2
Trabalhos Relacionados

Diversos trabalhos já abordaram o problema de min-RWA. Algoritmos aproximativos foram propostos para topologias de rede em anéis [8, 55, 59], árvores [28, 29, 41, 50] e malhas [20, 78]. Erlebach e Jansem [30] provaram que min-RWA também é NP-Difícil para essas topologias. Esta tese concentra-se no estudo de heurísticas para min-RWA em topologias arbitrárias. Alguns trabalhos decompõem min-RWA em dois subproblemas: subproblema de roteamento e subproblema de atribuição de comprimentos de onda [7, 48, 63, 69, 70]. Outros tratam os dois problemas simultaneamente [64, 88]. Um *survey* com uma classificação funcional de heurísticas para min-RWA pode ser encontrado em [22]. As principais heurísticas na literatura são detalhadas a seguir.

2.1
Heurística de Bannerjee e Mukherjee

Na heurística proposta por Bannerjee e Mukherjee [7], min-RWA é decomposto em dois subproblemas. Primeiramente, resolve-se um problema de roteamento, que consiste em calcular uma rota para cada caminho ótico. Em seguida, resolve-se um problema de atribuição de comprimentos de onda, que consiste em atribuir um comprimento de onda para cada caminho ótico, utilizando o menor número possível de comprimentos de onda. Vale salientar que esta decomposição não é exata; a solução ótima dos dois subproblemas não garante a obtenção da solução ótima de min-RWA.

O problema de roteamento foi formulado como um problema de minimização do congestionamento da rede, ou seja, encontrar uma rota para cada caminho ótico de forma a minimizar o número máximo de caminhos óticos que usam um mesmo arco. Esta escolha se deve ao fato de que o congestionamento de um arco impõe um limite inferior ao número de comprimentos de onda necessários para multiplexar os caminhos óticos.

O problema de roteamento é modelado como um problema de multi-fluxos inteiros. Sua formulação, apresentada na Figura 2.1, utiliza variáveis binárias $f_{ab}^{i} = 1$ se o caminho ótico $t_i \in \mathcal{T}$ atravessa o arco $(a, b) \in A$; caso contrário $f_{ab}^{i} = 0$. A função objetivo consiste em minimizar o número f_{max}

PUC-Rio - Certificação Digital Nº 0410880/CA

de caminhos óticos que atravessam o enlace mais congestionado da rede. A equação (2-2) garante que este valor é maior ou igual ao número de caminhos óticos que passam por qualquer enlace da rede. As restrições de conservação de fluxo (2-3) obrigam que os caminhos óticos sejam contínuos e as restrições (2-4) estabelecem a integralidade das variáveis f_{ab}^i.

$$
\begin{aligned}
Minimizar \quad & f_{max} && && (2\text{-}1)\\
sujeito\ a \quad f_{max} \ &\geq \ \sum_{t_i \in \mathcal{T}} f_{ab}^i \quad \forall (a,b) \in A && && (2\text{-}2)\\
\sum_{a \in X} f_{ab}^i - \sum_{a \in X} f_{ba}^i \ &= \ \begin{cases} -1 & \text{se } b \text{ é o nó origem de } t_i \\ 1 & \text{se } b \text{ é o nó destino de } t_i \\ 0 & \text{caso contrário} \end{cases} && \begin{matrix} \forall b \in X, \\ \forall i \in \{1, ..., |\mathcal{T}|\} \end{matrix} && (2\text{-}3)\\
f_{ab}^i \in \{0,1\} \quad & \forall (a,b) \in A,\ \forall i \in \{1, ..., |\mathcal{T}|\} && && (2\text{-}4)
\end{aligned}
$$

Figura 2.1: Formulação do problema de roteamento.

Bannerjee e Mukherjee [7] propuseram uma heurística baseada em arredondamento aleatorizado [79] para resolver o problema de roteamento. O algoritmo consiste nos três passos seguintes: resolver a relaxação linear do problema, aplicar o procedimento chamado de *path stripping* para calcular um conjunto de possíveis rotas para cada caminho ótico, e selecionar uma rota para cada caminho ótico.

Primeiramente, relaxam-se as restrições de integralidade (2-4), permitindo-se valores fracionários para as variáveis f_{ab}^i, e resolve-se o problema de programação linear resultante. Dada a solução da relaxação linear, denota-se por $\bar{f}_{ab}^i$ a quantidade de fluxo do caminho ótico $t_i \in \mathcal{T}$ que passa pelo arco $(a,b) \in A$.

Em seguida, o procedimento *path stripping* baseia-se no fluxo fracionário de cada caminho ótico $t_i \in \mathcal{T}$ para calcular um conjunto de rotas P_i no qual t_i possa ser roteado. Este procedimento também associa um peso a cada rota, cujo valor será usado na etapa seguinte. Ele pode ser descrito em três passos que são executados para cada caminho ótico $t_i \in \mathcal{T}$. Inicialmente, o conjunto de rotas P_i está vazio:

(i) Calcular uma rota p entre os nós terminais do caminho ótico t_k, usando apenas arcos $(a,b) \in A$ tais que $\bar{f}_{ab}^i > 0$.

