

Um Estudo Empírico de Métodos Hiper-Heurísticos

Igor Ribeiro Sucupira (Aluno de Mestrado)

Flávio Soares Corrêa da Silva (Orientador)

Departamento de Ciência da Computação

Instituto de Matemática e Estatística

Universidade de São Paulo

São Paulo – 2005

Meta-Heurísticas

- Definição resumida: "Uma meta-heurística é uma estrutura algorítmica geral que pode ser empregada na resolução de diversos problemas de otimização" (Metaheuristics Network - www.metaheuristics.net).
- Exemplos: *Simulated Annealing*, Algoritmos Genéticos, Busca Tabu, Reconexão de Caminhos, *Ant Colony Optimization*, Algoritmos de Estimação de Distribuição e muitas outras.

- Por que utilizar meta-heurísticas?
 - Idéias imediatas para a resolução de problemas novos.
 - Muitas vezes apresentam resultados excelentes quando enriquecidas com técnicas específicas para o problema.
- Características ligeiramente indesejáveis:
 - Em geral, se a implementação não é bem elaborada, os resultados são medíocres.
 - As implementações bem sucedidas freqüentemente se aproveitam de características específicas do problema.
- As hiper-heurísticas surgiram como uma tentativa de reduzir esses problemas.

Hiper-Heurísticas

- Definição:
 - Uma hiper-heurística é um algoritmo capaz de lidar com qualquer (?) problema de otimização, desde que receba informações e procedimentos específicos do problema a ser resolvido.
 - Essas informações e procedimentos devem ser facilmente construtíveis e a qualidade da heurística resultante deve depender o mínimo possível das características específicas do problema.
- Ideal: algoritmos facilmente reutilizáveis encontrando soluções "aceitáveis" para diversos problemas.

- O termo "hiper-heurística" surgiu no ano 2000. Desde então, alguns algoritmos já foram implementados: algoritmos genéticos, *simulated annealing*, *ant colony optimization* e outras idéias.
- Resultados muito bons, mas poucos autores realizaram experimentos que ajudassem a responder às perguntas:
 - Qual é o comportamento esperado da hiper-heurística quando aplicada a diversas instâncias do problema considerado? (experimentos com mais instâncias)
 - Como a hiper-heurística se comporta em problemas distintos?
 - Qual seria a diferença nos resultados se fosse implementado um algoritmo específico sofisticado? (experimentos com problemas bem conhecidos)

Uma Hiper-Heurística de Busca por Entornos

- Primeira hiper-heurística a ser implementada.
- Artigo elaborado em 2000 por Soubeiga et al.
- O algoritmo recebe:
 - Uma solução.
 - Um conjunto de operadores/heurísticas. Cada uma destas deve ser capaz de receber uma solução e produzir outra.
 - Um função capaz de avaliar as soluções.

- Cada iteração seleciona uma heurística para aplicar à solução atual.
- Heurística selecionada é aplicada repetidamente, até que não seja capaz de melhorar a solução.
- Função de decisão baseada em três critérios:
 1. Desempenho histórico de cada heurística.
 2. Desempenho histórico de cada seqüência com duas heurísticas.
 3. Tempo decorrido desde a última aplicação de cada heurística.
- Para cada heurística H_i :

$$f(H_i) = \alpha f_1(H_i) + \beta f_2(H_i, H_j) + \gamma f_3(H_i)$$

- α , β e γ são ajustados dinamicamente.
- f_1 e f_2 também levam em conta o consumo de tempo de cada heurística.
- Aplicação a três problemas de escalonamento (poucas instâncias).
- Principal resultado observado:
 - Funções de decisão triviais resultam em soluções muito piores.

Um Algoritmo Genético Hiper-Heurístico

- Desenvolvido por Cowling et al. em 2002.
- Recebe: as mesmas estruturas e abstrações recebidas pela hiper-heurística anterior.
- Cada cromossomo é uma seqüência de heurísticas. Exemplo:

H03 H01 H07 H08 H04 H04 H02

- Para avaliar um cromossomo, aplica-se a seqüência à melhor solução já encontrada, comparando-se o custo desta com o custo da solução resultante.

