DA – Projeto 2 G12

Agência de viagens

Realizado por:

João Pereira
Jorge Sousa
Nuno Pereira

Descrição do problema

- Uma agência de viagens dispõe de veículos em vários locais. Cada uma fará um trajeto de uma origem para um destino, com uma duração, capacidade e custo por pessoa.
- O objetivo é desenvolver um sistema capaz de gerir pedidos de transporte de grupos de pessoas de uma origem a um destino.
- O problema é subdividido em 2 cenários:
  - Os grupos não se podem dividir
  - Os grupos podem dividir-se

Estruturas de dados comuns aos problemas

- **Grafo** – implementamos os grafos com recurso a um unordered map, em que as chaves representam o número dos nós e o valor é o nó em si. Cada nó contém um unordered set de arestas que contém as arestas que o ligam aos nós adjacentes.
- Decidimos recorrer a unordered sets e unordered maps, pois, nos casos em que são usados, não precisamos de ordenação e assim conseguimos inserir, remover e aceder a elementos em tempo constante (amortizado).
- **Heap de máximos** – implementada com recurso a sets.

Cenário 1 – Grupos que não se separam

- Problemas:
  1. Maximizar a dimensão do grupo e indicar um encaminhamento.
  2. Maximizar a dimensão do grupo e minimizar o número de transbordos, sem privilegiar nenhum dos critérios.

Cenário 1.1 – Formalização

- Dados e variáveis
  - Grafo F, F = (V, E).
- Objetivo:
  - Maximizar n, em que n é o número de elementos do grupo, $n \in \mathbb{N}^{+}$.
- Restrições:
  - O grupo não se pode separar.
- Resultado:
  - Inteiro que representa o número máximo de elementos de um grupo para viajar do nó origem ao nó destino.
  - Vetor com lista de inteiros, em que cada um representa um nó pertencente ao encaminhamento.

Cenário 1.1 – Algoritmos relevantes

- Utilizamos uma adaptação do algoritmo de Dijkstra para descobrir o caminho de capacidade máxima, que é o que nos permite transportar o maior número de pessoas, sem o mesmo se separar.

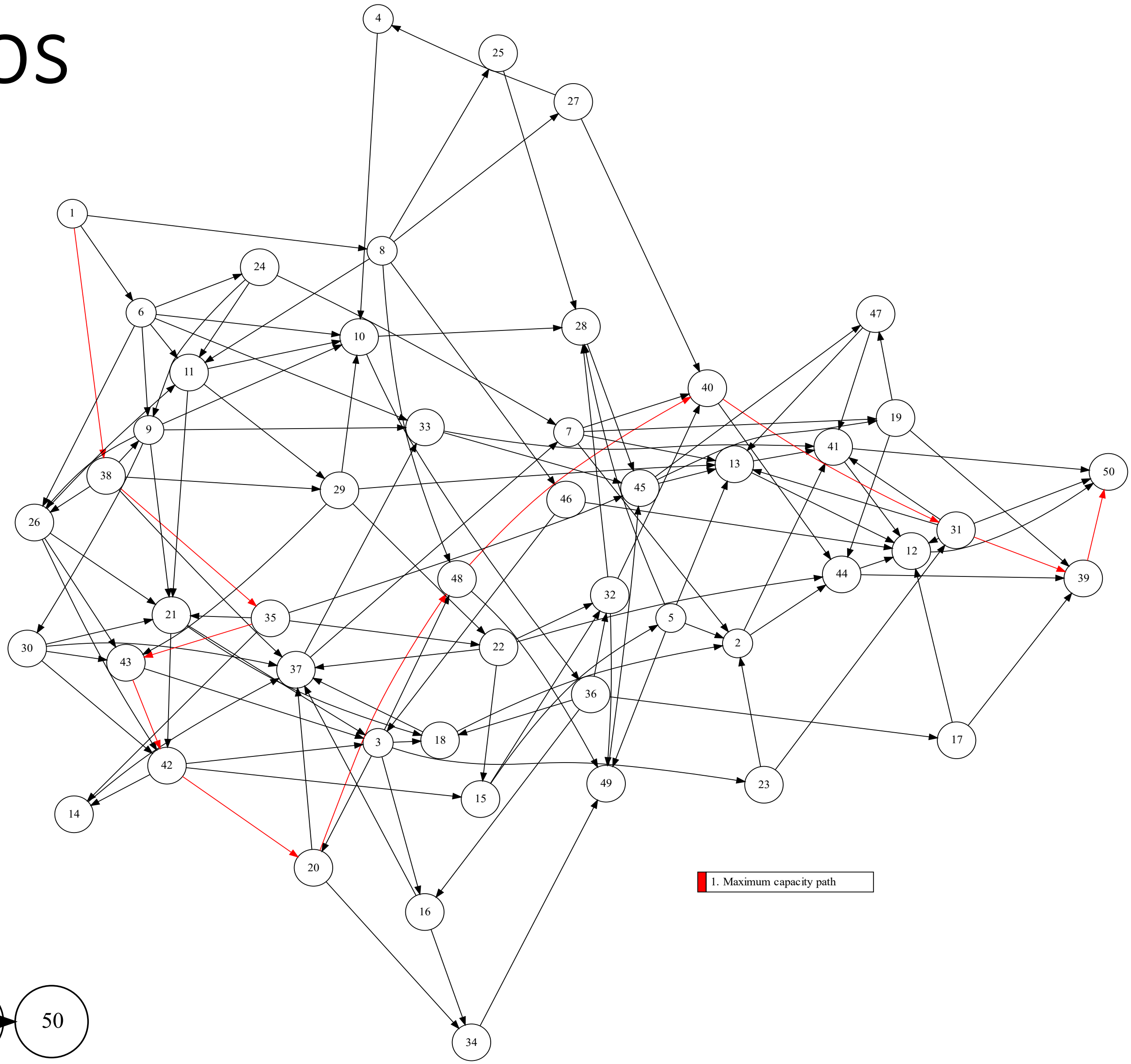
Cenário 1.1 – Análise da complexidade

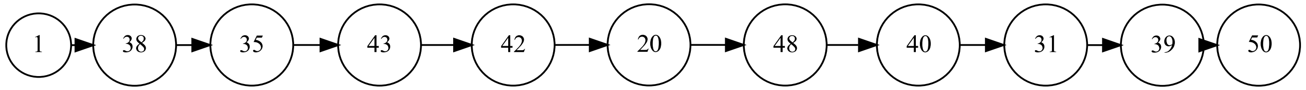
- Complexidade temporal
  - Como estamos a utilizar uma fila de prioridade com recurso a uma heap de máximos, conseguimos uma complexidade de O(|V| + |E| $\log_2 |V|$), em que V representa o número de nós e E o número de arestas.

- Complexidade espacial
  - Estamos apenas a usar um vetor para guardar as capacidades em cada nó e um set para a implementação da heap de máximos, logo temos complexidade igual a O(V), em que V representa o número de nós.

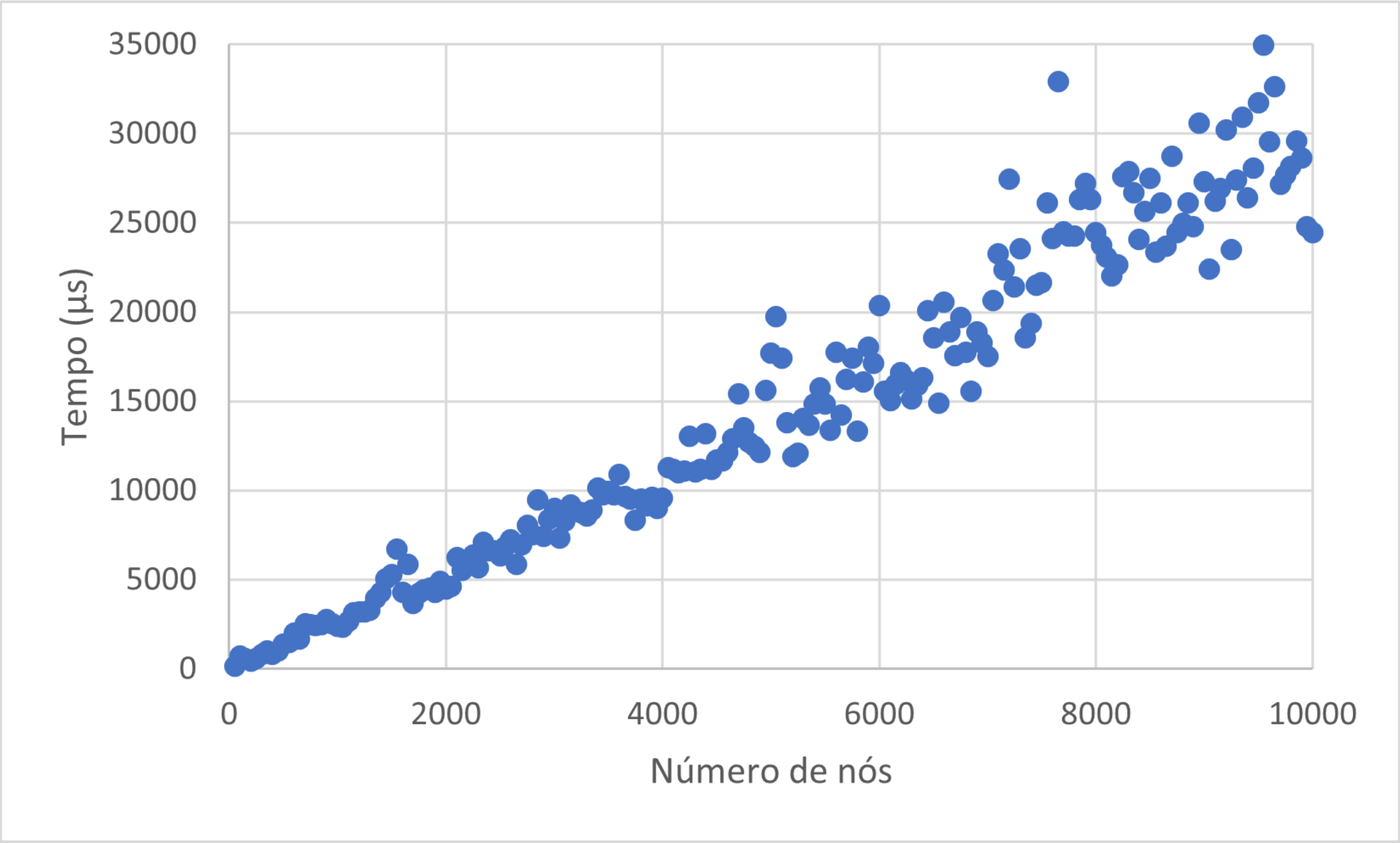
Cenário 1.1 – Resultados

Grafo com o caminho de capacidade máxima no dataset 1 e nós que formam o caminho.





