

HIPS: Um framework para escalonamento distribuído de processos em sistemas de produção utilizando sistemas multi-agentes

HIPS: A framework to distributed scheduling in production systems using multi-agent system

Arnoldo Uber Junior, M. Sc., & Ricardo Azambuja Silveira, Ph.D.

IATE: Grupo de Pesquisa em Inteligência Artificial e Tecnologia Educacional

Programa de Pós Graduação em Ciência da Computação – PPGCC

Universidade Federal de Santa Catarina – UFSC, Brasil

arnoldo.u.jr@gmail.com, silveira@inf.ufsc.br

Recibido para revisión 1 de diciembre de 2009, aceptado 4 de junio de 2010, versión final 6 de julio de 2010

Resumo: O estudo de técnicas de escalonamento de processos remete à criação dos primeiros sistemas operacionais (SO), com os algoritmos escalonadores de processos com e sem preempção. Porém a utilização de escalonadores de processo atinge outras áreas além dos SO, afeta todos os problemas onde há um conjunto de tarefas a serem executadas e um conjunto de unidades executantes. Este artigo propõe um framework denominado HIPS (Hybrid Intelligent Process Scheduler) para escalonamento distribuído de processos, utilizando a teoria de agentes e a técnica heurística de busca Algoritmos Genéticos (AG). Na modelagem do framework e aplicação foi utilizado a metodologia MaSE (Multi-agent System Engineering), que especifica etapas para análise e projeto de sistemas multiagentes. O desenvolvimento do framework e aplicação foi integrado à plataforma JADE (Java Agent Development Framework). A fim de validar o framework, desenvolveu-se um estudo de caso utilizando o framework HIPS e os resultados e limitações obtidos com esse estudo de caso, comparados a outro escalonamento de processos.

Palavras-chave: Sistemas Multiagentes(SMA), Escalonamento de Processos, Job Shop Schedule (JSS), Algoritmos Genéticos.

Abstract—The study of techniques for processes schedulers is linked to the creation of the first operational systems (OS), with the process schedulers algorithms with and without preemption. However, the use of processes schedulers reaches other areas besides OS. It affects all the problems where there is a set of tasks to be executed and a set of executing units. This work proposes a framework named HIPS (Hybrid Intelligent Process Scheduler) for distributed processes scheduler, using the agents theory and the Genetic Algorithms (GA) heuristic search technique. In the framework modeling and application the MaSE (Multi-agent System Engineering) was used, which specifies stages for analyses and project of multi agents systems. The framework development and application were integrated to

a JADE (Java Agent Development Framework). For the framework validation, the study of a case was developed using the HIPS framework and the results and limitations obtained in this study were compared to another process scheduler.

Keywords—Multi-agent Systems (MAS), Process Scheduling, Job Shop Schedule (JSS), Genetic Algorithms.

I. INTRODUÇÃO

A demanda por informação em tempo real torna-se realidade a cada instante, e os sistemas de produção necessitam de um fluxo de informação contínuo e ágil, com o objetivo de aperfeiçoar os processos de fabricação, visando atender a grande variedade de configurações que seus produtos podem compor, de forma flexível, diminuindo custos, melhorando a qualidade e entregando seus produtos em dia, na quantidade e conforme o solicitado.

O estudo de técnicas de escalonamento de processos remete à criação dos primeiros sistemas operacionais (SO) com os algoritmos escalonadores de processos sem preempção (FCFS - primeiro a chegar primeiro a executar, SJF - primeiro o mais curto, etc.), e os algoritmos com preempção (Round-Robin, Prioridade, SRTF - primeiro o processo que falta menos tempo, Filas multinível, etc.).

A utilização de escalonadores de processo atinge também outras áreas, além dos SO, e que envolvem os problemas em que há um conjunto de tarefas a serem executadas e um conjunto de unidades executantes, e que o tempo de execução final das tarefas é diretamente afetado pela seqüência de execução adotada.

Segundo [2], escalonamento, é a alocação de recursos em relação ao tempo para executar um conjunto de tarefas. Escalonadores são os dispositivos que atribuem tarefas para os seus respectivos recursos, determinando a sequência cronológica e satisfazendo as regras para atribuição.

Tais tipos de problemas são encontrados em áreas como: sequenciamento de projetos, Job-Shop Scheduling (JSS), planejamento de produção, etc. Por se tratar de um problema que tende a ser NP - Completo e possibilitar vasta aplicação, o problema JSS foi estudado neste trabalho, resultando no framework proposto neste artigo.

As ferramentas que possibilitem a utilização destas técnicas, de forma independente, ou em conjunto, formando novas tecnologias híbridas, permitem a solução satisfatória de problemas complexos através de outras perspectivas, tornando, o que antes era inimaginável, em soluções reais.

Este trabalho propõe, portanto, um framework para escalonamento de processos distribuídos, aplicado a sistemas de produção industrial, que utiliza a teoria de agentes e a técnica de busca heurística Algoritmos genéticos.

II. CARACTERIZAÇÃO DO PROBLEMA

A maioria dos problemas de programação de produção estudados aplica-se ao ambiente conhecido como Job Shop. O Job Shop tradicional é caracterizado por permitir diferentes fluxos das ordens entre as máquinas e diferentes números de operações por ordem, que são processadas apenas uma vez em cada máquina de acordo com [9].

