

Artigo

Uma avaliação comparativa de algoritmos inspirados na natureza para projeto de redes de telecomunicações

Stanisław Kozdrowski ^{1, *}, Mateusz Żotkiewicz ¹, Kacper Wnuk ¹, Arkadiusz Sikorski ¹
e Sławomir Sujecki ^{2,3}

¹ Instituto de Ciência da Computação, Universidade de Tecnologia de Varsóvia, Nowowiejska 15/19, 00-665 Varsóvia, Polónia; mżotkiew@tele.pw.edu.pl (M.Ż.); kacper.wnuk.stud@pw.edu.pl (KW); Arkadiusz.Sikorski.stud@pw.edu.pl (AS)

² George Green Institute, Universidade de Nottingham, Nottingham NG7 2RD, Reino Unido; Sławomir.Sujecki@nottingham.ac.uk

³ Departamento de Telecomunicações e Telemática, Universidade de Ciência e Tecnologia de Wrocław, 50-370 Wrocław, Polónia

* Correspondência: s.kozdrowski@elka.pw.edu.pl

Recebido: 23 de julho de 2020; Aceito: 25 de setembro de 2020; Publicado: 29 de setembro de 2020

Resumo: O assunto do estudo foi uma aplicação de algoritmos metaheurísticos inspirados na natureza para otimização da configuração de nós em redes ópticas. O principal objetivo da otimização foi minimizar o dispêndio de capital, que inclui os custos dos recursos do nó óptico, como transponders e amplificadores utilizados numa nova geração de redes ópticas. Para tanto, é proposto um modelo que leva em consideração os fenômenos físicos na rede óptica. Algoritmos metaheurísticos inspirados na natureza selecionados foram implementados e comparados com um algoritmo determinístico de referência, baseado em programação linear inteira. Para os casos estudados os resultados obtidos mostram que existe uma grande vantagem na utilização de algoritmos metaheurísticos. Em particular, o algoritmo evolutivo, o algoritmo de abelhas e o algoritmo de busca de harmonia apresentaram desempenho superior para os conjuntos de dados considerados correspondentes a grandes redes ópticas; o algoritmo baseado em programação inteira falhou em encontrar uma solução subótima aceitável dentro do tempo computacional máximo assumido. Todos os métodos de otimização foram comparados para instâncias selecionadas de redes de teletransmissão realistas de diferentes dimensões sujeitas a conjuntos de demanda de tráfego extraídos de dados de tráfego reais.

Palavras-chave: metaheurísticas; algoritmo evolutivo; algoritmo de abelhas; algoritmo de busca de harmonia; inteligência de enxame; otimização baseada em enxame; programação linear inteira mista e inteira; otimização combinatória; desenho de redes ópticas; recursos do nó óptico

1. Introdução

Do ponto de vista da operadora de rede, a demanda cada vez maior por serviços de dados de alta velocidade se traduz na necessidade de atualizar continuamente as redes para aumentar a taxa de transmissão de dados por fibra óptica. Um engenheiro que é responsável pela tarefa de atualização contínua de redes de multiplexação por divisão de comprimento de onda densa (DWDM) [1] precisa de ferramentas de software específicas que permitam estimar o desempenho da rede e otimizar a arquitetura com relação aos custos operacionais e de capital. Assim, a tarefa específica considerada nesta contribuição relacionada à otimização de rede DWDM foi formulada como um problema de programação inteira (IP) e resolvida pelos solucionadores de propósito geral disponíveis [2] No entanto, os resultados de otimização de IP mostram que mesmo se a abordagem de IP for aplicável, ela é ineficiente para redes maiores, uma vez que a tarefa de design é, em geral, NP-completa [3] As limitações de eficiência numérica são particularmente agudas no contexto de roteamento e atribuição de comprimento de onda (RWA) [4 - 6] e roteamento e alocação de espectro (RSA) [7 - 10] problemas,

que estão no centro da otimização da rede DWDM. Além disso, as restrições e funções de custo são, em um caso geral, não lineares e, portanto, podem complicar ainda mais a situação.

Uma vez que a invenção de algoritmos exatos computacionalmente eficientes é bastante improvável, os métodos de otimização discreta heurística parecem ser os mais adequados para a otimização de redes DWDM com tamanho realístico e alta modularidade de recursos de nó, levando em consideração deficiências da rede óptica, como atenuação ou um sinal óptico para relação de ruído (OSNR) [11 , 12]. Portanto, o objetivo principal deste artigo é estudar as propriedades de algoritmos heurísticos que são aplicáveis à otimização de redes DWDM, em que particular atenção é dada a algoritmos inspirados na natureza, com operadores específicos de problemas que foram especificamente adaptados pelos autores para o problema estudado. Os desempenhos de todos os algoritmos de otimização foram comparados para redes de teste selecionadas com complexidade diferente. Todas as redes usadas para comparação foram otimizadas, sujeitas a conjuntos de demanda de tráfego extraídos de dados de tráfego reais [13].

Este artigo está organizado da seguinte forma. Na seção 2 o problema é formulado e os modelos de rede são descritos. Devido a limitações de espaço, decidimos reduzir a apresentação do problema ao mínimo e consultar nosso artigo anterior. Na seção 3 metodologia geral é descrita. Na seção 4 as metaheurísticas propostas são descritas. A seguir, na Seção 5 um testbed de quatro estruturas de rede é apresentado e resultados de uma série de experimentos numéricos testando os algoritmos são fornecidos e comparados. Finalmente, Seção 6 fornece um resumo dos resultados da pesquisa.

2. Formulação do problema

Nesta seção, o problema de otimização de rede UW-WDM é formulado usando MIP [10]. Para este propósito, os seguintes conjuntos são definidos:

N conjunto de todos os nós;

T conjunto de transponders;

S conjunto de fatias de frequência;

E conjunto de bordas;

$P(n, n'; t)$ conjunto de caminhos entre nós $n, n' \in N$ para transponder $t \in T$; $p \subseteq E$; B conjunto de bandas;

S_b conjunto de fatias de frequência usadas pela banda $b \in B$; $S_b \subseteq S$; S_t conjunto de fatias de frequência que podem ser usadas como início; $b \in B$

fatias para transponder $t \in T$; $S_t \subseteq S$.

Observe que as restrições de orçamento de energia são levadas em consideração no modelo, limitando conjuntos $P(n, n'; t)$. Nesta abordagem, os conjuntos são pré-computados e contêm apenas caminhos que são viáveis do ponto de vista do orçamento de energia.

Usamos as seguintes variáveis:

y_{est} variável binária, igual a 1 se banda b é usado no limite e e 0 caso contrário;

$x_{tnn'ps}$ variável binária que é igual a 1 se transponders t são instalados entre os nós n e n' ;

roteado no caminho p e começar na fatia de frequência $s \in S_t$ e 0 caso contrário.

No modelo, as seguintes constantes são usadas:

$\xi(b)$ custo de usar banda b em uma única borda;

$\xi(tb)$ custo de usar um par de transponders t na banda b ; $v(t)$ taxa de bits

fornecida pelo transponder t ; $d(n, n')$ taxa de bits exigida do nó n para o nó n' ;

;

$u(t, s, s')$ constante binária que é igual a 1 se o transponder t usando largura de banda começando na fatia de frequência s também usa fatia de frequência s' ; 0 caso contrário.

No modelo de custo, EDFAs, pré-amplificadores, boosters, ILAs e transponders estão incluídos em $\xi(b)$ e $\xi(tb)$ mas não WSSs e filtros, uma vez que os últimos dispositivos não podem estar sujeitos à otimização.

O modelo de otimização (é o seguinte:

$$\min \sum_{b \in B} \xi(b) \sum_{e \in E_{\text{estar}}} y_e + \sum_{t \in T} \xi(t) \sum_{n, n' \in N} \sum_{p \in P(n, n', t)} \sum_{s \in S_t} x_{tnn'ps} \quad (1)$$

$$\sum_{t \in T} \sum_{p \in P(n, n', t)} \sum_{s \in S_t} v(t) x_{tnn'ps} \quad n_{ps} \geq d(n, n') \quad \forall n, n' \in N \quad (2)$$

$$\sum_{t \in T} \sum_{n, n' \in N} \sum_{p \in P(n, n', t)} \sum_{e \in ps} u(t, s, s') \cdot x_{tnn'ps} \leq y_{\text{estar}} \quad \forall e \in E, \quad \forall b \in B, \quad \forall s' \in S_b \quad (3)$$

O assunto de minimização é o custo de amplificadores e transponders instalados em (1)

Restrições (2) garantir que todas as demandas sejam satisfeitas. Restrições (3) certifique-se de que cada fatia de frequência em cada borda não seja usada mais de uma vez. Além disso, eles garantem que o uso de uma banda em uma borda resulte na instalação de amplificadores apropriados.

