Un algoritmo de búsqueda dispersa para el problema de torneos con equipos viajeros

Felipe A. Ramírez González[a]

[a]*Departamento de Informática*
Universidad Técnica Federico Santa María; Valparaíso, Chile.

Abstract

El problema de torneos con equipos viajeros (*Traveling Tournament Problem* (TTP), en inglés) es un problema de optimización que pertenece a la clase de problemas de planificación de actividades, donde el objetivo es minimizar la distancia que un conjunto de equipos debe recorrer para participar en los partidos de un torneo. En este trabajo se propone un nuevo enfoque para resolver el TTP basado en búsqueda dispersa (*scatter search* en inglés) con un componente dominante de búsqueda local provisto por *hill climbing* para la explotación y control de divergencia. Los resultados muestran que el algoritmo propuesto tiene un desempeño comparable a métodos de la literatura al resolver instancias de benchmark clásicas.

Key words: Scatter Search; Hill Climbing; Traveling Tournament Problem.

1. Introducción

En la actualidad los problemas de asignación de recursos y planificación de actividades han adquirido gran importancia para la inteligencia de negocios pues es un paso vital en el correcto funcionamiento de sistemas de producción alrededor del mundo. Los deportes como negocio no están fuera de esta categoría, ya que representan gran parte de la actividad económica alrededor del mundo, y en consecuencia la automatización de agendas deportivas para torneos de gran escala se ha vuelto necesaria para una vasta cantidad de deportes a nivel internacional. La importancia de una correcta planificación está dada por sus implicancias; desde la inversión en que los clubes tendrán que incurrir para el traslado del equipo técnico y los jugadores, hasta las variaciones en rendimiento de los mismos como consecuencia de las horas de viaje.

El problema de planificación de deportes se conoce en la literatura como *Traveling Tournament Problem* (TTP), y fue propuesto por Easton y otros en [8]. Este problema consiste en minimzar la distancia total que cierta cantidad par de equipos debe recorrer al participar en un torneo con modalidad *todos contra todos* doble. Este formato (conocido como *double round-robin* en inglés) es aquel empleado en la mayoría de los torneos modernos, y consiste en que todos los posibles pares de equipos deben enfrentarse dos veces en el torneo; una vez en calidad de local y otra vez de visita. Las soluciones al TTP son agendas para la realización de los partidos con el formato todos contra todos doble en una serie de rondas que minimiza la distancia viajada por los equipos, y donde la factibilidad de la agenda está sujeta a varias restricciones asociadas a la permanencia de los equipos en su ciudad o fuera de ella.

En cuanto al espacio de búsqueda del problema la cantidad de agendas posibles de generar explota exponencialmente con la cantidad de equipos asociados. Para un torneo de tan sólo 20 equipos la cantidad de agendas posibles es del orden de $10^{1}30$, sin embargo instancias reales del problema consideran más de 30 equipos, como la *National Football League* (NFL) de Estados Unidos que considera 32 equipos distribuidos de costa a costa. El TTP está en la categoría de problemas NP-difícil, por lo tanto, encontrar la agenda óptima (con métodos completos) para este tipo de instancias en tiempo polinomial es computacionalmente imposible. Por este motivo surge la necesidad de abordar el problema con enfoques incompletos que entreguen soluciones factibles en un tiempo razonable.

En este artículo se aborda el problema de torneos con equipos viajeros mediante una técnica evolutiva incompleta sin precedentes de aplicación en el área; la búsqueda dispersa (*scatter search* en inglés), y se encuentra estructurado como sigue. En la Sección 2 se formaliza el TTP y se enuncia la función objetivo a optimizar junto a las restricciones asociadas. En la Sección 3 se presenta la metodología seguida para construir el algoritmo propuesto y en la Sección 4 se presentan los resultados computacionales de experimentos para medir y comparar su rendimiento. En la Sección 5 se presenta una breve discusión bibliográfica y finalmente la Sección 6 concluye acerca del trabajo realizado.

