Tesis de Licenciatura

Generación automática de manos de Bridge con restricciones

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Angel Mario Traversi
LU 2057/86
e-mail: atraversi@dc.uba.ar

Director: Lic. Ariel Arbiser
e-mail: arbiser@dc.uba.ar
Para Sol y Merlina
Índice

AGRADECIMIENTOS IV

RESUMEN V

INTRODUCCIÓN 1

CAPÍTULO 1 2

BREVE DESCRIPCIÓN DEL JUEGO 2
   Inicio 2
   Remate 2
   Carteo 3
   Objetivo 4
   Definiciones 4

CAPÍTULO 2 6

REVISIÓN Y EVALUACIÓN DE GENERADORES EXISTENTES 6
   Big Deal 6
   Deal 3 7
   Hands 9
   Dealer 9
   JavaShuffle 10
   PlayBridge 10
   Bridge Baron 11

CONCLUSIONES 11

CAPÍTULO 3 12

PROBLEMA BÁSICO 12
   Requisitos de un buen generador 16
   Condiciones atómicas y restricciones de un buen generador 17

CONCLUSIONES 17

CAPÍTULO 4 18

EL PROBLEMA Y UNA SOLUCIÓN 18
   Condición general 18
   Idea general usando asignaciones 19
   Heurísticas de Distribución 21

CAPÍTULO 5 28

MANOS PARA MÁS DE UN JUGADOR 28
   Smartgen 28
   Puntos de Distribución 30
Agradecimientos

Hace 10 años que terminé de cursar la última materia de la Licenciatura y la Tesis era mi asignatura pendiente. Quiero agradecer a Sol, mi esposa, por haberme apoyado en su realización. Siempre estuvo a mi lado, incluso cuando la estaba terminado de escribir y comenzaban las contracciones del alumbramiento de Merlina, nuestra hija.

También quiero agradecer al Lic. Ariel Arbiser, mi director de Tesis, por su entusiasmo y ayuda para realizar este proyecto.
Resumen

El Bridge es un juego de naipes para cuatro jugadores en dos parejas, que tiene por objeto estimar el número de bazas que una pareja puede lograr y luego cumplirlas. En este juego hay varias facetas apropiadas para la aplicación de técnicas de búsqueda de soluciones a problemas de satisfacción de restricciones, posibles de encarar desde distintos puntos de vista. Uno de los problemas pocas veces encarado hasta ahora es el de la generación de manos que cumplan ciertas condiciones adecuadas para distintos tipos de estrategias y jugadas, tanto en la etapa del remate como la del corte. En esta tesis se propone un modelo para la satisfacción de restricciones aplicables a las manos de los cuatro jugadores, contemplando gran variedad de condiciones, muchas tomadas y construidas a partir de esquemas reales de valoración de la mano empleados por los jugadores de Bridge, que incluye darle puntajes a los honores (As, K, Q, J) presentes en la mano, a los fallos (ausencia de cartas de un palo), semifallos (palos con una sola carta), palos largos (más de 4 cartas), etc. También se contempla el especificar rangos en el número de cartas de determinados palos, y la posibilidad de cartas fijas, preasignadas a ciertos jugadores. Por otro lado se describe una manera de obtener condiciones en base a la información descubierta durante el remate. Se implementa un generador de manos en base a este modelo, y se hace una comparación de la performance entre este y otros varios existentes a la fecha, así como con el simple esquema de "generate and test". Por último, se muestra una serie de aplicaciones a varios aspectos del Bridge (y otros juegos): al estudio del juego en sí, a programas que juegan y a la enseñanza del juego.

Abstract

The game of Bridge is a 4-player card game in teams of two, its object being to estimate the number of tricks a team can win, and then to win them. There are many facets in this game which are suitable for applying constraint satisfaction problem techniques, which can be faced from many points of view. One problem which has been given little attention is hand generation satisfying given conditions, adequate for certain plays and strategies for the bidding stage as well as for the card playing stage. In this thesis we propose a model for constraint satisfaction affecting all the four players' hands. We allow a big family of conditions, many of them taken from real schemes for hand valuation used by Bridge players, such as scoring for honor cards (As, K, Q, J) present in hand, voids (suits with no cards), singletons (suits with a single card), long suits (more than 4 cards), etc. We also allow specifying a range of the number of cards in some suits, and the possibility of fixed cards -assigned for some players. Moreover, we describe a way to obtain conditions based on information gathered during the bidding. We implement a hand generator based in this model, and we compare its performance with other current generators, including the simple generate-and-test model. Last, we show several applications to various aspects of Bridge (and other games): the game itself, playing programs and teaching of the game.
Introducción

El Juego de Bridge, particularmente el Bridge por computadora, constituye un dominio específico donde se pueden aplicar muchas de las técnicas de inteligencia artificial, incluyendo planeamiento, toma de decisiones y aprendizaje automático.

La generación de manos de Bridge es un problema de restricciones que, habiendo recibido poca atención hasta ahora, ayuda en forma grande a casi toda estrategia de aprendizaje no solo para los humanos sino también para las máquinas, como por ejemplo, para probar la fuerza de juego de un programa. Se sabe que algunos programas juegan mejor que otros en ciertos tipos de manos, por ejemplo jugando a slam o a sin triunfo. En esta tesis se plantean modelos para la generación de manos de Bridge usando la mayoría de los tipos comunes de manos. Con esta meta se estudiaron algoritmos específicos para generar manos normales, manos equilibradas, manos altamente desequilibradas, manos para game, manos para slam, etc. La mayoría de estos tipos de manos se pueden analizar considerando no solo la mano de un solo jugador sino también la de su compañero, o la de los oponentes, o la de los cuatro jugadores simultáneamente. Por este motivo nos hemos ocupado también de proveer algoritmos para la generación de dos o cuatro manos simultáneamente, que en conjunto satisfagan ciertas condiciones.

Otro importante uso posible de la generación de manos con restricciones es durante el juego. Durante alguna fase del juego, podría ser posible generar manos teniendo en cuenta los datos adquiridos hasta el momento (subasta, carteo, etc.) y estimar las cartas de los otros jugadores o simular el resto del partido.

Actualmente todos los generadores de manos distan mucho de ser óptimos. Todos los que he evaluado son del tipo conocido como Generate & Test (GAT): se genera una mano al azar y luego se testean las restricciones. Si se cumplen, listo. Sino se vuelve a generar otra mano. Así hasta encontrar la mano requerida o parar luego de un cierto número de intentos. La idea principal de esta Tesis es hacer una aporte al respecto y presentar algoritmos para los que, usando la mayor cantidad y calidad de restricciones, puedan generar la o las manos pedidas.

En esta tesis se encara el problema de la generación de manos desde el punto de vista de un problema de restricciones y se lo ataca de varias maneras. Se describen algoritmos y heurísticas con el fin de resolver parcialmente este problema aplicado a la generación de manos variadas y se muestra que el esquema es bueno para los diversos tipos de manos adecuadas a determinado tipo de juego en Bridge.
Capítulo 1

La ley es dura, pero es la ley.

No es el fin de esta tesis explicar el juego del Bridge. Si bien vamos a detallar en parte la idea básica del juego y algunos conceptos claves que utilizaremos de aquí en más, para mayores detalles y referencias sobre el juego en sí y sus principales ideas y problemas se puede consultar cualquier libro o reglamento de Bridge, por ejemplo [TRU 87], [FRA 95], [GOR 73], [GOR 85].

Breve descripción del Juego

El juego del Bridge es un juego de naipes de origen inglés derivado del Whist, un juego de bazas que se practicaba en el siglo XVI. Se llama baza al grupo de cartas jugadas en un determinado momento, una por cada jugador. El Bridge surgió durante el siglo XIX y, desde su aparición, no ha cesado de extenderse en todo el mundo. En la actualidad cuenta con niveles de organización, reglamentación y participación que normalmente no se dan en los demás juegos de naipes, y existen clubs y organizaciones en todo el mundo.

Se trata de un juego para cuatro jugadores en parejas de dos, con dos partes diferenciadas: subasta y carteo. En la subasta o remate, los cuatro jugadores van declarando cuántas bazas se comprometen a ganar y el equipo que prometió el mayor número de bazas se dice que establece el contrato. Éste es el equipo declarante. Luego, en la etapa de carteo, se van jugando las bazas de forma tal que el equipo declarante trata de conseguir como mínimo el total de bazas contratado. El otro bando, los defensores, tiene como objetivo limitar al máximo el número de bazas ganadas por el declarante, es decir, trata de impedir el cumplimiento del contrato.

Inicio

El jugador que por sorteo le correspondió la posición de Norte es el dador. Reparte la baraja completa (52 cartas) a los cuatro jugadores. Cada jugador tiene 13 cartas, es decir, son 13 las bazas que se jugarán en el carteo.

Remate

Las declaraciones que se van haciendo en el remate siguen determinados sistemas de acuerdo a las cartas en la mano y los cantos del compañero y los adversarios. Existe una gran cantidad de esos sistemas, la mayoría basada en la valoración parcial de la mano. A partir del total de puntos surgidos en esta valoración el jugador puede llevar a cabo el remate, haciendo sus declaraciones, respondiendo a las de su compañero y/o interviendo sobre las declaraciones de los adversarios, con el objeto de interferir o bien de dar información al compañero que permita mejorar la declaración propia y con el fin de llegar al mejor contrato posible por ambas manos.

En el contrato, además de quedarse estipulado el número de bazas a ganar por el equipo declarante, también se define un palo llamado de triunfo que vale más que los otros palos en el momento del carteo, o bien se define sin triunfo (ST) lo que significa que ninguno de los palos será considerado como triunfo.

Entre los sistemas de remate más conocidos figuran: Bridge World Standard, Acol, Precision, Mayor Quinto y Goren.

Valoración de la mano
Si bien las reglas del remate son sencillas, no es para nada trivial conseguir un sistema de remate, es decir, un protocolo que de algún modo optimice la comunicación con el compañero, así como la interferencia o perjuicio a los contrarios, de modo de conseguir una buena o adecuada declaración. Para esto se conocen varios sistemas de remate, que se basan principalmente en el valor y características que tiene la mano de quien debe cantar. Una de las facetas de esta valoración es la valoración de la mano. Consiste en lo siguiente.

Tras recibir las 13 cartas luego del reparto, el jugador pasa a valorar la fuerza de su mano, que reside en la cantidad de cartas altas (fuerza de honores) que posea y en la longitud de los distintos palos (fuerza de distribución). Para ello suele utilizarse un sistema arbitrario de asignación de puntos ideado por Milton Work, que permite calcular la fuerza global de la misma, y que hoy en día se usa en todo el mundo por virtualmente todos los jugadores de Bridge. (Estos puntos no tienen nada que ver con los puntos ganados o perdidos en el corte sino que son "internos" y provisorios a la hora de rematar y eventualmente a la hora de jugar las cartas).

**Puntos de honor (PH):** Se asigna a cada carta el siguiente valor:

- Por cada as (A): 4 PH
- Por cada rey (K): 3 PH
- Por cada dama (Q): 2 PH
- Por cada jack (J): 1 PH

El resto de las cartas no representan ningún valor en PH. Se suman así los valores de las cartas y el total constituye en el valor de la mano en PH.

Es de destacar que la baraja completa contiene en sus 52 cartas un total de 40 PH (10 de cada palo) por lo que a cada jugador le corresponde en promedio 10 PH en cada mano. Es decir, si en una mano determinada un bando reúne por ejemplo 25 puntos, ello indica que los adversarios tienen 15 PH.

Ha habido y hay otros modos de valorar la mano [LAW 83], por ej. se ha propuesto asignar puntos de manera “exponencial” a A (8 pts.), K (4 pts.), Q (2 pts.) y J (1 pta.). Incluso otros sistemas más “precisos” otorgan puntaje también a las demás cartas. Nosotros sólo consideraremos el sistema clásico comentado arriba ya que está universalmente aceptado y permanentemente usado y muchas pruebas empíricas lo han confirmado a lo largo de los años y corroborado por numerosos experimentos, pruebas estadísticas, torneos y programas basados en esto, lo que hoy en día lo hace un sistema universalmente aceptado, base de otros posibles sub sistemas y variantes.

**Puntos de distribución (PD):** Se añaden a los PH para valorar la fuerza total de la mano en relación con un posible contrato a palo. No se tienen en cuenta en un contrato a sin triunfo.

Estos puntos se asignan a los palos cortos (pocas cartas de ese palo), que permitirán fallar en un contrato a palo. Se asignan los siguientes:

- Por cada fallo (ninguna carta de un palo): 3PD
- Por cada semifallo (una carta de un palo): 2PD
- Por cada dubletón (dos cartas de un palo): 1PD

Estos puntos no se tienen en cuenta cuando el juego es a sin triunfo, por la razón de que no existen las bazas por fallo.

**Puntos por palo largo (PL):** Consiste en asignar 1 punto adicional por cada carta en exceso de 4 de un palo dado. Por ej, con 7 cartas de trébol se suma 3 puntos. No todos los sistemas actualmente lo usan, y muchos lo usan combinado con los otros pero en circunstancias especiales.

**Carteo**

En esta etapa, los jugadores juegan las cartas por turnos y respetando condiciones tales como la de jugar siempre una carta de igual palo de quien jugó la primera de cada baza, y otras condiciones particulares.

El jugador que haya jugado la carta más alta del palo de salida, o la carta de triunfo más alta en caso en que haya alguna, ganará la baza e iniciará la baza siguiente.
Si el equipo declarante cumple el contrato (gana como mínimo las bazas indicadas) se anotará una determinada cantidad de puntos. Si no cumple el contrato esos puntos los ganará el equipo defensor (multas).

**Puntos por bazas:** se cuentan 40, 30 o 20 puntos dependiendo de si se juega a ST, a un palo alto (pica, corazón) o a un palo bajo (diamante o trébol).

**Game o manga:** 100 o más puntos ganados por bazas.

**Pequeño slam:** recibe este nombre el contrato a 12 bazas (todas menos una). Además de los puntos usuales de cumplir el contrato se ganan puntos extras.

**Gran slam:** contrato a ganar todas las bazas (13). Los puntos extras se duplican con respecto al pequeño slam.

Los puntajes y premios además suelen variar según la pareja se encuentre vulnerable (es decir, ya haya ganado un game) o no vulnerable (no haya ganado un game). Esta variación en el puntaje es tanto para los premios como para las multas, y hace cambiar ciertas decisiones de juego fundamentalmente en el remate.

**Objetivo**

El Bridge es un juego de sumar puntos, a diferencia de otros de suma cero en donde sólo interesa ganar.

En el bridge duplicado (o de competición) el objetivo de cada mano es tratar de conseguir mejor puntuación que otras parejas que juegan las mismas cartas y en la misma posición (obviamente estarán previamente repartidas por un árbitro).

En cambio, en una partida libre, el objetivo del juego consiste en ganar dos games para completar un rubber. Rubber es el conjunto de dos games ganados por el mismo equipo sobre un total de tres.

El objetivo general a largo plazo es hacer la máxima cantidad de puntos posibles. El objetivo particular o local por juego o rubber es cumplir los contratos propios (haciendo como se dice un buen juego de declarante o carteador) y evitar que los contrarios cumplan sus contratos (haciendo una buena defensa).

**Definiciones**

Los siguientes son términos del Bridge usados a lo largo del texto y que no han sido mencionados hasta ahora:

**Suit:** todas las cartas de un mismo palo en una mano

**Palos mayores:** Picas y corazones

**Palos menores:** Diamantes y tréboles

**Distribución:** $w.x.y.z$ es el número de cartas de cada palo sin precisar cuál es el palo, obviamente con $w+x+y+z=13$. Hay veces que se utiliza la 'x' como comodín cuando no interesa precisar el número exacto de cartas en alguno de los palos. Por ejemplo 0.x.x.x indica que en un palo no hay cartas y en los restantes no importa cuáles son los valores exactos.

**Fallo:** distribución $0.x.y.z$ con $x, y, z > 0$. Vale decir, un palo sin cartas.

**Semifallo:** distribución $1.x.y.z$ con $x, y, z > 1$. Vale decir, un palo con una sola carta.

**Doubleton:** distribución $2.x.y.z$ con $x, y, z > 2$. Vale decir, un palo con sola dos cartas.

**Palo largo:** palo con mayor cantidad de cartas en una mano. Data una distribución $w.x.y.z$, es el max($w,x,y,z$).

**Bazas perdedoras (Losing tricks):** mide el potencial de la mano en cuanto a bazas a ganar [KLI 91]. En algunos casos mide mejor que los PHD. Se usa este concepto al jugar contratos a palo. Cada palo tiene como máximo 3 perdedoras.

Por cada suit de 3 o más cartas, contar 1 perdedora por cada A, K o Q que no esté (que falte).
En el caso de dubletones: teniendo AK: 0 perdedoras; Ax, Kx: 1; AQ: 1/2 perdedora y 2 perdedoras el resto. En los semifallos: 1 perdedora para todos los semifallos excepto que sea un A. En ese caso es 0. Y en los fallos: 0 perdedoras.

Mano balanceada o equilibrada: Son las distribuciones 4.3.3.3, 4.4.3.2 y 5.3.3.2. En algunos libros exige que el 5 corresponda a un palo menor (nosotros no lo tendremos en cuenta ya que no es la mayoría).

