

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

PAULO ROBERTO FERREIRA JR.

**Estrutura Organizacional Emergente em  
Sistemas Multiagentes**

Proposta de Tese de Doutorado

Profa. Dra. Ana C. Bazzan  
Orientador

Porto Alegre, janeiro de 2006

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b>	4
<b>LISTA DE SÍMBOLOS</b>	5
<b>LISTA DE FIGURAS</b>	7
<b>LISTA DE TABELAS</b>	8
<b>RESUMO</b>	9
<b>ABSTRACT</b>	10
<b>1 INTRODUÇÃO</b>	11
<b>2 ORGANIZAÇÃO EM SISTEMAS MULTIAGENTES</b>	15
2.1 Coordenação e Organização	15
2.2 TÆMS	16
2.3 Trabalhos Relacionados	20
2.3.1 Auto-organização	20
2.3.2 Alocação Distribuída de Tarefas	20
<b>3 INTELIGÊNCIA COLETIVA</b>	25
3.1 Insetos Sociais	25
3.1.1 Características dos insetos sociais	26
3.1.2 Evolução dos insetos sociais	26
3.1.3 Organização das sociedades de insetos	27
3.1.4 Mecanismos de regulação da divisão de trabalho	28
3.1.5 Comunicação	30
3.2 Modelos Teóricos para Divisão de Trabalho	31
3.3 Otimização baseada em Colônias de Formigas	33
3.4 Trabalhos Relacionados	34
3.4.1 Divisão de Trabalho e Alocação de Tarefas	34
3.4.2 Formigas Forrageando e Otimização	35
<b>4 ABORDAGEM PROPOSTA</b>	38
4.1 Introdução	38
4.2 Idéias Básicas	38
4.3 Versão Preliminar	40
4.3.1 Estímulo	41
4.3.2 Polietismo Temporal	42

<b>4.4</b>	<b>Limitações da Versão Preliminar</b>	43
<b>4.5</b>	<b>Extensões e Modificações</b>	44
4.5.1	Estímulo	46
4.5.2	Especialização	47
4.5.3	Polimorfismo	49
<b>4.6</b>	<b>Cenários</b>	49
4.6.1	RCPSP	49
4.6.2	GAP	51
4.6.3	Considerações	52
<b>4.7</b>	<b>Discussão</b>	53
<b>5</b>	<b>RESULTADOS PRELIMINARES</b>	55
<b>5.1</b>	<b>Cenários Abstratos</b>	55
5.1.1	Cenário I	55
5.1.2	Cenário II	57
5.1.3	Cenário III	58
<b>5.2</b>	<b>Cenários Reais</b>	58
5.2.1	Cenário Simplificado	61
5.2.2	Cenário com Mudanças Dinâmicas	62
<b>6</b>	<b>CONCLUSÕES PRELIMINARES</b>	65
<b>6.1</b>	<b>Atividades Futuras</b>	65
<b>6.2</b>	<b>Cronograma</b>	68
	<b>REFERÊNCIAS</b>	69

## LISTA DE ABREVIATURAS E SIGLAS

ACO	<i>Ant Colony Optimization</i>
B&B	<i>Branch-and-Bound</i>
COP	<i>Constraint Optimization Problem</i>
CSP	<i>Constraint Satisfaction Problem</i>
DCOP	<i>Distributed Constraint Optimization Problem</i>
DiMES	<i>Distribute Multi-Event Scheduling</i>
DTC	<i>Design-to-Criteria</i>
E-GAP	<i>Extended GAP</i>
GAP	<i>Generalized Assignment Problem</i>
GPGP	<i>General Partial Global Planning</i>
JH	<i>Juvenile Hormone</i>
LA-DCOP	<i>Low-communication Aproximation</i>
NLE	<i>Non-Local Effect</i>
MSP	<i>Machine Sequencing Problem</i>
OptAPO	<i>Optimal Asynchronous Partial Overlay</i>
TG	<i>Task Group</i>
TS	<i>Ttask Structure</i>
RCPSP	<i>Resource Constraint Project Scheduling Problem</i>
SMA	<i>Sistemas Multiagentes</i>
QAF	<i>Quality Accumulation Function</i>

## LISTA DE SÍMBOLOS

$a_i$	idade do agente $i$
$a_{ij}$	unidades de capacidade do recurso $i$ consumidas por $j$
$A_i$	idade máxima estimada para o agente $i$
$\mathcal{A}$	idade do método que está a mais tempo disponível para ser executado
$c_j$	custo do método $j$
$\hat{c}_j$	custo normalizado do método $j$
$c_{ij}$	custo de alocar a atividade $i$ ao recurso $j$
$capability_i(j)$	competência do agente $i$ em realizar a tarefa $j$
$cost(\mathcal{S}_k)$	custo total do escalonamento $\mathcal{S}_k$
$d_j$	duração do método $j$
$\hat{d}_j$	duração normalizada do método $j$
$D(\mathcal{S}_k)$	desempenho dos agentes, no escalonamento $\mathcal{S}_k$
$end(j)$	unidade de tempo em que a atividade é concluída
$i$	agente, recurso (GAP/RCPSP)
$\mathcal{I}$	conjunto de recursos
$j$	tarefa, método ou atividade (RCPSP)
$\mathcal{J}$	conjunto de atividades
$k^*$	índice do melhor escalonamento em $\mathcal{S}$
$k^+$	índice da melhor alocação obtida em um escalonamento em $\mathcal{S}$
$m_j$	idade do método $j$
$M$	conjunto de métodos
$\mathcal{P}_j(\mathcal{S}_j)$	conjunto de predecessores diretos de $j$
$\mathcal{P}_j^*(\mathcal{S}^*_j)$	conjunto de todos os predecessores de $j$
$q_j$	qualidade do método $j$
$\hat{q}_j$	qualidade normalizada do método $j$
$q_{ij}$	qualidade em que o agente $i$ executa o método $j$

$\mathcal{Q}$	conjunto de recursos $r$
$QAF_{TG}(\mathcal{S}_k)$	qualidade acumulada total segundo as QAFs para o escalonamento $\mathcal{S}_k$
$r$	recurso
$s_j$	estímulo associado com a tarefa $j$
$s'_j$	estímulo local do método $j$ na rodada $t$
$s_j(t)$	estímulo do método $j$ na rodada $t$
$s_j(t - \mu)$	estímulo do método $j$ em uma rodada anterior $t - \mu$
$start_k(j)$	função que retorna a iteração (tempo) em que o método $j$ começou a ser executado no escalonamento $k$
$start_{k^*}(j)$	iteração onde o método $j$ teve sua execução iniciada no melhor escalonamento $k \in \mathcal{S}$
$\mathcal{S}$	conjunto de escalonamentos
$ \mathcal{S}_k $	duração total $ \mathcal{S}_k $ do escalonamento $\mathcal{S}_k$
$ \mathcal{S}_{k^*} $	duração total do melhor escalonamento $\mathcal{S}_{k^*}$
$T_{\theta_{ij}}(s_j)$	tendência do agente $i$ se engajar na tarefa $j$
$x_j$	estímulo associado com a QAF relacionada ao método $j$
$X(j)$	demanda associada com as QAFs relacionadas ao método $j$ , $0 \leq X(j) \leq 1$
$y_{ij}$	é o valor da $i$ -ésima linha e a $j$ -ésima coluna da matriz $y$
$Y$	matriz de alocação de tarefas a recursos
$\alpha, \beta, \gamma, \varphi$	constantes para equilíbrio dos parâmetros de $s_j$
$\delta$	taxa de desconto para diminuir o impacto do estímulo com o passar do tempo
$\theta_{ij}$	limiar interno do agente $i$ em relação a tarefa $j$
$\vartheta$	taxa de desconto para diminuir o impacto do estímulo das tarefas em $\mathcal{H}^I$
$\Gamma(j)$	viés de atualização do estímulo em $j$
$\kappa$	é a constante relacionada com o tipo de QAF
$\lambda$	taxa de desconto para o papel da competência na inicialização do limiar interno
$\mu$	constante que define o intervalo de rodadas entre as atualizações
$\xi$	coeficiente de aprendizado
$\rho$	coeficiente de esquecimento
$\varrho$	constante <i>inteira</i> ( $0 \leq \varrho \leq 1$ ) que determina se o polimorfismo será utilizado no cálculo da atualização do limiar interno
$\sigma$	constante para determinar o peso de $X(j)$ no cálculo da tendência
$\varsigma$	taxa de desconto para diminuir o impacto do estímulo das tarefas em $\mathcal{F}^j$

## LISTA DE FIGURAS

Figura 2.1:	Exemplo de uma <i>task structure</i> em TÆMS, figura originalmente publicada em (HORLING et al., 1999) . . . . .	17
Figura 4.1:	Estrutura de tarefas em TÆMS para uma instância simples do RCPSP.	51
Figura 5.1:	Estrutura de tarefas em TÆMS, figura originalmente publicada em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005). . . . .	56
Figura 5.2:	Mudança na qualidade em relação ao tempo (linha preta); As mudanças no ambiente são representadas pela sobra cinza, figura originalmente publicada em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005).	58
Figura 5.3:	Número de métodos e qualidade para um número variado de agentes, figura originalmente publicada em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005) . . . . .	59
Figura 5.4:	Estrutura de tarefas em TÆMS para o MSP (as sub-tarefas para as máquinas B, C , D com a peça 1 e todas as sub-tarefas para a peça 2 foram omitidas). . . . .	60
Figura 5.5:	Estrutura de tarefas para a peça 1 e a máquina A (todas as demais foram omitidas) com a configuração da máquinas. . . . .	61
Figura 5.6:	Melhor escalonamento possível para produzir uma peça sem tempo de configuração. . . . .	61
Figura 5.7:	Produção de cinco artefatos sem tempo de configuração. . . . .	63
Figura 5.8:	Produção de cinco artefatos com tempo de configuração. . . . .	64

## LISTA DE TABELAS

Tabela 2.1:	Funções que determinam o cálculo da qualidade da tarefa - QAF. . . .	18
Tabela 2.2:	Relacionamentos entre tarefas e/ou métodos. . . . .	19
Tabela 4.1:	Constantes relacionadas a QAF. . . . .	42
Tabela 4.2:	Duração $d_j$ e demanda de agentes $R_j$ para cada tarefa. . . . .	50
Tabela 5.1:	Escalonamento utilizando o DTC . . . . .	56
Tabela 5.2:	Melhor escalonamento (primeira variante) . . . . .	57
Tabela 5.3:	Melhor escalonamento (segunda variante) . . . . .	57
Tabela 5.4:	Sequência de Restrições das Peças . . . . .	59
Tabela 5.5:	Tempo de Processamento das Peças para cada Máquina . . . . .	60



## RESUMO

Agentes atuando em comunidade devem agir de maneira coerente para atingir um objetivo comum. A coordenação nos sistemas multiagentes previne o comportamento caótico dos agentes, permite que o sistema lide com restrições globais e a interdependência entre os agentes, e permite que o sistema seja composto por agentes com diferentes competências. A coordenação pode ser baseada na estrutura organizacional, onde a comunidade de agentes atua a favor de um objetivo comum através da forma como estão organizados. Em ambientes dinâmicos a organização dos agentes deve se adaptar a mudanças nos objetivos do sistema, na disponibilidade de recursos, nos relacionamentos entre os agentes, e assim por diante. Esta flexibilidade é um problema chave nos sistemas multiagentes e está relacionada à modelos de aprendizagem e adaptação como os observados nos insetos sociais. O presente trabalho propõe uma abordagem para a geração e adaptação da organização de um sistema multiagentes, em tempo de execução, utilizando como base os modelos teóricos de organização das colônias de insetos sociais. Esta abordagem enfoca a alocação e o escalonamento dinâmicos de tarefas distribuídos entre agentes com diferentes competências e em ambientes de larga escala. Dois cenários principais serão utilizados para experimentar e analisar a qualidade da abordagem proposta. Estes cenários estão baseados em problemas de pesquisa operacional denominados *Resource Constraint Project Scheduling Problem* (RCPSP) e o *General Assignment Problem* (GAP). Pretende-se que este trabalho contribua para o avanço do estado-da-arte em sistemas multiagentes uma vez que a abordagem proposta compõe uma metodologia completa para a especificação e resolução de problemas, através destes sistemas, enfocada em ambientes dinâmicos e de larga escala.

**Palavras-chave:** Organização em Sistemas Multiagentes, Comportamento Coletivo e Emergente em Agentes, Alocação de Recursos e Tarefas em Sistemas Multiagentes, Adaptação e Aprendizado.

## Emergent Organization in Multiagent Systems

### ABSTRACT

A community of individual agents must work in a coherent manner to reach some common goal. The coordination process in multiagent systems prevents chaotic behavior of the agents, makes the system able to deal with global constraints and inter-agents dependencies, and allows the system to be composed of agents with different capabilities. This process is normally based on the organizational structure, where the community of agents work towards the system goal due to its organization. However, in dynamic environments, agents must be able to adapt to the changing goals of the system, to the resources available, to their relationships to another agents, and so on. This problem is a key one in multiagent systems and relates to models of learning and adaptation, such as those observed among social insects. We propose a new approach to generate and adapt the multiagent organization, on the fly, based on the theoretical models of social insects colonies organization. This approach focus on distributed dynamic scheduling and task allocation using agents with different capabilities in large scale environments. We use two main scenarios to experiment and validate the proposed approach: the Resource Constraint Project Scheduling Problem (RCPSP) and the General Assignment Problem (GAP). We intend to contribute to advances the state-of-the-art in multiagent systems through this widely applicable methodology to describe and solve problems using multi-agents systems in dynamic and large scale environments.

**Keywords:** Organization in Multiagent Systems, Collective and Emergent Agent Behavior, Task and Resource Allocation in Agent Systems, Adaptation and Learning.

# 1 INTRODUÇÃO

O paradigma baseado nos sistemas multiagentes tem a coordenação como ponto central. A coordenação é o processo no qual os agentes se engajam para garantir que o conjunto deles que compõe o sistema irá atuar de uma maneira coerente. Quando agentes distribuídos trabalham para cumprir determinado objetivo, eles devem agir como uma unidade, coordenando suas ações, minimizando esforços redundantes, compartilhando recursos, etc.

A principal questão abordada por este trabalho é:

**Como coordenar um grande número de agentes em um sistema multiagentes para atuar em ambientes dinâmicos e de larga escala na busca de um objetivo comum?**

Assume-se que:

- Os agentes são cooperativos: todos desejam realizar o maior número possível de tarefas com a máxima eficiência para que os objetivos do sistema sejam alcançados;
- Os agentes podem ter competências diferentes para lidar com tarefas diferentes, compondo um sistema heterogêneo;
- A coordenação deve se dar de forma distribuída, utilizando um canal de comunicação entre os agentes;
- Ambientes de larga escala envolvem centenas de agentes realizando centenas de tarefas;
- Quanto à dinamicidade do ambiente, os agentes podem ser solicitados a responder em tempo real as demandas pela execução de tarefas, ou receber estas demandas com tempo suficiente para a realização de um planejamento.

A coordenação de um sistema multiagentes, atuando em ambientes com as características descritas acima, pode ser obtida através de sua estrutura organizacional. A dimensão funcional desta organização define os objetivos do sistema, as tarefas necessárias para atingir estes objetivos, como estas tarefas se dividem em sub-tarefas e como estas se relacionam entre si. A dimensão estrutural desta organização define que agentes atuam em quais tarefas e em que tempo isto deve ocorrer.

Desta forma, a dimensão funcional descreve o ambiente em que o sistema está inserido e quais os passos necessários para que os objetivos deste sistema sejam atingidos. Tal

dimensão pode ser especificada utilizando uma linguagem independente de domínio como o TÆMS. A principal dificuldade com relação ao problema em questão está na definição da dimensão estrutural da organização.

Este trabalho parte do pré-suposto de que é possível conseguir:

**A emergência da dimensão estrutural da organização através da utilização de modelos de divisão de trabalho em colônias de insetos sociais.**

As colônias de insetos sociais têm uma das estratégias de sobrevivência mais bem sucedidas da natureza. Esta estratégia é centrada na organização das colônias, cujas principais características são a divisão do trabalho e a especialização. As necessidades das colônias se modificam durante o passar do tempo. Estas modificações estão associadas com a fase de desenvolvimento da colônia, a época do ano, a disponibilidade de alimentos, a pressão de predadores e as condições climáticas. Apesar dessas drásticas variações nas condições da colônia os insetos sociais têm notório sucesso ecológico.

Uma colônia de insetos sociais opera sem coordenação explícita. Um inseto pertencente à casta operária individualmente não conhece as necessidades da colônia como um todo; este tem apenas informações locais simples, capacidade de comunicação extremamente limitada e ninguém é encarregado de coordenar as ações. Através da agregação destes trabalhadores, o comportamento da colônia emerge sem qualquer tipo de planejamento ou coordenação explícita. A chave para este comportamento emergente está na plasticidade da divisão de trabalho nas colônias. Estas colônias respondem à variação nas condições do ambiente através do ajuste das taxas de trabalhadores engajados nas diversas tarefas que precisam ser executadas.

**Este trabalho propõe uma abordagem para gerar e adaptar a dimensão estrutural da organização dos sistemas multiagentes, através da alocação e re-alocação de tarefas entre os agentes, utilizando como base os modelos teóricos de divisão do trabalho dos insetos sociais.**

O problema em questão pode ser redefinido como um problema de alocação/escalonamento dinâmico distribuído de tarefas em larga escala entre agentes com diferentes competências. Por alocação entende-se o processo de atribuir agentes a tarefas em tempo real, onde não é possível despendar tempo planejando, ou não se tem informação suficiente, e os agentes devem aprender com as experiências passadas. Por escalonamento entende-se o processo de determinar que agentes realizam quais tarefas e organizar a execução das tarefas com relação ao tempo. Neste último, os agentes dispõem de tempo e informação suficientes para planejar a alocação das tarefas e sua disposição no tempo deve ser feita de maneira a minimizar o tempo total necessário para a realização das mesmas.

Dois problemas da área de pesquisa operacional modelam partes diferentes, porém complementares, do problema descrito acima. O *Generalized Assignment Problem* (GAP) e o *Resource-Constrained Project Scheduling Problem* (RCPSP). Ambos são NP-difíceis. Estes problemas não são originalmente dinâmicos nem distribuídos. O número de tarefas e agentes (ou recursos) são fixos. Contudo, podem ser facilmente estendidos para lidar com essas características.

O GAP busca maximizar o ganho ao atribuir  $n$  tarefas a  $m$  agentes, sendo que cada

tarefa é associada a um agente apenas de acordo com restrições de competência destes agentes. No RCPSP tarefas devem ser escalonadas considerando um conjunto limitado de agentes (recursos na definição original) necessários para a realização de cada tarefa. As tarefas são relacionadas por restrições de interdependência de execução, ou seja, determinadas tarefas devem ser executadas antes de outras. O objetivo é buscar um escalonamento de todas as tarefas no menor tempo possível.

Estes problemas compõem dois cenários para a experimentação da abordagem proposta neste trabalho. Além de modelar vários aspectos de aplicações reais, estes problemas foram largamente estudados e existem conjuntos de dados disponíveis que podem ser usados para comparação. Pretende-se ainda estender os dois cenários, ou a combinação deles, para lidar com outros aspectos que envolvem a relação entre tarefas que não são levados em conta nestes problemas. Finalmente, objetiva-se analisar a organização emergente obtida na aplicação da abordagem em cada um destes cenários. Serão estudadas as alocações resultantes para levantar padrões de organização e verificar qual o papel da especialização neste processo.

As contribuições esperadas com este trabalho são:

- A abordagem proposta é a única abordagem para coordenação de agentes que oferece simultaneamente os mecanismos necessários para a alocação (em tempo real) e o escalonamento (com planejamento) de tarefas de forma distribuída;
- A abordagem proposta é a única abordagem para coordenação distribuída de agentes para cenários dinâmicos e de larga escala que considera as diferentes formas como as tarefas podem se relacionar entre si;
- Pretende-se resolver de forma aproximada o GAP (e suas extensões). O GAP vêm sendo utilizado para modelar muitos problemas de alocação de tarefas, incluindo cenários de coordenação de equipes de resgate e salvamento, onde um número muito restrito de soluções distribuídas foram propostas;
- Pretende-se resolver de forma aproximada o RCPSP. Este problema modela inúmeras aplicações reais, principalmente relacionadas ao escalonamento de tarefas em manufaturas. Uma abordagem distribuída, como a que está sendo proposta, junta-se ao esforço de tantas outras soluções para tentar resolver este problema crítico no estilo moderno de produção;
- Extensões aos modelos teóricos de organização dos insetos sociais estão sendo propostas para resolver problemas de forma distribuída e que contribuem para o desenvolvimento da pesquisa na área de Inteligência Coletiva;
- Serão estudadas as organizações emergentes dos modelos teóricos de organização dos insetos sociais. O papel dos modelos de especialização na forma como as tarefas são alocadas e sua influência no desempenho das soluções ainda são desconhecidos.

Além disso, a abordagem proposta compõe uma metodologia completa para a especificação e resolução de problemas utilizando sistemas multiagentes. Pretende-se que esta seja aplicável em diferentes problemas: uma vez que a dimensão funcional da organização for projetada, a dimensão estrutural emerge e os agentes se coordenam para a cumprimento dos objetivos do sistema. Em outras palavras, uma vez que o problema seja descrito os agentes serão capazes de resolvê-lo.

Este trabalho está organizado da seguinte forma:

- Capítulo 2** Neste capítulo são abordados alguns fundamentos necessários para a compreensão deste trabalho a respeito da coordenação e a organização dos sistemas multiagentes. Além disso, serão apresentados os trabalhos relacionados a este na área de sistemas multiagentes;
- Capítulo 3** Neste capítulo serão discutidos os aspectos acerca dos insetos sociais relevantes a este trabalho, onde o principal foco é a forma como a organização emerge nas colônias a partir da interação entre os elementos que a compõe. Além disso, serão apresentados os trabalhos relacionados a este na área de inteligência coletiva;
- Capítulo 4** Neste capítulo a abordagem proposta será detalhada e os cenários onde esta abordagem será experimentada serão formalizados e discutidos;
- Capítulo 5** Neste capítulo serão apresentados os resultados preliminares obtidos com uma versão mais incipiente da abordagem proposta;
- Capítulo 6** Neste capítulo serão apresentadas as conclusões preliminares e o cronograma para a realização das próximas etapas deste trabalho.

## 2 ORGANIZAÇÃO EM SISTEMAS MULTIAGENTES

Este capítulo está organizado da seguinte forma: na seção 2.1 serão discutidos os aspectos que envolvem a coordenação de agentes, a maneira como se obtém coordenação a partir da organização e as dificuldades de se obter uma organização eficiente em ambientes dinâmicos; na seção 2.2 o modelo de organização que este trabalho utiliza será detalhado; e na seção 2.3 serão abordados os trabalhos relacionados na área de sistemas multiagentes.

### 2.1 Coordenação e Organização

Como garantir que uma comunidade de agentes individuais trabalhem juntos de uma maneira coerente para atingir um objetivo comum? O processo de coordenação responde esta questão. A coordenação impede o comportamento caótico do sistema, permite que o sistema lide com restrições globais e a interdependência entre os agentes, e permite que o sistema seja composto por agentes diferentes com diferentes competências. De uma maneira mais geral, a coordenação aumenta o desempenho dos sistemas multiagentes.

Existem muitas maneiras de coordenar um sistema multiagente (SMA), classificadas em quatro categorias por (NWANA; LEE; JENNINGS, 1996): estrutura organizacional, contratos, planejamento multiagente e negociação. No artigo citado acima os autores provêm uma visão geral sobre cada categoria, criticando e comparando suas principais características. Este trabalho está interessado na coordenação baseada na estrutura organizacional, onde a comunidade de agentes age como uma unidade. Os agentes atuam a favor de um objetivo comum através de forma como estão organizados.

Existem muitos trabalhos na literatura que apresentam diferentes abordagens para a definição da organização de um SMA. (LEMAITRE-LEÓN; EXCELENTE-TOLEDO, 1998) classificaram estas abordagens em dois grupos: centrada nos agentes e centrada na organização. Nas abordagens centradas em agentes, o SMA não tem uma representação explícita da organização. Cada agente constrói sua própria representação de acordo com sua visão do comportamento dos outros agentes. Nas abordagens centradas na organização, esta é explicitamente definida, um modelo de organização deve ser usado para representar os aspectos envolvidos na interação entre os agentes.

(KIRN ST.; GASSER, 1998) formaliza o processo de modelar organizações em sistemas multiagentes, discutindo as atividades e elementos relacionados com o projeto formal da estrutura organizacional. Muitos modelos foram propostos na literatura. (HÜBNER; SICHMAN; BOISSIER, 2002) propõe separar os modelos em duas dimensões: funcional e estrutural. Na dimensão funcional o modelo indica como quais as atividades devem ser realizadas para atingir o objetivo do sistema. Na dimensão estrutural o modelo indica quais agentes estarão envolvidos com cada atividade que compõe a solução do problema.

Na abordagem proposta neste trabalho será utilizado o modelo TÆMS (DECKER; LESSER, 1993) como a base da organização funcional do SMA. A linguagem TÆMS é utilizada aqui para modelar a estrutura de tarefas e as atividades necessárias para atingir o objetivo do sistema. Este modelo será detalhado na seção 2.2..

