

HEURÍSTICA PARA GERENCIAR A COOPERAÇÃO DE UM GRUPO DE ROBÔS EM EXECUÇÃO DE TAREFAS

D. R. F. Barbosa¹, R. S. Maia¹ e A. A. S. Souza¹

¹Departamento de Computação – Universidade do Estado do Rio Grande do Norte
danilorfb@gmail.com - rosiery@uern.br – andersonabner@uern.br

Artigo submetido em novembro/2011 e aceito em fevereiro/2012

RESUMO

Propomos uma heurística que define as regras para uma cooperação eficiente de um sistema multirrobô, empregado para a resolução de tarefas em um ambiente controlado. Essa heurística possibilita que um grupo de robôs tenha melhor desempenho na

resolução de uma lista de tarefas sem a intervenção de algum agente externo ao ambiente. Por meio das regras definidas para as trocas de experiências, o ambiente obteve ganho de conhecimento, melhor utilização dos robôs e redução no tempo de execução das tarefas.

PALAVRAS-CHAVE: grupo de robôs, heurísticas, cooperação.

HEURISTIC TO MANAGE THE COOPERATION OF A GROUP OF ROBOTS IN TASKS EXECUTION

ABSTRACT

We propose a heuristic that defines the rules to an efficient cooperation of a multi-robot system used to execute tasks in a controlled environment. This heuristic enables a group of robots to have better performance during the execution of a task list without the

intervention of an external agent. The environment increased knowledge, better utilization of robots and reduction on tasks execution time by the rules defined to exchanges of experiences.

KEY-WORDS: group of robots, heuristics, cooperation.

HEURÍSTICA PARA GERENCIAR A COOPERAÇÃO DE UM GRUPO DE ROBÔS EM EXECUÇÃO DE TAREFAS

1. INTRODUÇÃO

Os sistemas aplicados em problemas em que tarefas são executadas por mais de um robô são chamados de sistemas multirrobôs e, de acordo com [Navarro-Serment et al., 2002], existe uma tarefa que é comum a todos os componentes do ambiente. É possível ver exemplos de aplicações com sistemas multirrobôs em [Chen et al., 2008], [Carrascosa et al., 2008], [Mes et al., 2007], [Toksari, 2007] e [Rooker and Birk, 2005]. Portanto, a principal característica desses sistemas é a cooperação em grupo para resolução da tarefa global, o que torna de suma importância pesquisar alternativas que modifiquem essa cooperação, tornando-a mais eficiente, e se possível, com troca de conhecimentos entre os componentes.

Alguns trabalhos, como [Chaimowicz et al., 2001] e [Kambayashi et al., 2009], utilizam mecanismos que coordenam a cooperação de forma eficiente, porém sem ganho de conhecimento por parte do ambiente.

Pensando nesse contexto da cooperação, o objetivo desse trabalho é gerenciar um ambiente simulado composto por um grupo de robôs heterogêneos, proporcionando uma capacidade de autonomia e de aquisição de conhecimento, por meio de uma heurística.

As seções seguintes desse artigo informam: no item 2, o desenvolvimento do trabalho; no item 3, os experimentos avaliados e seus resultados; e o item 4, a conclusão obtida acerca dos resultados.

2. CONFIGURAÇÃO DO SISTEMA MULTIRROBÔ

Inicialmente o ambiente multirrobô foi inspirado segundo os conceitos formalizados por [Maia et al., 2010], já que foi configurado para ser avaliado em conformidade com uma aplicação real de execução cooperativa de tarefas. Ele possui um conjunto finito de robôs e outro conjunto finito de tarefas. Estas compõem a missão do ambiente em um tempo qualquer e devem ser realizadas pelos robôs. As equações (1) e (2) abaixo definem esses conjuntos, sabendo que $m \in \mathbb{N}$ e $n \in \mathbb{N}$.