Nesta contribuição é considerado o problema de minimização de CAPEX e diversas técnicas são aplicadas a fim de reduzir o tempo de cálculo. A motivação para reduzir o tempo de cálculo do problema de minimização de CAPEX é dupla. Em primeiro lugar, a minimização do CAPEX deve ser realizada cada vez que uma nova capacidade é necessária após uma solicitação vinda de um cliente. Conseqüentemente, menor tempo de cálculo é claramente benéfico. Em segundo lugar, foi observado que o tempo de cálculo dos métodos de otimização padrão (por exemplo, ILP) cresce rapidamente com o número de nós da rede [7] Assim, para grandes redes, a aplicação de métodos heurísticos torna-se indispensável.

3. Metodologia

3.1. Modelo de Solução

Para cada algoritmo, o mesmo modelo de solução foi aplicado. A estrutura genérica do modelo de solução é mostrada na Figura 1. Assim, para implementar um algoritmo, são realizadas as seguintes etapas:

- Para todos os pares de nós no conjunto N , um conjunto de transponders T devem ser selecionados aqueles que são capazes de atender as demandas para o par de nós. O conjunto é um vetor de 4 elementos $T = \{n \cdot t_1, n \cdot t_2, n \cdot t_3, n \cdot t_4\}$, onde o primeiro elemento é o número de transponders 40G usados, o segundo elemento é o número de 100 transponders G usados, o terceiro elemento é o número de 200 transponders G usados e o quarto elemento é o número de 400 transponders G usados. Por exemplo, $T = \{2, 1, 0, 3\}$ significa que este conjunto consiste em dois transponders de 40 G, um transponder de 100 G e três transponders de 400 Gb. O conjunto de transponders é escolhido entre quatro soluções pré-calculadas, geradas por programação linear.
- Para cada transponder selecionado, um caminho deve ser determinado, através do qual ele transmitirá os dados. O caminho $p \in P$ é selecionado aleatoriamente desenhando um dos k - conjunto de elementos de caminhos pré-definidos. É necessário escolher um caminho para poder atribuir
- cada transponder $t \in T$ um pedaço único de largura de banda $s \in S$ para transmissão de dados. Isso é feito usando um algoritmo RMLSA heurístico (nível de modulação de roteamento e alocação de espectro) [14] que lida sequencialmente com cada demanda e usa a classificação do caminho mais longo primeiro (LPF).

3.2. Operadores

Esta seção descreve resumidamente os operadores usados para modificar uma solução, ou seja, *operador de realização de mudança*, *operador de mudança de caminho*, *operador de cruzamento*.

Operador *mudar o operador de realização* calcula uma nova solução para uma determinada demanda. Para cada demanda, calculamos previamente quatro conjuntos de transponders capazes de satisfazê-los. Depois de escolher um dos conjuntos pré-calculados, para cada transponder $t \in T$ do conjunto, um dos caminhos predefinidos é escolhido aleatoriamente.

Operador *operador de mudança de caminho* altera um caminho atribuído a um transponder. Um novo caminho é escolhido aleatoriamente a partir de um conjunto de caminhos pré-definidos. Por isso *operador de mudança de caminho* altera o identificador de caminho selecionado alocado para o transponder específico para um valor permitido aleatório. Portanto, se k caminhos estão disponíveis, então *operador de mudança de caminho* muda o identificador de caminho para um valor selecionado aleatoriamente de um conjunto de

$\{1, 2, \dots, k\}$. Neste estudo para cada demanda, $k = 3$ foi assumido.

Para implementação de *operador de crossover*, um crossover uniforme [15] foi usado. Nesta aplicação, cromossomos descendentes são criados como uma concatenação de vetores de demanda escolhidos aleatoriamente

dos pais. Nesta variante de crossover, { a C o genótipo é formado a partir dos genes dos pais em

com base no seguinte esquema:

$$C_j = \begin{cases} \text{UMA}_{eu} \text{ se } \zeta_{eu} < p_e \\ \text{B}_{eu} & \text{de outra forma,} \end{cases}$$

Onde ζ_{eu} significa a execução de uma variável aleatória com um $VOCÊ(0, 1)$ distribuição uniforme, desenhada para cada eu item no

genótipo separadamente, e p_e é a probabilidade de substituição do gene. Um típico p_e

o valor é: $p_e = 1$

Um exemplo de crossover, baseado na rede polonesa é ilustrado na Figura 2. Nisso

Por exemplo específico, o vetor de demanda dois está sujeito a cruzamento, ou seja, a demanda dois para a prole foi trocada quando comparada aos pais.

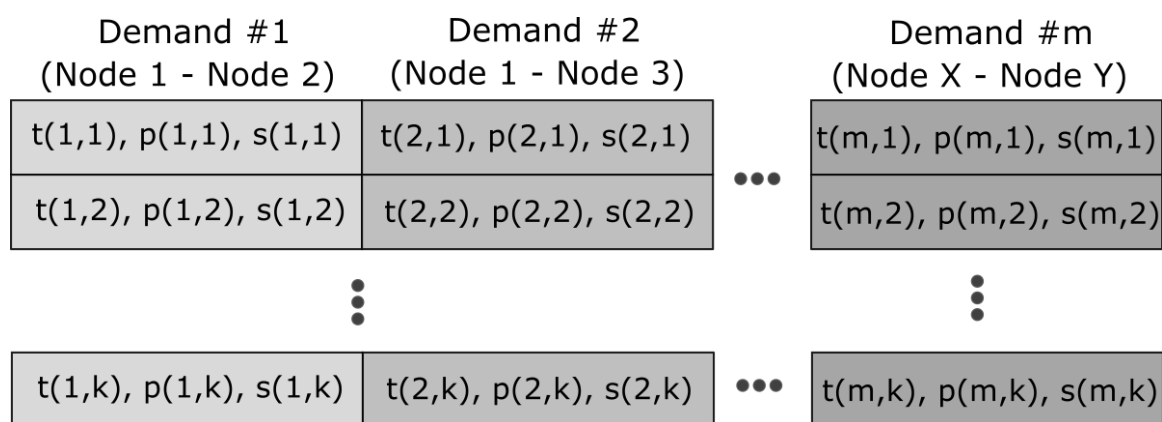


Figura 1. Estrutura de um modelo. Cada demanda consiste em k Vetores de 3 elementos; k pode ser diferente para diferentes demandas.

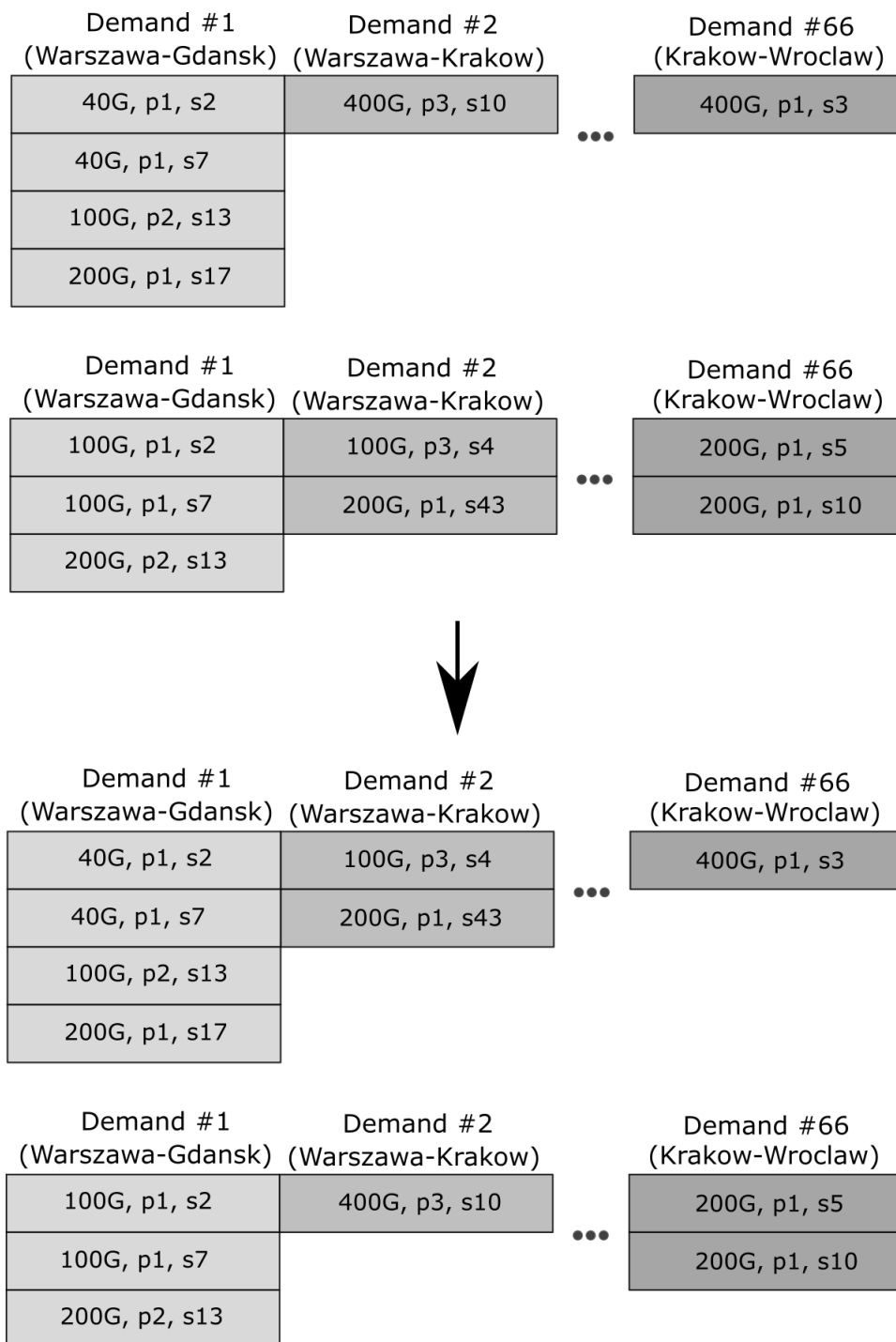


Figura 2. Um exemplo de crossover, baseado na rede polonesa.

