LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

Heurística Híbrida de Geração de Colunas com Aprendizado por Reforço para Resolução do Problema de Planejamento de Rede com Restrições de Caminho Mínimo

Cláudio André da Silva Alves
Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil
claudioalves@cos.ufrj.br

Abilio Pereira de Lucena Filho
Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil
abiliolucena@cos.ufrj.br

Pedro Henrique González
Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil
pegonzalez@cos.ufrj.br

RESUMO

As cidades estão enfrentando desafios significativos relacionados à infraestrutura de rede, transporte, logística e distribuição de recursos. Como exemplo, abordaremos o problema de dois níveis chamado Problema de Planejamento de Redes com Restrições de Caminho Mínimo (FCNDP-SPC), que trata do roteamento de múltiplos produtos desde suas origens até seus destinos. Cada produto deve ser transportado pelo caminho mais curto possível, considerando o comprimento dos arcos na rede criada. Para encontrar melhores soluções, integramos a geração de colunas com aprendizado por reforço, utilizando o aprendizado para determinar de maneira adaptativa quando prosseguir com o *soft-fixing*. Os resultados positivos encontrados fortalecem a contribuição desse estudo para o avanço do conhecimento de métodos híbridos.

PALAVRAS CHAVE. cidades inteligentes, aprendizado por reforço, geração de coluna.

ABSTRACT

Cities are facing significant challenges related to network infrastructure, transportation, logistics and resource distribution. As an example, we address the two-level problem known as Fixed Charge Network Design Problem with Shortest-path constraints (FCNDP-SPC), which deals with routing multiple products from their origins to their destinations. Each product must be transported via the shortest path possible, considering the arc lengths in the created network. To achieve better solutions, we integrate column generation with reinforcement learning, using the learning process to determine when to proceed with soft-fixing adaptively. The positive results obtained enhance the contribution of this study to advancing the knowledge of hybrid methods.

KEYWORDS. smart cities, reinforcement learning, column generation.

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

1. Introdução

A medida que os problemas complexos emergem em diversas áreas cada vez mais presentes no dia a dia, por exemplo, roteamento de veículos e alocação de recursos, a busca por alternativas de soluções eficientes torna-se cada vez mais crucial. Problemas de Programação Inteira Mista (MIP) apresentam um conjunto diversificado de desafios, exigindo abordagens inovadoras para sua resolução. Diante da crescente complexidade e tamanho desses problemas, técnicas e métodos para encontrar soluções eficientes e escaláveis estão sendo investigadas. Isso inclui o desenvolvimento de algoritmos avançados de otimização, como heurísticas e meta-heurísticas, além da integração de técnicas de inteligência artificial.

O planejamento de redes em cidades inteligentes é crucial para melhorar a eficiência e qualidade dos serviços prestados. Propostas voltadas para a aplicação em cidades inteligentes como a implantação de faixa reversíveis para veículos autônomos [Conceição et al., 2020], gerenciamento de tráfego em situações de emergência [Wan et al., 2017], ou transporte de materiais nocivos onde o governo deseja minimizar a exposição da população em caso de incidentes e por outro lado, o transportador quer minimizar o custo de transporte [Erkut e Gzara, 2008; Amaldi et al., 2011; Adamec et al., 2016]. Algumas dessas demandas vislumbradas como fundamentais para as cidades inteligentes já podem ser observadas em algum grau atualmente e podem ser modeladas como o Problema de Planejamento de Rede.

O Problema de Planejamento de Rede (FCNDP) consiste em selecionar um subconjunto de arestas de uma determinada rede, de forma que um conjunto de mercadorias possa ser transportado de suas origens até seus destinos. O objetivo é minimizar a soma dos custos fixos e dos custos variáveis que dependem das demandas de cada mercadoria. Aqui estamos interessados numa variante do FCNDP, chamada Problema de Planejamento de Rede com Restrições de Caminho Mínimo (FCNDP-SPC). Este problema combina o FCNDP com múltiplos problemas de caminhos mínimos, um para cada mercadoria.

O FCNDP-SPC envolve dois agentes distintos tomando decisões ao mesmo tempo, em vez de uma sequência. No primeiro nível, o líder tem como responsabilidade selecionar um conjunto de arestas a serem abertas, visando minimizar os custos fixos e variáveis. No segundo nível, o seguidor deve determinar os caminhos mínimos na rede para o envio de cada mercadoria. A influência entre os agentes não é direta, a escolha do seguidor é influenciada pela rede definida pelo líder no nível superior, enquanto a decisão do líder é afetada pelos custos variáveis das rotas estabelecidas pelo seguidor no nível inferior.