Segundo [9] o problema JSS é um problema combinatório, o que chega a se tornar NP - completo em determinadas situações (enumeração explícita ou implícita de todas as alternativas possíveis para garantir a solução ótima), como exemplificado na Figura 1. Desta forma, algoritmos otimizantes são computacionalmente viáveis quando aplicados a problemas reais pequenos, com objetivos limitados. Para problemas de porte similar aos encontrados no ambiente real, costuma-se sacrificar a obtenção de uma solução ótima por métodos heurísticos, que resultem em uma solução subótima, com tempo computacional aceitável.

A solução proposta nesse artigo leva em consideração um problema composto de características de job shop e flow shop o que será citado neste apenas como JSS.

Um ambiente do tipo JSS é aquele em que os materiais se deslocam na fábrica com rotas dependentes do tipo de trabalho a ser executado. Já um do tipo flow shop caracteriza-se pelo fato dos materiais e peças se deslocarem na fábrica com rotas constantes. As situações reais de produção se enquadram entre esses dois tipos, ou como uma combinação de ambos [4].

Um cenário hipotético de produção foi desenvolvido para representação do problema e segue o modelo abaixo representado na Figura 1.

O modelo mostrado na Figura 1 representa o setor de tinturaria de fios [1] [15], que possui um fluxo de produção composto por sete fases distintas (numeradas de 2 a 8 no modelo), onde há processo de transformação de material em todas elas, porém não necessariamente em sequência. O modelo apresenta ainda o Planejamento, Programação e Controle da Produção (PPCP) com o número um e o estoque final de produto, com o número nove, que dependendo do setor a ser escalonado, pode ser o estoque de produto acabado ou um estoque intermediário de outro setor, dando início a um novo modelo.

Cada fase de produção possui um conjunto de processos distintos, possuindo recursos específicos para a realização daquela etapa na produção do produto. Estoque intermediário pode ocorrer no processo, em virtude dos recursos de produção não atenderem a demanda ou problemas durante a execução. Uma vez executado o processo no recurso de produção, o produto intermediário resultante alimenta outro estoque intermediário, que irá abastecer os recursos produtivos da fase seguinte, que pode ou não, estar na sequência apresentada.

Dependendo do tamanho deste estoque intermediário, o número de alternativas para escolha do processo a executar e da sequência que será executada pode se tornar uma operação que exige um tempo considerável do supervisor de produção. Esta situação é multiplicada um número considerável de vezes, se levarmos em consideração que a escolha local da fase, ocasionará uma mudança de sequenciamento em todas as fases subsequentes, podendo causar adiantamento ou atrasos nas entregas dos produtos, no final do fluxo.

O sequenciamento de uma fase ou recurso determinado pela imposição de outro, gera comumente uma perda de eficiência local, fazendo com que os objetivos sejam atingidos em detrimento da eficiência global do sistema. [10] definiu como possíveis aplicações industriais eficientes de agentes: desenvolvimento de produto, processos de planejamento e escalonamento e operações de controle de equipamentos.

Portanto há necessidade de uma ferramenta que possibilite a comunicação entre as fases de produção, buscando atingir um equilíbrio entre eficiência local da fase e a eficiência global do setor, através da comunicação, cooperação e auto-organização dos recursos, o que se pretende atingir através da modelagem e desenvolvimento da aplicação TNT (TiNTuraria) implementada utilizando o framework HIPS.

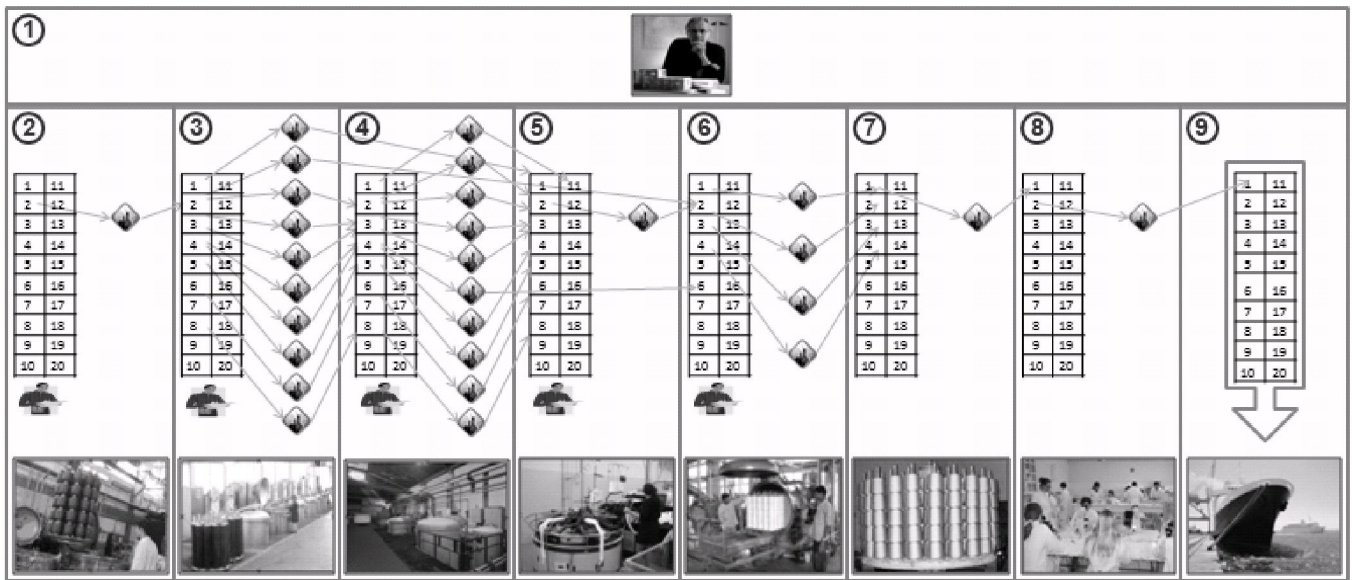


Figura 1. Cenário de Produção

III. REFERENCIAL TEÓRICO

O desenvolvimento do framework HIPS baseou-se na Teoria de Sistemas Multiagente e na técnica de busca heurística Algoritmos Genéticos.