2. Descripción del Problema

Como se adelantó en la introducción de este artículo, el *Traveling Tournament Problem* (TTP) consiste en planificar un torneo en modalidad *double round-robin*, donde un número par n de equipos (T) deben enfrentarse todos contra todos en dos oportunidades, una vez en calidad de local y otra de visita. Este formato requiere de $2(n-1)$ rondas (o fechas), en cada una de las cuales se juegan $n/2$ partidos. Una instancia del problema viene dada por la matriz de distancias D de tamaño $n \times n$, donde D_{ij} representa la distancia entre las ciudades donde los equipos T_i y T_j tienen calidad de local. La distancia recorrida por cada equipo en una planificación dada se calcula asumiendo que los equipos comienzan el torneo en sus ciudades y deben retornar una vez terminado.

El objetivo del problema es encontrar una planificación donde la distancia recorrida por todos los equipos sea mínima, sujeta adicionalmente a las siguientes restricciones:

- Un equipo no puede permanecer con un rol (local o visita) por más de tres rondas consecutivas.
- Un partido del equipo T_i en la ciudad del equipo T_j no puede estar junto a un partido de del equipo T_j en la ciudad del equipo T_i.

Una variación al TTP es el *Mirrored Traveling Tournament Problem* (mTTP), el cual refleja la realidad de los torneos Latinoamericanos. Esta variación consiste en que cada equipo juega una vez contra los otros en las primeras $n-1$ rondas, y luego en las últimas $n-1$ rondas se repiten los mismos partidos pero con los roles de local y visita invertidos. Cabe mencionar esta variación pues el algoritmo propuesto basa parte de su funcionamiento en la generación de soluciones para este caso particular del TTP.

3. Metodología

La metodología utilizada para resolver el problema descrito se basa en el algoritmo de Búsqueda Dispersa (Scatter Search) originalmente propuesto por Fred Glover en 1977 y recientemente adaptado [4] para resolución de problemas de optimización combinatoria. Este método, al igual que los algoritmos genéticos, tiene un enfoque evolutivo por lo que opera utilizando un conjunto de soluciones que *evolucionan* mediante procesos de combinación y búsqueda local. La estructura general del método se muestra en el Algoritmo 1.

Algorithm 1 Scatter Search

```
Generar soluciones
Mejorar soluciones con búsqueda local
Construir RefSet
repeat
  repeat
    Combinar soluciones
    Mejorar soluciones con búsqueda local
    Actualizar RefSet
  until combinacion no produce mejores soluciones o cantidad máxima de evaluaciones alcanzada
  Reconstruir RefSet
until cantidad máxima de evaluaciones alcanzada
```

El algoritmo comienza por generar un conjunto P de tamaño *popsize* de soluciones iniciales factibles y diversas con el método descrito en la Sección 3.2. A continuación estas soluciones son mejoradas aplicando un número determinado de iteraciones de búsqueda local como se especifica en la Sección 3.3. A partir de esta población inicial mejorada se construye el conjunto de referencia *RefSet*, el cual contiene b soluciones elegidas del conjunto P en dos fases. Primero se trasladan las mejores $b/2$ soluciones de P a *RefSet*. Luego para cada solución en *RefSet* se identifica la solución en P menos distante, donde la medida de distancia está dada por la suma de variables distintas entre ambas. Finalmente se traslada a *RefSet* aquella solución en P que es la más veces menos distance. El proceso se repite hasta que se hayan agregado las $b/2$ soluciones que faltaban. Notar que según lo anterior *popsize* debe ser al menos tan grande como b.

Luego, en el bucle principal del algoritmo, se combinan todos los posibles pares de soluciones en *RefSet* de acuerdo al operador especificado en la Sección 3.4. La combinación exhaustiva de soluciones genera un nuevo conjunto *Offspring* de soluciones factibles, el cual es mejorado mediante búsqueda local. El conjunto de referencia es actualizado seleccionando las b mejores soluciones de los conjuntos *RefSet* y *Offspring*. Se itera desde la combinación hasta que ya no haya soluciones seleccionadas desde *Offspring*.

