

1

Introdução

Problemas de otimização combinatória aparecem freqüentemente em vários setores da economia. Como exemplos desses problemas pode-se citar o projeto de redes de telecomunicação, a construção de agendas para tripulações, a produção de rotas eficientes para coleta de lixo, entre outros. Grande parte dos problemas de otimização combinatória são intratáveis por natureza ou são grandes o suficiente para tornar inviável o uso de algoritmos exatos. Nesses casos, métodos heurísticos são usualmente empregados. Esses métodos freqüentemente obtêm a solução ótima para um problema, porém nenhuma garantia de convergência é fornecida na maior parte dos casos.

GRASP (*Greedy Randomized Adaptive Search Procedure*) [43, 45, 48] é uma metaheurística de partidas múltiplas usada para obter soluções para problemas de otimização combinatória. Cada iteração é formada por duas etapas: uma fase construtiva, encarregada de produzir uma solução, e uma fase de busca local, que explora a vizinhança da solução construída, retornando um mínimo local. A melhor solução produzida durante as iterações será o resultado obtido pelo algoritmo.

A metaheurística GRASP tem sido usada para obter soluções de qualidade para muitos problemas de otimização combinatória [48]. Possui várias vantagens sobre outras metaheurísticas. Sua implementação é simples, usando-se algoritmos para construção de soluções e de busca local desenvolvidos para serem usados em outras abordagens. Além disso, o ajuste de parâmetros é mais simples em um procedimento GRASP do que em outras metaheurísticas.

Várias extensões e melhoramentos têm sido propostos para a metaheurística GRASP. Muitas dessas extensões consistem na hibridização do método com outras metaheurísticas. Em geral, verifica-se que abordagens híbridas possibilitam a obtenção de soluções de qualidade superior àquelas obtidas pelo algoritmo básico, tendo em alguns casos [47, 117] permitido melhorar as soluções conhecidas para problemas em aberto na literatura.

O religamento de caminhos incorporado à metaheurística GRASP têm

permitido a obtenção de soluções de qualidade superior às aquelas obtidas pelo algoritmo puro [7, 5, 27, 113, 117]. O procedimento de religamento de caminhos permite à metaheurística GRASP intensificar a busca em regiões onde soluções de qualidade tenham sido encontradas. A técnica consiste em explorar trajetórias que conectam soluções de alta qualidade, começando de uma *solução inicial* e gerando um caminho na vizinhança dessa solução na direção de outra solução, chamada de *solução guia*. Em geral, uma dessas soluções será produzida pela busca local da heurística e a outra será uma solução armazenada em um conjunto de soluções de elite.

Tanto a metaheurística GRASP pura, como a sua abordagem híbrida com religamento de caminhos, usam uma seqüência de números aleatórios durante a sua execução. Cada número aleatório dessa seqüência é usado para iniciar a fase construtiva do método em uma iteração. A fase construtiva é realizada por um algoritmo semi-guloso, onde a escolha de cada elemento inserido na solução é feita de forma randomizada, porém controlada por uma função gulosa. Dessa forma, para um número fixo de iterações, a qualidade da solução obtida pelo algoritmo GRASP puro ou pela abordagem híbrida com religamento de caminhos irá depender da seqüência de números aleatórios usada. Da mesma forma, fixando-se a qualidade de solução que deseja-se obter, o tempo de execução do algoritmo puro ou do algoritmo com religamento de caminhos também dependerá da seqüência de números aleatórios usada. Os resultados obtidos para a metaheurística GRASP dependem da semente inicial utilizada para a geração de números aleatórios diferentes. Esse fato leva à obtenção de conclusões imprecisas quando variantes do método desenvolvidas para um mesmo problema são comparadas.

A medida que tem sido observado o aparecimento de novas heurísticas para problemas de otimização combinatória, o uso de computadores paralelos também vêm se tornando mais freqüente. Visto que a maior parte dos problemas interessantes em otimização combinatória são problemas NP-difíceis e os tempos computacionais para obter soluções para esses problemas são elevados, é natural que algoritmos paralelos sejam considerados.

Uma preocupação freqüente durante o processo de paralelização de um programa seqüencial é como esse algoritmo irá comportar-se quando executado em paralelo. A situação ideal consiste em alcançar uma aceleração linear, isto é, que o tempo de execução do programa seqüencial, dividido pelo tempo de execução do programa paralelo executando em ρ processadores seja igual a ρ . Portanto, metodologias que auxiliem na análise do comportamento de uma estratégia paralela a partir do comportamento da sua versão seqüencial são de grande utilidade no processo de desenvolvimento de uma

aplicação paralela.