Mano no balanceada o desequilibrada: todas las menos excepto las balanceadas anteriores.

Tricolor: distribución 4.4.4.1

Digamos que un palo es apto para que sea palo de triunfo en general cuando se tienen en total 8 o más cartas entre los dos jugadores. Y una mano es apta para carteo en sin triunfo, cuando está balanceada. En algunos casos no se pide que las manos de ambos jugadores estén balanceadas, sólo una. Y muchas veces se pide que alguno de ambos jugadores tengan paradas en todos los palos, es decir cartas que permitan tomar al menos una baza al cabo de pocas bazas (por ej, K Q perderá a lo sumo una y ganará la siguiente).

Existen muchos sistemas de juego, que no comentaremos aquí porque escapan al objetivo de esta tesis. Referimos al lector a [KAN 83].

Para más información en el Apéndice 6 incluimos un glosario con los términos más comunes del Bridge.
Capítulo 2

Dealers choice is no big deal.

Revisión y evaluación de generadores existentes

En esta sección se hará una evaluación de diversos generadores de manos existentes. Hemos decidido incluirla como una medida del estado del arte tanto para este tipo de programas como para las técnicas que ellos emplean. Comenzamos con el objetivo de evaluar distintos generadores y llegar a sus flaquezas y motivar de este modo la necesidad de construir un generador fruto de la experiencia de los anteriores, así como el desarrollo de una técnica que permita enmarcarlo en un contexto más formal y que permita en el futuro mejorarla en diversos aspectos así como llevar dicha técnica a otros generadores o bien otros problemas análogos para terrenos diversos, tales como otros juegos y otros problemas relacionados con búsqueda de soluciones con satisfacción de restricciones.

Sorpresivamente hemos comprobado que muchos de los programas importantes de Bridge existentes (como Bridge Baron, por ej) poseen un simple generador del estilo GAT. Esto es realmente desventajoso por motivos obvios. El paradigma GAT, en principio aplicable a muchos y variados problemas de encontrar soluciones o respuestas posibles, es en principio uno de los métodos más ineficientes como es fácil observar. Se trata de generar una mano al azar o casi al azar para luego testear la validez de las condiciones deseadas. En caso de que estas sean satisfactorias, se devuelve la mano así obtenida. En caso en que no (lo más probable en la mayoría de las situaciones), se vuelve a intentar, y así hasta que se de una mano adecuada o bien hasta que se agote el tiempo disponible o hasta que se haya hecho el número de intentos máximo establecido de antemano.

Es inmediato comprobar que no hay garantía alguna de que este proceso pare alguna vez (y que devuelva por consiguiente la mano deseada).

Generadores existentes

Los siguientes son algunos de los generadores de manos de Bridge que se pueden encontrar en la actualidad. Algunos admiten restricciones y otros solo se limitan a generar manos al azar:

- Big Deal
- Deal 3
- Dealer
- Hands
- JavaShuffle
- PlayBridge
- Bridge Baron

Big Deal

[http://www.xs4all.nl/~sater]

Es un generador de manos sin restricciones, gratuito y los fuentes son de dominio público. Solo genera manos aleatorias sin ningún tipo de restricción. Fue desarrollado en Alemania a pedido de la Dutch Bridge Federation para ser utilizado en las olimpiadas de Bridge del 2000 en Maastricht. Fue programado en C por Hans van Staveren. El diseño e implementación de las rutinas matemáticas a cargo de Jeroen Kuipers, supervisado por el Profesor de Matemáticas Koos Vrieze de la Universidad de Maastricht.
El propósito fue desarrollar un programa que genere manos lo más al azar posible y que sea seguro. Los generadores utilizados con anterioridad por la federación tuvieron serios problemas (manos repetidas, secuencia de manos repetidas, manos conocidas por los jugadores, etc).

Estudiaron el por qué de estos problemas y trazaron tres requerimientos mínimos:

1. El software tiene que ser capaz de general cualquier posible mano real de bridge.
2. Tiene que generar cualquier mano con la misma probabilidad, sin ser influenciado por ninguna circunstancia como la mano anterior, el número de tablero, etc.
3. Tiene que ser imposible predecir una mano, por más que se vean y estudien manos anteriores.

Para ello hicieron hincapié en dos temas: randomización y seguridad.

Ellos dicen que la mayoría de los generadores de manos en el mejor de los casos utilizan 32 bits para el la semilla de un pseudo-generador random (PGR). Y que hay casos en que se solo se usan 20 bits (en DOS, por ejemplo). Eso significa que a lo sumo hay $2^{32}$ posibles series de valores del PGR. Las posibles manos de Bridge están en el orden de $0.7 \times 2^{56}$ (ver Capítulo 3 – Problema básico para más detalle acerca de éste valor).

Acorde a la Paradoja del Cumpleaños (Birthday Paradox) en $2^{32}$ seguramente se van a repetir valores como mínimo $\sqrt{2^{32}} \approx 2^{16} = 65000$ corridas. En el caso de 20 bits, mucho peor, las valores pueden repetirse cada $\sqrt{2^{56}} = 2^{10} = 1024$ corridas. Este es el principal problema que resolvieron. Para ello utilizaron semillas de 160 bits de longitud.

$\sqrt{2^{160}} = 2^{80}$ es un número que si bien no cubre todas las posibles manos, es enormemente grande. Finalmente, la semilla se consigue capturando teclas al azar al momento de generar las manos.

El otro tema fue la seguridad: evitar la ingeniería inversa, es decir, que no sea posible deducir la semilla a partir de una mano dada ni predecir una mano según la anterior. Para ello se utilizó un hash encriptado.

El programa genera una cierta cantidad de manos y la salida la hace en distintos formatos conocidos: duplicado, BRI, DGE, Bernasconi, Bore Hand, valores separados por coma, etc.

Debido a que este generador no considera restricciones, no se analizó su performance.

**Deal 3**

[http://thomaso.best.vwh.net/bridge/deal]


Para las restricciones primero usó un interprete parecido a Perl que era un tanto lento. Luego, en 1992, fue cambiado por el Tcl [TCL 03]. Tcl es un lenguaje interpretado que tiene posibilidades para llamar rutinas en C. Las sucesivas versiones de Deal (y a la vez, de Tcl) lo fueron mejorando en performance.

El lenguaje Tcl admite funciones, procedimientos, variables locales y globales, constantes, instrucciones del tipo IF THEN, ciclos, etc. La ventaja es su gran poder expresivo para definir restricciones. La desventaja es que no es fácil para un usuario común.

El sistema además ofrece funciones de sistema como:

- `[hcp north]` retorna los puntos de honor de Norte
- `[hcp north spades]` retorna los puntos de honor de Norte en Picas
- `[balanced]` dice si la mano es balanceada o no
- `[diamonds west]` indica la cantidad de diamantes para Este
Se pueden definir condiciones que acepten manos, o que las rechacen. Para ello existen las sentencias accept y reject. Por defecto el sistema rechaza la mano, a menos que el script la acepte.

Algunos ejemplos de scripts:

a) Encontrar una mano en las que Sur tiene 10 o más puntos de honor en corazones o espadas

```
main
{
    if {([hcp south hearts spades]>10} { accept }
}
```

b) Aquí las cartas de Sur están fijadas “a mano”. Y para Norte se deben dar las condiciones de 5 o más cartas de espadas y más de 15 puntos de honor.

```
south is "- 98532 A864 T962"
main
{
    if {[spades north]>=5 && [hcp north]>=12} { accept }
}
```

c) Este es un script que acepta una mano si Norte o Sur tienen puntaje fuerte sin triunfo.
Para evaluar esta situación, definimos una función booleana genérica notrump que chequea que la mano no sea balanceada y que el PH esté en cierto rango. Luego en el procedimiento principal se invoca una vez para Norte y otra para Sur con rango de PH alto (entre 15 y 17).

```
proc notrump {hand min max}
{
    if {![balanced $hand]} { return 0 }

    set hc [hcp $hand]
    if {$hc < $min || $hc>$max} { return 0 }

    return 1
}

set NTmin 15
set NTmax 17
main
{
    if { [notrump north $NTmin $NTmax]} { accept }
    if { [notrump south $NTmin $NTmax]} { accept }
}
```

Las variables, cuando se utilizan, se escriben precedidas con $, cuando se definen, no. En varios aspectos el Tcl tiene la misma sintaxis del lenguaje C.

El output del programa se puede adaptar y modificar vía Tcl.
Las pruebas efectuadas sobre este generador pueden verse en el Apéndice 3.

La documentación explica como un ejemplo puede hacerse de varias formas y que unas son más eficientes que otras. Lo malo es que hay que ser un programador que conozca Tcl para poder hacer una consulta ‘difícil’ en forma ‘eficiente’, y un usuario regular no podría.

Como conclusión resaltamos que lo más interesante de este generador es el poder expresivo que tiene para las consultas. Lo malo, además de que utiliza el esquema GAT, es lo difícil de usar para usuarios comunes. Otra cosa que no está bien resuelta son las cartas preasignadas. Esta opción requiere que se preasiguen las 13 cartas (la mano completa) del jugador en cuestión.

**Hands**

[http://www.whiteoaks.com/hands]

Es otro generador sin restricciones. Fue escrito en lenguaje C por Morris Jones y la estructura de datos (en lo que se refiere a cómo modelar una mano) fue ‘prestada’ por Matthew Clegg’s del programa OKBRIDGE [OKB 03]. Es de dominio público.

Fue pensado para generar manos para Bridge duplicado. En cada corrida indica, para cada tablero, las cartas de cada jugador, dado y vulnerabilidad. Incluye distintos formatos de salida.

Se recomienda el uso de la versión Unix que corre en Solaris ya que su PGR es de 48 bits.

Debido a que este generador no considera restricciones, no se analizó su performance.

**Dealer**

[http://www.dombo.org/henk/dealer.html]

Es uno de los generadores con restricciones más antiguos. Es del tipo GAT. Se puede usar para generar manos para aprender remate de equipo o partnership bidding, para generar estadísticas y para recrear partidas ya jugadas. Es gratuito y con código fuente accesible. El código original es de Hans van Staveren y fue actualizado por Henk Uijterwaal (actual coordinador del código). Hay versiones para Windows y varias plataformas Unix/Linux.

Como entrada se ingresan los siguientes parámetros (hay defaults para la mayoría de ellos):
- número máximo de manos al azar (el máximo número de intentos que ejecutará)
- cuántas manos producir (es decir, que cumplan la condición)
- cartas preasignadas (se pueden asignar cartas manualmente)
- condiciones
- acciones

En las condiciones se pueden incluir varias funciones build-in unidas por conectores lógicos. Las funciones incluyen:

- **Hep** (puntos de honor): por jugador y por jugador-palo
- **Shape** (patrones): distribuciones con patterns indicados como “4333” (4 picas, 3 corazones, etc. la posición indica el palo) o “any 5332” (no importa el palo). Se pueden combinar con + y - (disyunción y conjunción negada, respectivamente)
- **HasCard** (forzar carta): que incluya determinada carta
- **Control**: por jugador y por jugador-palo (Control: puntaje que se calcula sumando por cada A 2 puntos y por cada K 1 punto)
- **Loser** (bazas perdedoras): por jugador y por jugador-palo

Las acciones:
- **Imprimir** una mano, todas las manos, la mano de norte, etc
- **Frecuencias**: calcula y muestra el histograma de valores de la expresión ingresada en el rango indicado,
- **Promedios**: calcula y muestra el promedio de la expresión ingresada para todas las manos que satisfacen las condiciones.

Se pueden utilizar variables: `<identificador>=<expresión>`

Algunos ejemplos:

a) **retornar una mano en la que norte tenga más de 30 puntos de honor**

   condition
   hcp(north)>30
   produce
   1

b) **retornar una mano balanceada de sur**

   condition
   shape(south, any 4333 + any 4432 + any 5332 )
   produce
   1

Es interesante que incluya funciones para generar estadísticas.
La forma de ingresar las consultas le da buen poder expresivo (no tanto como el Deal 3) pero suficiente. En punto flaco es que no se puedan definir restricciones usando **puntos de distribución**.
Y obviamente sufre el problema de todo programa GAT: puede que nunca llegue a la solución.

**JavaShuffle**

[http://www.ripe.net/home/henk/bridge/software/shuffle.html]

Es un applet de java, escrito por Henk Uijterwaal (actual coordinador del código del generador Dealer). Es un generador de manos aleatorias sin restricciones.
Se ejecuta a través de un navegador de internet que soporte java. El applet corre en el servidor. Utiliza un PGR de 48 bits y la semilla se compone con la dirección IP y la fecha y hora de la consulta.

Debido a que este generador no considera restricción alguna, no se analizó su performance.

**PlayBridge**

[http://www.playbridge.com]

Es un generador con restricciones, escrito por Richard Podkowik. El copyright está a nombre de Jay Peak Systems, Inc. Se usa a través de un navegador de internet. Solo podemos decir que son páginas ASP, ya que no tenemos otro dato sobre cómo fue programado.

Tiene tres formas de configurar las restricciones:

1. One-hand (una mano)
2. Advanced (avanzado)
3. Pick-cards (elección de cartas)
One-hand permite definir restricciones únicamente sobre uno de los cuatro jugadores. Los parámetros que contempla son: jugador, puntos de distribución, puntos de honor, patrones de distribución, palo largo, cantidad de tableros y formato de salida. Este modo genera, en los restantes jugadores, manos al azar.

Advanced contempla restricciones para dos jugadores. Son mucho menos variadas que en el caso anterior. Solo incluye rango de cantidad de cartas por palo y rango de puntos de honor.

Pick-cards permite definir qué cartas se asignan a un solo jugador. Se pueden definir las 13 cartas o menos, en cuyo caso completa con cartas al azar.

Este generador es el menos interesante debido a que solo permite definir restricciones para uno o dos jugadores.

No hay documentación que diga si es del tipo GAT o no. Una página del mismo sitio llamada Shuffle Project explica la habilidad que tiene el generador en generar manos reales. Muestra estadística basada en miles de manos generadas y compara promedios de distribuciones encontradas con tablas de la Enciclopedia de Bridge ACBL [ACB 03], con resultados muy próximos a los reales. Esto nos hace presumir que el esfuerzo principal del algoritmo está puesto en la generación al azar de las manos.

**Bridge Baron**

[http://www.bridgebaron.com]

Bridge Baron es uno de los mejores programas de juego de bridge disponibles a la fecha. Cuenta con un módulo generador de manos, totalmente arcaico. Curiosamente y contrariamente a la calidad de este programa de juego, el generador no es más que un GAT puro, sin ningún tipo de heurística o inteligencia. Podemos decir que tiene las posibilidades usuales: PH, PL, PD, cartas por palo, con mínimo y máximo para cada uno, así como la posibilidad de pedir manos adecuadas para un remate determinado. La generación por tanto no siempre es efectiva y puede no resolverse.

Una vez generadas las manos, el usuario tiene la posibilidad de jugar esas manos y tratarlas como manos usuales: grabarlas, verificar el juego por la computadora, competir con ellas, etc.

**Conclusiones**

En este primer estudio sobre los generadores existentes, hemos notado en forma permanente tres problemas principales:

1) muchos son del estilo GAT o similar
2) algunos carecen de algunas condiciones combinadas (por ej PH + PD)
3) algunos carecen de condiciones para manos múltiples, es decir para más de un jugador

En los capítulos que siguen, nosotros intentamos resolver en parte estos problemas, y a la vez dejar caminos abiertos para ciertas líneas de investigación que, podrían llevar a mejoras substanciales en este problema y similares.
Capítulo 3

A veces no interesa tener una respuesta, sino comprender la pregunta.
Kung-Fu

Problema básico

Comenzaremos este capítulo viendo una implementación de un generador de manos GAT y describiremos sus inconvenientes. A continuación mostraremos, a modo de ejemplo, cómo modificar (dirigir) la generación teniendo en cuenta una restricción arbitraria: obtener una mano balanceada.

Según hemos visto en los generadores estudiados, el siguiente es el algoritmo utilizado para modelarlos:

```
Algoritmo [3.A] GAT
    repetir
    Generate( Mano )
    hasta Test( Mano )
    fin
```

`Generate` es un procedimiento que en forma random elige las cartas para la mano.
`Test` es una función que evalúa las cartas elegidas y si cumplen las restricciones definidas, retorna verdadero.
`Mano` representa la mano de cada uno de los 4 jugadores.

`Generate` es trivial: reparte las 52 cartas entre los cuatro jugadores en forma random:

```
Procedure Generate(Mano)
    cartas_sin_asignar := las 52 cartas del mazo
    para i := (norte, este, sur) hacer // set itera por 3 jugadores
        repetir // se elige una carta al azar
            r := nro random entre 1 y length(cartas_sin_asignar) ;
            carta := cartas_sin_asignar[r] ;
            // se agrega esa carta a la lista del jugador i
            Mano[i] += carta ;
        hasta length(Mano[i]) = 13 ; // listo las 13 cartas del jugador
        fin para;
        // luego las 13 cartas que quedaron en el mazo se dan a Oeste
        Mano[ oeste ] := cartas_sin_asignar ;
    fin ;
```

Respecto a la implementación de `Test`, tenemos:
Procedure Test(Mano)
  Result := false ;
  // se pregunta por cada una de las restricciones si se cumplen
  si Mano.PH > Restricciones.maxPH or Mano.PH < Restricciones.minPH
    entonces EXIT;
  para cada palo hacer
    si Mano[palo].PH > Restricciones.maxPH[palo] or
       Mano[palo].PH < Restricciones.minPH[palo] entonces EXIT;
  fin para ;
  si Mano.PD > Restricciones.maxPD or Mano.PD < Restricciones.minPH
    entonces EXIT;
  si Mano.Balanceada <> Restricciones.Balanceada entonces EXIT;
  // y así con cada restricción...
  fin ;

Como se ve, no es algo complicado de resolver computacionalmente. Es probable que la complejidad mayor esté en la interface de usuario con respecto a la forma de especificar las restricciones, es decir, el lenguaje a usar, más que en el algoritmo para resolverla.