Segundo (LESSER, 1998), a estrutura organizacional de um sistema multiagentes é um dos aspectos mais significativos para seu sucesso. A organização dos agentes depende das características do sistema quanto a forma de atingir seus objetivos, perceber o ambiente e determinar as atividades e interações entre os agentes. O problema está em definir que tipo de organização se adequa melhor a estas necessidades.

Uma maneira simples de resolver este problema é definir a organização de forma estática, o que significa encontrar os requisitos e projetar a organização mais apropriada. Uma vez que isto é feito *off-line*, as vantagens de uma organização bem definida se tornam desvantagens em ambientes dinâmicos ou que mudam frequentemente.

Agentes atuando em ambientes dinâmicos devem estar habilitados a lidar com requisições de tarefas aparecendo a todo momento, mudanças nos recursos disponíveis, falhas não previstas, entre outros. Estas questões tornam difícil o projeto de uma organização que considere todas as futuras situações possíveis. A medida que os sistemas multiagentes estão sendo utilizados em problemas dinâmicos, estruturas organizacionais estáticas com definições rígidas se tornam ineficientes.

Desta forma, sistemas multiagentes precisam lidar com a dinâmica dos cenários como a variação no número de agentes, mudanças no ambiente e nos objetivos do sistema. A questão é como produzir uma estrutura organizacional dada uma situação em particular. (HORLING; LESSER, 2004) abordam estratégias para o projeto de organizações em sistemas multiagentes, onde vários paradigmas são considerados. Neste artigo, a adaptação da organização é tratada como um tópico importante a cerca do projeto de organizações de sistemas multiagentes que atuam em ambientes dinâmicos.

O processo de gerar, adaptar e modificar a organização dinamicamente em sistemas multiagentes é usualmente chamada de *auto-organização*. Alguns autores preferem denominar esse processo de *adaptação organizacional* (HORLING et al., 2000) ou ainda *organização auto-projetada*. (SCHILLO et al., 2002) definem que um sistema multiagentes é auto-organizado se os agentes: estão habilitados a determinar a estrutura organizacional mais apropriada para o sistema em tempo de execução; e são capazes de mudar esta estrutura conforme o ambiente muda. No presente trabalho, o termo *adaptação da organização* será utilizado para representar essa característica. Na seção 2.3.1 serão discutidas abordagens que utilizam aspectos de auto-organização para adaptar a estrutura organizacional do sistema.

Além das dificuldades impostas pelos ambientes dinâmicos discutidas acima, uma nova característica, também do ambiente, tem sido foco das atenções dos pesquisadores. Cada vez mais os sistemas multiagentes vêm sendo aplicados em ambientes de larga escala, onde centenas de agentes devem realizar centenas de tarefas. Quando os ambientes dinâmicos tornam-se de larga escala os problemas relacionados a estes crescem exponencialmente. Não se conhecem abordagens de auto-organização eficientes para este tipo de ambiente.

## 2.2 TÆMS

A plataforma TÆMS (DECKER; LESSER, 1993), *Design-to-Criteria* (DTC) (WAGNER; LESSER, 2000) e *General Partial Global Planning* (GPGP) (DECKER; LESSER,



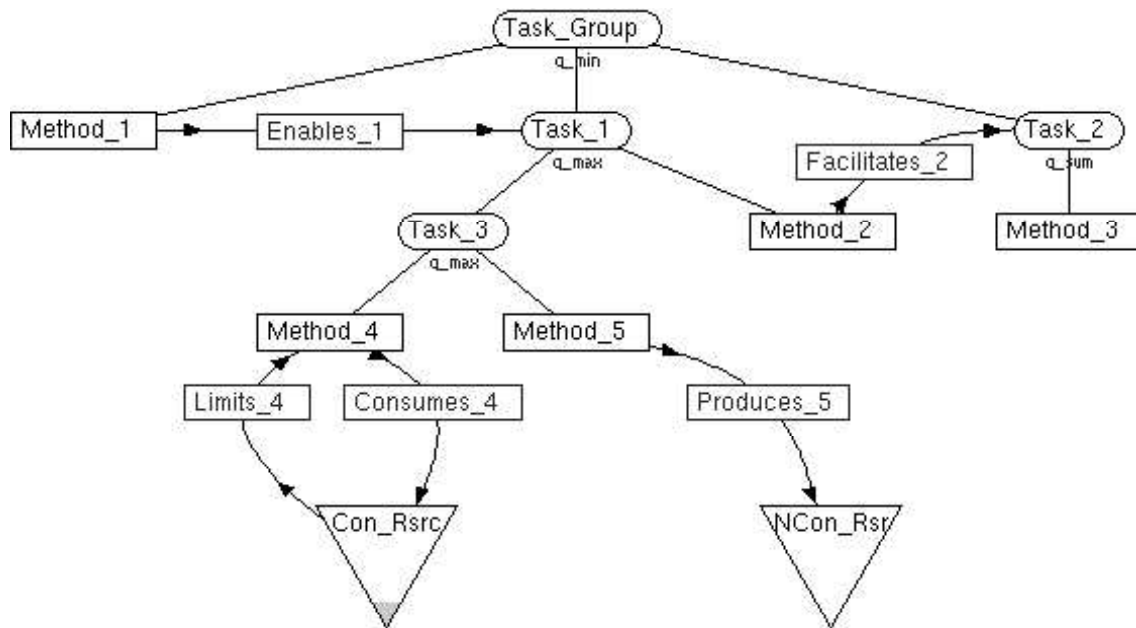


Figura 2.1: Exemplo de uma *task structure* em TÆMS, figura originalmente publicada em (HORLING et al., 1999)

1995) é utilizada como uma ferramenta independente de domínio para a descrição da estrutura de tarefas associadas à agentes e a determinação de como estes agentes serão coordenados para que tais tarefas sejam escalonadas.

O TÆMS é uma linguagem concebida para descrever a estrutura de tarefas, ou *task structure* (TS), dos agentes (DECKER; LESSER, 1993). Essencialmente, uma TS é uma forma de representação da decomposição de tarefas em uma árvore. O nível mais alto de uma árvore (raiz) é denominado TG (*task group*) e representa o objetivo do agente. Uma TS então é uma seqüência de métodos e tarefas que descrevem como aquele conjunto de tarefas será executado para que o objetivo do sistema seja atingido. Os métodos são terminais (folhas na árvore) e representam as ações primitivas que os agentes podem executar. As subdivisões de uma tarefa são denominadas *sub-tarefas* e a tarefa pais de uma tarefa na árvore é denominada *super-tarefa*.

A figura 2.1 traz uma TS com várias tarefas relacionadas a um TG, representadas por elipses. As anotações imediatamente abaixo das elipses, são intituladas *Quality Accumulation Function* - QAF, e representam a forma como a qualidade da resolução das sub-tarefas ou métodos é usada para compor a qualidade ou o custo da tarefa. Pode-se dizer que as QAFs determinam a semântica correta para as diferentes combinações e ordenações possíveis de resolução de sub-tarefa, considerando seus efeitos locais. A tabela 2.1 traz as diferentes QAFs que podem ser utilizadas em TÆMS.

Todos os métodos de uma TS em TÆMS tem uma distribuição de probabilidade associada aos possíveis resultados de sua execução. A qualidade da execução dos métodos é combinada segundo uma QAF e determina a qualidade da super-tarefa deste métodos. Este procedimento de calcular a qualidade da super-tarefa em função de suas sub-tarefas segue recursivamente até o TG (das folhas até a raiz), onde então se determina a qualidade da TS como um todo.

Os arcs entre métodos ou tarefas, na figura 2.1, indicam relacionamentos entre tarefas que não pertencem a mesma super-tarefa, onde a execução de um método terá um

Tabela 2.1: Funções que determinam o cálculo da qualidade da tarefa - QAF.

QAF	Descrição
<b>min</b>	A qualidade da tarefa é a menor qualidade das subtarefas ou métodos, e todas as subtarefas ou métodos devem ser escalonados obrigatoriamente (AND lógico entre elas).
<b>max</b>	A qualidade da tarefa é a maior qualidade das subtarefas ou métodos, e as subtarefas ou métodos podem ser escalonados em qualquer uma de suas combinações (OR lógico entre elas).
<b>sum</b>	A qualidade da tarefa é a soma da qualidade das subtarefas ou métodos, e as subtarefas ou métodos podem ser escalonados em qualquer uma de suas combinações, quanto mais métodos ou subtarefas escalonados maior será a qualidade da tarefa.
<b>all</b>	A qualidade da tarefa é a soma da qualidade das subtarefas ou métodos, e todas as subtarefas ou métodos devem ser escalonados.
<b>seq min</b>	A qualidade da tarefa é a menor qualidade das subtarefas ou métodos, e todas as subtarefas ou métodos devem ser escalonados obrigatoriamente (AND lógico entre elas) em uma determinada ordem.
<b>seq max</b>	A qualidade da tarefa é a maior qualidade das subtarefas ou métodos, e as subtarefas ou métodos podem ser escalonados em qualquer uma de suas combinações (OR lógico entre elas) que respeitem uma determinada ordem.
<b>seq sum</b>	A qualidade da tarefa é a soma da qualidade das subtarefas ou métodos, e as subtarefas ou métodos podem ser escalonados em qualquer uma de suas combinações que respeitem uma determinada ordem, quanto mais métodos ou subtarefas escalonados maior será a qualidade da tarefa.
<b>seq last</b>	A qualidade da tarefa é a qualidade da subtarefa ou método terminado por último, e as subtarefas ou métodos devem ser todos escalonados e respeitarem uma determinada ordem.
<b>exactly one</b>	A qualidade da tarefa é a maior qualidade das subtarefas ou métodos, e somente uma dessas subtarefas ou métodos podem ser escalonados (XOR lógico entre elas).
<b>last</b>	A qualidade da tarefa é a soma da qualidade das subtarefas ou métodos, e as subtarefas ou métodos podem ser escalonados em qualquer uma de suas combinações.

Tabela 2.2: Relacionamentos entre tarefas e/ou métodos.

Relacionamento	Descrição
<b>enable</b>	Relacionamento com comportamento restritivo. O método ou tarefa só poderá ser escalonado se o método ou tarefa relacionada for escalonado.
<b>disable</b>	Relacionamento com comportamento restritivo. O método ou tarefa não poderá ser escalonado se o método ou tarefa relacionada for escalonado.
<b>facilitate</b>	Relacionamento com comportamento não restritivo. O método ou tarefa será escalonado com uma qualidade maior se o método ou tarefa relacionada for escalonado.
<b>hinder</b>	Relacionamento com comportamento não restritivo. O método ou tarefa será escalonado com uma qualidade menor se o método ou tarefa relacionada for escalonado.

efeito positivo ou negativo, dependendo da qualidade e duração de sua execução, no outro método. Este relacionamento é denominado *non-local effect* (NLE). A tabela 2.2 traz os diferentes relacionamentos que podem existir entre métodos e tarefas, destacando os dois tipos em que podem se enquadrar, rígido e leve.

Além dos recursos de modelagem mostrados acima, o TÆMS permite modelar aspectos do ambiente através da modelagem de recursos, triângulo na figura 2.1. Um recurso é um conceito abstrato que possui um determinado nível e pode ser descrito como consumível ou não consumível. Relacionamentos entre método e recurso descritos como *produces* indicam que, quando o método for realizado com sucesso, uma determinada quantidade do recurso será produzida. Por outro lado, os relacionamentos intitulados *consumes* são usados para indicar que o método relacionado precisa consumir uma determinada quantidade do recurso para que possa ser realizado.

O TÆMS oferece um ferramental completo para representar os meios necessários para os agentes atingirem seus objetivos. Realizar a análise deste processo e determinar um curso apropriado para as ações dos agentes dadas restrições temporais é o papel do escalonador. O escalonador utilizado pela plataforma de desenvolvimento e simulação de agentes que incorpora o TÆMS é denominado *Design-to-Criteria* (DTC) (WAGNER; LESSER, 2000).

Em geral o número de escalonamentos possíveis para qualquer TS é significativamente grande e é intratável computacionalmente determinar cada um deles. O conjunto de restrições determinado pelas QAFs e pelas relações entre as tarefas impede a adoção de uma abordagem convencional de determinação da melhor solução. O DTC utiliza uma bateria de técnicas para lidar com estas características.

O GPGP é um mecanismo de coordenação de agentes cooperativos que permite a integração das soluções locais dos agentes, evita que tarefas sejam executadas de forma redundante e indica a compatibilidade entre os objetivos individuais.

Através destas ferramentas é possível construir a estrutura de tarefas para a resolução de qualquer problema. Esta estrutura é chamada de modelo objetivo do ambiente, e não é observado pelos agentes. Contudo, cada agente tem uma visão subjetiva desta estrutura de tarefas, que é utilizada para prever as ações dos outros agentes. A visão subjetiva contém as tarefas e relacionamento que o agente acredita ser o modelo completo de suas alternativas.

## 2.3 Trabalhos Relacionados

Nesta seção são discutidos os trabalhos relacionados a este que foram apresentados pela comunidade de sistemas multiagentes. Primeiramente, na seção 2.3.1, serão discutidas algumas abordagens que utilizam a adaptação para obter uma organização mais eficiente quando o ambiente é dinâmico. Em seguida, na seção 2.3.2, serão apresentadas as abordagens mais recentes para a alocação distribuída de tarefas entre agentes.

### 2.3.1 Auto-organização

Os primeiros trabalhos em auto-organização de sistemas multiagentes enfocavam o balanço de carga adaptativo. (ISHIDA; YOKOO; GASSER, 1990) apresenta uma abordagem para melhorar o desempenho de sistemas de produção. Nesta abordagem, um agente em particular (denominado *problem solver*) compartilha uma coleção de recursos com outros agentes. Requisições para resolver problemas chegam continuamente em taxas variáveis. A abordagem então faz a reorganização dos agentes para que estes possam atender a estas requisições, compondo e decompondo os agentes e re-allocando conhecimento para resolver os problemas.

(SHEHORY et al., 1998) apresenta uma abordagem abstrata seguindo as mesmas ideias. Nesta abordagem agentes sobrecarregados podem migrar, passar tarefas para outros agentes, se combinar com outros agentes e se clonar. A clonagem dos agentes aumenta o número de tarefas realizadas pelo sistema, aumentando seu desempenho.

(SO; DURFEE, 1993) apresentam uma abordagem de auto-organização baseada na alocação de tarefas. Nesta abordagem um conjunto de agentes autônomos está engajado na resolução de problemas através da cooperação. Os agentes percebem sua eficiência na realização de cada tarefa de maneira local e global. Quando um limiar específico de resposta é ultrapassado inicia-se um processo de reorganização.

A estrutura organizacional nesta abordagem inclui como as tarefas são decompostas em subtarefas, como as subtarefas são alocadas entre os agentes, quantos agentes estão envolvidos com cada tarefa, etc. Esta abordagem é muito próxima da que está sendo proposta neste trabalho em relação a idéia geral. Contudo, ela apresenta muitas limitações: as tarefas são decompostas em tarefas independentes, não sendo consideradas a interdependência entre tarefas; os agentes não interagem entre si com o objetivo de se comprometer com a realização de tarefas; não existe o compartilhamento de recursos pelos agentes; etc.

Recentemente, uma abordagem que pretende ser mais genérica, baseada em TÆMS e composta com um método para auto-diagnóstico foi proposta em (HORLING; BENYO; LESSER, 2001). Esta abordagem envolve coordenação de alto nível baseada na especificação da organização, o planejamento e a alocação das tarefas entre os agentes e seu escalonamento no tempo.

Esta abordagem mostrou bons resultados, mas algumas questões sobre sua eficiência em termos mais gerais continuam em aberto: com relação à comunicação, especialmente com um grande número de agentes, quão eficiente é a organização resultante?

### 2.3.2 Alocação Distribuída de Tarefas

Recentemente, as pesquisas tem se voltado para um aspecto da organização em sistemas multiagentes: a alocação distribuída de tarefas. O *Distributed Constraint Optimization Problem* (DCOP) é um formalismo para modelar um grande conjunto de problemas que envolvem a alocação de tarefas e têm sido tema de muitos avanços nos últimos anos. Os DCOPs são uma versão distribuída do *Constraint Optimization Problem* (COP), que

por sua vez é derivado do *Constraint Satisfaction Problem* (CSP). Nos DCOPs agentes colaboram para obter soluções sob um conjunto distribuído de restrições. A seguir, tais problemas serão discutidos na ordem em que foram concebidos, ou seja, CSP, COP e DCOP.

Os CSPs consistem em problemas de atribuição de valores a variáveis, respeitando restrições de valores entre elas. Cada restrição neste caso é proposicional (falso ou verdadeiro) e o objetivo final é obter uma solução que respeite todas as restrições. Ao longo dos anos foram propostas muitas soluções para os CSPs, onde as mais recentes e aplicáveis são baseadas em técnicas de busca exaustiva como a busca em profundidade (*Depth First Search*) e em largura (*Breadth First Search*). Este tipo de representação, baseada na satisfação de restrições, não é adequada para a grande maioria dos problemas reais onde a solução pode ter graus de qualidade ou custo. Outro aspecto dos problemas reais é que estes normalmente têm muitas restrições o que torna impossível satisfazer todas elas. Para problemas deste tipo se deseja minimizar o número de restrições violadas e ainda otimizar seus valores associados.

Com esse intuito, uma classe de problemas foi derivada dos CSPs e denominada *Constraint Optimization Problem*. Neste tipo de problema, as restrições não são consideradas proposicionais e valores são associados a cada restrição de acordo com os valores que podem assumir as variáveis relacionadas por esta. Uma função objetivo global está associada ao problema e o objetivo é maximizá-la ou minimizá-la. Esta função global utiliza os valores associados a cada restrição. Assim, cada valor atribuído à variáveis relacionadas por uma restrição gera um valor diferente para a restrição, que é determinado por esta, e que é utilizado pela função objetivo global.

Abordagens utilizadas para solucionar CSPs também são utilizadas em COPs que, adicionalmente, oferecem recursos para que heurísticas de aproximação possam ser utilizadas com eficiência uma vez que se pode prever se o caminho que está sendo utilizado na busca por uma solução é ou não promissor pela aproximação da função objetivo. O método *Branch-and-Bound* (B&B) é o mais popular para a solução deste tipo de problema.

Os DCOPs diferem dos COPs pois cada variável é associada a um agente e somente este tem controle sobre seu valor. As restrições são distribuídas entre os agentes que compõe o sistema e um processo de interação entre eles deve ser utilizado para se obter uma solução.

Formalmente um DCOP consiste de  $n$  variáveis  $V = \{x_1, x_2, \dots, x_n\}$  que podem assumir valores de um domínio discreto  $D_1, D_2, \dots, D_n$  respectivamente. Cada variável é associada a um agente que tem controle sobre seu valor. O objetivo do agente é escolher um valor para sua variável que otimize uma função objetivo global. Esta função é descrita como o somatório sobre o conjunto de restrições valoradas relacionadas a pares de variáveis. Assim, para um par de variáveis  $x_i, x_j$  existe uma função de custo definida como  $f_{ij} : D_i \times D_j \rightarrow N$ .

Uma solução para COP baseada no algoritmo Branch-and-Bound (B&B) é uma das mais conhecidas soluções também para DCOP. Contudo, neste caso a comunicação entre os agentes precisa ser síncrona o que aumenta significativamente o tempo de execução do algoritmo. O principal problema dos métodos para a solução de DCOPs é que estes ou são soluções que precisam de sincronismo na comunicação, deixando o processo lento em demasia, ou não garantem a qualidade da solução, podendo não obter a solução ótima.

Um método assíncrono completo para a solução de DCOPs denominado Adopt foi proposto em (MODI et al., 2003). Com isso, o Adopt é considerado um método eficiente para a solução de DCOPs (MAHESWARAN et al., 2004), garantindo a qualidade da

solução e permitindo a comunicação assíncrona. A idéia principal do Adopt é fazer com que os agentes troquem os valores das variáveis enquanto não detectam que existe uma solução melhor que a que está sendo investigada por eles. Esta estratégia permite que o método seja assíncrono, pois os agentes podem tomar decisões apenas com informação local. Soluções parciais podem ser abandonadas e reconstruídas até que a solução ótima seja encontrada.

Uma outra abordagem completa para a solução de DCOPs, baseada na mediação cooperativa, foi proposta em (MAILLER; LESSER, 2004). A mediação cooperativa é uma técnica parcialmente centralizada de mediação que é a base para o algoritmo chamado *Optimal Asynchronous Partial Overlay* (OptAPO). O OptAPO é um algoritmo distribuído e completo para a solução de DCOPs. Os autores desta abordagem mostraram que o OptAPO tem um desempenho melhor que outros algoritmos para DCOP em problemas abstratos como o problema de coloração de grafos.

Uma questão importante sobre as abordagens para a solução de DCOPs é se estas têm desempenho suficiente para serem aplicadas em problemas reais. É importante identificar se o número e o tamanho das mensagens trocadas entre os agentes permitem a aplicação da abordagem na prática. Em abordagens distribuídas, a comunicação normalmente demanda bastante esforço e pode causar uma sobrecarga na rede. O tempo total consumido pelos agentes é aceitável para aplicações reais? Os problemas reais normalmente demandam planejamento e ação no menor tempo possível. A grande maioria das abordagens para DCOP apresentam bons resultados em cenários simples mas não se tem conhecimento sobre seu desempenho em cenários mais complexos.

(MAHESWARAN et al., 2004) mostrou como mapear problemas de escalonamento para DCOP utilizando uma plataforma re-utilizável denominada *Distribute Multi-Event Scheduling* (DiMES). Além de realizar este mapeamento, os autores testaram a eficiência do Adopt em alguns cenários reais mapeados para DCOP através do DiMES e duas heurísticas foram propostas para aumentar o desempenho do Adopt nestes cenários. Foi mostrado que o tempo de convergência do Adopt para cenários reais é cem vezes maior que o esperado, destacando a diferença considerável existente entre o desempenho do algoritmo em cenários abstratos simples em relação a cenários mais complexos e próximos da realidade. Segundo os autores, as heurísticas apresentadas reduziram o tempo de convergência aos valores esperados e habilitaram o Adopt a lidar com problemas complexos. Mesmo assim, o número de mensagens trocadas pelos agentes permaneceu bastante alto.

Uma vez que o OptAPO mostrou um desempenho melhor que o Adopt no cenário de coloração de grafos, era de se esperar um comportamento similar no cenário de escalonamento. (FERREIRA JR.; BAZZAN, 2005) discute que essa premissa não é verdadeira, apresentando resultados que mostram a ineficiência do OptAPO nos mesmo cenários utilizados em (MAHESWARAN et al., 2004).

Mais recentemente, um algoritmo para DCOP, denominado DPOP, baseado em programação dinâmica foi proposto (PETCU; FALTINGS, 2005). Neste algoritmo, conjuntos com todas as soluções parciais possíveis para todas as variáveis são incrementalmente computados pelos agentes de acordo com uma ordem pré-determinada. Os autores argumentam que o número de mensagens trocadas entre os agentes, com estes conjuntos com todas as soluções possíveis, cresce linearmente em relação ao tamanho do problema. Já o tamanho das mensagens parece crescer exponencialmente, mas uma análise mais detalhada dessa questão não é apresentada. O DPOP é muito superior ao Adopt quanto ao número de mensagens trocadas, mas estas mensagens são bem maiores que as utilizadas no Adopt. Ainda não se tem experimentos sobre o desempenho do DPOP em termos de

tempo de computação e nem tão pouco sobre o impacto do tamanhos das mensagens sobre o desempenho do algoritmo.

Como problemas complexos, se considerou até agora cenários de escalonamento que envolvem algumas dezenas de agentes e tarefas. Na seção 2.1 foi salientado que os sistemas multiagentes têm sido aplicados em cenários de larga escala. Como visto acima, os algoritmos ótimos para DCOP não são eficientes em cenários complexos. O Adopt precisou de heurísticas adicionais para lidar com este tipo de problema enquanto que o OptAPO não foi capaz. Além disso, esses algoritmos não são considerados para cenários em larga escala, pois a complexidade para se determinar a solução ótima neste tipo de cenário é muito alta. Além disso, são limitados os experimentos encontrados na literatura sobre o desempenho destes algoritmos em cenários dinâmicos (OLIVEIRA; BAZZAN; LESSER, 2005).

Neste sentido, (SCERRI et al., 2005) propõem um algoritmo para DCOP especificamente para ambientes dinâmicos e em larga escala, denominado LA-DCOP (*Low-communication Approximation* DCOP). Este algoritmo diferencia-se dos demais discutidos aqui por não ser completo, buscando uma solução aproximada para DCOP. Os autores destacam que este algoritmo superou todas as outras abordagens de alocação distribuída de tarefas usando um número menor de mensagens e obtendo um melhor desempenho em cenários de larga escala.

O cenário utilizado em (SCERRI et al., 2005) baseia-se na alocação de tarefas à agentes que tem competências diferentes para atuar em cada tarefa, conhecido como GAP. O objetivo é realizar alocações que maximizem o somatório da eficiência dos agentes em cada instante. A eficiência é a medida da competência do agente de realizar a tarefa que ele está realizando no momento. Dezenas de tarefas aparecem e desaparecem com o passar do tempo e centenas de agentes devem realizá-las.

