

Busca Local Iterada

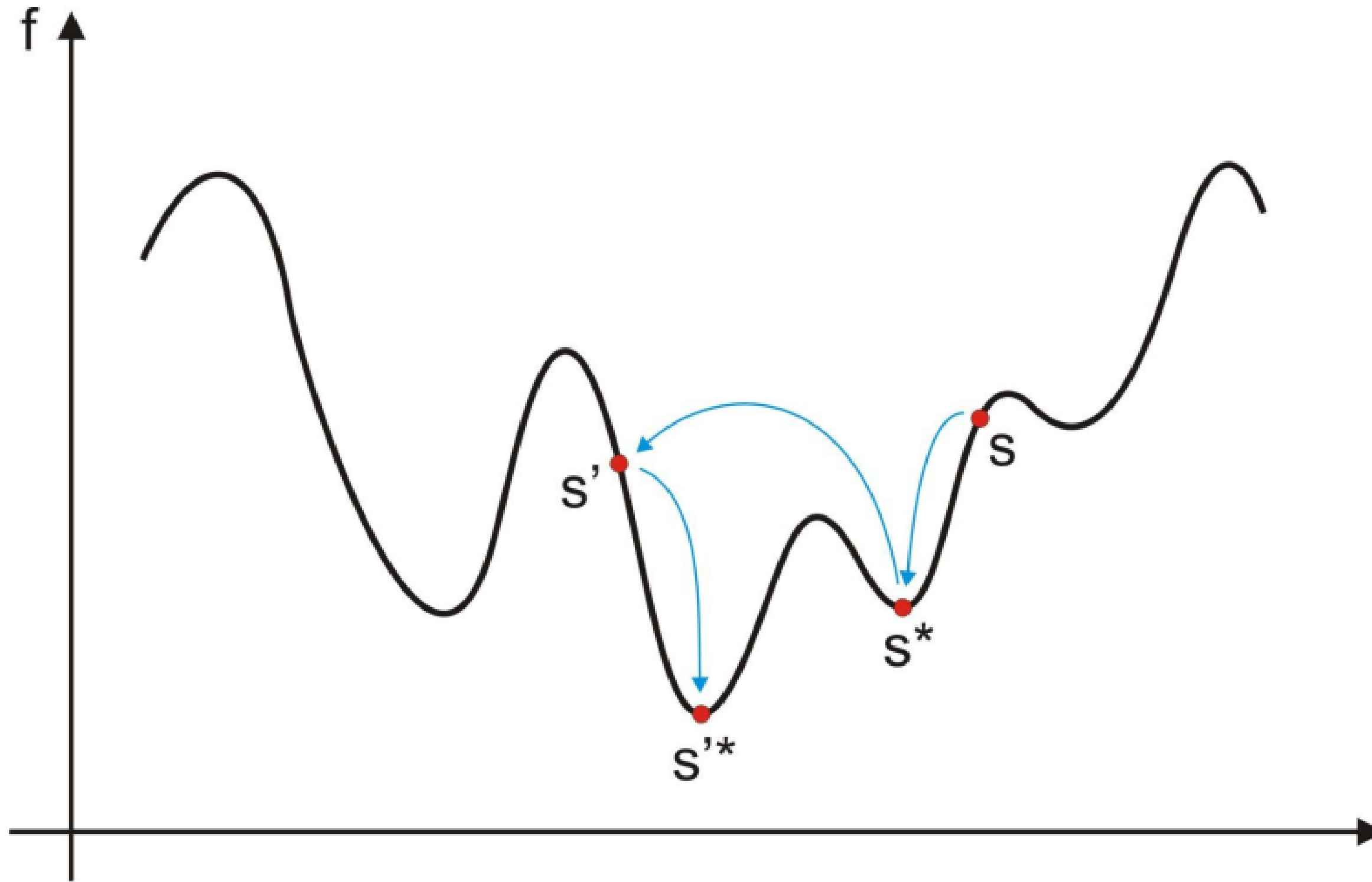
Felipe Soares, Kevin Perdomo, Lukelly Silva, Mateus Ramos



Busca Local Iterada

- **Técnica de otimização e busca heurística usada para resolver problemas de otimização combinatória**
- **Combinação de busca Local com perturbação iterada;**
- **Busca Local para encontrar o menor resultado da vizinhança;**
- **Perturbação iterada para evitar vícios na busca local;**