(ii) Seja $\sigma = min_{(a,b) \in A} \bar{f}_{ab}^i$, subtrair σ de $\bar{f}_{a'b'}^i$, para todos os arcos (a',b') na rota p, e inserir p em P_i com peso igual a σ.

PUC-Rio - Certificação Digital Nº 0410880/CA

(iii) Se ainda existir uma rota entre os nós terminais de t_i usando apenas arcos $(a, b) \in A$ tais que $\bar{f}^i_{ab} > 0$, voltar ao passo 1. Caso contrário, tratar o próximo caminho ótico.

Finalmente, uma única rota em P_i é selecionada aleatoriamente para cada caminho ótico $t_i \in \mathcal{T}$, com probabilidades proporcionais aos pesos atribuídos pelo procedimento de *path stripping*. Uma vez que as rotas estão fixadas, constrói-se um grafo de conflitos $G = (V, E)$ e atribui-se um comprimento de onda para cada caminho ótico resolvendo-se um problema coloração de vértices, como descrito no Capítulo 1. Esta etapa é repetida um número determinado de vezes. Após cada uma delas, o problema de atribuição de comprimentos de onda é resolvido. Ao final do algoritmo, a melhor solução encontrada é retornada.

O problema de coloração de vértices é NP-Difícil [40]. Qualquer heurística para GCP [11, 21, 38, 42, 46, 89] pode ser usada para a atribuição dos comprimentos de onda. Bannerjee e Mukherjee utilizaram a heurística `Smallest-Last` [66]. Esta heurística utiliza uma pilha para ordenar os vértices do grafo. A ordenação é realizada da seguinte forma: seleciona-se o vértice de menor grau, que é removido de G e colocado na pilha, repetindo-se o processo até que todos os vértices estejam empilhados. Em seguida, os vértices são coloridos de acordo com a ordem em que são retirados da pilha, ou seja, começando com o último vértice empilhado e terminando com o primeiro. Os vértices são coloridos com a menor cor que não é usada para colorir seus vizinhos (cores são associadas a números).

2.2 Heurística de Hyytiä e Virtamo

As heurísticas de Hyytiä e Virtamo para min-RWA são apresentadas em dois trabalhos [48, 49]. Primeiramente, eles tratam min-RWA no caso onde as rotas para os caminhos óticos são conhecidas a priori. Neste caso, min-RWA reduz-se ao problema GCP num grafo de conflitos, como o descrito no Capítulo 1. Quando as rotas não são fixadas a priori, Hyytiä e Virtamo sugerem um algoritmo para calcular e fixar uma rota para cada caminho ótico. A escolha das rotas é baseada nos seguintes princípios:

(i) Caminhos óticos que atravessam um único arco podem usar qualquer comprimento de onda livre neste arco.

(ii) Rotas mais longas podem levar a soluções com um número maior de comprimentos de onda, pois reservam comprimentos de onda em vários

PUC-Rio - Certificação Digital Nº 0410880/CA

arcos. Portanto, é preferível rotear os caminhos óticos por rotas mais curtas.

(iii) O número de caminhos óticos que atravessam um mesmo arco define um limite inferior para o número total de comprimentos de onda.

Assume-se em [48] que um conjunto de rotas candidatas P_i é fornecido para cada caminho ótico $t_i \in \mathcal{T}$. Este conjunto pode ser construído aleatoriamente ou por um algoritmo de k-caminhos mais curtos. O conjunto de rotas candidatas é filtrado de acordo com dois parâmetros que devem ser ajustados: o número máximo r de rotas candidatas para cada caminho ótico t_i e a diferença máxima d entre o número de arcos de uma rota em P_i e o número de arcos na rota mais curta entre os vértices terminais de t_i em N. Denota-se por P_i' as r rotas mais curtas em P_i que têm no máximo d arcos a mais que a rota mais curta de t_i em N.

O algoritmo de roteamento proposto em [48] seleciona aleatoriamente uma rota de P_i' para cada caminho ótico $t_i \in \mathcal{T}$. Uma vez que as rotas estão selecionadas, constrói-se um grafo de conflitos e atribui-se um comprimento de onda para cada caminho ótico resolvendo-se um problema de coloração de vértices, como descrito no Capítulo 1. O número de comprimentos de onda utilizados pela heurística `DSATUR` [11] é usado como uma estimativa da qualidade das rotas que foram selecionadas. Este procedimento é executado por um número fixo de iterações. Ao final, as rotas selecionadas na iteração onde a heurística `DSATUR` utilizou o menor número de comprimentos de onda são fixadas e aplica-se uma nova heurística para atribuir os comprimentos de onda definitivamente.

