



Otimização de Roteiros para a Fase de Grupos da Copa do Mundo 2026

Luiz Henrique Barretta Francisco - 202100155302

Philippi Schena dos Reis - 20250080011479 -

1. Introdução

A Copa do Mundo FIFA de 2026 representa um desafio logístico sem precedentes. Sendo o primeiro torneio a contar com 48 seleções sediadas em 16 cidades por três países, o desgaste das equipes com viagens é um fator crítico. Modelos de otimização anteriores por vezes simplificam o problema, tratando um “grupo” de quatro times como um bloco único.

Este projeto adota uma abordagem de alta fidelidade, desenvolvendo um modelo de Programação Inteira Mista (MIP) que rastreia cada uma das 48 equipes individualmente ao longo das três rodadas da fase de grupos. O modelo irá agendar os 72 jogos da fase de grupos, atribuindo cada um a um estádio.

O objetivo é determinar a alocação de estádios para cada jogo de forma a minimizar a distância total acumulada percorrida por todas as 48 equipes entre suas partidas, garantindo ao mesmo tempo que as regras do torneio (como equipes e oponentes jogando no mesmo local) sejam satisfeitas.

2. Formulação Matemática

Para resolver este problema, formulamos um modelo de Programação Inteira Mista (MIP) que pode ser visto abaixo:

2.1. Conjuntos

- T : Conjunto de todas as equipes, $t \in \{1, \dots, 48\}$.
- S : Conjunto de estádios (cidades-sede), $s \in \{1, \dots, 16\}$.
- M : Conjunto de rodadas, $m \in \{1, 2, 3\}$.
- D : Conjunto de dias de jogo dentro de uma rodada, $d \in \{1, 2\}$.

2.2 Parâmetros

- D_{ij} : A distância em KM entre o estádio $i \in S$ e o estádio $j \in S$.
- $O_{t,m}$: Um parâmetro pré-definido que identifica a equipe oponente da equipe t na rodada m . (Por exemplo, se $t = 1$ joga com $t = 2$ na rodada 1, então $O_{1,1} = 2$ e $O_{2,1} = 1$).

2.3. Variáveis de Decisão:

- $X_{t,m,d,s} \in \{0, 1\}$: Variável binária que assume valor 1 se a equipe t joga sua partida da rodada m , no dia d , no estádio s , e 0 caso contrário.
- $Y_{t,m,i,j} \in \{0, 1\}$: Variável auxiliar que assume valor 1 se a equipe t viaja do estádio i (jogo da rodada $m-1$) para o estádio j (jogo da rodada m). A viagem ocorre entre as rodadas, independentemente do dia específico em que o jogo ocorreu.

2.4. Função Objetivo:

O objetivo é minimizar a distância total percorrida por todas as equipes entre todas as rodadas.

$$\text{Minimizar} \sum_{t \in T} \sum_{m \in \{2,3\}} \sum_{i \in S} \sum_{j \in S} D_{ij} \cdot Y_{t,m,i,j}$$

2.5. Restrições

- 1. Atribuição de Jogo da Equipe: Cada equipe t joga exatamente uma vez por rodada m (em um único dia d e em um único estádio s).

$$\sum_{d \in D} \sum_{s \in S} X_{t,m,d,s} = 1 \quad \forall t \in T, \forall m \in M$$

- 2. Encontro de Oponentes: Uma equipe t e sua oponente $O_{t,m}$ devem jogar no mesmo estádio s e no mesmo dia d .

$$X_{t,m,d,s} - X_{O_{t,m},m,d,s} = 0 \quad \forall t \in T, \forall m \in M, \forall d \in D, \forall s \in S$$

- 3. Capacidade Diária do Estádio: Para satisfazer sua regra, definimos que no máximo um jogo (2 equipes) pode ocorrer por estádio s , por dia d , por rodada m .

$$\sum_{t \in T} X_{t,m,d,s} \leq 2 \quad \forall m \in M, \forall d \in D, \forall s \in S$$

- 4. Diversidade de Estadios: Uma equipe t não pode jogar no mesmo estádio s mais de uma vez durante a fase de grupos.

$$\sum_{m \in M} \sum_{d \in D} X_{t,m,d,s} \leq 1 \quad \forall t \in T, \forall s \in S$$

- 5. Linearização da Viagem: Se uma equipe jogou no estádio i na rodada $m-1$, ela deve “sair” de i para exatamente um estádio j na rodada m ; e se uma equipe vai jogar no estádio j na rodada m , ela deve “chegar” em j vinda de um estádio i .