El gran problema es su ineficacia. Es claro que no hay garantía de que en un tiempo razonable se pueda generar una mano pedida.
Todo depende de las restricciones que usemos. Si las restricciones acotan mucho la mano a generar –la restricción es muy fuerte– la probabilidad de que se dé esa mano es menor y puede suceder que el algoritmo [3.A] tarde muchos intentos (tiempo) para encontrarla o nunca la encuentre.

Entonces resulta conveniente plantear una solución más eficiente.

Lo que proponemos es utilizar las restricciones en el momento de elegir las cartas.

Si bien la cantidad de manos de Bridge no es un número infinito, el número es muy grande y justifica el esfuerzo de mejorar la técnica para generarlas. La cantidad de posibles manos de Bridge se calcula en:

\[
\binom{52}{13} \binom{52-13}{13} \binom{52-26}{13} = 53,644,737,765,488,792,839,237,440,000 =
5.3644 \times 10^{28} \equiv 0.7 \times 2^{96}
\]

A continuación veremos un algoritmo de ejemplo sobre cómo utilizar las restricciones al generar una mano.

Supongamos que la restricción consiste en:

- Generar una mano balanceada.

El siguiente es un algoritmo que genera manos balanceadas. Primero decide al azar cuál es la distribución a usar (4333, 4432 o 5332), luego elige al azar las cartas para cada uno de los palos:

**Algoritmo GenerarManoBalanceada [3.B]**

Sea el mazo modelado en 4 listas $S_1, S_2, S_3, S_4$ que representan cada uno de los palos.
\[ r := \text{nro random entre 1 y 3} \]  // decide cual distribución usar

Si \( r = 1 \)  // usa distribución 4 3 3 3
Elegir 4 cartas random de \( S_1 \)
Elegir 3 cartas random de \( S_2 \)
Elegir 3 cartas random de \( S_3 \)
Elegir 3 cartas random de \( S_4 \)

Sino Si \( r = 2 \)  // usa distribución 4 4 3 2
Elegir 4 cartas random de \( S_1 \)
Elegir 4 cartas random de \( S_2 \)
Elegir 3 cartas random de \( S_3 \)
Elegir 2 cartas random de \( S_4 \)

Sino Si \( r = 3 \)  // usa distribución 5 3 3 2
Elegir 5 cartas random de \( S_1 \)
Elegir 3 cartas random de \( S_2 \)
Elegir 3 cartas random de \( S_3 \)
Elegir 2 cartas random de \( S_4 \)

Fin

Fin

Claramente este algoritmo encuentra una mano balanceada.

Ahora, extendemos el ejemplo:

- \textit{Generar manos balanceadas para Norte y Sur.}


Seguimos con otro ejemplo más restrictivo:

- \textit{Manos balanceadas para norte, este y sur.}

Supongamos que ejecutamos [3.8] una vez para cada jugador. El resultado en la mayoría de los casos serán tres manos balanceadas, pero puede ocurrir que no. Por ej.:

Para Norte: obtiene una distribución 5 3 3 2 con \( S_1 \) picas.
Para Este: obtiene una distribución 5 3 3 2 con \( S_1 \) picas también (a esta altura en picas solo quedan tres cartas para repartir).
Para Sur: si \( S_1 \) es picas otra vez, no tendrá cartas suficientes para ninguna de las tres posibles distribuciones balanceadas.

Como vemos el algoritmo puede fallar en el caso de tres manos balanceadas.

Pero nuestro objetivo no es conseguir un algoritmo que genere directamente la mano pedida. Sino queremos dirigir el resultado para que converja en las restricciones. La idea es que la mano se debe encontrar necesariamente en el primer intento, sino que lleve \textit{pocos intentos} encontrarla.

Entonces, el algoritmo [3.8] es válido para el último ejemplo, si lo usamos en el siguiente ciclo:

\[
\text{Repetir}
\begin{align*}
\text{GenerarManoBalanceada(norte)}; \\
\text{GenerarManoBalanceada(este)}; \\
\text{GenerarManoBalanceada(sur)}; \\
\text{Oeste := resto del mazo};
\end{align*}
\]

\text{Hasta EsBalanceado(norte) and EsBalanceado(este) and EsBalanceado(sur) ;
Es decir, generamos tres manos balanceadas y testeamos si realmente los son. Sino, repetimos nuevamente, hasta que se de la restricción buscada.

El algoritmo anterior seguramente se puede mejorar, pero no pretendemos hacer eso ahora. Lo que queremos mostrar es el concepto: dirigir la generación y luego verificar que se cumplan las restricciones.

Número de Intentos

Tratemos de medir el término *pocos intentos*.

Utilizaremos el número de intentos como una medida de la eficiencia de un generador. Resulta más adecuado que usar el tiempo en encontrar una mano, debido a que el tiempo es dependiente del hardware utilizado y el número de intentos no.

El número de intentos es muy fácil de calcular: consiste en contar cada una de las veces en que el generador repartió cartas más allá de cuál fue el resultado.

Todos los generadores con restricciones estudiados de una u otra manera lo informan luego de encontrar o no la mano.

Observación: el número de intentos en que un generador GAT encuentra una mano se encuentra en relación directa (creciente) respecto a lo restrictivas que son las condiciones para generarla. Muchas veces este número de intentos resulta en el peor caso exponencial en el número de condiciones, como es bien sabido.

La medida de cuánto restrictiva es una mano en este esquema (suponiendo un algoritmo random de selección de cartas uniformemente distribuido), se puede medir. Es análogo a calcular con qué frecuencia una mano de esas características se da en la realidad. Podemos medirlo en forma analítica o empírica -hay una valoración de esto en el Capítulo 5, Validación de las restricciones-.

Por ejemplo, una mano muy poco probable como PD(N) = 9 (una distribución 13.x.x.x para Norte), nunca podremos generarla probando distintos generadores GAT.

Podemos concluir que un generador GAT resuelve en:

- Pocos intentos, cuando las condiciones son poco restrictivas
- Muchos intentos, cuando las condiciones son medianamente a muy restrictivas
- No encuentra mano, cuando las condiciones son sumamente restrictivas

Para un generador dirigido, esta definición de *poco*, no es tan fácil.

En el Capítulo 5 veremos que la condición PD(N)=9 se puede resolver sin problemas con la heurística propuesta en el primer intento. Esto muestra que en un generador dirigido lo restrictivo de una condición no está en relación directa (creciente) con el número de intentos.

Por otro lado, cuando con el generador dirigido se resuelven manos que uno del tipo GAT no encuentra (veremos que esto sucede), no importa tanto si lo hace en pocos o muchos intentos. Se puede ver en nuestro esquema que en muchos casos lo hace, y de algún modo el número de intentos tiende a ser bajo. Si bien hemos visto empírica e intuitivamente evidencias de que resultaría imposible en la práctica construir un generador perfecto que de una manera eficiente óptima devuelva manos deseables, nuestro objetivo es aproximarlo desde distintos costados. Decimos "en la práctica" porque en teoría podría ser plausible dado que se podría contar con una base de datos (gigantesca) con todas las manos posibles, más aún, con todos los repartos posibles para cuatro jugadores, clasificada en forma de árbol según ciertas condiciones (de las maneras usuales al usar árboles, árboles B, árboles con balanceos adecuados, etc) y cuando se pida una mano con ciertas condiciones, sólo será cuestión de buscar en dicho árbol y así obtenerla (no sólo una sino todas las que satisfagan las condiciones). Esto es virtualmente imposible con las
capacidades de cómpute actuales en virtud de dos razones: 1) el número enorme de manos (ya comentado), y 2) la dificultad (en tiempo y espacio) en conseguir dicha clasificación, o bien una razonable.

Entonces, proponemos un algoritmo dirigido, al cual podemos ir analizándolo viendo que tan rápido converge a las restricciones en diversas pruebas. Para ello usaremos la media y desviación estándar del número de intentos. Esto constituye una manera de medir cuántos intentos efectivos son necesarios a largo plazo, lo cual nos sirve para comparar las distintas técnicas o variantes que estamos empleando.

**Requisitos de un buen generador**

Hemos notado que como mínimo existen cuatro puntos que debe contemplar un buen generador de manos con restricciones. Nos basamos para esto en la experiencia del uso de otros generadores y en el mismo juego de Bridge [FRA 95], [GOR 73] [GOR 85]:

- Completo
- Lenguaje adecuado para especificar las restricciones
- Debe contemplar las restricciones necesarias
- Eficiente en cuanto a encontrar la respuesta

**Completitud**

Un algoritmo de generación de manos es completo si todas las manos que satisfacen las restricciones son posibles de ser generadas. Lo llamaremos uniformemente completo si además la distribución es uniforme en la probabilidad de que sean generadas (UPD). Por un lado la restricción de completo que parece intuitivamente útil, no lo es tanto. Un GAT suele ser completo y no por eso eficiente o útil en la práctica. Un algoritmo podría ser completo trivialmente (por ej cada cierto tiempo obteniendo una mano al azar y verificando las condiciones) lo cual no aporta mejora alguna a nuestra idea. Por otro lado, parece difícil en principio establecer que un algoritmo sea uniformemente completo; no nos ocuparemos de esto en detalle en la presente tesis. Sólo nos limitamos a decir que los algoritmos posteriores parecen ser uniformemente completos al menos en forma aproximada tal como muchos ejemplos de prueba indican.

**Lenguaje adecuado para especificar las restricciones**

Se puede utilizar una variante de la lógica de primer orden como lenguaje para especificar condiciones. Es muy claro para especificar las cualidades de una mano de Bridge, indicando Puntos de honor, puntos de distribución, longitud del suit y otros parámetros como símbolos de predicado. El problema principal con esto es que casi cualquier combinación de fórmulas es expresable y ésta puede causar problemas de tractabilidad.

**Debe contemplar las restricciones necesarias**

Las restricciones a tener en cuenta deben ser las que todo jugador espera. Sumando las restricciones que todo profesor de Bridge necesita para enseñar a sus alumnos. Es decir, serán restricciones dadas por condiciones presentes en los manuales y sistemas de valoración de la mano en Bridge de acuerdo a sus tipos de cantos: aperturas, respuestas, redeclaraciones, intervenciones, obstrucción, convenciones, etc.
Eficiente en cuanto a encontrar la respuesta

Lo ideal sería encontrar la mano la más rápido posible, es decir, en el menor número de intentos. Eso se espera lograr al menos en la mayoría de los casos, si bien experimentalmente tenemos evidencias de que no es siempre posible lograrlo en todos los casos.

Condiciones atómicas y restricciones de un buen generador

El objeto de esta sección es establecer cuáles son los tipos de condiciones a manejar para un buen generador de manos adecuadas para Bridge.
Luego de un relevamiento de distintos sistemas de remate [KAN 83], [ROO 95], [SEA 99] hemos encontrado que las condiciones que aparecen son las que hacen explícita mención de PH, PL, PD, fallos, semifallos, distribución, y otros. Y muchas o todas ellas con uso de alguna de las siguientes relaciones: igual, menor o igual, distinto. Por ej, PH <= 13 (para la apertura), fallos + semifallos = 2 (para el doble informativo), distribución balanceada (para la apertura en sin triunfo), PL >=6 (apertura de obstrucción).

En vista de lo anterior, las siguientes son las condiciones atómicas que formarán parte de las restricciones a usar por nosotros:

- Puntos de honor
- Puntos de distribución
- Distribución w.x.y.z
- Fallo
- Semifallo
- Dobleton
- Palos mayores
- Palos menores
- Palo largo
- Bazas Perdidas
- Mano balanceada
- No balanceada
- Tricolor
- Bicolor
- Sin triunfo
- Slam

Además de estas condiciones es necesario que se puedan preasignar cartas a los jugadores. Preasignar significa poder especificar que tal carta del mazo debe formar parte de la mano. Este tipo de restricción es útil, entre otras cosas, para recrear manos ya existentes. O para poder crear manos más ricas o interesantes, al combinarlas con otras restricciones. Y una muy importante para nuestro generador: que pueda usarse para generar manos en la etapa del carteo.

Conclusiones

Este capítulo marca el objetivo principal de la tesis: la necesidad de mejorar la resolución de un problema que hasta el momento se hizo en mayor o menor medida en forma ineficiente.
También explicamos qué puntos deberían tenerse en cuenta a la hora de escribir un buen generador, desde distintos puntos de vista.
Y vimos que, con cierto esfuerzo, podemos dirigir la generación de una mano para que cumpla cierta restricción. Los próximos capítulos están destinados fundamentalmente a la profundización de estas ideas.
Capítulo 4

El problema y una solución

El objeto de este capítulo es la modelización (informal) del problema principal y encontrar alguna solución.

Condición general

En virtud de las consideraciones anteriores, se decidió agregar la siguiente condición general al algoritmo generador:

    para generar una mano que cumpla cierta condición, hay un mínimo número de cartas a obtener teniendo en cuenta esa condición

Esa condición general podrá ser aplicable a cualquier otro juego de cartas en donde haya que generar una mano. En el caso de Bridge, vamos a intentar refinar (tal vez informalmente) esta condición general hasta llegar al algoritmo de generación de la mano.

En otros términos, el reparto de cartas se hará teniendo en cuenta la condición en mayor o menor medida, hasta un número de carta determinado (vale decir no todas las cartas de la mano se elegirán usando ese criterio sino un número inicial). Por ej, si el objetivo fuese satisfacer un mínimo de PH, entonces cuando obtenidas esas cartas, el resto de la mano se puede completar al azar. Esto no es así si el PH requerido fuese exacto, pero de todos modos análogo será el proceder en este caso.

Una vez generada la mano, aún hace falta evaluar la condición pedida, para determinar si la mano satsfice las condiciones que se desearon se cumplieran. Esto en principio podría ser evitado si la condición general fuese lo suficientemente buena o fina como para que luego no haga falta una verificación posterior.

Lo anterior se puede expresar con los siguientes algoritmos:

    // retorna una mano que cumple las restricciones
    Algoritmo [4.A] ObtenerMano( Restricciones )
    Begin
        Repetir
            Mano := GenerarMano( Restricciones )
            hasta Cumple( Restricciones, Mano )
        fin;
        return Mano;
    fin;

    // arma una mano teniendo en cuenta las restricciones
    Begin
        Elijo cartas contemplando las Restricciones
        Si faltan elegir cartas, las tomo al azar
        return Mano;
    fin;
El *Cumple* es similar a *Test* usado en el esquema GAT [3.A].

**Idea general usando asignaciones**

En esta sección haremos un bosquejo de un primer modelo para generación de manos con cierta generalidad, y la razón por la cual no es todo deseable. Este modelo consiste en información asociada a las cartas, que estará dada en función de las condiciones a utilizar para las manos que se deseen generar.

Consideraremos asignaciones de valores a las cartas, vale decir, funciones "discretas" $f: C \rightarrow N$. Por una cuestión de comodidad lo haremos de este modo en lugar de considerar funciones con rango $[0,1]$ y cuya suma total sea 1, es decir, en lugar de funciones de distribución usaremos a nivel práctico asignaciones de números naturales (en nuestro caso, a cada naipes o eventualmente grupo de naipes). La probabilidad estará dada por la proporción de la asignación de cada naipes respecto de todos.

**Asignaciones**

Un hecho que planteamos aquí es el siguiente: dada una condición deseada para repartir una mano, existe una asignación adecuada que haga obtener esa mano (en un número reducido de intentos).

Vale decir, ¿es sólo una cuestión de asignar probabilidad a cada naipes? Como veremos, la respuesta es negativa. Si fuese el caso, al menos para ciertos tipos de condiciones, podríamos pensar en un algoritmo naif como sigue. Es de observar que no siempre será fácil dar, dada una condición, una asignación a las cartas de modo de obtener manos deseadas. Pero veamos el modelo.

Primero un procedimiento auxiliar:

```plaintext
Algoritmo repartir una mano de un mazo de N naipes usando una asignación f.
  Sea F la distribución acumulada de f
  Para cada carta a repartir
    Mientras se deba reintentar
      Elegir un nro r al azar de 1 a N
      Sea x tal que $F(x) \leq r$ y $F(x+1) > r$
      si la carta x no está marcada
        elegir la carta x
        marcar a x
      salir del mientras
      si no, reintentar
      fin mientras
    guardar la carta x en la mano a devolver
  fin para
  devolver la mano armada
fin
```

Y ahora damos el

```plaintext
Algoritmo naif generico
  Dada la condición C
  obtener la asignación f a partir de C
```
Repartir usando f

Observación: obtener f a partir de C podrá ser complicado, pero en muchos casos es posible.

