DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1: Sist. de Ecuaciones Lineales

September 9, 2021 — Métodos Numéricos

Grupo 13

Integrante	LU	Correo electrónico
Bogetti, Gianfranco	693/15	gianbogetti7@hotmail.com
Strobl Leimeter, Matias Nicolás	645/18	matias.strobl@gmail.com
Fabian, Florencia	230/19	flor.fabian@hotmail.com
Losiggio, Ignacio	751/17	iglosiggio@dc.uba.ar

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Abstract

Realizamos una descripción de múltiples métodos de scoring (CMM, WP y Elo) en conjunto con uno nuevo, Justice. Realizamos una evaluación de éstos métodos tanto contra datos sintéticos con un modelo probabilístico subyacente cómo contra datos propios de información deportiva públicamente disponible. Finalmente, damos un enfoque inicial para la búsqueda de falencias en éstos métodos al intentar maximizar las victorias de un equipo dado al mismo tiempo que minimizamos las derrotas.

Keywords— Ranking deportivo, CMM, Elo, Probabilidad

Contents

1 Introducción

Este trabajo tiene por objetivo analizar los distintos métodos de ranking para competencias deportivas. Como se sabe, existen distintas formas de puntuar equipos o jugadores individuales en una determinada disciplina. No necesariamente los métodos usados son los más justos, y encontrar un método que refleje la complejidad de la competencia no es una tarea fácil. Se puede tomar en consideración la cantidad de partidos ganados, el "puntaje" del contrincante e incluso datos cómo cuál equipo era el anfitrión y cuál el visitante.

El fin del trabajo es estudiar el comportamiento de diferentes medidas de desempeño deportivo respecto de una lista de partidos con sus resultados. De las medidas nos va a importar el orden "de mejor a peor" por sobre el significado numérico de las mismas, dado que éste orden es el que usa al construir tablas de posiciones que determinan el ranking de cada participante en un deporte y/o competición.

Se presentan en la sección 2 cuatro métodos de ranking distintos. Cada uno de estos se explica por separado en conjunto con los detalles implementativos que consideramos relevantes. En la sección 3 planteamos las preguntas a responder sobre éstos métodos. En la sección 4 describimos nuestro diseño experimental. Finalmente, en la sección 5 mostramos nuestros resultados y presentamos nuestras conclusiones respecto de las preguntas planteadas.

2 Métodos de Ranking

2.1 Winning Percentage (WP)

El método WP consiste en calcular para cada equipo el porcentaje de partidos ganados ($\frac{\#\text{Ganados}}{\#\text{Jugados}}$). Éste puntaje es un estimador razonable de la probabilidad de ganar un partido si tomamos *"todos los partidos son igual de difíciles"* cómo una suposición valida.

Nuestra implementación utiliza dos vectores. Uno dónde se almacena el conteo de los partidos ganados y otro dónde se almacena el conteo de partidos jugados. El resultado final se calcula dividiendo éstos vectores componente a componente. Un detalle importante respecto de ésta implementación es que no hay resultado posible para un equipo que jugó cero partidos.

La ventaja principal de éste método es su simplicidad tanto conceptual cómo de cálculo. Además, podemos actualizar los puntajes asignados con facilidad (si los guardamos cómo tuplas $(\#\text{Ganados}, \#\text{Jugados})$), simplemente sumando $(1, 1)$ ó $(0, 1)$ según corresponda). Finalmente, es fácil incorporar el concepto del "empate" a este método, por ejemplo sumando la idea de que *"un empate es media victoria"* (o $\frac{1}{3}$ de victoria, cómo es común en muchas ligas de fútbol).

Las desventajas del método provienen de que la suposición central es obviamente falsa. El no considerar el puntaje del contrincante al momento de actualizar el propio luego de un partido puede llevar a situaciones un tanto ridículas cómo tener equipos con 100% probabilidad de victoria (o derrota).

2.2 Colley Matrix Method (CMM)

El método de Colley está basado en la regla de sucesión de Laplace (también conocida cómo estimador de Laplace-Bayes) para estimar las probabilidades de que un equipo gane o pierda. Esta regla nos propone que vistos n eventos, de los cuales s fueron exitosos, entonces la probabilidad de que el próximo evento sea exitoso será:

$$P(X_{n+1} \mid \sum_{i=1}^{n} X_i = s) = \frac{s+1}{n+2} \tag{1}$$

Teniendo T equipos, el método CMM propone construir una matriz $C \in \mathbb{R}^{T \times T}$ y un vector $b \in \mathbb{R}^T$. El ranking buscado va a estar dado por el orden de los equipos en $r \in \mathbb{R}^T$ la solución del sistema $Cr = b$. Tomando índices $\forall$ i,j $\in$ {1,...T} podemos definir C y b de la siguiente manera:

$$C_{ij} = \begin{cases} -n_{ij}, & i \neq j \\ 2 + n_i, & i = j \end{cases} \tag{2}$$

$$b_i = 1 + \frac{w_i - l_i}{2}, \tag{3}$$

- n_{ij}: Cantidad de partidos que jugó i contra j
- n_i: Cantidad de partidos que jugó i

- w_i: Cantidad de victorias de i
- l_i: Cantidad de derrotas de i

Cada r_i final estima las probabilidades de el equipo i gane el próximo partido (basándose en la regla de sucesión de Laplace).

Observando C podemos ver que es cuadrada, simétrica y estrictamente diagonal dominante. La simetría proviene de que $n_{ij} = n_{ji}$, por lo que todos los elementos por fuera de la diagonal se ven reflejados. Por otra parte, es fácil notar que $\forall \sum_{j=1}^{n} n_{ij} = n_i$ (los equipos no juegan contra sí mismos) y con esto concluir que la suma de los elementos de toda fila (o columna) es 2.

Con las propiedades de C descriptas podemos concluir que la matriz tiene factorización LU (por estrictamente diagonal dominante). Esto nos garantiza la factibilidad de aplicar EG sin intercambio de filas o columnas.

2.3 Elo

El sistema de puntuación Elo es un método matemático, basado en cálculo estadístico, para calcular la habilidad relativa de los jugadores de deportes como el ajedrez. Debido a que existe más de una variación del método en uso, usamos la de USCF (Federación de Ajedrez de Estados Unidos). La puntuación de un jugador se determina a partir de sus resultados contra otros jugadores. Dadas dos puntuaciones A y B la idea central del método es que el se puede estimar la puntuación esperada de A (1 es ganar, 0 es perder y $\frac{1}{2}$ es tablas) de la siguiente manera:

$$E_A = \frac{1}{(1 + 10^{\frac{A-B}{400}})} \tag{4}$$

La actualización de este puntaje se realiza respecto de la desviación del puntaje esperado para el jugador. Éste puntaje no necesariamente tiene que referirse a un único partido (tomar la suma de los puntajes esperados y obtenidos a lo largo de un torneo sería una forma válida de actualizarlo). El cambio de puntaje es un simple ajuste lineal proporcional a la diferencia entre la puntuación esperada y la obtenida por un jugador. El máximo ajuste posible por partida, llamado K es asignado de la siguiente manera:

$$K = \begin{cases} 32 & \text{si Elo} < 2100 \\ 24 & \text{si Elo} \in [2100; 2400] \\ 16 & \text{si Elo} > 2400 \end{cases} \tag{5}$$

Dicho esto llamemos E_A al puntaje esperado para A, S_A el obtenido y R_A el Elo de A. Con ésto, el "próximo" R_A (escrito R'_A) será:

$$R'_A = R_A + K(E_A - S_A) \tag{6}$$

De todo lo dicho ahora hay varias cosas notables a destacar:

- El elo de los participantes luego de analizar los partidos $R_{AvB}, R_{CvD}, \ldots, R_{BvA}$ (actualizando los puntajes partido a partido) es **sensible al orden** de éstos.
- Al enfrentarse los jugadores A y B ocurre que $E_A + E_B = 1$.
- Luego de enfrentarse los jugadores A y B ocurre que $S_A + S_B = 1$.