4. Métodos

No artigo, vários métodos foram usados para resolver o problema apresentado na Seção 2, que pode ser dividido em duas classes. A primeira classe é formada por algoritmos de população e consiste no algoritmo das abelhas (BA) [16–18] e o algoritmo evolutivo (EA) [19–21]. A segunda classe consiste em algoritmos que geram uma solução em cada iteração. Aqui, o algoritmo de busca por harmonia (HS) [22] e o algoritmo stochastic hill climbing (HC) [23] foi considerado. Como referência, um método bem conhecido baseado em programação linear inteira (IP) foi usado [24, 25]. Este método está disponível por meio de um solver comercial CPLEX [2].

4.1. O Algoritmo das Abelhas

O algoritmo das abelhas é inspirado nas abelhas melíferas e sua maneira de procurar manchas de flores ricas em néctar [16]. Cada solução e seu valor da função objetivo podem ser percebidos como um canteiro de flores encontrado por uma abelha escuteira e uma quantidade de néctar disponível ali. A vizinhança de um canteiro de flores rico em néctar deve ser explorada por mais abelhas do que os outros canteiros de flores. Inicialmente, o espaço de busca é amostrado aleatoriamente de acordo com as abelhas exploradoras procurando aleatoriamente nos arredores da colmeia. Por isso N soluções aleatórias são geradas. Então, um grupo de m melhores soluções são escolhidas a partir do

população atual. Essas soluções são usadas para pesquisa local. Para o *elite* e fora de m soluções n_1 vizinhos aleatórios são gerados. Analogamente, para o restante *melhor* $m - e$ soluções n_2 vizinhos aleatórios são gerados.

Observe que n_1 deve ser maior que n_2 e o tamanho da vizinhança da pesquisa local é determinado pelo valor k . Para gerar um vizinho, *mudar o operador de realização* é usado k vezes. No fim de cada iteração m melhores soluções são escolhidas de *melhor*, *elite* e vizinhos recém-gerados. Para manter o tamanho da população constante, as populações atuais são preenchidas com $N - m$ soluções completamente aleatórias. O pseudo-código do algoritmo é apresentado em Algoritmo 1. Para o problema descrito neste artigo os hiperparâmetros N , m , e , n_1 , n_2 , k deve ser relativamente pequeno.

Algoritmo 1: O algoritmo das abelhas.

Entrada: N, m, e, n_1, n_2, k
Resultado: a melhor solução encontrada
 $abelhas \leftarrow \text{RandomSolutions} (N);$
enquanto *condição de parada não é atendida* **Faz**
 $melhor \leftarrow \text{SelectBest} (abelhas, m);$
 $elite \leftarrow \text{SelectBest} (melhor, e);$
 $melhor \leftarrow melhor - elite;$
 $elite_neighs \leftarrow \text{RandomNeighbors} (elite, n_1, k);$
 $best_neighs \leftarrow \text{RandomNeighbours} (melhor, n_2, k);$
 $abelhas \leftarrow elite_neighs + best_neighbors + best + elite;$
 $abelhas \leftarrow \text{SelectBest} (abelhas, m) + \text{RandomSolutions} (N - m);$
fim
Retorna $\text{SelectBest} (abelhas, 1);$

4.2. Harmony Search

No início do algoritmo, a harmonia da memória é inicializada [26]. É um conjunto de modelos de solução gerados e avaliados aleatoriamente. O tamanho da memória é constante. Em cada iteração, uma nova solução é criada usando as seguintes etapas:

- Para cada par de nós no conjunto N escolha a solução que satisfaça a demanda.
- Com probabilidade p_{uma} , uma solução é escolhida aleatoriamente na memória. Também há probabilidade p_m que a solução escolhida será modificada usando *operador de mudança de caminho*.
- Caso contrário, uma solução é criada usando *alterar o operador de realização*.

Após a criação, a solução é avaliada e se for melhor do que qualquer uma das soluções armazenadas na memória, a pior solução é substituída pela solução recém-criada. Devido a duas formas de gerar soluções, o algoritmo tem a opção de explorar e explorar o espaço da solução. Além disso, é certo que a melhor solução gerada será a solução final devido à forma de armazenar as soluções na memória. O pseudo-código do algoritmo é apresentado em Algoritmo 2.

Algoritmo 2: Pesquisa de harmonia.

Entrada: Número de iterações (MaxIters), tamanho da memória (MemSize)

Resultado: A melhor solução encontrada (BestSoFar)

$M \leftarrow$ população inicial aleatória [MemSize]; Avalie a memória (M);

$t = 0$;

enquanto $t < \text{MaxIters}$ **Faz**

$\text{NewSolution} \leftarrow \text{EmptySolution}()$;

E se $\text{Rand}(0, 1) < p_{\text{uma}}$ **então**

pra NodePair em N **Faz**

$\text{realização} \leftarrow \text{ChooseRealization}(\text{NodePair}, M)$;

E se $\text{Rand}(0, 1) < p_m$ **então**

$\text{ChangePath}(\text{realização})$;

fim

$\text{AddRealization}(\text{NewSolution}, \text{NodePair}, \text{realização})$;

fim

outro

$\text{CreateRandom}(\text{NewSolution})$;

fim

$\text{Avaliar}(\text{NewSolution})$;

$\text{CompareWithMemory}(\text{NewSolution}, M)$;

$t \leftarrow t + 1$

fim

Retorna $\text{TheBestFrom}(M)$;

4.3. Algoritmo Evolucionário

O EA, que foi empregado para o problema de otimização do nó óptico, foi inspirado na estratégia evolutiva $((\mu + \lambda)$ ES) dado em [27, 28], mas tanto o método de codificação quanto os operadores genéticos que foram aplicados aqui são de problemas específicos. Seu pseudo-código é dado em Algoritmo 3.

O algoritmo mantém uma população P_t de indivíduos. A população é inicializada aleatoriamente. Para cada gene, a demanda associada a ele é alocada aleatoriamente para um caminho possível. Então o a função objetivo de cada indivíduo é calculada de acordo com (1). O loop principal do algoritmo trabalha para o número especificado de iterações, que é um parâmetro do método. Dentro do loop principal, uma prole população O_t é gerado reproduzindo elementos selecionados aleatoriamente de P_t . Então, os cromossomos são acasalados aleatoriamente em pares, cruzados e mutados. Uma nova população P_{t+1} é gerado selecionando os melhores cromossomos da soma da população base e da população descendente.

Para poder usar o EA, a solução do problema deve ser codificada para formar um genótipo de um indivíduo. A estrutura do cromossomo já foi representada na Figura 1 e um exemplo do cromossomo baseado na rede polonesa está representado na Figura 3. A demanda 1 é realizada por meio de quatro transponders (primeiro, atravessando o caminho 1 e usando a fatia 2; segundo, 40 G percorrendo o mesmo caminho, usando a fatia 7; terceiro, 100 G transponder - caminho 2 e fatia 13; e quarto, 200 G transponder – path 1 e slice 17). A demanda 2 tem dois transponders e a última demanda 66 tem apenas um transponder, atravessando o caminho 1, usando a fatia 5. O crossover é realizado usando *operador de crossover* conforme descrito anteriormente. A mutação

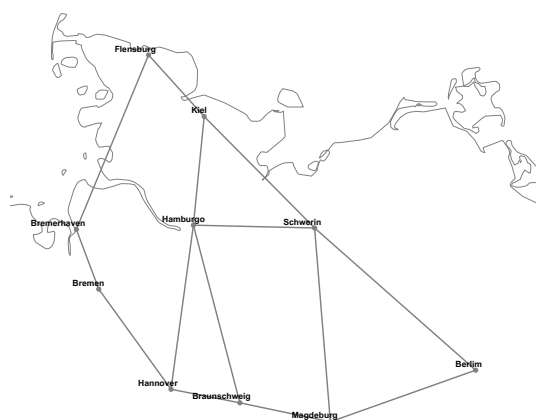
é realizado para cada cromossomo. Cada demanda é alterada separadamente com probabilidade p_m usando *operador de mudança de caminho*.