$$\mathbf{R} = \{r_1, r_2, \dots, r_m\} \quad \text{Equação (1)}$$

$$\mathbf{T} = \{t_1, t_2, \dots, t_n\} \quad \text{Equação (2)}$$

Para cada robô existe um conjunto de conhecimentos básicos e um conjunto de capacidades físicas, definidos nas equações 3 e 4, nas quais i, x e $y \in \mathbb{N}$. Cada tarefa também possui um conjunto de conhecimentos básicos (informações e/ou instruções que o robô precisa saber para executá-la) necessários para sua perfeita execução conforme a Equação 5, em que j e $k \in \mathbb{N}$, assim como um conjunto de características físicas (ou, todo o hardware constituinte do robô, como braço mecânico, roda, câmera, sonar, acelerômetro, dentre outros) necessárias que o robô deve ter para que seja possível resolvê-la, conforme a Equação

6, em que j e $w \in \mathbb{N}$. A composição formal de um robô é definida na Equação 7, em que $i \in \mathbb{N}$. Nela estão inseridos os conhecimentos e as capacidades físicas do robô. Seguindo a mesma ideia, as tarefas que compõem a missão do ambiente, em um determinado tempo, estão descritas conforme a Equação 8, em que $j \in \mathbb{N}$.

$$C(r_i) = \{c_1, c_2, \dots, c_x\} \quad \text{Equação (3)}$$

$$F(r_i) = \{f_1, f_2, \dots, f_y\} \quad \text{Equação (4)}$$

$$C(t_j) = \{c_1, c_2, \dots, c_k\} \quad \text{Equação (5)}$$

$$F(t_j) = \{f_1, f_2, \dots, f_w\} \quad \text{Equação (6)}$$

$$r_i = \{C(r_i)|F(r_i)\} \quad \text{Equação (7)}$$

$$t_j = \{C(t_j)|F(t_j)\} \quad \text{Equação (8)}$$

A função objetivo para uma aplicação multirrobô simulada nesse contexto foi adaptada da definida por [Maia et al., 2011], que visa maximizar o conhecimento total do ambiente em cada execução completa de uma dada missão $M=\{t_1, t_2, \dots, t_n\}$, em que $n \geq 0$, $n \in \mathbb{N}$ e M é uma lista de tarefas. Essa função está informada abaixo:

$$\max \sum_{i=1}^m \frac{C(r_i)}{J} \quad \text{Equação (9)}$$

Em que:

- $C(r_i)$ é o conjunto de conhecimentos de um robô r_i ;
- J é o instante avaliado;
- m é o total de robôs do ambiente e $m \in \mathbb{N}$.

2.1. COMPORTAMENTO DO AMBIENTE

Inicialmente, em um determinado tempo, é apresentada uma missão ao ambiente (composto por um grupo de robôs). Esta é uma lista de tarefas que deve ser executada. Imediatamente o ambiente verifica quais os robôs do grupo que estão ociosos e, a partir disso, as tarefas são designadas aos devidos robôs para execução, seguindo as regras que estão descritas na heurística desenvolvida (item 2.3).

O objetivo dos robôs é alcançado ao resolver a missão de tal maneira que todas as tarefas sejam resolvidas. Porém, para que uma tarefa seja resolvida, é necessário que pelo

menos um robô possua simultaneamente os conhecimentos básicos e as características físicas necessários e exigidos pela tarefa em questão. Desse modo, comparam-se os conjuntos de conhecimentos básicos e o conjunto de características físicas de cada tarefa com os respectivos conjuntos dos robôs. Aquele que for apto a resolver a tarefa, ou seja, aquele que possuir todos os conhecimentos e todas as capacidades físicas em comum com a tarefa irá resolvê-la.

2.3. HEURÍSTICA DOS PESOS

A heurística desenvolvida segue a ideia de explorar o robô apto menos utilizado a cada execução de uma tarefa da missão. Para isso, cada robô possui um atributo (denominado de peso) que identifica a quantidade de vezes que ele foi requisitado para resolução de tarefas (de uma mesma missão). Desse modo, o formalismo do robô passa a ser conforme a Equação 10, em que $P \in \mathbb{N}$ e P é o peso.