Os resultados da abordagem que está sendo proposta poderão ser comparados com os resultados obtidos pelo LA-DCOP. Porém, o LA-DCOP apresenta várias limitações em relação a abordagem que está sendo proposta por ter sido elaborado para resolver o GAP, mais especificamente uma versão deste com simplificações e extensões chamado (*Extended GAP*, ou simplesmente E-GAP), no qual:

1. Cada agente tem uma competência diferente para cada tarefa. Esta competência pode ser um número entre 0 e 1 que identificaria a qualidade da execução da tarefa pelo agente. Busca-se alocar agentes à tarefas de maneira que esta qualidade seja a maior possível.
2. O cenário é dinâmico, ou seja, tarefas surgem e desaparecem com o passar do tempo. Contudo, o número de tarefas é mantido constante, ou seja, as tarefas desaparecem e aparecem a uma mesma taxa. Essa simplificação é adotada porque o número de tarefas tem impacto no algoritmo proposto pelo artigo (o autor diz que este número influencia num limiar associada a cada tarefa que, neste caso, pode ser mantido único e constante. É preciso analisar que efeito a alteração desse valor e manutenção de mais de um deles teria sobre o desempenho do LA-DCOP).
3. É adotada uma restrição denominada de *AND* pelo autor. Esta restrição diz que algumas tarefas podem estar relacionadas entre si de maneira que todas tenham de ser executadas simultaneamente. A abordagem proposta não lida originalmente com um das características do problema definido acima, a demanda de mais de

um agente para a realização de uma tarefa. Com isso, tarefas desse tipo são decompostas em  $n$  tarefas relacionadas pela restrição *AND*. Esta restrição também é importante a medida que cada tarefa está associada a uma competência apenas.

Neste cenário não se considera que mais de um agente seja necessário para a realização de uma tarefa e que as tarefas podem ser inter-dependentes (onde a realização de uma facilita a realização de outra, por exemplo). Além disso, o LA-DCOP trata a tomada de decisão apenas em tempo real, não considerando a possibilidade de planejar ações caso haja tempo hábil e conhecimento para isso. Essa limitações restringem a aplicação do LA-DCOP quando se considera problemas reais onde estas características podem estar presentes. No capítulo 4 o GAP e sua extensão serão formalizados e discutidos com mais detalhes pois serão utilizados como cenários para os experimentos com a abordagem que está sendo proposta.



## 3 INTELIGÊNCIA COLETIVA

Este capítulo está organizado da seguinte forma: a seção 3.1 apresenta um resumo sobre o conhecimento dos pesquisadores sobre os insetos sociais do ponto de vista biológico, mas que tem como objetivo mostrar as características destes seres que motivaram o desenvolvimento de abordagens para a computação inspiradas no seu modo de vida; a seção 3.2 aborda os modelos teóricos para a organização dos insetos sociais que são a base da abordagem proposta neste trabalho; a seção 3.6 discute como a atividade de forragear das formigas deu origem a algoritmos aplicados a problemas de otimização; e a seção 3.4 traz os trabalhos relacionados a este na área de inteligência coletiva.

### 3.1 Insetos Sociais

Os insetos sociais possuem uma das estratégias de sobrevivência mais bem sucedidas da natureza, por isso, há uma imensa quantidade e variedade destes. Segundo (WILSON, 2000), existem mais espécies de formigas em 1 quilômetro quadrado de uma floresta brasileira do que todas as espécies de primatas existentes no mundo, e uma simples colônia de formigas possui mais habitantes do que todos os elefantes e leões da África somados.

A organização social desses insetos se apresenta aos biólogos como um tópico instigante de estudo e comparação. Entretanto, a riqueza e diversidade dos insetos sociais são tão grandes que, apesar do grande estudo sobre estas criaturas, ainda há muitas espécies pouco investigadas ou completamente desconhecidas. Existem cerca de 8.000 espécies de formigas descritas atualmente, mas acredita-se que ainda haja mais 4.000 espécies desconhecidas. Destas 12.000 espécies de formigas, apenas 1% foi estudada com algum detalhamento por pesquisadores.

Mesmo assim já existiam em 1973 aproximadamente 35.000 publicações sobre formigas, 3.000 publicações sobre vespas, 12.000 sobre cupins e 50.000 sobre abelhas. Todos esses números indicam que ainda há muito que explorar sobre o assunto. Toda a pesquisa realizada até agora fez com que os entomologistas definissem algumas questões fundamentais sobre o assunto, sendo elas:

- Quais são as qualidades únicas da vida social nos insetos? Isto é, quais são as características que diferenciam os insetos sociais dos demais insetos?
- Quais são os principais motivadores da evolução social?
- Quais são os passos evolucionários que levam a formas mais elevadas de organização social?
- Como as sociedades de insetos são organizadas?

### 3.1.1 Características dos insetos sociais

Para os entomologistas, para que um inseto seja considerado social ele deve possuir três qualidades fundamentais:

- Indivíduos da mesma espécie cooperam nos cuidados com a prole;
- Divisão reprodutiva do trabalho, com indivíduos mais ou menos estéreis;
- Sobreposição de pelo menos duas gerações no trabalho da colônia.

Qualquer inseto que não possua as três características poderá ser classificado como pré-social. O termo pré-social é usado para caracterizar qualquer espécie com um comportamento social maior que apenas o sexual, mas ainda assim, menor que o dos insetos sociais.

Um grupo de insetos sociais pode ser imaginado como um super-organismo. E podem se apresentar em diversos tamanhos diferentes, por exemplo, uma típica colônia de *Tetramorium caespitum* tem cerca de 10.000 formigas, 6,5 gramas e controla 40m<sup>2</sup> de solo enquanto uma colônia de *Dorylus wilverthi* é formada por 22.000.000 de formigas, pesa mais de 20 kg e controla até 50.000 m<sup>2</sup> de extensão.

### 3.1.2 Evolução dos insetos sociais

Quase a totalidade das espécies de insetos sociais pertencem à ordem das *Himenópteras*. Isso se deve principalmente a algumas características inerentes que contribuíram muito para a vida em sociedade, como o formato da mandíbula, particularmente propício para a manipulação de objetos, ou a relação estreita entre mãe e prole. Estas e outras características biológicas são pré-requisitos para a evolução dos insetos sociais. No entanto, elas são compartilhadas por diversos grupos de artrópodes, como aranhas e besouros, mas nenhum deles atingiu um nível tão elevado de sociabilidade.

A chave para o desenvolvimento social das *Himenópteras* está no seu modo de determinação sexual conhecido como haplodiploide, onde ovos não fertilizados tipicamente se desenvolvem como machos e os fertilizados se desenvolvem como fêmeas. Poucos grupos de artrópodes além das *Himenópteras* possuem essa forma de determinação sexual.

Baseado nestes fatos, (WILSON, 2000) desenvolveu uma teoria genética para o comportamento dos insetos sociais. Trabalhando com tradicionais axiomas de populações genéticas, ele primeiro deduziu o princípio de que o comportamento altruísta de um indivíduo para com um grupo de indivíduos está diretamente ligado ao seu coeficiente de relacionamento ( $r$ ) com estes indivíduos. O coeficiente de relacionamento é equivalente à fração média de genes compartilhada com indivíduos com algum parentesco. Desta forma, o  $r$  entre irmãos é  $1/2$ , entre meio irmãos é  $1/4$ , entre primos é  $1/8$  e assim por diante. Por exemplo, se um indivíduo sacrifica sua vida ou se torna estéril por algum motivo, para que este motivo seja corrigido na evolução ele deve resultar numa taxa de reprodução de irmãos maior que o dobro, ou maior que o quádruplo para meios irmãos e assim por diante.

(HAMILTON, 1964) concluiu através de cálculos que a relação genética entre a mãe e sua prole é de  $1/2$  e a entre as irmãs é de  $3/4$ , devido ao modo de determinação sexual das *Himenópteras*. A formiga fêmea recebe todos os genes de seu pai (posto que seus pais são homo-zigoto) e metade dos genes da mãe. Portanto, é geneticamente mais vantajoso a cada formiga que sua mãe (no caso a rainha) tenha filhos do que cada uma delas individualmente.

A teoria do comportamento social baseado na evolução genética de (HAMILTON, 1964) se aplica bem para himenópteras, como formigas, abelhas e vespas. No entanto, os cupins, que pertencem ao mesmo grupo que as baratas, não possuem o modo haplodiploide de determinação sexual, mas se igualaram às himenópteras em termos de evolução social. Uma característica importante dos cupins é de serem os únicos insetos comedores de madeira a depender de uma relação de simbiose com um protozoário intestinal. Esses protozoários são passados dos velhos para os jovens através, um comportamento que exige alguma organização social. Desta forma, os cupins iniciaram sua organização social devido a necessidade de alimentação e troca de protozoários entre indivíduos e somente depois passaram a cuidar da prole cooperativamente.

### **3.1.3 Organização das sociedades de insetos**

De acordo com (WILSON, 2000), assim que uma espécie chega até o nível “social” a organização da colônia pode avançar em dois sentidos: no aumento do grau de especialização das castas operárias, de forma física e/ou temporal – o polimorfismo e o polietismo temporal, respectivamente – e no alargamento do código de comunicação por onde os membros da colônia coordenam suas atividades. Com relação ao grau de especialização das classes operárias, vale comentar que (ROBISON, 1992) acrescenta ainda uma terceira característica, as diferenças individuais na especialização das classes.

#### *3.1.3.1 Especialização das castas operárias*

Como dito no parágrafo anterior, as castas podem ser de dois tipos: físicas e temporais. As formigas fêmeas se dividem em três castas: operárias, soldados e rainha. Já os machos não se dividem em castas, mas podem, num sentido mais abstrato, serem considerados como uma quarta casta. Em poucas espécies existe uma clara definição das três castas de fêmeas em todas as espécies de formigas. Em muitos casos não existem formigas soldado, por isso, na maioria das espécies, as mudanças fisiológicas e de comportamento ocorre com a idade do indivíduo. Essas mudanças representam saltos de uma casta temporal para outra.

De acordo com (GORDON, 1996), operárias jovens primeiro desempenham tarefas dentro da toca, como armazenar alimento e cuidar da prole, enquanto operárias mais velhas desempenham tarefas externas, como forragear. As formigas que cuidam da manutenção do formigueiro estão em transição entre o interior e o exterior, porque esse trabalho é feito em parte dentro do formigueiro, quando as formigas amontoam areia proveniente de túneis escavados, e em parte no exterior, quando a areia é levada para fora da boca do formigueiro. Até as larvas podem ser consideradas uma sub-casta temporal, em muitas espécies as larvas possuem uma glândula capaz de liberar um material líquido rico em nutrientes que é importante para a alimentação da colônia.

Os cupins apresentam uma organização similar às formigas, mas as castas se apresentam em ambos os sexos. Um aspecto interessante dos cupins é que estes possuem “reprodutores suplementares” e não apenas a rainha e o rei. Os reprodutores suplementares tornam as colônias praticamente imortais, pois se o reprodutor principal é removido, os outros podem substituí-lo.

Com abelhas e vespas, a divisão de castas é definida totalmente pelo polietismo temporal. Isto ocorre, segundo (WILSON, 2000) por que abelhas e vespas tomaram um caminho evolucionário diferente, por isso não há diferenças físicas entre vespas e abelhas, exceto entre a rainha e as operárias. A principal diferença morfológica existente nessas espécies é o tamanho das operárias. Operárias grandes tendem a forragear enquanto as

pequenas tendem a trabalhar dentro da colméia. Dentro dessas espécies, a rainha pode ter um comportamento passivo (máquina de botar ovos) ou um comportamento agressivo com suas irmãs operárias.

De acordo com (ROBISON, 1992) o polietismo temporal das abelhas está relacionado com a taxa de *Juvenile Hormone* (JH) existente no sangue, portanto, este hormônio está envolvido na regulação do desenvolvimento comportamental de adultos trabalhadores em colméias de abelhas. Os níveis de JH no sangue tipicamente aumentam com o passar do tempo; eles são baixos em abelhas que trabalham dentro da colméia e alto em abelhas encarregadas em forragear ou engajadas em defender a colônia. Além disso, forrageadoras precoces possuem um aumento precoce nos seus níveis de JH, abelhas “enfermeiras”, possuem um nível baixo e abelhas que passaram de forrageadoras a enfermeiras apresentaram uma redução no seu JH.

(SCHULZ; HUANG; ROBINSON, 1998) relata experiências em que um grupo de forrageadoras transportado para uma colônia acabava por inibir o desenvolvimento de novas abelhas para esta tarefa, e se o contrário ocorresse (fossem eliminadas abelhas forrageadoras, simulando a atividade de predadores), abelhas novas acabavam por se desenvolver mais depressa que o normal. A natureza desses processos inibitórios, segundo os próprios autores do artigo, ainda é desconhecida. No entanto, os resultados obtidos reforçam a idéia de que o nível de JH no sangue está diretamente relacionado com a mudança de comportamento da abelha.

### 3.1.4 Mecanismos de regulação da divisão de trabalho

Como os insetos regulam a divisão de trabalho para adaptarem-se as mudanças no ambiente é uma questão que está sendo amplamente estudada. Modelos que descrevem os mecanismos que controlam o comportamento de cada indivíduo membro da colônia são os mais usualmente utilizados. Destes modelos, destacam-se (ROBISON, 1992): modelo de divisão do trabalho baseada em diferenças de limiares de resposta, modelo de regulação hormonal na divisão do trabalho em colônias de abelhas e modelo baseado em diferenças genéticas.

#### 3.1.4.1 Modelo de divisão do trabalho baseado em diferenças de limiares de resposta

Insetos altamente sociais vivem em colônias com um grande número de indivíduos e diversos estímulos diferentes podem ser associados às diferentes tarefas que podem ser executadas. Provavelmente, as atividades que os indivíduos executam durante um determinado momento devem estar ligadas à quantidade de estímulo associado a essa tarefa específica e a sensibilidade a este estímulo deve estar associada a casta específica a que o indivíduo pertence. Diferentes respostas a estímulo foram observadas em diferentes castas físicas em formigas e cupins. Diferenças nos limiares de resposta de acordo com a idade foram observadas em abelhas, as abelhas mais velhas tendem a ter um comportamento mais defensivo do que trabalhadoras mais jovens. A probabilidade de um indivíduo executar uma tarefa está relacionada com dois fatores:

- a) magnitude do estímulo associado à tarefa;
- b) limiar de execução da tarefa pelo indivíduo.

A flexibilidade na execução de tarefas é uma consequência dos fatores ambientais atuando no indivíduo. Os limiares de resposta a uma determinada tarefa podem mudar de acordo com o tempo que o indivíduo executa a tarefa ou de acordo com as mudanças no

ambiente. Em abelhas, o JH tem sido considerado um dos fatores para as mudanças nos limiares de execução de tarefas específicas.

#### 3.1.4.2 *Regulação hormonal na divisão do trabalho em colônias de abelhas*

O JH é sintetizado e liberado pela glândula denominada *Corpora allata* e pode ser considerado um dos fatores principais que determinam o comportamento dos insetos. O JH III é o homólogo ao JH encontrado em abelhas operárias e a sua titulação aumenta na medida que a abelha envelhece.

- Controle hormonal do polietismo temporal. O JH está ligado ao polietismo temporal em abelhas, já que o sua baixa titulação está associada a tarefas dentro da colméia e quanto maior a sua titulação maior chance do indivíduo executar tarefas fora da colméia. Forrageamento precoce foi induzido em abelhas administrando-se oralmente esse hormônio e hormônios similares ao JH. A causa aparente da troca de atividade é a degeneração das glândulas hipofaríngeas, que estão naturalmente associadas as mudanças no desenvolvimento comportamental da CA.
- Controle hormonal da plasticidade no polietismo temporal. Os estímulos ambientais sobre a colônia estão associados as mudanças nas densidades demográficas de indivíduos da mesma atividade dentro da colônia e modulam o aumento normal do JH, isso resulta na plasticidade do polietismo temporal. Estudos mostram que se há falta de forrageadoras, a titulação do JH aumenta e leva indivíduos mais jovens a executarem tarefas externas, como forragear, que estão normalmente associadas a indivíduos mais velhos. Similarmente, a falta de indivíduos jovens para a execução de tarefas internas faz com que o JH aumente de modo mais lento, deixando os indivíduos jovens durante um tempo mais longo que o normal em suas tarefas internas. A modulação que o ambiente provoca nas titulações do JH fornece o mecanismo de regulação de recursos de trabalho dentro da colônia para adaptar-se às mudanças das necessidades da colônia. Trabalhos sugerem que o JH pode ter papel similar em outras espécies de insetos sociais, principalmente onde há forte polietismo temporal.

#### 3.1.4.3 *Modelo baseado em diferenças genéticas*

Para explicar a evolução comportamental e física entre operárias não reprodutivas, o modelo de seleção a nível de colônia necessita que as seguintes características estejam presentes:

- a) variação no comportamento da operária dentro da colônia resultante do seu próprio genótipo e/ou do genótipo da rainha;
- b) variação no sucesso reprodutivo entre colônias devido a diferentes distribuições das características herdadas pelas operárias;
- c) mudanças na distribuição das características herdadas pelas operárias como uma consequência da seleção a nível de colônia.

Embora as diferenças comportamentais entre os indivíduos de uma mesma colônia sejam na maioria das vezes vistas como consequências dos estímulos hormonais e do ambiente, diversos estudos sugerem que o comportamento também está relacionado as

variabilidades genéticas entre os indivíduos. Cada colônia de abelhas tem diversas sub-famílias, já que a mesma rainha tem seus ovos fertilizados por zangões diferentes.

Experimentos mostram que cada sub-família de chamadas “super-irmãs” apresentam tendências iguais a execução das mesmas tarefas e que indivíduos com idade similar mas de sub-famílias diferentes desempenham tarefas diferentes de acordo com o seu genótipo. Estudos genéticos foram utilizados para encontrar diferentes formas de divisão do trabalho, até então desconhecidas. O estudo com abelhas e formigas mostram as diferenças nas tendências em forragear, patrulhar e na manutenção do ninho em indivíduos com a mesma idade, mas como traços genéticos diferentes. A variabilidade genética pode ser um dos fatores chave para a divisão do trabalho não apenas em abelhas, mas como em outros insetos sociais, como foi observado nas formigas.

A variabilidade genética também está relacionada as diferenças nos limiares de execução de certas tarefas e a sensibilidade à estímulos externos. Indivíduos altamente especializados geneticamente na execução de uma tarefa, tendem a ter limiares muito altos para a execução de outras tarefas, tornando-se praticamente fixos a alguma tarefa em particular.

### 3.1.5 Comunicação

As formas de comunicação entre os insetos sociais são variadas, inclui pequenas pancadas, sons, toques de antenas, liberação de substâncias químicas, entre muitos outros. Devido a grande variedade de formas de comunicação torna-se muito complicado estudar todos os detalhes das interações entre os indivíduos de uma colônia, por isso, a comunicação dos insetos sociais respeita três generalizações:

- A maior parte da comunicação é química, através de feromônios. O conhecimento de sinais visuais e esparso é muito simples. Em alguns grupos, particularmente os cupins e formigas subterrâneas, esta forma de comunicação não possui nenhum papel no dia-a-dia da colônia. Os sons também são pouco identificáveis pelos insetos sociais e não são definidos em nenhum sistema de comunicação importante. Os toques são amplamente utilizados nas colônias de insetos, mas também não foi identificada nenhuma espécie de código capaz de carregar grandes quantidades de informação. Por outro lado, os sinais químicos se apresentam em praticamente todas as categorias de comunicação, principalmente através de feromônios, capazes de estimular ou inibir diversas características fisiológicas nos membros de uma colônia.
- Qualquer forma de comunicação sempre encontra paralelo em espécies pré-sociais. Por exemplo, as hierarquias de dominância que possuem papel chave nas sociedades de abelhas e vespas possuem um precedente no comportamento territorial de muitas espécies solitárias e substâncias de alarme são em muitos casos simples modificações de secreções defensivas.
- O comportamento complexo da colônia emerge da integração de simples padrões individuais através da comunicação. (WILSON, 2000) apresenta nove categorias de resposta nos insetos sociais. São elas: alarme; atração simples ou múltipla; recrutamento (para um novo ponto de coleta de comida ou um ponto específico do ninho); assistência; *trophallaxis* (troca de líquido oral e anal); troca de partículas de comida; efeitos de grupo através do estímulo ou inibição de uma atividade específica; reconhecimento de membros da mesma colônia ou membros de castas diferentes; determinação de casta, por inibição ou estimulação.

### 3.2 Modelos Teóricos para Divisão de Trabalho

Modelos teóricos e matemáticos sobre a organização das colônias de insetos sociais foram concebidos baseados nos estudos e observações realizadas pelos entomologistas. Tais modelos foram construídos em sua maioria com o objetivo de simular o funcionamento das colônias para tentar compreender sua organização. Em (GORDON, 2002), capítulo 8, são apresentados alguns destes modelos que, em sua maioria, consideram a alocação de tarefas orientada pelo padrão de iterações que cada indivíduo experimenta.

Os modelos apresentados até então produzem resultados que se assemelham ao comportamento observado de colônias reais, mas não demonstram o modo como os insetos operam. Assim, estes modelos ainda não puderam ser utilizados para o objetivo a que foram concebidos, pois não se sabe se os detalhes empíricos modelados são suficientes.

Um dos modelos apresentados mais recentemente em (BONABEAU; THRAULAZ; DORIGO, 1999)<sup>1</sup>, capítulo 3, já por pesquisadores da computação, pretende ser um modelo genérico que cobre todos os aspectos que envolvem a divisão do trabalho nas colônias. Neste modelo cada indivíduo da colônia tem um limiar de resposta a estímulos para realizar determinada tarefa. Um indivíduo passa a executar uma tarefa quando o estímulo para executar esta tarefa ultrapassa seu limiar associado.

Seja  $s$  a intensidade de um estímulo associado a uma atividade em particular, onde  $s$  pode ser o número de encontros com outros indivíduos, uma concentração química ou qualquer outro fator quantitativo que possa ser sentido por um indivíduo. O limiar de resposta  $\theta$ , expresso em unidades de intensidade de estímulo, é uma variável interna que determina a tendência de um indivíduo, respondendo ao estímulo  $s$ , realizar a tarefa associada.

A equação 3.1 mostra uma possível função para a probabilidade de um indivíduo atender a resposta de um estímulo. Outras funções podem levar ao mesmo padrão de resultados esperado pelo modelo. Qualquer número inteiro  $n$  maior que 1 pode ser usado com expoente de  $s$ , que determina o índice de crescimento de  $\theta$ , porém o autor utiliza em todos seus exemplos  $n = 2$ .

$$T_{\theta}(s) = \frac{s^2}{s^2 + \theta^2} \quad (3.1)$$

Desta forma, se o valor de  $\theta$  for muito pequeno, a probabilidade do indivíduo atender ao estímulo tende a 0 e se o valor de  $\theta$  for muito grande tal probabilidade irá tender a 1. Com  $s = \theta$  a probabilidade é exatamente 1/2.

A equação 3.2 traz outro exemplo de função para a probabilidade de um indivíduo atender a resposta a um estímulo.

$$T_{\theta}(s) = 1 - e^{-\frac{s}{\theta}} \quad (3.2)$$

O comportamento das funções apresentadas acima é equivalente, variando apenas a curva de resposta aos estímulos.

Considerando o estímulo associado a tarefa de recolher um objeto como o número de encontros do indivíduo com objetos daquele tipo, como grãos que precisam ser carregados para o ninho ou cadáveres para o cemitério. Se, a cada encontro com um novo ítem, o indivíduo tiver uma probabilidade  $\rho$  de recolher aquele objeto, então a probabilidade do indivíduo não responder aos primeiros  $N$  encontros é  $(1 - \rho)^N$ . Com isso, a probabilidade

<sup>1</sup>Em (THRAULAZ; BONABEAU; DENEUBOURG, 1998) são apresentados outros modelos nesta linha e são discutidas as diferenças entre tais modelos e o modelo do autor.

$P(N)$  de um indivíduo responder a  $N$  encontros é  $P(N) = 1 - (1 - \rho)^N = 1 - e^{N \ln(1-\rho)}$ , que é a equação 3.2 com  $s = N$  e  $\theta = \frac{-1}{\ln(1-\rho)}$ .

O limiar de resposta a um estímulo, valor de  $\theta$  nas funções acima, pode variar para cada indivíduo para refletir o polimorfismo e o polietismo temporal dos insetos sociais. Indivíduos fisicamente diferentes ou de diferentes idades podem ter tendências diferentes de executar determinadas tarefas.

No modelo apresentado em (BONABEAU; THRAULAZ; DORIGO, 1999), capítulo 3, um indivíduo torna-se mais sensível ao estímulo associado a uma tarefa em que está engajado e menos sensível aos estímulos associados as demais tarefas. Cada indivíduo no modelo tem um limiar de resposta para cada tarefa. Este limiar é atualizado (aumentando ou diminuindo) de acordo com dois coeficientes diferentes. O limiar de resposta  $\theta_{ij}$  de um indivíduo  $i$  quando executando a tarefa  $j$  durante o intervalo de tempo  $\Delta t$  é dado pela equação 3.3.

$$\theta_{ij} = \theta_{ij} - \xi \Delta t_{ij} \quad (3.3)$$

onde:

$\xi$  coeficiente de aprendizado

O limiar de resposta  $\theta_{ij}$  de um indivíduo  $i$  quando não executando a tarefa  $j$  durante o intervalo de tempo  $\Delta t$  é dado pela equação 3.4.

$$\theta_{ij} = \theta_{ij} + \rho \Delta t_{ij} \quad (3.4)$$

onde

$\rho$  coeficiente de esquecimento

O estímulo  $s$  associado a uma tarefa deve variar conforme sua execução é satisfatória ou não. Uma tarefa pode ser executada de forma ineficiente, seja por um ou mais indivíduos, o que deve aumentar seu estímulo associado para fazer com que outros indivíduos se engajem na sua execução.

