Minimum Broadcast Time

Pedro Fiorio e Rafael Zardo

INF 284 2023.1

Contexto

O Minimum Broadcast Time(MBT) é um problema clássico de disseminação de dados em rede. Ele é apresentado no livro Computers and Intractability (Garey and Johnson,1979) como NP-Difícil.

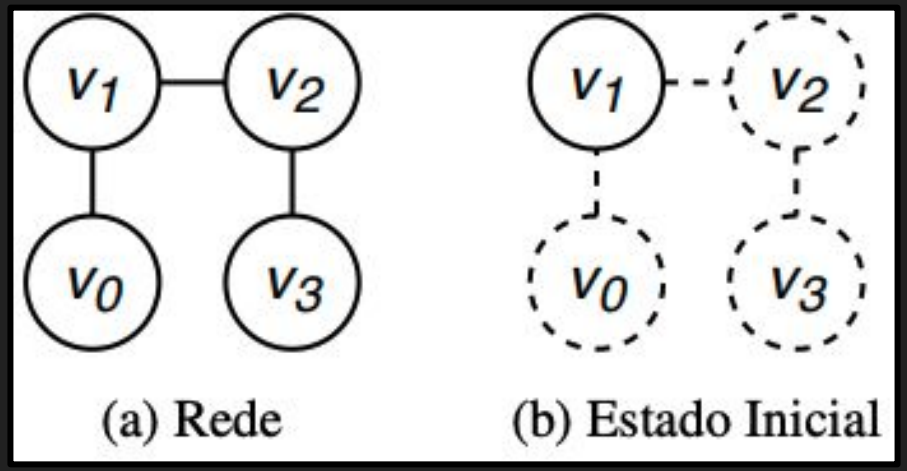
Aplicações:

- Redes de sensores sem fio
- Comunicação entre multiagentes
- Redes de satélites
- Direct memory access

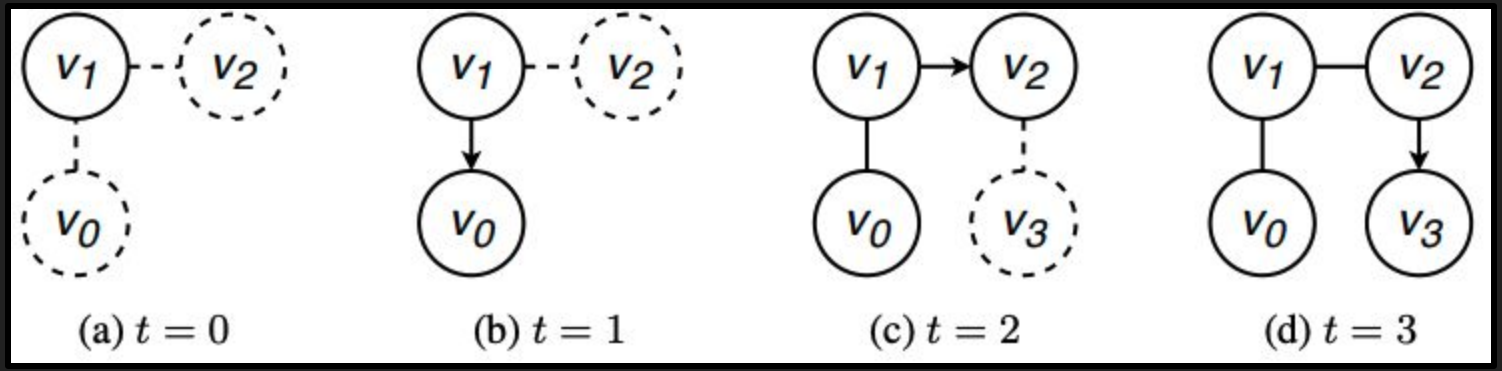
Objetivo

O problema pode ser exposto da seguinte maneira, imagine que um ou alguns dispositivos possuem uma mensagem a ser transferida aos demais dispositivos. Todos os dispositivos pertencem à mesma rede e devem receber a mensagem no menor tempo possível. No entanto, há algumas limitações. Cada dispositivo só pode realizar uma transferência por vez aos seus vizinhos imediatos. Quando um dispositivo recebe a mensagem, ele também fica responsável por enviar a mensagem para seus respectivos vizinhos imediatos.