Um agente de software é conceituado como uma entidade que funciona de forma contínua e autônoma em um ambiente em particular, geralmente habitado por outros agentes, e que seja capaz de intervir no seu ambiente, de forma flexível e inteligente, sem requerer intervenção ou orientação humana constante. De um modo ideal, um agente que funcione, continuamente, por longos períodos de tempo, pode ser capaz de aprender com a experiência e, se ele habita um ambiente com outros agentes, for capaz de comunicar-se e cooperar com eles, e ainda mover-se de um local para outro [14].

Segundo [12] agentes em um SMA podem decidir cooperativamente para atingir objetivos globais, onde geralmente atingem soluções melhores que um sistema centralizado com igual poder computacional. O uso de um SMA é justificado quando uma decisão distribuída é necessária.

A comunicação forma a base da cooperação e é constituída pelo transporte de mensagens entre os agentes através da rede, pelos protocolos de comunicação e pelos métodos de comunicação resultantes, por sua vez, os protocolos e as estratégias de cooperação, são construídos sobre os métodos de comunicação e são de fundamental importância na definição de: quem, quando e como devem atuar os agentes na solução do problema [14].

Com o objetivo de concentrar os esforços na análise do problema a ser resolvido e no planejamento de sua

implementação, abstraindo-se toda a implementação da linguagem de comunicação, protocolos e mensagens e focando no desenvolvimento da solução para o problema, foi utilizada uma plataforma para desenvolvimento de sistemas multiagentes conhecida por JADE [3].

JADE (Java Agent Development Framework) é um framework para desenvolvimento de aplicações baseadas em agentes em conformidade com as especificações FIPA (Foundation for Intelligent Physical Agents) para sistemas multiagentes [7].

Uma sucinta definição de AG seria a proposta por [8] onde descreve os AG como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural. Complementando esta definição, [8] cita que os AG são técnicas heurísticas de otimização global, que se opõe aos métodos como gradiente (hill climbing).

[11] defini os AG como sendo uma técnica na qual os estados sucessores são gerados pela combinação de dois estados pais, em vez de serem gerados pela modificação de um único estado.

Os AG são técnicas probabilísticas e não determinísticas, ou seja, com uma mesma população inicial e parâmetros, pode ser encontradas soluções distintas a cada execução.

Deve ficar claro, que o uso de AG não busca a solução ótima para o problema, mas sim uma solução satisfatória de acordo com uma função ou tempo de processamento estabelecido. Durante o processamento serão encontrados máximos e mínimos locais, próximos ou não do máximo global.

IV. FRAMEWORK HIPS

O framework proposto neste trabalho foi denominado Hybrid Intelligent Process Scheduler (HIPS) e permite modelar problemas de escalonamento de processos em cenários de produção que atendem as especificações de problemas do tipo JSS, além de executar e monitorar o sistema multiagente associado à aplicação desenvolvida a partir dele.

O framework HIPS é composto pelo ambiente de modelagem, chamado de Hips Architect e por três pacotes de classes que compõem o JHIPS.

O HIPS Architect é a ferramenta responsável pela modelagem do problema JSS, e pela criação do modelo do cenário de produção que se deseja administrar. Ele permite configurar os layouts de produção e definir os parâmetros de otimização e sua forma. Desta maneira, ele compõe o ambiente multiagente necessário para a execução do controle da produção. No Hips Architect pode, então, ser executado o cenário projetado, integrado a plataforma JADE.

A Figura 2 apresenta a interface do HIPS Architect, configurando o agente de fase.

O pacote de classes JHIPS implementa as classes base necessárias para representar o cenário de produção modelado no Hips Architect, estruturando e carregando as definições configuradas no modelo.

Os pacotes de classes JHIPS são classes Java que utilizam a framework JADE e são divididos em três grupos:

- JHIPS Ontology: são as classes que compõem a ontologia do sistema multiagente;
- JHIPS Base: são os agentes base HIPS, possuem as estruturas básicas para execução do sistema multiagente para representar um cenário de produção modelado no HIPS Architect;
- JHips Tools: são classes acessórias fornecidas pelo framework para acesso a dados e AG.

A forma de interação entre a produção, sistema ERP e o framework HIPS é representado na Figura 3.