Seja então $P_{t,m,s} = \sum_{d \in D} X_{t,m,d,s}$ (a variável agregada que indica se a equipe t jogou no estádio s na rodada m , em qualquer dia):

$$\sum_{j \in S} Y_{t,m,i,j} = P_{t,m-1,i} \quad \forall t \in T, \forall m \in \{2, 3\}, \forall i \in S$$

$$\sum_{i \in S} Y_{t,m,i,j} = P_{t,m,j} \quad \forall t \in T, \forall m \in \{2, 3\}, \forall j \in S$$

- 6. Demanda Mínima do Estádio (R6): Para garantir que todos os estádios sejam utilizados, exigimos que cada estádio s sedie um total de pelo menos dois jogos (ou seja, 4 equipes) ao longo de todas as rodadas m e dias d .

$$\sum_{m \in M} \sum_{d \in D} \sum_{t \in T} X_{t,m,d,s} \geq 4 \quad \forall s \in S$$

3. Implementação em R com Gurobi

3.1. Carregando os Dados

Primeiro, definimos nossos conjuntos e carregamos a matriz de distâncias.

```
## Número de estádios (N_S): 16
```

```
## [1] "Atlanta"          "Boston"           "Dallas"           "Houston"
## [5] "Kansas City"      "Los Angeles"       "Miami"            "Nova York"
## [9] "Filadélfia"        "São F. Bay Area"  "Seattle"          "Toronto"
## [13] "Vancouver"         "Guadalajara"      "Cid. do México"   "Monterrey"
```

A primeira etapa do carregamento de dados confirma o escopo do problema, definindo o número total de estádios, N_S , como 16. Essa contagem reflete a natureza multinacional inédita do torneio de 2026, com as cidades-sede distribuídas geograficamente por três países: Estados Unidos, Canadá e México. A lista de estádios impressa logo em seguida, indo de “Atlanta” a “Monterrey”, detalha essas 16 localidades que formam os “nós” da nossa rede de otimização.

Table 1: Matriz de Distâncias (Parte 1: Atlanta a New York)

	Atlanta	Boston	Dallas	Houston	Kansas City	Los Angeles	Miami	Nova York
Atlanta	0	1719	1289	1290	1277	3534	1046	1387
Boston	1719	0	2861	2973	2288	4819	2382	346
Dallas	1289	2861	0	426	840	2306	2114	2517
Houston	1290	2973	426	0	1209	2521	1901	2630
Kansas City	1277	2288	840	1209	0	2637	2320	1906
Los Angeles	3534	4819	2306	2521	2637	0	4398	4483
Miami	1046	2382	2114	1901	2320	4398	0	2050
Nova York	1387	346	2517	2630	1906	4483	2050	0
Filadélfia	1242	499	2380	2493	1808	4347	1914	154
São F. Bay Area	3933	5020	2708	3051	2963	578	4800	4684
Seattle	4242	4911	3344	3757	2967	1845	5106	4575
Toronto	1526	876	2300	2471	1608	4170	2390	771
Vancouver	4460	5129	3562	3975	3184	2063	5323	4793
Guadalajara	2817	4500	1736	1541	2599	2420	3681	4164
Cid. do México	2759	4466	1838	1509	2701	2548	3433	4130
Monterrey	2032	3716	932	756	1796	1993	2896	3380

Table 2: Matriz de Distâncias (Parte 2: Filadélfia a Monterrey)