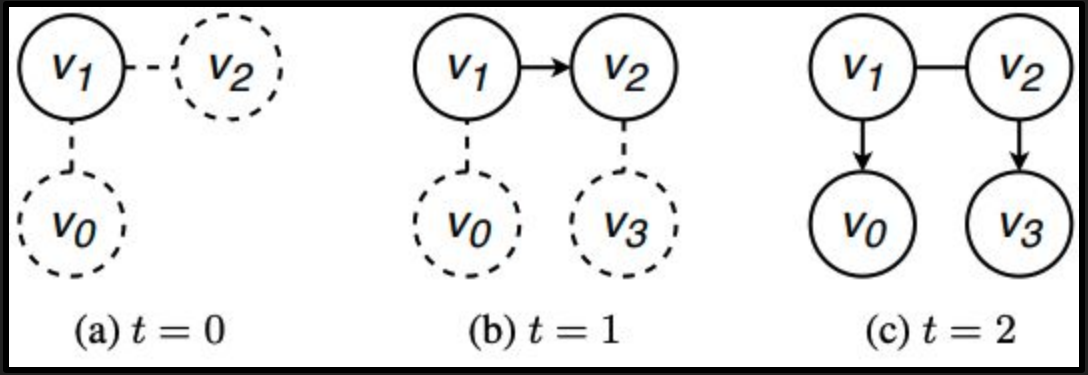
Exemplo



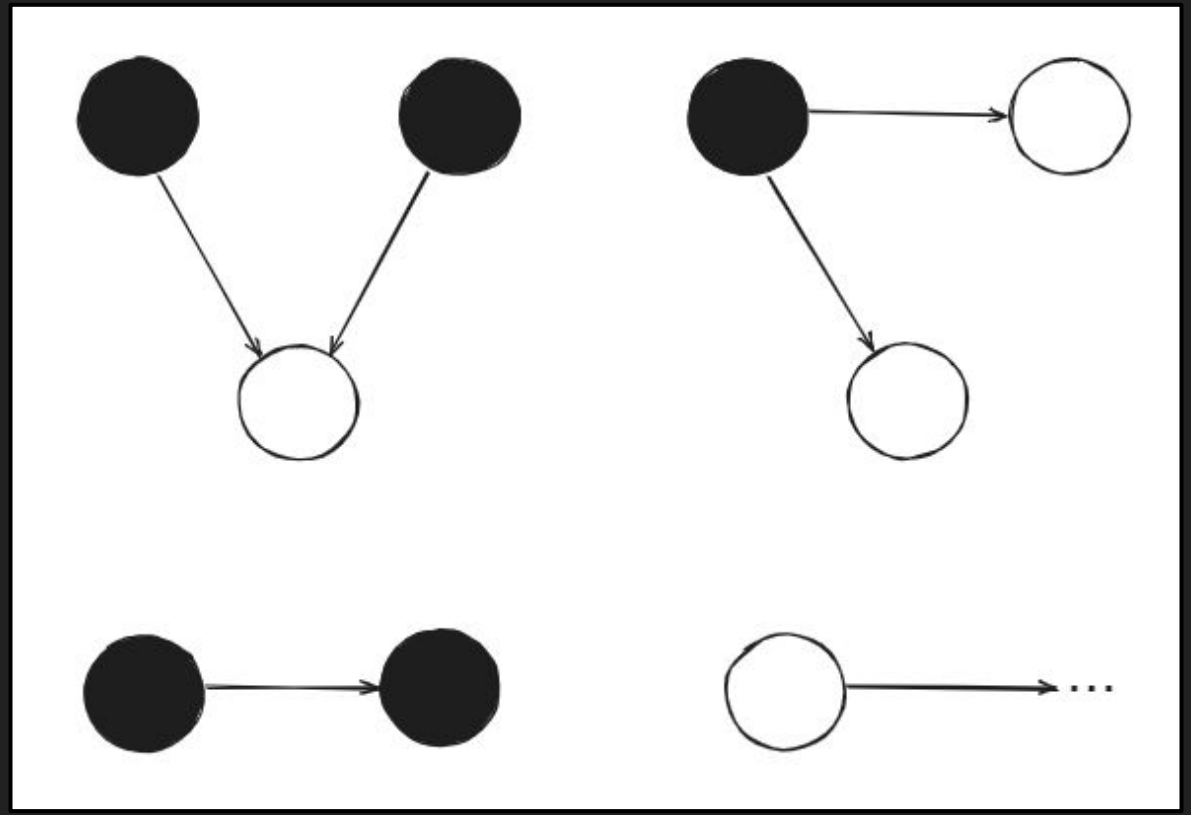
Exemplo - Solução Viável



Exemplo - Solução Ótima



Restrições

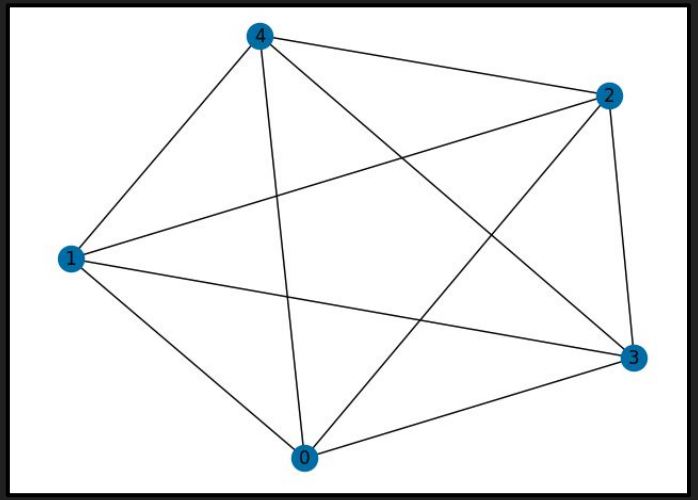


Métodos Já Empregados Por Outros

- ACO - Hasson and Sipper (2004)
- ILP - de Sousa et al. (2018)
- TreeBlock - de Sousa et al. (2018)
- ILP - Ivanova (2019)
- BRKGA - Alfredo Lima et al. (2020)

Representação de Solução