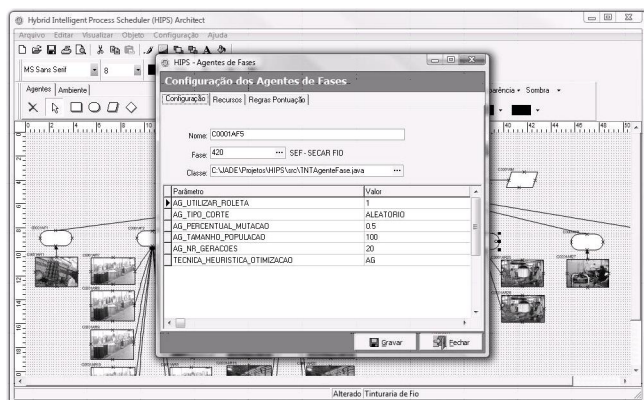


Figura 2. HIPS Architect

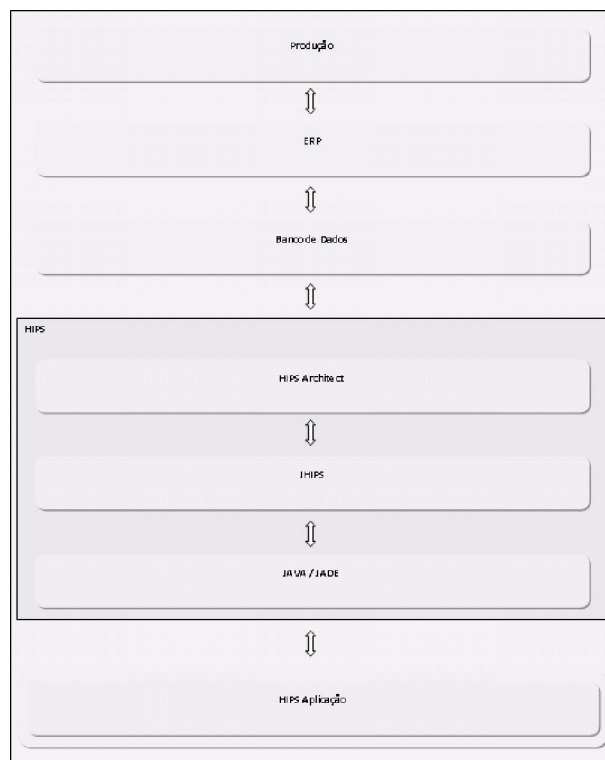


Figura 3. Interação ProduçãoxERPxHIPS

O pacote de classes JHIPS Base é o conjunto de agentes HIPS, que implementam as estruturas básicas para execução do sistema multiagente, capaz de representar um cenário de produção modelado na ferramenta HIPS Architect.

A classe de agente principal é HIPSAgent, integrada a plataforma JADE, herdando a classe `jade.core.AID`. Esta classe serve de base para os demais agentes, possuindo os métodos básicos de configuração ao iniciar `Setup()` e ao finalizar `takeDown()`.

As demais classes de agentes que herdam as características do HIPSAgent são:

- HIPSAgenteRecurso: responsáveis por gerenciar o recurso de produção;
- HIPSAgenteFase: administram a fase de produção, escalonando os processos entre os recursos disponíveis;
- HIPSAgenteMonitor: responsável por informar a situação atual do sistema multiagente, trazendo os status de cada agente iniciado e sua situação perante a produção;
- HIPSAgenteOrientador: coordena as prioridades de produção, definindo a ordem em que as tarefas serão executadas.

V. APLICAÇÃO

Com o objetivo de verificar a aplicabilidade do framework HIPS proposto, foi desenvolvida a aplicação HIPS TNT (TiNTuraria de fios).

Inicialmente foi feita a modelagem dessa aplicação através da metodologia MaSE [5], definindo os objetivos, utilizando os

agentes propostos no framework HIPS para construção dos diálogos e classes. Posteriormente o modelo construído foi sendo desenvolvido utilizando o framework HIPS, através do HIPS Architect e dos pacotes JHIPS para construção das classes de agentes.

O diagrama de papéis apresentado na Figura 4 mostra a interação entre o agente orientador, agente monitor, agentes de fases com outros agentes de fases e também, com os agentes recursos.