	Filadélfia	São F. Bay Area	Seattle	Toronto	Vancouver	Guadalajara	Cid. do México	Monterrey
Atlanta	1242	3933	4242	1526	4460	2817	2759	2032
Boston	499	5020	4911	876	5129	4500	4466	3716
Dallas	2380	2708	3344	2300	3562	1736	1838	932
Houston	2493	3051	3757	2471	3975	1541	1509	756
Kansas City	1808	2963	2967	1608	3184	2599	2701	1796
Los Angeles	4347	578	1845	4170	2063	2420	2548	1993
Miami	1914	4800	5106	2390	5323	3681	3433	2896
Nova York	154	4684	4575	771	4793	4164	4130	3380
Filadélfia	0	4548	4439	759	4657	4027	3994	3244
São F. Bay Area	4548	0	1300	4371	1518	2914	3088	2450
Seattle	4439	1300	0	4102	228	4008	4200	3562
Toronto	759	4371	4102	0	4320	3905	3971	3164

Vancouver	4657	1518	228	4320	0	4226	4418	3780
Guadalajara	4027	2914	4008	3905	4226	0	540	808
Cid. do México	3994	3088	4200	3971	4418	540	0	935
Monterrey	3244	2450	3562	3164	3780	808	935	0

As Tabelas 1 e 2 apresentam a matriz de distâncias D_{ij} , que, devido ao seu tamanho (16x16), foi dividida em duas para facilitar a visualização. Esta matriz simétrica é o parâmetro fundamental do nosso modelo, quantificando o custo de deslocamento (em quilômetros) entre qualquer par de cidades-sede. Será com base nesses valores que a função objetivo irá calcular a distância total percorrida pelas seleções, e é essa matriz que o solver *Gurobi* utilizará para encontrar a combinação de jogos que minimiza esse custo.