O intuito dessa heurística é minimizar os danos que podem ser causados pela superutilização de alguns robôs com maior quantidade de conhecimentos iniciais, bem como diminuir o tempo de término da missão completa. A operacionalização da aplicação dessa heurística está mais bem informada na próxima seção.

$$\mathbf{r}_i = \{\mathbf{C}(\mathbf{r}_i) | \mathbf{F}(\mathbf{r}_i) | \mathbf{P}(\mathbf{r}_i)\} \quad \text{Equação (10)}$$

A heurística desenvolvida aborda uma ideia principal que é a definição de cenários como situações passíveis de ocorrer no ambiente, quando exposto a uma missão. Esses cenários estão divididos em três categorias: o Ideal, o de Intercâmbio e o Pior cenário.

O Ideal é aquele no qual pelo menos um robô é dito completamente apto a resolver uma tarefa, pois possui simultaneamente o conhecimento e a capacidade física exigidos para que uma tarefa possa ser solucionada.

O Cenário de Intercâmbio ocorre quando não existe robô com o conhecimento necessário para solucionar a tarefa, mas existe pelo menos um robô que possui a capacidade física para solucioná-la. Neste caso, verifica-se se no conjunto de robôs aqueles que possuem completa ou parcialmente o(s) conhecimento(s) necessário(s) à resolução da tarefa e, caso seja possível obter todo o conhecimento exigido dentre os robôs que compõem o ambiente, ocorre o intercâmbio de conhecimentos. Nessa situação, os robôs que sabem realizar a tarefa transferem seu saber para o robô que possui a capacidade física para resolvê-la, tornando-o completamente apto para a resolução da tarefa.

Por fim, o Pior Cenário é aquele em que não existe no conjunto de robôs nenhum que possua a capacidade física para resolver uma tarefa ou que, caso exista pelo menos um com esta característica, não seja possível obter por intercâmbio todo o conhecimento exigido para a resolução da tarefa. A ordem de precedência dos cenários é apresentada no Algoritmo 1. Analisaremos a resolução de duas tarefas para demonstrar os cenários que podem acontecer.

Algoritmo 1 - Pseudocódigo do procedimento para executar uma missão pelo ambiente utilizando os cenários definidos.

```

01 Início
02   Atualizar_Instâncias(R,M)
03   Enquanto M<>{} Faça até que não haja mais tentativas de resolver cada tarefa
04      $t_j = \text{Buscar\_Próxima\_Tarefa}(M);$ 
05      $R\_Apto = \text{Buscar\_Robôs\_Aptos}(R);$  //Cenário Ideal
06     Se R_Apto<>{} Então
07        $r_i = \text{Escolher\_Robô}(R\_Apto);$ 
08       Executar_tarefa( $r_i, t_j$ )
09     Senão
10        $R\_Capaz = \text{buscar\_Robôs\_Fisicamente\_Capazes}(R);$  //Cenário de Intercâmbio
11       Se R_Capaz<>{} Então
12          $R\_IntelectualmenteAptos = \text{Buscar\_Robôs\_Com\_Conhecimento}(C(t_j));$ 
13         Se (R_IntelectualmenteAptos<>{}) e (R_IntelectualmenteAptos  $\supseteq$  C( $t_j$ )) Então
14            $r_i = \text{Escolher\_Robô}(R\_Capaz);$ 
15           Repita
16              $r_{doador} = \text{Escolher\_Robô}(R\_IntelectualmenteAptos);$ 
17             Efetuar_Intercâmbio( $r_{doador}, r_i$ );
18           Até que C( $r_i$ )  $\supseteq$  C( $t_j$ );
19           Executar_tarefa( $r_i, t_j$ )
20         Senão
21            $R\_Sábios = \text{Buscar\_Robôs\_Com\_Conhecimento\_Parcial}(C(t_j));$ 
22           Se (R_Sábios<>{}) e (R_Sábios  $\supseteq$  C( $t_j$ )) Então
23              $r_i = \text{Escolher\_Robô}(R\_Capaz);$ 
24             Repita
25                $r_{doador} = \text{Escolher\_Robô}(R\_Sábios);$ 
26               Efetuar_Intercâmbio( $r_{doador}, r_i$ );
27             Até que C( $r_i$ )  $\supseteq$  C( $t_j$ );
28             Executar_tarefa( $r_i, t_j$ )
29           Fim_Se
30         Senão
31           Retorne "impossível execução"; //Pior Cenário
32       Fim_Enquanto
33   Fim