Algoritmo 3: O algoritmo evolutivo ($\mu + \lambda$)-EA.**Entrada:** Número de iterações (MaxIters), tamanho da população (PopSize)**Resultado:** A melhor solução encontrada (BestSoFar) $P_0 \leftarrow$ população inicial aleatória [μ chrom.];Objetivo \leftarrow EvaluatePopulation (P_0); $t = 0$;**enquanto** $t < \text{MaxIters}$ **Faz** $O \leftarrow$ RandomlyReproduce (P_t) [λ chrom.]; $C \leftarrow$ Crossover (O); $M \leftarrow$ Mutate (C); ObjetivoNovo \leftarrow EvaluatePopulation (M); $P_{t+1} \leftarrow$ TheBestFrom ($P_t \cup M$, Objetivo, ObjetivoNovo) [μ chrom.]; Objetivo \leftarrow TheBestFrom
 (Objective, ObjectiveNew); $t \leftarrow t + 1$ **fim****Retorna** TheBestFrom (P_t , Objetivo);

Demand #1 (Warszawa-Gdansk)	Demand #2 (Warszawa-Krakow)		Demand #66 (Krakow-Wroclaw)
40G, p1, s2	100G, p3, s4	...	400G, p1, s5
40G, p1, s7	200G, p1, s43		
100G, p2, s13			
200G, p1, s17			

Figura 3. Um exemplo de um cromossomo, baseado na rede polonesa.**5. Resultados e discussão**

Resultados computacionais foram obtidos para sete redes ópticas com características diferentes. As redes correspondem a redes ópticas reais provenientes de países específicos e foram retiradas de [29] Mesa 1 e figuras 4 - 10 fornecer os parâmetros relevantes para ambas as redes e suas topologias.

**Figura 4.** Diagrama esquemático da rede de backbone ótica de transmissão nacional alemã de 10 nós.

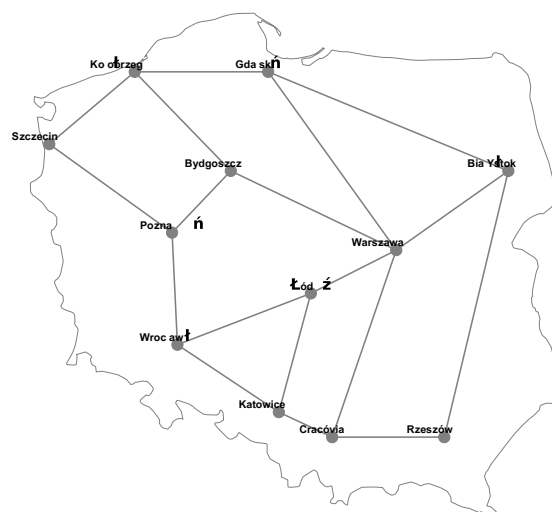


Figura 5. Diagrama esquemático da rede de backbone ótica de transmissão nacional polonesa.

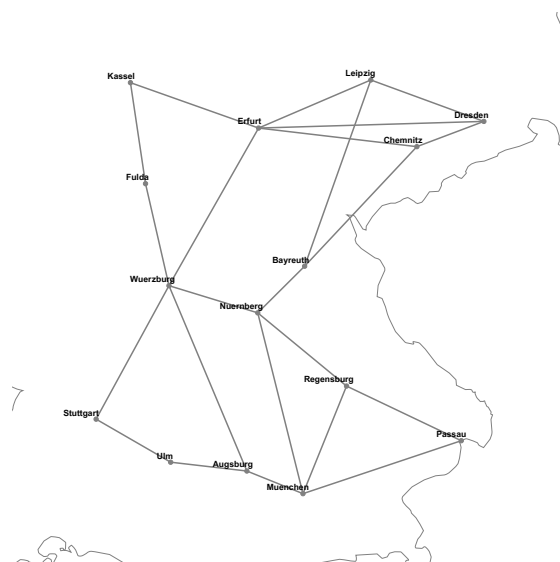


Figura 6. Diagrama esquemático da rede de backbone ótica de transmissão nacional alemã de 15 nós.

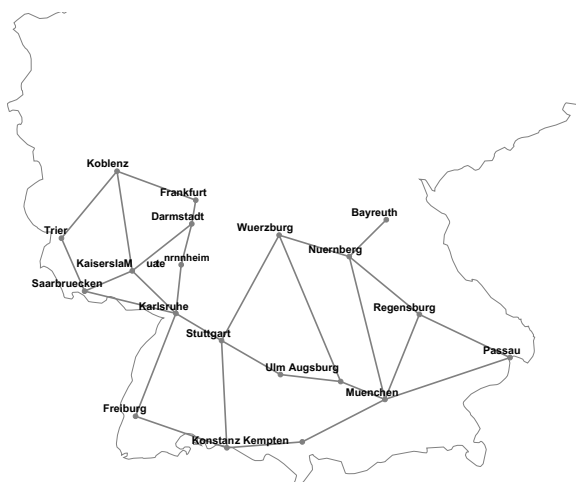


Figura 7. Diagrama esquemático da rede de backbone ótica de transmissão nacional alemã de 20 nós.

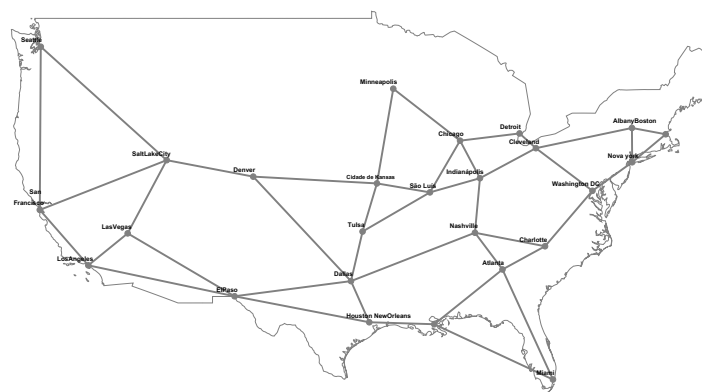


Figura 8. Diagrama esquemático de uma rede de backbone ótico de transmissão nacional dos EUA.

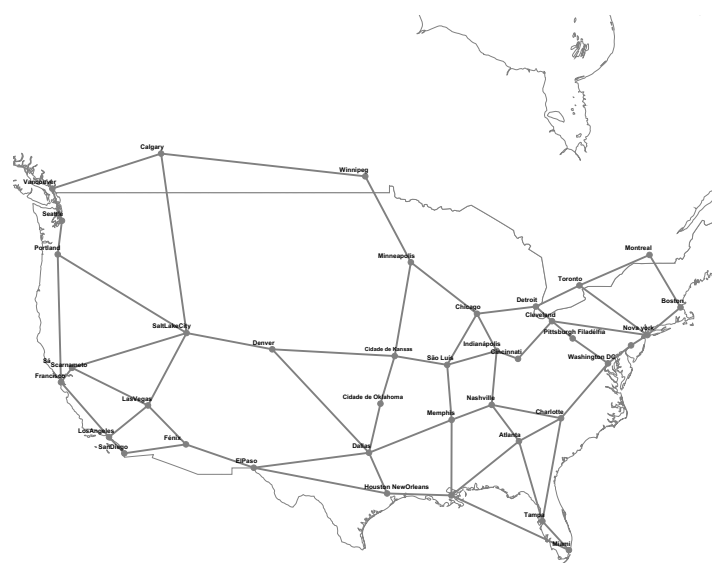


Figura 9. Diagrama esquemático de uma rede de backbone ótico de transmissão nacional americana.

Tabela 1. Parâmetros das redes analisadas.

Rede	# Nós	# Links	# Demandas
Alemanha10	10	14	45
polonês	12	18	66
Alemanha 15	15	23	105
Alemanha20	20	29	190
EUA	26	42	325
americano	39	61	741
Alemanha	50	88	12: 25h

As demandas que foram utilizadas na otimização dos custos da rede são dadas por uma matriz de demanda, que fornece os valores do fluxo de tráfego entre os nós selecionados expressos em gigabits por segundo (Gbps). Os cálculos foram realizados usando um mecanismo de resolução linear do CPLEX 12.8.0.0 em um Processador 2.1 GHz Xeon E7-4830 v.3 com 256 GB de RAM rodando no sistema operacional Linux Debian. Mesa 2 descreve em detalhes os conjuntos e suas configurações e apresenta as configurações constantes usadas durante o processo computacional.