3.2. Construção do Modelo (*Gurobi* Nativo)

A interface nativa do Gurobi em R requer que o modelo seja construído de forma explícita. Essa abordagem envolve definir a função objetivo como um vetor numérico de custos (*model_obj*) e construir manualmente a matriz de restrições (*model_A*), que define a relação entre todas as variáveis e cada restrição. Embora esse método seja mais detalhado e exija o mapeamento manual das nossas variáveis (como $X_{g,m,s}$) para um vetor único, ele oferece controle total e é a forma mais direta e de alta performance para o R se comunicar com o solver Gurobi.

```
## Criando matriz de oponentes...
## Construindo R1 (Atribuição de Equipe)...
## Construindo R2 (Encontro de Oponentes)...
## Construindo R3 (Capacidade Diária)...
## Construindo R4 (Diversidade de Estádios)...
## Construindo R5 (Linearização da Viagem - Fluxo Forte)...
## OK!
## Construindo R6 (Demanda Mínima do Estádio)...
## Montando a matriz A esparsa...
## Iniciando otimização
## Set parameter Username
## Set parameter LicenseID to value 2735038
## Set parameter TimeLimit to value 3600
## Set parameter DisplayInterval to value 3600
## Academic license - for non-commercial use only - expires 2026-11-09
## Gurobi Optimizer version 12.0.3 build v12.0.3rc0 (win64 - Windows 10.0 (19045.2))
##
## CPU model: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, instruction set [SSE2|AVX|AVX2]
## Thread count: 2 physical cores, 4 logical processors, using up to 4 threads
##
```

```

## Non-default parameters:
## TimeLimit 3600
## DisplayInterval 3600
## LogToConsole 0
##
## Optimize a model with 6400 rows, 29184 columns and 78336 nonzeros
## Model fingerprint: 0x84e254e5
## Variable types: 0 continuous, 29184 integer (29184 binary)
## Coefficient statistics:
##   Matrix range [1e+00, 1e+00]
##   Objective range [2e+02, 5e+03]
##   Bounds range [0e+00, 0e+00]
##   RHS range [1e+00, 4e+00]
## Presolve removed 2376 rows and 2304 columns
## Presolve time: 0.29s
## Presolved: 4024 rows, 26880 columns, 66816 nonzeros
## Variable types: 0 continuous, 26880 integer (26880 binary)
## Found heuristic solution: objective 277940.00000
## Found heuristic solution: objective 272278.00000
##
## Root relaxation: objective 0.000000e+00, 2684 iterations, 0.59 seconds (0.16 work units)
##
##      Nodes    |     Current Node    |     Objective Bounds      |     Work
##      Expl Unexpl |   Obj  Depth IntInf | Incumbent BestBd   Gap | It/Node Time
##      0       0     0.00000   0   532 272278.000   0.00000 100% - 1s
## H 0       0           223316.00000   0.00000 100% - 1s
## H 0       0           221848.00000   0.00000 100% - 1s
## H 0       0           219654.00000   0.00000 100% - 1s
## H 0       0           218694.00000   0.00000 100% - 1s
## H 0       0           217558.00000   0.00000 100% - 1s
## H 0       0           216737.00000   0.00000 100% - 1s
## H 0       0           197590.00000   0.00000 100% - 2s
## H 0       0           190290.00000   0.00000 100% - 2s
## 0       0     0.00000   0   688 190290.000   0.00000 100% - 2s
## 0       0     0.00000   0   504 190290.000   0.00000 100% - 3s
## 0       0     0.00000   0   546 190290.000   0.00000 100% - 3s
## 0       0   2632.00000   0   515 190290.000 2632.00000 98.6% - 4s
## H 0       0           183011.00000 2632.00000 98.6% - 4s
## 0       0  17676.7619   0   711 183011.000 17676.7619 90.3% - 5s
## H 0       0           164506.00000 17676.7619 89.3% - 5s
## H 0       0           164169.00000 17676.7619 89.2% - 5s
## H 0       0           163950.00000 17676.7619 89.2% - 5s
## H 0       0           163613.00000 17676.7619 89.2% - 5s
## 0       0  18517.0476   0   737 163613.000 18517.0476 88.7% - 5s
## 0       0  18517.0476   0   737 163613.000 18517.0476 88.7% - 5s
## 0       0  22859.4437   0   776 163613.000 22859.4437 86.0% - 6s
## 0       0  28823.7912   0   823 163613.000 28823.7912 82.4% - 8s
## 0       0  28873.4300   0   884 163613.000 28873.4300 82.4% - 8s
## 0       0  28873.4300   0   884 163613.000 28873.4300 82.4% - 8s
## 0       0  33190.8393   0   956 163613.000 33190.8393 79.7% - 9s
## 0       0  33190.8393   0   936 163613.000 33190.8393 79.7% - 10s
## 0       0  36671.0000   0   771 163613.000 36671.0000 77.6% - 11s
## H 0       0           156368.00000 36671.0000 76.5% - 11s