A variação da intensidade de um estímulo associado a execução de uma tarefa, que reduz a intensidade do estímulo, e associado ao crescimento natural da demanda é dada pela seguinte equação.

$$s(t+1) = s(t) + \delta - \frac{\alpha N_{act}}{N} \quad (3.5)$$

onde:

$s(t+1)$  estímulo associado a tarefa  $j$  no tempo  $t+1$ .

$s(t)$  estímulo associado a tarefa  $j$  no tempo  $t$ .

$\delta$  crescimento da intensidade em função do tempo.

$\alpha$  fator de escala associado a eficiência da realização da tarefa.

$N_{act}$  número de indivíduos ativos na colônia.

$N$  número de indivíduos que podem ser tornar ativos na colônia.



Todo indivíduo tem a mesma probabilidade fixa de deixar de executar uma tarefa e pode retomar sua execução imediatamente se o estímulo que este indivíduo tem para executar a tarefa que acabou de abandonar for superior ao seu limiar.

Os modelos apresentados nesta seção serviram de base para a concepção de abordagens para a coordenação de agentes em alguns cenários específicos. Tais abordagens e cenários serão discutido na seção 3.4.

### 3.3 Otimização baseada em Colônias de Formigas

Outras características dos insetos sociais também já serviram de inspiração para a concepção de modelos teóricos que estão sendo aplicados em problemas computacionais. Modelos baseados na maneira como as formigas forrageiam deram origem a uma série de algoritmos para a solução de problemas de otimização combinatorial.

Atualmente há uma série de pesquisas sendo realizadas nessa linha, o que levou ao surgimento de uma nova área de pesquisa intitulada ACO (do inglês *Ant Colony Optimization*) (DORIGO, 1992). Abordagens de ACO para problemas como a escolha de rotas para veículos, ordenação sequencial, coloração de grafos, roteamento em redes de comunicação, entre outros, tem sido experimentadas com sucesso.

As abordagens de ACO, como dito acima, foram baseados na maneira como as formigas forrageiam, mais especificamente, na forma com que elas determinam o menor caminho entre a fonte de alimento e seu ninho. Enquanto se deslocam do ninho até a fonte de alimento encontrada, e de volta para o ninho, as formigas secretam uma substância química chamada feromônio. Esta substância, considerada como uma forma primitiva de comunicação, é percebida por outras formigas que escolhem seu caminho probabilisticamente de acordo com a concentração de feromônio em cada caminho possível. Uma colônia de formigas seguindo esta trilha química faz com que emergja o caminho mais curto entre o ninho e a fonte de alimento.

O modelo probabilístico que descreve este comportamento será apresentado a seguir. Neste modelo a probabilidade de escolher um caminho específico em um determinado momento é proporcional ao número de formigas que escolheu aquele caminho até tal momento. Assim, suponhamos que dois caminhos são possíveis até a fonte de alimento. O número de formigas que escolheu o primeiro é dado por  $U_m$  e o número de formigas que o escolheu o segundo é dado por  $L_m$ , após  $m$  ( $L_m + U_m$ ) formigas se deslocarem entre ninho e fonte de alimento. A probabilidade  $P_U(m)$  da  $(m + 1)$ -ésima formiga escolher o primeiro caminho é dada pela equação 3.6, onde  $k$  e  $h$  são constantes utilizadas para adequar o modelo a observações reais.

$$P_U(m) = \frac{(U_m + k)^h}{(U_m + k)^h + (L_m + k)^h} \quad (3.6)$$

A probabilidade  $P_L(m)$  que a mesma formiga escolha o segundo caminho é dada por  $P_L(m) = 1 - P_U(m)$ .

Quando um dos caminhos é mais curto a quantidade de formigas que irá selecioná-lo é maior. Isto ocorre porque as primeiras formigas a chegarem na fonte de alimento são as que escolheram o caminho mais curto e, por consequência, estas formigas voltam mais cedo para o ninho. Com isso, haverá mais feromônio depositado no caminho mais curto, fazendo com que outras formigas tentem a seguir este caminho.

Estas equações foram obtidas através de experimentos com formigas reais em cenários controlados. Utilizando o modelo descrito acima, as abordagens de ACO oferecem meta-

heurísticas onde formigas artificiais trabalham cooperativamente para encontrar boas soluções para problemas complexos. As formigas reais serviram de base para tais formigas artificiais. Contudo, foram dadas capacidades para estas últimas, que não são encontradas em suas irmãs reais, para torná-las ainda mais eficientes.

Em (DORIGO; DI CARO; GAMBARDELLA, 1999) pode-se encontrar mais detalhes sobre ACO, tanto em relação as diferenças entre formigas reais e virtuais, quanto em relação ao modelo discutido acima. Além disso, os autores apresentam uma visão geral da aplicação de ACO em problemas de otimização, como os descritos no início desta seção, e mostram o quão poderosa e eficiente pode ser esta abordagem se comparada com outras anteriormente propostas. Na seção 3.4 serão discutidas abordagens de ACO que têm relação com este trabalho.

### 3.4 Trabalhos Relacionados

A seguir serão discutidos os trabalhos relacionados a este que foram realizados na área de inteligência coletiva. Primeiramente, na seção 3.4.1, serão discutidas as abordagens que utilizam o modelo de divisão de trabalho nas colônias de insetos sociais, modelos estes que também serviram como base para a abordagem proposta neste trabalho. Em seguida, na seção 3.4.2, serão discutidas abordagens que utilizam a comunicação através dos feromônios para otimização. Tais abordagens estão relacionadas a este trabalho pois são utilizadas para resolver problemas em cenários próximos aos que são considerados para a experimentação e validação da abordagem proposta.

#### 3.4.1 Divisão de Trabalho e Alocação de Tarefas

(CAMPOS et al., 2001) propuseram a utilização dos modelos teóricos de divisão de trabalho discutidos neste capítulo para a coordenação de agentes que atuam em ambientes dinâmicos. Neste artigo os autores apresentam uma abordagem que é aplicada ao *Dynamic Flow Shop Scheduling Problem*.

Neste cenário, máquinas capazes de realizar diferentes tipos de tarefas são representados por agentes que se comportam como insetos sociais. Novas tarefas chegam em uma linha de produção e existe um custo de configuração das máquinas quando estas trocam de um tipo de tarefa para outro. Os agentes, representando as máquinas, escolhem se vão ou não realizar cada tarefa. O objetivo da abordagem é minimizar o tempo total de realização das tarefas especializando as máquinas por tipo de tarefa.

Experimentos foram realizados com 20 tipos de tarefas e utilizando de 6 a 15 máquinas. Cada máquina possui uma fila de 5 tarefas que podem ficar aguardando para serem realizadas. Uma vez que a tarefa é escolhida pelo agente ela entra nesta fila. A cada passo do experimento uma ou mais máquinas podem falhar. A dinamicidade do cenário está exatamente nestas falhas. As tarefas aguardando na fila de uma máquina que falha não são colocadas de volta na linha de produção.

Os resultados obtidos mostram que a abordagem baseada em insetos sociais tem desempenho equivalente a uma abordagem *market-based*, usada no artigo para comparação. Ambas abordagens tiveram desempenho melhor que abordagens que não utilizam agentes. Os autores detalham as diferenças e semelhanças entre as duas abordagens e argumentam que modelos baseados em insetos sociais, que se mostraram poderosos através de ACO para otimização estática e centralizada, podem ser eficientes também para otimização em cenários dinâmicos e distribuídos. No artigo a alocação de tarefas é vista como um problema de escalonamento que é continuamente resolvido pelos agentes em um ambiente

variável.

(CICIRELLO; SMITH, 2004) apresentam uma extensão da abordagem para o mesmo problema de (CAMPOS et al., 2001), agora intitulado (*Dynamic Shop Floor Routing problem*). Nesta nova abordagem são feitas algumas modificações nas equações apresentadas por (CAMPOS et al., 2001) que melhoram seu desempenho. Além disso é proposto um modelo de competição por dominância, onde agentes que decidem realizar a mesma tarefa disputam pelo direito de realizá-la.

Os aspectos dinâmicos deste cenário são os seguintes: diferentes tipos de tarefas são enviadas simultaneamente para o chão de fábrica; as taxas em que estas tarefas chegam são determinadas pela probabilidade de uma tarefa de um tipo diferente da anterior aparecer. Inicialmente, três casos com diferentes probabilidades para dois tipos de tarefas foram analisados: 50-50, 75-25 e 100-0. A abordagem apresentada foi capaz de especializar as máquinas a estas probabilidades. Mais tarde, uma distribuição variável foi utilizada para simular a dinamicidade do ambiente de uma maneira controlada: a distribuição inicial é alterada depois da primeira metade do experimento.

A adaptação com relação a troca da distribuição necessitou de uma quantidade de tempo considerável. Os autores argumentam que este tempo de adaptação não prejudica o desempenho da abordagem. Apesar desta limitação e do aspecto controlado da dinamicidade do ambiente os resultados mostraram que a utilização dos modelos teóricos dos insetos sociais é competitivo, e em alguns casos até superior, a outros sistemas baseados em agentes apresentados anteriormente (inclusive o de (CAMPOS et al., 2001), cujo modelo foi estendido).

Contudo, existem questões em aberto nas referidas abordagens quanto a especificidade das soluções propostas. Quando aplicado a cenários mais complexos, onde mais de uma máquina pode ser requisitada para realizar a mesma tarefa, a simples especialização não é suficiente. No cenário discutido na seção 5.2, denominado *Machine Sequencing Problem* (MSP), depois que uma tarefa é realizada ela precisa retornar à linha de produção para ser então realizada por outra máquina, em uma sequência pré-estabelecida.

Possivelmente algumas extensões aplicadas a especialização simples (como a utilizada nas abordagens em questão) poderiam lidar com o MSP. Porém, não se sabe o tipo de extensão que deveria ser utilizada para que a abordagem também obtenha bons resultados neste novo cenário. De maneira mais geral, não está claro se estas abordagens são capazes de lidar com outros problemas de escalonamento e alocação de tarefas.

### 3.4.2 Formigas Forrageando e Otimização

(MERKLE; MIDDENDORF; SCHMECK, 2002) apresenta uma abordagem para o *Resource Constraint Project Scheduling Problem* - RCPSP que, segundo os autores, é um problema geral de escalonamento que engloba problemas de *job-shop*, *flow-shop* e *open-shop*. Por este cenário ter sido largamente estudado existem disponíveis comparações de desempenho entre os mais diversos algoritmos e conjuntos de teste padrão que são usados nestas comparações. Já foram propostos algoritmos para RCPSP baseados em algoritmos genéticos, *simulated annealing*, *tabu search*, entre outros.

Muitas abordagens baseadas em ACO têm sido propostas para problemas de escalonamento que são especializações de RCPSP. Nestas abordagens muitas gerações de formigas artificiais buscam por uma boa solução. Cada formiga de uma geração constrói uma solução tomando decisões probabilísticas. Em geral, formigas que encontram uma boa solução marcam com feromônios o caminho que as levou até essa solução. As formigas das próximas gerações são atraídas pelos feromônios e tendem a buscar uma solução próxima

a solução encontrada pela geração anterior. Além dos feromônios, normalmente estas abordagens adicionam alguma heurística, especificamente relacionada com o problema, ao processo de tomada de decisão.

A abordagem proposta em (MERKLE; MIDDENDORF; SCHMECK, 2002) se difere das outras até então propostas por várias razões. Primeiramente, por combinar duas formas diferentes de considerar os feromônios para determinar as probabilidades de escalonamento de cada tarefa, ambas anteriormente propostas. Numa delas, denominada direta, os feromônios depositados em determinada posição possível para o escalonamento de cada tarefa são considerados diretamente para determinar a probabilidade da tarefa ser escalonada nesta posição. Na outra, denominada somatório, os feromônios de cada posição possível no escalonamento, até a posição que se está considerando, são somadas para compor a probabilidade.

Além disso, os autores propõe que a influência da heurística relacionada com o problema diminua com o passar do tempo, ou seja, as primeiras gerações são mais influenciadas pela heurística que as últimas. Esta modificação faz com que a heurística determinística não impeça que as formigas busquem melhores soluções de geração para geração, mas as orienta no início do processo.

Outra modificação importante proposta diz respeito a influência da formiga que encontra a melhor solução e é autorizada a atualizar os feromônios, chamada de elite. Em outras abordagens esta formiga mantém a melhor solução encontrada e atualiza os feromônios de acordo com essa solução em cada geração. Na abordagem proposta esta ação é limitada, permitindo à formiga elite atualizar os feromônios depois de um certo número de gerações.

Foram realizados experimentos comparando esta abordagem com outras baseadas em diferentes tecnologias. Quando o número de soluções que as abordagens puderam computar foi limitado, a abordagem em questão obteve melhores resultados. Quando esta limitação não foi considerada esta abordagem encontrou soluções mais eficientes que as até então conhecidas para 30% dos problemas utilizados nos experimentos.

O RCPSP será utilizado como um dos cenários para experimentar a abordagem que está sendo proposta neste trabalho e será formalizado e melhor discutido no capítulo 4.

Como comentado na seção 2.3.2, alguns problemas de escalonamento e alocação de tarefas também podem ser descritos como Problemas de Satisfação de Restrições - CSPs. Os CSPs são caracterizados por conterem um conjunto de variáveis, cada uma possuindo um domínio específico (conjunto discreto de valores diferentes que a variável pode assumir) e um conjunto de restrições com relação a distribuição dos valores dos domínios entre as variáveis. O problema é considerado resolvido quando o menor número possível de restrições é desrespeitado. Este tipo de problema pode ser mapeado como um grafo onde os nodos representam as variáveis associadas com cada um de seus valores possíveis e os vértices representam as restrições existentes com relação aos pares variável/valor representados pelos nodos interligados pelo vértice em questão.

(TARRANT; BRIDGE, 2004) propõem a aplicação de ACO em CSPs, apresentando um algoritmo básico e discutindo as diversas abordagens para se obter a solução do referido problema usando este algoritmo. Para seu funcionamento, uma quantidade inicial de feromônios é distribuída pelo grafo. Em seguida, formigas artificiais são aleatoriamente distribuídas nos nodos do grafo e se deslocam através dos nodos formando uma atribuição completa, isto é, percorrendo um nodo relacionado com cada variável. Ao final deste percurso são distribuídos feromônios de acordo com a qualidade da solução. Simultaneamente, os feromônios depositados evaporam segundo uma taxa pré definida. O algoritmo

termina quando uma solução para o problema é encontrada ou quando um determinado número de percursos foi realizado.

O objetivo dos autores foi determinar qual destas abordagens é a mais eficiente para no futuro comparar o algoritmo baseada em insetos sociais com outros já apresentadas. Assim, não se tem conhecimento ainda da diferença de desempenho entre ACO e as outras soluções propostas para CSPs, bem como não se sabe a diferença desta solução especificamente se comparada a outras também baseadas em ACO para problemas que possam ser representados como CSPs.

## 4 ABORDAGEM PROPOSTA

Este capítulo está organizado da seguinte forma: na seção 4.1 a abordagem proposta será contextualizada; na seção 4.2 serão discutidas as idéias que são a base da abordagem proposta; na seção 4.3 será apresentada uma versão preliminar da abordagem proposta na qual foram realizados os experimentos que permitiram obter os resultados preliminares do capítulo 5; na seção 4.5 a proposta propriamente dita será detalhada; na seção 4.6 os cenários onde a abordagem será aplicada serão formalizados; e na seção 4.6.3 serão discutidos aspectos relevantes a implementação da abordagem em sistemas multiagentes.

### 4.1 Introdução

Como mencionado no capítulo 1, este trabalho enfoca o problema da alocação ou escalonamento dinâmico distribuído de tarefas em larga escala entre agentes com diferentes competências. Neste problema, tarefas devem ser escalonadas e executadas por agentes. O controle distribuído do processo de escalonamento implica na tomada de decisão dos agentes de modo distribuído, com estes tendo apenas uma visão local sobre as tarefas percebidas e sobre o desempenho obtido depois da tarefa realizada.

Como larga escala entende-se que mais de uma centena de agentes estão envolvidos na realização de mais de uma centena de tarefas. Cada agente tem uma competência diferente para realizar cada tarefa. As tarefas aparecem e desaparecem dinamicamente, ou seja, entram e saem do sistema. Algumas tarefas podem demandar mais de um agente para sua realização. Além disso, algumas tarefas podem interferir na execução de outras de muitas maneiras diferentes, sendo inter-dependentes, mutuamente exclusivas, etc. O objetivo é diminuir o tempo necessário para a realização das tarefas utilizando os agentes mais capacitados para a execução de cada tarefa.

Neste capítulo será proposta uma abordagem para este problema baseada nos modelos teóricos de organização dos insetos sociais discutidos na seção 3.2. O TÆMS, apresentado na seção 2.2, é utilizado para definir as tarefas e sua hierarquia, compondo a dimensão funcional da organização do sistema multiagente que tratará o problema. A abordagem em questão é utilizada pelos agentes para determinar quais tarefas estes estarão engajados e em que momento a tarefa será executada. Desta forma, a dimensão estrutural da organização emerge como resultado da interação dos agentes.

### 4.2 Idéias Básicas

O embasamento da abordagem proposta, como já mencionado, é a plasticidade na divisão do trabalho nas colônias dos insetos sociais. Esta plasticidade emerge da interação

entre os indivíduos e deles com o ambiente, utilizando padrões simples de comunicação e sem coordenação explícita.

Centenas, e até milhares, de indivíduos compõe uma colônia que se auto-organiza para atender suas necessidades em um ambiente dinâmico. A alocação de tarefas entre os indivíduos que compõem uma colônia deve ser tal que as variações do ambiente sejam absorvidas e que a energia gasta pela colônia seja a menor possível. Os indivíduos trocam de tarefa para satisfazer as demandas da colônia em relação as mudanças no ambiente e buscam desempenhar as tarefas para as quais tem maior competência, usualmente determinada por sua forma física.

Agentes podem decidir quais tarefas realizar e quando realizá-las inspirados na forma como os insetos sociais tomam este tipo de decisão, sem utilizar processos explícitos de coordenação e buscando a maior eficiência possível. Este comportamento é adequado para agentes que compõem sistemas multiagentes engajados em problemas dinâmicos e em larga escala.

Para representar as tarefas necessárias para que o SMA atinja seu objetivo, além da especificação da organização hierárquica destas tarefas e sua inter-dependência, a abordagem proposta utiliza a linguagem TÆMS. Esta linguagem, explicada em detalhes na seção 2.2, permite especificar de uma forma bastante rica essa estrutura de tarefas. Os agentes determinam de forma probabilística se devem se engajar na realização de uma tarefa utilizando como base o modelo teórico do limiar de resposta, apresentado em 3.2, equação 3.1. As tarefas, representadas em TÆMS, produzem estímulos para os agentes de acordo com suas características e relacionamentos.

Deve-se considerar que os agentes atuam em ambientes dinâmicos, onde a estrutura de tarefas descrita em TÆMS pode ser modificada em tempo de execução: métodos podem aparecer e desaparecer; o número de agentes pode mudar; e podem variar os relacionamentos entre os métodos. Contudo, a estrutura de tarefas pode mudar apenas quando o objetivo desta estrutura é atingido. Esta restrição é utilizada para que sempre se obtenha um escalonamento completo da estrutura de tarefas, possibilitando o aprendizado por parte dos agentes. Cada intervalo de tempo necessário para se obter um escalonamento completo foi denominado *rodada*. Cada passo em uma rodada para que um escalonamento seja obtido é denominado *iteração*.

O modelo de variação do estímulo não considera as mudanças em relação ao número de agentes e métodos. (BONABEAU; DORIGO; THERAULAZ, 2000) mostram que o sucesso emergente de tarefas pode ser obtida com o uso de limiares fixos mas com aplicabilidade limitada. A especialização dos indivíduos em conjuntos de tarefas é um aspecto do comportamento das colônias de insetos sociais que é importante e bastante estudado (ROBISON, 1992).

A especialização dos indivíduos de acordo com sua idade é denominado pelos biólogos de *polietismo temporal*. Nas abelhas esta é a principal forma de se obter a divisão de trabalho. Trabalhadores jovens realizam tarefas dentro da colméia e trabalhadores mais velhos realizam tarefas fora dela, como defender a colônia e forragear.

Propõe-se uma modificação no modelo de especialização dos insetos sociais, apresentado em 3.2. Nesta modificação, o limiar interno dos agentes é atualizado de acordo com sua idade e a idade das tarefas. Por idade de um agente e de um método entende-se o tempo em que os agentes estão atuando no sistema e o tempo em que o método está disponível para execução, respectivamente.

Através desta modificação, os agentes têm limiares de resposta altos para métodos que estão aguardando a mais tempo para serem executados e baixos para métodos novos. Isso

garante que as tarefas não permaneçam muito tempo aguardando para serem realizadas. Tarefas que foram percebidas a mais tempo pelo sistema estimulam mais intensamente os agentes, pois estes terão um limiar de resposta maior para estas tarefas. Além disso, agentes mais novos tem limiares mais baixos que agentes mais velhos com relação aos mesmos métodos. A idéia por traz disso é especializar agentes mais velhos em um conjunto maior de métodos em relação aos agentes mais novos, como ocorre com os insetos sociais.

A especialização dos indivíduos de acordo com sua forma é denominada pelos biólogos de *polietismo morfológico*, ou simplesmente *polimorfismo*. Nas colônias de formigas o polimorfismo tem um papel de suma importância para a divisão do trabalho. O polimorfismo pode ser modelado através do limiar de resposta discutido na seção 3.2. Para diferenciar os agentes e obter a especialização morfológica basta iniciar os limiares internos de cada agente de acordo com as tarefas as quais sua “forma” é mais adaptada. Agentes que partem com limiares altos para determinadas tarefas terão maior tendência a realizá-las.

Uma vez que no problema que está sendo considerado neste trabalho os agentes tem diferentes competências para realizar diferentes tarefas, o limiar interno do agente será iniciado para refletir esta competência. Esta inicialização só poderá ocorrer para as tarefas previamente conhecidas pelos agentes. Caso isso não ocorra, os agentes são iniciados com um limiar de nível intermediário para tais tarefas.

Como mencionado anteriormente, potencialmente cada inseto na colônia pode realizar todos os tipos de tarefas, mesmo quando sua morfologia é especialmente adaptada para alguma tarefa em específico e independentemente de sua idade. Assim, a despeito do polietismo temporal e do polimorfismo, os indivíduos de uma colônia de insetos sociais podem se engajar em tarefas que não estão associadas a sua forma ou idade para atender à demandas de acordo com as necessidades da colônia. Se o número de indivíduos engajados em uma tarefa não é suficiente ou se esses indivíduos não estão atendendo tal demanda a contento, outros indivíduos podem se engajar nesta tarefa.

Propõe-se condicionar a atualização dos limiares internos, seguindo o modelo discutido em 3.2, equações 3.3 e 3.4, ao sucesso do agente em desempenhar uma tarefa. Assim, quanto mais eficiente é o agente na realização de uma tarefa, de acordo com sua competência, mais especializado nesta tarefa o agente será. A influência do polietismo temporal nesta especialização é importante para garantir que o sistema não gere agentes super-especializados que não atendam a novas tarefas.

### 4.3 Versão Preliminar

A versão preliminar da abordagem em questão foi previamente proposta em (OLIVEIRA; FERREIRA; BAZZAN, 2004) e (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005). Nesta versão, algumas idéias básicas da abordagem foram consideradas: os agentes tem uma tendência a se engajar na execução de uma tarefa; as tarefas produzem estímulos para os agentes; os agentes utilizam um limiar interno que varia com sua idade para, junto com o estímulo, determinar a tendência mencionada a acima; e as tarefas e suas inter-dependências são modeladas em TÆMS.

A tendência  $T_{\theta_{ij}}(s_j)$  do agente  $i$  se engajar na tarefa  $j$ , computada da mesma forma como fazem os insetos sociais, é determinada pela relação entre o estímulo da tarefa e o limiar interno do agente através da equação 4.1.



$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \quad (4.1)$$

onde:

$s_j$  estímulo associado com a tarefa  $j$

$\theta_{ij}$  limiar interno do agente  $i$  em relação a tarefa  $j$

O agente decide executar ou não uma tarefa de forma probabilística. Uma vez que a tarefa  $j$  é percebida, o agente  $i$  decide, com probabilidade dada por  $T_{\theta_{ij}}(s_j)$ , se irá se engajar na execução de  $j$ .