Algunos ejemplos

Ejemplificamos para el caso de un solo jugador.

Ejemplo 1
si la condición es: PH > 20
la asignación es
A -> 10
K -> 8
Q -> 5
J -> 4
2 .. 10 -> 1

Debemos modificar estos valores de manera razonable para condiciones PH > K para otras constantes K. Ver el siguiente ejemplo.

Ejemplo 2
si la condición es: PH = 0
la asignación es
A -> 0
K -> 0
Q -> 0
J -> 0
2 .. 10 -> 1

Discusión

Este esquema de algoritmo parece andar bien con PH pero tiene limitaciones con PD. Por ej, si la condición es PD = k, o bien PD > k, entonces no parece haber una asignación adecuada. Así como en estos ejemplos, si bien esto no constituye una prueba formal, es de destacar que una asignación no siempre permite dar manos para condiciones adecuadas. Esto se puede subsanar enriqueciendo el concepto de asignación. Si por ej la condición es

PD = 9

se puede pensar en una asignación al conjunto P, C, D, T dando un 1 a 3 de ellos (elegido al azar) y 0 al resto. Aunque esto requiere una preselección al azar de uno de ellos.

Y esto no se hace extensivo a otros valores posibles para PD.

Por ej, PD = 8, 7, ... etc: requiere elegir una combinación de palos y nros. de cartas.

Entre las desventajas o limitaciones también podemos destacar las siguientes. Como se dijo, no parece natural ni fácil obtener la asignación adecuada a partir de las condiciones deseadas. Por otro lado, no parece además sencillo
generalizar esto a dos manos, tres manos, etc.; tan solo podría hacerse esto modificando la asignación entre jugador y jugador.

Hemos explorado diversas mejoras a este modelo, llegando a la conclusión de que el conjunto dado de posibles condiciones, dada su riqueza, obliga a analizar cada una de manera distinta. Vale decir, no se trata igual un tipo de condición que otra. Es decir, para pedir PH > k no se tendrá (ni podrá) tratar las manos como cuando se desea pedir NCP >= n, o PD >= d, o distribución = w:x:y:z, etc.

Esto se ve confirmado por el hecho de que son muchas las facetas a tener en cuenta en las manos de bridge, en principio dada la complejidad de las posibilidades del juego.

**Familia de restricciones**

Debido a que la condición es en cierta medida el eje central del generador, se hace necesario caracterizar o clasificar las condiciones de modo tal de poder manipularlas en forma correcta.

Las dos actuaciones que un jugador humano tiene al recibir las cartas en el reparto son, la de contar cuántas cartas son del mismo palo y, cuántos honores se tienen. Más precisamente, cuántos son los puntos de distribución, y cuántos los puntos de honor. Para el primero solo se tiene en cuenta el palo de las cartas. Para el segundo, su valor. Estos datos son independientes, por lo menos como punto de partida para evaluar la mano. Luego se podrán sumar, o verificar cuántos PH se tiene por palo, o cuántas cartas tiene el palo más largo, etc. Todo esto proviene de sistemas reales de juego, conocidos en general por toda la comunidad de jugadores de bridge.

Lo mismo sucede cuando una persona manualmente intenta armar una mano que cumpla ciertas condiciones. Por un lado elige las cartas de forma tal de conseguir cierta longitud de tipos. Luego toma las cartas que suman cierto PH. Puede hacerse en el orden inverso, incluso en un mismo paso, no importa, pero son las dos características determinantes.

La idea es aplicar este mismo razonamiento en el generador.

Luego, se pueden definir dos grupos o familias de restricciones:

- Las relacionadas con la distribución
- Las relacionadas con los puntos de honor

Esta división es útil para clasificar las restricciones. También ayuda a estudiar y plantear heurísticas que utilicen las restricciones para dirigir el resultado.

Para terminar, hay una tercera variante de restricción: las cartas preasignadas. Generar una mano con algunas cartas fijas o preasignadas puede que sea de las condiciones más restrictivas que podamos tener, sobre todo si la cantidad de cartas es alta. Este tipo de restricción es muy preciso comparado con los anteriores. Realmente es tan puntual que solo admite una forma de resolverlo eficientemente: repartiendo directamente la o las cartas al jugador en cuestión. Debido a ésta particularidad, la ubicamos en una tercera familia.

El uso de cartas preasignadas es una restricción fuerte y sumamente útil. La ventaja que tiene esta posibilidad es que se puede usar un generador así para la toma de decisiones durante el juego de Bridge (ver Cartas Preasignadas).

**Heurísticas de Distribución**

Sea un algoritmo de selección de cartas que consiste, primero en elegir un palo y luego en elegir una carta de ese palo. Es decir, se toma al azar un palo (un número entre 1 y 4) y a continuación se toma al azar un valor de carta (un número entre 1 y 13). Hacemos esto 13 veces y conseguimos una mano completa para un jugador. Solo debemos
tener cuidado de no elegir dos veces la misma carta, es decir, hay que validar que la carta no esté elegida con anterioridad.

Este esquema claramente elige una mano en forma totalmente aleatoria. Pero si de alguna manera aligeramos la etapa de selección del palo, podremos conseguir manos con cierta tendencia hacia determinado palo. De eso se trata cuando buscamos dirigir la distribución.

Una forma de lograr esto consiste en definir una probabilidad por palo y hacer que la elección del palo se base en esa probabilidad. Normalmente cada palo tendría probabilidad 1/4 (uniforme), pero si buscamos una mano con palo largo en picas, entonces aumentamos la probabilidad del palo pica. El mismo razonamiento pero a la inversa lo usamos para conseguir pocas cartas en un palo, asignando probabilidad < 1/4.

El siguiente es un algoritmo que encuentra un fallo utilizando este esquema:

```
Begin
    // todos los palos la misma probabilidad
    PP := 1
    PC := 1
    PD := 1
    PT := 1
    // excepto a uno que se pone en cero (el palo se elige random)
    segun random entre 1 y 4 hacer
        1: PP := 0
        2: PC := 0
        3: PD := 0
        4: PT := 0
    fin;
    // se ranquean las probabilidades
    P1 := PP;
    P2 := P1 + PC ;
    P3 := P2 + PD ;
    P4 := P3 + PT ;
    // se eligen las 13 cartas
    para I:= 1 TO 13 hacer
        P := random entre 0 y P4-1
        Si P < P1 entonces
            Se elige una carta de picas
        Sino si p < p2 entonces
            Se elige una carta de corazones
        Sino si p < p3 entonces
            Se elige una carta de diamantes
        Sino
            Se elige una carta de treboles
    endfor
Fin;
```

La probabilidad quedó expresada como un número entero >= 0. Si a los cuatro palos le ponemos 1, significa que cada palo tiene la misma chance (1/4) de salir. En cambio, si un palo tiene 0 y el resto 1, el de 0 tiene chance 0 y el resto 1/3. Esta es una forma cómoda de expresar probabilidad y estará dada por la proporción de la asignación de cada palo respecto de todos.
En este algoritmo, supongamos que la probabilidad 0 se asigna a PC, luego P2 es igual a P1 y nunca se ejecutará la acción que genera una carta de corazones. Claramente el algoritmo [4.C] genera fallos.

Otro ejemplo: conseguir una mano con palo largo en trébol. Para ello modificamos los números que expresan probabilidad de forma tal que trébol tenga más chance que los demás palos. Una forma sería:

\[
\begin{align*}
PP & := 1 \\
PC & := 1 \\
PD & := 1 \\
PT & := 2 \\
\end{align*}
\]

Aquí trébol tiene el doble de chance que los demás palos (2/5 contra 1/5). Recordemos que la probabilidad es \( P_x / (PP+PC+PD+PT) \) con \( x = P, C, D \) o \( T \).

Si cambiamos \( PT := 3 \), la chance es mayor aún para trébol (1/2 contra 1/6 en los otros palos).

**Refinamiento**

Rescribimos el algoritmo [4.C], generalizando para cualquier entero que expresa probabilidad por palo:

```
Algoritmo [4.D] ConseguirMano( PP, PC, PD, PT )
P1 := PP;     
P2 := P1 + PC; 
P3 := P2 + PD; 
P4 := P3 + PT; 
para i := 1 TO 13 hacer
    P := random(P4); //encuentra un número entre 0 y P4-1
    Si p < P1 entonces
        ElegirCarta(Pica)
    Sino si p < p2 entonces
        ElegirCarta(Corazon)
    Sino si p < p3 entonces
        ElegirCarta(Diamante)
    Sino
        ElegirCarta(Trebol)
    endfor
Fin;
```

ElegirCarta: elige una carta al azar del palo indicado (esto veremos que va a cambiar cuando incluyamos las heurísticas de puntos de honor).

Podemos forzar a que la suma de los enteros PP+PC+PD+PT sea 13. Esto indica que expresamos la probabilidad en factores de 1/13. Y, como una mano tiene 13 cartas, lo que en definitiva estamos diciendo es cantidad de cartas por palo.

Ejemplo:

\[
\begin{align*}
PP & := 0 \\
PC & := 3 \\
PD & := 3 \\
PT & := 7 \\
\end{align*}
\]
Indica que se quiere un fallo en picas, 7 cartas de trébol y 3 del resto. O lo que es lo mismo: una distribución 0.3.3.7 con los palos p.c.d.t.

En una palabra, usamos la distribución como forma de expresar la probabilidad de las cartas.

Con

\[
\begin{align*}
PP & := 3 \\
PC & := 3 \\
PD & := 3 \\
PT & := 4
\end{align*}
\]

se busca una mano balanceada

Y con

\[
\begin{align*}
PP & := 4 \\
PC & := 4 \\
PD & := 4 \\
PT & := 1
\end{align*}
\]

se busca una tricolor.

Supongamos que queremos generar una distribución 7.x.x.x con el palo largo en picas. Entonces:

**Algoritmo [4.E] 7_Picas**

// se define cantidad de cartas por palo (uno de ellos con 7)

\[
\begin{align*}
PP & := 7 \\
PC & := \text{Random entre } 0 \text{ y } (13 - PP) \\
PD & := \text{Random entre } 0 \text{ y } (13 - PP - PC) \\
PT & := 13 - PP - PC - PD \\
// claramente PP+PC+PD+PT = 13 \\
\text{Result} & := \text{ConseguirMano( PP, PC, PD, PT)};
\end{align*}
\]

Fin;

Este método dirige bien el resultado, sobre todo en los casos de distribuciones desbalanceadas y fuertemente desbalanceadas. Por ejemplo, si queremos generar una mano con distribución 13.0.0.0, éste algoritmo la encuentra en un solo intento. Por las pruebas que hicimos un algoritmo GAT [3.A] nunca encuentra una distribución 13.0.0.0.

Testeando 5000 corridas del algoritmo anterior contra uno GAT, buscando una distribución 7.x.x.x, los resultados fueron:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Número de intentos promedio ($x$)</td>
<td>3.36</td>
<td>13.56</td>
</tr>
<tr>
<td>Desviación standard ($S$)</td>
<td>2.83</td>
<td>12.85</td>
</tr>
</tbody>
</table>

$x$ es la media de los 5000 intentos y $S$ es la desviación standard.

[4.E] es unas 4 veces más rápido. Además $S$ demuestra que converge mucho mejor que GAT.

Estos son los resultados de 5000 corridas con algunas distribuciones desbalanceadas.
<table>
<thead>
<tr>
<th></th>
<th>6.x.x.x</th>
<th>8.x.x.x</th>
<th>9.x.x.x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x$</td>
<td>$S$</td>
<td>$x$</td>
</tr>
<tr>
<td>GAT [3.A]</td>
<td>4.52</td>
<td>4.03</td>
<td>53.90</td>
</tr>
</tbody>
</table>

En cambio en las distribuciones balanceadas, los números se invierten. Es mejor GAT que [4.E]:

<table>
<thead>
<tr>
<th></th>
<th>4.3.3.3</th>
<th>4.4.3.2</th>
<th>5.3.3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x$</td>
<td>$S$</td>
<td>$x$</td>
</tr>
<tr>
<td>GAT [3.A]</td>
<td>14.13</td>
<td>13.64</td>
<td>6.26</td>
</tr>
</tbody>
</table>

Este efecto contrario se debe a que las probabilidades por palo se parecen (se achatan) en los casos balanceados, entonces el resultado no es el esperado.

**Variante**

Se hace necesario encontrar un método mejor para dirigir la distribución.

Una variante del método anterior consiste en ir modificando la probabilidad de cada palo a medida que se van eligiendo las cartas. Esto parece sensato: las cartas se eligen de a una en un ciclo, y a medida que van saliendo, la probabilidad de los 4 palos en la siguiente iteración no debería ser la misma que la de la iteración actual. Veamos esto en un ejemplo:

- *Generar un semifalso (distribución 1.x.x.x)*

Un posible estado inicial del algoritmo sería:

```plaintext
PP := 3
PC := 1
PD := 5
PT := 4
```

Al cabo de i iteraciones se elige un corazón, supongamos i = 4. Claramente no tiene sentido que la probabilidad de corazones continúe siendo 1/13 para las restantes 13-4 iteraciones. Lo ideal sería que la probabilidad de corazones cambie a 0 para que no se elija ninguno más.

Éste es el razonamiento a seguir con cada uno de los palos. Expresamos la probabilidad de ellos indicando cuántas cartas por palo se buscan y luego, al ir eligiendo las cartas, se resta en uno el contador respectivo.

Otro ejemplo:

- *Generar una distribución 4.3.3.3*

Un posible estado inicial sería:

```plaintext
PP := 3
PC := 3
PD := 3
PT := 4
```

Y una posible corrida:
En la 1\textsuperscript{a} iteración se elige una carta de corazón => PC se decrementa y pasa a valer 2
En la 2\textsuperscript{a} iteración la carta es de trébol => PT pasa a valer 3
En la 3\textsuperscript{a} iteración se elige una carta de corazón => PC = 1

... En la 9\textsuperscript{a} iteración se elige una carta de corazón => PC = 0. Ahora corazón tiene la cantidad de cartas que buscaba y no van a salir más corazones porque su probabilidad pasó a ser 0.
Así hasta la iteración 12 en que las variables valen

\begin{verbatim}
PP := 0
PC := 0
PD := 1
PT := 0
\end{verbatim}

La última iteración elige diamante (todos los otros palos tienen probabilidad 0).

Claramente este método converge directamente en la distribución.

Luego, redefinimos el algoritmo [4.D] como:

\begin{verbatim}
Begin
para I := 1 TO 13 hacer
   // el ranqueo se hace en cada pasada para que la probabilidad
   // cambie
   P1 := PP;
P2 := P1 + PC ;
P3 := P2 + PD ;
P4 := P3 + PT ;

   P := random(P4) ; //encuentra un número entre 0 y P4-1
   Si P < P1 entonces
       ElegirCarta(Pica)
       PP := PP - 1 ; // se resta en uno el contador
   Sino si p < p2 entonces
       ElegirCarta(Corazon)
       PC := PC - 1 ; // se resta en uno el contador
   Sino si p < p3 entonces
       ElegirCarta(Diamante)
       PD := PD - 1 ; // se resta en uno el contador
   Sino
       ElegirCarta(Trebol)
       PT := PT - 1 ; // se resta en uno el contador
   endif
endfor
Fin;
\end{verbatim}

Con este cambio, los test anteriores arrojan los siguientes valores:
<table>
<thead>
<tr>
<th></th>
<th>4.3.3.3</th>
<th></th>
<th>4.4.3.2</th>
<th></th>
<th>5.3.3.2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Algoritmo [4.F]</td>
<td>X</td>
<td>S</td>
<td>X</td>
<td>S</td>
<td>x</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

La mejora es evidente. Converge directamente.
Y con las distribuciones desbalanceadas pasa lo mismo. Todos los promedios dan un intento y $S = 0$. 
Capítulo 5

La unión hace la fuerza.

Manos para más de un jugador


Primero vamos a definir un algoritmo que elige cartas para 4 jugadores. Es una extensión de lo que venimos describiendo en párrafos anteriores. Consiste en repartir las manos en forma secuencial, jugador por jugador, corriendo el algoritmo [4.F] una vez para cada jugador. El juego de Bridge tiene como particularidad que se reparten todas las cartas del mazo. Eso determina que si ya repartí las cartas de tres jugadores, lo que queda en el mazo es precisamente la mano del jugador número 4. Entonces, en realidad alcanza con ejecutar el algoritmo tres veces con los parámetros correspondientes a tres de los cuatro jugadores y el 4to jugador recibirá una mano que consiste en las cartas que quedaron en el mazo sin repartir.

El principal problema de un esquema de más de un jugador es que se consuman la o las cartas antes de tiempo. Es decir que en la mano del 2do o 3º jugador puede suceder que haya suces que no tengan la cantidad de cartas necesarias para completar la distribución pedida. Pensemos en un ejemplo en el que hay que asignar dos distribuciones con palos largos: 7.3.3.0 para Norte y 8.1.2.1 para Oeste (sin especificar palos). La mano de Norte se elige con problemas (supongamos con palo largo en picas elegido al azar). Luego se comienza a dar las cartas a Oeste y nuevamente por azar sale palo largo en picas. En el momento en que se intenta dar la 7ma carta de picas a Oeste, el algoritmo detecta que no quedan más picas!. ¿Qué sucedió? El palo quedó exhausto y se consumieron todas sus cartas antes de terminar la distribución.