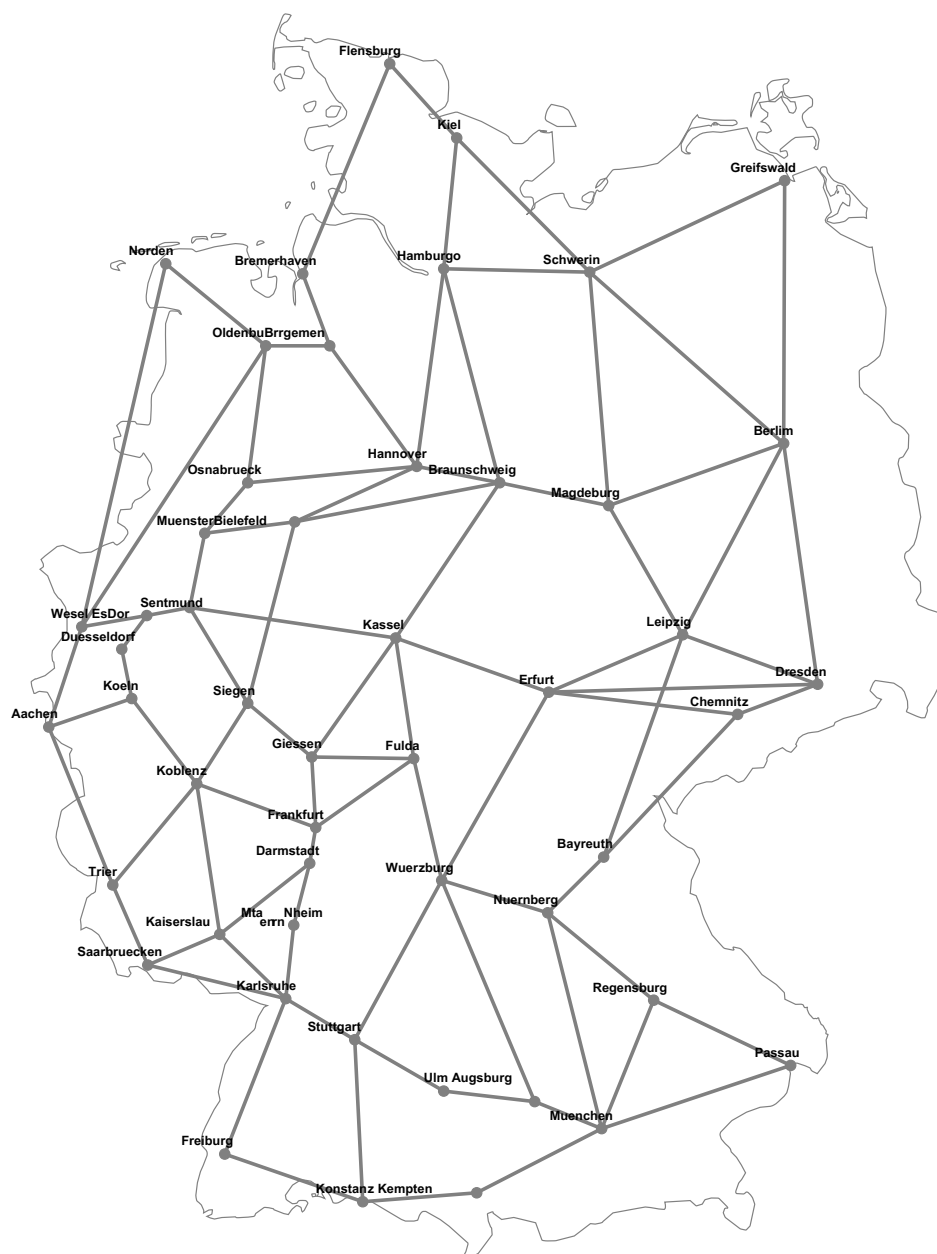


Figura 10. Diagrama esquemático da rede de backbone ótica de transmissão nacional alemã de 50 nós.

Os seguintes parâmetros de modelagem foram usados para algoritmos heurísticos: $p_a = 0,95$, $p_m = 0,05$ e $memory_size = 30$ para HS; $N = 50$, $m = 12$, $e = 6$, $n_1 = 8$, $n_2 = 2$ e $k = 5$ para BA; e $\mu = 100$ $\lambda / \mu = 2$, $p_c = 0,9$ e $p_m = 0,05$ para EA. Todos esses valores foram selecionados empiricamente após uma série de testes.

No primeiro experimento numérico, uma comparação de métodos de otimização foi realizada. Para redes polonesas, alemãs10, alemãs15 e alemãs20 (Figura 4 - 7 e figura 11), que são discutidos primeiro, todos os algoritmos considerados alcançaram a solução ótima. Os resultados dos algoritmos heurísticos são caracterizados por um pequeno desvio padrão. Essa observação apóia a afirmação de que as simulações são bastante repetitivas e que convergem consistentemente para efetivamente o mesmo valor. Além disso, os resultados das tabelas 3 - 6 mostram que os valores médios estão próximos dos valores mínimos. Esse é particularmente o caso do algoritmo BA. Assim, o algoritmo BA apresentou em média o melhor desempenho entre todos os algoritmos heurísticos em termos de consistência.

Mesa 2. Configurações definidas e constantes.

Definir	Definir configurações
N	na tabela 1
E	na tabela 1
S	768 slots
B	2 bandas
T	4 transponders ($t_1 = 40$ G, $t_2 = 100$ G, $t_3 = 200$ G e $t_4 = 400$ G)
S_i	$S_1 = \{1 \dots 380\} \cup \{385 \dots 764\}$ $S_2 = \{1 \dots 380\} \cup \{385 \dots 762\}$ $S_3 = \{1 \dots 378\} \cup \{385 \dots 762\}$ $S_4 = \{1 \dots 376\} \cup \{385 \dots 760\}$
Constante	Configurações constantes
$d(n, n') [Gbps]$	por exemplo, $d(1, 2) = 70$
$\xi(b)$	$\xi(1) = 1, \xi(2) = 2$
$\xi(tb)$	$\xi(1, 1) = 2, \xi(2, 1) = 5, \xi(3, 1) = 7, \xi(4, 1) = 9$ $\xi(1, 2) = 2,4, \xi(2, 2) = 6, \xi(3, 2) = 8,4, \xi(4, 2) = 11,8$

Já os resultados obtidos pelo algoritmo HS são caracterizados pelo maior desvio padrão e pelo maior valor médio da função de custo. Assim, o algoritmo HS teve o pior desempenho entre todos os algoritmos heurísticos. Também é importante notar que o software CPLEX levou cerca de 6 min para encontrar a solução ótima de 1321. Portanto, em média, a abordagem IP venceu a competição para a instância de rede polonesa. Por fim, nota-se que, em média, o algoritmo heurístico mais rápido acabou sendo o algoritmo HC. Um pouco pior, em termos de tempo, foi o algoritmo HS. Ambos os algoritmos estavam próximos da abordagem IP em termos de tempo. Os algoritmos de enxame (BA e EA) finalmente alcançaram o ótimo, mas precisaram de cerca de 20 vezes mais tempo.

Os algoritmos implementados também foram aplicados às grandes redes, ou seja, as redes dos EUA, American e German50 (Figuras 8 - 10). Esse foi o objetivo principal deste artigo. Analogamente, os resultados foram coletados em Tabelas 7 - 9. Além disso, as curvas de convergência com mais de 50 execuções independentes são representadas nas Figuras 12 - 14 para as redes dos EUA, American e German50. Em todos os três casos, a abordagem IP encontrou soluções muito mais distantes do valor ideal do que as encontradas por algoritmos heurísticos. Em todos os casos estudados, os maiores valores mínimos foram obtidos pelos algoritmos HC e HS.

É importante notar que a abordagem ILP tem uma vantagem sobre os métodos heurísticos, ou seja, ILP, por fornecer o limite inferior para o mínimo procurado. Os resultados obtidos mostram que para os casos estudados o ILP é particularmente eficaz quando aplicado às redes Alemã10, Polonesa, Alemã15 e Alemã20, já que neste caso um algoritmo ILP encontra o ótimo. Para as redes dos EUA, Estados Unidos e Alemanha, conforme mencionado, o ILP encontra a banda inferior e, portanto, permite estimar o gap de otimalidade. No entanto, é notado que para redes com muitos nós excessivamente o gap de otimalidade pode ser tão grande a ponto de impedir qualquer conhecimento prático sobre a qualidade das soluções viáveis obtidas.

Considerando os resultados médios, o algoritmo HC teve o pior desempenho em todos os casos. Por outro lado, os melhores resultados em todos os casos considerados para as redes dos EUA, Estados Unidos e Alemanha foram obtidos de forma consistente usando algoritmos BA e EA. Os algoritmos BA e EA alcançaram o mínimo e, o mais importante, o valor médio mais baixo para a rede americana. Esta observação apóia a afirmação de que os algoritmos de enxame (BA e EA) são os mais adequados para otimização de redes ópticas com um grande número de nós (ou seja, EUA, americano e alemão). Os resultados dos algoritmos BA e HC apresentam os maiores valores de desvio padrão. Isso sugere que os algoritmos BA e HC exploram o espaço da solução de forma mais robusta e, portanto, fornecem resultados mais diversos.

Nos casos mais complexos, redes americanas, americanas e alemãs de 50 nós (Figuras 8 - 10), algoritmos baseados principalmente na exploração do espaço de solução (HC e HS) convergiram para um determinado ponto, e então, devido à pequena exploração, pararam em um dos extremos dos espaços de solução local. Neste ponto,

Os algoritmos EA e BA mostram sua superioridade. Isso ocorre porque nos algoritmos EA e BA é dada mais ênfase à exploração do que nos dois algoritmos anteriores (HC e HS), o que permite que os algoritmos EA e BA evitem a armadilha de extremos locais e permite que continuem com otimização adicional.

As figuras 11 - 14 também mostram a influência da população de inicialização pelos algoritmos HS, EA e BA em contraste com o algoritmo HC. Mais de uma solução no início fornece uma solução de base potencialmente melhor, de modo que esses algoritmos encontram uma solução de custo mais baixo no início (o gráfico começa com um valor de custo muito mais baixo).