En nuestra implementación del método los participantes parten de un puntaje de 1500 puntos. La implementación propuesta procesa una lista acotada de partidos entre jugadores. Es interesante notar que Elo **puede** ser calculado incrementalmente (es decir, dar un ranking *"partido a partido"*).

2.4 Justice

Finalmente, inventamos un nuevo método para analizarlo en conjunto con los anteriores al que llamamos *"Justice"*.

Justice calcula el WP de cada equipo, y le suma el término $\frac{r_i}{n_i}$ que representa del rendimiento de sus rivales. Para el equipo i el puntaje final P_i será:

$$P_i = \frac{w_i + r_i}{n_i} = \frac{w_i}{n_i} + \frac{r_i}{n_i} = \text{WP}_i + \frac{r_i}{n_i} \tag{7}$$

- w_i: Cantidad de victorias del equipo i

- n_i: Cantidad de partidos jugados por el equipo i
- r_i: $w_i - n_i + \sum_{j=1}^{n_i} \mathrm{WP}_{\text{Partidos[i].contrincante}}$

Por ejemplo, si el equipo i ganó contra el j y perdió contra el k tendremos:

$$r_i = 1 - 2 + \mathrm{WP}_j + \mathrm{WP}_k = -1 + \frac{w_j}{n_j} + \frac{w_k}{n_k}$$

Este método de ranking toma en consideración tanto el rendimiento del equipo en cuestión cómo el de sus contrincantes. La manera en la que está construido no nos ofrece a simple vista un mecanismo incremental de cálculo sencillo cómo en WP o Elo, dado que los resultados de una partida pueden tener repercusiones en el puntaje de todo el resto de los participantes.

3 Preguntas

Durante el resto de este trabajo expondremos los resultados de nuestro análisis. Las siguientes fueron las preguntas de partida utilizadas para las la sección 5.

- ¿Cómo se comparan los resultados de nuestra implementación de CMM con los ofrecidos por la cátedra?
- ¿Cuáles de los métodos ofrecidos son *justos*? ¿Cómo afecta la dificultad del *schedule* a la calidad de los resultados?
- ¿Cuál método aproxima mejor "la realidad" dado un torneo de eliminación directa?
- ¿Y dado uno Round-Robin?
- ¿Qué características notables tiene cada uno de los métodos en la práctica?

4 Diseño experimental

4.1 Comparación de rankings

Durante el desarrollo de nuestra experimentación tuvimos la necesidad de comparar rankings múltiples veces. Con fin de agilizar nuestro trabajo decidimos utilizar el coeficiente de correlación de Spearman (ρ). Éste está especialmente pensado para trabajar con rankings, por lo que sólo usa el orden dado por cada método sin enfocarse en el scoring asignado por él.

Nuestra implementación de ρ esta disponible en `experimentacion/compare_rankings.py`.

4.2 Creación de torneos

En varios de nuestros experimentos generamos nuestros propios "torneos". Decidimos partir de la suposición de Elo sobre cómo ese puntaje estima la probabilidad de victoria de todo par de participantes. Este enfoque nos ofrece una *ground-truth* contra la que comparar los rankings. Finalmente, suponemos la población de jugadores/equipos normalmente distribuida.

Con estos supuestos creamos una lista de equipos resultado de muestrear una $N(\mu = 1500, \sigma = 500)$. Estos equipos fueron sometidos a múltiples torneos. El resultado de cada partido se calculó usando el estimador propuesto por Elo (E_i) cómo p en un ensayo de Bernoulli. Se consideró 1 victoria del anfitrión y 0 victoria del visitante. La lista de equipos y partidos puede ser recreada determinísticamente por medio del script `experimentacion/sample_tournament.py`.

4.2.1 Eliminación directa

Estos torneos requieren que la cantidad de equipos sea potencia de 2. Si el requisito tomamos una permutación aleatoria de los equipos (simulando un sorteo) y emparejamos los equipos ronda-a-ronda hasta quedarnos con el ganador.

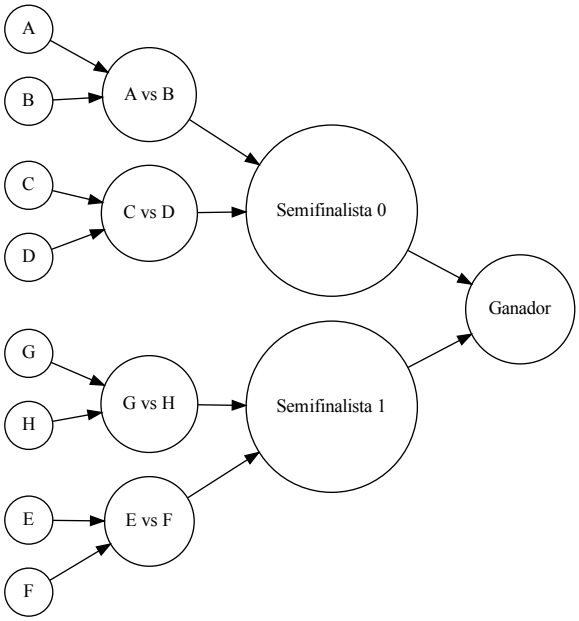

Figure 1: Torneo de eliminación directa generado para $n = 8$

4.2.2 Round-Robin

Un torneo Round-Robin resulta sencillo conceptualmente: todos los equipos juegan contra todos los equipos una vez (un torneo ida y vuelta se puede construir concatenando dos torneos de un partido). Cómo algunos de los métodos implementados son sensibles al orden de las partidas es importante generar el "fixture" de forma correcta (un ejemplo dónde esto no se cumple puede verse en los datos de prueba dentro de la carpeta `test_completos`). Para la planificación del torneo usamos la estrategia descripta en https://nrich.maths.org/1443.

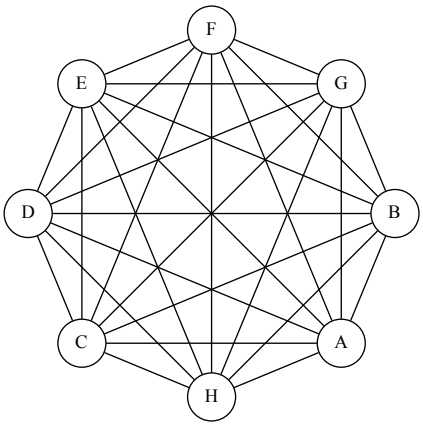

Figure 2: Torneo Round-Robin generado para $n = 8$

4.2.3 Torneo *"a la FIFA™"*

Finalmente, utilizamos el formato de torneo de la FIFA World Cup™ con la variación de que en caso de empate en la fase de grupos gana el equipo con menor orden ($T_0 < T_1 < \cdots < T_{31}$). El fixture está basado en el publicado

para la copa que se ocurrirá en Qatar el año próximo, disponible en https://www.fifa.com/tournaments/mens/worldcup/qatar2022.