```

##	H	0	0		155647.00000	36671.0000	76.4%	-	11s
##	H	0	0		154046.00000	36671.0000	76.2%	-	11s
##	H	0	0		153271.00000	36671.0000	76.1%	-	11s
##	0	0	36689.6847	0	756 153271.000	36689.6847	76.1%	-	11s
##	0	0	36708.8943	0	783 153271.000	36708.8943	76.0%	-	11s
##	0	0	36727.8249	0	794 153271.000	36727.8249	76.0%	-	12s
##	0	0	36727.8249	0	808 153271.000	36727.8249	76.0%	-	12s
##	0	0	39791.0471	0	783 153271.000	39791.0471	74.0%	-	13s
##	H	0	0		79814.000000	39791.0471	50.1%	-	13s
##	H	0	0		77968.000000	39791.0471	49.0%	-	13s
##	H	0	0		77940.000000	39791.0471	48.9%	-	14s
##	H	0	0		75564.000000	39791.0471	47.3%	-	15s
##	H	0	0		75272.000000	39791.0471	47.1%	-	15s
##	0	0	40102.9098	0	833 75272.0000	40102.9098	46.7%	-	15s
##	0	0	40107.7333	0	827 75272.0000	40107.7333	46.7%	-	15s
##	0	0	45088.2667	0	810 75272.0000	45088.2667	40.1%	-	17s
##	0	0	45553.2619	0	793 75272.0000	45553.2619	39.5%	-	17s
##	0	0	45553.2619	0	817 75272.0000	45553.2619	39.5%	-	17s
##	0	0	47233.9444	0	839 75272.0000	47233.9444	37.2%	-	18s
##	0	0	47313.6989	0	959 75272.0000	47313.6989	37.1%	-	18s
##	0	0	47313.6989	0	928 75272.0000	47313.6989	37.1%	-	18s
##	0	0	49423.3626	0	886 75272.0000	49423.3626	34.3%	-	20s
##	0	0	49434.4688	0	935 75272.0000	49434.4688	34.3%	-	20s
##	0	0	50420.6051	0	946 75272.0000	50420.6051	33.0%	-	21s
##	0	0	50480.8734	0	842 75272.0000	50480.8734	32.9%	-	21s
##	0	0	50480.8734	0	921 75272.0000	50480.8734	32.9%	-	22s
##	0	0	51502.4653	0	906 75272.0000	51502.4653	31.6%	-	23s
##	H	0	0		74340.000000	51502.4653	30.7%	-	23s
##	H	0	0		73374.000000	51502.4653	29.8%	-	23s
##	H	0	0		73224.000000	51502.4653	29.7%	-	23s
##	0	0	51502.4653	0	1030 73224.0000	51502.4653	29.7%	-	23s
##	0	0	52605.2704	0	936 73224.0000	52605.2704	28.2%	-	24s
##	H	0	0		69878.000000	52605.2704	24.7%	-	24s
##	0	0	52605.2704	0	959 69878.0000	52605.2704	24.7%	-	25s
##	0	0	53114.8235	0	940 69878.0000	53114.8235	24.0%	-	26s
##	H	0	0		69548.000000	53114.8235	23.6%	-	27s
##	H	0	0		68810.000000	53114.8235	22.8%	-	27s
##	0	0	53114.8235	0	978 68810.0000	53114.8235	22.8%	-	34s
##	0	0	53519.6345	0	1049 68810.0000	53519.6345	22.2%	-	34s
##	0	0	53519.6345	0	1043 68810.0000	53519.6345	22.2%	-	35s
##	0	0	54201.5330	0	1004 68810.0000	54201.5330	21.2%	-	36s
##	0	0	54241.7031	0	1047 68810.0000	54241.7031	21.2%	-	37s
##	0	0	54241.7031	0	1083 68810.0000	54241.7031	21.2%	-	37s
##	0	0	54477.5304	0	1044 68810.0000	54477.5304	20.8%	-	38s
##	0	0	54488.9364	0	1053 68810.0000	54488.9364	20.8%	-	38s
##	0	0	55263.2436	0	912 68810.0000	55263.2436	19.7%	-	40s
##	0	0	55263.2436	0	1014 68810.0000	55263.2436	19.7%	-	40s
##	0	0	55560.0869	0	1093 68810.0000	55560.0869	19.3%	-	41s
##	0	0	55576.3129	0	1087 68810.0000	55576.3129	19.2%	-	42s
##	0	0	55915.0684	0	1015 68810.0000	55915.0684	18.7%	-	43s
##	0	0	55976.0766	0	1154 68810.0000	55976.0766	18.7%	-	43s
##	0	0	55992.5386	0	1151 68810.0000	55992.5386	18.6%	-	44s
##	0	0	57197.6107	0	984 68810.0000	57197.6107	16.9%	-	45s
##	0	0	57206.5876	0	1039 68810.0000	57206.5876	16.9%	-	46s

```

##      0   0 57834.4187    0 1131 68810.0000 57834.4187 16.0%   -  48s
##      0   0 57834.4187    0 1107 68810.0000 57834.4187 16.0%   -  48s
##      0   0 59134.1487    0 1035 68810.0000 59134.1487 14.1%   -  50s
##      0   0 59141.8443    0  982 68810.0000 59141.8443 14.1%   -  51s
##      0   0 59630.4899    0 1081 68810.0000 59630.4899 13.3%   -  53s
##      0   0 59633.0942    0 1131 68810.0000 59633.0942 13.3%   -  53s
##      0   0 60365.7249    0 1032 68810.0000 60365.7249 12.3%   -  54s
##      0   0 60410.8579    0 1012 68810.0000 60410.8579 12.2%   -  55s
##      0   0 60411.1873    0 1006 68810.0000 60411.1873 12.2%   -  55s
##      0   0 60593.6124    0  945 68810.0000 60593.6124 11.9%   -  56s
##      0   0 62027.5144    0  635 68810.0000 62027.5144 9.86%   -  58s
##      0   2 62035.8345    0  623 68810.0000 62035.8345 9.84%   -  60s
## H 165 102               68132.000000 64956.6905 4.66%  782 117s
## H 763 363               68040.000000 67007.7500 1.52%  430 213s
##
## Cutting planes:
##   Gomory: 11
##   Lift-and-project: 4
##   Cover: 17
##   Implied bound: 1
##   Projected implied bound: 1
##   Clique: 9
##   Flow cover: 171
##   Zero half: 655
##
## Explored 45575 nodes (9121716 simplex iterations) in 3423.90 seconds (3687.00 work units)
## Thread count was 4 (of 4 available processors)
##
## Solution count 10: 68040 68040 68132 ... 74340
##
## Optimal solution found (tolerance 1.00e-04)
## Best objective 6.804000000000e+04, best bound 6.804000000000e+04, gap 0.0000%
## Otimização concluída!