En estos casos se hace necesario incluir algún chequeo y solución. Lo que proponemos es reformular las probabilidades de forma tal que tengan en cuenta las cartas que están en el mazo sin asignar. Con esto se logra que el algoritmo encuentre si o si una mano. Puede que no sea la mano que se busca, pero el algoritmo finalmente encuentra una mano y termina.

En el ejemplo anterior, al detectar que picas se quedó sin cartas, recalcula los valores PT, PC y PD basándose en la cantidad de cartas que falta asignar y el valor PP lo fuerza a 0 para que no se vuelvan a elegir picas. Recordemos que es perfectamente válido que el generador no encuentre una mano en algún intento. No es el objetivo converger siempre en el primer intento.

Otra consideración importante es el orden de los jugadores. En una primera aproximación parecía suficiente que el orden se elija al azar. En un enfoque posterior, resultó mucho mejor el hecho de darles prioridad a los jugadores que tengan restricciones. Pensemos que las restricciones pueden existir para algunos jugadores si y para otros no. Podemos tener un sinfín de consultas con uno, dos o tres jugadores parametrizados y el resto libre. En estos casos resulta productivo generar primero las manos de los jugadores con restricciones y luego la de los otros. Si, por ejemplo, solo Norte y Oeste tiene restricciones, el algoritmo genera primero la mano de Norte y luego la de Oeste (o al revés ya que los toma al azar). A continuación el tercer jugador se elige al azar entre Sur y Este (ambos sin restricciones). Por último, como ya mencionamos, el 4to jugador recibe las cartas que quedaron en el mazo sin asignar.

Smartgen

Aplicando los conceptos anteriores, definimos el ciclo principal del generador dirigido:
Algoritmo [5.A] SmartGen
Repetir
    Se eligen al azar tres jugadores dándole prioridad a los que tienen restricciones
    Mano1 := GenerarMano( Rj1 ) ;
    Mano2 := GenerarMano( Rj2 ) ;
    Mano3 := GenerarMano( Rj3 ) ;
    Mano4 := resto del mazo sin asignar
    Hasta Cumple(Mano1,Rj1) AND Cumple(Mano2,Rj2) AND Cumple(Mano3,Rj3) AND Cumple(Mano4,Rj4); 
Fin ;

Rj; son las restricciones del jugador i
GenerarMano toma las restricciones del jugador y llama a ConseguirMano
Cumple tomas las restricciones de un jugador y evalúa si la mano las cumple.

El siguiente es un algoritmo similar al [4.F] pero con los cambios mencionados para contemplar varios jugadores:

Algoritmo [5.B] ConseguirMano( PP, PC, PD, PT )
para I:= 1 TO 13 hacer
    // se fuerzen probabilidades en 0 para los palos exustos
    si NO quedan cartas de pica entonces PP := 0 ;
    si NO quedan cartas de corazón entonces PC := 0 ;
    si NO quedan cartas de diamante entonces PD := 0 ;
    si NO quedan cartas de trébol entonces PT := 0 ;
    // se reasigna cantidad de cartas para que la mano se termine
    si PP+PC+PD+PT = 0 entonces begin
        PP := 13 - nro cartas que restan asignar de pica
        PC := 13 - nro cartas que restan asignar de corazón
        PD := 13 - nro cartas que restan asignar de diamante
        PT := 13 - nro cartas que restan asignar de trébol
    fin ;
    // ranqueo
    P1 := PP;
    P2 := P1 + PC ;
    P3 := P2 + PD ;
    P4 := P3 + PT ;
    P := random(P4) ; // encuentra un número entre 0 y P4-1
    si P < P1 entonces
        ElegirCarta(Pica)
        PP := PP - 1 ; // se resta en uno el contador
    sino si p < p2 entonces
        ElegirCarta(Corazon)
        PC := PC - 1 ; // se resta en uno el contador
    sino si p < p3 entonces
        ElegirCarta(Diamante)
        PD := PD - 1 ; // se resta en uno el contador
    sino
        ElegirCarta(Trebol)
Generación automática de manos de Bridge con restricciones

PT := PT - 1 ; // se resta en uno el contador
endif
endfor
Fin;

Volvamos al ejemplo de Norte: 7.x.x.x y Oeste: 8.x.x.x. Testeando el algoritmo [5.B] contra GAT en 5000 corridas, resultó:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Número de intentos promedio ($\mu$)</td>
<td>1.37</td>
<td>100.86</td>
</tr>
<tr>
<td>Desviación standard ($\sigma$)</td>
<td>0.71</td>
<td>99.11</td>
</tr>
</tbody>
</table>

A continuación vamos a modificar el test de manera de hacerlo paulatinamente más difícil, incluyendo restricciones a más jugadores. Se verá que el algoritmo [5.B] encuentra soluciones mucho mejor que GAT, incluso cuando GAT no las encuentra.

Agregamos al ejemplo anterior una restricción para Sur: distribución 6.x.x.x. Sobre 500 corridas, resultó:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Número de intentos promedio ($\mu$)</td>
<td>2.83</td>
<td>381.32</td>
</tr>
<tr>
<td>Desviación standard ($\sigma$)</td>
<td>2.21</td>
<td>391.92</td>
</tr>
</tbody>
</table>

Ahora cambiamos la restricción de Sur: distribución 4.3.3.3. Sobre 100 corridas, resultó:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Número de intentos promedio ($\mu$)</td>
<td>2.12</td>
<td>2024.38</td>
</tr>
<tr>
<td>Desviación standard ($\sigma$)</td>
<td>1.63</td>
<td>2017.94</td>
</tr>
</tbody>
</table>

Nuevamente cambiamos la restricción de Sur por distribución 0.x.x.x. Sobre 100 corridas, resultó:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Número de intentos promedio ($\mu$)</td>
<td>2.95</td>
<td>606.74</td>
</tr>
<tr>
<td>Desviación standard ($\sigma$)</td>
<td>2.31</td>
<td>585.34</td>
</tr>
</tbody>
</table>

Ahora agregamos una restricción para Este: distribución 7.x.x.x. Los 4 jugadores tienen restricciones. Sobre 100 corridas, resultó:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Número de intentos promedio ($\mu$)</td>
<td>11.75</td>
<td>26854.46(*)</td>
</tr>
<tr>
<td>Desviación standard ($\sigma$)</td>
<td>9.66</td>
<td>10765.17(*)</td>
</tr>
</tbody>
</table>

(*) GAT en 31 corridas no encontró mano (sobre 30000 intentos por corrida)

Puntos de Distribución

Hasta aquí hemos explicado un algoritmo eficiente para generar manos, empleando como restricción una distribución x.x.x.x para cada jugador.
Es necesario también que el generador pueda resolver consultas basadas en PD. En los siguientes párrafos del capítulo demostraremos que las restricciones sobre PD pueden expresarse como una distribución x.x.x.x. Entonces, este tipo de restricciones también se pueden resolver usando el algoritmo [5.B].

Veamos. Los PD cuantifican el desequilibrio de la mano: 0 PD es balanceada, 9 PD es la distribución 13.0.0.0.

1 PD se consigue \( \Rightarrow \) con un dubletón (2.x.x.x).
2 PD se consiguen \( \Leftrightarrow \) con un semifallo (1.x.x.x) ó con dos doubletones (2.2.x.x).
3 PD se consiguen \( \Leftarrow \) con un fallo (0.x.x.x), ó con un semifallo y un dubletón (1.2.x.x) ó con tres dubletones (2.2.2.x). En todos los casos \( x > 2 \), sino el PD aumenta.

Es decir, podemos expresar los PD como distribuciones. Es más, podemos definir muy fácilmente por extensión todos los posibles PD y las distribuciones equivalentes:

<table>
<thead>
<tr>
<th>PD</th>
<th>Distribución</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.x.x.x</td>
</tr>
<tr>
<td>2</td>
<td>1.x.x.x</td>
</tr>
<tr>
<td>2</td>
<td>2.2.x.x</td>
</tr>
<tr>
<td>3</td>
<td>0.x.x.x</td>
</tr>
<tr>
<td>3</td>
<td>1.2.x.x</td>
</tr>
<tr>
<td>3</td>
<td>2.2.2.x</td>
</tr>
<tr>
<td>4</td>
<td>0.2.x.x</td>
</tr>
<tr>
<td>4</td>
<td>1.1.x.x</td>
</tr>
<tr>
<td>4</td>
<td>1.2.2.x</td>
</tr>
<tr>
<td>5</td>
<td>0.1.x.x</td>
</tr>
<tr>
<td>5</td>
<td>0.2.2.x</td>
</tr>
<tr>
<td>5</td>
<td>1.1.2.x</td>
</tr>
<tr>
<td>6</td>
<td>0.0.x.x</td>
</tr>
<tr>
<td>6</td>
<td>0.1.2.x</td>
</tr>
<tr>
<td>6</td>
<td>1.1.1.x</td>
</tr>
<tr>
<td>7</td>
<td>0.0.2.x</td>
</tr>
<tr>
<td>7</td>
<td>0.1.1.x</td>
</tr>
<tr>
<td>8</td>
<td>0.0.1.x</td>
</tr>
<tr>
<td>9</td>
<td>0.0.0.x</td>
</tr>
</tbody>
</table>

Si por ejemplo hay que generar una mano con 8 PD, convertimos PD a distribución con la tabla anterior y obtenemos 0.0.1.x. La x se convertirá en un 12 (la suma de la distribución debe ser 13) y así llegamos a la distribución 0.0.1.12. Luego aplicamos el algoritmo [5.B] y generamos la mano.

Si en cambio, tenemos que conseguir 5 PD, fijándonos en la tabla vemos que hay tres posibles distribuciones: 0.1.x.x, 0.2.2.x y 1.1.2.x. Entonces tomamos una de ellas al azar, reemplazamos las x por valores adecuados y corremos el algoritmo [5.B] para generar la mano.

Básicamente la idea consiste en buscar los PD en la tabla, y elegir una distribución (si hay más de una, la tomo al azar). Luego reemplazamos las x por valores al azar (mayores que 2, para no sumar más PD) y tal que los 4 valores sumados den 13:
Algoritmo [5.C] PD-a_Distribucion( PD, Distribución )
Se busca PD en la tabla
Se toma una distribución relacionada. Si hay varias, se elige una de ellas al azar
Asigno valores a los x:
con valores al azar > 2
y tal que la suma de los 4 valores de la distribución sea 13
fin;


<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x S</td>
<td>X S</td>
</tr>
<tr>
<td>0</td>
<td>1  0</td>
<td>14.54  13.51</td>
</tr>
<tr>
<td>1</td>
<td>1  0</td>
<td>3.44   2.90</td>
</tr>
<tr>
<td>2</td>
<td>1  0</td>
<td>2.79   2.14</td>
</tr>
<tr>
<td>3</td>
<td>1  0</td>
<td>4.80   4.04</td>
</tr>
<tr>
<td>4</td>
<td>1  0</td>
<td>16.01  16.73</td>
</tr>
<tr>
<td>5</td>
<td>1  0</td>
<td>84     82.30</td>
</tr>
<tr>
<td>6</td>
<td>1  0</td>
<td>893.01 833.29</td>
</tr>
<tr>
<td>7</td>
<td>1  0</td>
<td>26 veces sobre 100 no encontró mano</td>
</tr>
<tr>
<td>8</td>
<td>1  0</td>
<td>Nunca encontró mano</td>
</tr>
</tbody>
</table>

Estos son test con dos jugadores (100 corridas):

<table>
<thead>
<tr>
<th>Restricciones</th>
<th>Resultados</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Heurísticas de Puntos de Honor

Intentamos lograr un algoritmo que dado cierto valor en PH, indique cuáles son las cartas que hay que elegir para cumplirlo.

El generador que estamos proponiendo, cómo vimos en los párrafos anteriores, primero tiene en cuenta la distribución. Con ello calcula un número de cartas por palo y usando el algoritmo [5.B] elige las 13 cartas del jugador. Definimos una función ElegirCarta que elige al azar una carta cualquiera del palo indicado (sin tener en cuenta el valor en honores de las cartas). Entonces, lo que proponemos ahora es modificar ElegirCarta de manera de que elija las cartas teniendo en cuenta el PH que hay que conseguir.
Como expresamos anteriormente, si una persona trata de armar una mano *manualmente* en la que se deba cumplir cierta cantidad de PH, primero elige las cartas de honores de forma tal que la suma sea igual al PH en cuestión y luego completa la mano con cartas bajas.

Nos basamos en esta idea para *dirigir* el resultado por PH: primero elegir las cartas adecuadas que sumen cierto PH y luego completar la mano con cartas bajas (que no van a modificar el PH).

Veamos un ejemplo y un posible desarrollo:

*Generar una mano 4.3.3.3 con 8 PH*

1. Repartimos los 8 PH en PH por palo:
   - 4 PH para el palo 1
   - 2 PH para el palo 2
   - 2 PH para el palo 3
   - 0 PH para el palo 4

2. Convertimos los puntos en cartas:
   - 4 PH $\Rightarrow$ A
   - 2 PH $\Rightarrow$ Q

   y resulta

   palo 1: AXXX
   palo 2: QXXX
   palo 3: QXX
   palo 4: XXX

3. Reemplazamos las x por cartas bajas, consiguiendo así la mano.

Veamos en detalle cada paso:

En el *paso* 2, convertimos los PH en cartas. 4 PH es un A, 3 PH es una K, etc.

Hay casos en que existe más de una combinación de cartas que satisfacen el PH.

Por ej: hay dos formas de elegir cartas del mismo palo para 7 PH:

AK (4+3)

ó

AQJ (4+2+1)

En un caso así, elijo al azar entre una variante u otra.

Ahora defino todas las combinaciones posibles de PH y honores por palo:

<table>
<thead>
<tr>
<th>PH</th>
<th>Honores</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>J</td>
</tr>
<tr>
<td>2</td>
<td>Q</td>
</tr>
<tr>
<td>3</td>
<td>K</td>
</tr>
<tr>
<td>3</td>
<td>Q, J</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>K, Q</td>
</tr>
<tr>
<td>5</td>
<td>A, J</td>
</tr>
</tbody>
</table>
Con esta tabla tengo asegurado qué cartas elegir para alcanzar el PH indicado.

El paso 3 es trivial: repartir cartas bajas hasta llegar a las 13 cartas que forman la mano completa de un jugador.

Veamos ahora el paso 1: El interrogante que surge es ¿cómo repartimos PH total en PH por palo?

Debemos definir una relación entre la cantidad de cartas por palo y el PH posible por palo. La idea se ve mejor en un ejemplo: ¿cuantos puntos de honor puedo conseguir con dos cartas del mismo palo?

- 0 PH: si las dos cartas son bajas
- 1 PH: J y una carta baja
- 2 PH: Q y una carta baja
- 3 PH: K y una carta baja
- 4 PH: A y una carta baja
- 5 PH: A y J
- 6 PH: A y Q
- 7 PH: A y K
- 8 PH: imposible con dos cartas del mismo palo

Luego, con dos cartas podemos tener entre 0 y 7 PH.

Ahora podemos extender esto a todos los valores de cartas posibles y llegamos a la siguiente tabla:

<table>
<thead>
<tr>
<th>Nro. de cartas de un palo</th>
<th>Rango posible de PH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
</tr>
<tr>
<td>4</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.10</td>
</tr>
<tr>
<td>6</td>
<td>0.10</td>
</tr>
<tr>
<td>7</td>
<td>0.10</td>
</tr>
<tr>
<td>8</td>
<td>0.10</td>
</tr>
<tr>
<td>9</td>
<td>0.10</td>
</tr>
<tr>
<td>10</td>
<td>1.10</td>
</tr>
<tr>
<td>11</td>
<td>3.10</td>
</tr>
<tr>
<td>12</td>
<td>6.10</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
</tr>
</tbody>
</table>

Vemos que con 10 cartas el rango posible de PH va de 1 a 10. Significa que en 10 cartas como mínimo seguro que hay un honor (son 9 las cartas no-honores). En el caso de mínimo puntaje, ese honor es una J, por eso el rango comienza desde 1. El caso de máxima es tener los 4 honores, es decir, 10 puntos. Así quedó definido el rango 1 a 10.

Lo mismo pasa con 11 cartas. Seguro que dos o más son honores. El caso de mínima es J y Q, es decir 3 PH.
Luego, con la tabla [T.2] tenemos el conocimiento necesario para repartir PH por palo:

**Algoritmo [5.D] RepartirPHPorPalos( minPH, maxPH, PP, PC, PD, PT, PhP, PhC, PhD, PhT) Repetir**  
PhP = se toma el rango de [T.2] según PP y se elige un nro. al azar del rango  
PhC = se toma el rango de [T.2] según PC y se elige un nro. al azar del rango  
PhD = se toma el rango de [T.2] según PD y se elige un nro. al azar del rango  
PhT = se toma el rango de [T.2] según PT y se elige un nro. al azar del rango  
Hasta PhP+PhC+PhD+PhT in [minPH..maxPH]  
Fin ;

Como vemos en [5.D] el PH total está expresado en un rango ya que son comunes, en los libros de Bridge, las restricciones del tipo PH entre tal y tal número. En particular, si minPH y maxPH son iguales, tenemos el caso de PH total exacto.

