

Comparação da Busca Tabu com uma Heurística Simples no Escalonamento de Barramento de Campo

Lucy María Franco Vargas, Rômulo Silva de Oliveira

Departamento de Automação e Sistemas - Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brasil

{lucy,romulo}@das.ufsc.br

Resumo. *O objetivo deste artigo é comparar a meta-heurística Busca Tabu com uma heurística simples na resolução do problema de alocação e escalonamento de blocos funcionais em barramentos de campo que seguem a especificação Foundation Fieldbus. Nas experiências são empregados cenários baseados em aplicações reais de controle distribuído.*

1. Introdução

Sistemas de controle com realimentação em que os laços de controle são fechados por uma rede de tempo real são chamados Sistemas de Controle Baseados em Rede (NCS – *Networked Control Systems*). A principal característica de um NCS é que a informação (entrada de referência, saída da planta, entrada do controle, etc.) é trocada através de uma rede entre os componentes do sistema de controle (sensores, controlador, atuadores, etc.) [Wei 2001].

O *Foundation Fieldbus* (FF) é um sistema de comunicação digital, serial e bidirecional, que permite a comunicação entre equipamentos de supervisão, de controle e dispositivos de campo [Fieldbus Foundation 2003]. O controle da comunicação e de acesso ao *fieldbus* é responsabilidade de um dispositivo chamado Escalonador Ativo de Enlace (LAS – *Link Active Scheduler*). O LAS administra a permissão dada aos dispositivos que desejam transmitir dados no barramento, de tal forma que somente um único dispositivo acesse o barramento em cada instante. O FF especificou um modelo de Aplicação do Usuário onde cada dispositivo de campo cumpre sua parte na operação total do sistema implementando um ou mais blocos funcionais (FB – *Function Blocks*). Os FB descrevem as funções disponíveis em um dado dispositivo e definem como estas podem ser acessadas.

Depois de ser definida a estratégia de controle em termos de blocos funcionais, é necessário fazer a alocação de cada bloco aos diversos dispositivos existentes na rede. Por alocação entende-se decidir qual bloco executa em qual dispositivo de campo. Após a alocação dos blocos nos dispositivos, é necessário fazer o escalonamento destes para garantir uma correta ordem de execução e comunicação. O escalonamento refere-se a decisão de “qual bloco executa quando no processador que foi alocado” e também “qual mensagem é transmitida quando no barramento de comunicação”.

Para a resolução de problemas como este e muitos outros classificados como *NP-Hard* ou NP-Completo pela teoria de complexidade computacional podem ser utilizados métodos conhecidos como meta-heurísticos. Em [Rafael 2003], os métodos meta-heurísticos são definidos como um marco geral para criar novos algoritmos híbridos combinando diferentes conceitos derivados da inteligência artificial, da evolução biológica e dos mecanismos estatísticos. Atualmente existem vários destes métodos que tem demonstrado sua eficácia em uma grande variedade de problemas, como é o caso do método Busca Tabu.

A Busca Tabu é uma meta-heurística que guia uma heurística de busca local para explorar o espaço de soluções de tal modo que seja evitado ser apanhado em mínimos locais. Está baseada em princípios gerais da inteligência artificial (IA). Toma da IA o conceito de memória e o implementa através de simples estruturas com o objetivo de dirigir a busca considerando a história dela [Rafael 2003].

Neste artigo é feita uma comparação do Busca Tabu com uma heurística simples na resolução do problema de alocação e escalonamento de blocos funcionais e mensagens periódicas do *Foundation Fieldbus*.

2. Descrição do problema

Uma estratégia de controle têm n laços de controle os quais são executados em m dispositivos de campo. O laço de controle $L[i]$, $i = 1, \dots, n$, é composto de nb_i tarefas blocos funcionais e np_i tarefas de comunicação (mensagens periódicas).

Uma tarefa $T_j[i]$, $j = 1, \dots, (nbi + npi)$, (bloco funcional ou de comunicação) do laço de controle $L[i]$ pode ser definida como: $T_j[i] = (C_j[i], P_j[i], D_j[i])$, onde $C_j[i]$ é o tempo de computação ou transmissão requerido para a execução da tarefa $T_j[i]$, o período $P_j[i]$ é o tempo fixo no qual a tarefa $T_j[i]$ deveria ser executada repetidamente e o *deadline* $D_j[i]$ é o tempo máximo que a tarefa $T_j[i]$ têm para ser executada. Estes tempos, $C_j[i]$, $P_j[i]$ e $D_j[i]$, são conhecidos; o $C_j[i]$ das mensagens que são enviadas entre blocos funcionais no mesmo dispositivo é considerado igual a 0. O *deadline* $D_j[i]$ é considerado igual ao período da tarefa. Algumas das tarefas $T_j[i]$ do laço de controle $L[i]$ têm relações de precedência.