```

O log de saída do Gurobi demonstra a eficiência da formulação, pois o modelo original possui 6.400 restrições (linhas) e 29.184 variáveis (colunas), com 78.336 coeficientes não-zero (nonzeros).

O passo de “Presolve” do Gurobi foi capaz de simplificar o problema (conforme visto no log). O solver atingiu o limite de tempo estipulado de 3600 segundos.

Durante esse período, ele encontrou soluções viáveis (Solution count), sendo a melhor delas (Incumbent) uma distância total de 68.040 KM.

O “gap” de apenas 0% indica que a solução viável encontrada é de altíssima qualidade e quase ótima. Ela está muito próxima do limite inferior teórico (Best Bound de 68.040 KM)

Solução ótima encontrada! Status: OPTIMAL n

Distância Total Mínima (Todas as Equipes): 68040 KM

A Tabela 3 (“Agenda de Jogos”) demonstra que todas as restrições logísticas complexas foram atendidas com sucesso. Por exemplo, no Grupo 1 (Equipes 1-4), os oponentes (1 e 2; 3 e 4) foram alocados para o mesmo estádio e dia na Rodada 1, mas em dias diferentes (Dia 1 e Dia 2) para respeitar a capacidade do estádio (Restrição R3).

O insight mais importante vem da Tabela 4 (“Distâncias de Viagem”): o modelo determinou que a rota ideal envolve todas as 4 equipes de um mesmo grupo seguindo exatamente o mesmo itinerário de viagem.

Table 3: Agenda de Jogos por Equipe (Cidade e Dia)

