma ação passada não influencia uma ação futura.
- Agentes reativos possuem uma organização etológica, sendo assim, seu modelo organizacional se assemelha ao dos animais.
- Sistemas multiagente reativos possuem um número elevado de agentes, sendo dezenas, centenas ou até mesmo milhares.

### **3.3. Sistemas Multiagente Híbridos**

Sistemas multiagente híbridos são uma comunidade de agentes deliberativos e agentes reativos, os quais se unem para otimizar a resolução de tarefas. Agentes deliberativos apresentam uma dificuldade para reagir a uma situação inesperada que exija uma resposta em um tempo rápido, já os agentes reativos tem o problema de realizar ações alternativas quando alguma nova situação surge e se difere de seus objetivos principais [Bastos 1998]. Utilizar um modelo híbrido em sistemas multiagente proporciona otimizar tarefas em uma comunidade, uma vez que, cada tipo de agente resolve a tarefa que mais se adequa às suas características.

## **4. Trabalhos Relacionados**

O problema das n-rainhas por ser muito antigo, já foi abordado por diversos pesquisadores, com as mais variáveis técnicas de solução. A seguir são apresentados alguns trabalhos que propuseram diferentes soluções para a resolução do problema.

[Oliveira and Pozo 2014] utilizou uma meta heurística metafórica das formigas artificiais que deveriam supervisionar o caminho que foi entreposto o feromônio, direcionando para a escolha da localização das peças no tabuleiro, abordando de forma detalhada o problema e a solução para n-rainhas, através de experimentos em algoritmos baseados em probabilidade, com fundamentação nos valores das matrizes, assegurando proporcionalidade e ponderação lógica.

No trabalho de [Zeni 2007], é proposta a solução através de um jogo, permitindo que a solução possa ser visualizada em todo o seu contexto, de acordo com as configurações 8x8 ou 4x4, permitindo avançar ou diminuir os graus de dificuldade. O autor explica que o jogo é somente para um competidor, sem opositores, com o desafio padrão de um quebra-cabeça, não necessitando de conhecimentos matemáticos específicos para jogar.

Em [Solnon 2002] se utiliza a otimização para solucionar o problema de restrições através de metaheurísticas para n-rainhas, satisfazendo as limitações de posições para valores a serem atribuídos às variáveis. O trabalho emprega ACO (Ant Colony Optimization), combinando técnicas de buscas locais, explorando um espaço maior de tentativas, proporcionando que as soluções sejam encontradas em menor tempo e a um custo mais baixo.

[Gambardella and Dorigo 1996] apresenta a solução algorítmica Ant-Q, que trabalham soluções de simetria e assimetria para otimização combinatória, resolvendo o problema de n-rainhas através de abordagens heurísticas baseadas em redes neurais, passando a conhecer respostas de qualidade que só poderiam ser encontradas em algoritmos especializados. Estes mesmos autores, no trabalho [Gambardella and Dorigo 1996], trazem o algoritmo ACS que também utiliza a solução de desenvolvimento combinatório, mostrando que o ACS é capaz de encontrar boa elucidação para esses casos.

Este artigo apresenta como proposta uma alternativa para a resolução do problema de n-rainhas, diferente das soluções apresentadas nos trabalhos mencionados acima. A ideia proposta aqui é a utilização de sistemas multiagente híbridos, onde um agente deliberativo - que pensa e raciocina - realiza a parte lógica e marca o local onde os agentes reativos devem ficar.

## **5. Materiais e Métodos**

### **5.1. Jadex**

Jadex é um software para o desenvolvimento de aplicações envolvendo agentes cognitivos. Nesse sistema os agentes são modelados nas linguagens XML e Java, o que difere de muitos outros softwares para desenvolvimento de agentes, pois utilizam linguagens próprias para a modelagem. O Jadex é baseado no modelo BDI, onde os agentes possuem crenças que são armazenadas em sua base, desejos que podem ser definidos como estados a serem atingidos e planos que são os procedimentos utilizados para atingir seus objetivos [de Nunes 2007].

No Jadex, o agente é uma caixa preta que recebe e envia mensagens, como mostra a figura 1. As Mensagens, eventos internos e novas metas, servem como entrada para a relação interna do agente e o mecanismo de deliberação. Tendo base nos resultados obtidos no processo de deliberação, os eventos são enviados para os planos que já estão

em execução ou para novos planos instanciados a partir da biblioteca de planos. Planos em execução tem a permissão de acessar e alterar a base de crenças, enviar mensagens para outros agentes, criar novos subconjuntos de nível superior ou de sub-eventos e causar eventos internos. Já o mecanismo de reação e deliberação normalmente é o mesmo para todos os agentes [Braubach et al. 2003].