#### 4.3.1 Estímulo

Como mencionado na seção 2.2, o método é o elemento da estrutura de tarefas em TÆMS que representa o que os agentes realmente devem executar. Todos os métodos em TÆMS possuem distribuições de probabilidade associadas a qualidade, custo e duração. Assim, a tarefa  $j$  tem qualidade  $q_j$ , custo  $c_j$ , e duração  $d_j$ . Estes parâmetros são usados para calcular o estímulo  $s_j$ . A intensidade deste estímulo está associada com o resultado da execução do respectivo método. Cada método  $j$  tem um estímulo  $s_j$  determinado pela equação 4.2.

$$s_j = \varphi * (\alpha * \hat{q}_j - \beta * \hat{c}_j - \gamma * \hat{d}_j + \beta + \gamma) + (1 - \varphi) * x_j \quad (4.2)$$

onde:

$\hat{q}_j$  qualidade normalizada do método  $j$

$\hat{c}_j$  custo normalizado do método  $j$

$\hat{d}_j$  duração normalizada do método  $j$

$x_j$  estímulo associado com a QAF relacionada ao método  $j$

$\alpha, \beta, \gamma, \varphi$  constantes

Na equação 4.2 as constantes são utilizadas para atribuir pesos diferentes a qualidade ( $\alpha$ ), custo ( $\beta$ ) e duração ( $\gamma$ ), cuja soma de ser igual a um. O estímulo cresce de acordo com a qualidade (diretamente proporcional) e decresce de acordo com o custo e a duração (inversamente proporcional). Para conseguir isso, o custo e a duração são subtraídos da qualidade. Como custo e duração assumem valores da mesma ordem de magnitude da qualidade (essas variáveis são normalizadas), as constantes  $\beta$  e  $\gamma$  (relacionadas ao custo e a duração) são adicionadas a equação. Isso é utilizado para equilibrar a influência das variáveis inversamente e diretamente proporcionais ao estímulo. A normalização da qualidade, custo e duração é feita através da proporção destas variáveis em relação ao valor máximo total (somatório) de seus valores em toda a estrutura de tarefas. A equação 4.3 ilustra este cálculo para o custo.

$$\hat{c}_j = \frac{c_j}{\sum_{j=0}^M c_j} \quad (4.3)$$

Além disso, a constante  $\varphi$  é utilizada para dar diferentes pesos ao estímulo associado com a qualidade, custo e duração de execução do método ( $\alpha * \hat{q}_j - \beta * \hat{c}_j - \gamma * \hat{d}_j + \beta + \gamma$ )

e o estímulo relacionado com a sucessão emergente dos métodos ( $x_j$ ), discutido a seguir. Nesta versão da abordagem foi utilizado  $\varphi = 0.5$ .

O estímulo  $s_j$  para cada método  $j$  é recalculado toda vez que um método é executado por um agente (o que será chamado de iteração). A atualização do estímulo é realizada de acordo com o modelo de sucessão emergente de tarefas discutido em (BONABEAU; THRAULAZ; DORIGO, 1999). Nas colônias de insetos sociais, a realização de uma tarefa aumenta a demanda pela realização de outra tarefa que esteja relacionada. Por exemplo, a limpeza do ninho gera uma pilha de sujeira na entrada deste ninho, que produz a necessidade de limpar a entrada. Nesta versão da abordagem, a execução de um método influencia o estímulo de todos os outros métodos da mesma tarefa no TÆMS de acordo com a QAF. Esta influência está representada na equação 4.2 pela variável  $x_j$ .

Suponha a existência de  $M$  métodos na TS percebida por um agente. Quando qualquer método  $k$  do conjunto de  $M$  métodos é realizado, todos os métodos  $j$  de  $M$ , com exceção de  $k$ , terão seus estímulos atualizados segundo a equação 4.4.

$$x_j = x_j + \kappa \quad (4.4)$$

onde:

$\kappa$  é a constante relacionada com o tipo de QAF segundo a tabela 4.1

Esta influência é recursiva em relação as tarefas da TS que o método está inserido. A constante  $\kappa$  está associada a QAF (discutida em detalhes na seção 2.2) usada para modelar a influência dos métodos que são relacionados entre si. Foram adotados valores pequenos para o  $\kappa$  ( $0 < \kappa \leq 1$ ) porque o estímulo  $x_j$  é cumulativo, aumentando a cada iteração. O estímulo  $s_j$  é limitado para não ser menor que 0 nem maior que 1, variando apenas entre estes limites.

Tabela 4.1: Constantes relacionadas a QAF.

QAF	$\kappa$
SeqMax, Max, SeqMin, Min	0
SeqSum, Sum, All	0.01
ExactlyOne	-0.01

### 4.3.2 Polietismo Temporal

Por definição, cada inseto na colônia pode realizar qualquer tipo de tarefa, mesmo que sua morfologia ou idade seja especialmente adaptada para alguma delas especificamente. Nesta versão preliminar todos os agentes tem igual competência para execução de todas as tarefas, não foi considerada nenhuma propensão específica em relação a sua forma física como acontece com os insetos sociais.

O modelo de limiar de resposta original, mostrado na seção 3.2, utiliza duas constantes como coeficiente relacionados ao aprendizado e ao esquecimento. A versão da abordagem que se está discutindo propõe estender este modelo para que a aprendizagem e o esquecimento variem de acordo com a idade dos agentes, modelando o polietismo temporal.

O limiar de resposta  $\theta_{ij}$  do indivíduo  $i$  quando realiza a tarefa  $j$  é atualizado pela equação 4.5. Os limiares de resposta das tarefas que não estão sendo realizadas por este mesmo agente é atualizado pela equação 4.6.

$$\theta_{ij} = \theta_{ij} - \frac{a_i}{A_i} * \frac{\mathcal{A} - m_j}{\mathcal{A}} \quad (4.5)$$

$$\theta_{ij} = \theta_{ij} + \frac{a_i}{A_i} * \frac{m_j}{\mathcal{A}} \quad (4.6)$$

onde:

$a_i$  idade do agente  $i$

$A_i$  idade máxima estimada para o agente  $i$

$m_j$  idade do método  $j$

$\mathcal{A}$  idade do método que está a mais tempo disponível para ser executado

Neste modelo de limiar de resposta todos os agentes iniciam com os mesmos valores para o limiar  $\theta_{ij}$ , usualmente um valor intermediário. Quando um método é executado pelo agente, o limiar de resposta é atualizado. Para que um agente se especialize na execução de um método é necessário que este agente selecione este método alguma vezes.

Nas equações 4.5 e 4.6 são consideradas a idade do agente (proporcional ao tempo máximo previsto no TÆMS para a execução das tarefas) e a idade dos métodos (proporcional ao tempo do método que está a mais tempo disponível, ou seja, o método mais velho). A idade dos agentes e dos métodos é calculada a cada iteração. A idade do método aumenta até que este seja executado por um agente. Os agentes tem probabilidades associadas com relação a permanecerem ativos ou não.

## 4.4 Limitações da Versão Preliminar

Na versão preliminar da abordagem o estímulo é computado com o objetivo de fazer os agentes respeitarem as restrições impostas pela organização da tarefas em TÆMS, ou seja, os agentes devem ser estimulados a cumprir o objetivo do sistema de acordo com a estrutura de tarefas. São consideradas as relações locais, ditadas pelas QAFs, e algumas não locais, representadas pelos NLEs restritivos. O polietismo temporal foi considerado porque esta versão da abordagem já se propunha a lidar com problemas dinâmicos.

Quando uma tarefa estava relacionada com outra através de um NLE do tipo *disable*, se a primeira tarefa fosse executada a segunda passava a não ser percebida mais pelos agentes. Da mesma forma, se duas tarefas se relacionavam através de um NLE do tipo *enable*, a segunda tarefa só passava a ser percebida depois que a primeira fosse realizada. Os NLEs do tipo *facilitate* e *hinder* não foram considerados por esta versão da abordagem. Os recursos modelados em TÆMS, também descritos na seção 2.2, são tratados de forma semelhante aos NLEs. Quando uma tarefa demandava um recurso para sua realização, esta tarefa só era percebida pelos agentes se houvesse quantidade suficiente de recurso disponível.

Além disso, quando um agente passava a executar uma tarefa, esta tarefa não era mais percebida pelos demais agentes. Desta forma, apenas um agente poderia estar envolvido na execução de uma tarefa a cada instante. Para representar tarefas que necessitassem do envolvimento de mais de um agente, utilizou-se a sub-divisão desta tarefa em duas sub-tarefas.

A versão preliminar da abordagem é limitada em muitos sentidos por não considerar todas as idéias descritas na seção anterior. Esta versão da abordagem não considera a especialização e o polimorfismo, não lidando com as diferentes competências dos agentes e seu desempenho na realização das tarefas. Outra limitação importante desta primeira versão é a forma como o estímulo é determinado, utilizando as características dos métodos (qualidade, custo e duração) e as QAFs

que definem as relações entre as tarefas. A atualização do estímulo, neste caso, foi feita apenas em função da QAF.

Quando é possível ter a informação completa sobre as tarefas que precisam ser executadas e há tempo suficiente para o planejamento, os agentes poderiam realizar várias rodadas para, influenciados pelo estímulo modificado em função de seu desempenho, diminuir o tempo total utilizado. Dessa forma, poderia-se obter um escalonamento melhor a cada rodada.

A primeira extensão da abordagem proposta, em relação a sua versão preliminar, diz respeito a forma como a QAF é incorporada na abordagem. A proposta agora é utilizar a QAF como parte da tendência, influenciando igualmente na tomada de decisão do agente, mas não mais compondo o estímulo. A atualização do estímulo será então modificada para que o desempenho dos agentes na solução do problema seja o único responsável disso.

As relações entre as tarefas, expressas em TÆMS, são agora utilizadas para computar uma demanda que é associada a cada tarefa. Esta demanda compõe a tendência do agentes de realizar cada tarefa, mas não influencia seus estímulos em relação a cada uma delas. O estímulo é atualizado para motivar aos agentes a realizar tarefas o mais cedo possível, para que o tempo total necessário para a realização de todas as tarefas seja minimizado.

Além disso, quando não há conhecimento sobre a duração das tarefas nem tempo disponível para melhorar o escalonamento através da realização de várias rodadas, planejar não é possível. Quando é este o caso, apenas interessa a alocação dos agentes mais competentes em relação a cada tarefa. Com isso, a versão preliminar foi modificada para que o desempenho dos agentes influencie sua especialização de modo que estes aprendam com as tentativas, para obter alocações cada vez mais eficientes em relação a suas competências. Como comentado, a versão preliminar não considerava este tipo de especialização nem as diferentes competências.

A seguir serão detalhadas as extensões e modificações feitas na versão preliminar da abordagem proposta para que todas as idéias básicas sejam consideradas e, com isso, habilitá-la a lidar com o problema em questão neste trabalho.

## 4.5 Extensões e Modificações

A tendência do agente  $i$  se engajar no método  $j$  é agora determinada pela relação entre o estímulo do método, o limiar interno do agente e a demanda produzida pelas QAFs relacionadas a tarefa  $j$ , como mostra a equação 4.7.

$$T_{\theta_{ij}}(s_j) = \sigma \frac{s_j^2}{s_j^2 + \theta_{ij}^2} + (1 - \sigma)X(j) \quad (4.7)$$

onde:

$s_j$  estímulo associado com a tarefa  $j$

$\theta_{ij}$  limiar interno do agente  $i$  em relação a tarefa  $j$

$X(j)$  demanda associada com as QAFs relacionadas ao método  $j$ ,  $0 \leq X(j) \leq 1$

$\sigma$  constante para determinar o peso de  $X(j)$  no cálculo da tendência

Como já mencionado, cada tarefa em TÆMS é dividida em sub-tarefas de maneira recursiva, compondo uma estrutura em forma de árvore, a partir da TG (nodo raiz) até sua forma indivisível, os métodos (nodos folha). Somente os métodos são de fato executados pelos agentes. Cada método  $j$  possui qualidade  $q_j$ , custo  $c_j$  e duração  $d_j$  como atributos.

As QAFs dão significado aos relacionamentos entre as sub-tarefas de cada tarefa. Quando uma tarefa está sendo executada ela pode demandar que outra tarefa também seja executada, pois só assim a super-tarefa que abriga as duas será realizada por completo. Ou ainda, uma tarefa

sendo executada inibe outras tarefas da mesma super-tarefa, pois esta só precisa que uma de suas sub-tarefas seja executada. A tabela 2.1 mostra todas as possibilidades de QAFs.

A equação 4.8 modela parte do que se discutiu acima. O coeficiente de execução  $e_j$  associado a tarefa  $j$  é calculado pela proporção de tarefas que estão sendo (ou já foram) executadas em relação ao número total de tarefas da super-tarefa onde  $j$  está inserida. A função  $P(j)$  retorna a super-tarefa da tarefa  $j$ . O valor de  $e_j$  é zero quando a tarefa  $j$  está concluída.

$$e_j = \begin{cases} \frac{1+n_{P(j)}}{N_{P(j)}} & \text{se } n_{P(j)} \neq N_{P(j)} \\ 0 & \text{caso contrário} \end{cases} \quad (4.8)$$

onde:

$n_{P(j)}$  número de sub-tarefas da super-tarefa de  $j$  que foram ou estão sendo executadas

$N_{P(j)}$  número total de sub-tarefas da super-tarefa de  $j$

Para garantir o significado das QAFs, a função  $X(j)$  retorna a demanda associada com as QAFs relacionadas ao método  $j$ . Assim, se uma super-tarefa precisa que todas suas sub-tarefas sejam executadas para que possa ser realizada,  $X(j)$  deve aumentar a tendência dos agentes em relação as sub-tarefas quando uma delas for iniciada. Quando o caso é o contrário, e a super-tarefa necessita que uma, e apenas uma, de suas sub-tarefas sejam executadas,  $X(j)$  deve diminuir a tendência dos agentes de executarem as outras sub-tarefas quando uma das sub-tarefas está em execução. Isto deve se dar de acordo com a semântica das QAFs e recursivamente em relação a toda TS até sua raiz (TG). A equação 4.9 é responsável por este comportamento, onde define-se  $X(j) \forall j \mid P(j) \neq TG$ . O  $X$  foi utilizado para representar essa função pois na versão preliminar a demanda associada as QAFs utilizava o  $x$  (minúsculo).

$$X(j) = \begin{cases} e_j + \psi X(P(j)) & QAF \in \{min, all, max, seq\_max, seq\_min\} \\ \omega * [e_j + \psi X(P(j))] & QAF \in \{sum, seq\_sum\} \\ \lfloor 1 - e_j \rfloor & QAF \in \{exactly\_one\} \end{cases} \quad (4.9)$$

onde:

$P(j)$  super-tarefa de  $j$

$\psi$  taxa de desconto para diminuir a demanda das tarefas mais distantes na TS

$\omega$  taxa de desconto para diminuir peso da QAF quando as relações não forem restritivas

$\lfloor e_j \rfloor$  valor inteiro resultante do arredondamento por corte de  $e_j$ , (*floor*)

A taxa de desconto  $\omega$  é utilizada para diminuir o peso da QAF quando as relações não forem restritivas, ou seja, relações que não sejam do tipo AND como as QAFs *min* e *all* e do tipo XOR como a QAF *exactly\_one*. As relações que não são restritivas indicam a preferência por uma restrição mas não sua obrigatoriedade, o que diminui o impacto que a QAF correspondente deve causar sobre a tendência. A taxa de desconto  $\psi$  é utilizada para determinar a diferença entre os pesos das QAFs conforme o nível de relação das tarefas, ou seja, de acordo com a distância na hierarquia que se encontram estas tarefas. Conforme a função sobe recursivamente na TS o peso da QAF correspondente diminui de acordo com  $\psi$ , onde  $0 \leq \psi \leq 1$ .

Um agente tem probabilidade  $p$  de abandonar a execução de um método enquanto não há outros agentes envolvidos em métodos relacionados diretamente por um super-tarefa. Se todos os métodos requeridos por uma tarefa estão sendo executados então os agentes não poderão abandoná-los ( $p = 0$ ). A implicação desta hipótese será discutida na seção 4.7.

#### 4.5.1 Estímulo

O estímulo local  $s_j(t)'$  relacionado ao método  $j$  na rodada  $t$  é dado pela equação 4.10. Neste caso, o estímulo local decresce com o passar do tempo e apenas cresce segundo uma função que reflete o melhor escalonamento obtido pelos agentes.

$$s_j(t)' = s_j(t - \mu)' * \delta + \Gamma(j) \quad (4.10)$$

onde:

$s_j(t)'$  estímulo local do método  $j$  na rodada  $t$

$s_j(t - \mu)'$  estímulo local do método  $j$  em uma rodada anterior  $t - \mu$

$\delta$  taxa de desconto para diminuir o impacto do estímulo com o passar do tempo

$\Gamma(j)$  viés de atualização do estímulo em  $j$

$\mu$  constante que define o intervalo de rodadas entre as atualizações

A atualização do estímulo local segundo a equação 4.10 se dará com frequência  $\mu$ . Desta forma, durante  $\mu$  rodadas os agentes armazenam as soluções obtidas e, apenas na última delas, utilizam a melhor solução como base para o cálculo do viés de atualização  $\Gamma(j)$ .

Suponha que  $M$  métodos são escalonados e que  $start_k(j)$  é a função que retorna a iteração (tempo) em que o método  $j$  começou a ser executado no escalonamento  $k$ . São armazenados  $\mu$  escalonamentos, uma para cada rodada. O conjunto de escalonamentos pode ser representado por  $\mathcal{S}$ , onde  $\mathcal{S}_k = \{start_k(1), start_k(2), \dots, start_k(j), \dots, start_k(M)\}$ .

Quando a TS é totalmente executada (término de uma rodada), o escalonamento resultante pode ter seu desempenho medido de acordo com sua duração total. Esta duração total  $|\mathcal{S}_k|$  do escalonamento  $\mathcal{S}_k$  é dada pelo número da iteração em que o último método foi terminado, como mostra a equação 4.11.

$$|\mathcal{S}_k| = \max_{j=1}^M [start_k(j) + d_j] \quad (4.11)$$

onde:

$start_k(j)$  iteração onde o método  $j$  teve sua execução iniciada no escalonamento  $k$

$d_j$  duração do método  $j$

A duração  $d_j$  de cada método  $j$ , utilizada na equação 4.11 para compor a duração total do escalonamento, é o tempo real de execução do método durante a respectiva rodada. Assim, se os métodos em TÆMS forem definidos com uma distribuição de probabilidade para a duração, uma vez que o método foi executado, uma das durações da distribuição será utilizada.

O melhor escalonamento em  $\mathcal{S}$ , cujo índice é  $k^*$ , é determinado pela equação 4.12. O escalonamento  $\mathcal{S}_{k^*}$  é aquele que tem o menor tempo de duração dentre os armazenados em  $\mathcal{S}$  nas últimas  $\mu$  rodadas.

$$k^* = \underset{k=1}{\operatorname{argmin}}^{\mu} |\mathcal{S}_k| \quad (4.12)$$

onde:

$|\mathcal{S}_k|$  duração total  $|\mathcal{S}_k|$  do escalonamento  $\mathcal{S}_k$

O viés de atualização  $\Gamma(j)$  é calculado pela equação 4.13. O objetivo deste viés é estimular as tarefas a começarem o mais cedo possível para que o escalonamento tenha seu tempo total reduzido a cada rodada.

$$\Gamma(j) = \frac{start_{k^*}(j)}{|\mathcal{S}_{k^*}|} \quad (4.13)$$

onde:

$start_{k^*}(j)$  iteração onde o método  $j$  teve sua execução iniciada no melhor escalonamento  $k \in \mathcal{S}$   
 $|\mathcal{S}_{k^*}|$  duração total do melhor escalonamento  $\mathcal{S}_{k^*}$

Uma vez que o estímulo local é determinado, o agente compõe este estímulo local do método com os estímulos dos métodos que se relacionam com este através de NLEs não restritivas para compor o estímulo do método. Suponha que os conjuntos  $\mathcal{H}^l$  e  $\mathcal{F}^j$  contêm os métodos que se relacionam com o método  $j$  através dos NLEs não restritivos *hidden* e *facilitate*, respectivamente. O conjunto  $\mathcal{F}^j$  contêm as tarefas que são facilitadas pelo método  $j$  e o conjunto  $\mathcal{H}^l$  contêm as tarefas que são dificultadas pela execução do método  $j$ . O estímulo  $s_j$  relacionado ao método  $j$  é calculado em função de seu estímulo local  $s'_j$  e os estímulos das tarefas que pertencem a estes conjuntos, como mostra a equação 4.14;

$$s_j = s'_j - \vartheta \sum_{k \in H} s_k + \varsigma \sum_{k \in F} s_k \quad (4.14)$$

onde:

$s'_j$  estímulo local do método  $j$  na rodada  $t$

$\vartheta$  taxa de desconto para diminuir o impacto do estímulo das tarefas em  $\mathcal{H}^l$

$\varsigma$  taxa de desconto para diminuir o impacto do estímulo das tarefas em  $\mathcal{F}^j$

#### 4.5.2 Especialização

Na abordagem que está sendo proposta aqui, os agentes podem ter competências diferentes para executar tarefas diferentes. Com isso, assim como o desempenho dos agentes quanto ao tempo total do escalonamento influencia o estímulo, o desempenho destes agentes com relação a sua competência de executar as tarefas deve influenciar sua especialização.

Agentes devem se especializar nas tarefas para as quais têm maior competência, além de possuírem o polietismo temporal descrito na seção 4.3.2.

A qualidade  $q_{ij}$  em que o agente  $i$  executa o método  $j$  deve ser calculada em função de sua competência. Cada agente  $i$  tem uma competência  $capability_i(j)$ , onde  $0 \leq capability_i(j) \leq 1$ , para executar a tarefa  $j$ . Assim, a qualidade  $q_{ij}$  da execução de um método é dada pela equação 4.15.

$$q_{ij} = q_j * capability_i(j) \quad (4.15)$$

onde:

$q_j$  qualidade do método  $j$

$capability_i(j)$  competência do agente  $i$  em realizar a tarefa  $j$

O TÆMS irá considerar a qualidade  $q_{ij}$  de cada método  $j$ , de acordo com os agentes  $i$  que os executaram, para determinar a qualidade total acumulada no TG para cada escalonamento  $\mathcal{S}_k \in \mathcal{S}$ . A qualidade total da estrutura de tarefas  $QAF_{TG}(\mathcal{S}_k)$ , para o escalonamento  $\mathcal{S}_k$ , é computada segundo as QAFs de cada tarefa que se relaciona com o método  $j$  na estrutura de tarefas.

O custo  $cost(\mathcal{S}_k)$  do escalonamento  $\mathcal{S}_k$  é o somatório dos custos  $c_j$  de cada método  $j$  executado, como mostra a equação 4.16.

$$cost(\mathcal{S}_k) = \sum_{j=1}^M c_j \quad (4.16)$$

onde:

$c_j$  custo do método  $j$

O desempenho  $D(\mathcal{S}_k)$  dos agentes, no escalonamento  $\mathcal{S}_k$ , de acordo com o custo e a qualidade acumulada é dado pela equação 4.17.

$$D(\mathcal{S}_k) = \begin{cases} \frac{QAF_{TG}(\mathcal{S}_k)}{cost(\mathcal{S}_k)} & \text{se } cost(\mathcal{S}_k) \neq 0 \\ QAF_{TG}(\mathcal{S}_k) & \text{caso contrário} \end{cases} \quad (4.17)$$

onde: onde:

$QAF_{TG}(\mathcal{S}_k)$  qualidade acumulada total segundo as QAFs para o escalonamento  $\mathcal{S}_k$

$cost(\mathcal{S}_k)$  custo total do escalonamento  $\mathcal{S}_k$

Para o cálculo deste desempenho assumi-se que, se o custo total do escalonamento  $cost(\mathcal{S}_k)$  for igual a zero, o desempenho será determinado apenas pela qualidade acumulada total  $QAF_{TG}(\mathcal{S}_k)$ . Caso esta qualidade seja igual a zero, o desempenho do agente também será zero e o limiar interno não será atualizado de acordo com esse desempenho, sendo influenciado apenas pelo polimorfismo.

O desempenho  $D(\mathcal{S}_k)$  dos agentes será utilizado para, a cada  $\mu$  rodadas, atualizar o limiar interno dos agentes com o objetivo de especializá-los em relação a melhor alocação obtida.

A melhor alocação obtida em um escalonamento em  $\mathcal{S}$ , cujo índice é  $k^+$ , é determinado pela equação 4.18. O escalonamento  $\mathcal{S}_{k^+}$  é aquele cujos agentes obtiveram o melhor desempenho dentre os armazenados em  $\mathcal{S}$  nas últimas  $\mu$  rodadas.

$$k^+ = \underset{k=0}{\operatorname{argmax}}^\mu D(\mathcal{S}_k) \quad (4.18)$$

onde:

$D(\mathcal{S}_k)$  desempenho dos agentes no escalonamento  $\mathcal{S}_k$

Cada agente  $i$  deve ter seu limiar interno atualizado em relação a tarefa  $j$  que executou no escalonamento  $\mathcal{S}_{k^+}$ . O limiar de resposta  $\theta_{ij}$  de um agente  $i$  em relação ao método  $j$  é atualizado segundo a equação 4.19, quando o agente  $i$  executa  $j$  em  $\mathcal{S}_{k^+}$ , e pela equação 4.20, caso contrário.

$$\theta_{ij}(t) = \theta_{ij}(t - \mu) - \xi * \left( \frac{a_i}{A_i} * \frac{\mathcal{A} - m_j}{\mathcal{A}} \right)^\varrho \quad (4.19)$$

onde:

$\xi$  coeficiente de aprendizado

$a_i$  idade do agente  $i$

$A_i$  idade máxima estimada para o agente  $i$

$m_j$  idade do método  $j$

$\mathcal{A}$  idade do método que está a mais tempo disponível para ser executado

$\varrho$  constante inteira ( $0 \leq \varrho \leq 1$ ) que determina se o polimorfismo será utilizado no cálculo da atualização do limiar interno.



$$\theta_{ij}(t) = \theta_{ij}(t - \mu) + \rho * \left( \frac{a_i}{A_i} * \frac{m_j}{\mathcal{A}} \right)^e \quad (4.20)$$

onde:

$\rho$  coeficiente de esquecimento

Os coeficientes  $\xi$  e  $\rho$  são utilizados para aumentar e diminuir, respectivamente, os limiares internos dos agentes em relação as tarefas onde os agentes obtiveram o melhor desempenho. Em ambas equações o modelo baseado no polietismo temporal, apresentado na seção 4.3.2, é também considerado. A constante  $\alpha$  é utilizada para determinar o peso do polietismo temporal na variação do limiar.