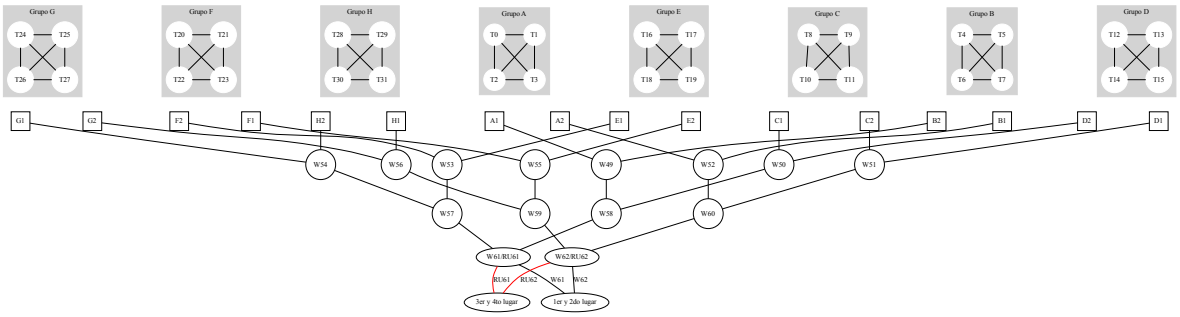

Figure 3: Torneo FIFA generado

5 Resultados y discusión

5.1 Márgenes de error de nuestra implementación de CMM

Tomando los resultados ofrecidos por la cátedra en `tests/` cómo *"ground truth"* nos interesa ver cómo se desvía nuestra implementación de CMM basada en eliminación gaussiana de ellos. Para esto desarrollamos un pequeño script (`experimentacion/compare_results.py`) que ejecuta nuestra implementación y la compara con los resultados provistos. Tomando $X_i = |P_{i,\text{Cátedra}} - P_{i,\text{Nuestro}|}$ los resultados fueron los siguientes:

Test	#equipos	#partidos	$\bar{X}$	σ	$\min X_i$	$\max X_i$
`test1`	6	13	2.118×10^{-7}	1.282×10^{-7}	1.083×10^{-7}	4.169×10^{-7}
`test2`	6	10	2.599×10^{-7}	1.201×10^{-7}	7.989×10^{-8}	4.197×10^{-7}
`test-prob-1`	6	11	3.312×10^{-7}	1.020×10^{-7}	2.218×10^{-7}	4.775×10^{-7}
`test-prob-2`	6	12	1.706×10^{-7}	1.409×10^{-7}	2.117×10^{-8}	3.486×10^{-7}
`test_completo_10_1`	10	45	2.702×10^{-7}	1.373×10^{-7}	8.796×10^{-9}	3.440×10^{-7}
`test_completo_100_4`	100	12410	8.263×10^{-7}	3.107×10^{-7}	2.494×10^{-7}	1.379×10^{-6}
`test_completo_100_8`	100	22318	2.497×10^{-6}	3.142×10^{-7}	1.817×10^{-6}	3.055×10^{-6}
`test_completo_1000_2`	1000	749273	1.493×10^{-5}	3.745×10^{-7}	1.367×10^{-5}	1.613×10^{-5}
`test_completo_1000_8`	1000	2248231	6.738×10^{-5}	4.533×10^{-7}	6.578×10^{-5}	6.893×10^{-5}

Cómo podemos ver tanto $\bar{X}$ cómo σ parecen crecer con el tamaño de los equipos. Por la manera en la que EG funciona esto es esperable: por cada equipo nuevo agregamos $2n + 1$ elementos a la matriz (una fila y una columna) y eliminación gaussiana realiza una cantidad de operaciones primitivas proporcional a n^2. Entendemos que otros métodos de cálculo podrían dar lugar a resultados más rápidos y más precisos.

Sumado a esto también quisimos visualizar la distribución de éstos errores esperando inicialmente que esté centrada alrededor del 0, los resultados se muestran a continuación.

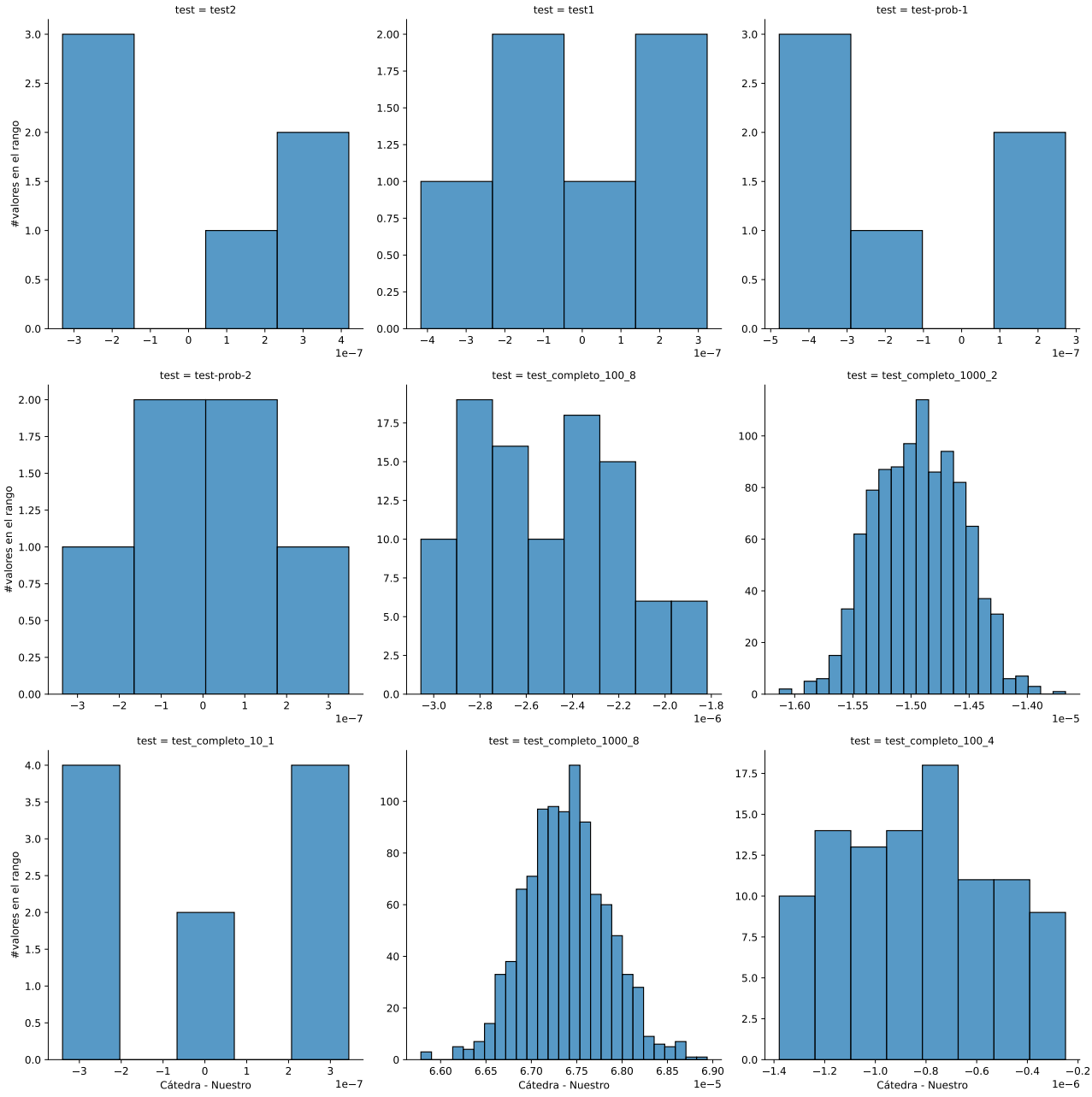

Figure 4: Histograma de $P_{i,\text{Cátedra}} - P_{i,\text{Nuestro}}$ para cada test provisto