Cada tarefa $T_j[i]$ no laço de controle $L[i]$ vai ser escalonada $ns_j[i]$ vezes no macrociclo:

$$ns_j[i] = \frac{\text{macrociclo}}{P_j[i]}, \text{ onde } P_j[i] \text{ é o período da tarefa } T_j[i]. \quad (1)$$

Uma solução de escalonamento neste contexto consiste da definição da alocação e também da construção de uma grade de tempo informando que tarefa (bloco funcional ou de comunicação) executa a cada instante. A grade terá o tamanho do macrociclo.

Neste trabalho é considerada como a melhor solução a que apresenta o menor valor no resultado da fórmula abaixo. As constantes na fórmula foram utilizadas para permitir que a meta-heurística possa dar prioridade àqueles conflitos com o maior valor da constante e portanto resolve-los primeiro que os outros. Estes valores para as constantes foram definidos após um conjunto de experiências preparatórias.

$$\text{Avaliação} = (100 * \text{NumeroPrecedenciasIgnoradas}) + (1 * \text{DeadlineNaoRespeitado}) + (10 * \text{ConflitosBarramento}) + (1 * \text{ConflitosBlocos}).$$

3. Heurística Simples

Existem blocos funcionais que só podem executar em processadores específicos (por exemplo os blocos AI e AO). Na heurística implementada a alocação dos blocos que não têm um processador predefinido, quer dizer, que podem executar em qualquer dispositivos conectados à rede, foi feita da seguinte forma: os blocos PID são alocados nos respectivos dispositivos atuadores e os outros blocos funcionais que não têm um processador específico para execução são colocados em qualquer dos outros dispositivos sensores.

O escalonamento é dividido em duas partes, primeiro é feito o escalonamento nos dispositivos processadores e depois no barramento. Para o escalonamento nos processadores,

é atribuído como início de execução de cada bloco sucessor o final de execução do seu bloco predecessor e o início dos blocos que não têm um predecessor é igual a 0. No escalonamento no barramento, o início de execução das mensagens enviadas entre os blocos funcionais é igual ao final de execução do bloco que envia a mensagem mais o final de execução da última mensagem que foi transmitida, isto para evitar o conflito entre as mensagens.

Depois da alocação e escalonamento dos blocos funcionais e mensagens enviadas entre estes blocos, é feita uma verificação do início de execução dos blocos sucessores, isto para que o bloco sucessor execute depois de receber a mensagem que lhe foi enviada. A seguir é apresentado o pseudo-código da heurística implementada.

Início

Alocação dos blocos. Alocar os blocos que podem executar em qualquer processador.

Blocos PID nos atuadores.

Se houver outros blocos, alocar-los em algum dos dispositivos sensores.

Escalonamento nos processadores. Atribuir os inícios de execução dos blocos.

Início de execução do sucessor = Final de execução do predecessor.

Início de execução dos blocos sem predecessor = 0.

Escalonamento no barramento. Atribuir os inícios de execução das mensagens.

Início de execução da mensagem = Final de execução do bloco origem + última mensagem transmitida.

Se (É mensagem externa) **então**

Última mensagem = Início de execução da mensagem + tempo de computação da mensagem.

Atualização do início de execução. Para todos os blocos que recebem uma mensagem (sucessores).

Se (Início de execução do sucessor < final de execução da mensagem) **então**

Início de execução do sucessor = Final de execução da mensagem.

Fim

4. Experiências

O tempo para transmissão de uma mensagem através do barramento foi definido como sendo 30ms e 0ms para as mensagens internas. Para as experiências, a cada bloco é atribuído um tempo de computação aleatório que varia entre o valor mínimo de 10ms e o valor máximo de 40ms para os blocos que têm que ser executados em processadores específicos e entre os valores 40ms e 100ms para aqueles blocos que podem executar em qualquer processador.

Os laços de controle usados variam desde os mais simples até os mais complexos para poder gerar as diferentes cargas computacionais. A definição de carga computacional procurou usar configurações e valores realistas, tirados da literatura. Foram considerados três tipos de cargas computacionais, chamadas de pequena, média e grande. São geradas selecionando aleatoriamente os laços de controle que formarão cada um destes tipos de carga. A carga pequena é formada por 2 laços de controle, a média composta de 4 laços e as cargas grandes são formadas por 5 laços de controle. Foram geradas aleatoriamente 24 estratégias pequenas e 100 estratégias médias e grandes. O algoritmo do Busca Tabu executava durante 3 minutos, e fornecia a melhor solução que tivesse encontrado.