Nossa representação para uma solução é um vetor de prioridade, em que o valor de cada posição representa um vértice. Estar mais à esquerda na solução significa ter maior prioridade (tanto para enviar, quanto para receber informação).



H4,5

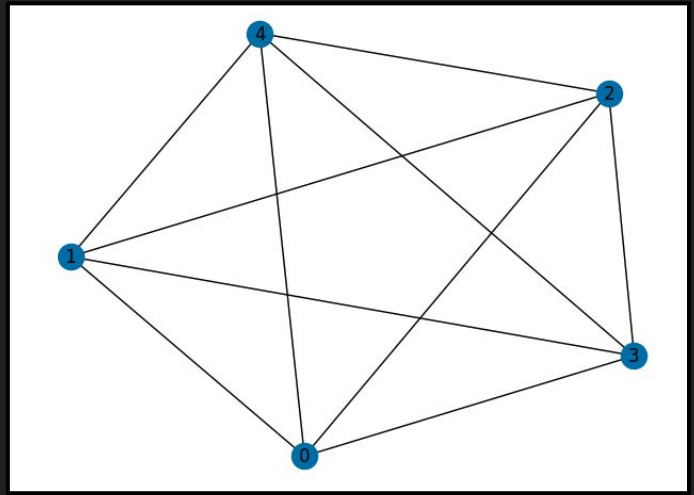
Possíveis soluções:

1. [0,4,2,1,3]
2. [4,1,3,2,0]
3. [1,4,2,0,3]

OBS: O primeiro item da lista, é sempre o primeiro detentor da informação no grafo.