Mauttone et al. [2008] formulam o FCNDP-SPC como um problema inteiro de dois níveis e utilizam a metaheurística Busca Tabu como método de resolução. Gonzalez et al. [2016] propõe uma heurística com fixação de variáveis, seguida da aplicação do *local branching* e por fim adicionam uma estratégia de perturbação baseada em ciclos. Os resultados computacionais apresentados demonstram um excelente equilíbrio entre qualidade das soluções e tempo computacional. Bouras [2019] apresenta novas formulações para o problema baseadas em desigualdades de caminho e ciclo. Além disso, essas formulações são avaliadas utilizando a técnica de *branch-and-cut* e planos de corte, com resultados bastante promissores.

Durante a revisão bibliográfica sobre o problema não foram encontradas abordagens que usassem heurísticas de geração de colunas para a resolução do FCNDP-SPC, apesar de essas técnicas serem comumente empregadas em problemas que possuem estruturas interessantes para aplicar decomposições, por exemplo a divisão em subproblemas independentes ou apresentam a estrutura de bloco-diagonal [Bueno, 2005].

Assim como ocorre em diversos problemas, soluções eficientes dependem das formulações escolhidas, resolver o problema como um MIP pode demandar muitos recursos e ainda assim

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

não ser factível. A geração de colunas é uma técnica avançada utilizada para abordar grandes problemas de forma iterativa. Em vez de considerar todas as variáveis de uma vez, essa técnica adiciona variáveis apenas quando necessário, permitindo uma solução mais eficiente e reduzindo a carga computacional. Isso torna possível resolver problemas de grande escala que seriam impraticáveis de outra forma devido ao consumo excessivo de tempo e memória. Adicionalmente, podemos usar uma heurística de aprimoramento como o *soft-fixing* [Berthold, 2006] para encontrar soluções viáveis. Na sua aplicação, temos uma decisão sequencial, a cada passo de seguir ou não com a execução na busca de melhores soluções que sejam próximas a solução atual. É justamente nesse ponto que o aprendizado por reforço pode ser aplicado, decidindo quando continuar ou interromper o *soft-fixing*, adaptando-se dinamicamente às características do problema em questão e potencialmente melhorando a qualidade das soluções encontradas [Barto et al., 1989]. Técnicas de aprendizado de máquina tem sido utilizadas em problemas de otimização combinatória para melhorar a eficiência, a escalabilidade e a solução de problemas complexos [Bengio et al., 2021]. Desse modo, o presente trabalho tem como objetivo aplicar uma heurística de geração de coluna com uso do aprendizado por reforço, onde a decisão de prosseguir ou não com o *soft-fixing* é determinada por uma entidade tomadora de decisões.

O restante deste trabalho está estruturado da seguinte forma: Na Seção 2, o problema é detalhado juntamente com as formulações usadas para o problema. Na Seção 3, o método proposto é apresentado bem como o referencial teórico. Na Seção 4, os experimentos computacionais para validar o método são apresentados. Na Seção 5, as conclusões e direções futuras da pesquisa são apresentadas.

2. Descrição do problema e formulação matemática

O FCNDP-SPC é definido como um grafo não direcionado $G(V, E)$ onde V é o conjunto de nós que representam as instalações e E é o conjunto de arestas não capacitadas que conectam as instalações. O conjunto K representa as mercadorias que devem ser transportadas pela rede, sendo que, para cada mercadoria, existe uma demanda a ser entregue através de um caminho mais curto entre sua origem $o(k)$ e seu destino $d(k)$. Para cada aresta temos seu tamanho c_{ij}, o custo fixo de abertura f_e e o custo variável g_{ij}^k para transportar cada mercadoria pela aresta ij. A formulação de dois níveis é apresentada pelas Equações (1) a (7). A variável binária y_e é igual a 1 se a aresta (i, j) é utilizada na construção da rede e 0 caso contrário. A outra variável, x_{ij}^k representa o fluxo da mercadoria k no arco (i, j). Os arcos no conjunto A^E são obtidos bidirecionando as arestas em E, as expressões δ_i^+ e δ_i^- representam os conjuntos de todos os arcos saindo e entrando no nó $i \in V$, respectivamente.

