Politecnico di Torino
Dipartimento di Scienze Matematiche "G. L. Lagrange"

DISMA
ECCELLENZA 2018·2022

Sport Scheduling Optimization

Candidato:
Davide Omento

Relatore:
Prof. Paolo Brandimarte

Laurea in Matematica per l'Ingegneria

Introduzione allo scheduling sportivo

L'ambito sportivo è oggi un vasto settore economico, con le competizioni che attraggono un vasto pubblico globale e, di conseguenza, generano notevoli ricavi finanziari. Le leghe dunque si affidano ad esperti nel settore che si occupano di ottenere la pianificazione migliore per massimizzare i profitti.

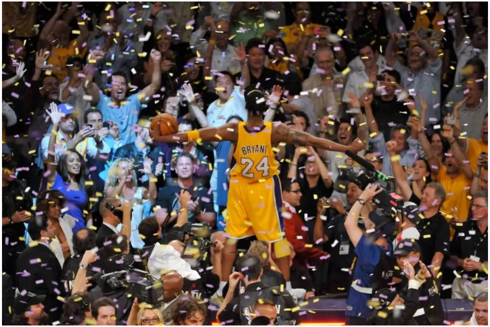

Figure: Kobe Bryant

Ottimizzazione di calendari Round Robin

Il formato dei campionati Round Robin, noti anche come tornei all'italiana, prevede che ciascuna squadra si scontri con tutte le altre partecipanti.

Il nostro obbiettivo è ottenere un calendario di questo tipo che minimizzi il costo delle partite. Per fare ciò ci appoggiamo al seguente modello:

- $T = \{1, 2, ..., n\}$ è l'insieme delle n squadre(n pari), indicizzate da i, $1 \leq i \leq n$.
- $R = \{1, 2, ..., n - 1\}$ è l'insieme dei turni di gioco, indicizzati da r, $1 \leq r \leq n - 1$.
- $M = \{1, 2, ..., (n/2) * (n - 1)\}$ è l'insieme delle partite, indicizzate da m, $1 \leq m \leq (n * (n - 1))/2$.
- $c_{m,r}$ è il costo della partita m se disputata nel turno r.

E' stata introdotta poi una variabile decisionale $x_{m,r}$ con $m \in M, r \in R$ per valutare se la partita m viene giocata nel turno r ($x_{m,r} = 1$) oppure no ($x_{m,r} = 0$).
Il problema può dunque essere modellizzato come:

$$\min \quad \sum_{m \in M} \sum_{r \in R} c_{m,r} x_{m,r}$$

$$\text{s.t.} \quad \begin{cases} \sum_{r \in R} x_{m,r} = 1 & m \in M \\ \sum_{m \in M_i} x_{m,r} = 1 & i \in T, r \in R \\ x_{m,r} \in \{0,1\} & m \in M, r \in R \end{cases}$$

Risultati

Le soluzioni sono state calcolate grazie ad un Optimization Toolbox di MATLAB. Valutiamo grazie a questa tabella quanto tale approccio ci consente di risparmiare rispetto ad un calendario non ottimizzato:

Table: Calendario non ottimizzato vs calendario ottimizzato

N° di squadre	Costo non ottimizzato	Costo ottimizzato	Riduzione (%)
4	32.83	22.66	31
6	85.66	55.16	36
8	141.5	83.5	41
10	250.66	126.66	49
12	355.66	158.5	55

Ottimizzazione del calendario NBA

Il campionato della stagione regolare NBA consiste in:

- 1230 partite in circa 170 giorni.
- 30 squadre suddivise tra due **conference** e sei **division**.
- 82 partite per squadra (41 in casa e 41 in trasferta).

Inoltre, a differenza dei campionati Round Robin, le squadre si affrontano più volte se nella stessa **conference** o stessa **division** e meno volte se sono in **conference** o **division** diverse.

Figure: Logo NBA

Vincoli e obiettivo

Vincoli:

Ogni squadra:

- gioca al massimo una partita al giorno.
- gioca al massimo due partite in tre giorni consecutivi
- gioca al massimo cinque partite in otto giorni consecutivi.
- gioca al massimo diciotto back to back(due partite in due giorni consecutivi) durante la stagione.

Obiettivo:

- ottenere una pianificazione accettabile, ossia in linea con i vincoli imposti, mentre si cerca di ridurre al minimo la somma dei viaggi di tutte le squadre coinvolte.

Il problema viene chiamato NBA Scheduling Problem(NBASP).

Modello matematico

Il modello matematico dell'NBASP assume le seguenti ipotesi:

- $T = \{1, 2, ..., n\}$ è l'insieme delle $n = 30$ squadre.
- $\delta_{i,i'}$ è la distanza in kilometri tra l'arena della squadra i e quella della squadra i'.
- $S = \{1, 2, ..., p\}$ è l'insieme dei giorni disponibili per giocare(circa 170).
- $G = \{1, 2, ..., u\}$ è l'insieme delle partite.