Los otros parámetros de entrada son PP, PC, PD, PT que representan la cantidad de cartas por cada uno de los palos -la distribución-.

Los parámetros de salida son PhP, PhC, PhD, PhT que representan los valores de PH para cada palo.

**Observaciones**

El algoritmo anterior siempre termina, salvo cuando la cantidad de cartas es incoherente con el PH a repartir. Claramente, no se puede tener una distribución 11.0.1.1 con 2 PH. entre las 11 cartas del mismo palo seguro que hay dos honores y nunca pueden sumar 2, como mínimo suman 3 (J+Q).

Entonces, redefinimos el algoritmo [5.B] ConseguirMano para que tenga en cuenta no solo la heurística de distribución sino también la de puntos de honor:

**Algoritmo [5.E] ConseguirMano( PP, PC, PD, PT, minPH, maxPH )**  
RepartirPHPorPalos( minPH, maxPH, PP, PC, PD, PT, PhP, PhC, PhD, PhT)  
para I:= 1 TO 13 hacer  
// se fuerza probabilidad en 0 para los palos exauutos  
si NO quedan cartas de pica entonces PP := 0 ;  
si NO quedan cartas de corazón entonces PC := 0 ;  
si NO quedan cartas de diamante entonces PD := 0 ;  
si NO quedan cartas de trébol entonces PT := 0 ;  
// se reasigna cantidad de cartas para que la mano se termine  
si PP+PC+PD+PT = 0 entonces begin  
PP := 13 - nro cartas que restan asignar de pica  
PC := 13 - nro cartas que restan asignar de corazón  
PD := 13 - nro cartas que restan asignar de diamante  
PT := 13 - nro cartas que restan asignar de trébol  
fin ;  
// se ranquean los nros. De cartas  
P1 := PP ;  
P2 := P1 + PC ;  
P3 := P2 + PD ;
P4 := P3 + PT;
P := nro al azar entre 0 y P4-1
Si P < P1 entonces
  ElegirCarta(Pica, PH)
  PP := PP - 1; // se resta en uno el contador
Sino si p < p2 entonces
  ElegirCarta(Corazon, PhC)
  PC := PC - 1; // se resta en uno el contador
Sino si p < p3 entonces
  ElegirCarta(Diamante, PhD)
  PD := PD - 1; // se resta en uno el contador
Sino
  ElegirCarta(Trebol, PhT)
  PT := PT - 1; // se resta en uno el contador
fin si
Fin para
Fin;

Y el auxiliar:

Algoritmo [5.5] ElegirCarta( Palo, PH )
  // primero se elijen los honores luego las cartas bajas
  Si todavía no se elegieron todos los honores necesarios para llegar a PH entonces
  Se buscan los honores necesarios en la tabla [T.1] indizada por PH. Si hay más de una combinación de honores entonces en una forma random se elige una al azar
  end si
  Marco el honor como elegido
Sino
  Elijo y marco al azar carta baja
Fin si
Fin;

Las siguientes son algunas condiciones de ejemplo comparadas con GAT (valores sobre 200 corridas). Son solo restricciones sobre PH:

<table>
<thead>
<tr>
<th>Restricciones</th>
<th>Norte &gt;= 25 PH</th>
<th>Cada jugador con 10PH</th>
<th>Norte: 5PH en pica, 5PH en trebó, 2PH en diamante y 1PH en corazón</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algoritmo</td>
<td>x</td>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>1.35</td>
<td>0.70</td>
<td>162.33</td>
</tr>
<tr>
<td>GAT [3.A]</td>
<td>2147.01</td>
<td>2225.37</td>
<td>882.73</td>
</tr>
</tbody>
</table>

(*) GAT en 54 corridas no encontró mano (sobre 30000 intentos por corrida)
Cartas preassignadas

La asignación de cartas específicas como parte de la condición es un campo que consideramos poco explotado por generadores existentes, y que da como ya se dijo múltiples ventajas. Más allá del hecho intuitivo de que permite una mayor flexibilidad en el lenguaje de condiciones usado, puede prestar servicios a la hora del juego. En efecto, un programa jugador podría tener en cuenta durante el carteo las condiciones inducidas a partir del remate en el resto de los jugadores, y juntamente con las cartas que ya fueron jugadas por estos, generar manos adecuadas (i.e. compatibles) con esa información y así poseer mejor información del contenido de esas manos.

El obtener manos con cartas preassignadas no requiere en principio de un algoritmo particular. Se puede resolver en forma simple asignando la o las cartas directamente al jugador indicado. En este sentido, lo óptimo consiste en realizar esta asignación, para los cuatro jugadores, antes del ciclo normal de reparto con restricciones. De esta manera garantizamos que las cartas involucradas no se elijan antes para otro jugador con lo cual podría llegar a bloquearse la búsqueda, como la misma intuición lo indica.

Reformulamos el algoritmo [5.4]:

Algoritmo [5.6] SmartGen
    Se reparten las cartas preassignadas a los jugadores respectivos
    // luego la etapa de reparto dirijido
    Repetir
    Se eligen al azar tres jugadores dándole prioridad a los que tienen restricciones
    Mano1 := GenerarMano( RJ1 )
    Mano2 := GenerarMano( RJ2 )
    Mano3 := GenerarMano( RJ3 )
    Mano4 := resto del mazo sin asignar
    Hasta Cumple(Mano1,RJ1) AND Cumple(Mano2,RJ2) AND Cumple(Mano3,RJ3) AND Cumple(Mano4,RJ4);
    Fin;

Validación de las restricciones

Con el fin de obtener un generador eficiente, es decir, que encuentre la mano buscada lo antes posible, es importante estudiar el impacto (en tiempo) que tiene la etapa de verificación de las restricciones. Nos referimos a la función Cumple(restricciones_jugador, mano_jugador)

Este punto vale para cualquier generador de manos, sea GAT o no.

Intentamos que se detecte, lo antes posible, que una mano no cumple las restricciones -si es que es una mano negativa-, para poder reintentar -es decir, repartir nuevamente- cuanto antes. Tratamos de producir el backtracking en el momento más temprano que se pueda.

Para ello, planteamos dos temas:

1. Ejecutar las verificaciones en forma eficiente
2. Ordenar los parámetros a verificar de forma tal de validar primero los más restrictivos.

Como dijimos al principio del capítulo, la función Cumple es una secuencia de condiciones que validan todos los parámetros posibles. En cuanto una condición falla, la función retorna falso.
La mayoría de las condiciones que se evalúan son rangos de números enteros: que el PH de la mano esté en tal rango, que el PH de tal palo esté en tal rango, que la cantidad de cartas de tal palo esté en tal rango, etc. Obviamente el calculo de esas propiedades debe hacerse una sola vez y en la forma más eficiente posible.

 Uno de los parámetros más lento para verificar es el patrón de una distribución. Si la restricción es Norte con distribución w.x.y.z sin especificar palo, tendremos que comparar la distribución de la mano encontrada contra w.x.y.z sin importar el orden en que se sucedan w, x, y, z. Una forma de hacerlo es ordenando numéricamente w, x, y, z en ambas distribuciones y comparar los cuatro valores una sola vez.

 Si la distribución es más genérica como 9.x.x.x (en la que x.x.x es cualquier combinación tal que sume 13-9) solo tendremos que verificar que la distribución encontrada tenga un 9.
 En el caso de una 3.x.x.x tendremos que verificar que el 3 aparece dos veces en la distribución encontrada.

 Veamos ahora una forma más eficiente de verificar una distribución.

 Consideremos [DEA 03] la desviación standard S de una distribución w₁, w₂, w₃, w₄, calculada del siguiente modo:

\[ S = \sqrt{\frac{1}{4} \sum_{i=1}^{4} (w_i - x)^2} \]

con \( x = \frac{13}{4} = 3.25 \) (promedio del largo de un palo)

La siguiente tabla muestra los valores calculados para todas las distribuciones posibles:

<table>
<thead>
<tr>
<th>Distribución</th>
<th>S</th>
<th>Suma de cuadrados</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.3.3</td>
<td>0.5000</td>
<td>43</td>
</tr>
<tr>
<td>4.4.3.2</td>
<td>0.9574</td>
<td>45</td>
</tr>
<tr>
<td>5.3.3.2</td>
<td>1.2583</td>
<td>47</td>
</tr>
<tr>
<td>4.4.4.1, 5.4.2.2</td>
<td>1.5000</td>
<td>49</td>
</tr>
<tr>
<td>5.4.3.1</td>
<td>1.7078</td>
<td>51</td>
</tr>
<tr>
<td>6.3.2.2</td>
<td>1.8930</td>
<td>53</td>
</tr>
<tr>
<td>5.5.2.1, 6.3.3.1</td>
<td>2.0616</td>
<td>55</td>
</tr>
<tr>
<td>5.4.4.0, 6.4.2.1</td>
<td>2.2166</td>
<td>57</td>
</tr>
<tr>
<td>5.5.3.0</td>
<td>2.3629</td>
<td>59</td>
</tr>
<tr>
<td>7.2.2.2, 6.4.3.0</td>
<td>2.5000</td>
<td>61</td>
</tr>
<tr>
<td>7.3.2.1, 6.5.1.1</td>
<td>2.6300</td>
<td>63</td>
</tr>
<tr>
<td>6.5.2.0</td>
<td>2.7538</td>
<td>65</td>
</tr>
<tr>
<td>7.3.3.0, 7.4.1.1</td>
<td>2.8723</td>
<td>67</td>
</tr>
<tr>
<td>7.4.2.0</td>
<td>2.9860</td>
<td>69</td>
</tr>
<tr>
<td>6.6.1.0, 8.2.2.1</td>
<td>3.2016</td>
<td>73</td>
</tr>
<tr>
<td>7.5.1.0, 8.3.1.1</td>
<td>3.3040</td>
<td>75</td>
</tr>
<tr>
<td>8.3.2.0</td>
<td>3.4034</td>
<td>77</td>
</tr>
<tr>
<td>8.4.1.0</td>
<td>3.5940</td>
<td>81</td>
</tr>
<tr>
<td>7.6.0.0</td>
<td>3.7749</td>
<td>85</td>
</tr>
<tr>
<td>9.2.1.1</td>
<td>3.8622</td>
<td>87</td>
</tr>
<tr>
<td>8.5.0.0, 9.2.2.0</td>
<td>3.9476</td>
<td>89</td>
</tr>
<tr>
<td>9.3.1.0</td>
<td>4.0311</td>
<td>91</td>
</tr>
<tr>
<td>9.4.0.0</td>
<td>4.2720</td>
<td>97</td>
</tr>
<tr>
<td>10.1.1.1</td>
<td>4.5000</td>
<td>103</td>
</tr>
<tr>
<td>10.2.1.0</td>
<td>4.5735</td>
<td>105</td>
</tr>
<tr>
<td>10.3.0.0</td>
<td>4.7170</td>
<td>109</td>
</tr>
<tr>
<td>11.1.1.0</td>
<td>5.1881</td>
<td>123</td>
</tr>
<tr>
<td>11.2.0.0</td>
<td>5.2519</td>
<td>125</td>
</tr>
<tr>
<td>12.1.0.0</td>
<td>5.8524</td>
<td>145</td>
</tr>
<tr>
<td>13.0.0.0</td>
<td>6.5000</td>
<td>169</td>
</tr>
</tbody>
</table>

Tabla 1.3

Pag. 38
Como vemos, las distribuciones balanceadas tienen la desviación estándar más baja. Más precisamente $S < 1.26$. De esta forma podemos verificar si una mano es balanceada o no sin comparar distribuciones. Es más, para simplificar los cálculos, en vez de $S$ usamos la suma de los cuadrados.

Esta propiedad de las distribuciones también se puede usar para verificar la igualdad de dos patrones cualquiera, solo hay que considerar que hay casos en que la suma de cuadrados da el mismo valor para distribuciones diferentes, como 7.2.2.2 y 6.4.3.0. Considerando esos casos especiales –que son pocos–, la validación será más eficiente que ordenar listas de números. Para comprobar esto, probamos comparaciones de patrones usando ordenamiento de listas contra suma de cuadrados. Este último es unas 4 veces más rápido.

Con respecto al orden en que realizamos las verificaciones, proponemos el siguiente:

1. Verificar las cartas preasignadas
2. Puntos de honor (totales)
3. Puntos de honor x palo
4. Si la mano es balanceada
5. patrón de distribución
6. Rango de cartas x palo
7. Puntos de distribución

Desde ya que cada item solo se debe verificar si forma parte de la restricción.

El item 1 es el más restrictivo de todos, por eso lo debemos chequear en primer término.
Los items 2 y 3 se refieren a los puntos de honor
Los restantes se refieren a distribución

Hemos comprobado que si validamos primero los puntos de honor y luego los de distribución, el tiempo consumido por la función Cumple es menor. En una palabra es mejor validar los PH antes que la distribución. Esto lo hemos podido comprobar en forma empírica a lo largo del desarrollo.
El hecho tiene que ver con que las condiciones sobre puntos de honor son más restrictivas que las condiciones sobre distribución. Esta afirmación se puede confirmar observando lo siguiente.

Uso de tablas

Según la American Contract Bridge League [PLA 03], [ACB 03], las distribuciones se dan al azar en la siguiente proporción:

<table>
<thead>
<tr>
<th>Distribución</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.3.2</td>
<td>21.5512</td>
</tr>
<tr>
<td>4.3.3.3</td>
<td>10.5361</td>
</tr>
<tr>
<td>4.4.4.1</td>
<td>2.9932</td>
</tr>
<tr>
<td>5.3.3.2</td>
<td>15.5168</td>
</tr>
<tr>
<td>5.4.3.1</td>
<td>12.9307</td>
</tr>
<tr>
<td>5.4.2.2</td>
<td>10.5797</td>
</tr>
<tr>
<td>5.5.2.1</td>
<td>3.1739</td>
</tr>
<tr>
<td>5.4.4.0</td>
<td>1.3433</td>
</tr>
<tr>
<td>5.5.3.0</td>
<td>0.8952</td>
</tr>
<tr>
<td>6.3.2.2</td>
<td>5.6425</td>
</tr>
<tr>
<td>6.4.2.1</td>
<td>4.7021</td>
</tr>
<tr>
<td>6.3.3.1</td>
<td>3.4482</td>
</tr>
<tr>
<td>6.4.3.0</td>
<td>1.3262</td>
</tr>
<tr>
<td>6.5.1.1</td>
<td>0.7053</td>
</tr>
<tr>
<td>6.5.2.0</td>
<td>0.6511</td>
</tr>
<tr>
<td>6.6.1.0</td>
<td>0.0723</td>
</tr>
<tr>
<td>7.3.2.1</td>
<td>1.8808</td>
</tr>
</tbody>
</table>
Con respecto a los puntos de honor, la proporción es:

<table>
<thead>
<tr>
<th>PH</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3639</td>
</tr>
<tr>
<td>1</td>
<td>0.7884</td>
</tr>
<tr>
<td>2</td>
<td>1.3561</td>
</tr>
<tr>
<td>3</td>
<td>2.4624</td>
</tr>
<tr>
<td>4</td>
<td>3.8454</td>
</tr>
<tr>
<td>5</td>
<td>5.1862</td>
</tr>
<tr>
<td>6</td>
<td>6.5541</td>
</tr>
<tr>
<td>7</td>
<td>8.0281</td>
</tr>
<tr>
<td>8</td>
<td>8.8922</td>
</tr>
<tr>
<td>9</td>
<td>9.3562</td>
</tr>
<tr>
<td>10</td>
<td>9.4051</td>
</tr>
<tr>
<td>11</td>
<td>8.9447</td>
</tr>
<tr>
<td>12</td>
<td>8.0269</td>
</tr>
<tr>
<td>13</td>
<td>6.9143</td>
</tr>
<tr>
<td>14</td>
<td>5.6933</td>
</tr>
<tr>
<td>15</td>
<td>4.4237</td>
</tr>
<tr>
<td>16</td>
<td>3.3109</td>
</tr>
<tr>
<td>17</td>
<td>2.3617</td>
</tr>
<tr>
<td>18</td>
<td>1.6051</td>
</tr>
<tr>
<td>19</td>
<td>1.0362</td>
</tr>
<tr>
<td>20</td>
<td>0.6435</td>
</tr>
<tr>
<td>21</td>
<td>0.3779</td>
</tr>
<tr>
<td>22</td>
<td>0.2100</td>
</tr>
<tr>
<td>23</td>
<td>0.1119</td>
</tr>
<tr>
<td>24</td>
<td>0.0559</td>
</tr>
<tr>
<td>25</td>
<td>0.0264</td>
</tr>
<tr>
<td>26</td>
<td>0.0117</td>
</tr>
<tr>
<td>27</td>
<td>0.0049</td>
</tr>
<tr>
<td>28</td>
<td>0.0019</td>
</tr>
</tbody>
</table>
Digamos de paso que es posible generar ambas tablas tanto de manera analítica como de manera experimental. Nosotros nos basamos en esta información ya publicada por la ACBL.

Como vemos en [T.5], ningún porcentaje supera el 10%. En cambio en [T.4] hay casos en que sí.