Cuando se cumple que la combinación optimizada no produce mejores soluciones se reconstruye el *RefSet* reemplazando sus $b/2$ peores soluciones por las mejores encontradas en una nueva población P mejorada. Se vuelve a iterar desde la combinación hasta que se alcanza un número determinado de llamadas a la función de evaluación.

3.1. Representación

Las soluciones al problema están representadas por una matriz S de $2(n-1) \times n$, donde n es la cantidad de equipos que participan del torneo. El dominio de cada casilla S_{ij} es el intervalo de enteros $[-n, n]$ que excluye el 0, donde la instanciación $S_{ij} = k$ indica que en la ronda i, el equipo j juega de local contra el equipo k si $k > 0$ o de visita si $k < 0$. Según esta representación, la instanciación anterior impone que $S_{ik} = -j$, ya que la planificación de un partido define variables para dos equipos. La consistencia de este tipo de información redundante está garantizada por el diseño de las heurísticas utilizadas en el algoritmo.

3.2. Soluciones iniciales

Inspirado por el trabajo de los autores en [2], las soluciones iniciales de la población P son generadas mediante el Método del Polígono, el cual utiliza una representación compacta para generar soluciones factibles en conjunto con una heurística de asignación de roles. Este método construye soluciones para el mTTP, sin embargo estas pueden ser igualmente utilizadas como punto de partida para encontrar soluciones al TTP.

Sea C, arreglo de tamaño n, la representación compacta de una potencial solución, donde cada casilla C_i está asociada a un equipo participante en un orden arbitrario. Se define al equipo base como el primer elemento del arreglo, y se construye una matriz de decodificación de tamaño $n \times (n-1)$ donde en cada fila se copia el arreglo C con un desplazamiento hacia la izquierda pero conservando el equipo base. Las filas de la matriz de decodificación entregan información acerca de los equipos que se deben enfrentar en cada ronda; el equipo base se enfrenta al equipo de la posición 2 y los equipos de las posiciones $i = 3, ..., (n/2)+1$ se enfrentan a los equipos de las posiciones $n-i+3$. Sólo se definen la mitad de rondas del torneo ya que la otra mitad se genera repitiendo la misma secuencia de partidos pero con roles intercambiados. Finalmente los roles local/visita

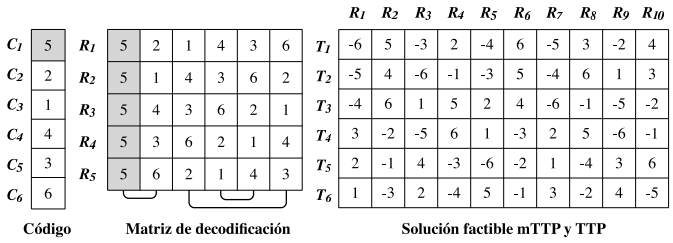

Figura 1: Decodificación de la representación compacta y generación de soluciones factibles mediante el Método del Polígono.

son asignados mediante una heurística aleatoria alternante para asegurar que las restricciones de permanencia no sean violadas. Un ejemplo de este procedimiento puede verse en la Figura 1.

Las soluciones iniciales son creadas aplicando este método a códigos generados aleatoriamente, lo que genera una población altamente diversa (considerando las posibles configuraciones del arreglo compacto y la aleatoriedad de la asignación de roles) y al mismo tiempo factible.

3.3. Búsqueda local

La búsqueda local utilizada en este algoritmo consiste en generar el vecindario completo a una solución para cierto movimiento y elegir la mejor, lo que en la literatura se conoce como *Hill Climbing* con selección de la mejor mejoría.

El movimiento a utilizar en cada iteración de la búsqueda se elige con probabilidad uniforme de tres posibles, los cuales se enuncian y describen a continuación.