Para sorpresa nuestra los errores no siempre se centraban alrededor del cero. Es una suposición que parece cierta para los tests más chicos (`test2`, `test1`, `test-prob-1`, `test-prob-2`, `test_completo_10_1`) pero no sucede con los casos de prueba de 100 participantes o más (`test_completo_100_8`, `test_completo_1000_2`, `test_completo_100_4`, `test_completo_1000_8`). Qué el error esté centrado fuera del cero nos da la pauta de que **nuestra implementación converge a números levemente distintos**. Esto puede darse tanto por defectos propios de la implementación cómo por precisión perdida por el método de resolución de sistemas de ecuaciones lineales utilizado.

5.2 Comparación de los métodos de ranking

5.2.1 *"Justicia"* de los métodos

En primera instancia, necesitamos una definición de *justo* con la que trabajar. Desde propuesto por la cátedra hay dos interpretaciones levemente distintas que nos interesa analizar:

1. Justo bajo sufijo: Dada una lista de resultados $r_1, r_2, \ldots, r_n$ cambiar **un sufijo** de la lista sólo cambia

los puntajes de los equipos que participan en éste sufijo.

2. Justo bajo infijo: Dada una lista de resultados $r_1, r_2, \ldots, r_n$ cambiar **una subcadena** de la lista sólo cambia los puntajes de los equipos que participan de esa subcadena.

Es interesante notar que Justo bajo infijo implica Justo bajo sufijo (dado que todo sufijo es una subcadena).

De los métodos vistos en la sección 2 WP resulta Justo bajo infijo, Elo Justo bajo sufijo, finalmente tanto CMM cómo Justice no muestran ninguna de estas propiedades. Una pequeña explicación justificando cada uno de los casos:

- **Winning Percentage:** Observemos que el orden de los partidos no importa. Hecho esto, modificar los resultados de un partido sólo modifica la cantidad de victorias de los equipos participantes. Agregar o quitar un partido modifica la cantidad de victorias y la cantidad de partidos de los equipos participantes.

  Dado que los números modificados sólo se utilizan en el cálculo de los partidos que participaron de los equipos en cuestión y que el orden de los partidos no cambia el resultado final el método de ranking resulta Justo bajo infijo.

- **CMM:** Es fácil observar que el método no es sensible al orden de los partidos (dada la construcción de C). Podemos descartar que sea sólo Justo bajo sufijo por esta razón. Pese a esto, el método no resulta Justo bajo infijo.

  Miremos el siguiente par de sistemas que nos sirven de contraejemplo:

$$\begin{pmatrix} 6 & -2 & -2 \\ -2 & 5 & -1 \\ -2 & -1 & 5 \end{pmatrix} \begin{pmatrix} 3/4 \\ 7/24 \\ 11/24 \end{pmatrix} = \begin{pmatrix} 3 \\ -1/2 \\ 1/2 \end{pmatrix}$$

$$\begin{pmatrix} 6 & -2 & -2 \\ -2 & 5 & -1 \\ -2 & -1 & 5 \end{pmatrix} \begin{pmatrix} 5/8 \\ 7/16 \\ 7/16 \end{pmatrix} = \begin{pmatrix} 2 \\ 1/2 \\ 1/2 \end{pmatrix}$$

  Éstas matrices se corresponden a las construidas para resolver `cmm-fairness-a.txt` y `cmm-fairness-b.txt` dónde sólo el partido final se modifica. Éstos ficheros se encuentran en `experimentacion/contraejemplo/`. Pese a que sólo cambia el resultado de un partido del equipo 1 contra el equipo 2 esto afecta la puntuación del equipo 3. Sin embargo, no creemos que ésta modificación sea negativa, dado que es parte de la corrección por dificultad en el *schedule* propia del método.

- **Elo:** Cómo ya discutimos anteriormente, el sistema Elo es sensible al orden de los partidos. Ésto descarta toda posibilidad de ser Justo bajo infijo. Sin embargo, es fácil notar que la ecuación de actualización (6) sólo modifica los puntajes de los participantes de la partida en cuestión. Es por esto que dada toda secuencia de partidas agregar partidas al final o borrar partidas desde el final sólo modifica los puntajes de los jugadores que participaron de éstas. Por esto concluimos que Elo es Justo bajo sufijo.

- **Justice:** Al igual que CMM, Justice propaga los resultados de una partida a todos los jugadores que jugaron con los participantes de ésta. Suponemos fácil ver que al tener en cuenta el WP de los rivales no es posible ser Justo bajo sufijo.

5.2.2 Rendimiento de los métodos ante distintos tipos de torneo

Distintos tipos de torneo ofrecen distinta información sobre los participantes. Decidimos realizar un análisis sencillo comparando el tipo de torneo que creemos provee la menor cantidad de información (eliminación directa) con el que creemos provee la mayor (Round-Robin). Para realizar este análisis tomamos 100 torneos de cada tipo con distinto sorteo inicial, disponibles en `experimentacion/experimental-data` luego de correr `experimentacion/sample_tournament.py`. Más información sobre cómo se generan estos torneos está disponible en la sección 4.

Para la comparación de rankings decidimos usar el coeficiente de correlación de Spearman (ρ) dado que éste opera sobre el *orden* de los datos de la muestra en lugar del scoring asignado (rank-correlation). Dentro de esta comparación nos interesa saber cuáles rankings son *justos* y cómo se comportan de acuerdo a los cambios de *schedule*. Nuestra implementación de ρ está disponible en `experimentacion/compare_rankings.py`. Los rankings *"reales"* (usados para samplear los torneos) son generados en con el nombre `players.txt`.