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A^E} g_{ij}^k x_{ij}^k \tag{1}$$

$$\text{s.t} \quad y_e \in \{0, 1\} \qquad \forall e \in E, \tag{2}$$

onde x_{ij}^k é uma solução do problema:

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in A^E} c_{ij} x_{ij}^k \tag{3}$$

$$\text{s.t} \quad \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k \qquad \forall i \in V, \forall k \in K, \tag{4}$$

$$x_{ij}^k + x_{ji}^k \leq y_e, \qquad \forall e \in [i, j] \in E, \forall k \in K, \tag{5}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall (i, j) \in A^E, \forall k \in K \tag{6}$$

onde:

$$b_i^k = \begin{cases} -1 & \text{se } i = d(k), \\ 1 & \text{se } i = o(k), \\ 0 & \text{caso contrário.} \end{cases} \tag{7}$$

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

A Função Objetivo (1) minimiza os custos fixos e variáveis, enquanto a Função Objetivo (3) tem como finalidade encontrar o caminho mínimo para cada mercadoria. As Restrições (2) e (6) garantem que as variáveis assumam valores binários, a Restrição (4) garante a conservação de fluxo, a Restrição (5) garante que só exista fluxo em arcos que foram abertos, e por fim a Restrição (7) com o vetor de demanda $b_k \in \{-1, 0, 1\}^{|V|}$ garante que cada mercadoria tenha sua demanda total associada ao seu par origem-destino.

Embora a modelagem em dois níveis facilite o entendimento do problema sua resolução não é trivial. Mauttone et al. [2008] descreve uma nova formulação com a aplicação da linearização "big M" onde a formulação anterior é convertida para um problema de um nível. As variáveis $\pi_i^k, k \in K, i \in V$ representam o caminho mínimo entre o nó i e o nó $d(k)$. A restrição 11 refere-se às condições de otimalidade de Bellman para $|K|$ pares $(o(k), d(k))$.

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A^E} g_{ij}^k x_{ij}^k \tag{8}$$

$$\text{s.t} \quad \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k \qquad \forall i \in V, \forall k \in K, \tag{9}$$

$$x_{ij}^k + x_{ji}^k \leq y_e, \qquad \forall e \in [i,j] \in E, \forall k \in K, \tag{10}$$

$$\pi_i^k - \pi_j^k \leq M_{e(ij)} - y_{e(ij)}(M_{e(ij)} - c_{ij}) - 2c_{ij}x_{ij}^k, \qquad \forall (i,j) \in A^E, \forall k \in K, \tag{11}$$

$$\pi_{d(k)}^k = 0, \qquad \forall k \in K, \tag{12}$$

$$\pi_i^k \geq 0, \qquad \forall i \in V \backslash d(k), \forall k \in K \tag{13}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall (i,j) \in A^E, \forall k \in K \tag{14}$$

$$y_e \in \{0,1\} \qquad \forall e \in E, \tag{15}$$

onde:

$$b_i^k = \begin{cases} -1 & \text{se } i = d(k), \\ 1 & \text{se } i = o(k), \\ 0 & \text{caso contrário.} \end{cases} \tag{16}$$

O FCNDP-SCP pertence a classe problemas *NP-hard*, problemas aqueles que não são conhecidos algoritmos que resolvem de maneira exata em tempo polinomial [Johnson et al., 1978]. Por isso se faz necessário utilizar técnicas avançadas de otimização e estratégias híbridas para encontrar soluções viáveis e eficientes para problemas de planejemaento de redes.

3. Método proposto

Nessa seção o método proposto neste trabalho é apresentado em detalhes. Na Seção 3.1 a heurística de geração de colunas é explicada bem como a decomposição do modelo em duas partes: o problema mestre e o subproblema. Na Seção 3.2, o *soft-fixing* é apresentado como abordagem para aprimorar as soluções obtidas por um modelo MIP. Na sequência, a Seção 3.3 apresenta os principais conceitos do aprendizado por reforço e por fim na Seção 3.4 a descrição da proposta é apresentada consolidando os elementos teóricos discutidos anteriormente em um método.

3.1. Heurística de Geração de Coluna

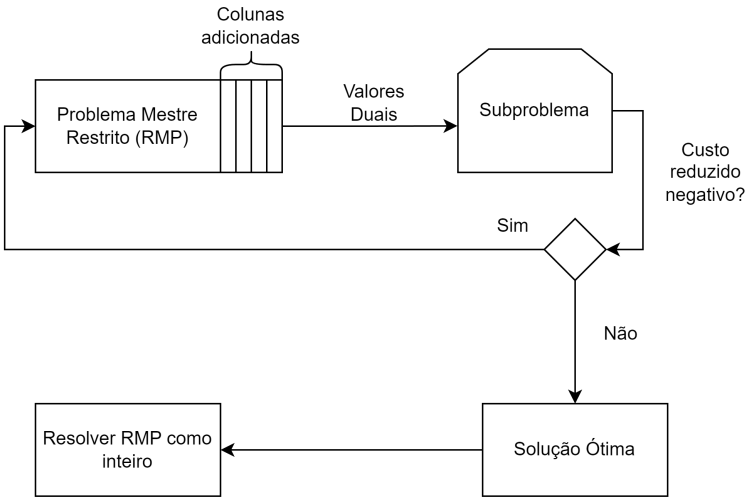
A geração de colunas é um método para resolver problemas lineares (LP) com um grande número de variáveis. O problema original é decomposto em duas partes, o problema mestre e o subproblema. Eles são separados de modo que para um problema $\min\{cx : Ax \geq b, Bx \geq d, x \geq 0\}$, eles possam ser separados de tal forma que resolver o problema sobre $Bx \geq d$ seja de mais fácil resolução e possa ser usado como subproblema [Wolsey, 2020].