```

3. RESULTADOS OBTIDOS

Em um primeiro experimento foi utilizada a configuração do ambiente de acordo com o exposto na Tabela 1: 10 robôs heterogêneos e uma missão M composta por 10 tarefas. A primeira tarefa que deverá ser resolvida por algum robô é a t_4 ; por conseguinte, a t_9 , e assim por diante até a resolução da última tarefa, a t_7 . A configuração dos robôs e das tarefas está na Tabela 1, que será utilizada na análise da resolução de tarefas.

Tabela 1: Configurações dos robôs, tarefas e missão.

ROBÔS	TAREFAS
$r_1 = \{[c_4]; [f_1]\}$	$t_1 = \{[c_1]; [f_3]\}$
$r_2 = \{[]; [f_1, f_2]\}$	$t_2 = \{[c_1, c_2, c_3]; [f_1, f_3]\}$
$r_3 = \{[c_1]; [f_2, f_3]\}$	$t_3 = \{[c_2, c_3]; [f_1, f_2]\}$
$r_4 = \{[c_1, c_4]; [f_1]\}$	$t_4 = \{[c_2]; [f_2]\}$
$r_5 = \{[c_2]; [f_1, f_2, f_3]\}$	$t_5 = \{[c_3, c_4]; [f_1]\}$
$r_6 = \{[c_1, c_3]; [f_2]\}$	$t_6 = \{[c_4]; [f_2]\}$
$r_7 = \{[c_3]; [f_1, f_3]\}$	$t_7 = \{[c_1, c_3]; [f_3]\}$
$r_8 = \{[]; [f_1, f_2, f_3]\}$	$t_8 = \{[c_3]; [f_1]\}$
$r_9 = \{[c_2, c_3]; [f_3]\}$	$t_9 = \{[c_4]; [f_1, f_2, f_3]\}$
$r_{10} = \{[c_2, c_3, c_4]; [f_1, f_3]\}$	$t_{10} = \{[c_1, c_4]; [f_1, f_2]\}$
$M = \{t_4, t_9, t_6, t_8, t_1, t_{10}, t_2, t_5, t_3, t_7\}$	

Inicialmente analisaremos a tarefa t_4 verificando o conjunto de robôs, à procura de algum que possua o conhecimento c_2 e a capacidade física f_2 . Caso exista algum robô com esta configuração, será este o responsável pela resolução de t_4 . Constata-se então que o robô r_5 possui c_2 e f_2 e, portanto, é completamente apto a resolver essa tarefa. Esta situação corresponde ao Cenário Ideal. Dando prosseguimento à resolução de M , após t_4 ter sido resolvida por r_5 , partiremos em busca de um robô que possa solucionar t_9 . Observamos que nenhum robô possui simultaneamente conhecimento e capacidade física que satisfaçam sua resolução. Já que o Cenário Ideal não foi concretizado, o segundo cenário que pode ocorrer é o de Intercâmbio de conhecimentos.