Nesse trabalho é proposta uma metodologia para análise do comportamento da metaheurística GRASP. Também são propostas estratégias de hibridização com o religamento de caminhos. Essas estratégias foram desenvolvidas para o problema de atribuição de três índices (AP3) e para o problema de escalonamento de tarefas conhecido na literatura como *job-shop scheduling problem* (JSP) e são analisadas de acordo com a metodologia proposta. A metodologia para análise do comportamento do método GRASP pode ser usada para prever de forma qualitativa como será a aceleração do algoritmo implementado em paralelo a partir da sua versão seqüencial. Essa propriedade da metodologia proposta será testada para estratégias paralelas desenvolvidas para AP3 e para JSP.

No Capítulo 2 a metaheurística GRASP é descrita. Além de explicar cada etapa do método, esse capítulo descreve os melhoramentos propostos para o algoritmo básico, tais como mecanismos alternativos usados na fase construtiva, o procedimento de GRASP Reativo, algoritmos híbridos e abordagens paralelas.

No Capítulo 3 é feita uma análise da distribuição da probabilidade do tempo necessário para a metaheurística GRASP obter uma solução com valor melhor ou igual a um valor alvo fornecido com entrada. Esse tempo computacional será chamado de tempo para valor alvo no restante desse trabalho. Esse capítulo baseia-se no estudo apresentado em [8]. Foram estudadas cinco abordagens da literatura, cujos códigos fontes estavam disponíveis. Uma metodologia padrão para análise gráfica é usada para comparar as distribuições empíricas e teóricas do tempo para valor alvo da heurística e para estimar os parâmetros dessas distribuições. Os resultados obtidos nesse capítulo descrevem uma consequência importante na paralelização do GRASP.

O Capítulo 4 descreve os trabalhos apresentados em [6, 5]. Novas formas de incorporar o religamento de caminhos em um algoritmo GRASP são propostas. Essas variantes do método estudado foram adaptadas para o problema de atribuição de três índices. O AP3 tem como aplicações o escalonamento de capitais de investimento, a atribuição de tropas militares, a otimização de cobertura por satélites, o escalonamento de práticas de ensino, a produção de placas de circuitos integrados e problemas em siderurgia [36, 51, 95, 94]. Nesse capítulo também é estudada a distribuição do tempo para valor alvo de todas as variantes de algoritmos GRASP com religamento de caminhos implementadas para o AP3.

No Capítulo 5 é proposta uma abordagem de GRASP com religamen-

to de caminhos para a obtenção de soluções aproximadas para o problema de escalonamento de tarefas JSP. Esse capítulo descreve os resultados apresentados em [7] e foi estendido em [17]. O procedimento de religamento de caminhos usado nessa abordagem é computacionalmente muito trabalhoso e por isso é realizado apenas para soluções que satisfaçam um critério de qualidade. Além disso, a fase construtiva é feita por duas funções gulosas de forma alternada, isto é, iterações ímpares do algoritmo GRASP usam uma função gulosa e iterações pares usam outra, o que produz uma maior diversidade de soluções iniciais para a busca local [27, 117]. A distribuição do tempo para valor alvo desse algoritmo é estudada nesse capítulo.

No Capítulo 6 são descritas duas estratégias de paralelização para abordagens de GRASP com religamento de caminhos. Na primeira estratégia, chamada de *independente*, os processos comunicam-se apenas para fazer a entrada de dados, a detecção do término do programa e a determinação da melhor solução global. A segunda abordagem, chamada de *cooperativa*, além de realizar a comunicação feita na abordagem independente, permite a troca de informações sobre soluções de qualidade obtidas durante as iterações do método. Os algoritmos propostos nos Capítulos 4 e 5 são paralelizados segundo as estratégias de paralelização propostas. A aceleração desses algoritmos paralelos é analisada e os resultados computacionais são comparados aos resultados teóricos estabelecidos no Capítulo 3.

Um problema de programação linear inteira com um número exponencial de restrições para alocação de custos de sistemas de transmissão é estudado no Capítulo 7. Esse problema de programação linear é resolvido por um esquema de geração de restrições violadas, implementado de forma seqüencial e paralela. A implementação paralela usa $\rho \geq 2$ processadores, onde um procedimento de *branch-and-bound* é executado por um processo, enquanto os $\rho - 1$ demais processos executam procedimentos de busca local. Nesses procedimentos de busca local, a cada vizinhança analisada o próximo movimento é escolhido de forma aleatória entre aqueles que produzem as melhores soluções. Técnicas usadas na fase construtiva da metaheurística GRASP para escolher o próximo elemento a ser inserido na solução são usadas para escolher o próximo movimento realizado pelos procedimentos de busca local. Mostra-se que os tempos para valor alvo desses procedimentos de busca local são distribuídos da mesma forma que os tempos para valor alvo da metaheurística GRASP.

No Capítulo 8 faz-se uma análise das contribuições desse trabalho e sugere-se pesquisas futuras e melhoramentos.