Hyytiä e Virtamo avaliaram diversas heurísticas para resolver o problema de atribuição de comprimentos de onda. Duas heurísticas gulosas foram analisadas. A heurística `Largest-Fit` [91] primeiro ordena os vértices do grafo em ordem decrescente de seus graus. Em seguida, atribui seqüencialmente a menor cor disponível para cada vértice. A heurística `DSATUR` [11] segue um princípio diferente. Define-se *color-degree* como o número de cores diferentes usadas para colorir os vizinhos de um vértice e *uncolored-degree* como o número de vizinhos não coloridos de um vértice. `DSATUR` seleciona repetidamente o vértice com maior *color-degree* e atribui a ele a menor cor disponível. Se dois vértices têm o mesmo *color-degree*, aquele com maior *uncolored-degree* é colorido primeiro. Hyytiä e Virtamo também estudaram heurísticas baseadas em *simulated annealing* [81, 82], algoritmos genéticos [82] e busca tabu [42, 46], além de um algoritmo de busca exaustiva [48].

PUC-Rio - Certificação Digital Nº 0410880/CA

Os experimentos computacionais demonstraram que a heurística de busca tabu obteve os melhores resultados. O procedimento, aqui chamado de `TabuGCP`, é basicamente uma busca local, onde os movimentos que aparentemente levam a soluções já visitadas são proibidos por um certo número de iterações. O objetivo do algoritmo é encontrar uma coloração válida com C cores. No entanto, `TabuGCP` pode ser usado num procedimento iterativo, onde sucessivamente o valor de C é decrementado até que não seja mais possível encontrar uma solução com um número menor de cores.

`TabuGCP` parte de uma solução inviável com C cores, onde vértices adjacentes podem estar coloridos com a mesma cor. A vizinhança de uma solução S é formada por todas as soluções que podem ser construídas trocando-se a cor de um dos vértices de S. O procedimento consiste em minimizar o número de pares de vértices adjacentes coloridos com a mesma cor. `TabuGCP` é interrompido quando uma solução viável é encontrada, ou quando um determinado número de iterações é executado sem que uma solução viável tenha sido encontrada.

2.3 Heurística de Li e Simha

Li e Simha [63] propuseram uma outra heurística de decomposição para min-RWA. Primeiramente, as k rotas mais curtas para cada caminho ótico são calculadas, onde k é um parâmetro a ser ajustado. Em seguida, uma rota (dentre as pré-calculadas) e um comprimento de onda são atribuídos a cada caminho ótico, resolvendo-se um Problema de Coloração de Partições num grafo de conflitos particionado, como descrito no Capítulo 1.

Foram propostas seis heurísticas construtivas para PCP, baseadas em heurísticas clássicas para GCP. Li e Simha [63] basearam seu estudo em três heurísticas conhecidas para GCP: `Smallest-Last` (SL) [66], descrita na Seção 2.1, `Largest-Fit` (LF) [91] e `DSATUR`[11], as duas últimas descritas na Seção 2.2. As heurísticas para PCP utilizam os seguintes princípios: (i) os vértices de maior grau em cada componente da partição são em geral mais difíceis de colorir que os vértices de menor grau, e (ii) quanto menor o número de vértices no grafo particionado, mais próxima a solução da heurística estará da solução ótima. As heurísticas propostas podem ser divididas em dois grupos: *heurísticas de uma fase* e *heurísticas de duas fases*. Nos dois grupos, as arestas entre vértices na mesma componente da partição são removidas, já que somente um vértice em cada componente é colorido.

As heurísticas de uma fase são extensões das mencionadas na seção anterior, adaptadas para o PCP. As novas heurísticas são chamadas de `onestepLF`

PUC-Rio - Certificação Digital Nº 0410880/CA

(*one step Largest-Fit*), `onestepSL` (*one step Smallest-Last*) e `onestepCD` (*one step Color-Degree*).

A heurística `onestepLF` é uma variante da heurística `Largest-Fit` [91]. Ela recebe um grafo particionado como entrada e repete os seguintes passos, até que todas as componentes tenham exatamente um vértice colorido: (i) identificar o vértice não empilhado de cada componente com o menor grau, (ii) dentre estes vértices, selecionar aquele com maior grau, (iii) atribuir ao vértice selecionado a menor cor que não é usada por nenhum dos seus vizinhos, e (iv) remover os demais vértices na mesma componente.

A heurística `onestepSL` é uma variante da heurística `Smallest-Last` [66]. Ela recebe um grafo particionado como entrada e inicializa uma pilha de vértices B. Em seguida, repete os seguintes passos até que um vértice de cada componente seja empilhado: (i) encontrar o vértice não colorido de cada componente com o menor grau, (ii) selecionar dentre estes vértices aquele com maior grau, (iii) colocar o vértice selecionado no topo de B, e (iv) remover os demais vértices na mesma componente. Por fim, `onestepSL` colore iterativamente os vértices na ordem em que são retirados de B, com a menor cor que não é usada para colorir seus vizinhos.