Existem dois robôs que têm a capacidade física exigida por t_9 : r_5 e r_8 . Então, verificamos se é possível compor o conjunto de conhecimentos requerido por t_9 por meio do intercâmbio de conhecimentos entre os robôs. Assim, é possível obter c_4 de r_1 , r_4 ou r_{10} , de tal maneira que após a transferência de conhecimento, r_5 ou r_8 se tornem aptos a resolver t_9 . Em uma aplicação real, a escolha entre r_5 ou r_8 para receber c_4 seria aleatória, tal qual a escolha para a doação do conhecimento entre r_1 , r_4 e r_{10} . A priori, sem a aplicação de nenhuma heurística para a determinação da escolha do robô, elegeremos r_5 para receber o conhecimento requerido, bem como r_1 para a doação de c_4 . Desse modo, a configuração de r_5 passa a ser $r_5 = \{[c_2, c_4]; [f_1, f_2, f_3]\}$. Após receber c_4 , r_5 guarda esse conhecimento permanentemente. O robô r_1 continua com a mesma configuração: $r_1 = \{[c_4]; [f_1]\}$. Logo, esta situação ilustra um Cenário de Intercâmbio, no qual o robô fisicamente apto “aprendeu” a resolver uma tarefa. A lógica de execução segue, portanto, a ordem definida pelos cenários: primeiro o Cenário Ideal é almejado; caso este não ocorra, busca-se o Cenário de Intercâmbio. Finalmente, se este também não ocorrer, o Pior Cenário será concretizado. Na configuração definida conforme a Tabela 1, não encontraremos a ocorrência do Pior Cenário, pois caso não

ocorra o Cenário Ideal, o Cenário de Intercâmbio sempre ocorrerá na realização das tarefas, que podem ser todas resolvidas e assim o objetivo dos robôs é alcançado.

Em uma aplicação real (simulada), a escolha de robôs para a realização de tarefas dá-se aleatoriamente, sem nenhum parâmetro que a justifique. Desse modo, apesar de ser possível o cumprimento da missão, alguns robôs podem se tornar muito sobrecarregados, enquanto outros possam ficar em estado de ócio, mesmo sendo capazes de obter os pré-requisitos para solucionar uma tarefa por meio do intercâmbio de conhecimentos. Por exemplo, r_5 e r_8 possuem capacidades físicas para resolver qualquer uma das tarefas definidas na Tabela 1. Caso ocorram situações em que seja necessário que algum desses dois robôs adquira conhecimentos por Intercâmbio, um deles (ou ambos) poderá se tornar completamente apto a resolver qualquer tarefa. Por exemplo, r_5 pode aprender todos os conhecimentos de todos os robôs por intercâmbio, e obter a seguinte configuração: $r_5 = \{[c_1, c_2, c_3, c_4]; [f_1, f_2, f_3]\}$. Portanto, qualquer tarefa que apareça na ordem da missão após r_5 adquirir essa configuração será sempre resolvida por ele, já que o Cenário Ideal é sempre o primeiro a ser investigado, e r_5 sempre satisfará essa condição. Assim, outros robôs ficarão sempre em estado de ócio, enquanto r_5 ficará sobrecarregado. A Tabela 2 mostra essa situação. Foram poucos os robôs utilizados (r_3, r_5 e r_7), sendo que o r_5 ficou sobrecarregado. Além disso, uma grande quantidade de robôs ficou em estado de ócio: $r_1, r_2, r_4, r_6, r_8, r_9$ e r_{10} .

Tabela 2: Resolução de $M = \{t_4, t_9, t_6, t_8, t_1, t_{10}, t_2, t_5, t_3, t_7\}$ por uma aplicação real.