Para a definição de um agente no Jadex, o sistema precisa ter conhecimento das propriedades que constituem o agente, crenças, objetivos e a biblioteca de planos conhecidos determinam o estado inicial do agente [Braubach et al. 2003]. O Jadex adota uma abordagem declarativa e processual para a definição dos componentes de um agente. Nele, os elementos do plano devem ser implementados como classes Java comuns que determinam uma classe de estrutura, proporcionando assim um acesso genérico às especificidades do BDI. Todos os outros tópicos são especificados usando uma linguagem XML.

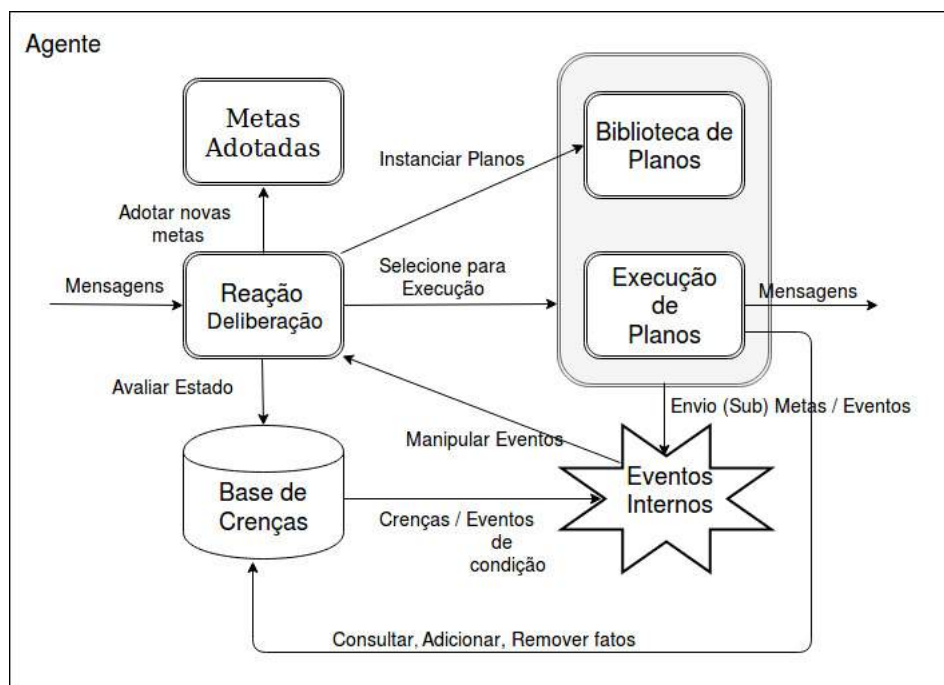


Figura 1. A arquitetura abstrata de Jadex adaptada de [Braubach et al. 2003]

## 6. Estudo de Caso

Para a realização deste estudo de caso desenvolveu-se um ambiente, figura 2, que é uma matriz de  $8 \times 8$ . No ambiente arquitetado foram criados oito agentes reativos nomeados de rainhas, cada uma das rainhas está posicionada inicialmente em uma linha da matriz. O objetivo da simulação é que cada rainha se locomova para uma posição que respeite as regras do problema das n-rainhas, ou seja, que não se ataquem entre si.

Na solução proposta existe um agente denominado mestre. Este realiza o raciocínio lógico, localiza uma solução para o problema e marca a posição onde cada rainha deve chegar. Abaixo figura 3 uma imagem demonstrativa deste procedimento.