A heurística `onestepCD` é uma variante da heurística `DSATUR` [11]. Ela recebe um grafo particionado como entrada e repete os seguintes passos até que todas as componentes tenham exatamente um vértice colorido: (i) encontrar o vértice não colorido de menor *color-degree* de cada componente, escolhendo em caso de empate aquele com menor *uncolored-degree*, (ii) selecionar dentre estes aquele com maior *color-degree*, selecionando em caso de empate aquele com maior *uncolored-degree*, (iii) atribuir ao vértice selecionado a menor cor que não é usada por nenhum dos seus vizinhos, e (iv) remover os demais vértices na mesma componente do vértice recém colorido.

As heurísticas de duas fases são `twostepLF` (*two step Largest-Fit*), `twostepSL` (*two step Smallest-Last*) e `twostepCD` (*two step Color-Degree*). Na primeira fase, seleciona-se um vértice em cada componente da partição e na segunda fase atribui-se uma cor a cada vértice selecionado. O processo de seleção dos vértices ocorre da seguinte forma: escolhe-se o vértice com o menor grau dentre aqueles que estão em componentes que contêm mais de um vértice, removem-se os demais vértices da mesma componente e repete-se o procedimento até que todas as componentes tenham apenas um único vértice. Ao final desta fase, as heurísticas `twostepLF`, `twostepSL` e `twostepCD` aplicam as heurísticas `Largest-Fit`, `Smallest-Last` e `DSATUR`, respectivamente, no grafo residual.

Li e Simha [63] realizaram uma série de experimentos computacionais

PUC-Rio - Certificação Digital Nº 0410880/CA

para avaliar o desempenho dessas heurísticas construtivas, mostrando que os melhores resultados foram obtidos por `onestepCD`.

2.4 Heurística de Manohar, Manjunath e Shevgaonkar

Esta seção apresenta a heurística `Greedy-EDP-RWA` proposta por Manohar, Manjunath e Shevgaonkar [64]. Ela parte do princípio de que quanto mais caminhos óticos utilizarem o mesmo comprimento de onda, menor será o número total de comprimentos de onda utilizados. Para que dois caminhos óticos possam utilizar o mesmo comprimento de onda, suas rotas não podem compartilhar um mesmo arco de N.

O problema de encontrar o maior subconjunto de caminhos óticos disjuntos em arcos - max-EDP (do inglês *Maximum Edge Disjoint Path Problem*) é NP-Difícil [54]. A heurística `BGAforEDP` para max-EDP [54] é apresentada na Figura 2.2. Os dados de entrada de `BGAforEDP` são o grafo N, o conjunto $\mathcal{T}$ de requisições de caminhos óticos, um vetor $\pi = [\ \pi(1), \ldots, \pi(|\mathcal{T}|)\ ]$ descrevendo a ordem na qual os caminhos óticos são considerados ($\pi(i) \in \{1, \ldots, |\mathcal{T}|\}$ e $\pi(i) \neq \pi(j)$, para todo $i, j = 1, \ldots, |\mathcal{T}|$), e o valor d do número máximo de arcos em cada rota. Kleinberg [54] sugere que o valor de d seja o máximo entre o diâmetro de N e a raiz quadrada do número de enlaces na rede. A heurística retorna o conjunto P de pares $(t_{\pi(i)}, p_{\pi(i)})$, onde $t_{\pi(i)} \in \mathcal{T}$ é um caminho ótico e $p_{\pi(i)}$ é sua rota. Na linha 1, o conjunto P é inicializado. O laço das linhas 2 a 9 é executado uma vez para cada caminho ótico em $\mathcal{T}$. Na linha 3, calcula-se o caminho mais curto $p_{\pi(i)}$ entre os nós terminais de $t_{\pi(i)}$. A linha 4 verifica se $p_{\pi(i)}$ atravessa no máximo d enlaces. Neste caso, o conjunto P é atualizado na linha 6 e todos os arcos de $p_{\pi(i)}$ são removidos da rede N na linha 7, para evitar que os próximos caminhos óticos analisados utilizem algum desses arcos. Por fim, o conjunto P é retornado na linha 10.

A heurística `Greedy-EDP-RWA` [64] para min-RWA é descrita a seguir. Primeiramente, aplica-se `BGAforEDP` e obtém-se um subconjunto de caminhos óticos $T_1 \subseteq \mathcal{T}$, cujas rotas são disjuntas em arcos. Em seguida, os caminhos óticos em T_1 são associados ao mesmo comprimento de onda ω_1. Novamente, aplica-se `BGAforEDP` para o conjunto de caminhos óticos restantes ($\mathcal{T} \setminus T_1$) e obtém-se um subconjunto T_2, cujos caminhos óticos são associados ao mesmo comprimento de onda ω_2. Este procedimento é repetido até que todos os caminhos óticos recebam uma rota e um comprimento de onda.