Perturbação iterada modificando os resultados



Perturbação iterada modificando os resultados

```
1   $s_0 \leftarrow \text{Gera\_Solução\_Inicial};$   
2   $s^* \leftarrow \text{Busca\_Local}(s_0);$   
3  Enquanto não é satisfeito um critério de parada,  
3.1     $s' \leftarrow \text{Perturbação}(s^*, \text{histórico});$   
3.2     $s'^* \leftarrow \text{Busca\_Local}(s');$   
3.3     $s^* \leftarrow \text{Critério\_de\_Aceitação}(s^*, s'^*, \text{histórico});$   
4  Fim-Enquanto;
```

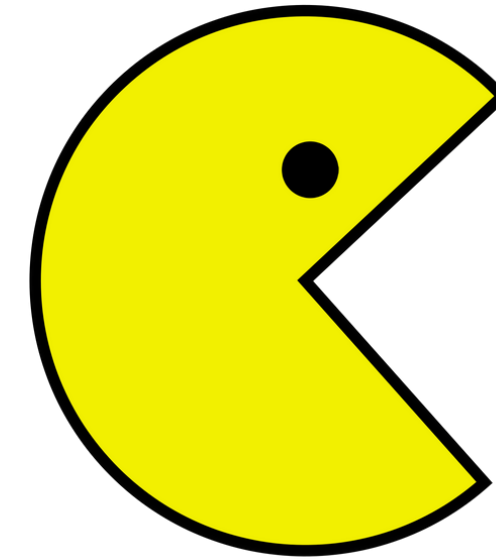
Componentes do ILS

- **Solução inicial** - Produz uma solução inicial, podendo ser aleatória ou fruto de uma heurística menor;
- **Busca Local** - Retorna uma solução melhorada em relação a inicial com base na sua vizinhança;