Após o agente mestre marcar a posição que cada rainha deve estar, as rainhas

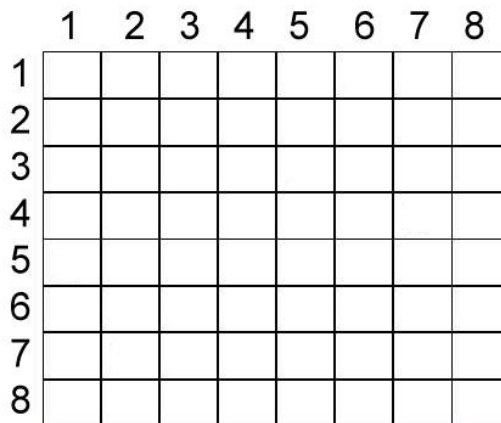


Figura 2. Ambiente de simulação

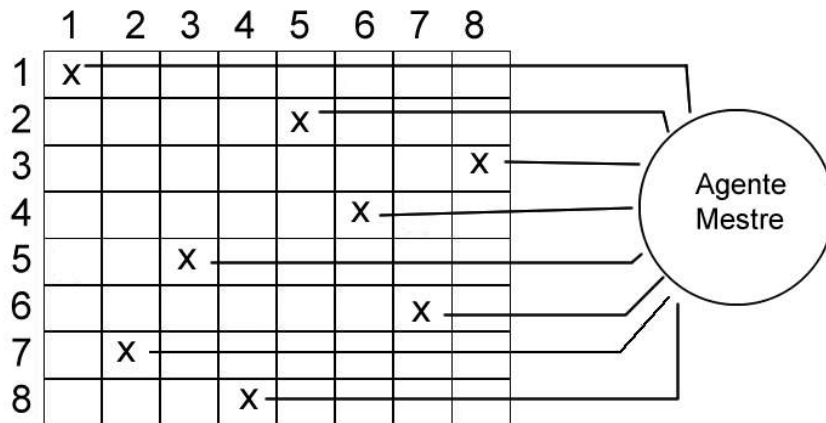


Figura 3. Procedimento de raciocínio lógico, localização e marcação de uma posição realizado pelo agente mestre

se locomovem até o lugar marcado. Ao Final das simulações as 8 rainhas devem estar posicionadas conforme mostra a figura 4.

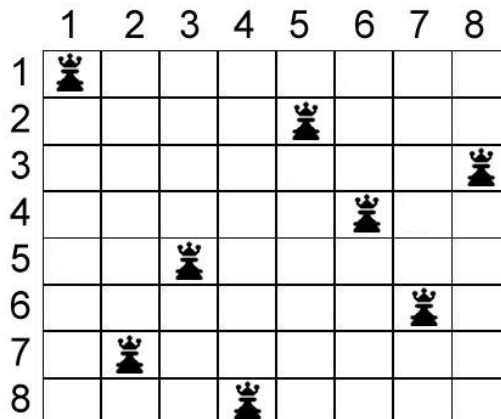


Figura 4. Posicionamento final das 8 rainhas

```

01     if (x == 1 && y == 1) {
02         return new Position((x = x + 4), (y++));
03     }else if (x == 5 && y == 2) {
04         return new Position((x = x + 3), (y++));
05     }...
06...

```

**Figura 5. Código na linguagem Java no software Jadex**

A parte lógica do agente mestre foi desenvolvida na linguagem Java. Inicialmente foram definidas as possíveis posições que cada agente poderia se alocar em uma ordem de numeração crescente. Dessa forma, a *Rainha1* se posiciona primeiro e em seguida a *Rainha2*, em uma posição que não viole a regra das *n*-rainhas. Na figura 5 é ilustrado o código que implementa o processo mencionado.

Inicialmente, a simulação contém os valores de coordenadas  $x$  e  $y$  iguais a 1, logo a primeira posição definida neste estudo de caso é  $x = 1$  e  $y = 1$ . Na figura 5, linha 01: *if(x==1 && y == 1)*, o agente mestre verifica se seu ponteiro de marcação está na linha 1 e na coluna 1. Caso seja verdade, então ele executará a ação descrita na linha 2: *(return new Position((x = x + 4), (y++)) ;)*, que é retornar a posição (5, 2)

O código da figura 5 - linha 03, executará a mesma lógica abordada anteriormente, se a marcação anterior estiver definida nas coordenadas  $x = 5$  e  $y = 2$ , a próxima marcação será  $x + 3$  e  $y + 1$ , o que acarretará na marcação da coordenada  $x = 8$  e  $y = 3$ . O processo mencionado é repetido incluindo mais regras até que as 8 rainhas estejam posicionadas corretamente no tabuleiro.