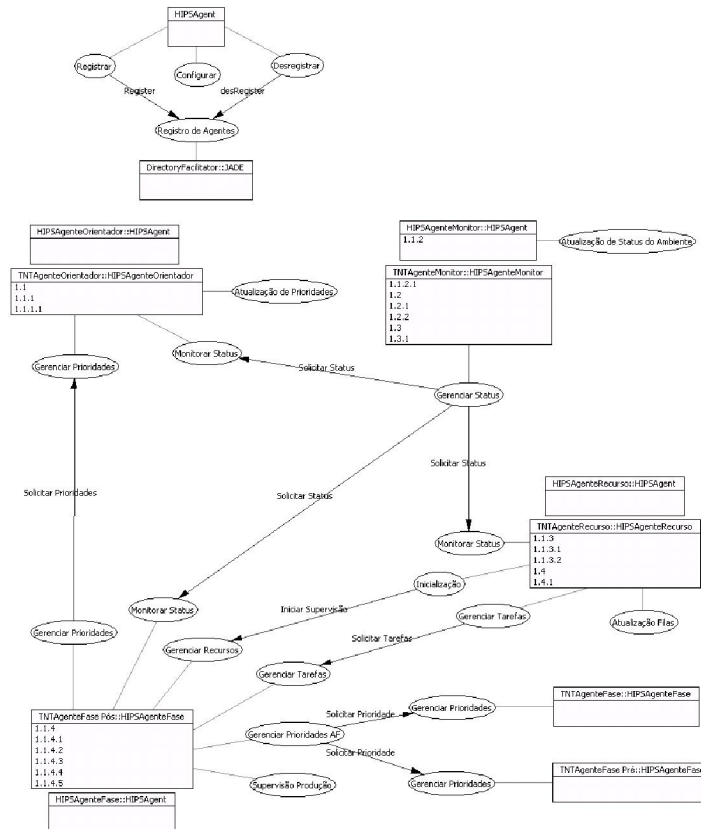


Figura 4. Diagrama de papéis HIPS TNT

O ciclo de vida definido para a aplicação HIPS TNT é iniciado com a instanciação do agente Orientador, TNTAgenteOrientador e do agente monitor, TNTAgenteMonitor. Posteriormente são iniciados os agentes de fases, TNTAgenteFase para cada uma das fases quem compõem o cenário de produção e por fim, a instanciação dos agentes recursos, TNTAgenteRecurso.

O TNTAgenteOrientador inicializa buscando as prioridades de produção, que foram definidas como sendo prioridades do agente orientador, as datas de entregas dos pedidos em produção, ou seja, tem maior prioridade os pedidos com data de entrega mais próxima, que possuam alguma ordem pendente de produção no cenário. A atualização das prioridades de produção é feita com periodicidade de um minuto, através de um comportamento próprio do agente orientador. Os pedidos e

ordens mencionados são disponibilizados através de integração, pelo sistema ERP implantado na indústria.

O agente TNTAgenteOrientador possui ainda outros dois comportamentos, que são informar o seu status ao agente TNTAgenteMonitor e retornar as prioridades de produção aos agentes de fases TNTAgenteFase.

O agente responsável por informar o usuário qual o status de cada agente no cenário de produção é o TNTAgenteMonitor. Seu comportamento é baseado na monitoração do cenário de produção, buscando novos agentes registrados na plataforma JADE, através de constantes consultas ao DF. Também é atribuição do agente monitor, solicitar aos agentes de fases, recursos e orientador os seus status atuais, apresentando ao usuário através de uma interface, as informações de cada agente registrado na plataforma.

Agentes de fase supervisionam a fase de produção definida para eles no HIPS Architect. Passam a exercer a monitoração dos recursos de produção, assim que os agentes recursos, TNTAgenteRecurso, são iniciados e buscam a sua inicialização junto ao agente de fase o qual são alocados.

O TNTAgenteFase aguarda a inicialização de seus recursos de produção. Assim que recebe a requisição de início dos agentes TNTAgenteRecursos, passam a buscar os processos a serem escalonados nos recursos disponíveis inicializados, que não fazem parte da lista fixa de processos de cada recurso, definida também através do HIPS Architect.

Carregados todos os processos a serem escalonados, o TNTAgenteFase executa o procedimento de escalonamento, que verifica os parâmetros passados através do HIPS Architect na definição da técnica de otimização. No caso da aplicação HIPS TNT, foi definido o uso de AG como técnica heurística de otimização.

A otimização através de AG é executada com periodicidade de um minuto, ou seja, executa a otimização, aguardasse um minuto e executa a otimização novamente. Cada execução atualiza a lista de processos pendentes de produção, verificado se há novos recursos, verifica a lista fixa de cada recurso, consultam as prioridades do agente orientador, TNTAgenteOrientador e dos agentes de fase vizinhos, ou seja, todos os agentes de fase, o qual as ordens a serem escalonadas possuam processos posteriores a serem escalonados. Através dessa consulta, é possível determinar as prioridades de produção do setor e também de cada fase de produção posterior, buscando otimizar o escalonamento nesse sentido. Para calculo da pontuação do seqüenciamento, ou seja, o fitness, ainda é

considerado os pesos definidos no HIPS Architect ao configurar o agente de fase, para cada uma das regras de pontuação.

Os agentes do tipo TNTAgenteFase ainda possuem os comportamentos de retornar o status ao TNTAgenteMonitor, retornar ordens de produção seqüenciadas ao agentes TNTAgenteRecurso e retornar aos demais agentes de fase a sua prioridade de produção, também calculada através do critério da data de entrega, porém difere do agente orientador, pois nesse caso analisa somente as ordens pertinentes a fase.

Os recursos de produção representados através do agente TNTAgenteRecurso inicializam buscando as informações do recurso de produção que controlam, posteriormente integrando ao sistema ERP, carregando os processos a serem executados que já estão fixos, ou seja, compõem a fila fixa de processos que não podem ser alterados através de escalonamento do TNTAgenteFase.

Após estarem iniciados e com as informações carregadas, buscam seu agente de fase, através das configurações definidas no HIPS Architect, informando o recurso de produção que estão controlando e o número de processos em sua fila fixa. Uma vez inicializados perante o seu agente de fase, passam a ter o comportamento de monitorar o recurso de produção através de integração com o sistema ERP. Notando a confirmação do término de algum dos processos, solicitam ao agente de fase a próxima ordem a ser adicionada a sua fila fixa, que trabalha da forma primeiro a entrar, primeiro a sair, ou seja, First In, First Out (FIFO).

A execução desse cenário de produção, ou seja, a aplicação HIPS TNT pode ser visualizada na Figura 5, onde foram utilizados também, alguns recursos da plataforma JADE para visualização dos agentes criados e dos diálogos sendo trocados naquele instante.