O pseudocódigo da heurística `Greedy-EDP-RWA` é apresentado na Figura 2.3. A heurística retorna um conjunto S de pares formados pela rota p_i e pelo comprimento de onda w_i atribuídos a cada caminho ótico $t_i \in \mathcal{T}$. Na

PUC-Rio - Certificação Digital Nº 0410880/CA

Procedimento `BGAforEDP`$(N, \mathcal{T}, d, \pi)$
1. $P \leftarrow \emptyset$;
2. **para** $i = 1, ..., |\mathcal{T}|$ **faça**
3. Calcule o caminho mais curto $p_{\pi(i)}$ em N entre os nós terminais de $t_{\pi(i)}$;
4. **se** $p_{\pi(i)}$ não atravessa mais do que d enlaces
5. **então faça;**
6. $P \leftarrow P \cup (t_{\pi(i)}, p_{\pi(i)})$;
7. Remova de N os arcos em $p_{\pi(i)}$;
8. **fim-se**;
9. **fim-para**;
10. **retorne** P;

fim `BGAforEDP`

Figura 2.2: Pseudocódigo do procedimento `BGAforEDP`.

linha 1, são inicializados o conjunto S e o contador de comprimentos de onda j. O laço das linhas 2 a 12 é executado até que todos os caminhos óticos recebam uma rota e um comprimento de onda. O valor de j é incrementado na linha 4. Na linha 5, constrói-se um vetor $\pi = [\ \pi(1), \ldots, \pi(|\mathcal{T}|)\ ]$ descrevendo uma ordenação aleatória dos caminhos óticos em $\mathcal{T}$ ($\pi(i) \in \{1, \ldots, |\mathcal{T}|\}$ e $\pi(i) \neq \pi(j)$, para todo $i, j = 1, \ldots, |\mathcal{T}|$). A heurística `BGAforEDP` é executada na linha 6 e retorna o conjunto P de pares (t_i, p_i), onde $t_i \in \mathcal{T}$ é um caminho ótico e p_i é sua rota. O laço das linhas 7 a 11 é executado para cada par $(t_i, p_i) \in P$. O caminho ótico t_i é associado ao j-ésimo comprimento de onda na linha 8 e o par (p_i, w_i) é adicionado à solução parcial corrente na linha 9. Na linha 10, o caminho ótico t_i é retirado do conjunto $\mathcal{T}$ e uma nova iteração é iniciada. Por fim, o conjunto S é retornado na linha 13.

A heurística `Greedy-EDP-RWA` pode apresentar soluções diferentes, dependendo da ordenação aleatória dos caminhos óticos gerada na linha 5. Em virtude disto, `Greedy-EDP-RWA` é usada dentro de um procedimento multi-partida. O procedimento, aqui chamado de multi-partida `Greedy-EDP-RWA`, consiste em executar a heurística `Greedy-EDP-RWA` um certo número de vezes, variando a permutação aleatória dos caminhos óticos. A melhor solução encontrada ao final de todas as iterações é retornada. Os resultados computacionais obtidos com esta heurística foram comparados com a heurística proposta por Bannerjee e Mukherjee [7]. A heurística multi-partida `Greedy-EDP-RWA` mostrou-se mais rápida e obteve soluções de mesma qualidade.

2.5

PUC-Rio - Certificação Digital Nº 0410880/CA

Procedimento `Greedy-EDP-RWA`$(N, \mathcal{T}, d)$
1. Faça $S \leftarrow \emptyset$ e $j \leftarrow 0$;
2. **enquanto** $\mathcal{T} \neq \emptyset$ **faça**
4. $j \leftarrow j + 1$;
5. Seja π uma ordenação aleatória dos elementos em $\mathcal{T}$;
6. $P \leftarrow$ `BGAforEDP`$(N, \mathcal{T}, d, \pi)$;
7. **para todo** $(t_i, p_i) \in P$ **faça**
8. Faça $w_i \leftarrow \omega_j$;
9. $S \leftarrow S \cup (p_i, w_i)$;
10. $\mathcal{T} \leftarrow \mathcal{T} \setminus \{t_i\}$;
11. **fim-para**;
12. **fim-enquanto**;
13. **retorne** S;

fim `BGAforEDP`

Figura 2.3: Pseudocódigo do procedimento `Greedy-EDP-RWA`.

PUC-Rio - Certificação Digital Nº 0410880/CA

Heurística de Noronha e Ribeiro

Noronha e Ribeiro [70, 69] utilizaram a mesma estratégia de decomposição de Li e Simha [63], mas utilizaram diferentes algoritmos nas duas fases. A primeira consiste em calcular um conjunto de k rotas alternativas para cada caminho ótico, usando o procedimento construtivo k-`EDR` descrito na Seção 2.5.1. A segunda fase consiste em escolher uma rota (dentre aquelas pré-calculadas) e atribuir um comprimento de onda a cada caminho ótico através de uma busca tabu para PCP. As duas fases são descritas em detalhes a seguir.

2.5.1 Escolha das Rotas

O procedimento k-`EDR` calcula um conjunto de até k rotas alternativas para cada caminho ótico em $\mathcal{T}$. O princípio básico de k-`EDR` é rotear o maior número possível de caminhos óticos através de rotas disjuntas, o que permite um maior aproveitamento dos comprimentos de onda disponíveis. Ela é baseada na heurística `BGAforEDP` descrita na Seção 2.4.