### 4.5.3 Polimorfismo

A equação 4.21 calcula o valor inicial do limiar interno  $\theta_{ij}$  como sendo determinado pela competência do agente  $capacity_i(j)$ . A constante  $\lambda, 0 \leq \lambda \leq 1$ , é utilizada para determinar o peso desta competência no limiar interno.

$$\theta_{ij} = \lambda * capacity_i(j) \quad (4.21)$$

onde:

$capacity_i(j)$  competência do agente  $i$  executar a tarefa  $j$ ,  $0 \leq capacity_i(j) \leq 1$

$\lambda$  taxa de desconto para o papel da competência na inicialização do limiar interno

## 4.6 Cenários

Como comentado no capítulo 1, dois problemas da área de pesquisa operacional serão utilizados neste trabalho para experimentar a abordagem proposta. O *Resource-Constrained Project Scheduling Problem* (RCPSP) e o *Generalized Assignment Problem* (GAP) representam partes complementares do problema em questão neste trabalho e modelam vários aspectos de aplicações reais.

Estes problemas não são originalmente dinâmicos e distribuídos, mas podem ser facilmente estendidos para lidar com essas características. Ambos são problemas NP-difíceis. A seguir, cada um deles será discutido e formalizado. Além disso, será detalhado como esses problemas serão utilizados como cenários para os experimentos da abordagem proposta. Na seção 4.6.3 serão discutidas as diferenças entre os cenários, seu papel na experimentação da abordagem e outros aspectos a respeito dos experimentos que serão conduzidos.

### 4.6.1 RCPSP

O *Resource-Constrained Project Scheduling Problem* (RCPSP) é um problema de otimização onde deve-se escalonar as atividades de um projeto de modo que o tempo total seja minimizado. Além disso, determinadas restrições de precedência entre as atividades devem ser respeitadas e os recursos necessários para as atividades do escalonamento, a cada unidade de tempo, não devem exceder a restrições de capacidade destes diferentes recursos.

O RCPSP é uma generalização de vários problemas de escalonamento. Os cenários complexos utilizados em (MAHESWARAN et al., 2004) para validar os algoritmos de alocação de tarefas, como descrito na seção 2.3.2, são problemas generalizáveis como RCPSPs. Esta característica contribuiu para a escolha deste problema para compor um dos cenários onde a abordagem proposta será experimentada. Vários problemas reais da área de sistemas multiagentes, como o agendamento múltiplo de reuniões e o controle de redes de sensores distribuídos, foram modelados como instâncias de RCPSPs na literatura.

Tabela 4.2: Duração  $d_j$  e demanda de agentes  $R_j$  para cada tarefa.

Tarefa	A	B	C	D	E
$d_j$	4	9	2	2	2
$R_j$	1	3	2	2	1

Suponha que  $\mathcal{J}$  represente o conjunto de  $n$  atividades de um projeto. Uma lista de atividades é uma permutação destas atividades tal que cada atividade aparece na lista antes das atividades que são suas sucessoras no que diz respeito as relações de precedência. Um conjunto de  $r$  recursos é denotado por  $\mathcal{Q}$ . A capacidade associada a cada recurso do tipo  $i$  é dada por  $R_i > 0$ , onde  $i \in \mathcal{Q}$ . Cada atividade  $j \in \mathcal{J}$  tem uma duração  $d_j$  e requer um conjunto de recursos  $r_{j,1}, \dots, r_{j,q}$ , onde  $r_{j,i}$  representa a necessidade do recurso do tipo  $i$  por unidade de tempo para que a atividade  $j$  seja escalonada. O conjunto de predecessores diretos de  $j$  é dado por  $\mathcal{P}_j(\mathcal{S}_j)$  e o conjunto de todos os predecessores de  $j$  é dado por  $\mathcal{P}_j^*(\mathcal{S}_j^*)$ .

Um escalonamento para um projeto é representado pelo vetor  $(start(1), start(2), \dots, start(n))$ , onde  $start(j)$  é a unidade de tempo em que a atividade  $j$  inicia. Assim, a unidade de tempo em que a atividade é concluída é dada por  $end(j) = start(j) + d_j$ . A unidade de tempo em que um escalonamento inicia é a mesma unidade de tempo onde começa a tarefa que inicia mais cedo  $\min\{start(j) | j \in \mathcal{J}\}$ , enquanto que a unidade de tempo em que o escalonamento termina é a mesma unidade de tempo onde termina a última atividade  $\max\{start(j) + d_j | j \in \mathcal{J}\}$ . O tempo total do escalonamento é a diferença entre as unidades de tempo em que este começa e termina.

Para ser considerado válido, um escalonamento deve satisfazer os dois tipos de restrições já mencionados:

1. A atividade  $j$  não pode iniciar antes que todas as suas predecessoras terminem,  $start(j) \geq start(h) + p_h \quad \forall h \in \mathcal{P}_j$ ;
2. As restrições dos recursos tem de ser satisfeitas, ou seja, para cada unidade de tempo  $t$ , o somatório dos recursos do tipo  $i$  requisitados por todas as atividades escalonadas não pode exceder a capacidade destes recursos,  $\sum_{j \in \mathcal{J}, start(j) \leq t < start(j) + d_j} r_{i,j} \leq R_i$ .

Para a adequação do RCPSP aos experimentos que serão conduzidos com a abordagem proposta serão consideradas algumas simplificações:

- Apenas um tipo de recurso  $q$  será utilizado,  $\mathcal{Q} = \{q\}$ ;
- As atividades serão consideradas como tarefas  $j$ , e cada  $j$  demandará uma quantidade diferente do recurso  $q$ ;
- Cada agente  $i$  será associado a uma unidade da capacidade  $R_q$  do recurso  $q$ ;
- Não serão consideradas as diferentes competências dos agentes com relação as tarefas, ou seja, cada agente  $i$  tem  $capability_i(j) = 1 \quad \forall j \in M$ ;

A figura 4.1 mostra uma instância simples do RCPSP representada através do TÆMS. Nesta instância são consideradas cinco tarefas  $A$ ,  $B$ ,  $C$ ,  $D$  e  $E$  que tem duração e demanda quanto ao número de agentes segunda a tabela 4.2. As tarefas  $C$  e  $D$  sucedem  $A$  e  $B$ , respectivamente, que por sua vez sucedem  $E$ .

Esta instância é modelada através de um TG, denominado  $T$ , que possui cinco subtarefas  $A$ ,  $B$ ,  $C$ ,  $D$  e  $E$ , cada uma dessas subtarefas tem tantos métodos quanto o número de agentes que estas tarefas demandam para sua execução (retângulos). A duração de cada tarefa está representada pela duração de cada método (número abaixo do métodos). Os atributos de qualidade e custo dos métodos foram desconsiderados. As restrições de precedência são representadas por relações de NLEs, do tipo *enable*, entre as tarefas (em cinza).

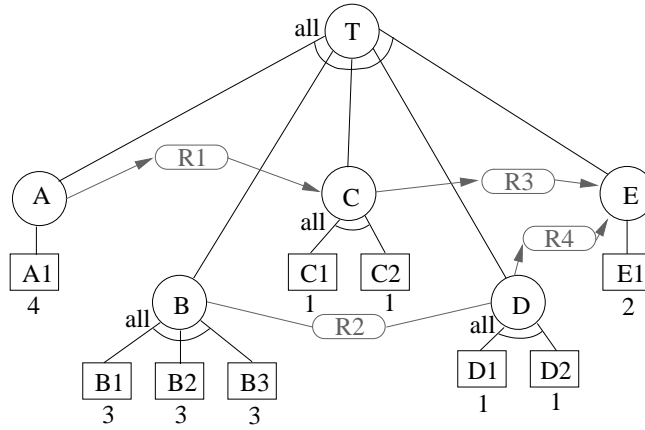


Figura 4.1: Estrutura de tarefas em TÆMS para uma instância simples do RCPSP.

#### 4.6.2 GAP

O *Generalized Assignment Problem* (GAP) é um problema de otimização onde busca-se alocar todas as tarefas à recursos, respeitando a capacidade destes recursos e minimizando o custo dessa alocação. Na seção 2.3.2 este problema já havia sido comentado e comparado com sua extensão denominada *E-GAP*, que foi apresentada também nesta seção.

O GAP é um problema geral de alocação de tarefas. Este problema foi estendido em (SCERRI et al., 2005) para modelar cenários de busca e salvamento em desastres (E-GAP), como os utilizados na Robocup Rescue. Segundo o autor, o E-GAP é um problema que representa as questões que envolvem a alocação de tarefas em ambientes dinâmicos e de larga escala. Agentes que atuam neste tipo de problema são denominados em (SCERRI et al., 2005) de *Extreme Teams*. Este cenário foi escolhido para experimentação da abordagem proposta por modelar problemas com as características que a abordagem proposta pretende tratar.

Este problema pode ser definido como segue. Seja  $\mathcal{J}$  o conjunto de tarefas e  $\mathcal{I}$  o conjunto de recursos. Cada recurso  $i \in \mathcal{I}$  tem uma capacidade  $b_i$ . Quando uma tarefa  $j \in \mathcal{J}$  aloca um recurso  $i$ , esta consome  $a_{ij}$  unidades de capacidade do recurso  $i$  e implica em um custo  $c_{ij}$ .

Considere a matriz de alocação  $Y$  onde  $y_{ij}$  é o valor da  $i$ -ésima linha e a  $j$ -ésima coluna dado pela equação 4.22.

$$y_{ij} = \begin{cases} 1 & \text{se } j \text{ está alocado por } i \\ 0 & \text{caso contrário} \end{cases} \quad (4.22)$$

O objetivo do GAP é encontrar  $Y$  que minimiza o custo  $c_{ij}$  respeitando as limitações dos recursos  $b_i$ , como mostrado na equação 4.23.

$$Y = \underset{A'}{\operatorname{argmin}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} * y'_{ij} \mid \forall i \in \mathcal{I}, \sum_{j \in \mathcal{J}} a_{ij} * y_{ij} \quad (4.23)$$

Para a adequação do GAP aos experimentos que serão conduzidos com a abordagem proposta, assume-se o seguinte:

- Os recursos terão capacidade para atender a apenas uma tarefa. Assim,  $b_i = 1$  e cada tarefa  $j$  demanda  $a_{ij} = 1$ ;
- Cada recurso  $i$  será atribuído a um agente, ou seja, um agente só pode estar alocado à uma tarefa por unidade de tempo;
- A competência  $\text{capability}_i(j)$  do agente  $i$  em relação a tarefa  $j$  é  $1 - c_{ij}$ , onde  $c_{ij}$  é o custo pago pelo recurso  $i$  quando alocado para tarefa  $j$ ;

No GAP, o custo depende do recurso em que a tarefa está alocada. Já na abordagem proposta, esta relação existe apenas com a qualidade, pois o custo está associado à tarefa diretamente, sendo independente do agente que a está executando. A qualidade será tratada pela abordagem como sendo o complemento de um do custo para o GAP.

Problemas como o GAP são modelados pelo TÆMS de forma trivial. As tarefas não se relacionam entre si de nenhuma maneira, o que significa para o TÆMS uma TG, com QAF do tipo *all* (pois todas as tarefas precisam ser executadas), e tantos métodos quantos forem as tarefas. Cada método tem qualidade igual a um, enquanto que a duração e o custo podem ser desprezados.

A abordagem proposta pode ser aplicada no E-GAP da mesma forma como foi considerada para a aplicação no GAP. As principais diferenças entre GAP e E-GAP foram comentadas na seção 2.3.2. Pretende-se utilizar o E-GAP na realização dos experimentos para que se possa comparar os resultados obtidos aqui com aqueles obtidos pelo LA-DCOP (SCERRI et al., 2005).

### 4.6.3 Considerações

Com as simplificações do RCPSP discutidas na seção 4.6.1, o modelo de especialização apresentado na seção 4.5.2 terá sua aplicação restrita ao polietismo temporal e o modelo de polimorfismo apresentado na seção 4.5.3 não será considerado. Estas simplificações não são obrigatórias para que a abordagem proposta possa ser aplicada ao RCPSP.

As questões relacionadas com a especialização e o polimorfismo serão melhor exploradas pelo GAP. Neste problema, não é considerado um escalonamento, pois as atividades não tem um tempo pré-determinado. Com isso, o modelo de estímulo, apresentado na seção 4.2, não é considerado.

O ferramental contido na abordagem em questão se propõe a resolver diversos problemas com características diferentes. Analisar os experimentos considerando todo este ferramental simultaneamente dificultaria demasiadamente este processo. Na simplificação do RCPSP algumas partes da abordagem proposta não são relevantes, o que permite que a parte que esteja sendo considerada possa ser experimentada separadamente. O que ocorre também com relação GAP. Tornando os dois problemas cenários complementares para os experimentos da abordagem proposta.

Existem significativas diferenças entre o GAP e o RCPSP:

1. Ao contrário do GAP, o RCPSP não considera a capacidade do agente de realizar uma tarefa. Ou seja, no RCPSP um agente necessário para a realização de uma tarefa tem capacidade máxima para realizá-la. No RCPSP não é possível otimizar a solução buscando maximizar a capacidade dos agentes envolvidos na realização de cada tarefa;
2. No RCPSP o tempo que o agente leva para realizar uma tarefa é previamente conhecido. É possível então definir um escalonamento para as tarefas de modo a minimizar o tempo total de execução. No GAP este tempo não é conhecido e o problema se resume a alocar agentes a tarefas sem considerar a otimização do tempo total;
3. No GAP as tarefas não são inter-relacionadas. Não existe a relação de precedência entre tarefas presente no RCPSP;

Pode-se estender os problemas discutidos na seção anterior para que contenham características dinâmicas e distribuídas:

- O processo de decidir sobre qual a melhor maneira de organizar as atividades para compor o escalonamento ou a alocação é considerado de forma centralizada. Pode-se assumir que cada recurso deve decidir se realiza ou não as tarefas em cada unidade de tempo. Assim, para que uma tarefa seja realizada, os recursos demandados por ela tem que perceber esta atividade e decidir, de forma independente, participar ou não. Atribuindo um agente a cada recurso, torna-se, tanto o RCPSP quanto o GAP, problemas distribuídos, onde não se tem controle centralizado sobre os recursos;

- Tais problemas consideram o escalonamento de um projeto ou a alocação de um recurso de maneira estática. Ao invés disso, pode-se considerar que um projeto no RCPSP deva ser escalonado periodicamente. O mesmo pode ocorrer com as alocações no GAP. O RCPSP pode ser visto como um problema dinâmico uma vez que, a cada novo escalonamento, o projeto pode ter suas definições modificadas. A alteração na precedência entre as atividades, na quantidade de recursos disponíveis, etc., faz com que o escalonamento para o projeto não seja necessariamente o mesmo em cada período. Quanto ao GAP, pode-se considerar que as tarefas tenham custos diferentes e demandam quantidades diferentes de recursos a cada período.

Este trabalho propõe uma abordagem para tratar problemas em larga escala. Os problemas descritos na seção anterior não fazem nenhuma restrição quanto ao tamanho de suas instâncias. Os cenários experimentados em (SCERRI et al., 2005), envolvendo o E-GAP, utiliza até 5000 agentes atuando em 2000 tarefas. Os casos de teste utilizados em (MERKLE; MIDDENDORF; SCHMECK, 2002) envolviam 120 tarefas e até 120 unidades de cada recurso, escalonados de forma centralizada.

Tanto o GAP quanto o RCPSP não consideram aspectos que envolvem a inter-relação entre as tarefas que podem acontecer em cenários reais. Tarefas podem estar inter-relacionadas pela possibilidade de realizar uma ou outra de forma alternativa, representados por uma restrição *XOR* entre elas. Ou ainda, duas tarefas podem ser realizadas, mas apenas uma será utilizada de fato para que o objetivo seja atingido, caracterizando um *OR*. Uma vez que o *OR* pode existir, a restrição *AND* poderia ser definida entre duas tarefas quando ambas precisam ser realizadas, independente de isso acontecer simultaneamente ou não (no E-GAP existe esta última restrição temporal). Muitas relações desse tipo podem ser definidas. O TÆMS define um grande número de relações entre tarefas, com os mais diferentes significados, através das QAFs.

Além da simulação dos dois cenários isolados, pretende-se conduzir uma série de experimentos que envolva a abordagem por completo. Estes experimentos poderão ser realizados na versão sem simplificações do RCPSP ou em outro cenário que tenha esse escopo. Além disso, pretende-se realizar modificações neste cenário completo para que relações do tipo das citadas no parágrafo anterior estejam presentes.

## 4.7 Discussão

Assim como os insetos sociais, os agentes trabalhando cooperativamente para resolver problemas em ambientes dinâmicos e de larga escala, devem coordenar-se de maneira eficiente. Padrões simples de comunicação e um número restrito de mensagens devem ser usados para garantir essa eficiência.

Considerando isso, como implementar a abordagem proposta em um sistema multiagentes? Duas possibilidades podem ser consideradas: adotar um agente onipresente e onisciente que represente o ambiente e que seja responsável por perceber as demandas da estrutura de tarefas, pela atualização do estímulo e pelo cálculo do desempenho dos agentes; ou permitir que cada agente construa seu modelo individual relacionado às demandas, ao estímulo e à seu desempenho.

Na primeira possibilidade, trata-se de um processo centralizado. A demanda das tarefas é determinada por este super-agente que percebe toda a TS e sabe que agentes estão engajados em que métodos. Todos os agentes calculam suas tendências, argüindo o super-agente sobre as demandas e os estímulos das tarefas, e comunica a este super-agente imediatamente que um método foi escolhido. O super-agente pode recalcular as demandas e aguardar por novos questionamentos de outros agentes.

Este mesmo super-agente guarda os escalonamentos intermediários e atualiza estímulos de acordo com o melhor escalonamento obtido. Além disso, este super-agente faz o cálculo do desempenho dos agentes em cada tarefa e envia à eles para que possam atualizar seus limiares internos.

Na segunda possibilidade, todos os agentes conhecem a TS e o processo não tem centralização, ou seja, os agentes mantêm um modelo próprio para determinar que agente está engajado em qual tarefa durante todo o tempo. Para isso, os agentes comunicam aos demais sua opção por um método cada vez que o processo de decisão ocorre. Os agentes podem calcular por si próprios as demandas e os estímulos, bem como guardar os escalonamentos intermediários e medir seu desempenho.

Qual a interferência do processo de centralização no sistema? Normalmente, quando isso ocorre, os agentes passam a depender das informações deste centralizador. Caso o super-agente falhe, o sistema falhará como um todo e a característica de robustez do sistemas multiagentes se perde. Como o problema em questão neste trabalho considera que os agentes, ou a comunicação entre eles, podem falhar, esta alternativa parece inadequada.

Utilizando a segunda alternativa, é preciso determinar o número e o tamanho das mensagens utilizadas para que se possa medir o desempenho geral do sistema em termos de tempo e comunicação. A princípio, a abordagem deve utilizar um número de mensagens que cresce em função do número de agentes e de métodos a serem escalonados. Cada método escolhido por um agente gera uma mensagem para cada um dos outros agentes. Além disso, quando a comunicação ou os agentes falham haverá uma inconsistência nos modelos dos demais agentes que não prejudicará a abordagem, apenas influenciando na qualidade dos resultados obtidos.

Na seção 4.5 foi considerado que os agentes não poderão abandonar um método cuja super-tarefa já tenha o número necessário de métodos realizados ou em realização ( $p = 0$ ). A falha na comunicação dos agentes poderia fazer com que um agente não saiba que o outro método relacionado está sendo executado na iteração corrente. Assim, este agente poderá abandonar a execução do método naquela iteração e retomar mais tarde, não impossibilitando a aplicação da abordagem proposta.

Outra questão importante sobre a implementação da abordagem, diz respeito ao forma que os agentes percebem os métodos. Na primeira versão da abordagem, somente as tarefas que podiam ser realizadas estimulam os agentes. Se uma tarefa dependesse que outra fosse realizada previamente, o agente só seria estimulado para realizar a segunda tarefa quando a primeira estivesse concluída. Esta hipótese tem impacto também sobre o tratamento dos recursos e das QAFs que envolvem sequência na execução das tarefas (*seq\_min*, *seq\_max*, *seq\_sum* e *seq\_last*).

É preciso estudar a implicação de considerar que os agentes tenham informação imprecisa sobre a TS, não tendo o conhecimento prévio sobre essas restrições. A princípio, o agente desconhecendo essas restrições determinaria seus estímulos e desempenho de forma equivocada e não teria meios para corrigir esta imprecisão.

## 5 RESULTADOS PRELIMINARES

Neste capítulo serão discutidos os resultados obtidos por uma versão inicial da abordagem que está sendo proposta, apresentada na seção 4.5. Primeiramente, na seção 5.1, serão apresentados os resultados obtidos com a experimentação em alguns cenários abstratos. Em seguida, na seção 5.2, serão apresentados os resultados obtidos em um problema de sequenciamento de máquinas em manufaturas. As simulações discutidas aqui foram geradas através de um simulador desenvolvido em JAVA e utilizando a API do TÆMS.

### 5.1 Cenários Abstratos

Nas próximas seções o desempenho da abordagem proposta será analisado em três diferentes cenários abstratos<sup>1</sup>, com diferentes propósitos:

**Cenário I** A idéia por trás deste cenário é validar a versão preliminar da abordagem, verificando se a dimensão estrutural da organização emerge a partir da interação entre os agentes. Em outras palavras, objetiva-se verificar a habilidade dos agentes para realizar todas as tarefas cooperativamente sem coordenação explícita entre eles;

**Cenário II** Neste cenário o ambiente é alterado dinamicamente para experimentar a capacidade de adaptação da organização emergente;

**Cenário III** O objetivo dos experimentos com este cenário é verificar o impacto do número de agentes na primeira versão da abordagem. Em ambientes dinâmicos o tamanho das organizações podem variar dinamicamente e a organização dever ser capaz de se adaptar.

Cada método em TÆMS tem probabilidades associadas a valores de qualidade, custo e duração que foram utilizadas nestes cenários. Com isso, os resultados nesta seção são apresentados e discutidos estatisticamente: os resultados apresentados aqui são médias sobre 1.000 repetições de cada experimento.

#### 5.1.1 Cenário I

A estrutura de tarefas em TÆMS da figura 5.1 foi utilizada como base para estas simulações. Esta TS abstrata pode representar, por exemplo, um problema típico de escalonamento de tarefas entre máquinas como em (CICIRELLO; SMITH, 2001). Ou ainda, esta TS pode ser relacionada com o cenário de manutenção de aeronaves discutido em (WAGNER; GURALNIK; PHELPS, 2003).

A tarefa  $T_1$  pode ser imaginada como o primeiro estágio de produção ou manutenção. Tarefas ou aeronaves dos tipos  $a$ ,  $b$  ou  $c$  demandam serem realizadas pelos agentes. Se a primeira a gerar

---

<sup>1</sup>Os cenários e os resultados apresentados e discutidos nesta seção foram publicados previamente em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005).

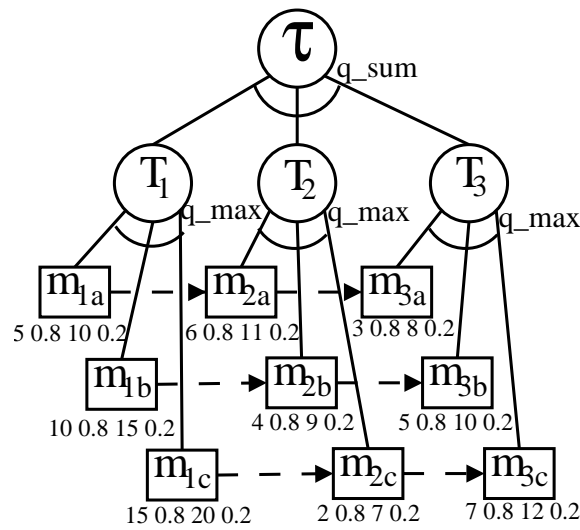


Figura 5.1: Estrutura de tarefas em TÆMS, figura originalmente publicada em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005).

demanda é tarefa do tipo *a*, por exemplo, então  $m_{1a}$  é alocado para algum agente. Desta forma, o método  $m_{2a}$  passa a estar liberado para execução, e assim por diante. Neste tipo de cenário a quantidade de NLEs do tipo *enable* faz com que o escalonamento das tarefas seja pouco flexível.

A figura 5.1 mostra somente a distribuição de probabilidade com relação a duração de cada método. Todos os métodos tem a mesma distribuição de probabilidade para o custo e a qualidade: o custo para todos os métodos é (0 0.8 1 0.2), o que significa custo 0 com probabilidade de 80% e custo 1 com probabilidade de 20%. A qualidade para todos os métodos é (5 0.8 4 0.2), o que significa qualidade 5 com probabilidade de 80% e qualidade 4 com probabilidade de 20%. A menos que seja explicitado, o limite de tempo utilizado (*deadline*) para o escalonamento é 25.

O DTC foi executado para que o escalonamento realizado pela versão preliminar da abordagem fosse comparado. A tabela 5.1 mostra o escalonamento realizado pelo DTC. Esta tabela contém os tempos de início e de fim para cada método, suas qualidades e custos. Não é mostrado na tabela que a qualidade total obtida é 14, 35, o custo total é 0, 6 e a duração total é 17, 0.