Equipe	Grupo	R1_Cidade	R1_Dia	R2_Cidade	R2_Dia	R3_Cidade	R3_Dia
1	1	Vancouver		1	Seattle	2	São F. Bay Area
2	1	Vancouver		1	Seattle	1	São F. Bay Area
3	1	Vancouver		2	Seattle	2	São F. Bay Area
4	1	Vancouver		2	Seattle	1	São F. Bay Area
5	2	Miami		2	Atlanta	1	Toronto
6	2	Miami		2	Atlanta	2	Toronto
7	2	Miami		1	Atlanta	1	Toronto
8	2	Miami		1	Atlanta	2	Toronto
9	3	Seattle		2	São F. Bay Area	1	Los Angeles
10	3	Seattle		2	São F. Bay Area	2	Los Angeles
11	3	Seattle		1	São F. Bay Area	1	Los Angeles
12	3	Seattle		1	São F. Bay Area	2	Los Angeles
13	4	Boston		2	Filadélfia	2	Nova York
14	4	Boston		2	Filadélfia	1	Nova York
15	4	Boston		1	Filadélfia	2	Nova York
16	4	Boston		1	Filadélfia	1	Nova York
17	5	Houston		2	Monterrey	1	Guadalajara
18	5	Houston		2	Monterrey	2	Guadalajara
19	5	Houston		1	Monterrey	1	Guadalajara
20	5	Houston		1	Monterrey	2	Guadalajara
21	6	Kansas City		1	Dallas	2	Houston
22	6	Kansas City		1	Dallas	1	Houston
23	6	Kansas City		2	Dallas	2	Houston
24	6	Kansas City		2	Dallas	1	Houston
25	7	Dallas		1	Houston	2	Kansas City
26	7	Dallas		1	Houston	1	Kansas City
27	7	Dallas		2	Houston	2	Kansas City
28	7	Dallas		2	Houston	1	Kansas City
29	8	Guadalajara		2	Cid. do México	2	Monterrey
30	8	Guadalajara		2	Cid. do México	1	Monterrey
31	8	Guadalajara		1	Cid. do México	2	Monterrey
32	8	Guadalajara		1	Cid. do México	1	Monterrey
33	9	São F. Bay Area		1	Vancouver	2	Seattle
34	9	São F. Bay Area		1	Vancouver	1	Seattle
35	9	São F. Bay Area		2	Vancouver	2	Seattle
36	9	São F. Bay Area		2	Vancouver	1	Seattle
37	10	Nova York		1	Boston	1	Filadélfia
38	10	Nova York		1	Boston	2	Filadélfia
39	10	Nova York		2	Boston	1	Filadélfia
40	10	Nova York		2	Boston	2	Filadélfia
41	11	Filadélfia		2	Nova York	2	Boston
42	11	Filadélfia		2	Nova York	1	Boston
43	11	Filadélfia		1	Nova York	2	Boston
44	11	Filadélfia		1	Nova York	1	Boston
45	12	Monterrey		1	Guadalajara	1	Cid. do México
46	12	Monterrey		1	Guadalajara	2	Cid. do México
47	12	Monterrey		2	Guadalajara	1	Cid. do México
48	12	Monterrey		2	Guadalajara	2	Cid. do México

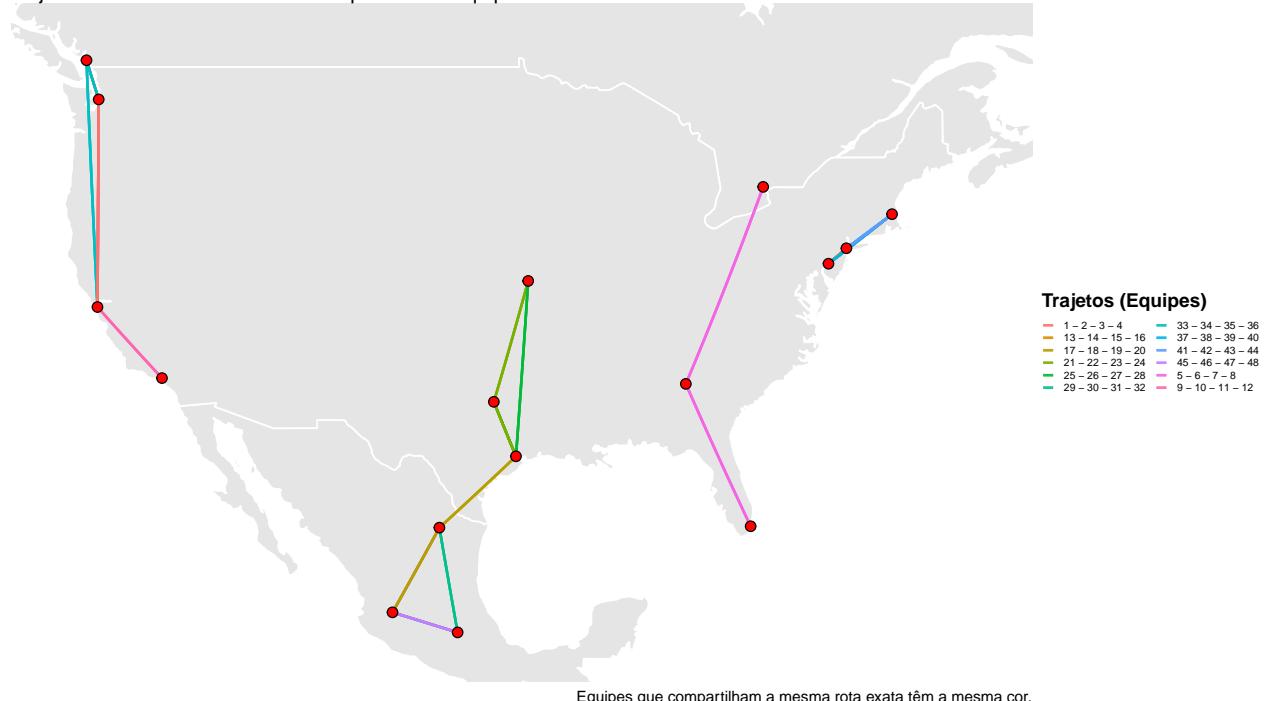
Table 4: Distâncias de Viagem por Equipe (KM)