A Figura 1 ilustra o funcionamento da técnica na resolução de um MIP. O Problema Mestre Restrito (RMP) contém todas as restrições do problema mestre mas apenas um conjunto das variáveis. Na sequência, subproblemas ou *pricing problem* são resolvidos com os valores duais

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

das restrições do problema mestre, onde os coeficientes da função objetivo são o custo reduzido. Desse modo, o intuito é identificar se existem variáveis que não estão contidas no RMP mas que podem melhorar a função objetivo, ou seja, possuem custo reduzido negativo em um problema de minimização. Caso contrário, se não existirem tais variáveis, é garantido que a solução atual do RMP é ótima e o processo termina. Por fim, o RMP é resolvido definindo as variáveis como inteiras.

Figura 1: Geração de Coluna aplicada a um MIP



Fonte: Elaborada pelo próprio autor

Gonzalez [2015] apresenta a decomposição do FCNDP-SPC que foi utilizada no presente trabalho. O problema mestre é responsável por utilizar as colunas geradas até o momento para encontrar uma solução viável e potencialmente ótima. Seja P_k o conjunto dos caminhos encontrados para cada mercadoria $(k \in K)$, z_p^k é definida como a variável binária associada a um caminho $p \in P_k$ da mercadoria $k \in K$ e r_p^k como seu custo. Para representar a utilização ou não de uma aresta em um caminho $p \in P_k$ para uma mercadoria $k \in K$, definiremos a variável multidimensional R_{ij}^{pk}, onde $(ij) \in A_E$, $k \in K$ e $p \in P_k$.

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{p \in P_k} r_p^k z_p^k \tag{17}$$

$$\text{s.t} \quad \sum_{p \in P_k} z_p^k = 1, \qquad \forall k \in K \tag{18}$$

$$y_e \geq \sum_{p \in P_k} z_p^k R_{pk}^{ij} + z_p^k R_{pk}^{ji}, \qquad \forall e \in [i,j] \in E, \forall k \in K, \tag{19}$$

$$\pi_i^k - \pi_j^k \leq M_{e(ij)} - y_{e(ij)}(M_{e(ij)} - c_{ij}) - 2c_{ij} \sum_{p \in P_k} z_p^k R_{pk}^{ji}, \qquad \forall (i,j) \in A^E, \forall k \in K \tag{20}$$

$$\pi_i^k = 0, \qquad \forall i = d(k), \forall k \in K, \tag{21}$$

$$\pi_i^k \geq 0, \qquad \forall i \in V, \forall k \in K, \tag{22}$$

$$z_p^k \in \{0,1\}, \qquad \forall k \in k, \forall p \in P_k, \tag{23}$$

$$y_e \in \{0,1\}, \qquad \forall e \in E. \tag{24}$$

Por outro lado, o subproblema tem a função de gerar novas colunas, ou seja, novas variáveis que possam melhorar a solução atual do problema mestre. Dessa forma, podemos resolver $|K|$ subproblemas de caminho mínimo, um para cada mercadoria $k \in K$.

$$\min \quad -\zeta_k + \sum_{(i,j) \in A^E} (g_{ij}^k - 2cij\lambda_{ji}^k + \beta_{e(ij)}^k) x_{ij}^k \tag{25}$$

$$\text{s.t} \quad \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k \qquad \forall i,j \in V, \tag{26}$$

$$x_{ij}^k \geq 0, \qquad \forall (i,j) \in A^E \tag{27}$$

onde:

$$b_i^k = \begin{cases} -1 & \text{se } i = d(k), \\ 1 & \text{se } i = o(k), \\ 0 & \text{caso contrário.} \end{cases} \tag{28}$$

Na Função Objetivo (25) do subproblema, os termos ζ_k, $\beta_{e(ij)}^k$ e λ_{ji}^k são as variáveis duais associadas as restrições 18, 19 e 20 respectivamente do problema mestre.

3.2. *Soft Fixing*

A heurística de aprimoramento *soft-fixing* é uma técnica na resolução de MIP onde uma grande parte das variáveis é fixada a seus valores de uma solução heurística inicial, permitindo que apenas um pequeno subconjunto de variáveis seja otimizado [Berthold, 2006]. Isso pode ser útil para focar o esforço computacional em uma parte mais gerenciável do problema, aumentando a eficiência e potencialmente encontrando melhores soluções.

