Simulación de campeonato de Speedcubing mediante un modelo estocástico

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires

Abstract
El presente trabajo busca analizar el efecto que tiene la modificación de parámetros típicos en la organización de un torneo de Speedcubing sobre el tiempo de espera promedio para sus competidores. Se ha realizado un estudio sobre un modelo estocástico que representa la dinámica de competición, con N puestos fijos, R puestos rotativos, y dos colas concatenadas. Los resultados muestran que existe un número óptimo R, con N y los demás parámetros fijos, para el cual los valores a optimizar se acercan al mínimo asintótico.

Palabras Clave
Simulación, modelo estocástico, dinámica de juegos, Speedcubing, sistema de colas, puestos de atención

Introducción

Este trabajo se ha realizado en el marco de la asignatura "Simulación" en la carrera de Ingeniería en Sistemas de la UTN FRBA. Se propone estudiar el comportamiento de un sistema de dominio conocido mediante una simulación, para analizar los efectos que los cambios de las variables exógenas de control tienen sobre aquellos resultados endógenos que sean de interés.

En el presente se desarrolla un modelo basado en un torneo de Speedcubing, el cual consta de número de estaciones fijas, compuestas por una mesa y un juez, en las que los competidores realizan sus soluciones. Estos aguardan en una zona llamada "zona de espera" el llamado de una persona con el rol de "runner", cuya función principal es llevar un cubo mezclado desde la zona de mezclas, en la cual una persona dedicada entrega un cubo con una mezcla particular cada cierto lapso, hacia la zona de competidores, para ubicar al competidor dueño del puzzle en cualquier estación que se encuentre libre, ni bien haya una.

Se identifican dos variables de control como la cantidad de estaciones y cantidad de runners a definir, número el cual los organizadores de estos campeonatos habitualmente tienen que definir en base a su experiencia previa, a modo de conseguir que el tiempo promedio entre llamados de un runner, y los porcentajes de tiempo ociosos de las estaciones y los runners sean todos mínimos.

Esto es de suma relevancia en este tipo de campeonatos, ya que el hecho de que la espera en la zona de competidores previa al llamado del runner no sea muy larga es de vital importancia para los participantes de este tipo de torneos. Asimismo, los recursos son limitados, por lo que no se puede contar con una cantidad muy grande de runners ni estaciones por limitaciones de presupuesto, espacio físico, o de personal. La adición de cada uno de estos elementos a la organización de una competencia representa una decisión y debería estar basada en datos concretos para evitar el desperdicio de los recursos disponibles.

Elementos del Trabajo y metodología

Se ha planteado un modelo siguiendo la metodología de evento a evento para la simulación de sistemas.

A continuación, se incluye una descripción detallada de la mecánica de este tipo de torneos, y las consideraciones puntuales para tener en cuenta.

En la vida real, una ronda comienza cuando todos los participantes inscriptos dejan su cubo resuelto, identificado con su nombre en la zona de mezclas, y se dirigen hacia la zona de espera, todos juntos. En este momento, las *N* estaciones se encontrarán vacías, cada una con un juez y un cronómetro esperando a ser utilizadas.

En la zona de mezclas, los *scramblers* dedicados aplicarán una secuencia de mezclado generada por computadora a cada cubo resuelto que tengan a su disposición, cubo que será trasladado desde esta zona hacia la zona de competidores mediante un *runner* disponible.

El runner, al llegar a esta zona, puede ver que hay una estación libre, en cuyo caso llamará al competidor correspondiente para sentarlo e iniciar la solución del puzzle, o bien ver que todas las estaciones están ocupadas, evento ante el cual tendrá que esperar a que se libere una.

Cuando el competidor termina su resolución (que consiste en la inspección del puzzle, el comienzo del intento, el armado del cubo, y el registro del resultado final por parte del juez), habrá finalizado su tiempo de uso de la estación y se dirigirá nuevamente hacia la zona de competidores.

Un runner que esté volviendo desde la zona de competidores hacia la zona de mezclas llevará su cubo de regreso, a modo de que el competidor pueda realizar sus cinco (5) intentos según el formato de la WCA [1].