Cenário 1.1 – Resultados



Cenário 1.2 – Formalização

- Dados e variáveis:
  - Grafo F, F = (V, E).
- Objetivo:
  - Maximizar n, em que n é o número de elementos do grupo, $n \in \mathbb{N}^{+}$.
  - Minimizar m, em que m representa o número de transbordos, m $\in \mathbb{N}^{+}$.
- Restrições:
  - O grupo não se pode separar.
  - Não privilegiar um critério em relação ao outro.
- Resultados:
  - Inteiro que representa o número máximo de elementos de um grupo para viajar do nó origem ao nó destino.
  - Vetor com lista de inteiros, em que cada um representa um nó pertencente ao encaminhamento.

Cenário 1.2 – Algoritmos relevantes

- Ao utilizarmos Breadth First Search, conseguimos encontrar o caminho com menos transbordos.

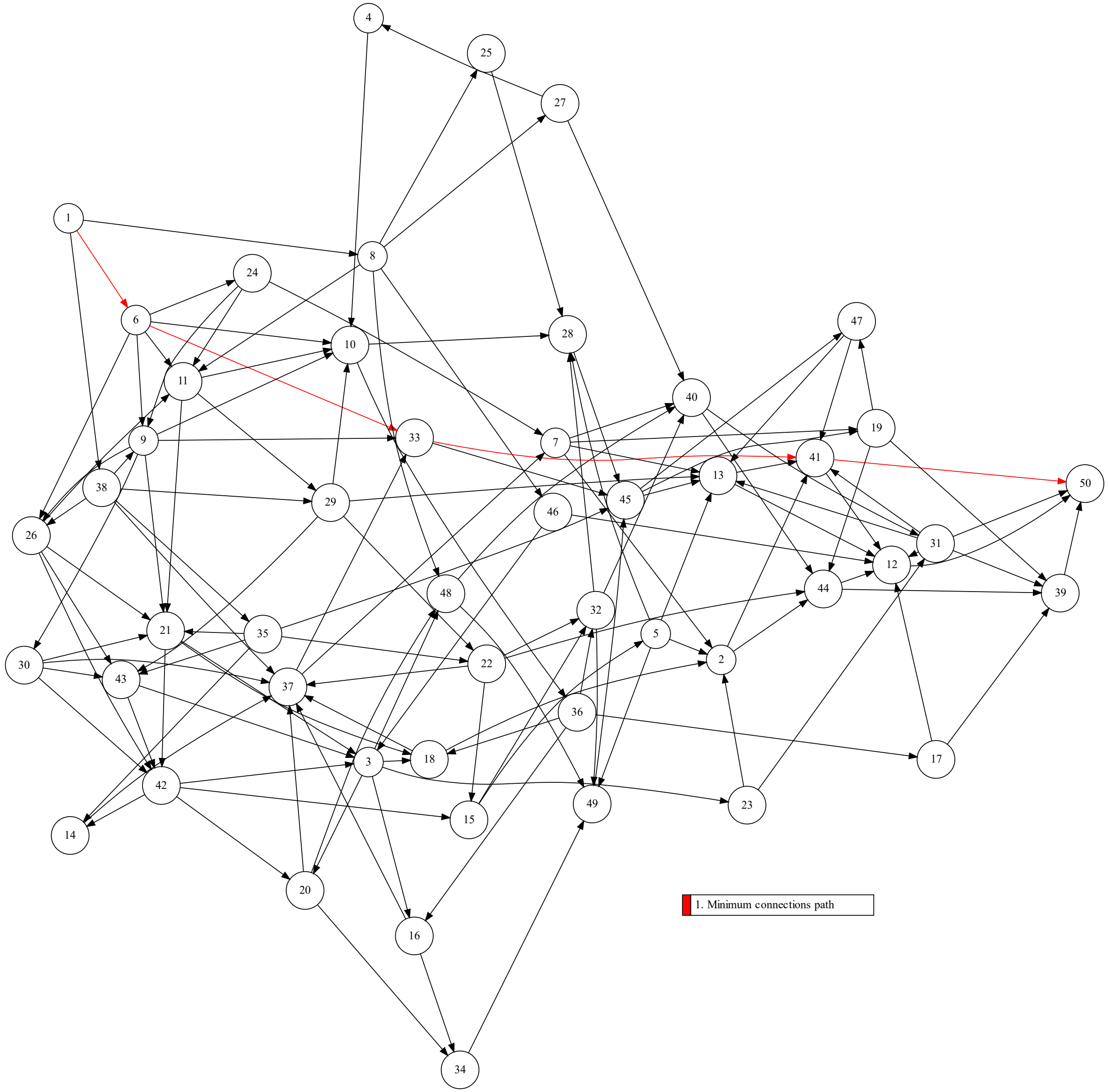
Cenário 1.2 – Análise da complexidade

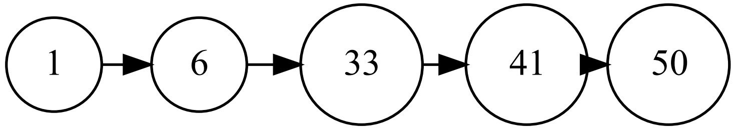
- Complexidade temporal
  - Como estamos apenas a usar Breadth First Search, a complexidade temporal é $O(V + E)$, em que V representa o número de nós e E o número de arestas.

- Complexidade espacial
  - É igual à complexidade espacial de Breadth First Search, logo $O(V)$, em que V representa o número de nós.

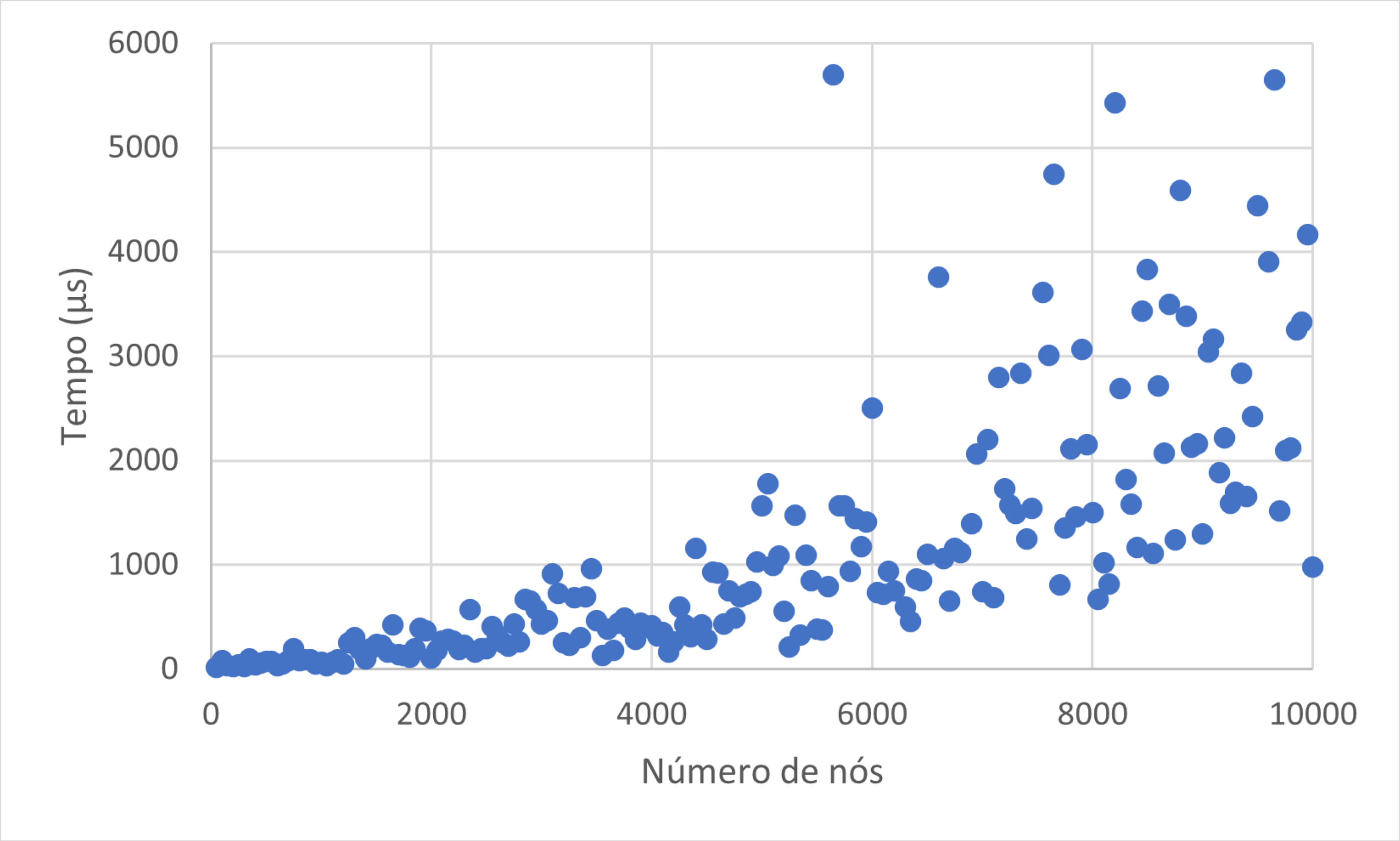
Cenário 1.2 – Resultados

Grafo com o caminho com menos transbordos no dataset 1 e nós que formam o caminho.

