

5

Conclusões e Extensões

O objetivo desta tese foi o de desenvolver novas heurísticas para o problema de síntese de redes a 2-caminhos (2PNDP), analisando-se variantes e combinações de GRASP e reconexão por caminhos e comparando-se os resultados obtidos pelos algoritmos desenvolvidos com os descritos na literatura.

Uma heurística GRASP para 2PNDP foi proposta. Um algoritmo guloso e dois procedimentos de busca local foram implementados, adicionando-se um mecanismo de memória de muito curto prazo em um destes dois últimos métodos.

Na heurística GRASP foram introduzidas várias técnicas de reconexão por caminhos, que permitiram melhorar o desempenho do algoritmo original. Foi proposta uma nova técnica de reconexão por caminhos mista que examina duas trajetórias distintas de investigação de vizinhanças em direções opostas.

Experimentos computacionais extensivos foram feitos com variantes de GRASP e reconexão por caminhos para 2PNDP. Ilustrou-se que as estratégias de reconexão por caminhos permitem melhorar o desempenho do algoritmo puro, fazendo com que soluções de qualidade sejam encontradas mais rapidamente. Os resultados obtidos pelas execuções da heurística GRASP combinada com a nova estratégia de reconexão por caminhos mista ilustraram que esta variante é extremamente promissora, pois melhorou ainda mais os resultados alcançados pelas outras variantes.

Quando a variante da heurística GRASP combinada com reconexão por caminhos mista foi comparada ao algoritmo aproximado guloso da literatura, foi possível mostrar que houve uma melhoria significativa na qualidade das soluções encontradas. Além disto, a heurística GRASP foi aplicada a instâncias bem maiores, com 500 nós, 124750 arestas e 5000 pares origem-destino, enquanto que aquela apresentada em [25] foi aplicada a problemas com, no máximo, 120 nós, 7140 arestas e 60 pares origem-destino.

Foram descritas e avaliadas implementações paralelas independentes e colaborativas da heurística GRASP proposta para 2PNDP. Na estratégia independente, cada processador executa $\text{MaxIter}/p$ iterações, onde MaxIter e p são, respectivamente, o número total de iterações do procedimento GRASP e o número de processadores. Em estratégias colaborativas, os processadores que executam em paralelo trocam e compartilham informações coletadas ao longo da trajetória que cada um deles investiga. Espera-se não apenas acelerar a convergência para a melhor solução, mas, também, encontrar melhores soluções do que através de estratégias independentes. No caso da estratégia colaborativa básica, as informações compartilhadas entre os processadores são as soluções de elite encontradas por cada um deles.

Inicialmente, foram comparadas as estratégias independente e colaborativa básica, variando-se o número de processadores no ambiente paralelo. Os experimentos computacionais foram realizados em um *cluster* de 32 processadores Pentium II 400MHz com 32 Mbytes, usando a implementação Red Hat 6.2 do sistema operacional Linux 2.2.14-5.0 e o *switch* IBM 8274 com 96 portas com velocidade de 10 Mbits/s para a comunicação entre os processadores. Foi possível ilustrar que o desempenho da estratégia colaborativa básica melhora progressivamente, graças à execução de um menor número de iterações, à cooperação entre os processadores e ao aumento do número de processadores no ambiente paralelo. O desempenho da estratégia colaborativa com oito processadores já foi sensivelmente superior ao da independente. Quando foram utilizados 16 processadores, houve problemas de memória no *cluster*, devido ao elevado número de mensagens que chegam ao processador mestre.

Foram discutidas e implementadas variantes da estratégia colaborativa básica, que diminuem o número de soluções enviadas ao processador mestre. Ilustrou-se que as melhores estratégias são as que enviam a mesma quantidade de soluções ao mestre que a estratégia colaborativa básica, mas que verificam o custo da solução antes de enviá-la. A verificação do custo antes do envio da solução é imprescindível para prevenir problemas de memória no *cluster*.

A estratégia colaborativa básica acrescida da verificação do custo da solução antes do seu envio foi comparada à estratégia independente. Foi possível ilustrar que o desempenho da estratégia colaborativa melhora progressivamente em relação ao desempenho da estratégia independente para 2, 4, 8 e 16 processadores. O desempenho da estratégia colaborativa com 8 e 16 processadores já foi superior ao da independente. Com 32 processadores, ocorre uma sensível queda de desempenho da estratégia

colaborativa. Esta queda de desempenho se deve ao aumento do número de mensagens trafegando na rede, devido ao aumento considerável do número de processadores no ambiente paralelo. Para mostrar que esta queda de desempenho era devida exclusivamente às limitações do *cluster* e que o estudo realizado nesta tese mostra como superar estas limitações, as duas estratégias foram testadas em um outro *cluster* com mais recursos. Observou-se então que não houve queda de desempenho da estratégia colaborativa quando foram usados 32 processadores. Este fato ilustra que a quantidade de memória disponível nos processadores e a velocidade da rede de comunicação influem no desempenho dos algoritmos.

Uma extensão interessante seria estudar uma implementação hierárquica para 2PNDP e compará-la com as melhores estratégias paralelas. Uma estratégia hierárquica teria dois ou mais processadores mestres controlando grupos de processadores escravos. Os processadores escravos só poderiam trocar informações com seu respectivo mestre. Em intervalos periódicos de tempo (ou de iterações) os mestres podem atualizar seus conjuntos de soluções de elite. Esta estratégia paralela se mostra interessante na medida que diminui o processamento no mestre.

Uma outra extensão interessante seria avaliar o grau de dificuldade das instâncias usadas para comparar as variantes seqüenciais de GRASP e reconexão por caminhos e desenvolver-se geradores de instâncias difíceis, de forma similar ao estudo feito por Rosseti et al. [92] para o problema de Steiner em grafos.

Embora todas as comparações entre heurísticas seqüenciais e paralelas tenham sido realizadas para 2PNDP, acredita-se que conclusões semelhantes poderão ser obtidas para k PNDP, com $k > 2$. Portanto, seria interessante implementar e avaliar os melhores algoritmos seqüenciais e paralelos para k PNDP, com $k \geq 3$.