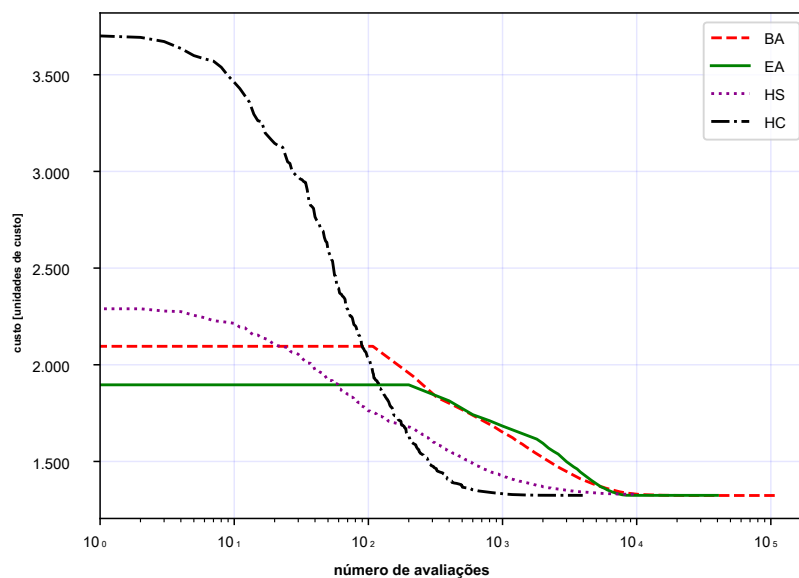


Figura 11. Comparação da convergência para a rede polonesa de todos os métodos considerados: colônia de abelhas artificial (BA), algoritmo evolutivo (EA), algoritmo de busca por harmonia (HS) e algoritmo de escalada estocástica (HC). Cada curva é uma média de 50 execuções independentes.

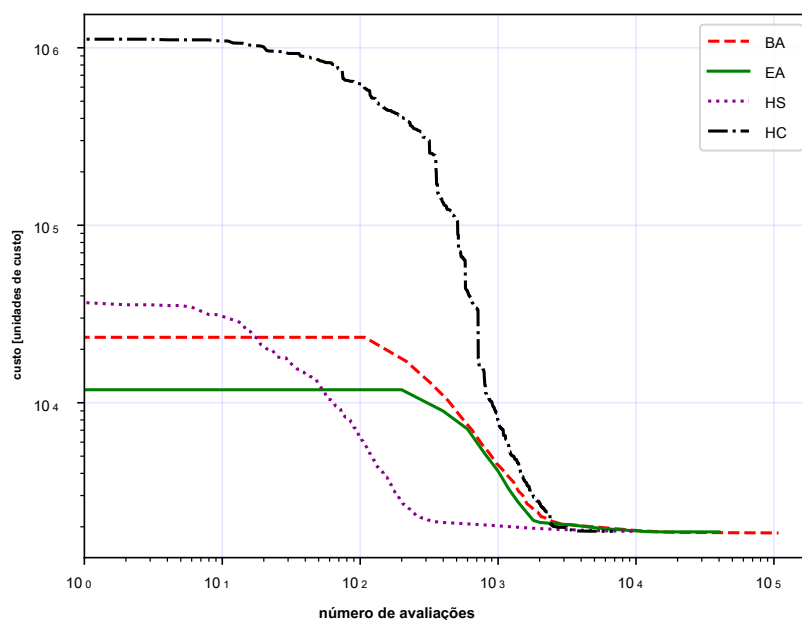


Figura 12. Comparação da convergência para a rede dos EUA de todos os métodos considerados: colônia de abelhas artificial (BA), algoritmo evolutivo (EA), algoritmo de busca por harmonia (HS) e algoritmo de escalada estocástica (HC). Cada curva é uma média de 50 execuções independentes.

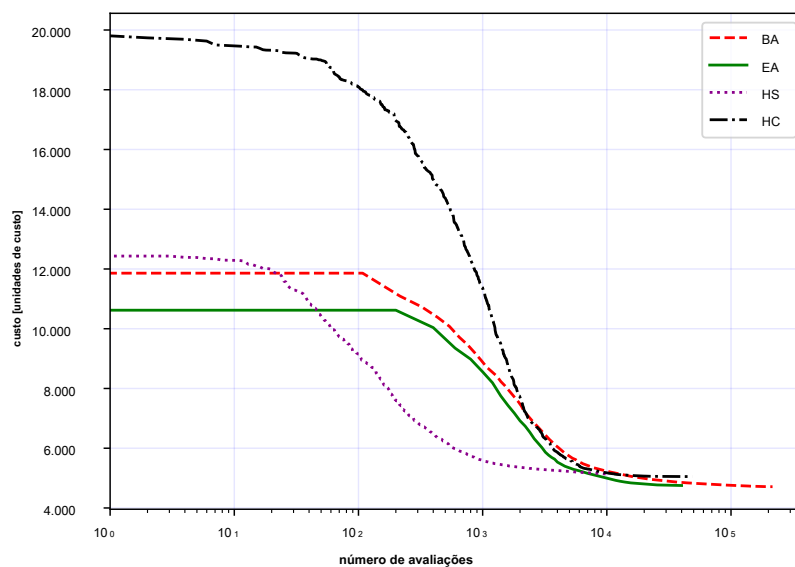


Figura 13. Comparação da convergência para a rede americana de todos os métodos considerados: colônia de abelhas artificial (BA), algoritmo evolutivo (EA), algoritmo de busca por harmonia (HS) e algoritmo de escalada estocástica (HC). Cada curva é uma média de 50 execuções independentes.

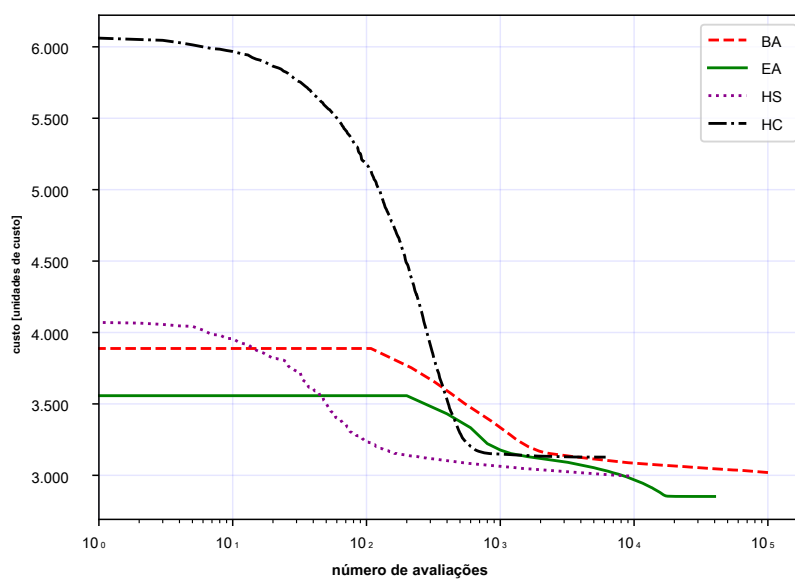


Figura 14. Comparação da convergência para a rede alemã de 50 nós de todos os métodos considerados: colônia de abelhas artificial (BA), algoritmo evolutivo (EA), algoritmo de busca por harmonia (HS) e algoritmo de escalada estocástica (HC). Cada curva é uma média de 50 execuções independentes.

Tabela 3. Comparação de todos os algoritmos considerados para a rede alemã de 10 nós.

Método	Alemão 10						
	Solução			# Avaliação	Tempo [seg.]		
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Lacuna [%]
BA	152	152,63	0,48	69.992	6990,8	170,7	-
EA	152	152,59	0,49	21.220	2170,0	51,0	-
HS	152	152,96	0,19	5293	529,18	25,86	-
HC	153	153,96	0,19	5226	522,18	25,86	-
IP	152	-	-	-	461,12	-	0,00

Além disso, nota-se que demorou mais de 6 dias para a abordagem IP encontrar uma solução, que está ainda mais longe do valor ótimo do que as soluções obtidas por algoritmos heurísticos. Na média,

levou quase três vezes menos tempo para o algoritmo BA do que para a abordagem IP, e quase seis vezes menos tempo foi necessário para o algoritmo EA. Isso se deve ao fato de que as iterações do algoritmo de escalada são computacionalmente muito eficientes quando comparadas com outros algoritmos, por exemplo, o algoritmo EA. Finalmente, a mesma quantidade de tempo foi necessária para completar os cálculos usando algoritmos HS e HC. Na verdade, os algoritmos HS e HC provaram ser os mais rápidos entre todos os algoritmos testados. No entanto, esses algoritmos não encontraram as melhores soluções, que eram fornecidas pelos algoritmos BA e EA.

Tabela 4. Comparação de todos os algoritmos considerados para a rede polonesa.