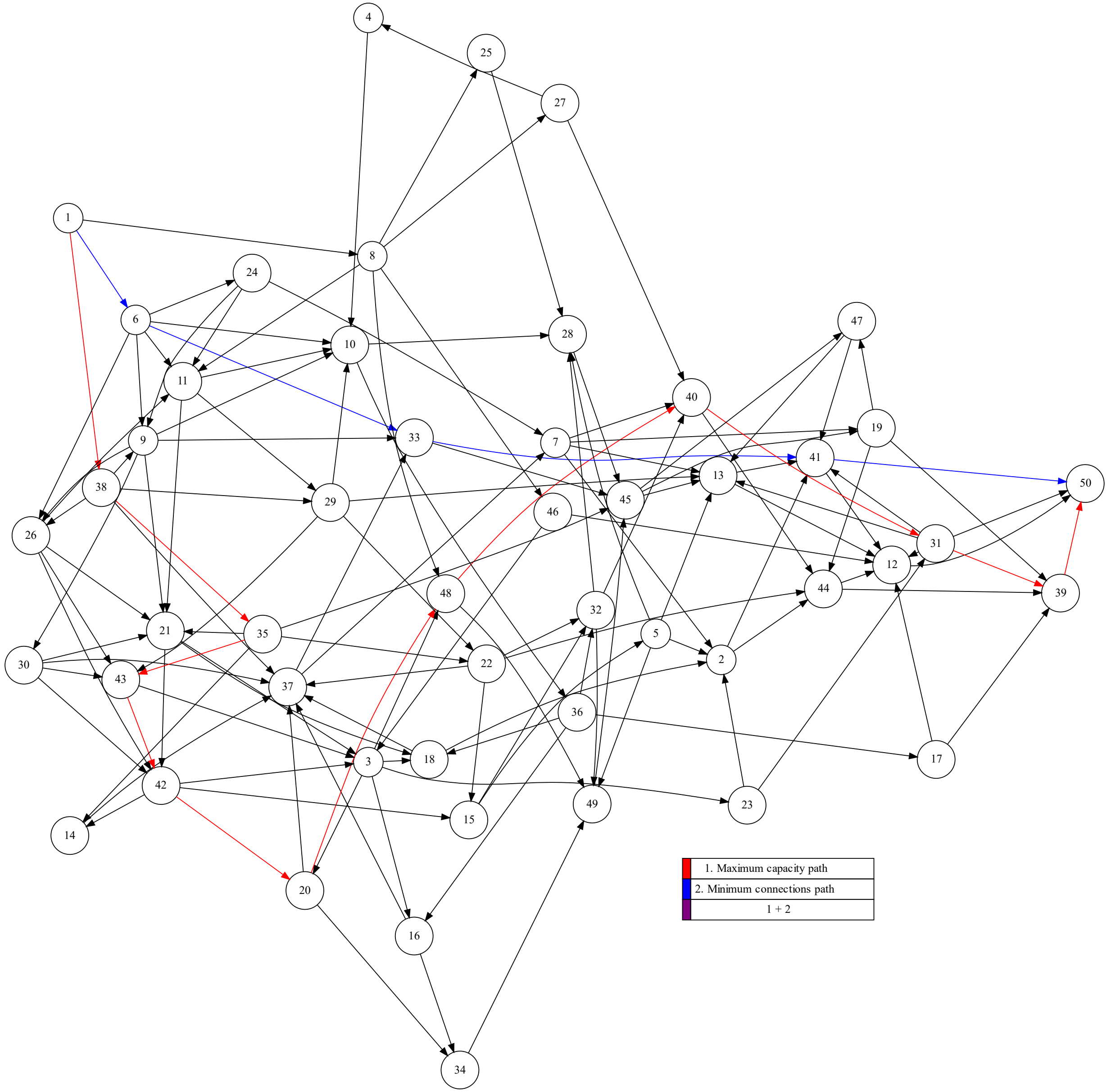




Cenário 1.2 – Resultados



Comparação dos resultados das alíneas anteriores



Cenário 2 – Grupos que se separam

- Problemas:
  1. Determinar um encaminhamento para um grupo, dada a sua dimensão.
  2. Corrigir um encaminhamento, se necessário, para que a dimensão de um grupo possa aumentar de um número de unidades dado.
  3. Determinar a dimensão máxima de um grupo e um encaminhamento.
  4. Determinar quando é que o grupo se reúne de novo no destino, partindo de um encaminhamento.
  5. Determinar o tempo máximo de espera e os locais onde haveria elementos que esperam esse tempo, partindo de um encaminhamento.

Cenário 2.1 – Formalização

- Dados e variáveis:
  - Grafo F, F = (V, E).
  - Dimensão do grupo, n.
- Objetivo:
  - Encontrar um grafo G, sendo G = (V, E), com $G.V \subseteq F.V \land G.E \subseteq F.E$.
- Restrições:
  - $n \leq G.fMáx$.
- Resultados:
  - Inteiro que representa o fluxo máximo.
  - Vetor de listas de inteiros, em que cada lista representa um caminho utilizado por uma divisão do grupo.

Cenário 2.1 – Algoritmos relevantes

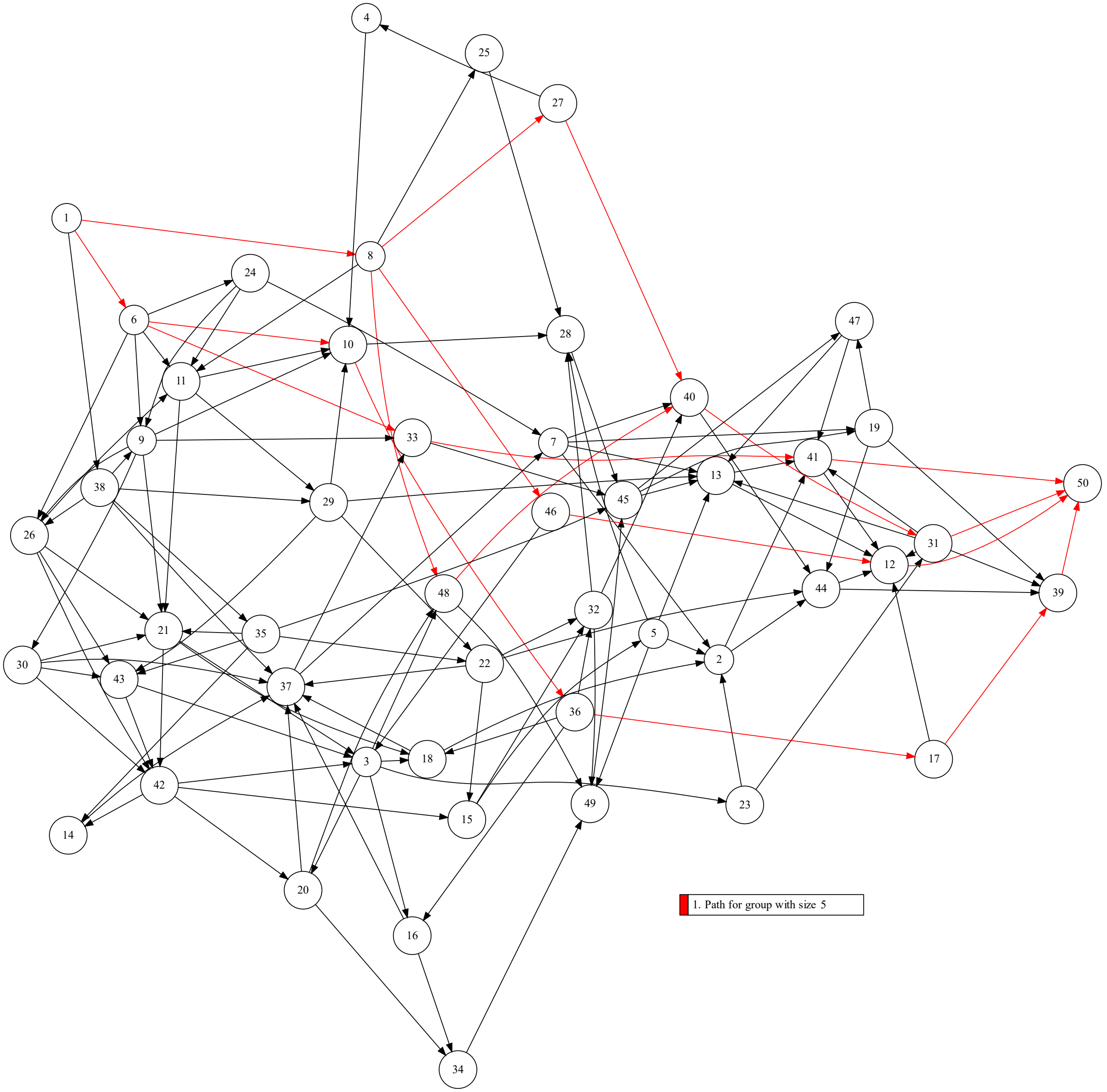
- Utilizamos o algoritmo de Edmonds-Karp, mas o mesmo termina quando o fluxo é maior ou igual à dimensão do grupo dada.