- **Perturbação** - Modifica a solução corrente guiando a uma solução intermediária, evitando que o algoritmo prenda em ótimos sub-ótimos locais;
- **Critério de Aceitação** - Decide de qual solução a próxima perturbação será aplicada;



Aleatório

usado quando o
número de iterações é
grande.



Algoritmo guloso

usado quando o
número de iterações é
pequeno ou para ter
mais velocidade.

Solução inicial

Busca Local

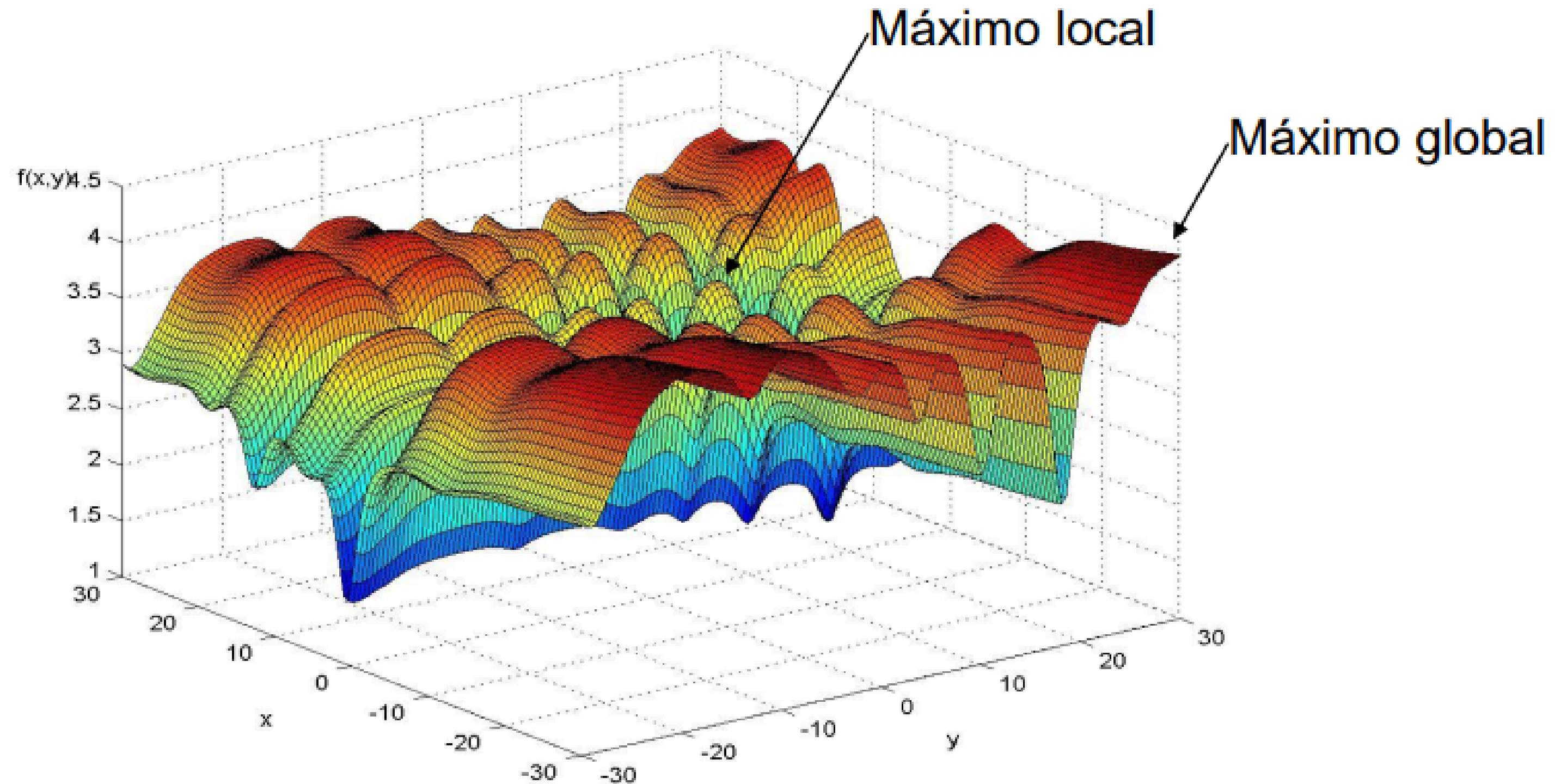
- **É o centro da ILS, porém não se prende a único algoritmo;**
- **Pode se escolher entre algoritmos de buscas para serem implementados;**
- **A única exigência é que esse algoritmo possa melhorar a solução dada;**
- **Quanto mais eficiente o algoritmo melhores os resultados, porém vale analisar o contexto;**