Avaliação de Solução

Dada uma solução, precisamos avaliá-la. Vamos utilizar o exemplo anterior para demonstrar o cálculo da avaliação.



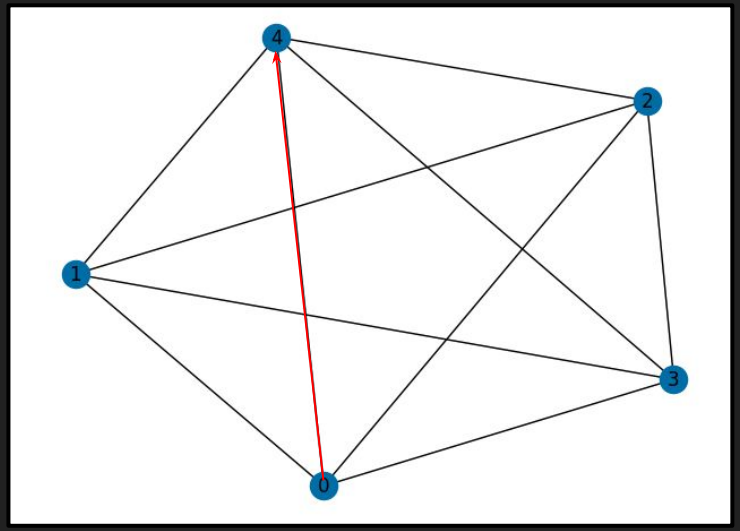
Solução:

[0,4,2,1,3]

Usaremos um vetor auxiliar para marcar quais vértices já foram informados:

[V,F,F,F,F]

Tempo 1



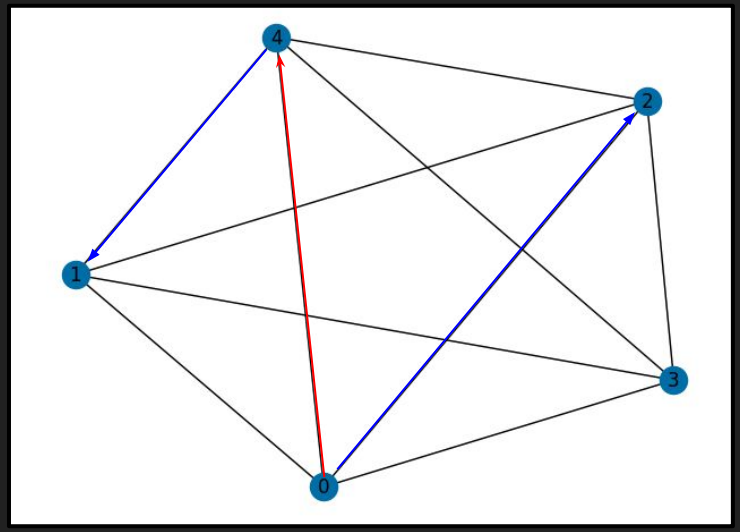
Solução:
[0,4,2,1,3]

[V,F,F,F,F]

Devemos verificar o vizinho de 0 com maior prioridade, que ainda não detêm a informação. Nesse caso 4, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.
[V,F,F,F,V]

Como não existe outro item que detêm a informação (passada no início do tempo), não há mais iterações do tempo 1.

Tempo 2



Solução:

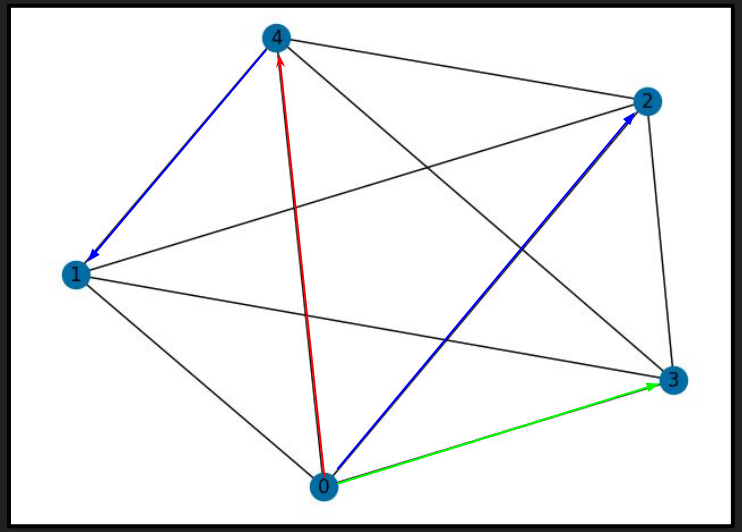
[0,4,2,1,3]
[V,F,F,F,V]

Devemos verificar o vizinho de 0 com maior prioridade, que ainda não detêm a informação. Nesse caso 2, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.
[V,F,V,F,V]

Além disso, devemos verificar o vizinho de 4 com maior prioridade, que ainda não detêm a informação. Nesse caso 1, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.
[V,V,V,F,V]

Como não existe outro item que detêm a informação (passada no início do tempo), acabaram as iterações do tempo 2.

Tempo 3



Solução:

[0,4,2,1,3]
[V,V,V,F,V]

Devemos verificar o vizinho de 0 com maior prioridade, que ainda não detêm a informação. Nesse caso 3, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.
[V,V,V,V,V]

Como a lista de visitados foi completa, retornamos o tempo atual, ou seja, 3.

Scatter Search - Indivíduo

Antes de introduzir como foi feito, faz-se necessário entender os cromossomos dos indivíduos. Cada indivíduo, possui um vetor de números (cromossomo), que codifica uma possível solução. Conforme exemplo abaixo:

Cromossomo:

[3, 2, 0, 4, 1]

Solução Gerada:

[2, 4, 1, 0, 3]

O valor de Cromossomo[i] diz qual é a prioridade de V_i

Scatter Search - População

Em seguida, geramos uma população inicial de forma totalmente determinística. Isso significa que os cromossomos são criados a partir das permutações de grupos da solução trivial ([0, 1, 2, 3, ..., n-1]) e das permutações do inverso da solução trivial. Para ilustrar, vamos usar o seguinte exemplo:

Cromossomo Trivial: [0, 1, 2, 3, 4]

50% da população:

é gerada a partir das permutações de [0, 1, 2, 3, 4].

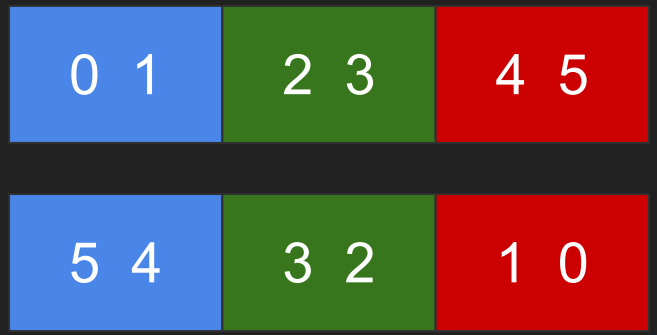
50% da população:

é gerada a partir das permutações de [4, 3, 2, 1, 0].