Método	polonês						
	Solução			# Avaliação	Tempo [seg.]		
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Lacuna [%]
BA	1321	1321,0	0,0	108.000	7650,8	165,2	-
EA	1321	1321,6	1,3	40.000	8562,1	275,3	-
HS	1321	1330,1	5,9	7641	420,3	90,1	-
HC	1321	1321,7	1,4	2421	410,8	80,8	-
IP	1321	-	-	-	367,0	-	0,00

Tabela 5. Comparação de todos os algoritmos considerados para a rede alemã de 15 nós.

Método	Alemão 15						
	Solução			# Avaliação	Tempo [seg.]		
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Lacuna [%]
BA	281	281,85	0,35	70.084	7008,0	170,7	-
EA	281	281,4	0,49	27.220	2710,0	62,0	-
HS	281	281,93	0,47	5862	582,37	105,5	-
HC	282	282,9	0,32	5440	544,5	24,8	-
IP	281	-	-	-	1547,10	-	0,00

Tabela 6. Comparação de todos os algoritmos considerados para a rede alemã de 20 nós.

Método	Alemão 20						
	Solução			# Avaliação	Tempo [seg.]		
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Lacuna [%]
BA	500	502,1	0,21	55.612	5561,0	417,7	-
EA	500	506,0	0,70	29.770	2870,0	20,7	-
HS	501	503,7	0,4	5245	524,5	47,1	-
HC	501	504,0	0,3	6255	625,3	562,0	-
IP	500	-	-	-	17304,75	-	0,00

Tabela 7. Comparação de todos os algoritmos considerados para a rede dos EUA.

Método	EUA						
	Solução			# Avaliação	Tempo [seg.]		
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Lacuna [%]
BA	1846	1851,6	8,6	108000	18113,2	187,4	-
EA	1846	1872,6	10,2	40.000	19.006,0	514,4	-
HS	1866	1884,8	10,3	9057	1696,3	230,5	-
HC	1862	1886,4	11,9	4868	1661,5	279,7	-
HY	1846	1864,4	8,8	21248	9717,8	455,0	-
IP	2366	-	-	-	3 · 10 ⁵	-	25,7

Tabela 8. Comparação de todos os algoritmos considerados para a rede americana.

Método	americano						
	Solução			# Avaliação		Tempo [seg.]	
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Lacuna [%]
BA	4658	4698,3	40,0	210.819	57.299,3	452,4	-
EA	4685	4751,9	26,2	40.000	62.470,9	706,9	-
HS	5047	5136,1	46,8	9205	4931,5	734,3	-
HC	4900	5051,4	85,5	23.407	30.819,3	6884,5	-
IP	6020	-	-	-	$6 \cdot 10^5$	-	33,2

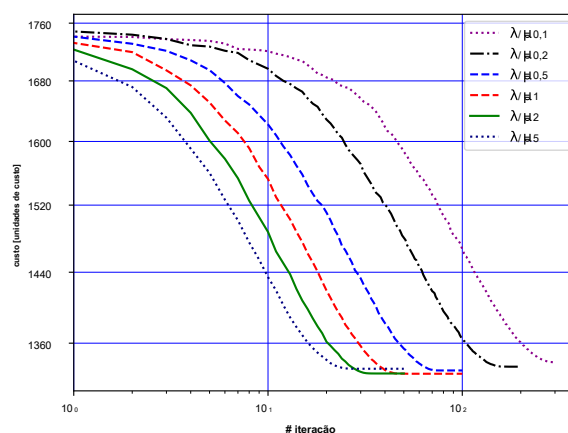
Tabela 9. Comparação de todos os algoritmos considerados para a rede alemã de 50 nós.

Método	Alemão 50						
	Solução			# Avaliação		Tempo [seg.]	
	Melhor	Significar	std. dev.	Significar	Significar	std. dev.	Gap = Vão [%]
BA	2968	3043,8	37,4	108.000	42.981,3	484,8	-
EA	2851	2852,0	1,0	40.000	52.929,6	750,6	-
HS	2979	2993,9	7,7	8805	5783,3	915,8	-
HC	3088	3127,4	22,3	2894	2525,3	1048,0	-
HY	2917	2959,7	13,2	21321	28166,6	3980,9	-
IP	3196	-	-	-	$6 \cdot 10^5$	-	13,1

No último experimento, o ajuste fino do EA foi realizado. Resultados relativos ao ajuste de μ e λ parâmetros são fornecidos, pois isso surgiu como um problema desafiador. O objetivo desta experiência era estabelecer as melhores configurações para μ e λ . A metodologia foi a seguinte: valores de $\mu = 100$ e $\mu = 200$ foram definidos experimentalmente; então $\lambda \in \{0,1, 0,2, 0,5, 1, 2, 5\}$ foram testados, $p_c = 0,9$ e $p_m = 0,05$.

Curvas de convergência apresentadas na Figura 15 - 18 têm formas semelhantes para diferentes λ valores. Se λ é pequeno, quase nenhum ganho é observado nas primeiras iterações. Após as iterações iniciais, no entanto, segue-se uma convergência rápida, após a qual os valores de custo calculados estabelecem um valor constante. Com grande λ , a convergência lenta inicial não ocorre de todo e o período de convergência rápida é seguido novamente pelo estabelecimento de um valor constante.

Assim, a análise do impacto dos parâmetros estudados na convergência conduziu à formulação da seguinte conclusão: se $\mu = 100$, λ deve ser mantido em cerca de 1-2 para produzir o melhor EA desempenho do algoritmo.

Figura 15. Curvas de convergência (custo vs. iterações) para diferentes combinações de μ e λ definições.

Cada curva é uma média de 50 execuções independentes. Valores de λ coeficientes são descritos na legenda da figura. Os resultados foram calculados para a rede polonesa.

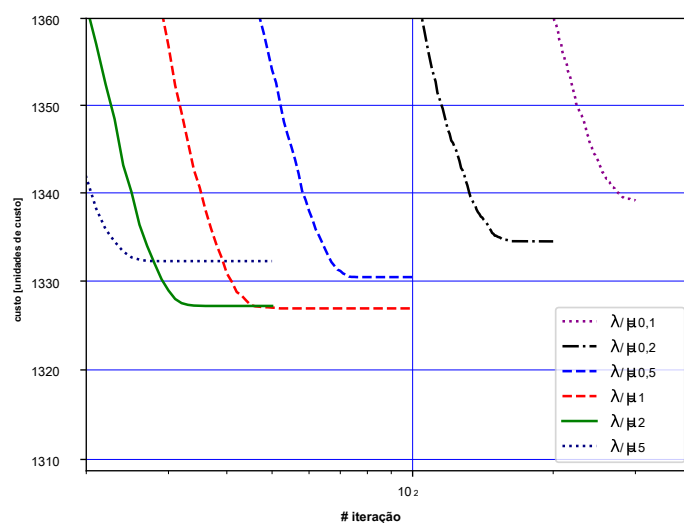


Figura 16. Curvas de convergência (custo vs. iterações) para diferentes combinações de μ e λ

Zoom da Figura 15 .

μ definições.

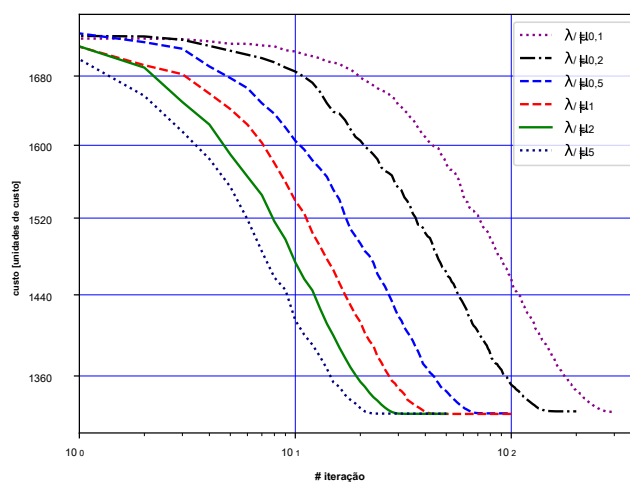


Figura 17. Curvas de convergência (custo vs. iterações) para diferentes combinações de μ e λ

Cada curva é uma média de 50 execuções independentes. Valores de λ

μ coeficientes são descritos na legenda

da figura. Os resultados foram calculados para a rede polonesa.

μ definições.

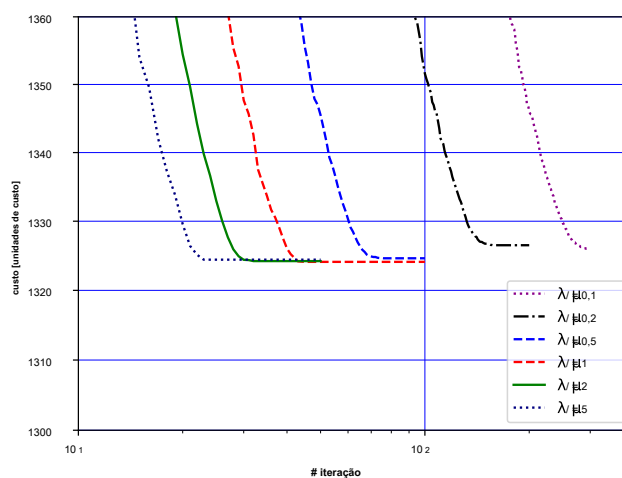


Figura 18. Curvas de convergência (custo vs. iterações) para diferentes combinações de μ e λ

Zoom da Figura 17 .