O objetivo é equilibrar a fixação de variáveis suficientes para reduzir a complexidade do problema, mas sem restringir tanto o espaço de solução que boas soluções não possam ser encontradas. Considerando uma solução viável do FCNDP-SPC representada por $s = \langle \bar{y}, \bar{x} \rangle \in P$, onde P é o poliedro definido pelas Restrições (9) - (15) e seja $\bar{y}$ uma solução heurística para um problema binário com n variáveis, podemos adicionar a Restrição (29) ao modelo de modo que pelo menos α variáveis sejam iguais a solução inicial.

$$\sum_{e \in E | \bar{y_e} = 1} \bar{y}_e y_e \geq \lceil \alpha \sum_{e \in E | \bar{y_e} = 1} \bar{y}_e \rceil \tag{29}$$

Para o FCNDP-SPC, essa restrição tem como objetivo encontrar novas soluções que possuam um sub-conjunto das arestas que foram abertas e o parâmetro α serve para definir a similaridade entre as soluções, sua variação serve para fazer intensificação quanto mais próximo de 1 e diversificação a medida que é decrementado.

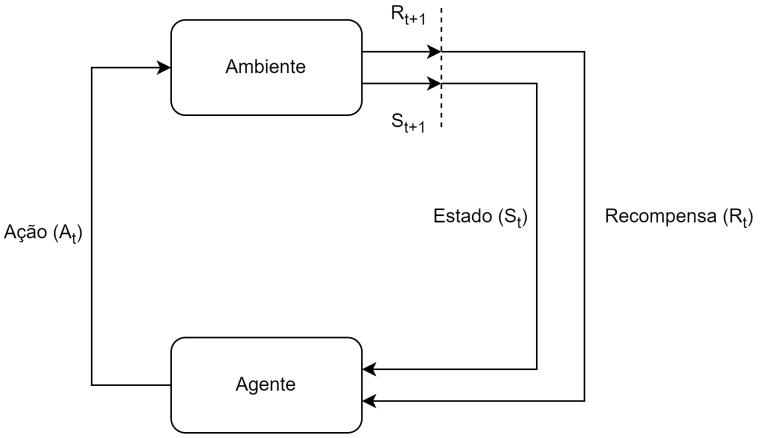
DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

3.3. Aprendizado por Reforço

O Aprendizado por Reforço ou *Reinforcement Learning* (RL) é um paradigma do aprendizado de máquina onde um agente aprende a tomar decisões interagindo com um ambiente [Arulkumaran et al., 2017]. O agente recebe *feedback* através de recompensas ou penalidades com base em suas ações. Através da exploração e exploração, o agente refina sua estratégia de tomada de decisão para maximizar as recompensas acumuladas ao longo do tempo. O Deep Q-Learning (DQN) combina o aprendizado por reforço com redes neurais para lidar com espaços de estado complexos e de alta dimensionalidade, a ideia é aproximar a função Q que representa a recompensa futura esperada por tomar uma ação em um determinado estado.

No aprendizado por reforço, os agentes tomam decisões sequenciais, ele tem sido utilizado e estudado na área da saúde [Esteva et al., 2019], na robótica [Levine et al., 2016] e em jogos [Mnih et al., 2015]. A Figura 2 resume os principais passos. Em um determinado tempo t, o agente recebe o estado atual S_t do ambiente. Com base no estado recebido e em sua política, o agente escolhe uma ação A_t para executar no ambiente. Após a transição para o próximo estado S_{t+1}, o ambiente fornece a recompensa R_{t+1}. O agente pode usar o conhecimento para melhorar sua política e aprender [Mnih et al., 2013].

Figura 2: Esquema básico do aprendizado por reforço



Fonte: Elaborada pelo próprio autor

Abordando o aprendizado por reforço para a proposta que estamos lidando nesse trabalho, o agente recebe como entrada um estado, contemplando a solução obtida pela geração de coluna, o valor do α que controla a similaridade entre a solução inicial e a solução que será encontrada usando o *soft-fixing*, a quantidade de iterações sem melhora, o valor da solução obtida resolvendo o RMP com a restrição de *soft-fixing*, e o valor da solução exata. Para aprender a política ótima de decisão usamos o algoritmo DQN (Deep Q-Network) [Mnih et al., 2013], utilizando uma rede neural *feedforward* com 3 camadas. Com essas informações, o agente é capaz de decidir se o processo do *soft-fixing* deve ser interrompido ou continuar na expectativa de encontrar soluções com melhores custos. O ambiente nesse caso é a própria execução do modelo com a restrição de

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

soft-fixing, a recompensa reflete a qualidade da solução e a eficiência computacional, ou seja, ela é positiva caso uma solução de melhor qualidade for encontrada e negativa caso contrário porque um tempo computacional adicional foi gasto sem sucesso.