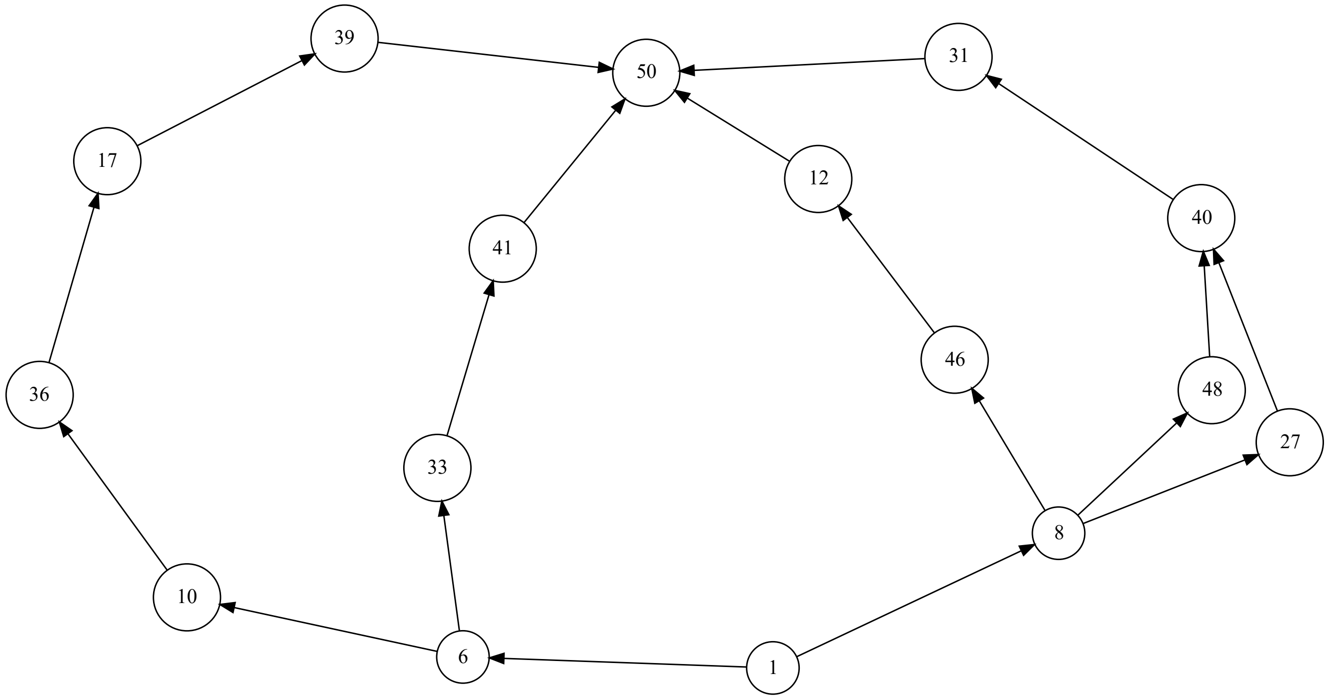
Cenário 2.1 – Análise da complexidade

- Complexidade temporal
  - Semelhante à complexidade temporal do algoritmo de Edmonds-Karp, ou seja, $O(|V||E|^2)$, em que V representa o número de nós e E o número de arestas.

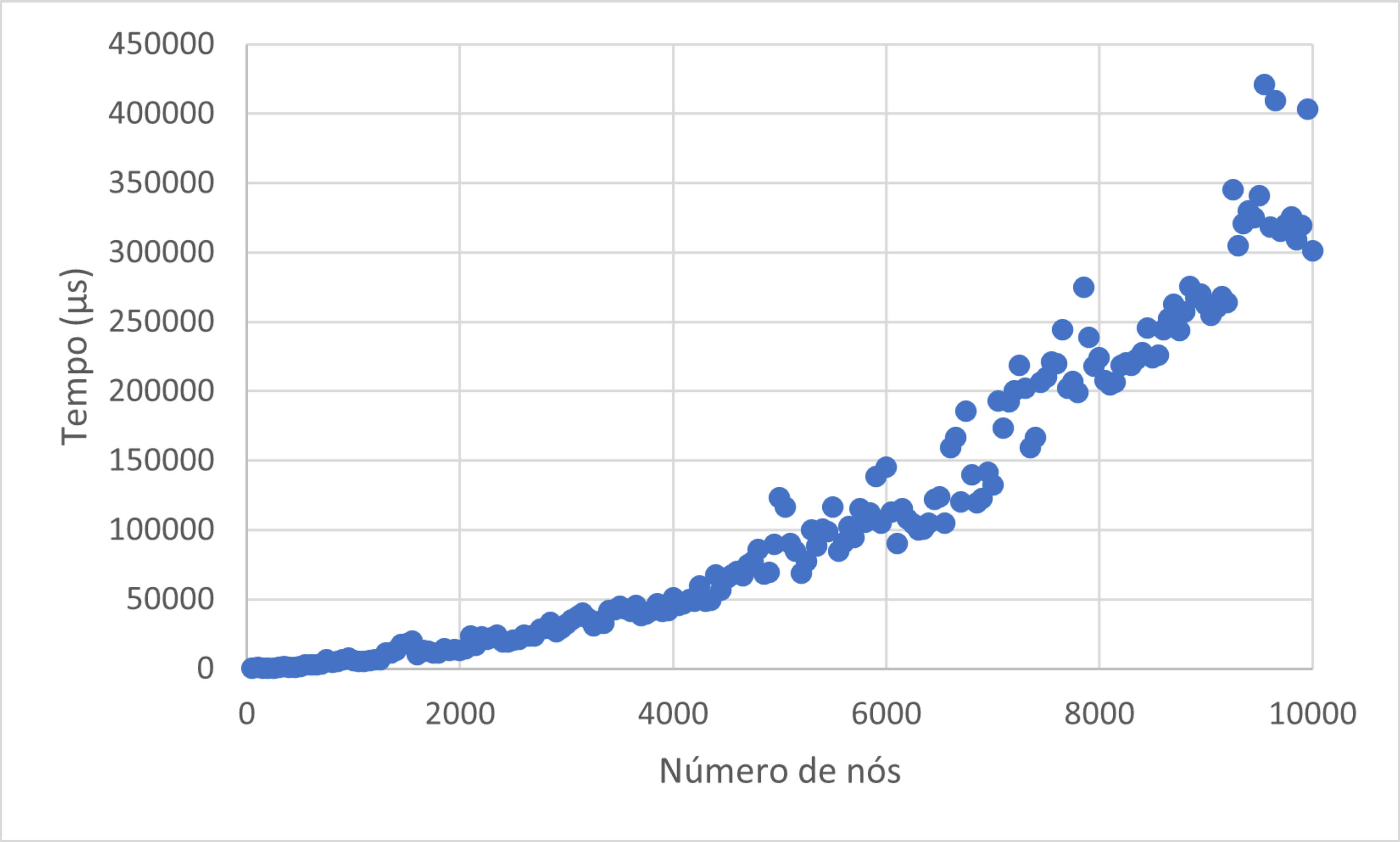
- Complexidade espacial
  - Utilizamos apenas um grafo que representa os caminhos tomados pelo grupo. Assim, temos complexidade espacial O(V), em que V representa o número de nós pertencentes aos caminhos usados pelo grupo.

Cenário 2.1 – Resultados





Cenário 2.1 – Resultados



Cenário 2.2 – Formalização

- Dados e variáveis:
  - Grafo F, F = (V, E).
  - Dimensão do grupo, n.
  - Aumento da dimensão do grupo, m.
  - Fluxo máximo de F, fMáx.
- Objetivo:
  - Encontrar um grafo G, sendo G = (V, E), com $G.V \subseteq F.V \land G.E \subseteq F.E$.
- Restrições:
  - $n + m \leq fMáx$.
- Resultados:
  - Inteiro que representa o fluxo máximo.
  - Vetor de listas de inteiros, em que cada lista representa um caminho utilizado por uma divisão do grupo.