μ definições.

6. conclusões

Em suma, as seguintes conclusões podem ser tiradas da análise dos casos estudados. Para as redes Alemã10, Polonesa, Alemã15 e Alemã20, que são instâncias de redes pequenas, quase todos os métodos considerados calcularam as soluções ótimas. No entanto, para o método IP, o tempo de cálculo foi o mais baixo para as redes alemãs10 e polonesas. Os métodos HS e HC também podem ser usados com sucesso para esses casos. No entanto, para os algoritmos de enxame (BA e EA), os tempos de cálculo foram relativamente longos. Observamos que o algoritmo EA é o mais eficiente para redes grandes, com os conjuntos de dados usados neste artigo (ou seja, EUA, americano e alemão), embora o algoritmo BA seja quase tão bom.

Finalmente, é observado que os resultados se referem a redes DWDM de relevância prática e, portanto, fornecem orientação adicional para operadores de rede que estão planejando expansões de rede DWDM, e uma vez que algoritmos de enxame fornecem resultados promissores quando aplicados à otimização de rede DWDM, em um futuro próximo planeja continuar a pesquisa sobre a otimização de grandes redes sujeitas a restrições não lineares adicionais que são típicas de redes ópticas.

Contribuições do autor: Conceptualização, SK; metodologia, SK e SS; software, KW, AS, SK e M.Ż.; validação, SK e SS; análise formal, SK, SS e M.Ż.; recursos, SK, M.Ż. e SS; escrita - original preparação do esboço, SK e SS; redação - revisão e edição, SK, M.Ż. e SS; visualização, AS, SK e SS; supervisão, SK Todos os autores leram e concordaram com a versão publicada do manuscrito.

Financiamento: Esta pesquisa não recebeu financiamento externo.

Conflitos de interesse: Os autores declaram não haver conflito de interesses.

Referências

- Richardson, D.; Fini, J.; Nelson, L. Space Division Multiplexing in Optical Fibers. *Nat. Fotônica* **2013**, *7*, 354–362. [CrossRef]
- ILOG. *Manual do usuário do CPLEX 11.0*; ILOG: Genebra, Suíça, 2007. Garey, MR; Johnson, DS *Computadores e intratabilidade; Um Guia para a Teoria da NP-Completeness*; WH Freeman & Co.: New York, NY, USA, 1990.
- Klinkowski, M.; Walkowiak, K. Routing and Spectrum Assignment in Spectrum Sliced Elastic Optical Path Network. *IEEE Commun. Lett.* **2011**, *15*, 884–886. [CrossRef]
- Klinkowski, M.; Żotkiewicz, M.; Walkowiak, K.; Pióro, M.; Ruiz, M.; Velasco, L. Resolvendo grandes instâncias do problema RSA em redes ópticas flexíveis elásticas. *IEEE / OSA J. Opt. Comum. Netw.* **2016**, *8*, 320–330. [CrossRef]
- Cai, A.; Shen, G.; Peng, L.; Zukerman, M. Novel Node-Arc Model and Multiiteration Heuristics for Static Routing and Spectrum Assignment in Elastic Optical Networks. *J. Light. Technol.* **2013**, *31*, 3402–3413. [CrossRef]
- Kozdrowski, S.; Żotkiewicz, M.; Sujecki, S. Otimização de Redes Ópticas Baseadas na Tecnologia CDC-ROADM. *Appl. Sci.* **2019**, *9*, 399. [CrossRef]
- Żotkiewicz, M.; Ruiz, M.; Klinkowski, M.; Pióro, M.; Velasco, L. Reotimização de redes ópticas flexíveis dinâmicas após reparos de falha de link. *IEEE / OSA J. Opt. Comum. Netw.* **2015**, *7*, 49–61. [CrossRef]
- Dallaglio, M.; Giorgetti, A.; Sambo, N.; Velasco, L.; Castoldi, P. Routing, Spectrum e Transponder Assignment in Elastic Optical Networks. *J. Light. Technol.* **2015**, *33*, 4648–4658. [CrossRef] Kozdrowski, S.; Żotkiewicz, M.; Sujecki, S. Ultra-Wideband WDM Optical Network Optimization. *Fotônica* **2020**, *7*, 16. [CrossRef]
- Shariati, B.; Mastropaolo, A.; Diamantopoulos, N.; Rivas-Moscote, JM; Klionidis, D.; Tomkos, I. Avaliação de desempenho ciente da camada física de redes SDM com base em pacotes SMF, MCFs e FMFs. *IEEE / OSA J. Opt. Comum. Netw.* **2018**, *10*, 712–722. [CrossRef]
- Poggiolini, P.; Bosco, G.; Carena, A.; Curri, V.; Jiang, Y.; Forghieri, F. O modelo GN de propagação não linear de fibra e suas aplicações. *J. Light. Technol.* **2014**, *32*, 694–721. [CrossRef] Yang, XS *Algoritmos Metaheurísticos Inspirados na Natureza*; Livros-guia: Nairobi, Quênia, 2010.

- 14 Christodoulopoulos, K. ; Tomkos, I. ; Varvarigos, EA Elastic Bandwidth Allocation em Redes Óticas Flexíveis Baseadas em OFDM. *J. Light. Technol.* **2011**, *29*, 1354–1366. [[CrossRef](#)] Syswerda, G. *Crossover uniforme em algoritmos genéticos*; ICGA'89; Morgan Kaufmann: San Mateo, CA, EUA, 1989; pp. 2–9. [[CrossRef](#)]
- 16 Pham, D. ; Ghanbarzadeh, A. ; Koç, E. ; Otri, S. ; Rahim, S. ; Zaidi, M. The Bees Algorithm - A Novel Tool for Complex Optimization Problems. *Intell. Prod. Mach. Syst.* **2006**, 454–459. [[CrossRef](#)]
- 17 Karaboga, D. ; Basturk, B. Arti fi cial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. Em Proceedings of the Foundations of Fuzzy Logic and Soft Computing, 12º Congresso Mundial da International Fuzzy Systems Association, IFSA 2007, Cancún, México, 18–21 de junho de 2007; Volume 4529, pp. 789–798. [[CrossRef](#)]
- 18 Yuce, B. ; Packianather, M. ; Mastrocinque, E. ; Pham, D. ; Lambiase, A. Método de otimização inspirado nas abelhas: The Bees Algorithm. *Insetos* **2013**, *4*, 646–662. [[CrossRef](#)] [[PubMed](#)] Bäck, H. *Proceedings of the Seventh International Conference on Genetic Algorithms: Michigan State University, East Lansing, MI, USA, 19-23 de julho de 1997*; Morgan Kaufmann Publishers Inc. : San Francisco, CA, EUA, 1997. Michalkiewicz, Z. *Algoritmos Genéticos + Estruturas de Dados = Programas de Evolução*; Springer: Berlin, Germany, 1996. Arabas, J. ; Kozdrowski, S. Aplicando um algoritmo evolucionário ao projeto de redes de telecomunicações.
- 20 *Evol. Comput. IEEE Trans.* **2001**, *5*, 309–322. [[CrossRef](#)]
- 22 Yang, XS Harmony Search as a Metaheuristic Algorithm. 2010. Disponível online: <http://xxx.lanl.gov/abs/1003.1599> (acesso em 20 de setembro de 2020).
- 23 Turky, AM; Abdullah, S. ; McCollum, B. ; Sabar, NR An Evolutionary Hill Climbing Algorithm for Dynamic Optimization Problems. Em Proceedings of the 6th Multidisciplinary Int. conf. on Scheduling: Theory and Applications (MISTA 2013), Ghent, Bélgica, 27–30 de agosto de 2013.
- 24 Tomlin, J. Branch and Bound Methods for Integer and Nonconvex Programming. No *Programação inteira e não linear*; Abbie, J., Ed.; North-Holland: Amsterdam, The Netherlands, 1970; pp. 437–450. Wolsey, L. *Programação inteira*; John Wiley & Sons: New York, NY, USA, 1998.
- 26 Geem, ZW; Kim, JH; Loganathan, G. A New Heuristic Optimization Algorithm: Harmony Search. *SIMULAÇÃO* **2001**, *76*, 60–68. [[CrossRef](#)]
- 27 Beyer, HG; Schwefel, estratégias de evolução da HP - uma introdução abrangente. *Nat. Comput.* **2002**, *1*, 3–52. [[CrossRef](#)]
- 28 Arabas, J. ; Kozdrowski, S. Inicialização da população no contexto de uma mutação tendenciosa e específica do problema. Em Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEEWorld Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, EUA, 4-9 de maio de 1998; pp. 769–774. [[CrossRef](#)]
- 29 Orlowski, S. ; Wessäly, R. ; Pióro, M. ; Tomaszewski, A. SNDlib 1.0-Survivable Network Design Library. *Redes* **2010**, *55*, 276–286. [[CrossRef](#)]