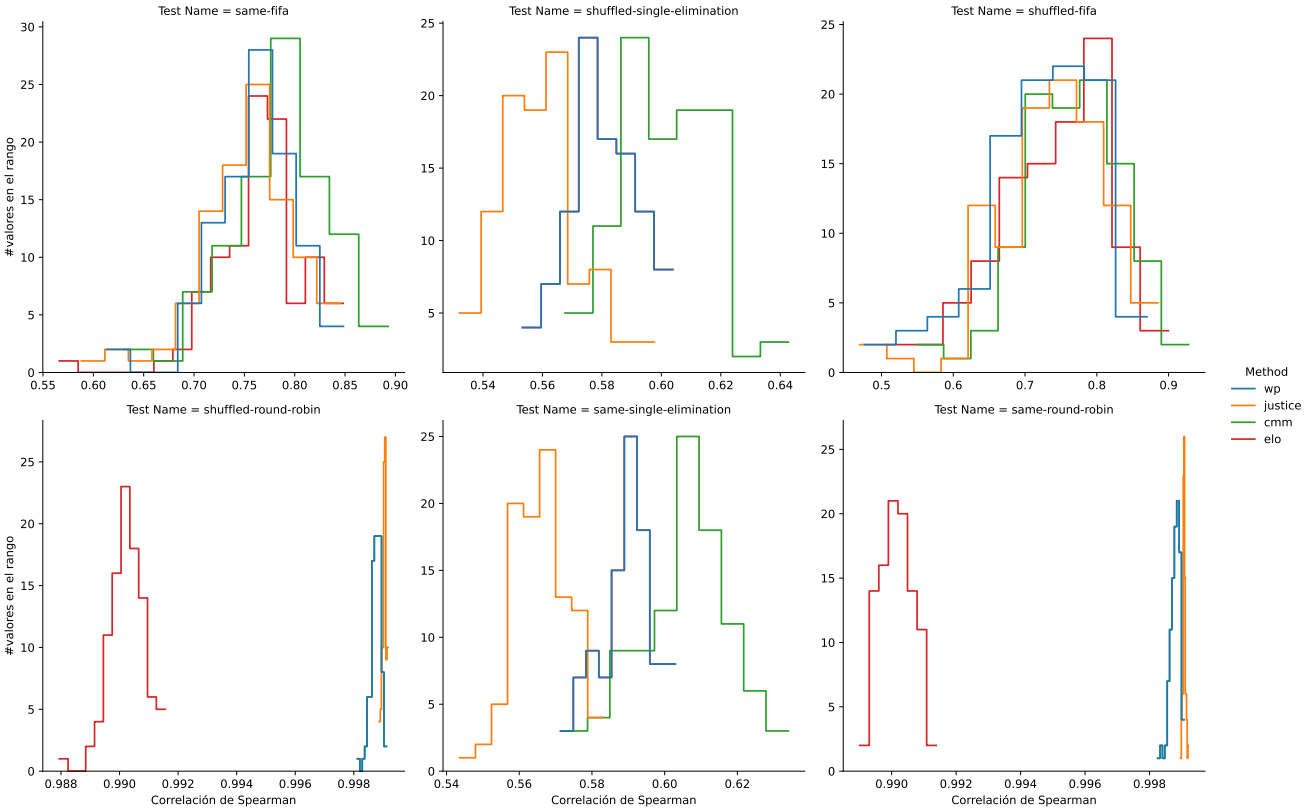

Figure 5: Histograma de la correlación de Spearman entre el ranking de `players.txt` y el generado por los métodos implementados para cada torneo generado.

En la figura 5 podemos notar que algunos métodos convergen fuertemente, dependiendo del tipo de torneo. Es éste el caso de CMM y WP que generan los mismos rankings para los torneos Round-Robin (`same-*` son casos de prueba que todos tienen el mismo fixture mientras que en `shuffled-*` se sortearon los equipos para cada torneo). Por otra parte, Elo y WP se comportan idénticamente en los torneos eliminación simple.

Es interesante notar también que los torneos de eliminación simple son los que menos correlación tienen (entre 0.55 y 0.65) lo cuál tiene sentido ya que al cada equipo jugar como **máximo** 10 partidos (recordemos son 1024 equipos) la dificultad del *schedule* juega un rol crucial. Por otra parte, los rankings generados para torneos Round-Robin encuentran una similaridad mucho más alta con el ranking original.

5.2.3 Rendimiento de los métodos ante torneos de eliminación directa

Sabemos entonces que *torneo-a-torneo* eliminación directa no nos ofrece un ranking demasiado similar al generado por nosotros. En la figura 6 podemos observar el comportamiento de nuestros métodos al ser utilizados sobre un torneo de eliminación simple. Los espacios sin puntaje son un defecto propio de la cantidad de partidos (no puede haber un WP de 0.25, por ejemplo).

Una pregunta razonable es si esto sigue ocurriendo al acumular información de múltiples torneos. Para esto es que generamos 100 torneos con el mismo *scheduling* y 100 torneos con distintos *schedulings*. Los gráficos correspondientes para éstos son 7 y 8 respectivamente.

5.2.4 Rendimiento de los métodos ante torneos Round-Robin

Finalmente, en la otra punta del espectro encontramos a los torneos Round-Robin. Éstos ofrecen una gran cantidad de información incluso aunque cada par de equipos se encuentre una única vez cómo. Esta observación puede notarse en la figura 9. Por la construcción misma del torneo podemos decir asigna igual dificultad de *schedule* a todos los equipos.

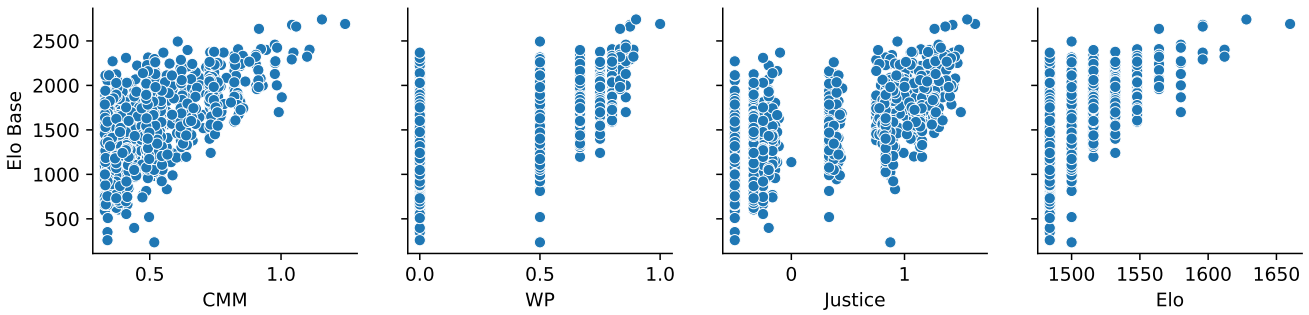

Figure 6: Comparación entre los puntajes Elo de referencia y los puntajes inferidos por los métodos implementados para la entrada `shuffled-single-elimination-3`.

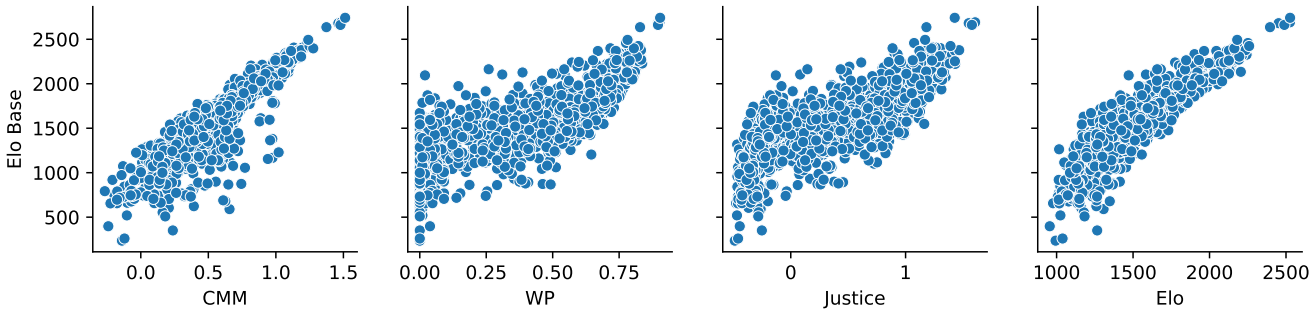

Figure 7: Comparación entre los puntajes Elo de referencia y los puntajes inferidos por los métodos implementados para la entrada `big-same-single`, generable con el objetivo `big-same-single.txt` de la carpeta de experimentación.