Dado que una simulación de estas características debería correr por un lapso largo simulado a modo de aproximarse más a su comportamiento real, este número de intentos es algo que será modificado por un número órdenes de magnitud mayor.

El intervalo entre mezclas generadas es conocido y se aproxima como una función de densidad de probabilidad uniforme entre los valores de 5 y 10 segundos. Asimismo, el tiempo que demora un runner en recorrer el trayecto entre la zona de mezclas y la zona de competidores se considera conocido y constante, dependiendo de las dimensiones físicas del salón del torneo.

Como hipótesis de simplificación se plantea que los runners solamente pueden trasladar de a un cubo a la vez, y que pueden traer de regreso cualquier cantidad de cubos.

Debido a esto, la simulación es equivalente a una en la que los competidores se retiran del recinto al finalizar su intento, y en la que llegan competidores nuevos dejando su cubo en la zona de mezclas cada un intervalo de tiempo igual al intervalo entre mezclas que conocemos.

El tiempo de solución, que representa el tiempo por el que un competidor ocupa una estación, se puede conocer y varía según los inscriptos del torneo. Según un torneo reciente, se estima este tiempo como una función de densidad de probabilidad definida como dos rectas por tramo en el intervalo [30;60] ∪ (60; 90], en el cual la probabilidad del valor 30 es la mitad del valor 60 y a su vez igual al 150% del valor en 90.

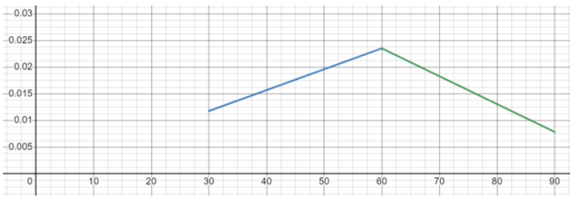

Imagen 1: Función de densidad de probabilidad del tiempo de solución

Para representar las funciones de densidad de probabilidad de manera programática utilizamos la técnica de la función inversa para *IM*, y el método de aceptación y rechazo para la función de *TS*.
No se tomaron muestras de la realidad debido a restricciones temporales, aunque las distribuciones planteadas reflejan la realidad de una manera suficientemente aproximada.
Como variables del sistema identificamos:

Control	**N**: número de estaciones **R**: número de runners
Datos	**IM**: intervalo entre mezclas, segundos **TS**: tiempo de solución, segundos
Estado	**NC**: cubos encolados, mezclados **NS**: competidores en el sistema
Resultados	**TPE**: tiempo promedio entre llamados del runner **PTOe**: promedio de tiempo ocioso de estaciones **PTOr**: promedio de tiempo ocioso de runners

Tabla 1: Variables del modelo

Es importante entender las variables de estado. En nuestro modelo, *NC* puede pensarse como la variable de estado típica de un modelo de *N puestos, 1 cola*, que representa la cantidad de sujetos encolados, así como la cantidad de sujetos que están siendo atendidos. Podemos conocer la cantidad de puestos libres simplemente comparando esta variable con el número fijo de puestos que conocemos que existen.
Por otro lado, *NS* también se puede interpretar como una variable basada en la misma idea.
Son entonces estas dos variables las que representan el estado de las dos colas concatenadas que se identifican en nuestro modelo: la cola de cubos mezclados esperando a ser atendidos por un runner, y la cola de runners esperando a ser atendidos por una estación.

En el caso de *NC*, los cubos pueden estar en la mesa de mezclas o en la mano de un runner caminando hacia la zona de competidores. Una vez se llegue allí, el cubo se cambiará de *NC* a *NS*, indicando que el runner está esperando a tener una estación liberada para llevar al competidor que corresponda. Finalmente, *NS* solo decrementará su valor en 1 cuando el competidor termine su solución.