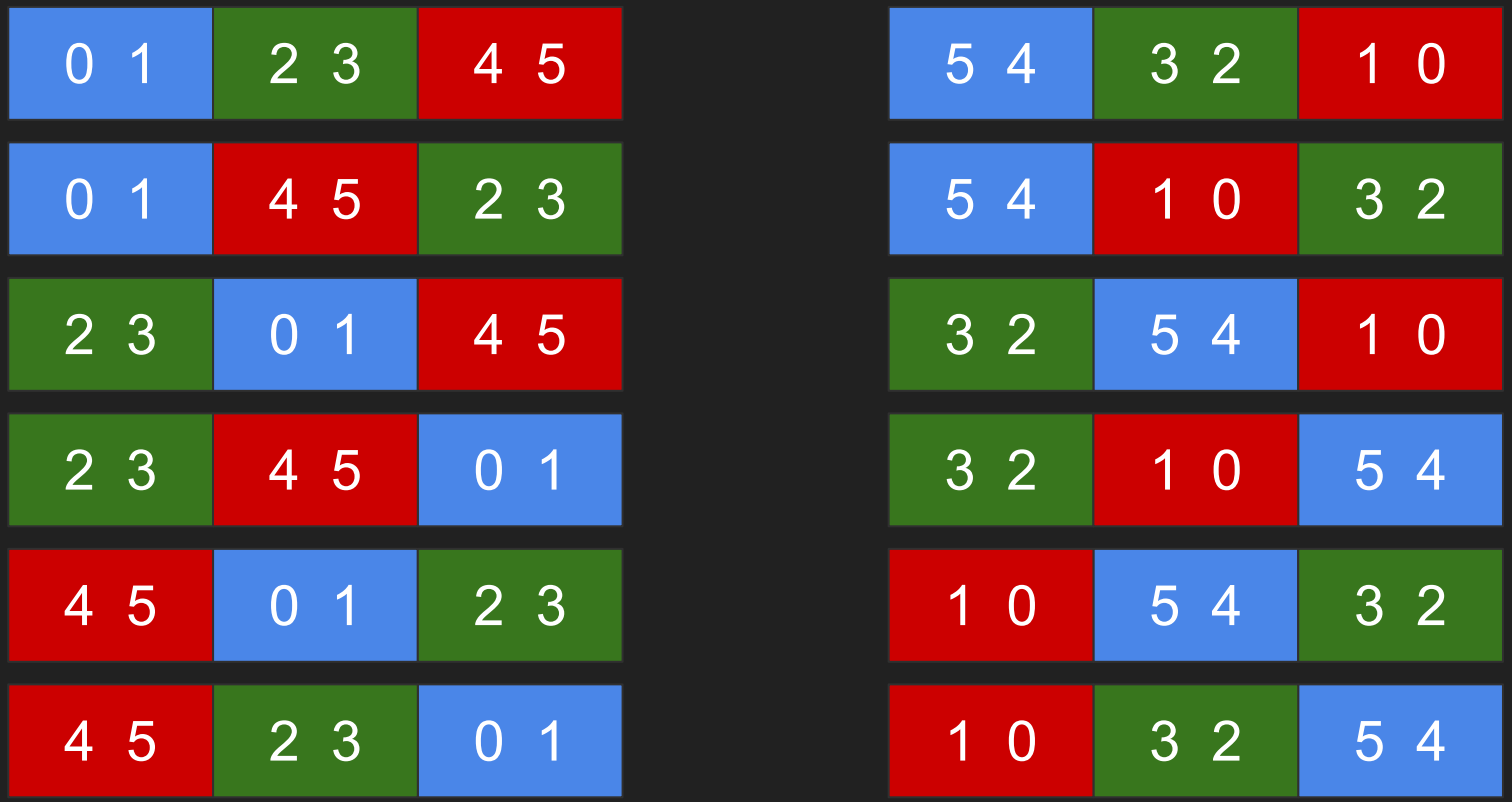
Essa abordagem garante que a população inicial seja diversificada e cubra uma ampla gama de possibilidades de solução.

Exemplo de Geração de População Inicial

- Cromossomo trivial: [0, 1, 2, 3, 4, 5]
- Cromossomo inverso: [5, 4, 3, 2, 1, 0]
- População: 6 indivíduos
- Dividimos os vetores em um número de grupos capaz de gerar 6 ou mais permutações



Exemplo de Geração de População Inicial



Exemplo de Geração de População Inicial

Scatter Search - Criar Conjunto de Referência

Em seguida, dado um tamanho para o conjunto de referência, precisamos criar o conjunto de referência a partir da população. Ele é criado da seguinte forma:

50% do conjunto de referência:

Consiste nos indivíduos da população que obtiveram os menores tempos para as instâncias. Esses indivíduos são selecionados com base no critério de desempenho, ou seja, aqueles que alcançaram os melhores resultados.