- **É o mecanismo responsável por não permitir que o algoritmo de busca fique preso a um resultado só;**
- **Deve ser forte o suficiente para evitar que a busca local volte ao mesmo resultado anterior;**
- **Porém fraco o bastante para que a busca não tenha q recomeçar a cada perturbação;**
- **Para encontrar o melhor resultado da melhor forma possível é necessário o equilíbrio perfeito desse passo;**

Perturbação

Perturbação

- Espaço de busca



Critério de aceitação

s' = solução atual

$f(s')$ = avaliação da melhor solução

s = solução atual

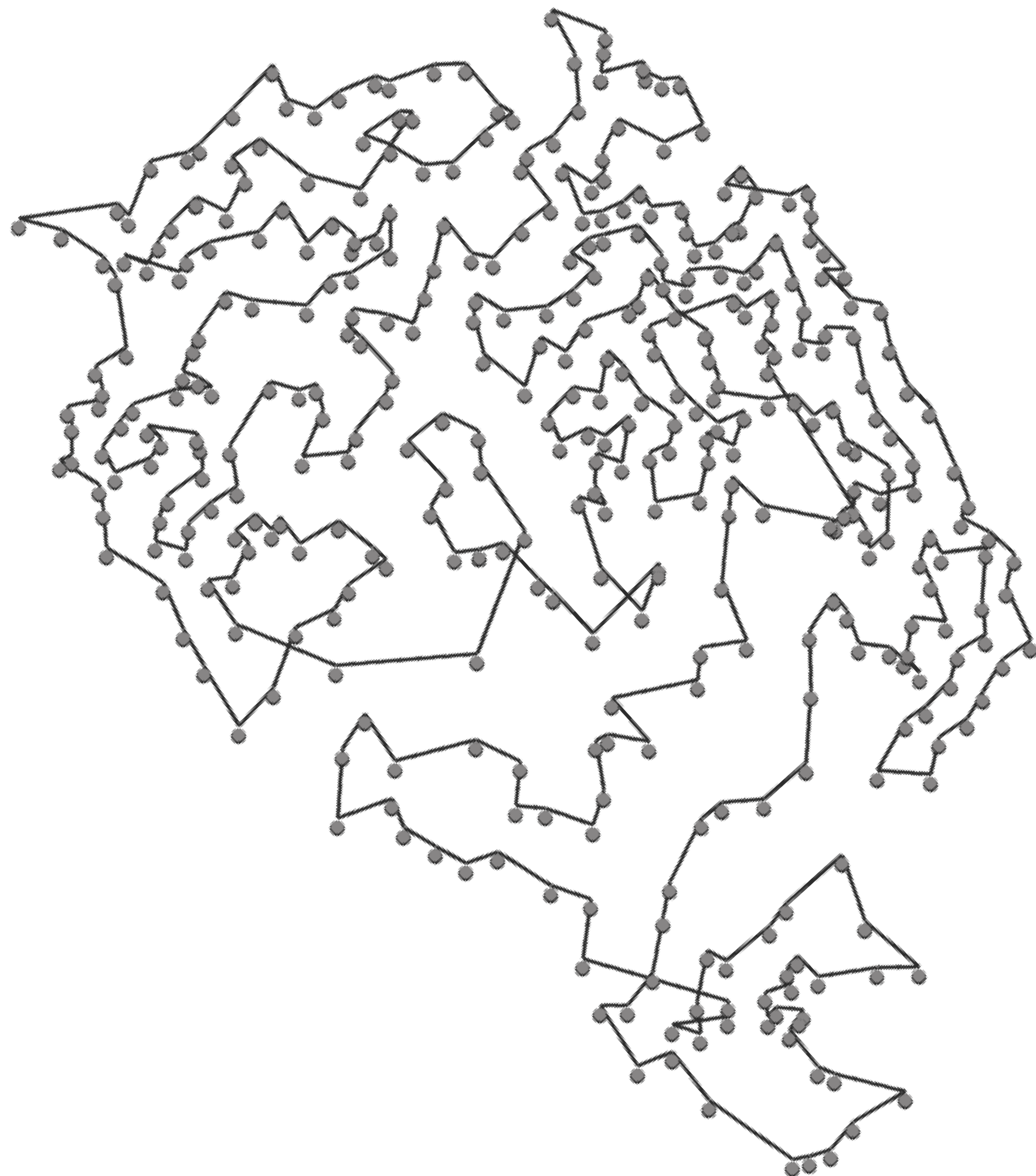
$f(s)$ = avaliação da solução atual

$\text{se } f(s') < f(s)$

- **É o critério usado pela busca para saber se ela vai atualizar a solução atual ou não após uma perturbação;**
- **Geralmente é feito pela comparação:**

$$\text{se } f(s') < f(s)$$

- **Porém, pode-se adicionar uma probabilidade de que um $f(s')$ seja escolhido, mesmo sendo pior.**
- **Isso acontece para que o algortimo possa varrer uma área maior de dados, trazendo uma variedade melhor de resultados para análise.**