O pseudocódigo da heurística k-`EDR` é descrito na Figura 2.4. Os dados de entrada de `BGAforEDP` são o valor de k, o grafo N, o conjunto $\mathcal{T}$ de requisições de caminhos óticos e o valor d do número máximo de arcos em cada rota. Como sugerido em Kleiberg [54], o valor de d é fixado no máximo entre o diâmetro de N e a raiz quadrada do número de enlaces na rede. A heurística retorna o conjunto P de pares $(t_{\pi(i)}, p_{\pi(i)})$, onde $t_{\pi(i)} \in \mathcal{T}$ é um caminho ótico e $p_{\pi(i)}$ é uma das rotas alternativas de $t_{\pi(i)}$. Na linha 1, o conjunto

P de rotas alternativas é inicializado. O laço das linhas 2 a 17 constrói até k rotas alternativas para cada caminho ótico em $\mathcal{T}$. Uma cópia $\mathcal{T}'$ de $\mathcal{T}$ é construída na linha 3. O laço das linhas 4 a 16 calcula uma única rota para cada caminho ótico. Uma cópia N' de N é construída na linha 5. Na linha 6, constrói-se um vetor $\pi = [\pi(1), \ldots, \pi(|\mathcal{T}|)]$ descrevendo uma ordenação aleatória dos caminhos óticos em $\mathcal{T}$ ($\pi(i) \in \{1, \ldots, |\mathcal{T}|\}$ e $\pi(i) \neq \pi(j)$, para todo $i, j = 1, \ldots, |\mathcal{T}|$). O laço das linhas 7 a 15 é repetido para cada caminho ótico em $\mathcal{T}'$. O caminho mais curto $p_{\pi(i)}$ entre os nós terminais do caminho ótico $t_{\pi(i)}$ no grafo N' é calculado na linha 8. A linha 9 verifica se $p_{\pi(i)}$ atravessa menos de d enlaces. Neste caso, os conjuntos $\mathcal{T}'$ e P são atualizados nas linhas 11 e 12, respectivamente. Na linha 15, todos os arcos de $p_{\pi(i)}$ são removidos do grafo N', para evitar que o próximo caminho ótico analisado utilize algum desses arcos. O conjunto P é retornado na linha 18. Este conjunto é utilizado para construir um grafo de conflitos particionado $G_P = (V, E, Q)$, como descrito no Capítulo 1.

PUC-Rio - Certificação Digital Nº 0410880/CA

Procedimento k-EDR$(k, N, \mathcal{T}, d)$
1. $P \leftarrow \emptyset$;
2. **para** $j = 1, \ldots, k$ **faça**
3. Faça $\mathcal{T}' \leftarrow \mathcal{T}$;
4. **enquanto** $\mathcal{T}' \neq \emptyset$ **faça**
5. Faça $N' \leftarrow N$;
6. Seja π uma ordenação aleatória dos elementos em $\mathcal{T}'$;
7. **para** $i = 1, ..., |\mathcal{T}'|$ **faça**
8. Calcule o caminho mais curto $p_{\pi(i)}$ em N' entre os nós terminais do caminho ótico $t_{\pi(i)}$;
9. **se** $p_{\pi(i)}$ não tem mais do que d arestas
10. **então faça;**
11. $\mathcal{T}' \leftarrow \mathcal{T}' \setminus \{t_{\pi(i)}\}$;
12. $P \leftarrow P \cup \{(t_{\pi(i)}, p_{\pi(i)})\}$;
13. Remova os arcos em $p_{\pi(i)}$ de N';
14. **fim-se**;
15. **fim-para**;
16. **fim-enquanto**
17. **fim-para**;
18. **retorne** P;

fim k-EDR

Figura 2.4: Pseudocódigo do procedimento k-EDR.

2.5.2

Busca Tabu para Coloração de Partições

Noronha e Ribeiro [70, 69] propuseram a heurística de busca tabu `TS-PCP` para colorir G_P. A heurística baseia-se na busca tabu proposta por Hertz e de Werra [46] para coloração de vértices. `TS-PCP` parte de uma solução inicial S com C cores, construída pela heurística `onestepCD`. A viabilidade de S é destruída para gerar uma solução S' com $C-1$ cores. Define-se um *conflito de coloração* como um par de vértices adjacentes coloridos com a mesma cor e `conflitos`(S') como o número de conflitos de coloração na solução S'. `TS-PCP` realiza uma busca tabu [42] a partir de S' em busca de uma solução viável com o mesmo número de cores. Se esta solução for encontrada, o algoritmo é reiniciado, buscando reduzir mais uma cor. O procedimento é repetido até que não seja mais possível encontrar uma solução viável com um número menor de cores.