Tabela 5.1: Escalonamento utilizando o DTC					
método	início	fim	qualidade	custo	duração
$m_{1a}$	0.0	6.0	4.8	0.2	6.0
$m_{2a}$	6.0	13.0	4.8	0.2	7.0
$m_{3a}$	13.0	17.0	4.76	0.2	4.0

O DTC lida com as distribuições de probabilidade da qualidade, custo e duração advindas da semântica do TÆMS, mas não de forma probabilística. Ao invés disso, o DTC calcula a média para cada distribuição. Nestes experimentos foram usadas ambas abordagens, a probabilística e a utilizada pelo DTC. Esta última é útil para a comparação dos resultados.

Usando a versão preliminar da abordagem, na variante não probabilística, foi produzido o escalonamento mostrado na tabela 5.2. A abordagem em questão produziu 1.000 escalonamentos, como se havia sido mencionado, um para cada repetição. O escalonamento mostrado nesta tabela foi o mais frequentemente obtido, o que significa que este escalonamento foi produzido 32, 7% das vezes. A qualidade total é 14, 4, o custo total 0, 6 e a duração total 17, 0. Isto significa que o resultado mais frequentemente obtido pela abordagem é o mesmo obtido através do DTC.

A mesma TS foi escalonada utilizando o tratamento probabilístico, segunda variante, para a qualidade, custo e duração. A tabela 5.3 mostra o escalonamento mais frequentemente obtido,



produzido 22, 3% das vezes. A qualidade total é 15, o custo total 0, 0 e a duração total 24, 0. Todos os métodos escalonados na primeira variante também estão presentes nesta segunda. Contudo, por causa da variação probabilística da duração, mais métodos foram escalonados dentro do limite de tempo. No total, 41, 5% dos escalonamentos contiveram, pelo menos, os 3 métodos escalonados pelo DTC.

Tabela 5.2: Melhor escalonamento (primeira variante)

método	início	fim	qualidade	custo	duração
$m_{1a}$	0.0	6.0	4.8	0.2	6.0
$m_{2a}$	6.0	13.0	4.8	0.2	7.0
$m_{3a}$	13.0	17.0	4.8	0.2	4.0

Tabela 5.3: Melhor escalonamento (segunda variante)

método	início	fim	qualidade	custo	duração
$m_{1a}$	0.0	5.0	5.0	0.0	5.0
$m_{2a}$	5.0	11.0	5.0	0.0	6.0
$m_{3a}$	11.0	14.0	5.0	0.0	3.0
$m_{1b}$	14.0	24.0	5.0	0.0	10.0

Os resultados gerais da abordagem não são tão bons quanto os obtidos pelo DTC. Contudo, a abordagem proposta é voltada para ambiente dinâmicos. Isto significa que os agentes através dessa abordagem podem se adaptar as mudanças no ambiente sem a necessidade de comunicação ou coordenação explícitas. Ambientes deste tipo são discutidos nas próximas seções.

### 5.1.2 Cenário II

Com o objetivo de medir o desempenho da versão preliminar da abordagem em ambientes dinâmicos, foram utilizadas quatro TSs aparecendo aleatoriamente com a mesma probabilidade. A primeira TS ( $TS_1$ ) é a mostrada na figura 5.1. As outras três variações são: a primeira delas não tem NLEs do tipo *enable* entre as tarefas ( $TS_2$ ); a segunda não possui a tarefa  $T_3$  ( $TS_3$ ); e a última teve seu limite de tempo alterado para 30 ( $TS_4$ ). Além disso, as distribuições de qualidade foram alteradas: na  $TS_2$  passou a ser (15 0.8 10 0.2), na  $TS_3$  (50 0.8 40 0.2), e na  $TS_4$  (100 0.8 90 0.2).

Somente TSs individuais podem ser escalonadas pelo DTC. Com isso, neste experimento, não foi possível comparar os resultados obtidos pela abordagem com os resultados do DTC.

A figura 5.2 mostra as mudanças na TS ocorrendo e os agentes se adaptando a elas. Como dito, as TSs são alteradas aleatoriamente. A sombra cinza no gráfico mostra qual a TS que está sendo escalonada no momento. Quando a sombra vai até o 300 significa que a  $TS_4$  está sendo executada. Quando a sombra vai até 100, 50, e 15, as TSs executadas no momento são  $TS_3$ ,  $TS_2$ , e  $TS_1$  respectivamente.

Idealmente, cada vez que a TS é trocada, os agentes devem se adaptar e a qualidade deveria mudar instantaneamente. Contudo, os agentes precisam de um determinado tempo para adaptar seu estímulo e limiar interno, o que faz com que um pequeno atraso seja percebido. A curva da qualidade não coincide com a sombra.

Devido aos aspectos probabilísticos da abordagem em questão, a curva do gráfico não é constante. Existe uma variação na qualidade dos escalonamentos para a mesma TS. Contudo, a figura 5.2 mostra que quando a TS muda, muda também a qualidade total associada a ela.

Estes resultados mostraram que modificando a TS dinamicamente o desempenho dos agentes é pouco perturbado, mantendo a qualidade dos escalonamentos produzidos. Cada vez que a estrutura muda, os agentes se adaptam a nova situação.

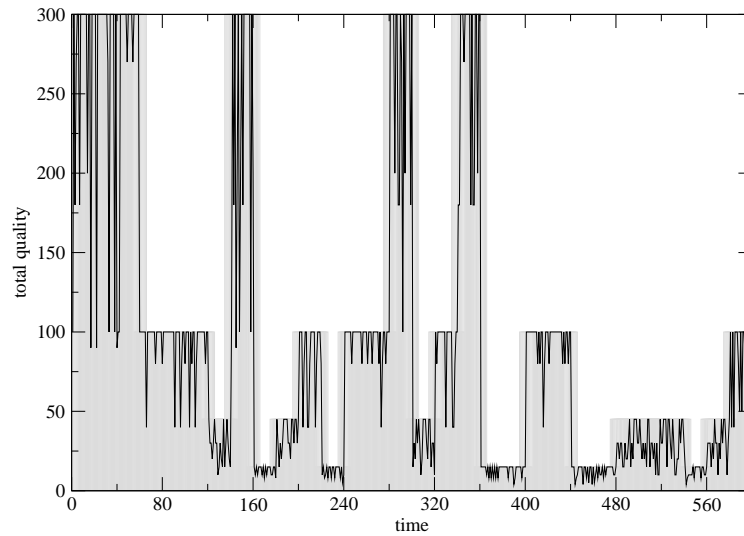


Figura 5.2: Mudança na qualidade em relação ao tempo (linha preta); As mudanças no ambiente são representadas pela sobra cinza, figura originalmente publicada em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005).

### 5.1.3 Cenário III

Com o mesmo objetivo descrito acima, o número de agentes disponíveis para a execução das tarefas da TS serão variados. Neste cenário também foi utilizada como base a TS da figura 5.1. Desta vez, todas as TS foram organizadas com sub-tarefas de um novo TG com QAF igual a *sum*.

Esta nova TS tem três vezes mais métodos que a estrutura base, ou seja, 27 métodos. Assim, foram variados o número de agentes entre 1 e 27. Para lidar com a natureza probabilística do problema, foram realizadas 100 repetições para cada vez que o número de agentes é modificado, totalizando 2700 repetições.

A figura 5.3 mostra a influência do número de agentes sobre o número de métodos escalonados e também sobre a qualidade. Na TS adotada, 9 dos 27 métodos não possuem NLEs do tipo *enable*. A medida que cresce o número de agentes, o número de métodos sem restrições executados aumenta. Quando o número de agentes é igual a 9, o número de métodos realizados se estabiliza pois, mesmo que mais agentes estejam disponíveis, estes não podem realizar métodos que não estiverem habilitados. O mesmo raciocínio se aplica a qualidade: a melhor qualidade é obtida após tal estabilização. A qualidade mais alta possível, por volta de 40, é obtida com a utilização de 9 agentes.

## 5.2 Cenários Reais

Nesta seção será discutida a aplicação da versão preliminar da abordagem proposta em um problema da área de pesquisa operacional, intitulado *Machine Sequencing Problem* (MSP) (STRUSEVICH, 1955), que é parte da classe de problemas de escalonamento em manufaturas<sup>2</sup>.

No MSP,  $n$  peças de um produto chegam em uma linha de produção e devem ser processadas por  $m$  máquinas em uma sequência específica. Esta sequência é estabelecida por restrições

<sup>2</sup>Os cenários e os resultados apresentados e discutidos nesta seção compõe um artigo recentemente submetido a uma conferência e que ainda está sob revisão (FERREIRA JR; BAZZAN, 2006).

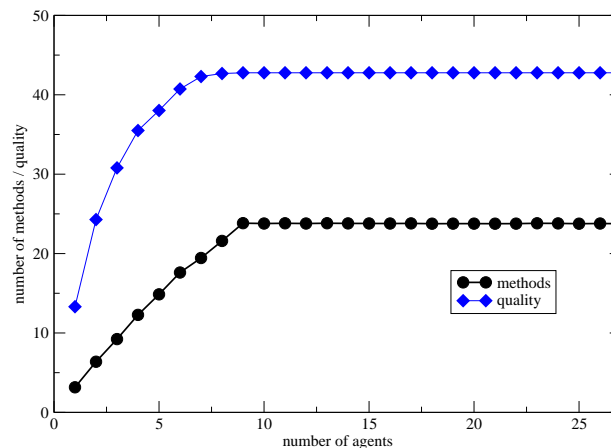


Figura 5.3: Número de métodos e qualidade para um número variado de agentes, figura originalmente publicada em (FERREIRA JR.; OLIVEIRA; BAZZAN, 2005)

Tabela 5.4: Sequência de Restrições das Peças

Peça	Sequência das Máquinas
1	A - D - C - B
2	B - A - C - D

tecnológicas. cada máquina gasta um determinado intervalo de tempo processando cada peça. A sequência pela qual as  $n$  peças são processadas por cada máquina  $i$  deve ser escolhida com o objetivo de minimizar o tempo total de produção.

A figura 5.4 traz uma instância do MSP modelada em TÆMS. O objetivo da TS é produzir um artefato que é composto por duas peças. Nesta abordagem, os agentes têm o papel de um trabalhador, selecionando uma peça e passando por cada uma das máquinas requeridas para processá-la. Se todas as máquinas estão ocupadas quando a peça chega, o agente precisa ficar aguardando até que a máquina se torne disponível.

Além disso, não há informação na TS sobre o que cada agente deve fazer. Estes são livres para escolher qualquer tarefa que esteja disponível e realizá-la. A abordagem proposta permite ao agente realizar as tarefas de acordo com as restrições sem qualquer coordenação explícita entre os agentes.

Na figura 5.4 são detalhadas apenas a estrutura de tarefas para a máquina A processando a peça 1 (*Machine\_A\_Part\_1*). Tarefas adicionais são necessárias (*Machine\_B\_Part\_1*, *Machine\_C\_Part\_1* e *Machine\_D\_Part\_1*) mas não são mostradas para manter a clareza da figura. Contudo, todas são bastante similares. Para simplificar, tarefas que não são detalhadas estão em cinza na figura 5.4. Além disso, cada artefato deve ter  $n$  sub-tarefas, uma para cada peça. Estas sub-tarefas são relacionadas pela QAF *all*, significando que as duas devem ser completadas. Qualquer instância de um MSP pode ser modelada desta forma, independente do número de peças, restrições e recursos.

A tabela 5.4 mostra a sequência de processamento de cada uma das peças. Por exemplo, a peça 1 deve ser processada pela máquina A, então a D e a C e finalmente a B. Esta sequência não está representada na figura 5.4 para mantê-la simples. Essas restrições são modeladas através de NLEs do tipo *enable* em cada tarefa das peças.

Usando a linguagem TÆMS, cada máquina é modelada como um recurso consumível que só pode ser utilizado por um único agente a cada vez. Quando um agente precisa usar uma máquina, ele primeiro a requisita, tornando-a inacessível para outro agente (relacionamento AI-

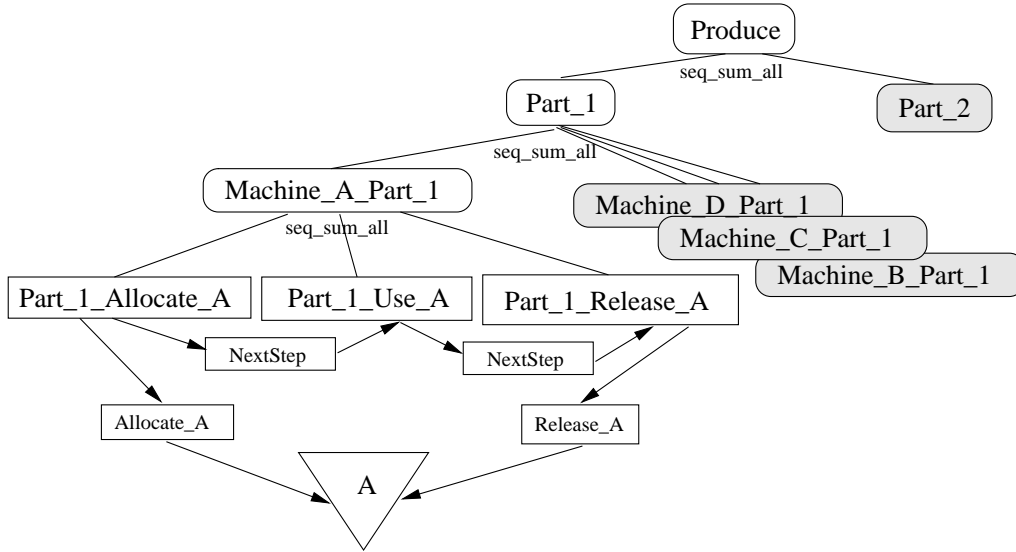


Figura 5.4: Estrutura de tarefas em TÆMS para o MSP (as sub-tarefas para as máquinas B, C, D com a peça 1 e todas as sub-tarefas para a peça 2 foram omitidas).

Tabela 5.5: Tempo de Processamento das Peças para cada Máquina

Peça	A	B	C	D
1	6	3	1	1
2	5	5	6	2

*locate\_A*). Quando o agente termina a execução da tarefa, ele libera a máquina (relacionamento *Release\_A*). Existem também NLEs do tipo *enable* (*NextStep*) relacionadas aos métodos *allocate* (e.g. *Part\_1\_Allocate\_A*), *use* (e.g. *Part\_1\_Use\_A*) and *release* (e.g. *Part\_1\_Release\_A*) que determinam a sequência desse processo.

A tabela 5.5 mostra a quantidade de tempo gasto por cada máquina no processamento de cada peça. Esta característica é modelada como duração na TS. Para simplificar, custo e qualidade não são considerados, foi adotado zero para ambos.

No MSP, uma quantidade de tempo específica pode ser necessário para configurar as máquinas quando duas peças diferentes serão processadas na mesma máquina em sequência.

A figura 5.5 mostra o modelo que envolve o tempo de configuração para a peça 1 utilizando a máquina A. Mais uma vez, para manter a clareza da figura o modelo completo do MSP não é mostrado. O modelo mostrado na figura 5.5 deve ser combinado com o da figura 5.4 para se tornar completo.

O método de alocação da máquina só pode ser realizado depois que a tarefa de configuração for executada. Dois recursos consumíveis foram utilizados para modelar o estado da configuração de cada máquina. Quando uma máquina é liberada, o recurso relacionado se torna disponível. Existe um relacionamento entre a tarefa de liberação e o recurso relacionado (e.g. the *Configure\_1* entre a tarefa *Part\_1\_Release\_A* e o recurso *A\_1*).

Um método de reconfiguração (*Reconfigure\_A\_Part\_1*) é escolhido (e.g. *Machine\_A\_2\_to\_1* ou *Machine\_A\_1\_to\_1*) pelo agente de acordo com o estado do recurso. Somente o método que tem recurso disponível será executado. Os métodos realizados consomem os recursos através de um relacionamento (*Configure\_A*). Se a configuração for necessária, ela consumirá 10 unidades de tempo.

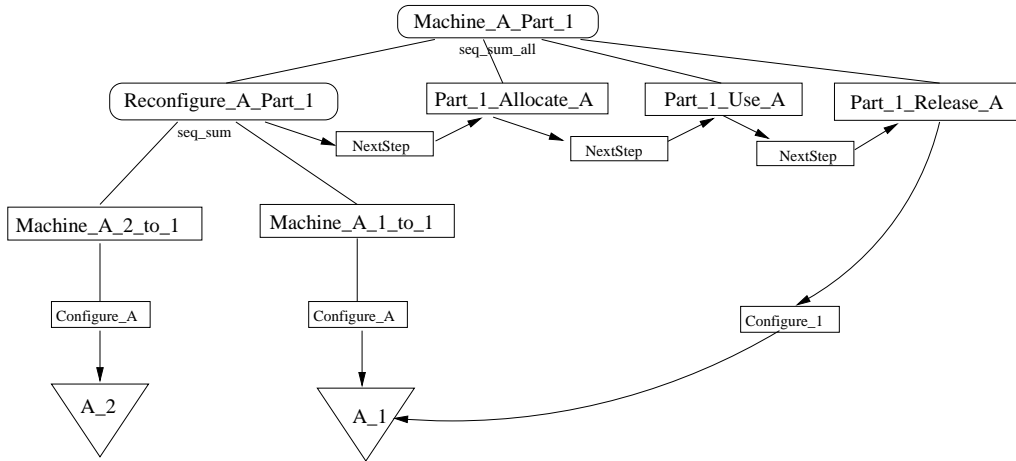


Figura 5.5: Estrutura de tarefas para a peça 1 e a máquina A (todas as demais foram omitidas) com a configuração da máquinas.

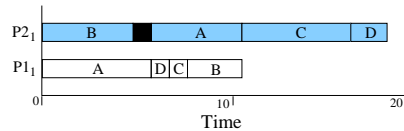


Figura 5.6: Melhor escalonamento possível para produzir uma peça sem tempo de configuração.

### 5.2.1 Cenário Simplificado

Neste cenário simples, são considerados dois agentes para produzir um artefato (composto pelas duas peças descritas anteriormente) e as máquinas não necessitam serem reconfiguradas (não é necessário tempo para configuração, neste caso). O Objetivo deste experimento é verificar a habilidade dos agentes em realizar todas as tarefas de forma cooperativa.

Para validação da abordagem, o resultado obtido pela primeira versão da abordagem é comparado com a solução ótima obtida pelo algoritmo de Akers e Friedman (AKERS; FRIEDMAN, 1955). Utilizando este algoritmo, é preciso conhecer que peça deve ser processada primeiro em cada máquina. Neste cenário com quatro máquinas, existem  $2^4$  escalonamentos possíveis. Os tempos de processamento são mostrados na tabela 5.5. Depois de elaborar os 16 escalonamentos, elimina-se aqueles que não são tecnicamente possíveis. Os escalonamentos restantes são examinados para que aqueles que não são ótimos sejam removidos.

A figura 5.6 mostra o melhor escalonamento possível, cujo tempo total é de 19 unidades. Este é o mesmo escalonamento obtido pela versão preliminar da abordagem proposta, pois os agentes não permanecem inativos quando existem tarefas à serem executadas. Estes agentes, como os insetos sociais, tentam realizar as tarefas o mais cedo possível. Nesta figura é possível visualizar em que intervalo de tempo cada peça foi processada. Os retângulos brancos representam a peça 1 e os cinzas representam a peça 2. Os retângulos pretos representam a quantidade de tempo uma peça não está sendo processada por nenhuma máquina durante o tempo total de processamento.

No cenário discutido aqui, onde a melhor solução pode ser determinada manualmente, a abordagem foi eficiente. O mesmo comportamento é esperado em outros cenários simples: a versão preliminar da abordagem deve obter o escalonamento ótimo ou, no pior caso, chegar a um caso perto do ótimo.

### 5.2.2 Cenário com Mudanças Dinâmicas

Neste cenário, as mudanças dinâmicas no ambiente são experimentadas. Inicialmente, o objetivo é produzir cinco artefatos (cinco peças 1 e cinco peças 2) sem tempo para configuração. Em seguida, a TS é alterada dinamicamente para que mais cinco artefatos sejam produzidos (cinco peças 1 e cinco peças 2), mas agora necessitando do tempo de configuração. Aqui se está interessado em permitir à abordagem decidir sobre a ordem de processamento em ambos estágios, sem coordenação ou comunicação explícitas.

Em nenhum dos estágios mencionados acima se pode aplicar o algoritmo de Akers and Friedman pois o número de soluções possíveis cresce exponencialmente com o número de peças. Assim, para comparar o desempenho da solução obtida, foi desenvolvido um algoritmo guloso utilizando duas estratégias diferentes: produzir cada artefato o mais cedo possível; produzir todas as peças 1 e, depois disso, produzir todas as peças 2.

A figura 5.7 mostra o tempo de produção de cada um dos cinco artefatos (dez peças) para a primeira estratégia gulosa (a), para a segunda estratégia gulosa (b) e para a abordagem proposta (c). Como pode ser observado, para este caso, ambas estratégias gulosas obtiveram resultados piores que a abordagem proposta. A organização baseada nos insetos sociais obteve uma solução melhor sem que uma estratégia previa fosse definida para este cenário especificamente. A dimensão estrutural da organização emergiu da simples interação entre os agentes.

Depois de produzir os cinco artefatos iniciais, onde não há tempo para reconfiguração, os estímulos das tarefas e os limiares internos dos agentes foram alterados de acordo com o as ações tomadas durante o escalonamento. As tendências dos agentes para executar cada método foi modificada de acordo com as características do problema. A decisão sobre quais tarefas executar e em que ordem, para que o objetivo do sistema seja atingido, foi guiada por esta tendência.

Neste ponto, é introduzida a modificação da demanda: mais cinco artefatos são demandados e o tempo para configuração da máquina se faz necessário. Com isso, os agentes devem manter o número de configurações o menor possível. Uma maneira simples de resolver este problema é acumular as demandas de artefatos e produzi-los em lotes de peças diferentes. Por exemplo, produzir um lote de peças 1 primeiro e, em seguida, produzir o lote de peças 2. A segunda estratégia gulosa é propícia quando se considera este tempo necessário para configuração das máquinas (a primeira estratégia não é razoável pois as peças 1 e 2 são intercaladas resultando em um número desnecessário de configurações).

Agora, o desempenho da solução depende diretamente da produção dos lotes de peças. O processo de decisão de acordo com a tendência computada anteriormente deve ser adaptado para lidar com esta nova característica o mais cedo possível. Esta adaptação ocorre, como poderá ser visto a seguir.

A figura 5.8 mostra o tempo de produção das peças com o tempo de configuração (representado por uma máquina hipotética  $X$ ). Pode-se observar que a abordagem proposta (a) cria os lotes de produção, reduzindo o tempo total necessário para a configuração. A produção de peças diferentes se sobrepõe duas vezes: na primeira vez, porque os agentes precisam aprender com os primeiros resultados para que se adaptem (peça  $P2_1$  como todas as peças  $P1$ ); a segunda sobreposição ocorre pois não existem mais peças 1 para serem processadas (peça  $P2_2$ ).

A abordagem proposta gera quatro reconfigurações a mais que a estratégia gulosa (b). A máquina  $B$  é usada para processar uma peça 1 no início do escalonamento, causando mais três reconfigurações. Contudo, mesmo com o dobro de reconfigurações, a versão preliminar da abordagem atingiu um resultado próximo ao obtido pela estratégia gulosa. Isto mostra que os agentes reagiram ao custo de configuração e adaptaram a organização para melhorar o processo de escalonamento.

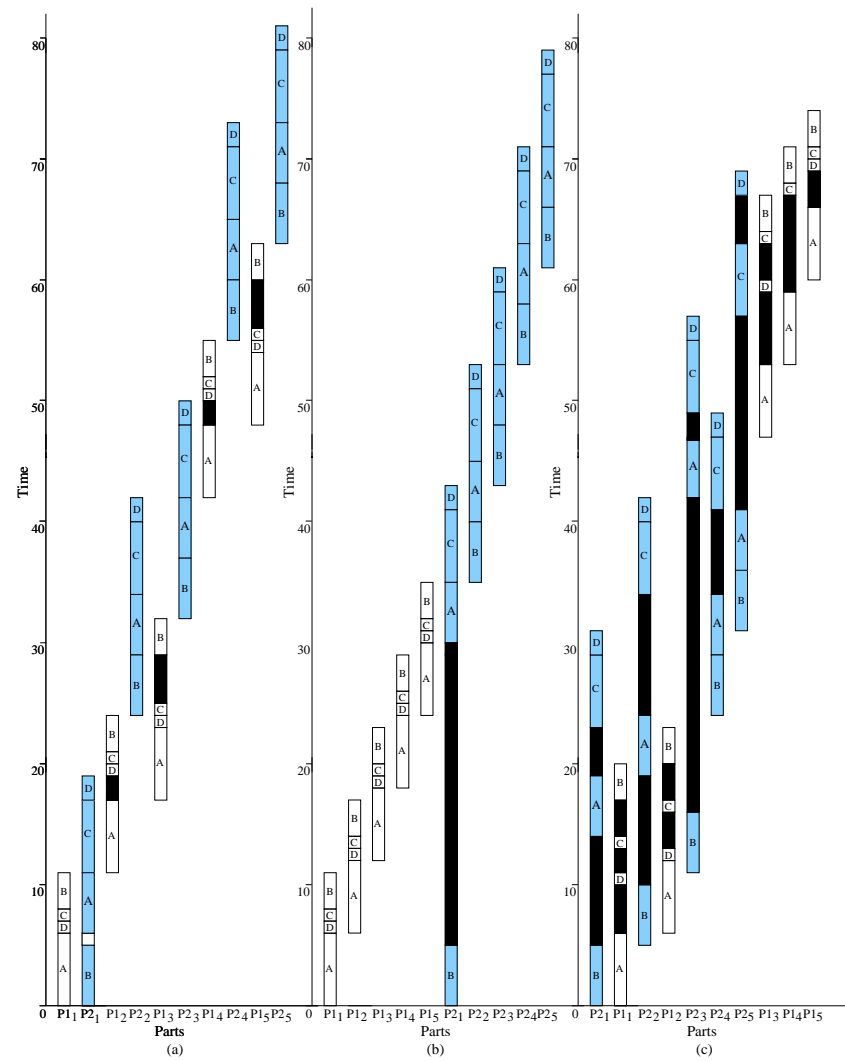


Figura 5.7: Produção de cinco artefatos sem tempo de configuração.