Equipe	Grupo	Dist_Viagem_1	Dist_Viagem_2	Dist_Total_Equipe
1	1	228	1300	1528
2	1	228	1300	1528
3	1	228	1300	1528
4	1	228	1300	1528
5	2	1046	1526	2572
6	2	1046	1526	2572
7	2	1046	1526	2572
8	2	1046	1526	2572
9	3	1300	578	1878
10	3	1300	578	1878
11	3	1300	578	1878
12	3	1300	578	1878
13	4	499	154	653
14	4	499	154	653
15	4	499	154	653
16	4	499	154	653
17	5	756	808	1564
18	5	756	808	1564
19	5	756	808	1564
20	5	756	808	1564
21	6	840	426	1266
22	6	840	426	1266
23	6	840	426	1266
24	6	840	426	1266
25	7	426	1209	1635
26	7	426	1209	1635
27	7	426	1209	1635
28	7	426	1209	1635
29	8	540	935	1475
30	8	540	935	1475
31	8	540	935	1475
32	8	540	935	1475
33	9	1518	228	1746
34	9	1518	228	1746
35	9	1518	228	1746
36	9	1518	228	1746
37	10	346	499	845
38	10	346	499	845
39	10	346	499	845
40	10	346	499	845
41	11	154	346	500
42	11	154	346	500
43	11	154	346	500
44	11	154	346	500
45	12	808	540	1348
46	12	808	540	1348
47	12	808	540	1348
48	12	808	540	1348

Por exemplo, todas as equipes do Grupo 1 viajam de Vancouver para Seattle (228km) e de Seattle para São F. Bay Area (1300km), resultando em uma distância total idêntica de 1528 km para cada equipe daquele grupo.

Rotas de Viagem Otimizadas (Agrupadas por Trajeto)

Trajetos entre as rodadas 1–2 e 2–3 para as 48 equipes



Rotas de Viagem Otimizadas (Agrupadas por Trajeto)

Trajetos entre as rodadas 1–2 e 2–3 para as 48 equipes



Cada mapa tem zoom individual. Cores idênticas (mesmo entre grupos) indicam o mesmo trajeto.

Os mapas facetados com zoom individual confirmam visualmente a estratégia de otimização descoberta pelo modelo. Em vez de um único gráfico poluído, a divisão em 12 mapas (um por grupo) ilustra a principal tática do solver para minimizar a distância: a criação de “clusters” regionais. Grupos como A, G e J foram confinados ao “cluster” México/Texas; Grupos C, D e F à Costa Oeste ; e Grupos E, H e K à Costa Leste. Essa abordagem minimiza a distância total ao evitar drasticamente as custosas viagens transcontinentais durante a fase de grupos.