RESOLUÇÃO DA MISSÃO	CONFIGURAÇÃO DA TAREFA	CONFIGURAÇÃO DO ROBÔ
t_4 foi resolvida por r_5	$t_4 = \{[c_2]; [f_2]\}$	$r_5 = \{[c_2]; [f_1, f_2, f_3]\}$
t_9 foi resolvida por r_5	$t_9 = \{[c_4]; [f_1, f_2, f_3]\}$	$r_5 = \{[c_2, c_4]; [f_1, f_2, f_3]\}$
t_6 foi resolvida por r_5	$t_6 = \{[c_4]; [f_2]\}$	$r_5 = \{[c_2, c_4]; [f_1, f_2, f_3]\}$
t_8 foi resolvida por r_7	$t_8 = \{[c_3]; [f_1]\}$	$r_7 = \{[c_3]; [f_1, f_3]\}$
t_1 foi resolvida por r_3	$t_1 = \{[c_1]; [f_3]\}$	$r_3 = \{[c_1]; [f_2, f_3]\}$
t_{10} foi resolvida por r_5	$t_{10} = \{[c_1, c_4]; [f_1, f_2]\}$	$r_5 = \{[c_1, c_2, c_4]; [f_1, f_2, f_3]\}$
t_2 foi resolvida por r_5	$t_2 = \{[c_1, c_2, c_3]; [f_1, f_3]\}$	$r_5 = \{[c_1, c_2, c_3, c_4]; [f_1, f_2, f_3]\}$
t_5 foi resolvida por r_5	$t_5 = \{[c_3, c_4]; [f_1]\}$	$r_5 = \{[c_1, c_2, c_3, c_4]; [f_1, f_2, f_3]\}$
t_3 foi resolvida por r_5	$t_3 = \{[c_2, c_3]; [f_1, f_2]\}$	$r_5 = \{[c_1, c_2, c_3, c_4]; [f_1, f_2, f_3]\}$
t_7 foi resolvida por r_5	$t_7 = \{[c_1, c_3]; [f_3]\}$	$r_5 = \{[c_1, c_2, c_3, c_4]; [f_1, f_2, f_3]\}$

Todavia, a escolha de robôs por uma aplicação heurísticamente enriquecida garante uma distribuição de tarefas de maneira mais equilibrada. Propõe-se uma heurística que garante uma maior eficiência, organização e distribuição equilibrada de tarefas com a aplicação de pesos nos robôs. Cada vez que um robô realiza uma tarefa, este tem seu peso incrementado em uma unidade. Supondo agora que a mesma missão definida na Tabela 1 será executada com o incremento de pesos em cada robô utilizado, $t_4 = \{[c_2]; [f_2]\}$ será

resolvida por $r_5 = \{[c_2]; [f_1, f_2, f_3]\}$, que é o único robô que atende ao caso ideal. já que r_5 resolveu t_4 , este que inicialmente tinha um peso 0, passa a ter um peso 1.

A próxima tarefa a ser resolvida é a $t_9 = \{[c_4]; [f_1, f_2, f_3]\}$, que como supradito, não há nenhum robô que se enquadre no Cenário Ideal para resolvê-la. Entretanto, há dois robôs candidatos à sua resolução: o $r_5 = \{[c_2]; [f_1, f_2, f_3]\}$ e o $r_8 = \{[]; [f_1, f_2, f_3]\}$, que possuem as capacidades físicas requeridas e se adequam ao Cenário de Intercâmbio. Na aplicação real, a escolha do robô para receber o conhecimento se dava de maneira aleatória. Agora, a Heurística dos Pesos dos robôs passa a ser um fator preponderante na escolha de quem receberá o conhecimento necessário à resolução de t_9 , ou seja, o c_4 . O r_5 tem peso 1, pois já resolveu uma tarefa (t_4), enquanto o r_8 tem peso 0, pois ainda está em estado de ócio. Desse modo, elegemos r_8 para receber o conhecimento c_4 , tornando-se completamente apto a resolver a tarefa em questão. Assim, esse robô, que antes possuía a configuração $r_8 = \{[]; [f_1, f_2, f_3]\}$ e peso 0, possuirá: $r_8 = \{[c_4]; [f_1, f_2, f_3]\}$, e terá seu peso incrementado em 1 unidade. Este princípio heurístico também se aplica caso exista mais de um robô que se enquadre no Cenário Ideal para resolver uma tarefa, para definir qual robô deverá resolvê-la, sendo este escolhido por possuir o menor peso dentre seus concorrentes. Além disso, também aplicamos este princípio para definir qual robô deve repassar seu conhecimento, elegendo o de menor peso. Caso os robôs possuam o mesmo peso, a escolha entre eles será aleatória. Essa operação é realizada até que não existam mais tarefas a serem executadas da missão.