3.4. Método

Uma heurística de geração de coluna foi introduzida por Abilio Lucena e Carlos Henrique Sabóia no ano de 2018, em um projeto de pesquisa para o CEPEL/ELETROBRAS. O procedimento foi inicialmente proposto para resolver uma formulação envolvendo exponencialmente muitas variáveis para o problema de Planejamento a Longo Prazo da Expansão da Geração e Transmissão de Energia Elétrica. O procedimento se tornou público na tese de doutorado de Carlos Henrique Sabóia [Sabóia, 2013].

A combinação da técnica de geração de colunas com heurísticas como o *soft-fixing*, potencializada pelo aprendizado por reforço, chamada aqui de GC-RLSF, representa uma abordagem promissora para resolver problemas de otimização. Ao adaptar dinamicamente a estratégia de busca durante o processo de resolução, podemos aproveitar melhor os recursos computacionais e encontrar soluções mais próximas do ótimo e em menor tempo.

4. Experimentos

Os experimentos foram executados em um notebook com processador Intel Core ^{TM} I5 10210U 1,6 GHz e 16GB de memória RAM. Os algoritmos foram implementados utilizando a linguagem de programação C++. O framework *PyTorch* [Paszke et al., 2019] foi utilizado para construir as estruturas do aprendizado por reforço, o *solver* CPLEX [IBM Corporation, 2022] na versão 22.1.1 foi utilizado. As instâncias foram obtidas do trabalho de Mauttone et al. [2008], o primeiro campo representa a quantidade de nós no grafo, seguido da densidade do grafo e por fim a quantidade de mercadorias a serem transportadas.

O parâmetro α que controla a proporção inicial de variáveis fixadas foi definido como $0,9$ e é decrementado gradualmente em $0,1$ unidades até o limite inferior de $0,1$ caso melhores soluções não sejam encontradas. Nesse caso, a cada iteração mais variáveis são liberadas permitindo um maior grau de liberdade para explorar a vizinhança. Quando uma melhor solução é encontrada, o parâmetro α é reinicializado.

Para realizar o treinamento do agente, uma amostra de 30% das instâncias foi utilizada. Para cada instância, cada técnica foi executada um total de 20 vezes. A Tabela 1 reúne os resultados obtidos, a primeira coluna apresenta o valor da solução exata, e para cada técnica utilizada como comparativo é apresentado o módulo do *gap* comparando com a solução exata e o tempo gasto em segundos.

A abordagem de aplicar a relaxação linear fornece soluções rapidamente porém o valor ótimo só foi alcançado em 15 instâncias de um total de 40 instâncias testadas. Ao aplicar a geração de coluna, com relação aos *gaps* não temos diferenças comparadas a relaxação linear e para as instâncias maiores e mais densas, por exemplo com 20 nós, o tempo de processamento chegou na média de 25 segundos em uma das instâncias avaliadas. A utilização de outros algoritmos para resolução do subproblema pode ser avaliada como um possível ponto de melhoria no tempo de processamento. A aplicação do *soft-fixing* foi extremamente benéfica, em apenas seis instâncias o valor ótimo não foi alcançado, e nesses casos o gap está em torno de 0,02%.