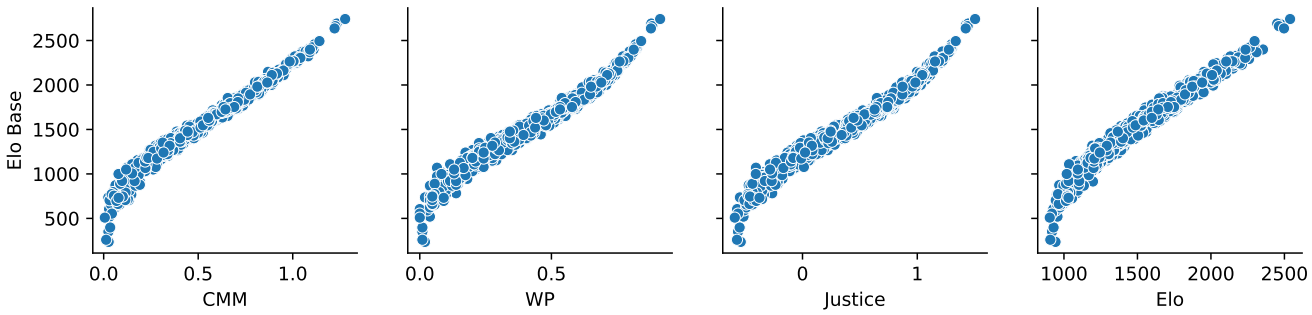

Figure 8: Comparación entre los puntajes Elo de referencia y los puntajes inferidos por los métodos implementados para la entrada `big-shuffled-single`, generable con el objetivo `big-shuffled-single.txt` de la carpeta de experimentación.

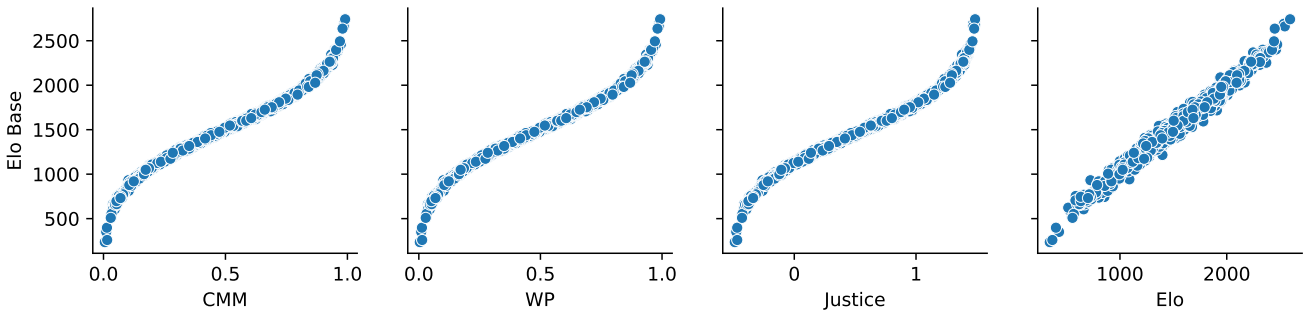

Figure 9: Comparación entre los puntajes Elo de referencia y los puntajes inferidos por los métodos implementados para la entrada `shuffled-round-robin-42`.

5.3 Comparación cualitativa con datos reales

En la presente sección, se utilizarán datos de competencias reales para compararlas con los resultados obtenidos de aplicar los distintos métodos a estos torneos con el fin de identificar y determinar características de los distintos algoritmos en función de los eventos ocurridos.

Para ello se eligieron 2 torneos, uno de ellos que consiste en enfrentamientos entre una determinada cantidad de equipos de distintas categorías, en donde luego los mejores equipos de dichas categorías pasan a instancias eliminatorias (NBA), y el otro es un torneo donde todos los equipos juegan contra todos la misma cantidad de partidos.

5.3.1 NBA

Primero se trabajara con el torneo de la NBA del año 2016. Dicha competencia consiste en principio de una instancia denominada Regular Season, donde cada uno de los 30 equipos tiene un total de 82 encuentros. Un importante detalle es que cada equipo se encuentra en una de las tres divisiones de 5 equipos que cada una de las dos conferencias tiene. Es decir que cada equipo disputará muchos más encuentros con equipos que pertenezcan a su división que con cualquier otro equipo dentro de su categoría que juegue en otra división. A su vez, cada equipo disputará más encuentros con equipos dentro de su propia categoría, que con aquellos que no.

Una vez finalizada la temporada, los mejores equipos de cada división pasan a la postemporada. Aquí se enfrentan "mejores" 8 equipos de cada conferencia en un torneo de eliminación directa.

El hecho de que no todos los equipos jueguen entre sí puede hacer que equipos "peores" terminen en posiciones más altas (por ejemplo, si la diferencia de nivel entre conferencias es muy pronunciada). Dado que cada método ajusta (o no) a la dificultad del schedule esto puede generar discrepancias entre los métodos de scoring.

En la Tabla 1 se puede ver el ranking de dicho torneo. El campeón fue Cleveland, siendo Golden State el equipo que salio en segundo lugar, Toronto en tercer lugar y Oklahoma el cuarto. Cuando dos equipos quedan descalificados en la misma instancia (por ejemplo en cuartos de final), el equipo que haya ganado más partidos en total, ocupará la posición mas alta.

En la tabla 2 se puede ver cuáles son los resultados para el mismo torneo según los métodos presentados.

Una de las observaciones mas interesantes es que todos los coinciden en un campeón (Golden State) que difieren del oficial. Esto se debe a que particularmente en dicho torneo, Golden State fue el equipo que ganó la mayor cantidad de partidos (y en consecuencia también el que menos perdió), sin embargo, perdió la final. Es decir, perdió el partido que lo coronaria campeón, frente a un rival (Cleveland) que previamente no había tenido un torneo tan bueno como él.

Generalizando estos conceptos, se puede ver que muchos equipos han variado también en sus posiciones finales. Esto nuevamente se debe a que, como bien se dijo previamente, ganar mas partidos no necesariamente implica obtener un mejor ranking en el torneo general, mientras que en la mayoría de los métodos como bien se explicó, no existe tanto el concepto de partidos mas importantes como si lo tiene el torneo.

Pos.	Equipo
1	Cleveland
2	Golden State
3	Toronto
4	Oklahoma
5	Miami
6	San Antonio
7	Portland
8	Atlanta
9	Indiana
10	Charlotte
11	LA Clippers
12	Boston
13	Houston
14	Dallas
15	Memphis
16	Detroit
17	Chicago
18	Utah
19	Washington
20	Orlando
21	Milwaukee
22	Denver
23	New York
24	Sacramento
25	New Orleans
26	Minnesota
27	Brooklyn
28	Phoenix
29	LA Lakers
30	Philadelphia

Table 1: Ranking oficial de la NBA, temporada 2016.

Pos.	Equipo	CMM
⇑1	Golden State	0.8741
⇑2	San Antonio	0.8040
⇓3	Cleveland	0.6924
⇓4	Toronto	0.6593
⇓5	Oklahoma	0.6555
⇑6	LA Clippers	0.6297
⇓7	Miami	0.5774
⇑8	Boston	0.5657
⇑9	Memphis	0.5632
⇓10	Atlanta	0.5588
⇓11	Charlotte	0.5582
⇓12	Indiana	0.5480
⇑13	Chicago	0.5160
⇓14	Portland	0.5159
⇓15	Houston	0.5113
⇓16	Dallas	0.5022
⇓17	Detroit	0.5002
18	Utah	0.4872
19	Washington	0.4837
⇑20	Milwaukee	0.4350
⇓21	Orlando	0.4322
22	Denver	0.4209
23	New York	0.4144
24	Sacramento	0.4128
25	New Orleans	0.3845
26	Minnesota	0.3200
⇑27	Phoenix	0.2989
⇓28	Brooklyn	0.2907
29	LA Lakers	0.2367
30	Philadelphia	0.1508

(a) Ranking de la NBA usando el método CMM.