Con esto en mente, identificamos los siguientes tres eventos:

Evento	EFNC	EFC	Condición
Mezcla	Mezcla	Runneada	NC <= R
Runneada	-	Solución	NS <= N
		Runneada	NC >= R
Solución	-	Solución	NS >= N

Tabla 2: Tabla de eventos independientes

Es entonces el evento de *Mezcla* aquel que sucede cuando un mezclador termina de aplicar una mezcla, evento independiente que se concatena mediante la función de densidad de probabilidad que llamamos *IM*; el evento *Runneada* el que representa a un runner llegando a la zona de competidores y encolándose para esperar la liberación de una estación; y *Solución* el evento que indica la finalización de uso de una estación por parte de un competidor.

El único evento concatenado es el de *Mezcla*, ya que los otros se generan condicionalmente sobre el vector de estado.
Con esa tabla como base obtenemos una tabla de eventos futuros como la siguiente:

Variable	Descripción
TPM	Tiempo de próxima mezcla, concatenado mediante la f.d.p. *IM*
TPR	Tiempo de próxima runneada, por cada runner (arreglo)
TPS	Tiempo de próxima solución, es decir, tiempo de liberación, por cada estación (arreglo)

Tabla 3: Tabla de eventos futuros

Partiendo con todas las variables distintas de TF inicialmente en cero (0), con el siguiente diagrama de flujo planteado, el sistema comenzará con un evento *Mezcla* y ejecutará su simulación por un lapso arbitrariamente grande de tiempo, que definimos en 3155760, el valor aproximado en segundos de un año.

Luego, es posible realizar la simulación y obtener los valores de las variables de resultados que deseamos conocer.