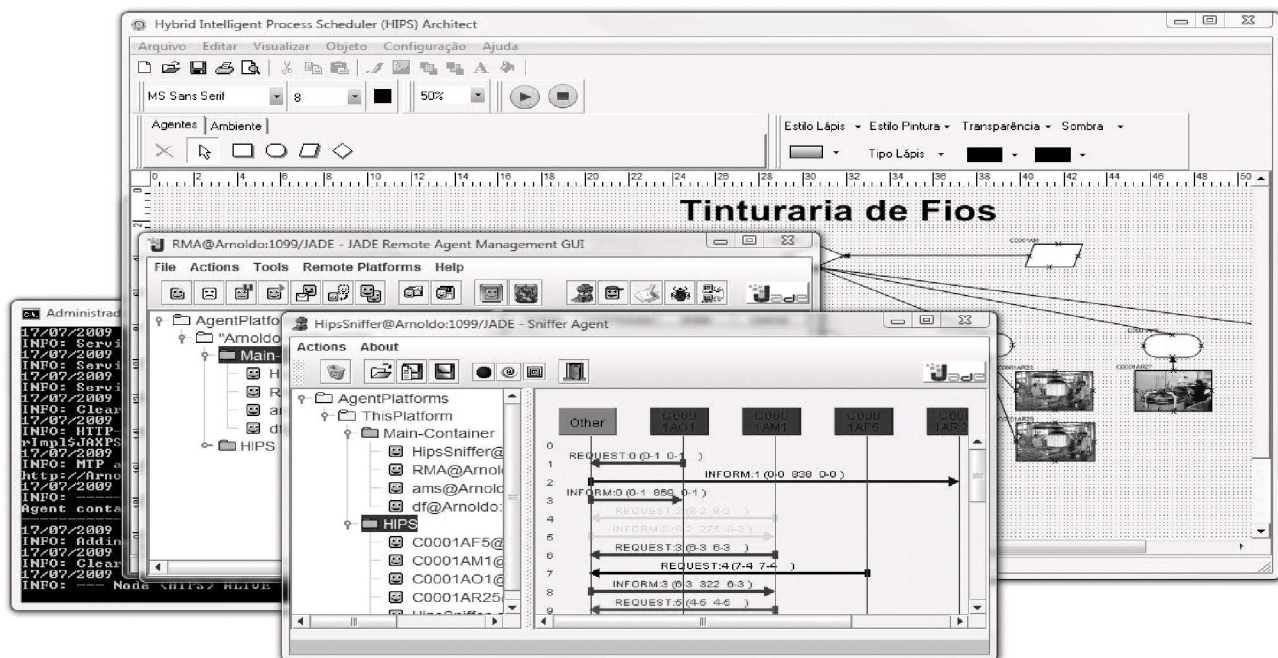


Figura 5. HIPS TNT em execução

VI. RESULTADOS OBTIDOS

Para realizar este comparativo, foram utilizadas informações reais de produção do setor de tinturaria de fios, onde constam todos os recursos produtivos deste setor e as ordens pendentes de produção, provenientes do sistema ERP.

A análise consiste em verificar o status atual do escalonamento de processos em uma das fases de produção, comparando-os a mesma fase de produção com seus recursos sendo escalonados pela aplicação HIPS TNT, durante um determinado período.

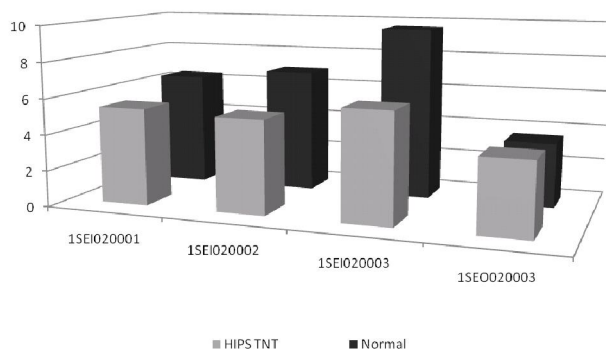
A fase escolhida para análise foi à quinta fase do cenário de produção apresentado na Figura 1, ou seja, a fase de secagem dos fios após a operação de tingimento. No cenário de produção real, esta fase possui seis recursos produtivos, porém a aplicação foi modelada com quatro recursos apenas, pois dois estavam em manutenção. O período para análise escolhido foi de uma semana, ou seja, sete dias contados a partir de uma segunda-feira.

A execução inicia com setenta e quatro ordens aguardando serem escalonadas na fase, entre seus recursos, porém três ordens em cada recurso obedecem ao escalonamento já existente, obedecendo aos parâmetros definidos no HIPS Architect. Apenas no final do dia, os agentes TNTAgenteRecursos solicitaram novas ordens ao agente TNTAgenteFase.

Quanto à alocação de tempo dos recursos, foi feito um comparativo entre os dois escalonamentos para indicar qual alocou melhor os recursos de produção disponíveis.

Determinando em tempo quase real a distribuição dos processos aos recursos de produção, o escalonamento HIPS TNT conseguiu um aproveitamento melhor de tempo, evitando paradas de processo para configuração de equipamentos. Esta diferença em relação ao escalonamento normal pode ser verificada na Figura 6, que apresenta os gráficos comparativos.