50% do conjunto de referência:

Inclui os indivíduos que não estão presentes no conjunto de referência e que possuem o maior número de valores diferentes em suas soluções em comparação com todos os indivíduos que já estão no conjunto de referência.

Essa abordagem visa garantir uma representação diversificada no conjunto de referência, incorporando tanto os indivíduos de melhor desempenho quanto aqueles com soluções menos semelhantes às já existentes. Dessa forma, o conjunto de referência abrange uma variedade de soluções promissoras para serem exploradas durante o processo de otimização.

Scatter Search - Criar Pais

Em seguida, criado o conjunto de referência, cria-se o conjunto de pais, a partir de todas combinações de indivíduos dois a dois presentes no conjunto de referência. Conforme exemplo abaixo:

Indivíduos: [1,2,3]

Conjunto de Pais: [1-2,1-3, 2-1, 2-3, 3-1, 3-2]

Scatter Search - Criar Filhos

Com o conjunto de pais, eles estão sujeitos a quatro tipos de reprodução para criar novos filhos. Toda a reprodução ocorre em torno dos cromossomos do indivíduo.

Primeira Reprodução: Média = (Cr1 + Cr2) / 2

Segunda Reprodução: Média Geométrica = sqrt(Cr1 * Cr2)

Terceira Reprodução: 1º gene do Pai, 2º gene da Mãe, 3º gene do Pai, 4º gene da Mãe….

Quarta Reprodução: Distância Euclidiana = $(Cr1 - Cr2)^2$

Esses quatro métodos de reprodução são utilizados para criar variedade genética nos filhos, combinando os cromossomos dos pais de maneiras diferentes. Cada método tem suas próprias características e pode levar a diferentes resultados na descendência.

Scatter Search - Atualizar o Conjunto de Referência

Após a geração dos filhos, é importante realizar uma troca entre os melhores filhos e os piores indivíduos do conjunto de referência. Nesse processo, caso um filho seja considerado superior a qualquer indivíduo do conjunto de referência, ele substituirá o membro correspondente no conjunto.

Essa estratégia de substituição tem como objetivo promover a evolução contínua da população, privilegiando os filhos mais promissores em relação aos indivíduos menos adaptados presentes no conjunto de referência. Dessa forma, o conjunto de referência é constantemente atualizado com os indivíduos mais aptos, permitindo que a próxima geração seja formada por descendentes que possuam características genéticas superiores.

Scatter Search - Iterar sem melhora

Após uma iteração sem melhora, ou seja, quando o indivíduo com o menor tempo encontrado não apresenta melhorias em relação às iterações anteriores, é necessário alterar o método de reprodução utilizado. Nesse caso, adota-se uma abordagem sequencial de tentativas, em que são testados diferentes métodos de reprodução até encontrar aquele que possa gerar melhorias no desempenho.

Se, em qualquer uma das tentativas, houver melhora no desempenho, os métodos de reprodução são reiniciados, e o processo continua iterando. Porém, caso nenhuma melhora seja obtida após todas as tentativas, o algoritmo do Scatter Search é finalizado, retornando a solução do melhor indivíduo.