Pos.	Equipo	WP
⇑1	Golden State	0.9091
⇑2	San Antonio	0.8507
⇓3	Cleveland	0.7121
⇓4	Toronto	0.6818
⇓5	Oklahoma	0.6716
⇑6	LA Clippers	0.6364
⇓7	Miami	0.5821
⇑8	Boston	0.5821
⇑9	Memphis	0.5821
⇓10	Atlanta	0.5672
⇓11	Charlotte	0.5606
⇓12	Indiana	0.5373
⇓13	Portland	0.5147
⇑14	Chicago	0.5077
⇓15	Houston	0.5075
⇓16	Dallas	0.5075
⇓17	Detroit	0.5075
18	Utah	0.4776
19	Washington	0.4697
20	Orlando	0.4394
21	Milwaukee	0.4265
22	Denver	0.4118
23	New York	0.4118
24	Sacramento	0.3940
25	New Orleans	0.3636
26	Minnesota	0.3134
27	Brooklyn	0.2836
28	Phoenix	0.2687
29	LA Lakers	0.2059
30	Philadelphia	0.1343

(b) Ranking de la NBA usando el método WP.

Pos.	Equipo	JUSTICE
⇑1	Golden State	1.290720
⇑2	San Antonio	1.159240
⇓3	Cleveland	0.909913
⇓4	Toronto	0.844214
⇓5	Oklahoma	0.830678
⇑6	LA Clippers	0.770065
⇓7	Miami	0.661968
⇑8	Boston	0.649385
⇑9	Memphis	0.644810
⇓10	Atlanta	0.628390
⇓11	Charlotte	0.620327
⇓12	Indiana	0.589049
⇓13	Portland	0.531111
⇑14	Chicago	0.526614
⇓15	Houston	0.520391
⇓16	Dallas	0.509195
⇓17	Detroit	0.507603
18	Utah	0.466122
19	Washington	0.456157
20	Orlando	0.369015
21	Milwaukee	0.360916
22	Denver	0.331447
23	New York	0.325531
24	Sacramento	0.306559
25	New Orleans	0.246910
26	Minnesota	0.128648
27	Brooklyn	0.069920
28	Phoenix	0.065462
29	LA Lakers	-0.061667
30	Philadelphia	-0.223182

(c) Ranking de la NBA usando el método JUSTICE.

Pos.	Equipo	ELO
⇑1	Golden State	1805.34
⇑2	San Antonio	1770.61
⇓3	Cleveland	1642.59
⇓4	Toronto	1641.53
⇓5	Oklahoma	1603.80
⇑6	LA Clippers	1594.04
⇑7	Charlotte	1578.85
⇑8	Boston	1574.77
⇓9	Miami	1569.95
⇓10	Portland	1564.21
⇓11	Atlanta	1556.18
⇓12	Indiana	1545.70
13	Houston	1538.43
⇑14	Memphis	1530.23
⇑15	Utah	1505.53
16	Detroit	1489.73
17	Chicago	1472.03
⇑18	Milwaukee	1471.16
19	Washington	1470.44
⇓20	Dallas	1463.20
⇑21	Denver	1441.76
⇓22	Orlando	1419.16
⇑23	New Orleans	1412.08
24	Sacramento	1405.15
⇓25	New York	1403.16
26	Minnesota	1362.17
27	Brooklyn	1333.39
28	Phoenix	1312.20
29	LA Lakers	1299.93
30	Philadelphia	1222.69

(d) Ranking de la NBA usando el método ELO.

Table 2: Rankings de la NBA 2015-2016 acorde a cada método.

5.3.2 Premier League

Otro torneo con el que se decidió trabajar es con el de la Premier League. Este consiste en 20 equipos que compiten entre si 2 veces, es decir, que cada equipo disputara un total de 38 encuentros.

Las formas de puntuación son bastantes sencillas. Dado un partido, el equipo que gane se lleva 3 puntos y el equipo que pierda suma 0. En caso de que haya un empate, cada equipo se lleva 1 punto.

Es decir, que se trata de un método bastante similar al de Winning Percentage, con la diferencia de que los empates reparten 1 punto a cada equipo, y que la victoria le otorga al ganador una suma de 3 puntos.

En este caso, se estará experimentando con el torneo de la temporada 2018/2019, en la cual, tal como puede observarse en la Tabla 3, el campeón fue Manchester City con un total de 98 puntos, mientras que el ultimo lugar se lo llevo Huddsersfield con 16 puntos.

En la Tabla 4 se pueden ver los ranking de cada equipo aplicado por los diversos métodos. Todos devolvieron diferencias con respecto al ranking obtenido por el método original. Esto se debe principalmente al hecho de que los métodos trabajados no se encargan en general de trabajar el caso de los empates por separado. Esto se puede observar en el hecho de que el campeón en los primeros 3 métodos difiera del campeón real, ya que este ultimo no fue el mayor ganador de la competencia, sino que los puntos otorgados por los empates lo hicieron posicionar en el primer lugar.

Otra observación interesante que se puede hacer es que, excepto algunas excepciones, el resultado obtenido por todos los métodos estudiados fueron bastante similar entre ellos. Esto se debe a que en este tipo de competencias, en donde no hay partidos mas importantes que otros en términos de puntos otorgados, y en donde todo equipo se enfrenta con cada uno exactamente en 2 ocasiones, se minimiza la variación del ranking.

Pos.	Equipo	Pts.
1	Manchester City	98
2	Liverpool	97
3	Chelsea	72
4	Tottenham	71
5	Arsenal	70
6	Manchester United	66
7	Wolves	57
8	Everton	54
9	Leicester	52
10	West Ham	52
11	Watford	50
12	Crystal Palace	49
13	Newcastle	45
14	Bournemouth	45
15	Burnley	40
16	Southampton	39
17	Brighton	36
18	Cardiff	34
19	Fulham	26
20	Huddersfield	16

Table 3: Ranking oficial de la Premier League.

Pos.	Equipo	CMM
⇧1	Liverpool	0.8809
⇩2	Manchester City	0.8571
⇧3	Arsenal	0.6429
⇩4	Chelsea	0.6190
⇩5	Tottenham	0.5952
6	Manchester United	0.5714
7	Wolves	0.5476
8	Everton	0.5238
⇧9	Newcastle	0.5238
⇩10	Leicester	0.5000
11	Watford	0.5000
⇩12	West Ham	0.4762
⇩13	Crystal Palace	0.4286
⇧14	Burnley	0.4286
⇩15	Bournemouth	0.3810
⇧16	Brighton	0.3571
⇩17	Southampton	0.3571
18	Cardiff	0.3333
19	Fulham	0.2619
20	Huddersfield	0.2143

(a) Ranking de la Premier League por CMM.