Tabela 3: Resolução de $M = \{t_4, t_9, t_6, t_8, t_1, t_{10}, t_2, t_5, t_3, t_7\}$ por uma aplicação enriquecida com a Heurística dos Pesos.

RESOLUÇÃO DA MISSÃO	CONFIGURAÇÃO DA TAREFA	CONFIGURAÇÃO DO ROBÔ
t_4 foi resolvida por r_5	$t_4 = \{[c_2]; [f_2]\}$	$r_5 = \{[c_2]; [f_1, f_2, f_3]; [1]\}$
t_9 foi resolvida por r_8	$t_9 = \{[c_4]; [f_1, f_2, f_3]\}$	$r_8 = \{[c_4]; [f_1, f_2, f_3]; [1]\}$
t_6 foi resolvida por r_8	$t_6 = \{[c_4]; [f_2]\}$	$r_8 = \{[c_4]; [f_1, f_2, f_3]; [2]\}$
t_8 foi resolvida por r_7	$t_8 = \{[c_3]; [f_1]\}$	$r_7 = \{[c_3]; [f_1, f_3]; [1]\}$
t_1 foi resolvida por r_3	$t_1 = \{[c_1]; [f_3]\}$	$r_3 = \{[c_1]; [f_2, f_3]; [1]\}$
t_{10} foi resolvida por r_2	$t_{10} = \{[c_1, c_4]; [f_1, f_2]\}$	$r_2 = \{[c_1, c_4]; [f_1, f_2]; [1]\}$
t_2 foi resolvida por r_{10}	$t_2 = \{[c_1, c_2, c_3]; [f_1, f_3]\}$	$r_{10} = \{[c_1, c_2, c_3, c_4]; [f_1, f_3]; [1]\}$
t_5 foi resolvida por r_{10}	$t_5 = \{[c_3, c_4]; [f_1]\}$	$r_{10} = \{[c_1, c_2, c_3, c_4]; [f_1, f_3]; [2]\}$
t_3 foi resolvida por r_2	$t_3 = \{[c_2, c_3]; [f_1, f_2]\}$	$r_2 = \{[c_1, c_2, c_3, c_4]; [f_1, f_2]; [2]\}$
t_7 foi resolvida por r_9	$t_7 = \{[c_1, c_3]; [f_3]\}$	$r_9 = \{[c_1, c_2, c_3]; [f_3]; [1]\}$

Assim, podemos observar que a eficiência na distribuição balanceada e realização de tarefas por uma aplicação heurísticamente enriquecida é superior a uma aplicação real. É possível chegar a essa conclusão mesmo sem diretamente levar em conta o tempo gasto para finalizar a missão em ambos os casos. Por exemplo, para uma aplicação real (simulada) de resolução de tarefas por um grupo de robôs, a missão demoraria mais tempo a ser concluída,

uma vez que poderá haver robô(s) sobrecarregado(s), que só poderá(ão) realizar outra tarefa após terminar a que está executando. Desse modo, várias tarefas ficam aguardando a sua vez de serem resolvidas. Com a utilização da Heurística dos Pesos, várias tarefas são resolvidas ao mesmo tempo por vários robôs, diminuindo consideravelmente o fator tempo para concluir a missão. É possível observar o balanceamento na distribuição de tarefas ao confrontarmos os dados obtidos na Tabela 3 com os dados da Tabela 2. Nesse caso, os robôs utilizados foram r_2 , r_3 , r_5 , r_7 , r_8 , r_9 e r_{10} . Já os em estado de ócio foram r_1 , r_4 e r_6 .

4. CONCLUSÃO

Portanto, com base nos resultados obtidos pelos experimentos realizados, podemos concluir que a utilização da Heurística dos Pesos contribui positivamente para gerenciar a cooperação de um grupo de robôs conforme a Tabela 4, distribuindo-os de forma balanceada e eficiente na realização de tarefas se comparada a uma aplicação sem heurística bem como a uma aplicação com o conceito de cenários. O emprego da Heurística dos Pesos evita a sobrecarga de trabalho em apenas alguns robôs e, conseqüentemente, diminui o tempo gasto na realização de todas as tarefas de uma missão, já que várias tarefas são resolvidas ao mesmo tempo. Além disso, também favorece a um ganho de conhecimento para o ambiente conforme a Figura 1, no qual vários robôs obtêm novos conhecimentos entre si, sem a influência de agentes externos.