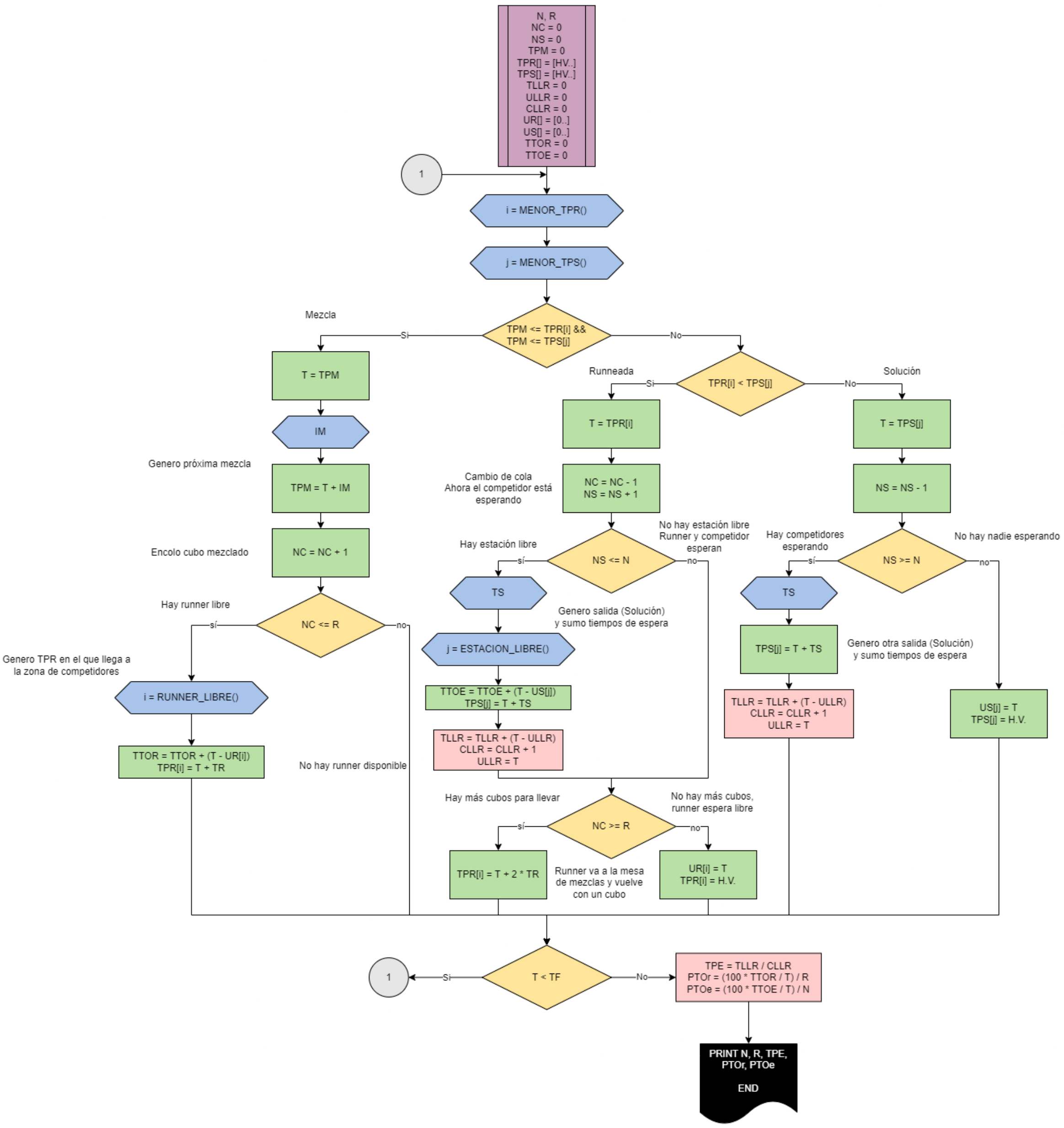

Imagen 2: Diagrama de flujo

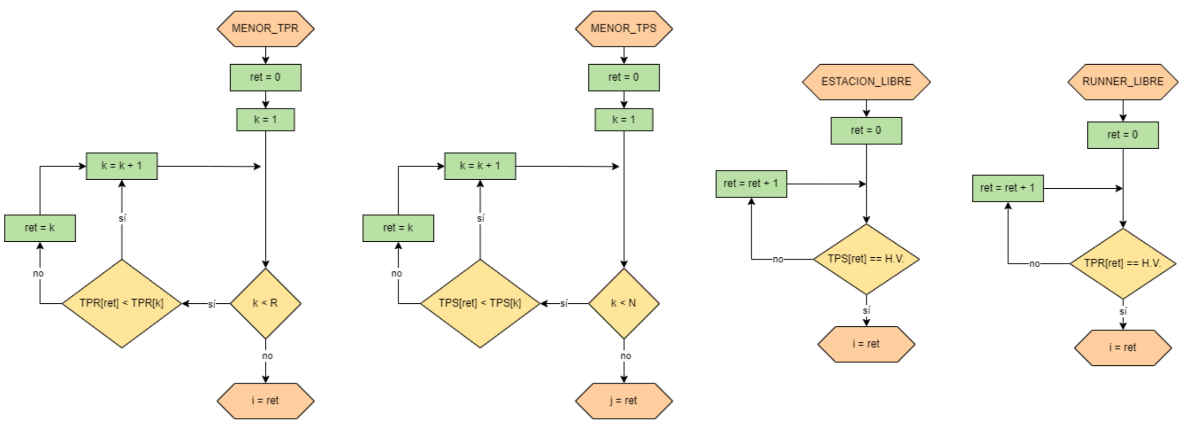

Imagen 3: Diagrama de flujo de las subrutinas

Resultados
Se han planteado condiciones iniciales de la simulación fijando el valor de *N* y la constante *TR* (tiempo de runneada) en magnitudes que se correspondan con las dimensiones de un salón conocido, y variando el valor de *R* (cantidad de runners) para conocer su valor óptimo.

Todas las demás variables comienzan en cero (0) a excepción de los arreglos en la tabla de eventos futuros, que comienzan con el valor máximo en todos sus elementos, representando así que se encuentran vacías todas las estaciones y libres todos los runners.

Al ejecutar estas simulaciones [2], podemos observar un comportamiento interesante.
La cantidad de runners, iniciada en uno (1), da lugar a un porcentaje de tiempo ocioso muy bajo de los mismos, prácticamente del 0%.
Conforme aumenta la cantidad de runners, se nota un decrecimiento en el promedio de tiempo entre llamados del runner, y un aumento del porcentaje de tiempo ocioso del runner muy pequeño. El porcentaje de tiempo ocioso de las estaciones también decrementa.