En cierta forma estas tablas nos muestran cuán probable es generar una mano en un caso y en el otro. Y la tabla de PH es más restrictiva que la de distribución.

**Restricciones usando fuerza combinada**

Algunas de las condiciones usadas en los libros de Bridge para el remate hacen referencia a la *valoración conjunta* de las manos de los jugadores de un mismo equipo:

- para *gran slam* se debe tener combinado entre 37 y 38 PHD y como mínimo 7 cartas del mismo palo
- para *pequeño slam* entre 33 y 34 PHD y como mínimo 6 cartas del mismo palo
- para *game ST (3 ST)* entre 25 y 25 PH...

Estas restricciones parece que fueron condiciones nuevas distintas de las habladas hasta ahora. Sin embargo no lo son.

Por ejemplo $\text{PH (N)} + \text{PH (S)} = 20$ se puede lograr usando una asignación al azar de naturales $x$, y que cumpla $x + y = 20$ (siempre son pocas) y luego generar en la forma usual a partir de ahí.

Así con ejemplos parecidos:

$$\text{PH (N)} + \text{PH (S)} < \text{PH (E)}$$
$$\text{NC(N,picas)} + \text{NC(S,picas)} = 7, \text{etc}$$

Incluso la asignación de naturales podría hacerse usando algún lenguaje adecuado para resolver eso. Entre ellos los lenguajes de satisfacción de restricciones -o Constraint Satisfaction Problems (CSP) (ver Apéndice 1)-, que, como se sabe, resuelven bien ese tipo de problemas o bien muchas de sus instancias, hasta tanto se apliquen correctamente.

De modo que usando esos lenguajes se resuelve la asignación de naturales. Luego estos valores pasan a ser restricciones de entrada de un generador del estilo del SmartGen.

Ej: $\text{PH (N)} + \text{PH (S)} = 20$

=> con CSP resolvemos $9 + 11 = 20$,

=> con SmartGen generamos una mano con 9 PH para Norte y 11 PH para Sur.

Cambia un poco la cosa, aunque no mucho, si la condición habla de *cartas preasignadas combinadas* -por ej para N y S-. Aquí simplemente se mejoraría el procedimiento general, que deberá repartir al azar estas cartas entre N y S, y luego hacer el reparto de la manera corriente.

**Conclusiones**

En este capítulo y el anterior hemos descripto el núcleo del generador que proponemos:

- presentamos distintas heurísticas para dirigir la generación en el sentido de las restricciones.
• describimos la forma de considerar cartas preasignadas
• nos referimos a cómo optimizar el chequeo de las manos generadas

Todo lo propuesto está codificado en SmartGen, un generador de manos de Bridge con restricciones escrito en lenguaje Delphi (ver Apéndice 4)

Con respecto a la fuerza combinada, es claro que es muy necesario contemplar este tipo de restricciones. Sobre todo para la etapa del remate donde la fuerza conjunta va tomada forma a medida que los cantos se suceden hasta jugar un papel preponderante en el momento de definir el contrato. Asimismo cualquiera de los tipos de restricciones que hemos analizado anteriormente se pueden pensar en forma conjunta, a saber: cartas que se poseen, PH, PD, L.T, puntos por palo, longitud de palos, etc. Todo se piensa en forma combinada, para la pareja de jugadores.
En menor medida podría contarse también con restricciones que indiquen valores conjuntos por dos jugadores que no formen un equipo, aunque esto no siempre revestirá interés en el juego o en los problemas clásicos de Bridge. Por ej, en algunos casos será interesante una mano en donde entre norte y este tengan todas las picas, o bien todos los PH, etc.

Dejamos pues planteada una forma de incorporarlo al esquema mencionado hasta ahora.
Capítulo 6

Divide y triunfarás. Ordena y progresarás.

Heurística general de ordenamiento

Los siguientes son otros mecanismos que fueron surgiendo al implementar SmartGen y que quedan como futuras extensiones para ser estudiadas en profundidad e implementadas.

Orden de los jugadores

Como vimos anteriormente, el generador asigna las manos jugador por jugador. El orden en que lo hace es determinante para que las restricciones puedan ser satisfechas lo antes posible.
Lo que proponemos aquí es ordenar los jugadores según alguna medida de complejidad en las restricciones. La mano del jugador con condición más restrictiva se resuelve primero.
Pensemos en restricciones muy fuertes como distribución 13.0.0.0 con palo largo en pica. Si este jugador no se atiende primero, se corre el riesgo de que se use algún naípe de pica en un jugador anterior y el reparto fracase.

Lo difícil es en general establecer una medida o ranking de complejidad para las condiciones. Esto daría lugar a un estudio adicional.

Una posibilidad es -como hacen algunos Prolog mejorados- ordenar según quién tiene la condición "más larga" o con "más predicados". Esto no siempre es bueno, sólo da la idea de la condición más compleja.

Otra, podría ser testear empíricamente estas condiciones en forma aislada, con GAT, a ver cuál de ellas arroja más manos (así sueltas) y eso daría la idea de cuál condición es más difícil de satisfacer.


Orden atómico

En esta sección hacemos algunas consideraciones respecto del orden de evaluación de las condiciones presentes en la generación.

Es sabido que en procesos inherentemente no determinísticos, cercanos al GAT en mayor o menor medida, en donde la prueba juega un papel importante del proceso, será conveniente y útil contar con un orden adecuado para la comprobación de las condiciones que se deseen al validar las respuestas buscadas.

Existen intérpretes Prolog que tienen en cuenta estos posibles ordenamientos de predicados, que de algún modo mejoran el simple backtracking que por desinformado o no inteligente no tiene en cuenta aspectos a veces obvios en el control de la búsqueda de respuestas.

Una técnica que puede emplearse es la adquisición automática del conocimiento de este orden de condiciones. Este sistema proviene de mecanismos en principio conocidos dentro de las probabilidades y también dentro del aprendizaje automático. Los detalles de la técnica a describir a priori escapan al alcance de esta tesis, pero nos limitaremos tan sólo sugerir en qué consiste su escencia, la manera básica de usarla y una posible justificación.
Ordenamiento general

En esta sección se intenta proponer un criterio de comparación de predicados que luego será usado al servicio del ordenamiento de predicados en la etapa de la verificación secuencial de condiciones para la generación de manos. Este criterio será lo suficientemente general como para ser tenido en cuenta en otros problemas de búsqueda de soluciones y satisfacción de restricciones expresables con condiciones lógicas, de algún modo restringidas.

La idea del ordenamiento de predicados surge a partir del deseo de contar con heurísticas de distintos tipos para el caso de nuestro problema, así como para otros. De algún modo se corresponde con las heurísticas de selección u ordenamiento presentes en los enfoques de búsqueda en espacios de estados para la resolución de ciertos problemas. En nuestro caso se ve esto combinado con el hecho de que nuestro problema exige la generación de respuestas (diferentes), no así el encontrar una simple respuesta cualquiera. Con lo cual hay una componente de generación y otra de verificación. La componente de generación puede estar basada parcialmente en estos predicados y sus valores así obtenidos. La componente de verificación también estará basada en este orden de verificación.

Nos basamos en probabilidades condicionales, mediante las cuales podemos establecer el siguiente criterio de comparación de predicados e incluso de combinaciones booleanas de éstos. De alguna manera este es un enfoque probabilístico a partir de conceptos bayesianos [REN 78]

Preliminares

Recordemos [FEJ 78] que en cualquier espacio de probabilidad, para todo par de variables aleatorias X, Y y valores a y b, vale que

\[ \text{Prob}(X = a \mid Y = b) = \text{Prob}(X = a \wedge Y = b) / \text{Prob}(Y = b) \]

hecho que usaremos en lo que sigue del capítulo. Como caso particular podemos tomar predicados P y Q en lugar de las variables, y tomar a = b = verdadero como valor que pueden tomar. Luego esta relación toma la forma

\[ \text{Prob}(P \mid Q) = \text{Prob}(P \wedge Q) / \text{Prob}(Q) \]

En virtud de esto podemos dar la siguiente

Definición

Dados dos predicados P y Q sobre un mismo dominio, diremos que P es preferible a Q, notado P \( \leq \) Q, si

\[ \text{Prob}(P \mid Q) \leq \text{Prob}(Q \mid P) \]

Es decir, si en nuestro espacio de probabilidad la probabilidad de que P sea verdadero sabiendo que Q es verdadero, es menor o igual que la probabilidad de que Q sea verdadero sabiendo que P es verdadero.

Notemos que en realidad la relación \( \leq \) depende del dominio de definición de los predicados, pero para simplificar la notación sólo lo indicaremos del modo antedicho.

La idea intuitiva detrás de esta definición es: será preferible intentar satisfacer un predicado P que cuando Q es cierto tenga menor probabilidad de tener éxito que en el caso contrario, es decir, que cuando P sea cierto Q deba satisfacerse.

Definición

Dados dos predicados P y Q sobre un mismo dominio, diremos que P es igualmente preferible que Q, notado P \( \approx \) Q, si

\[ \text{Prob}(P \mid Q) = \text{Prob}(Q \mid P) \]

En lo que sigue P, Q y R denotan predicados sobre un mismo dominio. En tanto sea necesario, se asumirá que las probabilidades involucradas son distintas de 0.
Lema 1

\[ P \approx Q \iff \text{Prob}(P) = \text{Prob}(Q). \]

Dem.
\[ P \approx Q \iff \text{Prob}(P \mid Q) = \text{Prob}(Q \mid P) \iff \frac{\text{Prob}(P \wedge Q)}{\text{Prob}(Q \wedge P)} = \frac{\text{Prob}(Q \wedge P)}{\text{Prob}(P)} \iff \text{Prob}(Q) = \text{Prob}(P) \]

(sea o no sea 0 el valor de \text{Prob}(P \wedge Q))

Observación

La relación \( \approx \) (igualmente preferible que) es una relación de equivalencia, es decir cumple las propiedades de reflexividad, simetría y transitivity.

Dem.
Simple verificación.

A continuación mostramos una equivalencia que da otra formulación en cierto modo más sencilla para la relación \( \leq \).

Lema 2

\[ P \leq Q \iff \text{Prob}(P) \leq \text{Prob}(Q) \]

Dem.
\[ P \leq Q \iff \text{Prob}(P \mid Q) \leq \text{Prob}(Q \mid P) \iff \frac{\text{Prob}(P \wedge Q)}{\text{Prob}(Q \wedge P)} \leq \frac{\text{Prob}(Q \wedge P)}{\text{Prob}(P)} \iff \text{Prob}(P) \leq \text{Prob}(Q) \]

(observando que en el caso \text{Prob}(P \wedge Q) = 0 la desigualdad también vale)

Esto dice que en realidad se podría haber definido \( P \leq Q \iff \text{Prob}(P) \leq \text{Prob}(Q) \), lo cual también concuerda con la intuición ya que, al intentar asignar cartas que cumplan un conjunto de predicados \( P, Q \) por ejemplo), será casi siempre preferible comenzar con un predicado \( P \) con menor probabilidad de tener éxito (que lo demás, en este caso \( Q \)). Si no tuviese éxito una asignación de cartas que cumpla el primer predicado \( P \), es esperable que tendrá menos éxito (o igual) dicho predicado una vez que se cumpla otro predicado \( Q \).

En la práctica, lo antedicho nos permitirá dado un conjunto de predicados ordenarlos si conocemos sus probabilidades. De hecho no todo es tan simple, ya que usualmente hay predicados compuestos (son una combinación booleana de otros más simples) que pueden ser candidatos a ser ordenados.

Lema 3

La relación \( \leq \) es un cuasi orden total, es decir reflexiva y transitiva y todo par de predicados es comparable.

Dem.

i) reflexividad: \( P \leq P \) ya que \( \text{Prob}(P \mid P) = 1 = \text{Prob}(P \mid P) \)

ii) transitividad: si \( P \leq Q \) y \( Q \leq R \), entonces aplicando el lema 2 dos veces, \( \text{Prob}(P) \leq \text{Prob}(Q) \) y \( \text{Prob}(Q) \leq \text{Prob}(R) \), lo que dice que \( \text{Prob}(P) \leq \text{Prob}(R) \), y aplicando el lema 2 nuevamente, \( P \leq R \).

Por último, que es total es evidente.

Observación: no necesariamente vale la propiedad antissimétrica: \( P \leq Q \) y \( Q \leq P \) no implican que \( P = Q \) (de hecho pueden ser predicados bien distintos, tan sólo coinciden en la probabilidad). Pero sí será cierto que \( P \approx Q \) (como se afirma en el lema 1).

Lo interesante de las relaciones \( \leq \) y \( \approx \) es que son congruenencias respecto de los conectivos booleanos \( \wedge, \vee \), es decir no varian si se combinan con esos conectivos e iguales predicados. Esto permitirá atacar la necesidad de comparar predicados compuestos por otros más elementales. Veamos primero

Lema 4
Si $P \leq Q$, entonces $\text{Prob}(P | R) \leq \text{Prob}(Q | R)$

Dem.

$P \leq Q$

$\Rightarrow$ (lema 2) $\text{Prob}(P) \leq \text{Prob}(Q)$

$\Rightarrow$ $\text{Prob}(P | R) \leq \text{Prob}(Q | R)$

(para esta última implicación ver cualquier texto elemental de Probabilidades [JAM 80], [REN 78]).

Ahora si veamos en detalle lo afirmado arriba.

**Proposición 1**

Si $P \leq Q$, entonces valen:

a) $P \land R \leq Q \land R$

b) $\neg Q \leq \neg P$

c) $P \lor R \leq Q \lor R$

Dem.

a) $P \leq Q$

$\Rightarrow$ (lema 2) $\text{Prob}(P) \leq \text{Prob}(Q)$

$\Rightarrow$ (lema 4) $\text{Prob}(P | R) \leq \text{Prob}(Q | R)$

$\Rightarrow$ $\text{Prob}(P | R) \text{Prob}(R) \leq \text{Prob}(Q | R) \text{Prob}(R)$

$\Rightarrow$ $\text{Prob}(P \land R) \leq \text{Prob}(Q \land R)$

$\Rightarrow$ $P \land R \leq Q \land R$

b) $P \leq Q$

$\Rightarrow$ (lema 2) $\text{Prob}(P) \leq \text{Prob}(Q)$

$\Rightarrow$ $1 - \text{Prob}(P) \geq 1 - \text{Prob}(Q)$

$\Rightarrow$ $\text{Prob}(\neg P) \geq \text{Prob}(\neg Q)$

$\Rightarrow$ $\neg Q \leq \neg P$

c) $P \leq Q$

$\Rightarrow$ (por la parte b) $\neg Q \leq \neg P$

$\Rightarrow$ (por la parte a) $\neg Q \land \neg R \leq \neg P \land \neg R$

$\Rightarrow$ $\text{Prob}(\neg Q \land \neg R) \leq \text{Prob}(\neg P \land \neg R)$

$\Rightarrow$ $1 - \text{Prob}(\neg Q \land \neg R) \geq 1 - \text{Prob}(\neg P \land \neg R)$

$\Rightarrow$ $\text{Prob}(\neg Q \land \neg R) \geq \text{Prob}(\neg P \land \neg R)$

$\Rightarrow$ (por las leyes de De Morgan) $\text{Prob}(Q \lor R) \geq \text{Prob}(P \lor R)$

$\Rightarrow$ $P \lor R \leq Q \lor R$

Nótese que la afirmación $P \approx \neg P$ podría o no satisfacerse, dependiendo exclusivamente de $\text{Prob}(P)$.

**Observación**

$P \approx \neg P \Leftrightarrow \text{Prob}(P) = \text{Prob}(\neg P) = \frac{1}{2}$.

Dem.

$P \approx \neg P \Leftrightarrow P \leq \neg P \land \neg P \leq P \Leftrightarrow$ (lema 4) $\text{Prob}(P) = \text{Prob}(\neg P)$, luego

$\text{Prob}(\neg P) = 1 - \text{Prob}(P)$, luego $\text{Prob}(P) = \frac{1}{2} = \text{Prob}(\neg P)$.

Denotemos con 1 al predicado siempre verdadero, y con 0 al predicado siempre falso.
Lema 5

Para todo predicado $P$, $0 \leq P \leq 1$.

Dem.

Prob($P$) $\geq$ Prob(0)

Prob ($P$) $\leq$ 1 = Prob(1)

Veamos que asimismo vale la

Proposición 2

a) $P \land Q \leq P$, $P \land R \leq R$

b) $P \leq P \lor Q$, $Q \leq P \lor Q$

Dem.

a) Usando la proposición 1 (a) y el lema 5, $P \land Q \leq P \land 1 \leq P$ (ya que $P \land 1 = P$)

Luego $P \land Q \leq P$

b) Usando la proposición 1 (c) y el lema 5, $P \leq P \lor 0$ (ya que $P \lor 0 = P$)

$P \lor 0 \leq P \lor Q$. Luego $P \leq P \lor Q \leq P$

Otra manera más simple de probar la proposición 2 es notar que $P \land Q$ es más restrictivo que $P$ y que $Q$, al igual que $P \lor Q$ es menos restrictivo que $P$ y que $Q$.

Generalicemos de alguna forma lo afirmado hasta ahora.