- **Intercambio de Equipos**

  Este movimiento intercambia el itinerario de dos equipos T_i y T_j, es decir, la secuencia de partidos que originalmente jugaba T_i es jugada por T_j. En la representación utilizada este movimiento se realiza intercambiando los valores de las casillas pertenecientes a las filas i y j de la matriz S donde T_i y T_j no juegan contra si mismos. Como consecuencia además se intercambian las variables con información redundante para mantener la consistencia.

- **Intercambio de Roles**

  Este movimiento se aplica sobre un partido en particular, intercambiando los roles local/visita de los dos partidos asociados a un par de equipos y los respectivos cambios en las variables con información redundante.

- **Intercambio de Rondas Completas**

  Este movimiento se aplica intercambiando dos columnas de la matriz S. En la mayoría de los casos este movimiento sólo genera soluciones TTP-factibles, ya que viola inmediatamente las restricciones definidas para el mTTP bajo esta representación.

Se debe notar que la aplicación de estos movimientos puede generar soluciones infactibles, por lo tanto antes de evaluar el efecto de los mismos sobre una solución, se comprueba que la solución generada sea factible, en caso contrario se descarta.

3.4. Combinación

Dada la cantidad de dependencias y redundancia entre variables de la representación utilizada, la combinación de soluciones se realiza aplicando una transformación que represente cromosomas más simples, haciendo referencia a la metáfora de los algoritmos genéticos. La forma compacta representante de una solución corresponde a la primera columna de su matriz S asociada, a lo que se denominó *firma de una solución*. Por ejemplo la firma de la solución de la Figura 1, es el vector $\{6, 5, 4, 3, 2, 1\}$.

La combinación se realiza copiando los primeros $n/2$ elementos de la firma del padre con mejor función objetivo y copiando los elementos restantes en el orden que aparecen en el segundo padre. Por ejemplo si las firmas de los padres son $F_a = \{2, 1, 3, 4, 6, 5\}$ y $F_b = \{6, 1, 5, 3, 4, 2\}$ y el valor de la función objetivo del padre a mejor que la de b, entonces la firma del hijo será $F_h = \{2, 1, 3, 6, 5, 4\}$.

Una vez calculada la firma del hijo, esta se trata como un código del Método del Polígono y a partir de ella se genera una nueva solución factible siguiendo el método descrito en detalle en la Sección 3.2.

4. Resultados

El algoritmo fue escrito en C++ y ejecutado en un Intel Core Duo a 1.83 Ghz con 1.5 GB de memoria RAM, para resolver varias instancias del problema descritas en la literatura. En esta comparación de resultados tanto los obtenidos con el algoritmo propuesto como los extraídos de la literatura corresponden a soluciones restringidas del problema, tal como se planteó en la Sección 2. Las instancias utilizadas se describen con mayor detalle en [8].

Los parámetros del algoritmo tamaño de la población, tamaño del *RefSet*, máximo número de evaluaciones y la cantidad de iteraciones de búsqueda local fueron elegidos mediante ensayo y error hasta encontrar un equilibrio en el compromiso velocidad de cómputo - calidad de la solución.

El Cuadro 1 muestra los resultados obtenidos con el algoritmo propuesto junto a la cota inferior y la mejor solución encontrada para cada instancia. Se exhibe además la distancia porcentual entre la solución hallada y la mejor solución existente junto al tiempo de ejecución para cada caso.

Los resultados revelan que la metodología de búsqueda dispersa tiene un desempeño comparable a los métodos utilizados en la literatura para ciertas instancias. En otros casos no se logró tener buenos resultados, lo que se puede atribuir a algunas deficiencias del algoritmo. Principalmente pueden deberse al patrón fijo que siguen las soluciones generadas mediante el Método del Polígono, que considera claves con un espacio de búsqueda notablemente más reducido que el de las soluciones. Además los movimientos propuestos son simples y de grano grueso, esto quiere decir que los saltos en el espacio de búsqueda son grandes y como consecuencia la probabilidad de visitar ciertas soluciones es bastante baja. Por estos motivos el algoritmo podría moverse alrededor de un valle en el espacio de búsqueda pero nunca llegar al punto más bajo.