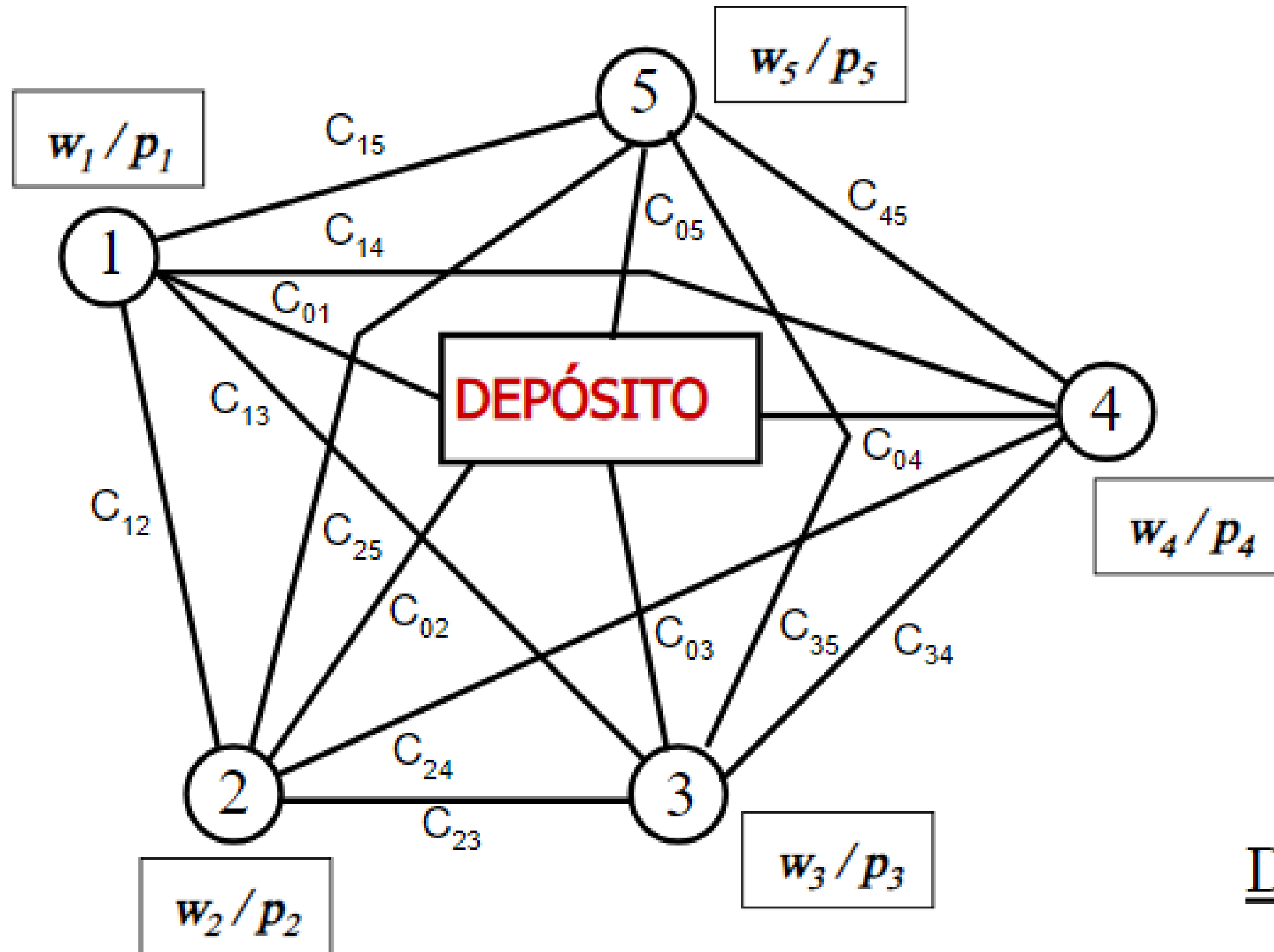


Problema do Caxeiro Viajante com Coleta de Prêmios

PCVCP

- **É uma extensão do PCV, adicionando um sistema de coleta de prêmios por cidade, visitar uma cidade lhe dá um prêmio, não visitar lhe faz pagar uma penalidade;**
- **Esse sistema de coleta de prêmios adiciona uma nova dimensão ao problema;**
- **NP-Difícil, pois quanto mais cidades adicionadas o número de rotas aumenta exponencialmente;**
- **Pode ser aplicado em diversas áreas, com grande potencial de resolução de problemas;**

PCVCP



Depósito:

$$w = 0$$

$$p = \infty$$

Função Objetivo

- **Cenário:**

Um caixeiro viajante precisa visitar cada cidade para coletar o prêmio associado, mas caso não visite a cidade, ele deve pagar uma penalidade;