Cenário 2.2 – Algoritmos relevantes

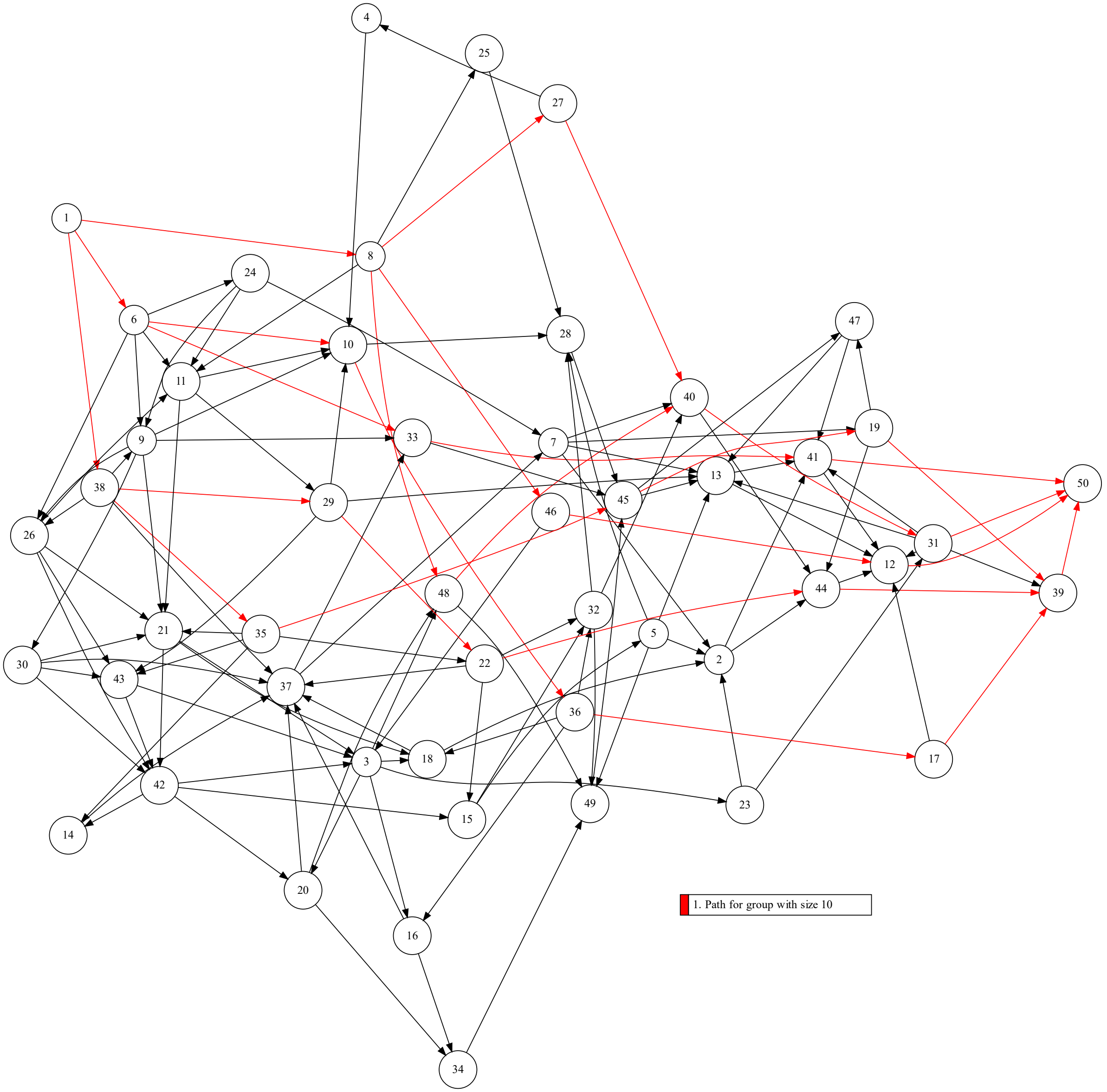
- Utilizamos o mesmo procedimento do cenário anterior, apenas alteramos o tamanho do grupo de modo a corresponder ao incremento dado.
- Assim, usamos o algoritmo de Edmonds-Karp até o fluxo ser maior ou igual que a nova dimensão do grupo.

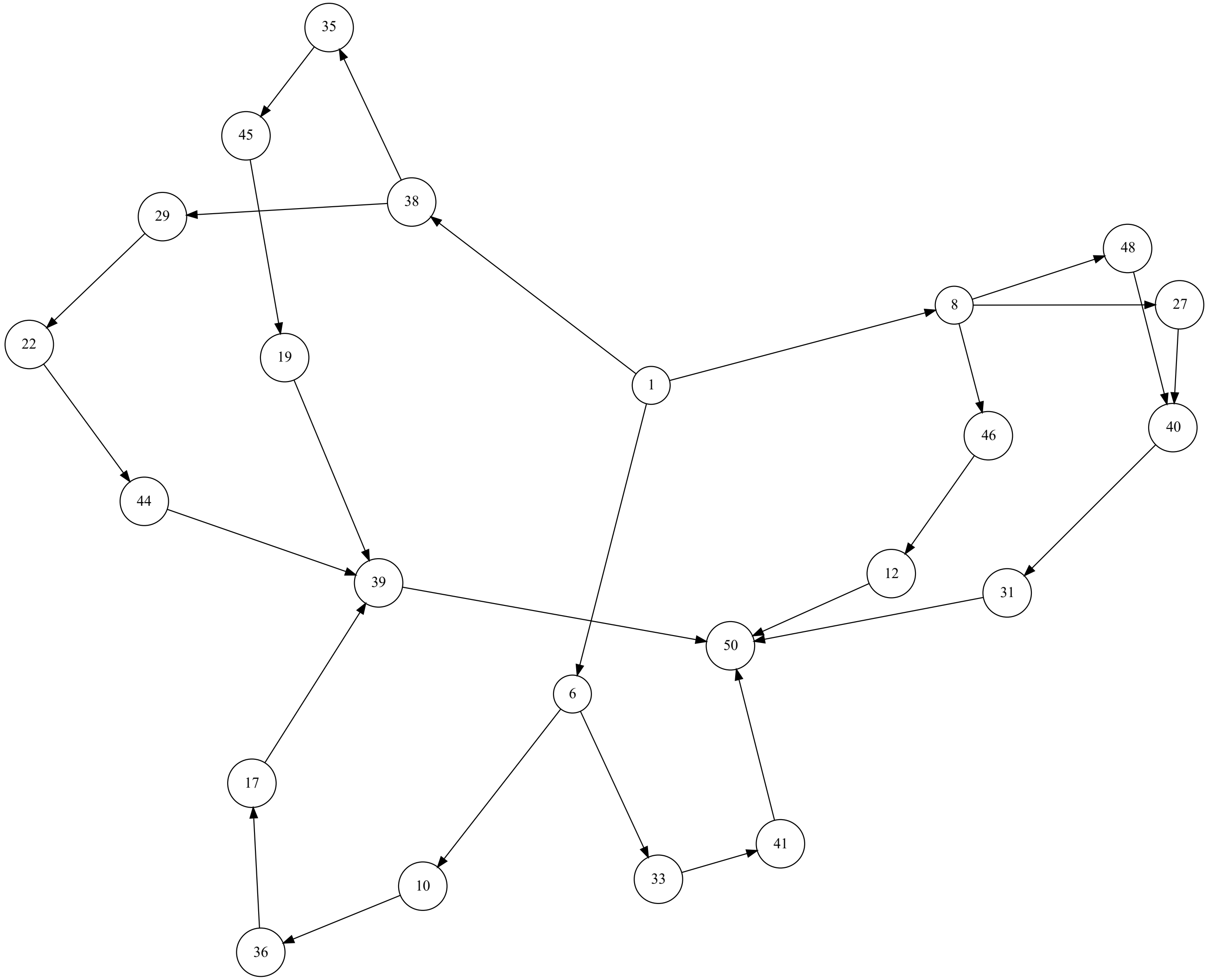
Cenário 2.2 – Análise da complexidade

- Complexidade temporal
  - Semelhante à complexidade temporal do algoritmo de Edmonds-Karp, ou seja, $O(|V||E|^2)$, em que V representa o número de nós e E o número de arestas.

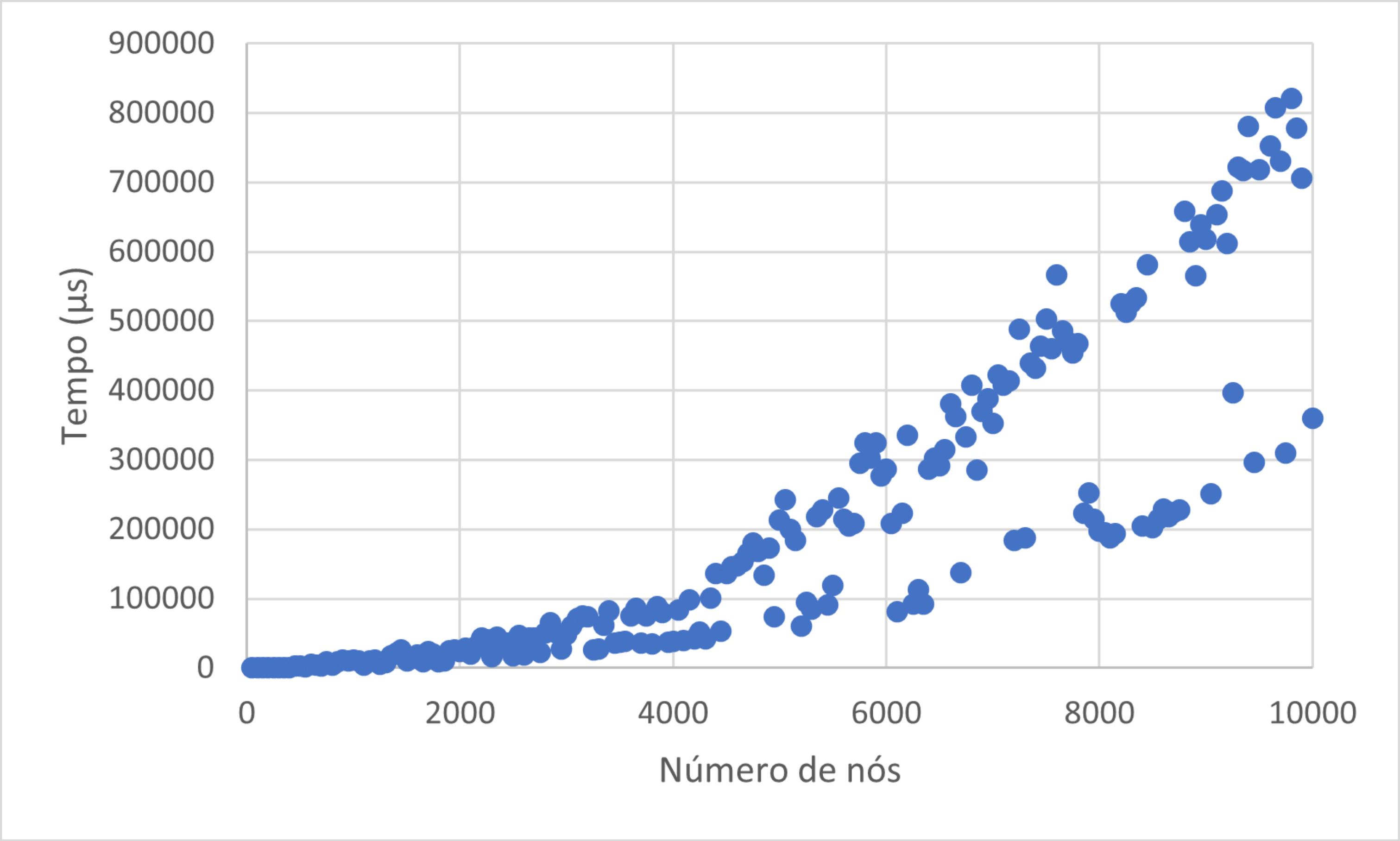
- Complexidade espacial
  - Utilizamos apenas um grafo que representa os caminhos tomados pelo grupo. Assim, temos complexidade espacial O(V), em que V representa o número de nós pertencentes aos caminhos usados pelo grupo.