`TS-PCP` é baseada numa busca local usando a vizinhança 1-opt. Cada vizinho de uma solução S' com $C-1$ cores é gerado através de um movimento (v, ω) que consiste em selecionar um vértice $v \in V$ e atribuir-lhe a cor ω, onde ω varia entre 1 e $C-1$. Para acelerar a convergência desse algoritmo, somente os vértices $v \in V$ envolvidos em conflitos de coloração são considerados.

O pseudocódigo de `TS-PCP` é descrito na Figura 2.5. Nas linhas 1 e 2, constrói-se uma nova solução S' a partir de S, colorindo-se aleatoriamente, com uma das cores $1, \ldots, C-1$, todos os vértices coloridos originalmente com a cor C. Como a solução S' provavelmente contém conflitos de coloração, os próximos passos procuram restaurar sua viabilidade. A lista tabu é inicializada na linha 3. O laço das linhas 4-9 é executado até que um certo número de movimentos tenham sido executados desde a última vez que uma solução viável foi encontrada. Uma busca local na vizinhança de S' é executada na linha 5, terminando quando um ótimo local é encontrado. Os vizinhos que pertencem à lista tabu e não satisfazem os critérios de aspiração não são considerados. Os critérios de aspiração permitem que o melhor vizinho de S' seja retirado da lista tabu quando ele tem menos conflitos de coloração do que a melhor solução conhecida ou quando todos os vizinhos de S' estão na lista tabu. Se a nova solução S' não tem conflitos de coloração, a melhor solução viável S é atualizada e uma nova tentativa de diminuir mais uma cor é iniciada a partir da linha 6. Caso contrário, `TS-PCP` executa o movimento (v, w) em S', movendo-se para o melhor vizinho de S' na linha 7. Como S' é um ótimo local, a solução resultante pode conter mais conflitos de coloração que S'. Portanto, o movimento (v, w) é inserido na lista tabu por um certo número de iterações na linha 8, impedindo que o procedimento `TS-PCP` retorne à solução corrente na próxima vez em que a busca local for executada. A última solução viável

PUC-Rio - Certificação Digital Nº 0410880/CA

encontrada S é retornada na linha 10.

Procedimento `TS-PCP`(G_P, S)
1. Seja C o número de cores em S;
2. Construa uma nova solução S' a partir de S com $C - 1$ cores;
3. Inicialize a lista tabu τ;
4. **enquanto** o critério de parada não seja satisfeito **faça**
5. $\quad S' \leftarrow$ `LocalSearch`(S', τ);
6. $\quad$ **se** `conflitos`$(S') = 0$ **então faça** $S \leftarrow S'$ e retorne à linha 1;
7. $\quad$ Efetue o movimento (v, w) que resulta no melhor vizinho de S';
8. $\quad$ Insira o par (v, w) em τ por `tabuTenure` iterações;
9. **fim-enquanto**
10. **retorne** S;

fim `TS-PCP`.

Figura 2.5: Pseudocódigo do procedimento `TS-PCP`.

PUC-Rio - Certificação Digital Nº 0410880/CA

2.6 Heurística de Skorin-Kapov

As melhores heurísticas disponíveis para min-RWA até o momento foram propostas por Skorin-Kapov [88]. Cada comprimento de onda é tratado por uma cópia do grafo $N = (X, A)$ que representa a topologia da rede. Caminhos óticos roteados na mesma cópia de N, usando rotas disjuntas em arcos, são multiplexados no mesmo comprimento de onda.

Neste trabalho, observou-se a semelhança entre min-RWA e o Problema de Empacotamento Unidimensional - BPP (do inglês *Bin Packing Problem*) [3, 52]. Dados um conjunto de itens com seus respectivos pesos e um conjunto de caixas (*bins* em inglês) com capacidade limitada, o BPP consiste em distribuir o conjunto de itens no menor número possível de caixas, de forma que a soma dos pesos dos itens em cada caixa não ultrapasse sua capacidade. Os comprimentos de onda são vistos como as caixas e os caminhos óticos como os itens do BPP. O min-RWA é então reformulado como o problema de distribuir o conjunto de caminhos óticos (itens) no menor número possível de comprimentos de onda (caixas).

Quatro heurísticas para min-RWA baseadas em heurísticas clássicas para BPP foram propostas: (i) `FF-RWA`, baseada na heurística First Fit, (ii) `BF-RWA`, baseada na heurística Best Fit, (iii) `FFD-RWA`, baseada na heurística First Fit Decreasing, e (iv) `BFD-RWA`, baseada na heurística Best Fit Decreasing. Observou-se em [88] que a primeira é equivalente à heurística Greedy-EDP-RWA [64], descrita na seção anterior, exceto pela ordem em que algumas operações são executadas.