Tempo de Alocação Recursos (Dias)



Tempo de Alocação Recursos (Dias)

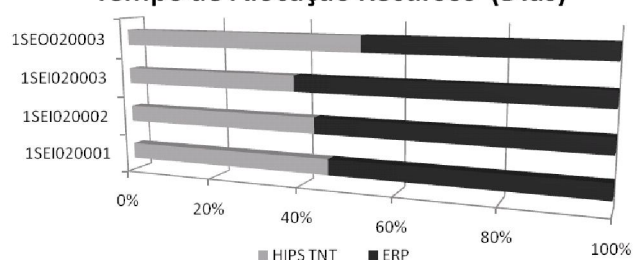


Figura 6. Gráficos do Tempo de Alocação dos Recursos

Os gráficos da Figura 8 apresentam sob duas perspectivas o tempo de alocação dos recursos de produção. Em três dos quatro recursos disponíveis, o escalonamento HIPS TNT ocupou menos tempo, onde no recurso que alocou mais tempo, esta situação é justificada pela melhor distribuição dos processos, procurando alocar todos os recursos, o menor tempo possível, conforme definição. Poderá haver situações, onde será mais vantajoso, alocar um recurso por mais tempo do que alocar todos por menos tempo, porém as regras de pontuação devem representar esta situação.

Outra análise leva em consideração o tempo total utilizado pelos escalonamentos, ou seja, a soma dos tempos de todos os recursos dos dois escalonamentos. A Figura 7 apresenta o gráfico desta comparação e destaca o excedente.

Tempo de Alocação de Recursos (Dias)

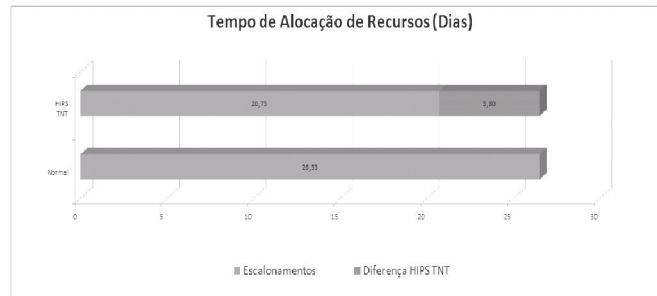


Figura 7. Gráficos do Tempo de Alocação dos Recursos Total

No gráfico da Figura 7, podem ser observadas as somatórias dos tempos de todos os recursos de cada escalonamento e em destaque, o número de dias gasto a mais pelo escalonamento normal.

Comparando diretamente o escalonamento normal e HIPS TNT, temos que o escalonamento HIPS TNT obteve uma economia de tempo de 5,8 dias, o que representa um ganho de 27,98% em relação ao escalonamento normal.

Considerando que as setenta e quatro ordens em produção, estavam movimentando 12.943,97Kg de fios, em termos práticos, utilizando o escalonamento HIPS TNT, poderiam ter sido secos mais 3621,72 Kg no mesmo período.

Através destes comparativos, foi possível verificar a quantidade de tempo utilizada a mais pelo escalonamento normal e a melhor distribuição dos processos entre os recursos disponíveis, realizado pela aplicação HIPS TNT. Benefícios que o escalonamento, quase que em tempo real, com técnicas heurísticas de otimização, pode proporcionar.

Importante ressaltar que o escalonamento normal não sofreu alterações desde o início do processo, ou seja, o planejado no início do período de análise foi considerado como válido, até o fim. Intervenções por parte da supervisão da fase, ou da gerência de produção poderiam auxiliar na obtenção de valores melhores de eficiência do escalonamento normal, porém tomadas de decisão deveriam ocorrer para essa situação e ações deveriam ser executadas no sistema ERP. Contudo esse processo não é on-line e o cenário de produção está em constante mudança.

VII. CONCLUSÃO

A utilização de sistemas multiagentes é uma estratégia adequada para manipular estes problemas, permitindo a definição de objetivos, papéis dos agentes e a forma como se relacionaram para atingir os objetivos.

O uso de uma metodologia para análise é de fundamental importância, ao modelar um problema utilizando sistemas multiagente, pois permite relacionar claramente os objetivos e a estrutura do ambiente, modelando a sociedade de agentes e guiando o pesquisador até o seu desenvolvimento.

Ao desenvolver o framework HIPS e a aplicação HIPS TNT, foi possível comprovar que a utilização de um framework para desenvolvimento dos agentes modelados, no caso, o JADE, possibilitou que os esforços fossem concentrados na análise do problema a ser resolvido e no planejamento de sua implementação, abstraindo-se toda a implementação da linguagem de comunicação, protocolos e mensagens e focando no desenvolvimento da solução para o problema.

O uso de Algoritmos Genéticos pelos agentes de fase permitiu ganhos na escolha de uma solução satisfatória local, evitando a análise de todo o conjunto de possibilidades ao seqüenciar os processos pendentes, que mudavam continuamente durante o ciclo de vida do SMA, sendo finalizados pelos agentes recursos ou sendo adicionados a fase. Outra vantagem importante dos AG é o tempo de processamento, possibilitando retornar um seqüenciamento satisfatório, não necessitando de uma grande quantidade de tempo, conforme necessidade dos agentes de fase.

A proposta do framework HIPS atingiu seus objetivos, onde ao desenvolver a aplicação HIPS TNT, foi constatado a aplicabilidade do framework, permitindo dividir o problema em partes, otimizando cada parte de maneira específica e satisfatória ao problema, resultando em benefícios decorrentes do

escalonamento de processos feito pelos agentes ao ambiente de produção.