Cenário 2.2 – Resultados





Cenário 2.2 – Resultados



Cenário 2.3 – Formalização

- Dados e variáveis:
  - Grafo F, F = (V, E).
- Objetivo:
  - Encontrar um grafo G, sendo G = (V, E), com $G.V \subseteq F.V \land G.E \subseteq F.E$.
  - Encontrar a dimensão máxima de um grupo.
- Resultados:
  - Inteiro que representa o fluxo máximo.
  - Vetor de listas de inteiros, em que cada lista representa um caminho que pode ser utilizado por uma divisão do grupo.

Cenário 2.3 – Algoritmos relevantes

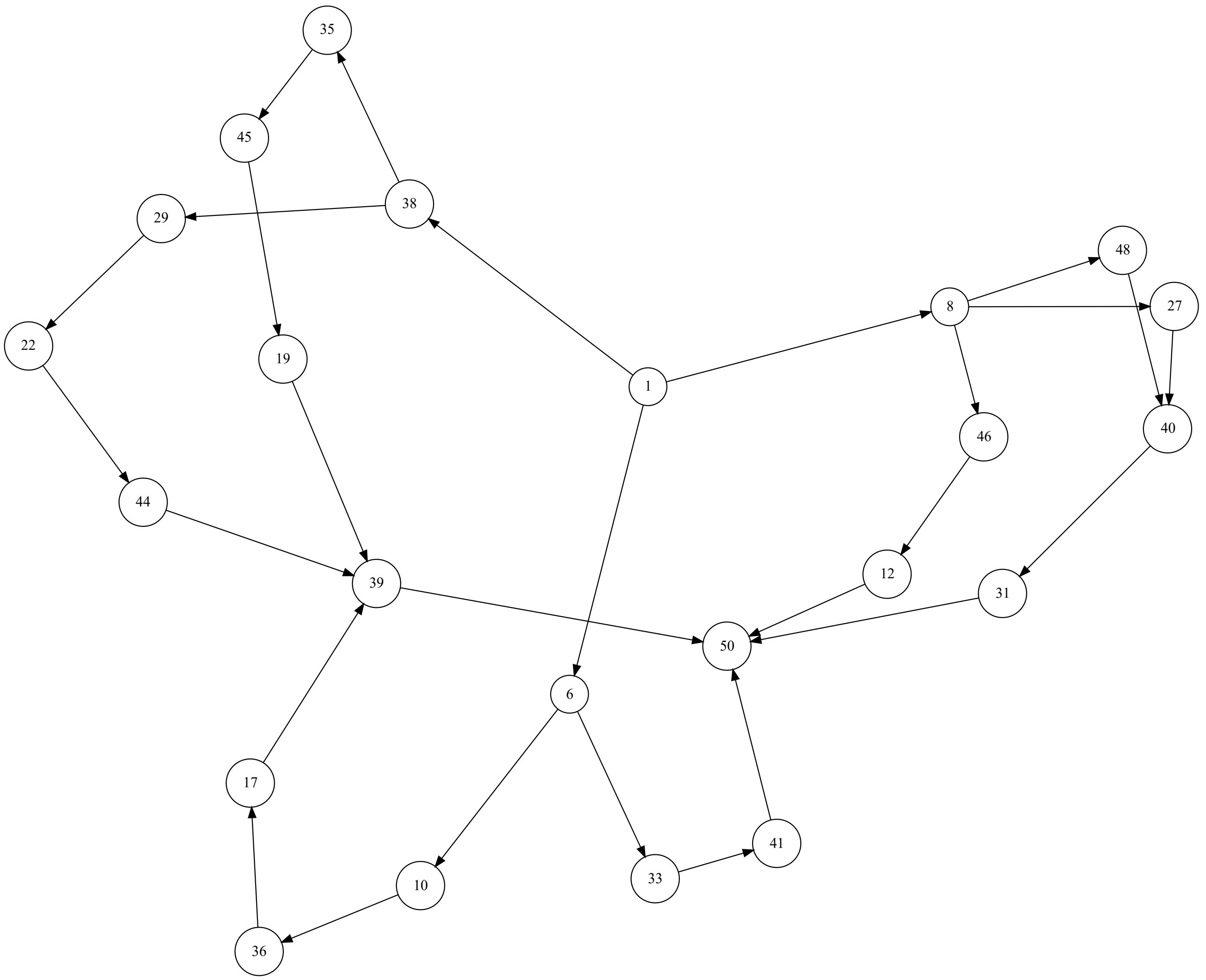
- Utilizamos o algoritmo de Edmonds-Karp, que nos dá diretamente o fluxo máximo num grafo.

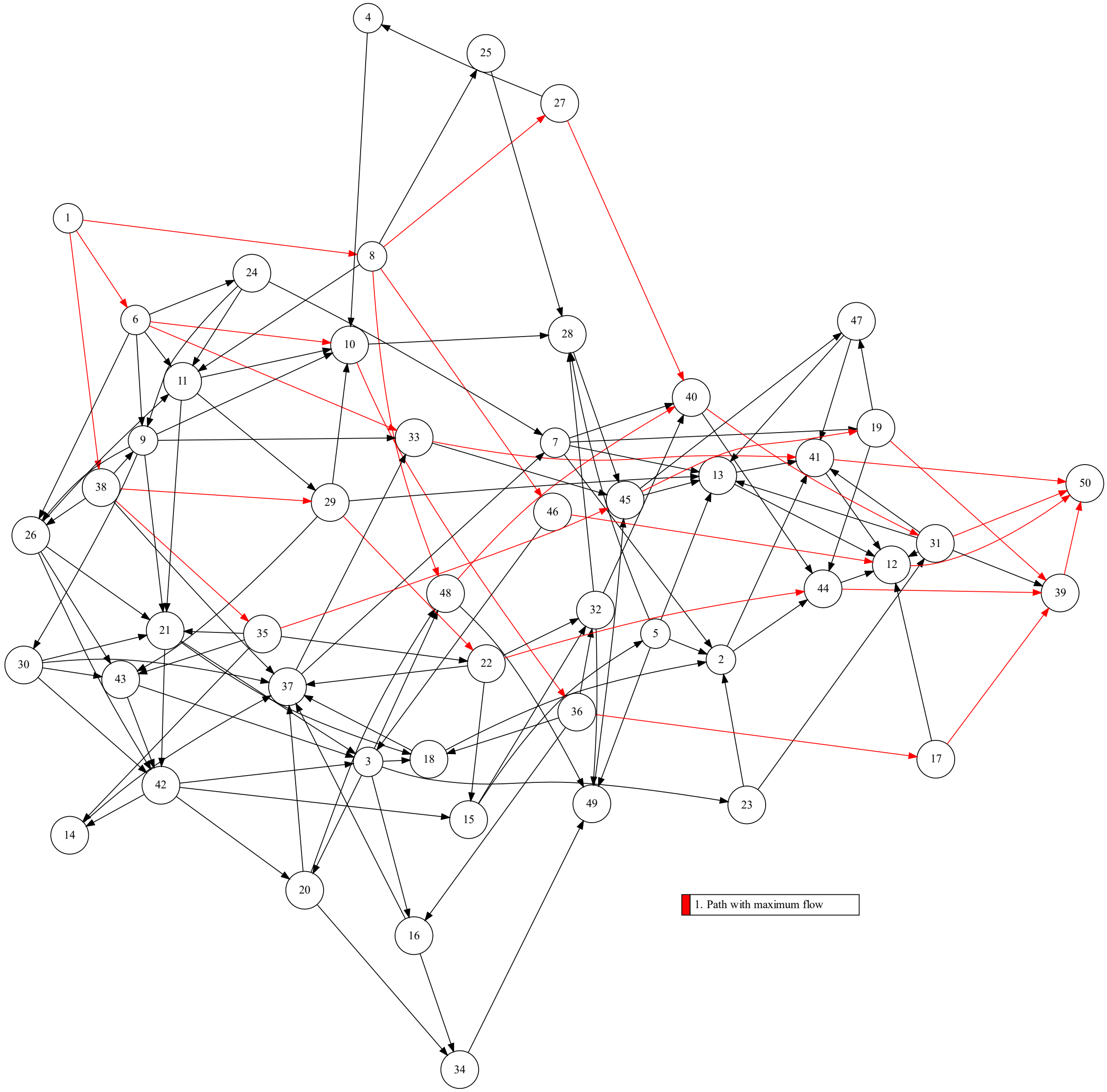
Cenário 2.3 – Análise da complexidade

- Complexidade temporal
  - Complexidade temporal do algoritmo de Edmonds-Karp, ou seja, $O(|V||E|^2)$, em que V representa o número de nós e E o número de arestas.

