

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

**CENTRO DE INFORMÁTICA - CIn UFPE**



**Jorge Guilherme Luna de Vasconcelos Cabral**

**Lucas Van-Lume Lima**

**RELATÓRIO FINAL - PROJETO IA  
(Soccer Calendar Generator)**

**Recife,  
2024**

## Apresentação do problema

No universo do futebol, a criação de calendários de confrontos entre as equipes é uma das principais problemáticas enfrentadas. Com o aumento do interesse dos telespectadores e a expansão de novas competições além das tradicionais, os desafios logísticos e organizacionais se intensificaram. Esses desafios impactam diretamente diversos aspectos, como o desgaste físico dos atletas, a complexidade na organização das atividades dos clubes, o rendimento financeiro da competição e até mesmo a logística de transmissão dos jogos, que depende de uma programação eficiente e bem estruturada.

A elaboração de calendários para competições como o Campeonato Brasileiro de 2025 envolve, além da programação dos confrontos em turnos de ida e volta, o atendimento a um conjunto de regras e restrições específicas que buscam equilibrar a competição e atender a diferentes partes interessadas. Cada time enfrenta todos os outros duas vezes – uma em seu estádio e outra no estádio adversário –, mas essa organização precisa levar em conta fatores como a logística de viagens, a disponibilidade de estádios, a equidade na sequência de mandos de campo, além de atender às necessidades de transmissão televisiva e interesse dos torcedores.

Entre as principais restrições que compõem este problema, destacam-se:

1. **Periodicidade e equilíbrio entre mandos:** As partidas devem ser organizadas em rodadas semanais predefinidas, com alternância entre jogos como mandante e visitante, para evitar que um time jogue consecutivamente em casa ou fora de casa, dentro de um turno ou entre turnos.
2. **Evitar conflitos de estádios e sobreposição de jogos:** Dois jogos não podem ser realizados no mesmo estádio na mesma data, considerando equipes que compartilham arenas ou eventos paralelos no local.
3. **Intervalos adequados para descanso:** Cada equipe deve ter pelo menos 72 horas de descanso entre partidas consecutivas, minimizando o desgaste físico dos atletas e o impacto no desempenho esportivo.
4. **Logística de viagens:** Deve-se reduzir a quantidade de deslocamentos excessivos ou sequências de jogos em locais distantes geograficamente, otimizando o planejamento das viagens e reduzindo os custos para os clubes.
5. **Rodadas e inversão de mandos:** As últimas rodadas do segundo turno devem seguir a mesma configuração das correspondentes rodadas do primeiro turno, porém com os mandos invertidos, garantindo um encerramento equilibrado da competição.

O objetivo deste projeto é otimizar a alocação de datas, horários e locais para o calendário do Campeonato Brasileiro de 2025, equilibrando todas essas restrições e prioridades. Trata-se de um problema de otimização de alta complexidade, que exige a busca por soluções eficientes e de qualidade. Para isso, será utilizado o algoritmo Simulated Annealing (SA), uma técnica reconhecida por sua eficácia em explorar grandes espaços de solução e escapar de mínimos locais, permitindo o refinamento progressivo de soluções até atingir configurações otimizadas. O método permite atender a múltiplas restrições e, ao mesmo tempo, minimizar penalidades relacionadas a logística, disponibilidade de estádios, descanso entre jogos e outros critérios relevantes.

# Simulated Annealing (SA)

O **Simulated Annealing (SA)** é um algoritmo inspirado por processos naturais, como redes neurais artificiais e algoritmos genéticos. Ele é baseado no **recozimento**, um processo térmico amplamente utilizado no aprimoramento de metais, como o aço. Nesse processo, o material é aquecido a altas temperaturas, permitindo que os átomos se movimentem livremente, e, em seguida, resfriado gradativamente, para que as moléculas se reorganizem em uma configuração estrutural mais estável.

No contexto computacional, o SA aplica esse princípio para resolver problemas de otimização. Inicialmente, o algoritmo opera em "temperaturas" elevadas, favorecendo uma busca exploratória no espaço de soluções. Nessa etapa, ele aceita soluções subótimas, simulando a liberdade atômica do recozimento e permitindo escapar de mínimos locais. Esse aspecto torna o SA altamente eficaz em problemas onde outros métodos de otimização podem facilmente se prender a soluções locais. No entanto, esse comportamento também pode ser visto como uma fraqueza, pois, em alguns casos, o algoritmo pode gastar tempo significativo explorando soluções subótimas sem convergir rapidamente para uma boa solução.

Conforme a temperatura é reduzida, a probabilidade de aceitar soluções piores diminui, e o algoritmo foca em exploração local, refinando progressivamente a solução final. Esse refinamento é um ponto forte do SA, pois permite que o algoritmo se aproxime de soluções ótimas com uma boa exploração do espaço de soluções. No entanto, essa transição de exploração para exploração local pode ser um desafio, pois se a temperatura diminuir muito rápido, o algoritmo pode "congelar" em uma solução subótima antes de encontrar a melhor solução possível.