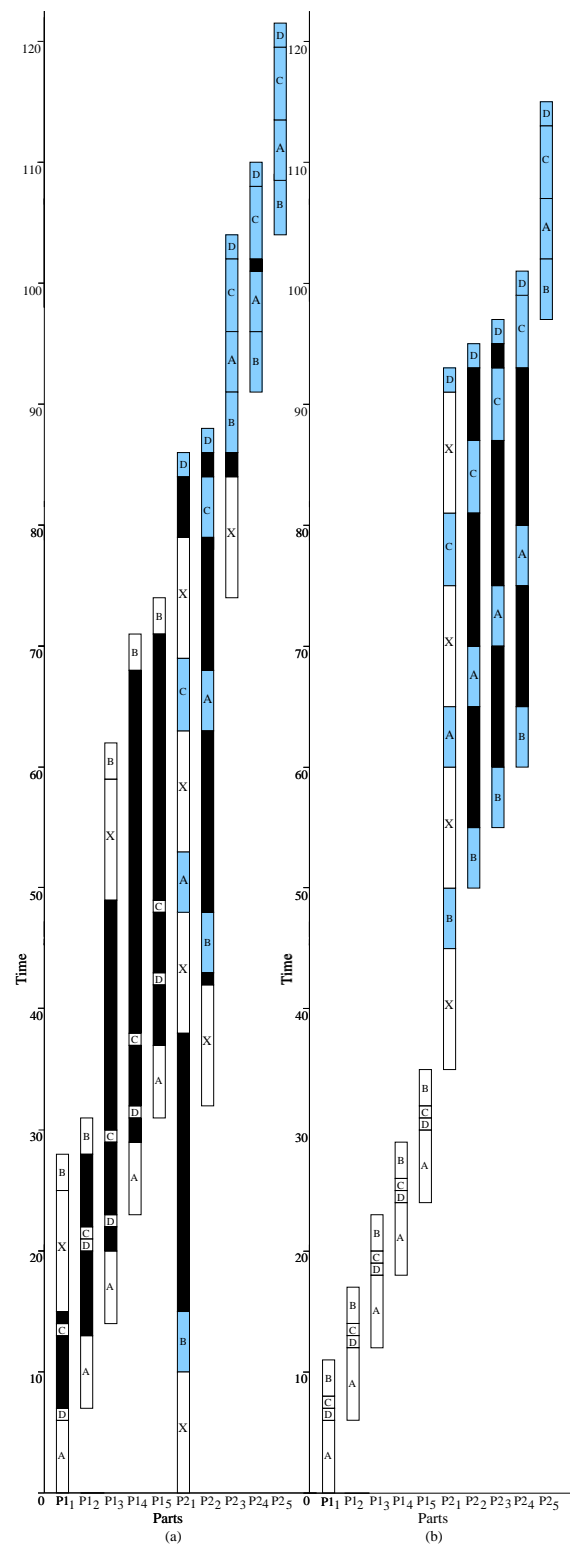


Figura 5.8: Produção de cinco artefatos com tempo de configuração.



## 6 CONCLUSÕES PRELIMINARES

Este trabalho propõe uma abordagem para a emergência da dimensão estrutural de uma organização para sistemas multiagentes com o objetivo de obter a coordenação destes agentes em ambientes dinâmicos e de larga escala. Esta abordagem baseia-se na alocação e re-alocação de tarefas entre os agentes.

Foi adotado o paradigma baseado em colônias de insetos sociais, onde existem plenas evidências de sucesso ecológico a despeito da não existência de coordenação explícita. Estes insetos se adaptam às mudanças no ambiente e às necessidades da colônia usando os mecanismos de divisão de trabalho explicados aqui.

Uma versão preliminar da abordagem foi proposta anteriormente contendo algumas das idéias básicas da abordagem proposta por este trabalho. Esta primeira versão possui várias limitações e não é capaz de tratar todo o escopo dos problemas agora considerados.

A abordagem proposta aqui faz com que os agentes decidam probabilisticamente se engajar ou não em determinada tarefa. Esta probabilidade é influenciada pelo estímulo da tarefa, por sua demanda em relação às outras tarefas e por um limiar interno do agente. O estímulo é atualizado de acordo com a eficiência dos escalonamentos obtidos, fazendo com que os agentes busquem concluir as tarefas sempre o mais cedo possível. O limiar interno é determinado pela competência do agente em realizar uma tarefa e é atualizado de acordo com seu desempenho na realização de cada tarefa. Pretende-se com isso que os agentes mais competentes para cada tarefa se engajem na execução destas..

Os problemas de pesquisa operacional mencionados servirão como cenários para a experimentação da abordagem. Tais problemas se complementam para cobrir o escopo de cenários em que a abordagem proposta pode atuar, permitindo que seu desempenho seja analisado detalhadamente e comparado com outras abordagens. Pretende-se mostrar que o paradigma dos insetos sociais pode ser utilizado para coordenar sistemas multiagentes quando atuando em ambientes dinâmicos e de larga escala.

Nos cenários discutidos nos resultados preliminares os agentes se adaptaram bem para completar todas as tarefas, confirmando a idéia de que os sistemas multiagentes podem se auto-organizar como os insetos sociais, obtendo o mesmo sucesso que estes quando atuando cooperativamente para a resolução de problemas distribuídos.

A seguir serão apresentadas as atividades que serão realizadas para dar continuidade a este trabalho.

### 6.1 Atividades Futuras

Pretende-se realizar as seguintes atividades:

1. Implementar a abordagem proposta em um ambiente de simulação para realizar os experimentos;

Pretende-se utilizar a linguagem JAVA para desenvolver um simulador que facilite a execução dos experimentos que serão descritos a seguir e análise de seus resultados.

2. Experimentar a abordagem proposta no cenário de RCPSP, analisando e comparando os resultados obtidos;

A tendência de um agente se engajar em uma tarefa é determinado pelo estímulo e o limiar interno, combinados com a demanda das tarefas determinada pelas QAFs da TS. A constante  $\sigma$  determina o peso desses dois fatores que compõem a equação 4.7. Pretende-se verificar o impacto da variação desse peso para medir a influência de cada um dos fatores mencionados. Para tanto, serão realizados experimentos com  $\sigma = \{0, 0.3, 0.5, 0.8, 1\}$ .

No RCPSP, os métodos são associados a super-tarefas por QAFs do tipo *max* que, por sua vez, são associadas por este mesmo tipo de QAF a um TG. As equações 4.8 e 4.9 serão utilizadas para determinar a demanda de um método em função das QAFs que o relaciona com os demais. Nesse caso é preciso medir o impacto da taxa de desconto  $\psi$  que determina o decaimento da influência do relacionamento a medida que o métodos se distanciam na TS. Para isso, serão realizados experimentos com  $\psi = \{0.1, 0.3, 0.5\}$ .

Para se obter um escalonamento melhor a cada rodada, o modelo de estímulo será utilizado. A equação 4.10 será utilizada para determinar a atualização do estímulo. É preciso medir o impacto da taxa de decaimento do estímulo em relação ao tempo, determinada pela constante  $\delta$ . Para experimentar isso, serão utilizados os seguintes valores  $\delta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ .

Outro aspecto importante relacionado a melhoria dos escalonamentos é a quantidade destes que serão armazenados para que o melhor seja utilizado para atualização dos estímulos. A constante  $\mu$  determina esse número de rodadas que se dá entre as atualizações. Serão experimentados os seguintes valores  $\mu = \{5, 10, 15\}$ .

Pretende-se executar 1000 rodadas e utilizar o escalonamento resultante para comparação com outra abordagem de destaque na resolução do RCPSP. Optou-se por utilizar o AS-RCPSP, apresentado em (MERKLE; MIDDENDORF; SCHMECK, 2002), pela eficiência demonstrada por essa abordagem e pela facilidade em realizar simulações comparativas com o AS-RCPSP dada a colaboração estabelecida entre um dos autores desta abordagem e o autor do trabalho em questão.

Serão também inseridas falhas nos agentes e na percepção dos agentes quanto as tarefas que outros agentes estão engajados. Os resultados obtidos mediante estas falhas serão comparados com os resultados sem considerar tais falhas.

3. Experimentar a abordagem proposta no cenário de E-GAP, analisando e comparando os resultados obtidos;

Algumas das variáveis consideradas nesse cenário já foram experimentadas no RCPSP, pretende-se então adotar os melhores valores obtidos para este cenário nestes novos experimentos.

No E-GAP, o desempenho dos agentes na realização das tarefas é medido apenas em função da qualidade da execução do método, ou seja, os custos são considerados iguais a zero. O modelo de especialização é então usado para motivar agentes mais competentes na execução de terminadas tarefas as escolham.

A equação 4.19 utiliza a constante  $\xi$  para determinar quanto o limiar interno do agente aumenta para determinada tarefa quando o desempenho total dos agentes foi o melhor, e o agente atuava nessa tarefa especificamente. A constante  $\rho$  na equação 4.20 tem o papel de diminuir o limiar interno para as atividades que o agente não estava executando nessa alocação de melhor desempenho. Serão experimentadas as seguintes combinações  $(\xi, \rho) = \{(0.1, 0.01), (0.05, 0.01), (0.02, 0.01)\}$ .

Segundo o modelo de polimorfismo, os agentes terão um limiar inicial maior para tarefas nas quais estes forem mais competentes. A equação 4.21 utiliza a constante  $\lambda$  para determinar quanto da competência do agente é utilizada para determinar seu limiar interno inicial. Para experimentar o impacto desse valor na especialização do agente serão utilizados nos experimentos os seguintes valores  $\lambda = \{0, 0.5, 1\}$ .

Outro aspecto a ser experimentado é o impacto do polietismo temporal na especialização. Para isso, as equações 4.19 e 4.20 utilizam a constante  $\rho$  que determina se este tipo de polietismo será considerado ou não. Serão considerados os seguintes valores  $\rho = \{0, 1\}$

Serão utilizadas 1000 rodadas e a qualidade de cada alocação será considerada para comparação com uma abordagem bem sucedida para o E-GAP. Para a realização desta comparação será utilizado o LA-DCOP, proposto em (SCERRI et al., 2005), a qual é a única abordagem para o GAP (E-GAP) que considera este problema distribuído, dinâmico e em larga escala.

Da mesma forma que no RCPSP, neste cenário também serão inseridas falhas para que se possa analisar o comportamento da abordagem.

4. Experimentar a abordagem proposta em uma combinação dos cenários anteriores para validá-la;

É preciso combinar os cenários acima para que abordagem tenha todo seu ferramental simulado simultaneamente. Os melhores valores para os parâmetros dos cenários RCPSP e E-GAP serão utilizados para realizar esse experimento. Aqui não existem abordagens que possam ser utilizadas para comparação dos resultados obtidos.

De qualquer forma, é preciso experimentar variações nesse cenário que permitam medir se as NLEs não restritivas estão sendo devidamente consideradas pela abordagem. Para isso é preciso medir o impacto das taxas de desconto  $\vartheta$  e  $\varsigma$  utilizadas para diminuir a influência dos outros métodos relacionados por NLEs deste tipo. Para tanto serão adotados os valores  $(\vartheta, \varsigma) = \{(0.1, 0.1), (0.3, 0.3), (0.5, 0.5)\}$ .

Além disso, alguns tipos de QAFs não foram experimentados para determinar o valor da demanda na equação 4.8. É preciso simular cenários onde as QAFs *exactly\_one* e *sum* sejam utilizadas. Com isso, se poderá variar a taxa de desconto  $\omega$  utilizada para diminuir o peso da QAF quando a relação é não restritiva. Serão realizados experimentos com  $\omega = \{0.1, 0.3, 0.5\}$ .

Pretende-se ainda simular com este cenário a questão levantada na seção 4.6.3 referente a percepção incompleta dos agentes em relação aos NLEs restritivos. Neste caso serão inseridas falhas na percepção destas restrições e medidos os resultados obtidos em relação ao mesmo cenário sem tais falhas.

5. Estudar as características das organizações de agentes obtidas nos cenários experimentados;

Pretende-se analisar a formação de grupos de agentes que se especializam em métodos, e como estes agentes trocam de grupo durante os experimentos. Serão feitas análises estatísticas da permanência dos agentes engajados na execução de cada tarefa e das taxas de trocas. Objetiva-se identificar se a especialização é capaz de gerar organizações razoavelmente estáveis.

6. Redação da tese;

7. Artigos previstos;

São previstos três artigos que conterão os resultados obtidos nas próximas etapas:

A os experimentos com o RCPSP devem compor um artigo para o *Sixth International Workshop on Ant Colony Optimization and Swarm Intelligence (Ants 2007)* deta-

lhando a abordagem proposta sob a ótica da inteligência coletiva, demonstrando os experimentos com a coordenação de agentes utilizando o RCPSP;

**B** os experimentos com o E-GAP devem compor um artigo para o *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007)* detalhando a abordagem proposta sob a ótica dos sistemas multiagentes, demonstrando os experimentos com a coordenação de agentes utilizando o E-GAP;

**C** os experimentos com a abordagem completa e o papel da estrutura organizacional emergente para a organização de agentes devem compor um artigo mais completo para o *Journal of Autonomous Agents and Multi-Agent Systems*.

8. Defesa da tese.

## 6.2 Cronograma

As atividades na seção anterior serão realizadas no seguinte período:

2006												
Etapas	1	2	3	4	5	6	7	8	9	10	11	12
1	•	•	•	•								
2					•	•	•					
3								•	•	•		
4											•	•
5							•			•		
6												
7										<b>B</b>		
8												

2007						
Etapas	1	2	3	4	5	6
1						
2						
3						
4	•	•				
5		•				
6			•	•	•	•
7			<b>A</b>			<b>C</b>
8						•

## REFERÊNCIAS

- AKERS, S.; FRIEDMAN, J. A non-numerical approach to production scheduling problems. In: OPERATIONS RESEARCH, 1955. **Annals...** [S.l.: s.n.], 1955. n.3, p.429–442.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. Inspiration for Optimization From Social Insect Behaviour. **Nature**, [S.l.], p.39–42, July 2000.
- BONABEAU, E.; THERAULAZ, G.; DORIGO, M. **Swarm Intelligence**: from natural to artificial systems. [S.l.]: Oxford Univ Press, 1999.
- CAMPOS, M.; BONABEAU, E.; THERAULAZ, G.; DENEUBOURG, J. Dynamic Scheduling and Division of Labor in Social Insects. In: ADAPTIVE BEHAVIOR, 2001. **Anais...** [S.l.: s.n.], 2001. v.8–2, p.83–96.
- CICIRELLO, V.; SMITH, S. Improved Routing Wasps for Distributed Factory Control. In: WORKSHOP ON ARTIFICIAL INTELLIGENCE AND MANUFACTURING: NEW AI PARADIGMS FOR MANUFACTURING, 2001. **Proceedings...** [S.l.: s.n.], 2001.
- CICIRELLO, V.; SMITH, S. Improved Routing Wasps for Distributed Factory Control. **Journal of Autonomous Agents and Multi-Agent Systems**, [S.l.], v.8, n.3, p.237–266, May 2004.
- DECKER, K. S.; LESSER, V. R. Quantitative Modeling of Complex Computational Task Environments. In: INTERNATIONAL WORKSHOP ON DISTRIBUTED ARTIFICIAL INTELLIGENCE, 12., 1993, Hidden Valley, Pennsylvania. **Proceedings...** [S.l.: s.n.], 1993. p.67–82.
- DECKER, K. S.; LESSER, V. R. Designing a Family of Coordination Algorithms. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, 1., 1995, San Francisco, CA. **Proceedings...** [S.l.]: AAAI Press, 1995.
- DORIGO, M. **Optimization, Learning and Natural Algorithms**. 1992. Tese (Doutorado em Ciência da Computação) — Politecnico di Milano.
- DORIGO, M.; DI CARO, G.; GAMBARDELLA, L. Ant Colony Optimization: A new meta-heuristic. In: CONGRESS ON EVOLUTIONARY COMPUTATION, 1999, Mayflower Hotel, Washington D.C., USA. **Proceedings...** IEEE Press, 1999. v.2, p.1470–1477.
- FERREIRA JR., P.; BAZZAN, A. Distributed Meeting Schedule through Cooperative Mediation: analysing optapo's performance in a complex scenario. In: SIXTH INTERNATIONAL WORKSHOP ON DISTRIBUTED CONSTRAINT REASONING (DCR2005) - NINETEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI2005), 2005. **Proceedings...** [S.l.: s.n.], 2005. p.101–113.

FERREIRA JR, P.; BAZZAN, A. L. Distributed Manufacturing Scheduling through Self-Organized Multiagent Systems. In: FIFTH INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS (AAMAS 06), 2006, Hakodate, Japan. **Anais...** [S.l.: s.n.], 2006. submitted.

FERREIRA JR., P. R.; OLIVEIRA, D.; BAZZAN, A. C. A Swarm Based Approach to Adapt the Structural Dimension of Agents' Organizations. **Journal of Brazilian Computer Society - JBCS - Special Issue on Agents Organizations**, [S.l.], v.11, n.1, p.63–73, July 2005.

GORDON, D. The organization of work in social insect colonies. **Nature**, [S.l.], v.380, p.121–124, 1996.

GORDON, D. **Formigas em ação**: como se organiza uma sociedade de insetos. [S.l.]: Jorge Zahar Editor, 2002.

HAMILTON, W. D. The genetic evolution of social behavior. **J Theoretical Biology**, [S.l.], v.7, p.18, 1964.

HORLING, B.; BENYO, B.; LESSER, V. Using Self-Diagnosis to Adapt Organizational Structures. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 5., 2001, Montreal. **Proceedings...** New York: ACM Press, 2001. p.529–536.

HORLING, B.; LESSER, V. A Survey of Multi-Agent Organizational Paradigms. **Knowledge Engineering Review**, [S.l.], v.19, n.4, p.281–316, december 2004.

HORLING, B.; LESSER, V.; VINCENT, R.; BAZZAN, A. L. C.; XUAN, P. Diagnosis as an Integral Part of Multi-Agent Adaptability. In: DARPA INFORMATION SURVIVABILITY CONFERENCE AND EXPOSITION, 2000, South Carolina. **Anais...** IEEE Computer Society, 2000. p.211–219. see also UMASS CSTR 1999-03.

HORLING, B.; LESSER, V.; VINCENT, R.; WAGNER, T.; RAJA, A.; ZHANG, S.; DECKER, K.; GARVEY, A. **The TAEMS White Paper**. 1999.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 16., 2002. **Proceedings...** Springer-Verlag, 2002. p.118–128.

ISHIDA, T.; YOKOO, M.; GASSER, L. An organization approach to adaptative production systems. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1990. **Anais...** [S.l.: s.n.], 1990. p.52–58.

KIRN ST.; GASSER, L. Organizational Approaches to Coordination in Multi-Agent Systems. **Informationstechnik und Technische Informatik (IT+TI)**, [S.l.], v.1, n.4, p.23–29, 1998.

LEMAITRE-LEÓN, C.; EXCELENTE-TOLEDO, C. Multiagent Organization Approach. In: IBEROAMERICAN WORKSHOP ON DISTRIBUTED ARTIFICIAL INTELLIGENCE AND MULTI-AGENT SYSTEMS, 2., 1998. **Anais...** [S.l.: s.n.], 1998.

LESSER, V. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. **Autonomous Agents and Multi-Agent Systems**, [S.l.], v.1, p.89–111, January 1998.

MAHESWARAN, R. T.; TAMBE, M.; BOWRING, E.; PEARCE, J. P.; VARAKANTHAM, P. Taking DCOP to the Real World: efficient complete solutions for distributed multi-event scheduling. In: THIRD INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 2004, Washington, DC, USA. **Anais...** IEEE Computer Society, 2004. v.1, p.310–317.

MAILLER, R.; LESSER, V. Solving Distributed Constraint Optimization Problems Using Cooperative Mediation. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 3., 2004, New York. **Proceedings...** New York: IEEE Computer Society, 2004. p.438–445.

MERKLE, D.; MIDDENDORF, M.; SCHMECK, H. Ant Colony Optimization for Resource-Constrained Project Scheduling. **IEEE Transactions on Evolutionary Computation**, [S.l.], v.6, n.4, p.333–146, 2002.

MODI, P. J.; SHEN, W.-M.; TAMBE, M.; YOKOO, M. An asynchronous complete method for distributed constraint optimization. In: SECOND INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 2003, New York, NY, USA. **Anais...** ACM Press, 2003. p.161–168.

NWANA, H. S.; LEE, L. C.; JENNINGS, N. R. Coordination in Software Agent Systems. **The British Telecom Technical Journal**, [S.l.], v.14, n.4, p.79–88, 1996.

OLIVEIRA, D.; BAZZAN, A. L. C.; LESSER, V. Using Cooperative Mediation to Coordinate Traffic Lights: a case study. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTI AGENT SYSTEMS (AAMAS), 4., 2005. **Proceedings...** New York: IEEE Computer Society, 2005. p.463–470.

OLIVEIRA, D.; FERREIRA, P.; BAZZAN, A. L. C. Reducing Traffic Jams with a Swarm-based Approach for Selection of Signal Plans. In: FOURTH INTERNATIONAL WORKSHOP ON ANT COLONY OPTIMIZATION AND SWARM INTELLIGENCE - ANTS 2004, 2004, Berlin, Germany. **Proceedings...** [S.l.: s.n.], 2004. p.416–417. (LNCS, v.3172).

PETCU, A.; FALTINGS, B. A Scalable Method for Multiagent Constraint Optimization. In: IJCAI 05, 2005, Edinburgh, Scotland. **Anais...** [S.l.: s.n.], 2005.

ROBISON, G. E. Regulation of Division of Labor in Insect Societies. **Annual Review of Entomology**, [S.l.], v.37, p.637–665, 1992.

SCERRI, P.; FARINELLI, A.; OKAMOTO, S.; TAMBE, M. Allocating tasks in extreme teams. In: AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 2005, New York, NY, USA. **Proceedings...** ACM Press, 2005. p.727–734.

SCHILLO, M.; FLEY, B.; FLORIAN, M.; HILLEBRANDT, F.; HINCK, D. Self-Organization in Multiagent Systems. In: THIRD INTERNATIONAL WORKSHOP ON MODELLING ARTIFICIAL SOCIETIES AND HYBRID ORGANIZATIONS (MASHO), 2002. **Anais...** [S.l.: s.n.], 2002.

SCHULZ, D. J.; HUANG, Z.-Y.; ROBINSON, G. E. Effects of colony food shortage on behavioral development in honey bees. **Behavioral Ecological Sociobiology**, [S.l.], n.42, p.295–303, 1998.

SHEHORY, O.; SYCARA, K.; CHALASANI, P.; JHA, S. Agent cloning: an approach to agent mobility and resource allocation. **IEEE Communications Magazine**, [S.l.], v.36, n.7, p.58–67, 1998.

SO, Y.; DURFEE, E. An organizational self-design model for organizational change. In: AAAI-93 WORKSHOP ON AI AND THEORIES OF GROUPS AND ORGANIZATIONS: CONCEPTUAL AND EMPIRICAL RESEARCH, 1993. **Anais...** [S.l.: s.n.], 1993. p.8–15.

STRUSEVICH, V. Shop scheduling problems under precedence constraints. In: OPERATIONS RESEARCH, 1955. **Annals...** [S.l.: s.n.], 1955. n.69, p.351–377.

TARRANT, F.; BRIDGE, D. When Ants Attack: comparing ant algorithms for constraint problems. In: PROCS. OF THE FIFTEENTH IRISH CONFERENCE ON ARTIFICIAL INTELLIGENCE AND COGNITIVE SCIENCE, 2004. **Anais...** [S.l.: s.n.], 2004. p.167–176.

THERAULAZ, G.; BONABEAU, E.; DENEUBOURG, J. Response Threshold Reinforcement and Division of Labour in Insect Societies. In: ROYAL SOCIETY OF LONDON SERIES B – BIOLOGICAL SCIENCES, 1998. **Proceedings...** [S.l.: s.n.], 1998. v.265, p.327–332.

WAGNER, T.; GURALNIK, V.; PHELPS, J. A key-based coordination algorithm for dynamic readiness and repair service coordination. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 2., 2003, New York. **Proceedings...** New York: ACM Press, 2003.

WAGNER, T.; LESSER, V. R. Design-to-Criteria Scheduling: real-time agent control. In: SPRING SYMPOSIUM ON REAL-TIME AUTONOMOUS SYSTEMS, AAAI, 2000, Stanford, CA. **Proceedings...** [S.l.: s.n.], 2000. p.89–96.

WILSON, E. O. **Sociobiology**: the new synthesis. [S.l.]: Harvard Univ Press, 2000.