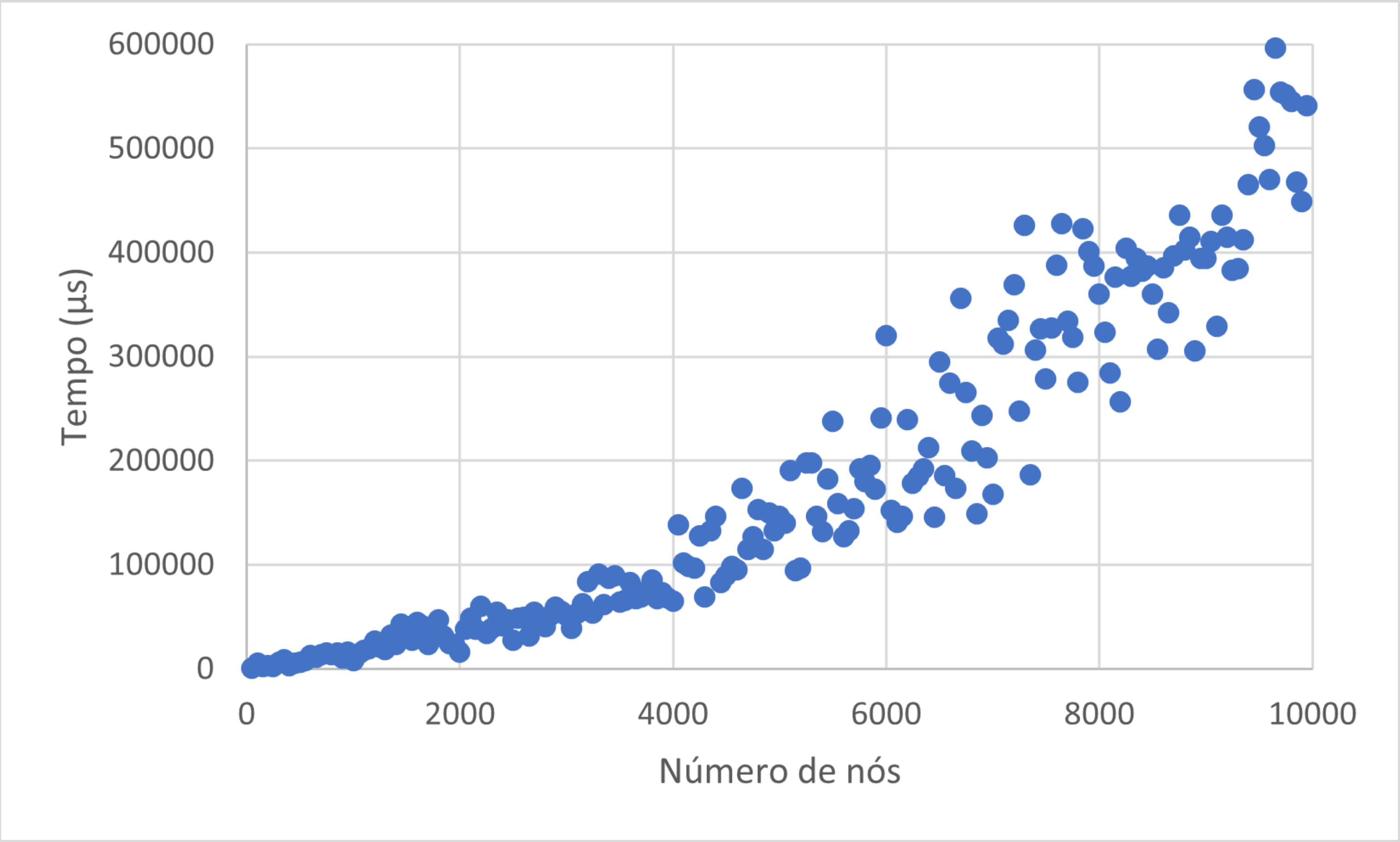
- Complexidade espacial
  - Utilizamos apenas um grafo que representa os caminhos tomados pelo grupo. Assim, temos complexidade espacial O(V), em que V representa o número de nós pertencentes aos caminhos usados pelo grupo.

Cenário 2.3 – Resultados

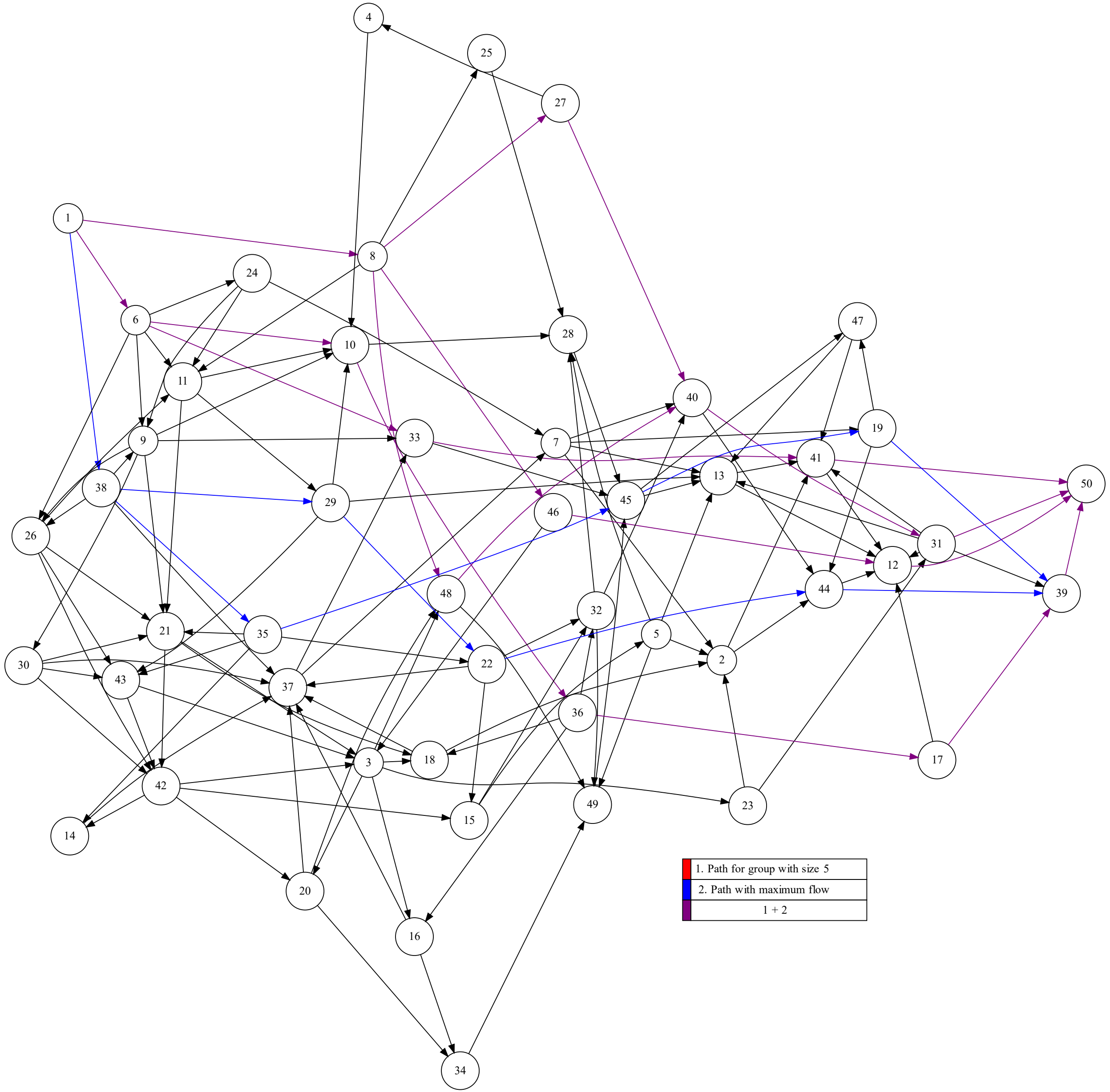




Cenário 2.3 – Resultados



Comparação de resultados das alíneas anteriores



Cenário 2.4 – Formalização

- Dados e variáveis
  - Grafo F, F = (V, E).
- Objetivo:
  - Determinar o maior dos tempos mínimos de chegada ao nó destino no grafo F.
- Restrições:
  - Escolher os caminhos com menor duração.
- Resultados:
  - Inteiro que representa o maior tempo de chegada entre os caminhos com menor duração.