Pos.	Equipo	WP
⇧1	Liverpool	0.9210
⇩2	Manchester City	0.8947
⇧3	Arsenal	0.6579
⇩4	Chelsea	0.6316
⇩5	Tottenham	0.6052
6	Manchester United	0.5789
7	Wolves	0.5526
8	Everton	0.5263
⇧9	Newcastle	0.5263
⇩10	Leicester	0.5000
11	Watford	0.5000
⇩12	West Ham	0.4737
⇩13	Crystal Palace	0.4210
⇧14	Burnley	0.4210
⇩15	Bournemouth	0.3684
⇧16	Brighton	0.3421
⇩17	Southampton	0.3421
18	Cardiff	0.3158
19	Fulham	0.2368
20	Huddersfield	0.1842

(b) Ranking de la Premier League por WP.

Pos.	Equipo	JUSTICE
⇧1	Liverpool	1.362880
⇩2	Manchester City	1.297780
⇧3	Arsenal	0.860111
⇩4	Chelsea	0.817175
⇩5	Tottenham	0.749307
6	Manchester United	0.703601
⇧7	Newcastle	0.619114
⇩8	Wolves	0.601108
⇩9	Everton	0.578947
⇧10	Watford	0.569252
⇩11	Leicester	0.530471
⇩12	West Ham	0.483380
⇧13	Burnley	0.385042
⇩14	Crystal Palace	0.361496
⇩15	Bournemouth	0.292244
17	Southampton	0.225762
16	Brighton	0.210526
18	Cardiff	0.174515
19	Fulham	0.001385
20	Huddersfield	-0.096953

(c) Ranking de la Premier League por JUSTICE.

Pos.	Equipo	ELO
1	Manchester City	1736.82
2	Liverpool	1735.24
3	Chelsea	1580.62
⇧4	Arsenal	1557.36
⇧5	Manchester United	1547.19
⇧6	Wolves	1533.27
⇩7	Tottenham	1531.24
8	Everton	1525.89
⇧9	Crystal Palace	1520.12
10	West Ham	1505.36
⇩11	Leicester	1496.81
⇧12	Newcastle	1492.13
⇩13	Watford	1460.67
⇧14	Burnley	1453.09
⇧15	Southampton	1448.99
⇩16	Bournemouth	1435.27
⇧17	Cardiff	1397.82
⇩18	Brighton	1390.98
19	Fulham	1360.76
20	Huddersfield	1290.37

(d) Ranking de la Premier League por ELO.

Table 4: Rankings de la Premier League acorde a cada algoritmo.

5.4 Estrategias para *"burlar"* los métodos de ranking

Dada una lista de resultados y un método de ranking podemos preguntarnos. ¿Cuál es el mayor ranking alcanzable con sólo k victorias? Es éste el tema que exploraremos en esta sección y como tal esperamos definir primero las limitaciones de nuestro análisis:

1. Nos enfocaremos en torneos que tienen un *schedule* fijo.

   Esto deja sólo torneos Round-Robin para lo discutido hasta ahora, pero existen otros tipos de torneos dónde los partidos son decididos con antelación.

2. Nos restringiremos a cambiar sólo el los partidos del equipo en cuestión. No modificaremos partidos de terceros aunque cambiar aquellos resultados impacte en nuestro ranking.

3. No daremos métodos exactos.

5.4.1 Respuesta obvia

La respuesta más sencilla es "probar todas las combinaciones". Sea n la cantidad de partidos jugados y k la cantidad de partidos que nos permitimos ganar entonces podemos ver los $\binom{n}{k}$ casos y elegir el que ponga al equipo en el ranking más alto. Cómo este número puede ser muy grande ($\binom{100}{10}$ es un número mayor a 10^{13}) buscaremos formas de minimizar el espacio de búsqueda.

5.4.2 Búsqueda basada en "sampling"

La forma más sencilla de minimizar el espacio de búsqueda es ignorarlo. Podemos tomar muchas selecciones de k partidos al azar y quedarnos con la que mejore el ranking. Ésta idea está implementada en `optimize-wins.cpp` dónde tomamos wins al azar para cada cantidad de victorias posibles y registramos el mayor rank obtenido entre éstas. Los resultados pueden verse en la figura 10.

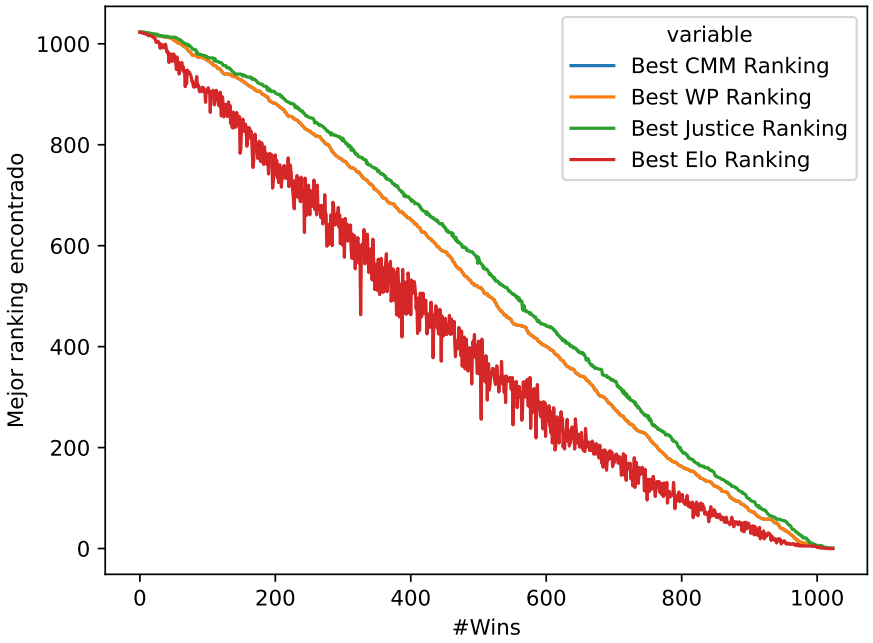

Figure 10: Comparación entre la cantidad de victorias y el mayor rank encontrado por parte del equipo 15 de `shuffled-round-robin-16`.

6 Conclusiones y trabajo futuro

En este trabajo presentamos cuatro métodos de scoring para competencias enfocados en ofrecer un ranking coherente para éstas. Discutimos sus propiedades y los evaluamos tanto en un ambiente sintético que nos ofrece un ranking de referencia cómo contra datos reales de múltiples competencias. Finalmente discutimos los primeros pasos a la hora de buscar maximizar el ranking para un método dado minimizando la cantidad de victorias.

Durante el desarrollo de este trabajo surgieron muchas más preguntas que no llegamos a desarrollar, éstas quedan cómo trabajo futuro:

- La matriz C en CMM es simétrica definida positiva, por lo que admite otros esquemas de factorización más precisos y eficientes. ¿Cómo afecta la precisión extra a los rankings? ¿Y al rendimiento de la implementación?
- Operar con números de coma flotante se puede hacer en distintos niveles de precisión. Trabajamos en posibilitar la compilación de nuestro código con tres de estos (números de 32, 64 y 80 bits/precisión extendida) y verificamos que pequeñas diferencias de ranking ocurrían entre éstos. ¿Cómo se contrastan estas diferencias en con los cambios de complejidad de usar otras representaciones de coma flotante?
- A la hora de intentar maximizar el ranking minimizando las victorias realizamos una búsqueda muy superficial sobre el espacio de posibilidades. ¿Cómo se compara ésta con una guiada? (por ejemplo utilizando un algoritmo genético)

7 Referencias y bibliografía

References

[1] Wesley N. Colley *Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained*, 2002.