- A avaliação de um cromossomo leva em conta o tempo, para evitar cromossomos muito grandes na população.
- Operadores de cruzamento:
 - Cruzamento de ponto único.
 - Selecionar a melhor subsequência de heurísticas de cada cromossomo e trocar essas subsequências entre os cromossomos.
- Operadores de mutação:
 - Alterar cada gene, com baixa probabilidade, substituindo-o por uma nova heurística, aleatoriamente escolhida.
 - Remover do cromossomo a pior subsequência de heurísticas.
 - Selecionar a melhor subsequência de outro cromossomo e inseri-la em uma posição aleatória no cromossomo que sofre mutação.

- Aplicação a um problema de escalonamento. 5 instâncias geradas a partir de uma instância “verdadeira”.
- Alguns dos resultados observados:
 - A hiper-heurística é capaz de superar implementações simples de algoritmos genéticos e meméticos.
 - Seqüências de heurísticas selecionadas no fim da execução da hiper-heurística (em uma das instâncias) resultam em soluções superiores às obtidas pela aplicação repetida de uma das heurísticas. Resultados comparáveis aos obtidos pelas meta-heurísticas.

Objetivos

- Contribuir para o estudo das possibilidades de aperfeiçoamento das hiper-heurísticas existentes.
- Introduzir novos métodos hiper-heurísticos baseados em meta-heurísticas, contribuindo para o estudo da qualidade desse tipo de abordagem.
- Avaliar a qualidade das hiper-heurísticas de forma mais consistente, aplicando-as a problemas para os quais existam diversas instâncias disponíveis, já utilizadas por outros autores.

- Avaliar a robustez das hiper-heurísticas (característica muito importante), aplicando-as a problemas distintos.
- Contribuir para a comparação entre as hiper-heurísticas de busca direta por entornos e as evolutivas indiretas, abordagens que diferem significativamente entre si.

Desenvolvimento

- Serão experimentadas algumas alterações na hiper-heurística de busca por entornos de Soubeiga et al.
- Alterações também no algoritmo genético hiper-heurístico de Cowling et al.
- Será implementada um hiper-heurística baseada na meta-heurística de reconexão de caminhos.
- Outra implementação: um algoritmo de estimação de distribuição hiper-heurístico.

- Duas possibilidades para as hiper-heurísticas populacionais:
 - População inicial gerada aleatoriamente.
 - População inicial obtida durante uma execução da hiper-heurística de busca por entornos.

Experimentos com a Hiper-Heurística de Busca por Entornos

- O tempo de execução deixará de ser considerado durante a avaliação do resultado da aplicação de uma heurística.
- Serão armazenadas informações sobre seqüências com mais heurísticas:

$$f(H_i) = \alpha f_1(H_i) + \beta f_2(H_i, H_j) + \gamma f_3(H_i, H_j, H_k) + \delta f_4(H_i, H_j, H_k, H_l) + \psi f_5(H_i)$$

Estendendo o Algoritmo Genético Hiper-Heurístico

- Um operador de mutação adicional:
 - Otimização local até que o indivíduo atinja $\frac{6}{5}$ de seu tamanho original (ou até que um ótimo local seja encontrado).

S0 [H02 H03 H03 H03 H09 H01 H04 H08 H05 H01] **S1**

S1 [H07 H06] **S2**

[H02 H03 H03 H03 H09 H01 H04 H08 H05 H01 H07 H06]

Algoritmo de Estimação de Distribuição

- Cada indivíduo é uma seqüência de heurísticas.
- Em cada geração, produz-se toda a população aleatoriamente, de acordo com probabilidades definidas nas gerações anteriores.
- Estratégia baseada nas estruturas da hiper-heurística de busca por entornos de Soubeiga. Porém, não precisamos do fator de diversificação.

- Indivíduos são construídos heurística-por-heurística.