De forma geral, o SA começa em um ponto aleatório no espaço de solução e avalia a qualidade dessa posição. Em seguida, realiza um movimento para um vizinho e avalia o novo ponto. Caso o resultado seja melhor, o algoritmo se move para essa nova posição. Se for inferior, ele ainda pode aceitar o movimento com uma probabilidade definida pela equação:

$$\text{probabilidade}(p) = \text{Exp}((X' - X) / T)$$

Tendo  $X'$  como a qualidade da solução vizinha,  $X$  é a qualidade da solução atual, e  $T$  é a temperatura. Trata-se de um processo eficiente, mas sua eficácia depende diretamente da estratégia de resfriamento adotada, o que pode ser tanto uma vantagem quanto uma limitação, dependendo do tipo de problema.

## Metodologia de Aplicação do Algoritmo SA

### Solução Inicial

Na fase inicial do processo, foi desenvolvida uma solução mínima viável, com o objetivo de criar uma base de referência para o calendário, embora ainda não otimizada. A estrutura adotada para essa representação é uma matriz, onde cada linha corresponde a uma rodada do campeonato e cada coluna representa um confronto dentro dessa rodada.

Cada célula da matriz armazena um objeto contendo informações cruciais para a organização do jogo, incluindo os times participantes, o estádio designado, a data e o horário da partida. Esta abordagem proporciona uma visualização clara da distribuição de jogos e permite um controle eficiente sobre os parâmetros envolvidos na formulação do calendário, funcionando como ponto de partida para as otimizações subsequentes.

## Estruturas de Vizinhança

A estrutura de vizinhança define como as soluções vizinhas são geradas a partir da solução atual. No caso da geração de um calendário esportivo, a vizinhança é explorada por meio de alterações na configuração inicial, empregando diferentes tipos de movimentos que ajustam os jogos, datas, horários e mandos. Esses movimentos são projetados para atender às restrições e otimizar os custos operacionais, logísticos e técnicos.

### Tipos de Movimentos

No contexto da aplicação do algoritmo SA para a geração do calendário, a noção de "movimento" refere-se à alteração na configuração dos jogos dentro da matriz representativa do calendário. Para garantir a exploração eficaz do espaço de soluções e otimizar o calendário conforme as restrições e objetivos definidos, diferentes tipos de movimentos foram considerados. Cada tipo de movimento permite uma modificação na distribuição dos jogos, o que impacta diretamente os custos associados às penalidades.

Os principais tipos de movimento adotados na metodologia foram:

1. **Troca de Jogos:**

Consiste na permuta entre dois jogos de uma mesma rodada. Esse movimento simples é útil para melhorar a compatibilidade com horários nobres e otimizar a logística geográfica, reduzindo penalidades relacionadas a viagens longas ou conflitos de estádio.

2. **Mudança de Mandante:**

Envolve a troca de mandante entre dois jogos, o que altera os estádios envolvidos. Esse movimento otimiza a logística da rodada, equilibrando as distâncias percorridas pelas equipes e reduzindo custos operacionais, além de minimizar penalidades relacionadas ao uso excessivo de um mesmo estádio.

3. **Alteração de Data e Horário:**

Ajusta a data e o horário de um jogo específico dentro de uma rodada. Esse movimento é crucial para atender às restrições de descanso mínimo entre partidas e à disponibilidade de estádios, além de melhorar a distribuição dos jogos ao longo da semana.

4. **Alteração de Datas da Rodada:**

Modifica as datas de todos os jogos de uma rodada, ajustando-as dentro do intervalo de uma mesma semana. No primeiro turno, as datas são redistribuídas aleatoriamente dentro do conjunto de possibilidades da semana, enquanto no segundo turno as datas são ajustadas respeitando a alternância de mandos entre os times. Esse movimento ajuda a evitar conflitos de agendamento e melhora a consistência do calendário.

## Função de Avaliação

A função de avaliação desempenha um papel central no algoritmo de Simulated Annealing, sendo responsável por quantificar a qualidade de uma solução com base nos critérios definidos. No contexto da geração de um calendário esportivo, essa função atribui um custo total à configuração atual, levando em conta diversas penalidades associadas ao descumprimento de restrições ou à subutilização de recursos disponíveis.

Os principais componentes avaliados pela função são:

1. **Conflitos de Estádios**

Penaliza situações em que dois ou mais jogos estão programados para ocorrer no mesmo estádio e na mesma data. Essa restrição visa garantir a viabilidade logística e evitar problemas operacionais.

2. **Intervalos Mínimos entre Jogos**

Avalia se há um intervalo adequado entre as partidas de cada time, garantindo um período mínimo de descanso de 72 horas. Descumprimentos dessa regra acarretam penalidades, pois impactam diretamente o desempenho técnico e o bem-estar dos jogadores.

3. **Distância entre Estádios**

Considera a logística de deslocamento entre as cidades dos clubes. Soluções que resultam em viagens excessivamente longas ou frequentes são penalizadas, dado o impacto no desgaste físico dos atletas e nos custos operacionais dos clubes.