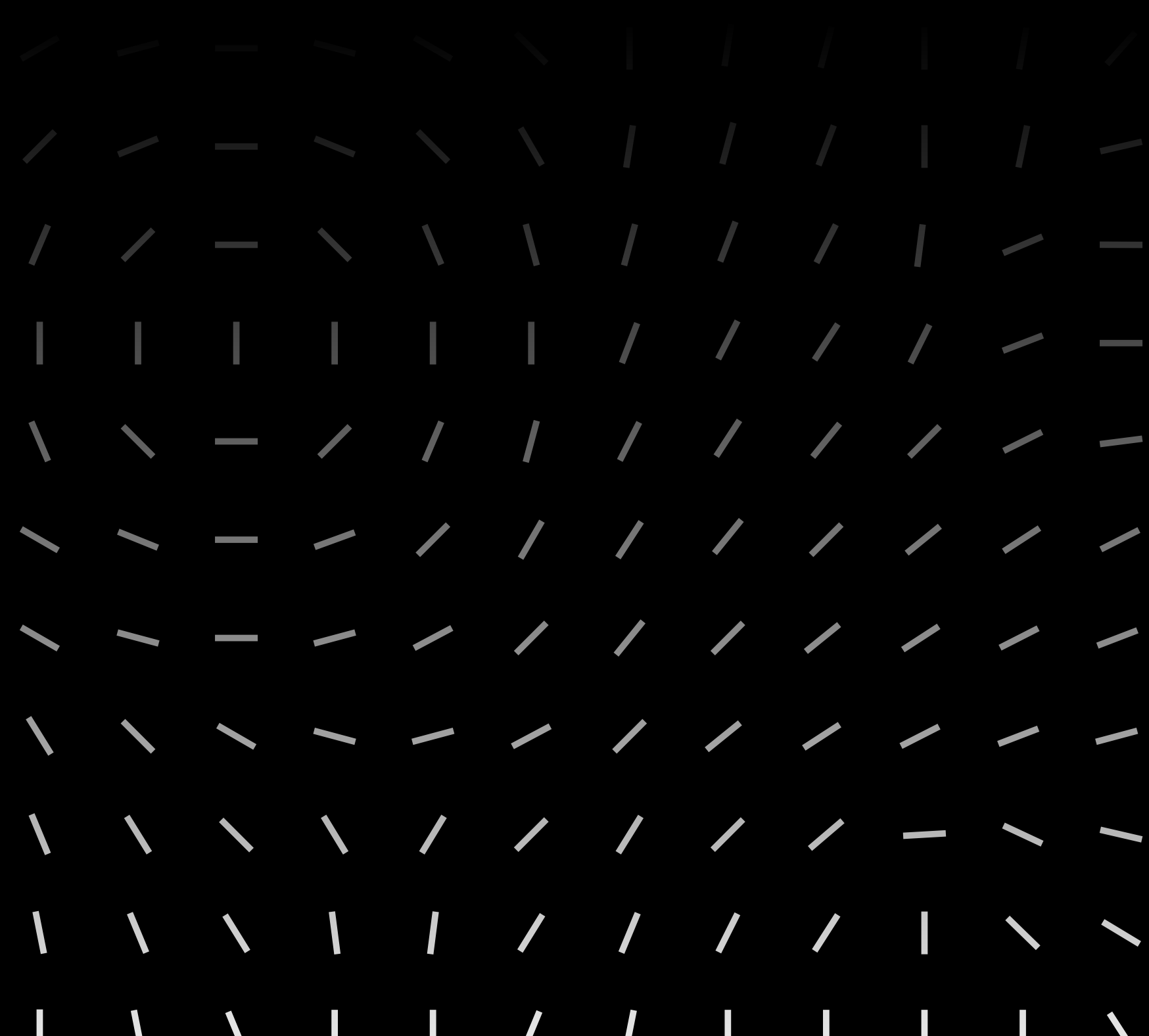
Objetivos:

- minimizar o custo de deslocamento
- minimizar a soma das penalidades
- coletar um prêmio mínimo pré-estabelecido;

Solução

- O algoritmo ILS interpreta o conjunto de cidades como um grafo;
- Cada cidade é um nó com um prêmio e cada aresta representa o custo de viagem;
- Com esse conceito se forma o espaço de busca.
- Inicia-se em um ponto aleatório do conjunto e, com ajuda da busca local, ele 'vasculha' as proximidades dessa ponto, buscando e salvando melhores resultados.
- Caso ele fique preso em um unico conjunto ele é redirecionado a outro ponto pela perturbação.

Vizinhança Utilizada



- **Definindo a Vizinhança**

- **Formas de definir**

- Trocar de cidades adjacentes
- Inversão de subconjuntos de cidades
- Troca de cidades não adjacentes
- Heurística 2-opt

- **Importância**

- Impacto na busca local iterada
- Qualidade e desempenho das soluções

Resultados Obtidos

- **Geração dos Problemas-Teste**
- **Realização de Testes**
- **Resultados dos Testes**
- **Abordagem alteranativa e calculo de Desvio**
- **Comparando Resultados**
- **Conclusões**

Resultados Obtidos

Tabela 1: Resultados dos experimentos computacionais.

		MÉTODO EXATO		GRASP+VNS		
PROBLEMA-TESTE	V	ÓTIMO GLOBAL	TEMPO (S)	FOMelhor	TEMPO (S)	DESVIO (%)
v10	11	1765	1	1765	0,10	0,00
v20	21	2302	65	2302	1,04	0,00
v30a	31	3582	86	3582	5,43	0,00
v30b	31	2515	100	2515	3,83	0,00
v30c	31	3236	1786	3236	7,83	0,05
v50a	51	-	10800	4328	132,45	0,42
v50b	51	-	10800	3872	43,76	0,31
v100a	101	-	-	6892	692,09	0,52
v100b	101	-	-	6814	446,81	0,12
v250a	251	-	-	15310	918,33	0,88
v250b	251	-	-	14678	996,72	0,76
v500a	501	-	-	28563	2145,79	0,67
v500b	501	-	-	28524	2410,21	0,82

Obrigade

e até semestre que vem 🥰