O pseudocódigo das heurísticas `FF-RWA` e `BF-RWA` é apresentado na Figura 2.6. Elas se diferenciam apenas pela linha 8, conforme descrito a seguir. Os dados de entrada são o grafo N, o conjunto $\mathcal{T}$ de requisições de caminhos óticos, um vetor $\pi = [\pi(1), \ldots, \pi(|\mathcal{T}|)]$ descrevendo a ordem na qual os caminhos óticos são considerados ($\pi(i) \in \{1, \ldots, |\mathcal{T}|\}$ e $\pi(i) \neq \pi(j)$, para todo $i, j = 1, \ldots, |\mathcal{T}|$), e o valor d do número máximo de arcos em cada rota. Como sugerido em [54], o valor de d é igual ao máximo entre o diâmetro de N e a raiz quadrada do número de enlaces na rede. As heurísticas retornam um conjunto S de pares $(p_{\pi(i)}, w_{\pi(i)})$, para $i = 1, \ldots, |\mathcal{T}|$, onde $p_{\pi(i)}$ é a rota associada com o caminho ótico $t_{\pi(i)}$ e $w_{\pi(i)}$ é o comprimento de onda com o qual ele é multiplexado.

Os conjuntos S (pares formados pela rota e pelo comprimento de onda atribuídos a cada caminho ótico) e Ω (cópias de N) são inicializados na linha 1. Os caminhos óticos são roteados e recebem um comprimento de onda nas linhas 2 a 12. Caso não exista uma cópia de N em Ω onde o caminho ótico $t_{\pi(i)}$ possa ser roteado com menos de d arcos, uma nova cópia de N é criada na linha 5 e adicionada ao conjunto Ω na linha 6. Uma cópia de N na qual o caminho ótico $t_{\pi(i)}$ possa ser roteado com menos de d arcos é encontrada na linha 8. No caso de existir mais de uma cópia de N que satisfaça essa propriedade, `FF-RWA` seleciona a primeira encontrada, enquanto `BF-RWA` escolhe aquela onde o caminho ótico $t_{\pi(i)}$ pode ser roteado com o menor número de arcos. Na linha 9, associa-se ao caminho ótico $t_{\pi(i)}$ a rota $p_{\pi(i)}$ com o menor caminho mais curto entre os nós terminais de $t_{\pi(i)}$ na cópia de N selecionada, assim como o comprimento de onda $w_{\pi(i)}$ correspondente. O par $(p_{\pi(i)}, w_{\pi(i)})$ é adicionado à solução parcial corrente na linha 10 e todos os arcos na rota $p_{\pi(i)}$ são removidos da cópia de N na linha 11. Por fim, o conjunto S é retornado na linha 13.

Os pseudocódigos das heurísticas `FFD-RWA` e `BFD-RWA` são idênticos aos de `FF-RWA` e `BF-RWA`, respectivamente, exceto que elas são executadas com o vetor π correspondente aos caminhos óticos ordenados em ordem decrescente do tamanho dos seus caminhos mais curtos. Os empates entre caminhos óticos com o mesmo tamanho de caminho mais curto são quebrados aleatoriamente. Conseqüentemente, as soluções retornadas por `FFD-RWA` e `BFD-RWA` podem variar a cada execução. Esta ordenação é usada em razão de caminhos óticos longos serem mais difíceis de serem roteados, devendo ser roteados primeiramente.

Skorin-Kapov [88] também sugere a aplicação das heurísticas `FF-RWA`, `BF-RWA`, `FFD-RWA` e `BFD-RWA` para uma versão multi-objetivo de min-RWA que procura minimizar tanto o número total de comprimentos de onda utilizados

PUC-Rio - Certificação Digital Nº 0410880/CA

Heurísticas `FF-RWA` e `BF-RWA`$(N, \mathcal{T}, d, \pi)$
1. Faça $S \leftarrow \emptyset$ e $\Omega \leftarrow \emptyset$;
2. **para** $i = 1, \ldots, |\mathcal{T}|$ **faça**
3. **se** não existem caminhos disjuntos em arcos disponíveis para rotear $t_{\pi(i)}$ com menos de d arcos em qualquer cópia de N em Ω
4. **então faça**
5. Crie uma nova cópia de N;
6. Adicione a Ω a nova cópia de N;
7. **fim-se**
8. Selecione uma cópia de N em Ω onde o caminho ótico $t_{\pi(i)}$ pode ser roteado com menos de d arcos;
9. Seja $p_{\pi(i)}$ o caminho mais curto de $t_{\pi(i)}$ nesta cópia de N e $w_{\pi(i)}$ o comprimento de onda correspondente;
10. $S \leftarrow S \cup (p_{\pi(i)}, w_{\pi(i)})$;
11. Remova todos os arcos em $p_{\pi(i)}$ desta cópia de N;
12. **fim-para**;
13. **retorne** S;

fim.

Figura 2.6: Pseudocódigo das heurísticas `FF-RWA` e `BF-RWA`.

quanto a média do comprimento do caminhos óticos. Como esta tese trata do problema clássico de min-RWA, o comprimento dos caminhos óticos não são considerados na função objetivo. Em outras palavras, considera-se que os arcos de N têm pesos iguais a 1.

PUC-Rio - Certificação Digital Nº 0410880/CA