Cenário 2.4 – Algoritmos relevantes

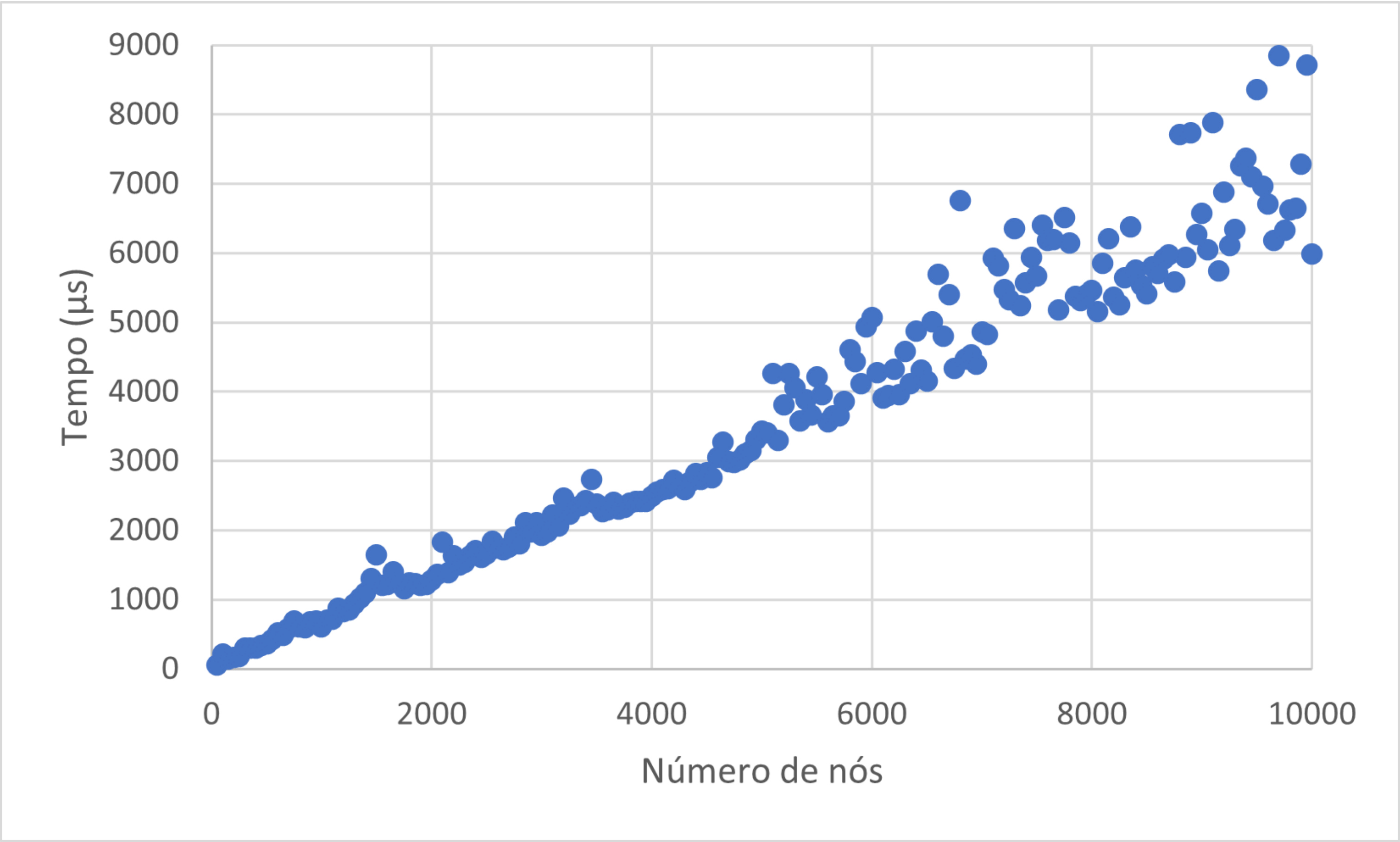
- Utilizamos um algoritmo de earliest start de modo a encontrar o maior dos tempos mínimos de chegada ao nó destino.

Cenário 2.4 – Análise da complexidade

- Complexidade temporal
  - Como utilizamos um algoritmo de earliest start, temos complexidade temporal O(V + E), em que V representa o número de nós e E o número de arestas.

- Complexidade espacial
  - Utilizamos dois unordered maps e uma fila para a implementação do algoritmo de earliest start, logo temos complexidade espacial O(V), em que V representa o número de nós.

Cenário 2.4 – Resultados



Cenário 2.5 – Formalização

- Dados e variáveis
  - Grafo F, F = (V, E).
- Objetivo:
  - Determinar o tempo máximo de espera e os nós onde existem elementos que esperam esse tempo.
- Restrições:
  - Escolher os caminhos de menor duração.
- Resultados:
  - Inteiro que representa o tempo máximo de espera.
  - Vetor de inteiros que contém os nós onde existem elementos que esperam o tempo máximo.

Cenário 2.5 – Algoritmos relevantes

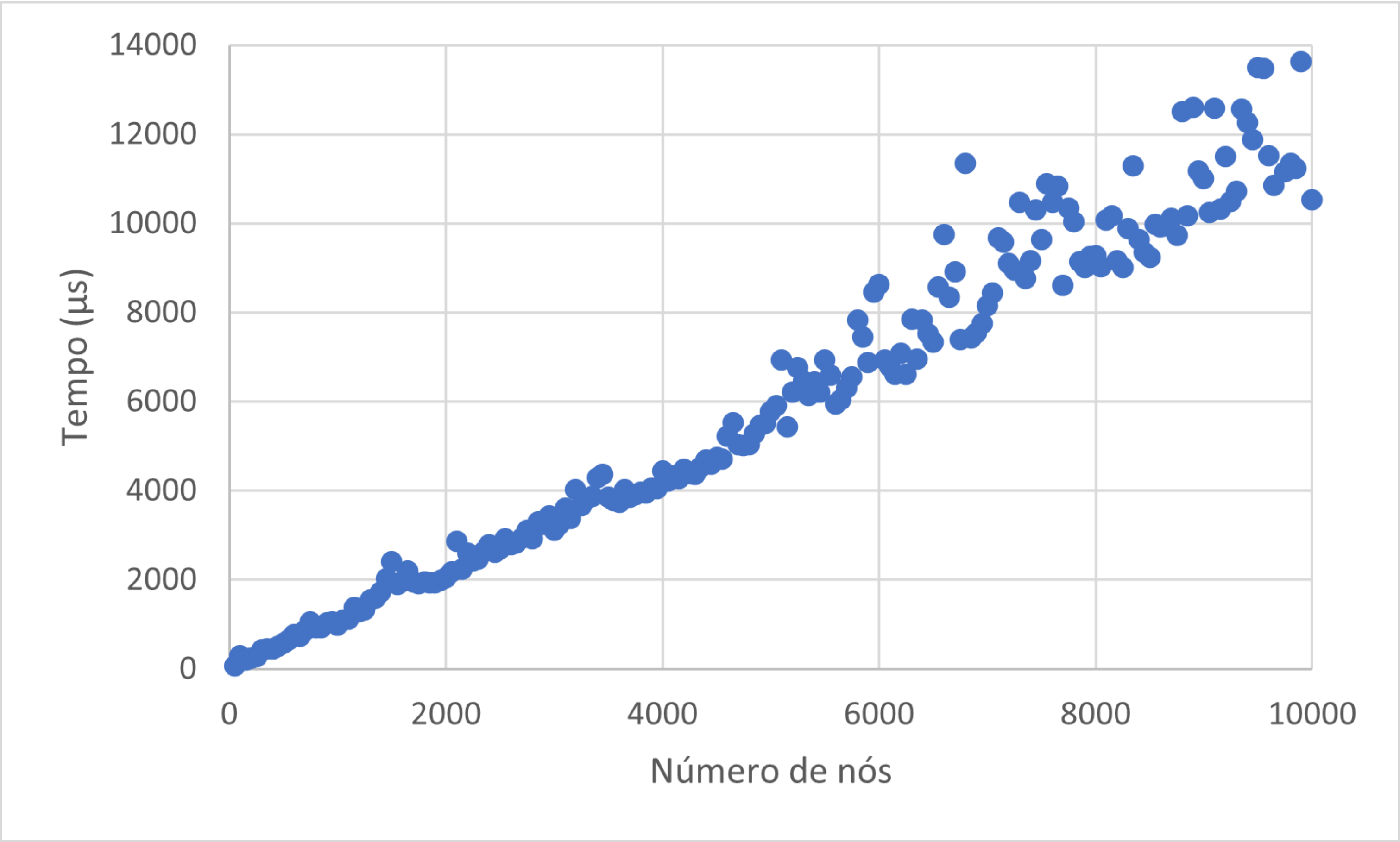
- Partindo dos resultados do cenário anterior, onde em cada nó colocamos os tempos máximos e mínimos de espera (tMáx, tMin), apenas precisamos de saber o tempo de espera maior, ou seja, tMáx – tMin e ver que nós têm esse tempo.

Cenário 2.5 – Análise da complexidade

- Complexidade temporal
  - O mesmo que o cenário 2.4, já que é baseado nele, logo O(V + E), em que V representa o número de nós e E o número de arestas.

- Complexidade espacial
  - Utilizamos apenas um multimap para manter os nós e os respetivos tempos máximos de espera, logo O(V), em que V representa o número de nós.

Cenário 2.5 – Resultados



Conclusão

- Oportunidades para melhorias
  - Em vez de utilizarmos uma matriz para representar a rede residual, poderíamos ter marcado arestas como residuais.
- Dificuldades
  - Interpretação do enunciado, principalmente nas alíneas do cenário 2.
- Participação
  - João Pereira – 1/3
  - Jorge Sousa – 1/3
  - Nuno Pereira – 1/3