Otro aspecto a notar es que, dada la arbitrariedad de la adaptación de soluciones a representación compacta con el sistema de firmas, pareciera ser que la combinación no funciona correctamente como proceso de preservar *genes* que se traducen en buenas soluciones. Esto se observa al aumentar el número máximo de evaluaciones de la función objetivo y fijar la cantidad de iteraciones de búsqueda local, lo que no provoca mejoras en la calidad de las soluciones con respecto a la parametrización original. Este fenómeno da evidencia de que la búsqueda local se lleva gran parte del trabajo en la convergencia a un óptimo.

En cuanto al comportamiento del tiempo de ejecución del algoritmo como función del tamaño de la instancia, se puede estimar que el crecimiento sigue una tendencia lineal. Esta propiedad es característica de los métodos incompletos, en comparación a técnicas completas que por lo general siguen una tendencia exponencial como función de la complejidad de la instancia. Además la implementación del algoritmo pudo resolver algunas instancias del problema con una solución no muy lejana a las del trabajo en [2] y en un menor tiempo.

5. Trabajo Relacionado

El reto científico que impone el TTP para las comunidades de investigación de operaciones sumado a sus aplicaciones concretas en el mundo del deporte a gran escala han significado la aparición de una vasta cantidad de literatura. Varios autores [1, 5, 6, 9] han abordado el problema en el contexto de diferentes deportes como fútbol, basquetbol, hockey, béisbol, rugby y fútbol americano, utilizando diversas técnicas de optimización combinatoria, como búsqueda tabú, algoritmos genéticos y simulated annealing. Uno de los trabajos que destaca es el de Pascal Van Hentenryck y otros [7], quienes abordaron el problema con un algoritmo simulated annealing con enfoque evolutivo basado en poblaciones, obteniendo varios de los mejores resultados conocidos para una gran cantidad de instancias. Otro trabajo destacado es el de los autores en [3], quienes utilizaron búsqueda tabú para resolver instancias de la National Football League de Estados Unidos, obteniendo algunos de los mejores resultados hallados para esas instancias.

6. Conclusiones y Trabajo Futuro

En este trabajo se ha presentado el problema de torneos con equipos viajeros (TTP) y se ha propuesto una técnica evolutiva incompleta basada en el algoritmo de búsqueda dispersa apoyado por un componente de búsqueda local para su resolución.

Los resultados son comparables a los de la literatura en algunos casos, sin embargo hay instancias en las que el algoritmo tiene un mal desempeño. Esto se atribuyó a deficiencias de diseño de la propuesta, principalmente por la arbitrariedad de la representación compacta propuesta para la fase de combinación de soluciones. A raíz de lo expuesto se puede concluir que la representación utilizada en este trabajo no es adecuada para aplicar enfoques evolutivos, dada el alto grado de redundancia e interdependencia entre las variables.

Trabajo futuro puede comprender el diseño de heurísticas generadoras de soluciones iniciales más variadas y al mismo tiempo factibles. Además introducir una nueva representación que permita diseñar movimientos con vecindarios más finos y al mismo tiempo favorezca la aplicación de operadores de combinación, para permitir el recorrido de una mayor parte del espacio de búsqueda y obtener descendencia convergente de una generación a otra.

Finalmente el algoritmo requiere de una parametrización más acabada y sistemática para optimizar tanto el uso de recursos computacionales como la calidad de la solución entregada.