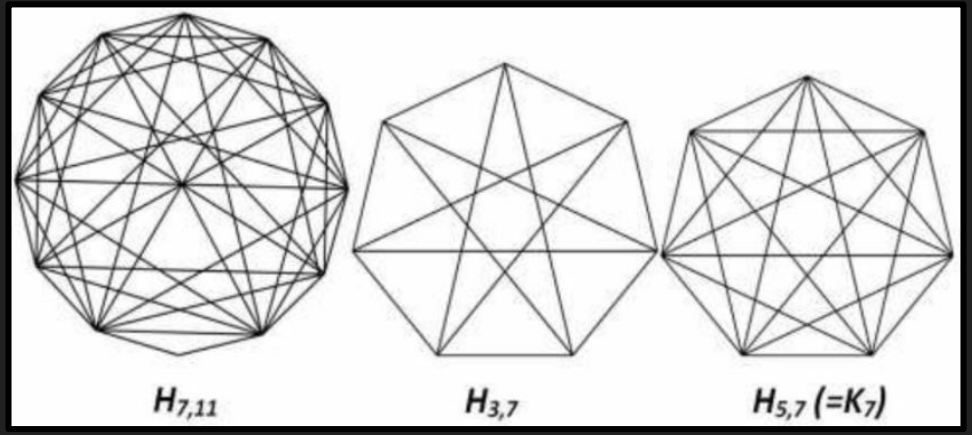
Busca Tabu

A solução inicial é o vetor de prioridade trivial. Os movimentos de melhoria são baseados na troca de elementos (swap), testando todas as combinações possíveis. Os movimentos válidos são aqueles que não estão na lista tabu ou que atendem ao critério de aspiração (melhor solução). Após encontrar todos os movimentos válidos, escolhemos o melhor e atualizamos a lista tabu.

Esse processo de busca é repetido até que um tempo definido seja alcançado.

Instâncias

O grafo de Harary H(k,n) é um exemplo particular de grafo k-conexo com n vértices de grafo tendo o menor número possível de arestas. Um grafo é k-conectado caso não fique desconexo removendo-se menos que k vértices.



Instâncias

17 Vértices	30 Vértices	50 Vértices	100 Vértices
$H_{2,17}$	$H_{2,30}$	$H_{2,50}$	$H_{2,100}$
$H_{3,17}$	$H_{3,30}$	$H_{3,50}$	
$H_{5,17}$	$H_{8,30}$	$H_{11,50}$	
$H_{6,17}$	$H_{9,30}$	$H_{20,50}$	
$H_{7,17}$	$H_{10,30}$	$H_{21,50}$	

- As mesmas utilizadas por Alfredo e de Souza

Instâncias	TreeBlock	ILP	BRKGA		Scatter Search		Busca Tabu	
	T	T	T	TE (s)	T	TE (s)	T	TE (s)
H2,17	9	9	9	0,2	9	0,006	9	2
H3,17	5	5	5	0,2	5	0,002	5	2
H5,17	5	5	5	0,2	5	0,002	5	2
H6,17	5	5	5	0,2	5	0,002	5	2
H7,17	5	5	5	0,3	5	0,002	5	2
H2,30	15	15	15	0,4	15	0,007	15	2
H3,30	9	9	9	0,4	9	0,006	9	2
H8,30	6	5	5	1,15	6	0,005	5	2
H9,30	6	5	5	0,6	5	0,004	5	2
H10,30	6	5	5	0,6	5	0,007	5	2
H2,50	25	25	25	0,11	25	0,021	25	2
H3,50	14	14	14	0,9	14	0,016	14	2
H11,50	7	-	6	0,1	6	0,009	6	2
H20,50	8	-	6	0,14	6	0,001	6	2
H21,50	7	-	6	0,14	6	0,010	6	2
H2,100	50	50	50	0,38	50	0,10	50	3,56

Conclusão

Ficamos extremamentes felizes com os resultados alcançados, mas temos alguns pontos de melhoria, até o fim do deadline do trabalho.

Pontos de Melhoria:

- Iniciar população com uma solução gulosa
- Usar uma solução inicial gulosa na busca tabu
- Pensar em mais formas de reprodução
- Criar busca local eficiente
- Testar novas instâncias para garantir eficácia do método

Fim!

Referências
- Heuristics for the Minimum Broadcast Time, Amaro de Sousa
- Algoritmo Genético de Chaves Aleatórias Viciadas para o Problema do Tempo de Transmissão Mínimo, Alfredo Lima Moura Silva, Rian Gabriel Santos Pinheiro, Bruno Costa e Silva Nogueira, e Rodrigo Jose Sarmento Peixoto