Aprimoramentos podem ser realizados na inclusão de técnicas de negociação entre os agentes de fase e orientador para troca de prioridades, utilizando estratégias próprias de negociação ou provindas da teoria dos jogos, como os jogos de informação incompleta, tais como os leilões [6].

Poderiam ser disponibilizadas outras opções de buscas, como as citadas em [13], como: Simulated Annealing, Ant Colonies, Tabu Search e métodos híbridos também permitem avanços no sentido da qualidade de otimização e tempo de processamento.

REFERÊNCIAS

- [1] ARAÚJO, MÁRIO, E.M. DE MELO E CASTRO. Manual de Engenharia Têxtil. Vol. 1. – Lisboa: Fundação Calouste Gulbenkian, 1986.
- [2] BAKER, K. Introduction to Sequencing and Scheduling. Durman, USA, 1974. 318 P.
- [3] BELLIFEMINE, FÁBIO, CAIRE, GIOVANNI, GREWOOD, DOMINIC. Developing multiagent systems with JADE. New York: Wiley, 2007. 300 P.
- [4] BORGES, FLÁVIO H., DALCOL, PAULO R. T.. Indústria de processos: comparações e caracterizações. ENEGEP. XXII Encontro Nacional de Engenharia de Produção. 2002.
- [5] DELOACH, S. A.; WOOD, M. “Developing Multiagent Systems with agentTool”. In: Proceedings of Lecture Notes in Artificial Intelligence. Springer - Verlag. Berlim, 2001.
- [6] FIANI, RONALDO. Teoria dos jogos com aplicações em economia, administração e ciências sociais. Rio de Janeiro: Elsevier, 2006. 388 P.
- [7] JADE. Java agent development framework. Disponível em: <<http://jade.tilab.com>>. Acesso em: 21 jun. 2009.
- [8] LINDEN, Ricardo. Algoritmos Genéticos: uma importante ferramenta da Inteligência Computacional. Rio de Janeiro: Brasport, 2006. 372p.
- [9] OLIVEIRA, RONALD LOPES, WALTER, CLÁUDIO. Escalonamento de um Job-Shop: um algoritmo com regras heurísticas. UFRGS, 2000.
- [10] PARUNAK, H.V.D., SAUTER, J., CLARK, S.J.. Toward the specification and design of industrial synthetic ecosystems. Springer-Verlag, Berlim, 1997.
- [11] RUSSELL, STUART; NORVIG, PETER. Inteligência artificial: tradução 2.ed. Rio de Janeiro: Editora Campus, 2004. 1040 P.
- [12] SACILE, ROBERTO, PAOLUCCI, MASSIMO. Agent-Based Manufacturing and Control Systems. Flórida, CRC Press LLC, 2005.
- [13] PETROWSKI, J. DRÉO A., TAILLARD, P. SIARRY E.. Metaheuristics for Hard Optimization: Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies,... - Methods and Case Studies. Berlim, Alemanha, 2006.
- [14] SILVEIRA, RICARDO AZAMBUJA. Modelagem orientada a agentes aplicada a ambientes inteligentes distribuídos de ensino: JADE - Java agent framework for distance learning environments. Tese. Porto Alegre: Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, 2001. 126f.:il.
- [15] VIEIRA, OACYR FEIJÓ. Controle de qualidade na indústria de fiação e tecelagem. Vol. II. Rio de Janeiro:



Arnaldo Uber Junior é gerente de TI da Operacional Têxtil Consultoria e Sistemas. Formado em Ciências da Computação, Especialista em desenvolvimento Web e Mestre em Ciências da Computação na área de Inteligência Computacional, é pesquisador do grupo de pesquisas IATE e desenvolve pesquisas na aplicação de IA na indústria, principalmente na modelagem e desenvolvimento de sistemas híbridos utilizando a Teoria de Agentes, Teoria dos Jogos, metas-heurísticas de busca, ferramentas para auxiliar na tomada de decisão e otimização da produção.



Ricardo Azambuja Silveira é doutor em Ciência da Computação pela UFRGS e Professor adjunto do Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC). Suas áreas de Interesse são: Inteligência Artificial, Sistemas Multiagente. Linguagens Formais e Compiladores Informática na Educação, Ambientes Inteligentes de Aprendizagem.

Universidad Nacional de Colombia Sede Medellín

Facultad de Minas



Escuela de Ingeniería de Sistemas

Misión

La misión de la Escuela de Ingeniería de Sistemas es fomentar y apoyar la generación o la apropiación de conocimiento, la innovación y el desarrollo tecnológico en el área de ingeniería de sistemas e informática sobre una base científica, tecnológica, ética y humanística.



Visión

La formación integral de profesionales desde el punto de vista científico, tecnológico y social que les permita adoptar, aplicar e innovar conocimiento en el campo de los sistemas e informática en sus diferentes aspectos, aportando con su organización, estructuración, gestión, planeación, modelamiento, desarrollo, procesamiento, validación, transferencia y comunicación; para lograr un desempeño profesional, investigativo y académico que contribuya al desarrollo social, económico, científico y tecnológico del país.



Escuela de Ingeniería de Sistemas
Dirección Postal:
Carrera 80 No. 65 - 223 Bloque M8A
Facultad de Minas. Medellín - Colombia
Tel: (574) 4255350 Fax: (574) 4255365
Email: esistema@unalmed.edu.co
<http://pisis.unalmed.edu.co/>