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

Tabela 1: Resultados obtidos comparando as abordagens

Instância	Solução exata	Relaxação Linear		Geração de Coluna		Geração de Coluna com Soft Fixing		GC-RLSF	
		Gap	Tempo	Gap	Tempo	Gap	Tempo	Gap	Tempo
10_0.300000_5_001	3942	0	0.0032	0	0.0361	0	0.0858	0	0.0627
10_0.300000_5_002	4552	0	0.0027	0	0.0223	0	0.0749	0	0.0634
10_0.300000_5_003	5762	0	0.0026	0	0.0421	0	0.0885	0	0.0680
10_0.300000_5_004	4811	0	0.0027	0	0.0199	0	0.0622	0	0.0512
10_0.300000_5_005	4831	0	0.0028	0	0.0339	0	0.0832	0	0.0827
10_0.300000_10_001	8331	0.0185	0.0045	0.0185	0.0621	0	0.1289	0	0.0972
10_0.300000_10_002	8812	0	0.0042	0	0.0527	0	0.1159	0	0.1170
10_0.300000_10_003	10016	0	0.0041	0	0.1100	0	0.1705	0	0.1324
10_0.300000_10_004	8750	0	0.0041	0	0.0764	0	0.1388	0	0.1165
10_0.300000_10_005	10130	0	0.0041	0	0.1087	0	0.1665	0	0.1331
10_0.300000_15_001	12490	0	0.0056	0	0.0820	0	0.1472	0	0.1150
10_0.300000_15_002	17417	0	0.0056	0	0.1138	0	0.1700	0	0.1439
10_0.300000_15_003	12378	0	0.0054	0	0.3216	0	0.2906	0	0.2782
10_0.300000_15_004	10988	0.0016	0.0056	0.0016	0.1459	0	0.1813	0	0.1533
10_0.300000_15_005	9066	0	0.0055	0	0.1617	0	0.1974	0	0.1648
20_0.300000_10_001	5978	0.0401	0.0221	0.0401	3.2494	0	3.3074	0	2.2765
20_0.300000_10_002	10469	0.0866	0.0285	0.0866	3.1800	0.0144	4.0070	0.0144	3.0716
20_0.300000_10_003	7020	0.0545	0.0253	0.0545	5.2348	0	5.7048	0	3.5542
20_0.300000_10_004	5484	0.1156	0.0263	0.1156	2.6203	0.0108	4.1396	0.0108	3.1948
20_0.300000_10_005	7932	0.0667	0.0249	0.0667	1.9106	0.0144	2.5702	0.0144	2.2347
20_0.300000_20_001	9488	0.0061	0.0377	0.0061	9.3303	0	5.9767	0	3.4767
20_0.300000_20_002	11521	0.0424	0.0370	0.0424	6.0073	0	4.5201	0	4.1222
20_0.300000_20_003	8270	0.0364	0.0364	0.0364	4.2376	0	4.2914	0	4.0339
20_0.300000_20_004	11901	0.0380	0.0633	0.0380	24.1752	0	21.4183	0	19.2775
20_0.300000_20_005	9656	0.0286	0.0459	0.0286	5.9058	0	6.1221	0	4.0945
20_0.500000_10_001	4784	0.0827	0.0827	0	11.2258	0	14.0607	0	11.0972
20_0.500000_10_002	7689	0.0505	0.0505	0	17.9093	0	18.8992	0	15.5916
20_0.500000_10_003	6184	0.0188	0.0188	0	2.9762	0	3.6667	0	2.1963
20_0.500000_10_004	5189	0.0561	0.0561	0	4.7653	0	5.5284	0	4.2947
20_0.500000_10_005	6051	0.0624	0.0624	0	7.1264	0	8.4011	0	5.8761
10_0.800000_10_001	5813	0	0.0043	0.0220	0.5626	0.0153	0.9712	0.0153	0.9723
10_0.800000_10_002	4830	0.0869	0.0055	0	0.1790	0	0.2864	0	0.1971
10_0.800000_10_003	3499	0.0625	0.0047	0	1.4345	0	1.6810	0	1.3429
10_0.800000_10_004	5364	0.0821	0.0044	0	0.4826	0	0.6277	0	0.6103
10_0.800000_10_005	4133	0.0280	0.0045	0.0670	0.7934	0	1.2264	0	1.1372
10_0.800000_15_001	6822	0.0009	0.0106	0.0009	0.5079	0.0166	0.7938	0.0166	0.7450
10_0.800000_15_002	5183	0.0020	0.0101	0.0020	0.6516	0.0255	0.9716	0.0255	0.7838
10_0.800000_15_003	4523	0.0165	0.0100	0.0165	0.5648	0	0.8384	0	0.7123
10_0.800000_15_004	7484	0.0164	0.0102	0.0164	0.3560	0	0.5718	0	0.4180
10_0.800000_15_005	3843	0	0.0102	0	1.4431	0	1.7157	0	1.2168

A técnica de *soft-fixing* na geração de colunas foi analisada em termos do tempo adicional gasto, e os resultados são bastante promissores. As diferenças observadas foram mínimas, com uma média de menos de 2 segundos. Este tempo adicional ínfimo sugere que a técnica não só é eficiente, mas também não impõe um ônus significativo ao processamento. Ao analisar o método proposto, esse cenário é ainda melhor, devido sua capacidade de interromper execuções que seriam infrutíferas. Em termos práticos, isso significa que o método conseguiu identificar quando a busca por melhores soluções não iria resultar em melhorias significativas e pode, portanto, encerrar a execução prematuramente.

Os resultados do método proposto mostraram um comportamento consistente em termos dos *gaps* obtidos. O desempenho da técnica em minimizar os *gaps* é comparável ao da abordagem clássica de geração de colunas, demonstrando que o *soft-fixing* e da versão com o uso do aprendizado por reforço não comprometem a qualidade das soluções finais.