4. **Horários Nobres**

Analisa a distribuição dos jogos em horários de maior audiência e apelo popular. Times com uma participação desigual em horários nobres geram desequilíbrios, resultando em penalidades que incentivam uma alocação mais justa.

5. **Datas Esperadas**

Verifica se os jogos de cada rodada estão alocados dentro das datas previamente definidas. Qualquer divergência resulta em penalidades, assegurando que o calendário seja compatível com as exigências de organização do campeonato.

6. **Sequência de Confrontos Desbalanceados**

Penaliza a ocorrência de confrontos consecutivos contra adversários de alto nível, o que pode gerar desequilíbrios técnicos e impacto psicológico negativo para os times.

A função de avaliação é projetada para considerar a soma ponderada dessas penalidades, garantindo que soluções com menores custos sejam mais atraentes. Além disso, sua formulação permite o balanceamento entre critérios conflitantes, como logística e horários de transmissão, fornecendo uma métrica única e abrangente para a otimização do calendário.

## Resultado

Os resultados obtidos pelo algoritmo **Simulated Annealing** destacam sua eficiência e robustez na otimização do calendário esportivo. Partindo de um custo inicial elevado de **56.000**, o algoritmo conseguiu reduzir esse valor para **5.091**, representando uma melhoria superior a **90%**. Esse avanço expressivo reflete a capacidade do método de explorar soluções vizinhas e realizar ajustes em elementos como datas, horários e mandos,

minimizando penalidades relacionadas à logística, disponibilidade de estádios e intervalos de descanso entre partidas.

### Evolução dos Custos durante a Execução do Simulated Annealing



(HC), também sejam viáveis, ambas apresentam limitações que comprometem sua eficiência e flexibilidade em cenários complexos.

O Constraint Programming, por exemplo, se destaca pela capacidade de modelar múltiplas variáveis e restrições. No entanto, o custo computacional elevado o torna impraticável para problemas de grande escala, como a alocação de datas e horários para múltiplos jogos em uma temporada. Já o Hill Climbing, embora seja uma técnica intuitiva e de rápida implementação, é mais suscetível a cair em **mínimos locais**. Sem um mecanismo que permita aceitar soluções piores temporariamente para escapar dessas armadilhas, como o oferecido pelo Simulated Annealing, o Hill Climbing tende a entregar resultados subótimos.

O Simulated Annealing, por sua vez, equilibra exploração e exploração ao longo do processo de resfriamento. Isso se mostrou crucial para o problema em questão, pois permitiu ao algoritmo escapar de mínimos locais e buscar configurações mais otimizadas. O gráfico de evolução do custo ao longo das iterações exemplifica essa vantagem: enquanto o SA reduziu o custo inicial de **56.000** para **5.091**, garantindo uma solução altamente otimizada, é provável que o HC estacionasse em um custo intermediário muito mais elevado devido à sua incapacidade de explorar vizinhanças menos promissoras temporariamente.

Apesar de suas vantagens, o Simulated Annealing também apresenta desafios e limitações:

- **Dependência da função de custo:** O desempenho depende fortemente de uma função de custo bem projetada, capaz de refletir com precisão os objetivos e restrições do problema.
- **Sensibilidade aos parâmetros:** O ajuste adequado da temperatura inicial, da taxa de resfriamento e do número de iterações é essencial. Parâmetros inadequados podem levar a soluções subótimas ou ao aumento do tempo de execução.
- **Escalabilidade:** Embora mais eficiente que o CP, o SA pode enfrentar dificuldades com uma quantidade muito grande de restrições ou em problemas de escala massiva.

Em síntese, o Simulated Annealing destacou-se por sua eficiência em encontrar soluções de alta qualidade e por sua flexibilidade em lidar com restrições complexas. Comparado ao Constraint Programming e ao Hill Climbing, o SA oferece um equilíbrio superior entre desempenho computacional e capacidade de otimização, sendo especialmente valioso para problemas que requerem um refinamento progressivo e criativo do espaço de busca.

## Conclusão

Neste trabalho, aplicamos o algoritmo **Simulated Annealing** para resolver o problema de construção do calendário do Campeonato Brasileiro de 2025, otimizando custos logísticos, operacionais e técnicos enquanto respeitávamos as restrições definidas. O algoritmo demonstrou grande eficiência, reduzindo o custo inicial de **56.000** para **5.091**, destacando-se em relação a outras abordagens, como o Hill Climbing, que provavelmente ficaria preso em mínimos locais, e o Constraint Programming, que sofre com custos computacionais elevados em problemas de grande escala.

A análise dos resultados, com suporte de gráficos que ilustram a convergência do custo ao longo das iterações, reforçou a flexibilidade e a capacidade do Simulated Annealing de escapar de soluções subótimas. No entanto, também evidenciou a importância de uma boa definição da função de custo e do ajuste cuidadoso dos parâmetros para alcançar resultados satisfatórios.

Assim, concluímos que o Simulated Annealing é uma ferramenta poderosa para problemas complexos de otimização, sendo especialmente adequado para cenários com múltiplas restrições e necessidade de alta adaptabilidade, como a elaboração de calendários esportivos.