Sia ora $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,p}\}$ la **schedule** ottenuta per la squadra i, dove $x_{i,j} \in \{0\} \cup G$. L'elemento $x_{i,j}$ indica la partita giocata nel giorno j se $x_{i,j} \in G$, altrimenti indica che non c'è nessuna partita in calendario per la squadra i nel giorno j.

Sia $count(a)$ un'operazione che conta quante volte l'argomento a è vero. Posso formalizzare i vincoli in forma matematica:

- **massimo due partite in tre giorni:**
  $\pi_i = count_{j=1,2,...,p-2}(x_{i,j}, x_{i,j+1}, x_{i,j+2} \neq 0) = 0 \quad \forall i = 1, ..., n.$
- **massimo cinque partite in otto giorni:**
  $\rho_i = count_{j=1,2,...,p-7}(count_{j'=j,j+1,...,j+7}(x_{i,j'} \neq 0) > 5) = 0 \quad \forall i = 1, ..., n.$
- **massimo diciotto back to back:**
  $\sigma_i = count_{j=1,2,...,p-1}(x_{i,j}, x_{i,j+1} \neq 0) \leq 18 \quad \forall i = 1, ..., n.$

In definitiva, il problema di ottimizzazione può essere formulato come:

$$\min \quad L = \sum_{i=1}^{n} l_i$$

$$\text{s.t.} \quad \begin{cases} \pi_i = 0 & \forall i = 1, ..., n \\ \rho_i = 0 & \forall i = 1, ..., n \\ \sigma_i \leq 18 & \forall i = 1, ..., n \end{cases}$$

Implementazione

Per prima cosa le squadre sono state numerate da 1 a 30 seguendo l'ordine alfabetico.

Il modello è stato implementato su MATLAB grazie a:

- Una matrice Λ con le distanze $\delta_{i,i'}$
- Dei vettori bidimensionali per rappresentare le partite.
- Una cella **schedule** di dimensione $n \times p$ contenente in posizione (i, j) il vettore bidimensionale con la partita che la squadra i deve disputare il giorno j.
- Un vettore **travel** con le distanze complessive percorse da ciascuna squadra.

A questo punto, l'obiettivo è ottenere una cella **schedule** che, rispettando i vincoli imposti, rende più piccola possibile la somma dei valori del vettore **travel**.

Applicazione dell'Hill Climbing Algorithm

Per ottenere una buona soluzione per l'NBASP è stato impiegato l'algoritmo Hill Climbing.
L'Hill Climbing è un algoritmo iterativo che parte da una soluzione iniziale qualsiasi di un problema e cerca poi di migliorarla apportando cambiamenti piccoli e graduali.

Questo metodo è stato applicato utilizzando due approcci differenti:

- **Spostamento del giorno di una partita**
- **Scambio tra squadra in casa e squadra in trasferta**

Risultati

Per valutare l'efficacia degli algoritmi proposti, sono stati acquisiti i dati relativi alle partite delle ultime tre stagioni NBA dal sito Basketball-Reference.com.

Le soluzioni proposte sono state ottenute calcolando la media dei risultati forniti dagli algoritmi in base a tre diverse soluzioni iniziali, al variare delle iterazioni.

- Spostamento del giorno di una partita:

N° di iterazioni	Riduzione (%)	Tempo di esecuzione(s)
10000	2.02	3.39
100000	9.23	32.57
1000000	21.84	297.32

- Scambio tra squadra in casa e squadra in trasferta:

N° di iterazioni	Riduzione (%)	Tempo di esecuzione(s)
1000	6.08	0.40
10000	11.79	2.79
100000	11.91	31.62

Si nota che il primo algoritmo è meno efficace con poche iterazioni, ma migliora progressivamente all'aumentare di esse. Al contrario, il secondo algoritmo funziona meglio con poche iterazioni ma si stabilizza senza ulteriori miglioramenti una volta raggiunta una soglia critica.

Conclusioni

- Calendari Round Robin:
  - L'ottimizzazione è fondamentale per calendari di questo tipo.
  - Esistono molti modelli a cui appoggiarsi che variano in base agli obiettivi dell'ottimizzazione.
- Calendario NBA:
  - Il primo algoritmo si dimostra più efficace del secondo in quanto, entro tempi ragionevoli, produce risultati significativamente superiori.
  - Il problema affrontato è stato semplificato rispetto alla complessità del problema reale che tiene conto di numerosi altri fattori(disponibilità delle arene, distribuzione delle partite giocate nei weekend,...).

Grazie per l’attenzione

Politecnico di Torino

1859