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

5. Conclusão e Trabalhos Futuros

No contexto das cidades inteligentes, onde há uma crescente demanda por soluções eficientes de planejamento de redes, abordagens híbridas que combinam técnicas de otimização tradicionais com aprendizado de máquina podem desempenhar um papel crucial na resolução desses desafios complexos.

Possíveis trabalhos futuros incluem avaliar a proposta com instâncias maiores e abordar outros algoritmos para resolução dos subproblemas. Outro ponto de interesse é atuar na estabilização da geração de colunas, impedindo que as variáveis duais oscilem entre as interações e permita uma convergência mais rápida. Essa abordagem também pode ser feita através do aprendizado por reforço onde o agente é responsável por decidir se naquele momento a coluna deve ser adicionada ou não ao problema mestre.

O emprego do aprendizado por reforço oferece uma oportunidade única para melhorar ainda mais a eficácia da geração de colunas. Ao introduzir uma entidade tomadora de decisões que aprende dinamicamente quando prosseguir ou interromper o processo de *soft-fixing*, adaptando-se às características específicas do problema em tempo real, podemos explorar de forma mais inteligente o espaço de soluções em busca de melhores resultados.

Agradecimentos

Os autores agradecem à UFRJ (23079.227622/2023-70), CNPq (307663/2021-3), CNPq (310185/2021-1) e FAPERJ (E-26/201.341/2022) pelo financiamento parcial desta pesquisa.

Referências

Adamec, V., Schüllerová, B., e Adam, V. (2016). Issues of hazardous materials transport and possibilities of safety measures in the concept of smart cities. In *Smart City 360°: First EAI International Summit, Smart City 360°, Bratislava, Slovakia and Toronto, Canada, October 13-16, 2015. Revised Selected Papers 1*, p. 790–799. Springer.

Amaldi, E., Bruglieri, M., e Fortz, B. (2011). On the hazmat transport network design problem. In *International Conference on Network Optimization*, p. 327–338. Springer.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., e Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

Barto, A. G., Sutton, R. S., e Watkins, C. (1989). *Learning and sequential decision making*, volume 89. University of Massachusetts Amherst, MA.

Bengio, Y., Lodi, A., e Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421.

Berthold, T. (2006). *Primal heuristics for mixed integer programs*. PhD thesis, Zuse Institute Berlin (ZIB).

Bouras, I. (2019). *Fixed charge network design problem with user-optimal flows*. PhD thesis, Université Montpellier.

Bueno, E. F. (2005). *Geração de Colunas em Problemas de Otimização Combinatória*. PhD thesis, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO.

Conceição, L., Correia, G. H. d. A., e Tavares, J. P. (2020). The reversible lane network design problem (rl-ndp) for smart cities with automated traffic. *Sustainability*, 12(3):1226.

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br

Erkut, E. e Gzara, F. (2008). Solving the hazmat transport network design problem. *Computers & Operations Research*, 35(7):2234–2247.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., e Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1): 24 – 29.

Gonzalez, P. H. (2015). *Studies On Network Design Problems*. PhD thesis, Universidade Federal Fluminense.

Gonzalez, P. H., Simonetti, L., Michelon, P., Martinhon, C., e Santos, E. (2016). A variable fixing heuristic with local branching for the fixed charge uncapacitated network design problem with user-optimal flow. *Computers & Operations Research*, 76:134–146.

IBM Corporation. *IBM ILOG CPLEX Optimization Studio*. IBM Corporation, 2022. URL `https://www.ibm.com/analytics/cplex-optimizer`. Versão 22.1.1.

Johnson, D. S., Lenstra, J. K., e Kan, A. R. (1978). The complexity of the network design problem. *Networks*, 8(4):279–285.

Levine, S., Finn, C., Darrell, T., e Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Mauttone, A., Labbé, M., e Figueiredo, R. (2008). A tabu search approach to solve a network design problem with user-optimal flows. In *ALIO/EURO Workshop on Applied Combinatorial Optimization*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., e Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., e Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library.

Sabóia, C. H. M. (2013). *Um Algoritmo Branch-and-Price para Instâncias de Grande Porte do Modelo Brasileiro de Planejamento da Expansão da Geração Elétrica a Longo Prazo*. PhD thesis, Programa de Engenharia de Sistemas e Computação - PESC/COPPE, Universidade Federal do Rio de Janeiro.

Wan, S., Lu, J., Fan, P., e Letaief, K. B. (2017). To smart city: Public safety network design for emergency. *IEEE access*, 6:1451–1460.

Wolsey, L. A. (2020). *Integer programming*. John Wiley & Sons.

DOI: 10.59254/sbpo-2024-193386
https://proceedings.science/p/193386?lang=pt-br