Instancia	**LB**	**BK**	**SS**	**Gap**	**CPU(s)**
Super4	63405	63405	63405	0.0 %	14.7
Super6	130365	130365	130437	0.1 %	22.0
Super8	182409	182409	269036	32.2 %	31.3
Super10	316329	316329	459811	31.2 %	42.3
Super12	367812	463876	618260	25.0 %	54.6
Super14	467839	571632	986763	42.1 %	70.9
NL4	8276	8276	8276	0.0 %	14.9
NL6	22969	23978	24085	0.4 %	22.2
NL8	38760	39721	47387	16.2 %	31.9
NL10	56506	59436	79510	25.2 %	42.1
NL12	107483	110729	146342	24.3 %	58.6
NL14	182797	188728	289641	34.8 %	75.3
NL16	248852	261687	434068	39.7 %	98.6
Circ4	20	20	20	0.0 %	14.3
Circ6	64	64	68	5.9 %	21.7
Circ8	128	132	166	20.5 %	31.0
Circ10	220	242	324	25.3 %	46.0
Circ12	384	404	558	27.6 %	55.1
Circ14	588	632	970	34.8 %	77.0
Circ16	832	916	1552	41.0 %	91.7
Circ18	1188	1294	2304	43.8 %	117.6
Circ20	1600	1732	3244	46.6 %	147.0
Galaxy4	416	416	416	0.0 %	15.03
Galaxy6	1365	1365	1396	2.2 %	23.0
Galaxy8	2373	2373	2891	17.9 %	38.3
Galaxy10	4443	4535	5869	22.7 %	50.1
Galaxy12	7034	7197	9683	25.7 %	76.0
Galaxy14	10255	10918	15801	30.9 %	102.4
Galaxy16		14900	23399	36.3 %	123.3
Galaxy18		20907	32455	35.6 %	164.1
Galaxy20		26289	42816	38.6 %	207.8
Galaxy22		35767	58234	38.6 %	163.2
Galaxy24		45910	76964	40.3 %	186.2
Galaxy26		60962	101478	39.9 %	216.9
Galaxy28		77577	133072	41.7 %	255.5
Galaxy30		96979	170194	43.0 %	283.5
Galaxy32		120683	214300	43.7 %	329.3
Galaxy34		147835	265796	44.4 %	424.2
Galaxy36		173827	322733	46.1 %	412.6
Galaxy38		210787	390798	46.1 %	468.5
Galaxy40		249230	474883	47.5 %	485.8

Cuadro 1: Comparación de resultados para varios conjuntos de instancias. *LB* es una cota inferior para la instancia, *BK* la mejor solución factible encontrada y *SS* la solución hallada por el algoritmo propuesto. *Gap* es la diferencia porcentual entre *SS* y *BK*. *CPU(s)* es el tiempo de ejecución del algoritmo en segundos.

Referencias

[1] Bryan C. Ball and Dennis B. Webster. Optimal scheduling for even-numbered team athletic conferences. *AIIE Transactions*, 9(2):161–169, 1977.

[2] Fabrcio Lacerda Biajoli, Luiz Antonio, Nogueira Lorena, Pesquisas Espaciais Inpe, Laboratrio Associado, Computao Matemtica, and Aplicada Lac. N.: Mirrored traveling tournament problem: An evolutionary approach. In *Lecture Notes in Artificial Intelligence Series*, pages 208–217, 2006.

[3] Luca Di Gaspero and Andrea Schaerf. A composite-neighborhood tabu search approach to the travelling tournament problem. *Journal of Heuristics*, 13:189–207, 2007.

[4] Fred Glover, Manuel Laguna, and Rafael Mart. Scatter search. In *Advances in Evolutionary Computing: Theory and Applications*, pages 519–537. Springer-Verlag, 2003.

[5] George L. Nemhauser and Michael A. Trick. Scheduling a major college basketball conference. *Oper. Res.*, 46(1):1–8, 1998.

[6] Jan A. M. Schreuder. Combinatorial aspects of construction of competition dutch professional football leagues. *Discrete Appl. Math.*, 35(3):301–312, 1992.

[7] Pascal Van Hentenryck and Yannis Vergados. Population-based simulated annealing for traveling tournaments. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 267–272. AAAI Press, 2007.

[8] Toby Walsh, Kelly Easton, George Nemhauser, and Michael Trick. *The Traveling Tournament Problem Description and Benchmarks*, volume 2239, pages 580–584. Springer Berlin / Heidelberg, 2001.

[9] M. B. Wright. Scheduling fixtures for basketball new zealand. *Comput. Oper. Res.*, 33(7):1875–1893, 2006.