## 7. Resultados

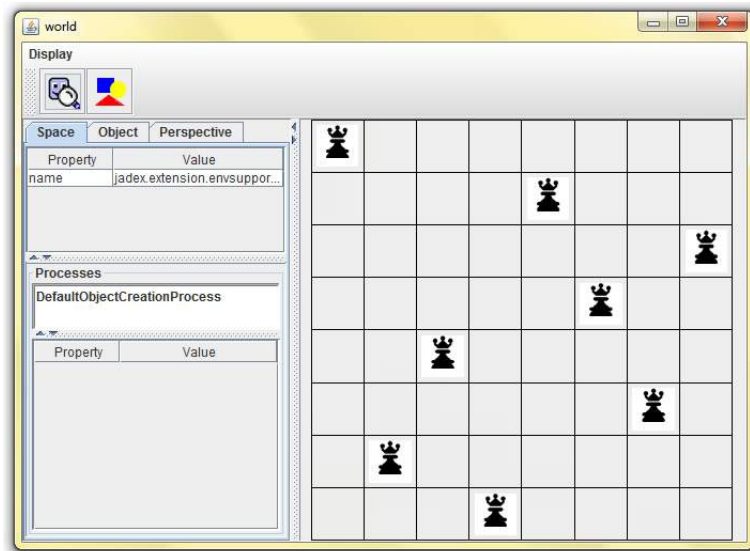
Através do uso de sistemas multiagente híbridos desenvolveu-se a resolução do problema de *n*-rainhas, no qual simulações foram realizadas para validar o desempenho da solução proposta. Ao longo do trabalho foram executadas 50 simulações, se obtendo o tempo médio de 07 segundos para resolver o problema das *n*-rainhas. O desvio padrão foi de 02 segundos.

As simulações apresentaram um posicionamento dos agentes com a combinação e disposição das 08 peças em um tabuleiro de xadrez 8x8, de modo que nenhuma rainha cometesse um ataque. A figura 6 demonstra como ficou o posicionamento final de cada rainha no ambiente de simulação desenvolvido no Jadex.

## 8. Discussão e Conclusão

Apesar do problema das 8 rainhas ser antigo e com várias soluções propostas, ainda existe um espaço que abrange novas propostas de solução para o problema. Usar sistemas multiagentes é uma alternativa válida, pois possibilita o uso de diversas técnicas para buscar uma resposta válida e ágil para o problema.

Este trabalho apresentou a proposta de solução do problema das *n*-rainhas usando sistemas multiagente do tipo híbrido, que possui um agente mestre com raciocínio lógico



**Figura 6. Posicionamento final de cada rainha no ambiente de simulação desenvolvido no software Jadex**

e 8 agentes chamados de rainhas que eram agentes reativos.

Após diversos testes foi possível constatar que a referida abordagem é válida para a solução, sendo importante enfatizar que o ambiente de desenvolvimento foi no Jadex, uma ferramenta para a criação de SMA.

Futuramente pretende-se investigar mais profundamente a parte de raciocínio dos agentes e desenvolver um algoritmo colaborativo, onde os agentes se comunicam e encontram a solução de forma rápida.

## Referências

- Alvares, L. O. (1997). Introdução aos sistemas multiagentes.
- Amandi, A. (1997). Programação de agentes orientada a objetos. Tese de doutorado. Universidade Federal do Rio Grande do Sul.
- Bastos, R. M. (1998). O planejamento de alocação de recursos baseado em sistemas multiagentes. Tese de doutorado. Universidade Federal do Rio Grande do Sul.
- Braubach, L., Lamersdorf, W., and Pokahr, A. (2003). Jadex: Implementing a bdi-infrastructure for jade agents.
- de Nunes, I. O. (2007). Implementação do modelo e da arquitetura bdi. *Monografias em Ciência da Computação*, 1.
- Ferber, J. and Gasser, L. (1991). Intelligence artificielle distribuée. France. Tutorial notes of the 11th conference on expert systems and their applications.
- Gambardella, L. M. and Dorigo, M. (1996). Solving symmetric and asymmetric tsps by ant colonies. In *International conference on evolutionary computation*, pages 622–627.



- Hu, X., Eberhart, R. C., and Shi, Y. (2003). Swarm intelligence for permutation optimization: a case study of n-queens problem. In *Swarm intelligence symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pages 243–246. IEEE.
- Oliveira, C. M. and Pozo, A. T. R. (2014). Resolução para o problema n-rainhas utilizando aco.
- Rezende, S. O. (2003). *Sistemas inteligentes: fundamentos e aplicações*.
- Solnon, C. (2002). Ants can solve constraint satisfaction problems. *IEEE transactions on evolutionary computation*, 6(4):347–357.
- Wooldridge, M. (2002). *An introduction to multiagent systems*. John Wiley & Sons.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons, 2 edition.
- Zeni, J. R. R. (2007). Um software para o problema das 8 rainhas. In *Anais do Workshop de Informática na Escola*, volume 1.