Tabela 4: Comparativo entre uma aplicação sem cooperação (S/H), com conceitos de cenários (C-C) e com conceito de cenários utilizando a Heurística dos Pesos (C-HP).

APLICAÇÃO	S/C	C-C	C-HP
FATORES	RESULTADOS		
Quantidade de tarefas resolvidas	4	10	10
Quantidade de tarefas não resolvidas	6	0	0
Quantidade de conhecimentos antes de resolver as tarefas	13	13	13
Quantidade de conhecimentos depois de resolver as tarefas	13	16	20
Quantidade de robôs utilizados	3	3	6
Quantidade de robôs em estado de ócio	7	7	4

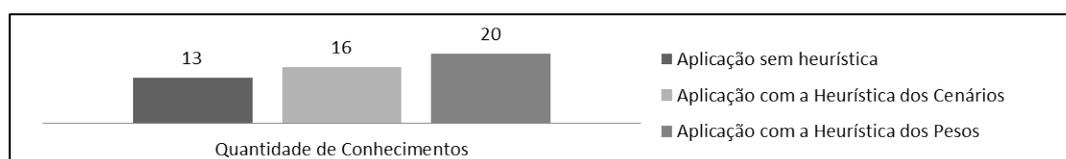


Figura 1 - Quantidade de conhecimentos dos robôs que compõem o ambiente.

REFERÊNCIAS BIBLIOGRÁFICAS

1. CARRASCOSA, C., BAJO, J., JULIAN, V., CORCHADO, J. M. and BOTTI, V. Hybrid multi-agent architecture as a real-time problem-solving model. *Expert Systems with Applications*. 34(1): 2-17, 2008.
2. CHAIMOWICZ, I., SUGAR, T., KUMAR, V. and CAMPOS, M. F. M. An architecture for tightly coupled multi-robot cooperation. *ICRA, IEEE*, pp. 2992-2997.
3. CHEN, D., JENG, B., LEE, W. and CHUANG, C. An agent-based model for consumer-to business electronic commerce. 34(1): 469-481, 2008.
4. KAMBAYASHI, Y., TSUJIMURA, Y., YAMACHI, H., TAKIMOTO, M. and YAMAMOTO H. Design of a multi-robot system using mobile agents with ant colony clustering. *System Sciences, HICSS'09. 42nd Hawaii International Conference*, pp. 1-10, 2009.
5. MAIA, R. S., SOUZA, A. A. S., and GONÇALVES, L. M. G. Um modelo do processo de aprendizagem proposto para um ambiente multi-robô. In *IX Congreso de La Sociedad Peruana de Computación, Trujillo. Jornada Peruana de Computación*, 2010.
6. MAIA, R. S., SOUZA, A. A. S., and GONÇALVES, L. M. G. Concepção de um formalismo de aprendizagem baseado em modelos sociais para um time de robôs em execução cooperativa de tarefas. In *X SBAI - Simpósio Brasileiro de Automação Inteligente, São João Del-Rei-MG*, 2011.
7. MES, M., HEIJDEN, M. V. D. and HARTEN, V. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. 181:59-75, 2007.
8. NAVARRO-SERMENT, L. E., GRABOWSKI, R., PAREDIS, C. J. J., and K. KHOSLA, P. Milli-bots: The development of a framework and algorithms for a distributed heterogeneous robot team. *IEEE Robotics and Automation Magazine*, 9:31-40, 2002.
9. ROOKER, M. N. and Birk, A. Combining exploration and ad-hoc networking in robocup rescue, environments. 8(8): 809-833, 2005.
10. TOKSARI, M. D. Ant Colony Optimization approach to estimate energy demand of turkey. 35: 3984-3990, 2007.