H02

H02 H04

H02 H04 H01

H02 H04 H01 H01

- Em cada passo, cada heurística H_i tem probabilidade:

$$u p(H_i) + v p(H_i, H_j) + w p(H_i, H_j, H_k) + z p(H_i, H_j, H_k, H_l)$$

Com $u, v, w, z \in (0, 1)$, $u + v + w + z = 1$ e $u < v < w < z$.

- Em seguida, cada indivíduo é estendido com otimização local.
- Inicialmente, todas as probabilidades são $\frac{1}{n}$, sendo n o número de heurísticas.

- No fim de cada geração, selecionam-se os N melhores indivíduos e atualizam-se as probabilidades de acordo com contagens realizadas nesses indivíduos:

$$p(H_i) = (1 - \alpha)p(H_i) + \alpha \frac{O_{H_i}}{T}$$

$$p(H_i, H_j) = (1 - \alpha)p(H_i, H_j) + \alpha \frac{O_{H_i H_j}}{O_{H_j} - F_{H_j}}$$

$$p(H_i, H_j, H_k) = (1 - \alpha)p(H_i, H_j, H_k) + \alpha \frac{O_{H_i H_j H_k}}{O_{H_j H_k} - F_{H_j H_k}}$$

$$p(H_i, H_j, H_k, H_l) = (1 - \alpha)p(H_i, H_j, H_k, H_l) + \alpha \frac{O_{H_i H_j H_k H_l}}{O_{H_j H_k H_l} - F_{H_j H_k H_l}}$$

- α é um valor dinâmico (crescente) em $(0, 1)$.

Hiper-Heurística de Reconexão de Caminhos

- Cada indivíduo é uma seqüência de heurísticas.
- Em cada geração, combina-se cada indivíduo com cada um dos demais, produzindo-se 6 indivíduos para cada par.
- Os indivíduos produzidos são estendidos com otimização local.
- Cada novo indivíduo será incluído na população (substituindo o pior) caso seja melhor que algum dos indivíduos desta.

I': H02 H03 H07 H01 H05 H11 H08

I'': H04 H09 H10 H01 H07

#H04# H03 H07 H01 H05 H11 H08

H04 #H09# H07 H01 H05 H11 H08

H04 H09 #H10# H01 H05 H11 H08

H04 H09 H10 H01 #H07# H11 H08

H04 H09 H10 H01 H07 H11 ##

H04 H09 H10 H01 H07 ##

I' ATINGIDO

H04 #H06# H10 H01 H07

- Quando um elemento for incluído na população, escolheremos um dos atuais componentes da população para ter seu tamanho reduzido (sem este passo, o tamanho médio dos indivíduos apenas cresceria).
- Redução de tamanho com o operador de mutação de Cowling et al.
- Avaliação de um indivíduo depende tanto do resultado de sua aplicação à melhor solução quanto do consumo de tempo desta aplicação (isto também contribui para evitar indivíduos muito extensos).

- Atualização dinâmica: combinações recomeçam sempre que a população é alterada.
- Se todos os indivíduos forem combinados e a população não se alterar, os $\lceil \frac{T-POP}{2} \rceil$ piores elementos serão removidos.
- Será mantido um conjunto extra de indivíduos, que substituirão os elementos removidos nessas ocasiões.
- Distância entre os indivíduos candidatos e os já presentes na população será considerada durante a (re)construção da população.

Metodologia

- Problemas:
 - Problema de empacotamento (*bin-packing*) unidimensional. Terceiro conjunto de instâncias de Scholl e Klein, que contém 10 instâncias bastante complexas (<http://www.wiwi.uni-jena.de/Entscheidung/binpp/index.htm>).
 - Problema de roteamento de veículos com restrições de capacidade. 15 instâncias propostas por Christofides e Eilon (<http://neo.lcc.uma.es/radi-aeb/WebVRP>).
 - Desafio ROADEF 2005. Conjunto A, com 16 instâncias (<http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005>).

- Para cada problema, todas as hiper-heurísticas partirão da mesma solução inicial, construída com um algoritmo facilmente implementável.
- Mesmo tempo de CPU (aproximadamente) para todas as execuções.

Mais Informações

O projeto pode ser obtido na minha página:
<http://www.ime.usp.br/~igorrs>