Somente foram consideradas estratégias com laços com períodos iguais (macrociclo = período). As figuras 1, 2 e 3 mostram o número de soluções satisfatórias que a meta-heurística e a heurística simples encontraram nas diferentes cargas utilizadas. Como mostram as figuras de 1 a 3, o Busca Tabu mostrou-se melhor nos três tipos de carga computacional que foram usadas. A simplicidade desta meta-heurística a torna atrativa para o problema tratado neste artigo. Em geral, softwares comerciais utilizam alocação manual [Smar 2002] e escalonamento baseado em heurísticas simples, soluções inferiores ao uso do Busca Tabu.

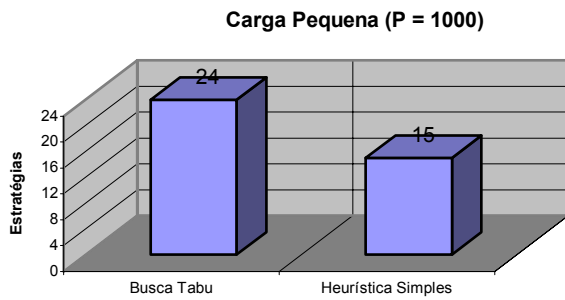


Figura 1. Resultados carga pequena.

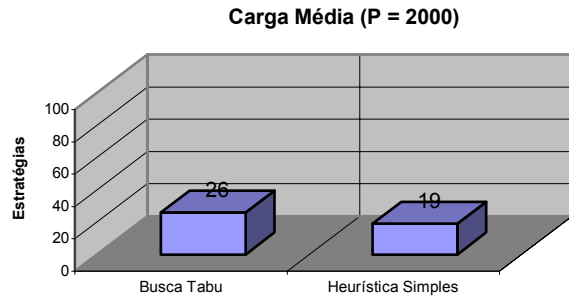


Figura 2. Resultados carga média.

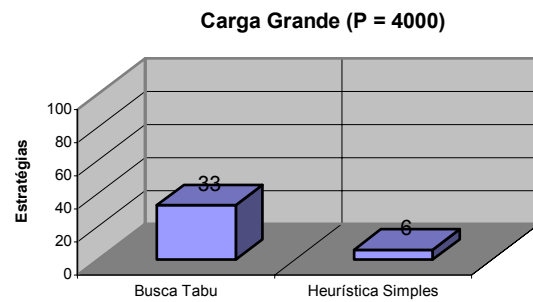
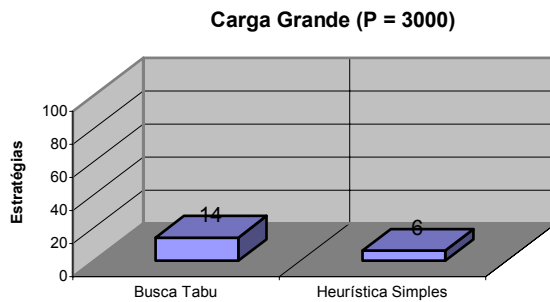


Figura 3. Resultados carga grande (com dois períodos diferentes).

5. Conclusões

Este artigo fez uma comparação entre o Busca Tabu e uma heurística simples na resolução do problema de alocação e escalonamento de blocos funcionais e mensagens periódicas do sistema *Foundation Fieldbus*. Foram usados diferentes cenários nas experiências realizadas e constatou-se que o Busca Tabu foi melhor que a heurística simples implementada nos três tipos de cargas definidos. Estas cargas foram geradas a partir da combinação de laços de controle realistas, tirados da literatura de controle. Os resultados relatados neste artigo, além de outros apresentados em [Vargas 2005], sugerem o Busca Tabu como uma forma eficaz de atacar o problema em questão. Pretende-se no futuro investigar se algoritmos mais complexos são capazes de ter a mesma eficácia do Busca Tabu. Também, pretende-se avaliar o impacto do tempo de execução do Busca Tabu na qualidade dos seus resultados.

Referências

- Fieldbus Foundation (2003), "Technical Overview", Austin Texas.
- Jonas, B. (2002), "Fieldbuses for Process Control: Engineering, Operation, and Maintenance", ISA – The Instrumentation, Systems, and Automation Society, 2002.
- Rafael, M. (2003), Procedimientos Metaheurísticos en Optimización Combinatoria, Matemáticas 1(1), 3-62.
- Smar (2002), "Function Blocks Instruction Manual".
- Vargas, L., "Estudo Sobre o Emprego de Meta-Heurísticas na Alocação e Escalonamento de Blocos Funcionais em Redes Foundation Fieldbus", Dissertação de Mestrado, PGEEL-UFSC, Fevereiro.
- Wei, Z., Michael, S.B. e Stephen, M.P. (2001), "Stability of Networked Control Systems", Control Systems Magazine, USA.