Sin embargo, llegado un número de runners, se obtiene el valor asintóticamente mínimo del porcentaje de tiempo ocioso de las estaciones, así como el mínimo tiempo entre llamados del runner, disparándose asimismo el porcentaje de tiempo ocioso de los runners.

Una vez alcanzado este número, aumentar la cantidad de runners tendrá efectos despreciables sobre los resultados distintos al porcentaje de tiempo ocioso de los runners, magnitud la cual aumentará indefinidamente.

La interpretación en la realidad de este comportamiento es que existe un número de runners con el cual las estaciones se utilizan de la mejor manera posible, logrando una coincidencia entre el tiempo de rotación de los runners y el tiempo promedio de ocupación de las estaciones considerando el intervalo entre mezclas. Una vez superado este umbral, tendremos runners de sobra que no estarán realizando ninguna tarea; simplemente observando cómo los *Ropt* runners realizan su trabajo y no necesitan ninguna ayuda para hacerlo mejor. Debido a esto, el tiempo ocioso se dispara.

Discusión
Vemos que existe una correlación entre la cantidad de runners y el cociente entre el *TR* y el valor medio de *IM*. Se interpreta esto como que, si la mesa de mezclas genera una mezcla cada un valor constante igual al valor medio de *IM*, serán suficientes cerca de 1+*TR/IM* runners para funcionar coordinadamente, redondeado hacia el mayor entero más cercano.

El cociente en esa expresión representa la cantidad de mezclas nuevas esperadas a ser generadas en el período de tiempo transcurrido entre que un runner toma su cubo y llega a la zona de espera (duración igual a *TR*). Deberían, entonces, existir esa cantidad de runners en su ausencia (de ahí la suma de 1) para poder manejar la cantidad de cubos esperada a ser generadas.

Por lo tanto, se satura la cantidad de runners con tener uno más que esa cantidad, y en ese punto ese runner de sobra es el que hace de “ventana”, haciendo las veces de un transporte instantáneo de un runner desde la zona de espera hacia la zona de mezclas, simplemente cambiando el tiempo ocioso de uno por el otro. El lector debe comprender que los runners necesitan trasladarse de regreso hacia la zona de mezclas, y este tiempo podría aprovecharse mejor si ya se tuviera un runner en esa zona, tomando la tarea del que está de camino antes de que ese llegue.

De ahí surge que en las simulaciones el valor de 2+*TR/IM* redondeado al siguiente entero es el punto de quiebre en el que se consigue el mínimo valor del tiempo promedio de espera entre llamados de runner, aproximadamente igual a la esperanza de *IM*, el mínimo porcentaje de tiempo ocioso de estaciones. El valor mínimo del porcentaje de tiempo ocioso de los runners ocurre con un runner menos, pero a costo del incremento de las otras dos variables de resultado.

Conclusión

Se extrae como conclusión la existencia de un valor óptimo de runners, *ceteris paribus*, para minimizar el porcentaje de tiempo ocioso de las estaciones, así como el promedio entre llamados.

No se consigue minimizar las tres variables de resultado en simultáneo, sino que el organizador que utilice esta simulación deberá analizar el balance entre la minimización del porcentaje de tiempo ocioso de los runners o bien la minimización de todo lo demás.

Se escogerá la segunda alternativa, pues es la solución que un organizador de un evento de Speedcubing considerará óptima, llevando al mínimo las variables de resultado que tienen un impacto en la percepción de los competidores sobre el campeonato, cediendo un lugar “extra” en las plazas de staff a modo de conseguir este resultado.

Agradecimientos
Se agradece a la cátedra de Simulación de la UTN FRBA por proveer del marco teórico y práctico para la realización del presente trabajo.

Referencias
[1] Reglamento de la WCA, artículo 9f8 - https://www.worldcubeassociation.org/regulations/#9f8
[2] Código fuente de la simulación - https://github.com/GuidoDipietro/simulacion-speedcubing