Definición

Se llama $f(P_1, ..., P_n)$ una combinación booleana de los predicados $P_1, ..., P_n$ que utiliza sólo los conectivos $\land, \lor$.

En ese caso a $f$ la llamaremos una combinación positiva de dichos predicados.

Corolario

Si $f(P_1, ..., P_n)$ es una combinación positiva de predicados, entonces:

a) si para algún $1 \leq i \leq n$, $P_i \leq Q_i$, entonces $f(P_1, ..., P_n) \leq f(P_1, ..., P_i, Q_i, P_{i+1}, ..., P_n)$

b) si para todo $1 \leq i \leq n$, $P_i \leq Q_i$, entonces $f(P_1, ..., P_n) \leq f(Q_1, ..., Q_n)$

Dem.

a) Usando la proposición 1, y aplicando inducción en la formación de $f$.

b) Usando (a) e inducción en $n$

Para que este resultado valga hemos pedido que la combinación sea positiva ya que de lo contrario el sentido de la desigualdad cambia o puede cambiar según el caso de qué combinación $f$ se trate. Si bien no profundizaremos en esto, digamos que aún en el caso en que no sea positiva se puede dar un resultado análogo al corolario parte (b) en donde el sentido de la desigualdad $f(P_1, ..., P_n) \leq f(Q_1, ..., Q_n)$ cambia o no en función de $f$ y de las ocurrencias positivas y negativas de los argumentos. No expondremos esto en detalle aquí.

Aplicación a la producción de manos

Es de destacar que en la práctica resultará difícil ordenar una familia de (finitos) predicados dado que deberíamos en general conocer las probabilidades de que sean verdaderos.
Pero esto sin embargo resulta sencillo para ciertos predicados de las manos de Bridge, muchas veces estudiados y con los que se cuenta con tablas al respecto bien conocidas (ver capítulo 5).

Sean por ejemplo predicados siguientes:

\[ PH = k \]
\[ PH > k \]
\[ PH < k \]
\[ PD = k \]
\[ PD < k \]
\[ PD > k \]
\[ Distrib = x, y, z, t \]

Por simplicidad en estos ejemplos, los predicados antedichos se deben pensar en cada caso con \( k \) constante. Para todos ellos existen tablas de probabilidades (muchas obtenibles de las tablas del capítulo 5). También pueden aproximarse o estimarse estas probabilidades en forma empírica, para lo cual nos podemos valer del generador (en modo GAT) aplicado a condiciones irrestrictas y generando un número muy grande de manos podemos establecer una aproximación a estas probabilidades mediante el número de manos favorables obtenidas sobre el total de manos.

Hay no obstante otros predicados de interés, a saber:

\[ PH + PD = k \]
\[ PH + PD < k \]
\[ PH + PD > k \]

Para el caso de estos predicados “más complejos” podemos descomponerlo en combinaciones booleanas (usando disyunciones o conjunciones) de los predicados correspondientes para \( PH \) y para \( PD \) (dado que los números de puntos en cualquier caso tienen rangos finitos), y aplicando el último corolario o bien las proposiciones 1 y 2 se podrá contar con esta información.

Lo mismo se aplicará al caso del predicado “ser una distribución balanceada”, vale decir:

\[ P(m) = m \] es una mano balanceada

En este caso \( P(m) \) resulta la disyunción siguiente (que utiliza predicados anteriores):
\[ Distrib = 4.3.3.3 \lor Distrib = 4.4.3.2 \lor Distrib = 5.3.3.2 \]

Y así con otros tipos de predicados combinados a partir de estos.

En resumidas cuentas, se puede obtener información de cómo ordenar siguiendo un criterio razonable un predicado que es una combinación booleana de otros predicados componentes, a veces a partir de éstos, para los cuales se conozcan las probabilidades de que sean verdaderos.

**Extensiones**

Lo anterior presupone el conocer estas probabilidades o estimaciones razonablemente buenas. Surge la pregunta de qué puede hacerse cuando esto no se cumple, es decir, para el caso en que no se cuente siquiera con estas aproximaciones. Se podrá apelar a ordenar los predicados compuestos en ese caso. Para el caso de los predicados simples, es decir que no puedan ser expresados como combinación booleana de otros conocidos, esto no es posible ya que al no tener indicios de su comportamiento probabilístico no se puede utilizar este sistema. Para el caso de ciertos predicados compuestos, es en cierto modo plausible utilizar esto si se les da por ejemplo un valor de probabilidad (suerte de estimación) a los predicados componentes, o bien utilizar el valor de \( \frac{1}{2} \) a algunos de ellos a partir de la idea de entropía máxima, esto es, se les asigna artificialmente este valor a estos componentes y se utilizan así entre los otros componentes del mismo predicado con los que se cuente información. La idea de asignar este valor sin compromiso a estos predicados los ubica aproximadamente en medio de otros predicados entre los que constituyen el conjunto que se esté utilizando.
De ser así, y cuando el número de estos valores artificiales sea bajo en relación al total, creemos que esto podrá mejorar el comportamiento general del generador aún sin conocer eficazmente todos estos valores.
La intuición dice de alguna manera que con estos valores se podrá tender a la larga a obtener mejores resultados o bien suficientemente aceptables.
Esta posibilidad puede llegar a ser aplicable a muchos otros problemas “suficientemente cercanos” al estudiado en la presente tesis.

Conclusiones

En este capítulo hemos formulado distintas mejoras al esquema planteado en el capítulo anterior, dando un marco más formal al criterio de ordenamiento de condiciones. Hemos seguido un esquema probabilístico bastante naïf, permitiendo esto su aplicación en más de un escenario posible. No obstante eso, reconocemos que en ciertas ocasiones aparecen dificultades adicionales tales como la necesidad de conocer ciertas distribuciones de probabilidad que normalmente podrían no poseerse. Esto naturalmente puede mejorar con ciertos ajustes que dependerán de la naturaleza de los predicados involucrados.
Tanto el ordenamiento de jugadores como el ordenamiento de predicados mejorarían los resultados actuales del generador en lo que respecta a eficiencia.
Hemos hecho algunas pruebas para comprobar experimentalmente las bondades y ventajas de ordenar correctamente los jugadores, por ejemplo forzando ‘manualmente’ el orden (de manera que primero se resuelva el jugador con condición más restrictiva y luego los demás siguiendo este criterio). Los resultados siempre fueron positivos.
Capítulo 7

La información y el tiempo valen oro.

Usando el Remate como fuente de restricciones

Hasta el momento hemos detallado modos de funcionamiento de un generador de manos basado en restricciones, que si bien son posibles de ser usadas en la realidad, aparecen en un esquema relativamente independiente del juego de Bridge en sí. Aunque los tipos de condiciones que estamos manejando son conocidas –de una u otra manera- por la mayoría de los jugadores de Bridge, existe otra manera de obtener tipos de restricciones o formulaciones de las mismas que no hemos atacado.

Una de las posibilidades más interesantes a la hora de generar manos de Bridge tiene que ver con poder utilizar el Remate como fuente de restricciones; en otros términos, el remate o parte de éste como restricción general. Esta característica le da al generador la oportunidad de ser usado no solo para training, sino en la etapa del remate de algún software que juegue al Bridge.

Consiste más precisamente en la necesidad de generar manos (para uno o más contendientes) que se adapten a un remate conocido total o parcialmente. De alguna manera representa un problema inverso a lo que hacen habitualmente los jugadores de bridge: dada una mano, cómo rematarla, es decir cuáles deben ser las sucesivas declaraciones.

En Pragmatic Reasoning in Bridge [GAM 93], se propone una forma de estimar manos basada en el remate. El esquema consiste en utilizar un simulador estocástico que reparte cartas al azar. Si las cartas satisfacen las restricciones resultantes del remate, la mano es aceptada, sino se intercambian pares de cartas en forma random y se vuelven a chequear las restricciones. Así hasta que la mano se acepte. Como puede observarse, es un esquema GAT.

En Reasoning by Agents in Computer Bridge Bidding [UEH 99] Uehara y Ando definen la generación de una imagen de la mano de cada uno de los otros jugadores, utilizando ECL/PS [ECL 02], un lenguaje de programación lógica con restricciones.

Nuestra propuesta del remate como restricción se basa en este artículo.

Para simplificar el concepto, solo nos vamos a referir a partnership bidding, es decir al remate entre jugadores del mismo equipo (los oponentes siempre "pasan"). El lenguaje que usaremos es Prolog con extensiones para programación lógica con restricciones –o Constraint Logic Programming (CLP)- [SIC 02], [CAR 97].

Hemos escrito una breve explicación sobre CLP en el Apéndice 1.

Se debe prestar atención al uso que le damos a la palabra restricción en el texto de éste capítulo. A veces nos referiremos a restricciones como los constrains de un CLP y a veces nos referiremos a restricciones como las condiciones de una mano.

La idea es escribir un programa CSP que dada una secuencia de cantos de remate, retorne la configuración de dos manos coherentes con esos cantos. Configuración significa: cantidad de cartas de cada palo y puntos de honor, es decir restricciones. Estos 5 parámetros luego se usan como entrada para SmartGen que en definitiva será el que genera la mano real.
Estructura

Básicamente el programa tendrá las siguientes partes:

1. Reglas del remate: predicados que modelan la apertura, respuesta, reapertura, etc.
2. Predicados auxiliares: balanceado, contar pd
3. Predicado get_cfg_hands(Bidding, Hand1, Hand2): es el punto de entrada.

Veamos un ejemplo:

- Encontrar manos para la secuencia de cantos 1 ST, 2 diamante:

  get_cfg_hands([ [1, st], [2, d] ], H1, H2).

la respuesta puede ser:

  H1 = [[3, 3, 3, 4], 15]
  H2 = [[0, 0, 7, 6], 7]

En H1, [3,3,3,4] se refiere a la cantidad de cartas por palo: 3 picas, 3 corazones, 3 diamantes y 4 trébol
Y 15 se refiere a los puntos de honor.
Lo mismo se aplica a H2.

Luego, utilizando SmartGen obtenemos las cartas reales.

Reglas del remate

Definimos una base de conocimiento con todas las reglas necesarias para la subasta: aperturas, respuestas, redeclaraciones, etc. Aquí se modela el sistema de remate que queramos. Nosotros mostraremos algunos ejemplos basados en el Mayor Quinto.

Las reglas de apertura tienen la siguiente forma: opening_bid( Bid, HandConfig ) en la que Bid se refiere al canto de apertura (parámetro de entrada) y HandConfig a los parámetros de la mano necesarios para realizar ese canto (de salida): cantidad de cartas por cada palo, puntos de honor y puntos de distribución.
% regla para cantar “1 pique” //////////////
opening_bid( [1,p], % canto
    [[NP,NC,ND,NT],PH,PD ] ) :- % parámetros
    notbalanced(NP,NC,ND,NT), % mano no balanceada
    sum( [PH, PD], #=, PHD ), % PHD puntos de honor+destribución
    PHD #>= 13, PHD #=> 18, % en el rango [13..20]
    NP #>= NC, % nro de cartas de pique mayor a los demás
    NP #>= ND,
    NP #>= NT.

% regla para cantar “1 ST” ///////////////
opening_bid( [1,st], % canto
    [[NP,NC,ND,NT],PH,_,PD ] ) :- % PD se ignora en ST
    balanced(NP,NC,ND,NT), % mano balanceada
    PH #>= 15, PH #<= 18. % entre 15 y 18 puntos de honor

Los operadores lógicos con # tienen el mismo significado que los operadores normales de Prolog cuando las variables están instanciadas. En cambio, cuando no están instanciadas el solver de CSP las mantiene como un goal retrasado (delayed goal) [SIC 02].

El predicado sum( lista, op, res) compara la suma las variables de dominio de la lista contra res usando el operador op.

De esta forma definimos toda la tabla de apertura.

En la práctica estas reglas podrán usarse de dos formas: si se instancia las variables de canto, el resultado será la configuración de cartas. Y al revés, si se instancia la configuración de cartas, obtendremos el canto. A nosotros nos interesa la primera forma.

Seguimos definiendo predicados. Ahora los que modelan las respuestas a la apertura:

% respuesta a 1 ST: paso
response_bid( [ [1,st], % 1er canto
    [paso] ], % 2do canto
    [[NP,NC,ND,_,NT],PH,_ ] ) :- % respuesta
    PH #=< 7, % PH mayor a 7
    NP #=> 5, % suits de más de 5 cartas (todo menos trebol)
    NC #=> 5,
    ND #=> 5,

% respuesta a 1 T: 3 T
response_bid( [ [1,t], [3,t] ], [[NP,_,NC,_,ND,NT],PH,PD] ) :-
    sum( [PH, PD], #=, PHD ),
    PHD #=> 11,
    NT #=> 5.

De esta forma definimos todas las respuestas a aperturas.
Luego deberán definirse los predicados de reapertura que tendrán la forma:

```prolog
re_bid( [Open, Response, Reopen], [[NP,NC,ND,NT],PH,PD] )
```

Y así hasta incluir todas las reglas necesarias.

**Predicado get_cfg_hand**

Primero vamos a definir un predicado para conseguir la configuración de una mano usando solo un canto de apertura. Luego lo extenderemos para dos jugadores y tres cantos.

La estructura de este predicado sigue los lineamientos generales de una rutina CSP escrita con Sicstus Prolog. Recomendamos leer [SIC 03] cap. 34.

```prolog
get_1_cfg_hand( OpenBid, [[NP,NC,ND,NT],PH,PD] ) :-
  %
  % aquí se definen las variables y sus dominios
  %
  domain( [NP,NC,ND,NT], 0, 13 ), % largos posibles de suit
  domain( [PH], 0, 37 ), % puntos de honor posible por mano

  %
  % ahora definimos las restricciones elementales
  %
  sum([NP,NC,ND,NT], #=, 13 ), % la cantidad total de cartas es 13
  calc_pd( PD, [NP,NC,ND,NT] ), % calculamos los puntos de distr.

  %
  % ahora la restricción que usa la base de conocimiento
  %
  opening_bid( OpenBid, [[NP,NC,ND,NT],PH,PD] ),

  %
  % aquí se asignan las variables de dominio y se hace backtracking
  % cuando las restricciones no se satisfacen
  %
  labeling([], [NP,NC,ND,NT,PH]).
```

Los predicados de **dominio** definen el valor que puede tomar cada variable. Las **restricciones** limitarán los valores del dominio. **Opening bid** actúa como una restricción más. El **labeling** asignará una a una las variables de dominio chequeando las restricciones. En cuanto un juego de asignaciones satisfaga todas las restricciones, el motor para y retorna la primer solución.

A continuación definimos el predicado para los dos jugadores. Las variables de dominio se duplicarán ya que hay un juego de variables para cada mano. Para simplificar el ejemplo, consideraremos solo tres cantos (apertura, respuesta y reapertura):

```prolog
get_cfg_hand( [OpenBid, ResponseBid, ReBid], [[NP1,NC1,ND1,NT1],PH1,PD1],
              [[NP2,NC2,ND2,NT2],PH2,PD2] ) :-
```

Pag. 53
% variables de dominio para cada mano
% domain( [NP1, NC1, ND1, NT1], 0, 13 ),
domain( [NP2, NC2, ND2, NT2], 0, 13 ),
domain( [PH1], 0, 37 ),
domain( [PH2], 0, 37 ),

% exactamente 13 cartas en cada mano
sum( [NP1,NC1,ND1,NT1], #=, 13 ),
sum( [NP2,NC2,ND2,NT2], #=, 13 ),

% no más de 13 cartas de cada palo
sum( [NP1,NP2], #=<, 13 ),
sum( [NC1,NC2], #=<, 13 ),
sum( [ND1,ND2], #=<, 13 ),
sum( [NT1,NT2], #=<, 13 ),

% no más de 40 puntos de honor en ambas manos
% (es una cota muy alta, pero posible)
sum( [PH1,PH2], #=<, 40 ),

% cálculo de PD para cada mano
calc_pd( PD1, [NP1,NC1,ND1,NT1] ),
calc_pd( PD2, [NP2,NC2,ND2,NT2] ),

% aquí usamos la base de conocimiento para cada canto
opening_bid( OpenBid, [[NP1,NC1,ND1,NT1],PH1,PD1] ),
response_bid( [OpenBid, ResponseBid], [[NP2,NC2,ND2,NT2],PH2,PD2] ),
re_bid( [OpenBid, ResponseBid, ReBid], [[NP1,NC1,ND1,NT1],PH1,PD1] ),
labeling([], [ NP1,NC1,ND1,NT1,PH1,NP2,NC2,ND2,NT2,PH2]).

Veamos la llamada a la base de conocimiento. Tanto para el predicado opening_bids como re_bids, usamos las mismas variables de dominio: las cartas del jugador 1. Aquí es donde el motor, aplicando abducción, combina las restricciones de ambos predicados y no para hasta conseguir una asignación que satisfaga ambas restricciones.

Para el predicado response_bid se usan las variables del jugador 2. La cosa es más sencilla para este jugador porque estamos considerando solo tres cantos (el jugador 2 cantó una vez sola). Al incluir más cantos sucedería lo mismo que con las variables del jugador 1: se incluirían en más de un predicado de la base de conocimiento.

Si bien no lo incluimos en el código descripto, se hace necesario definir alguna restricción que establezca la relación entre distribuciones y puntos de honor. Se sabe que, como mencionamos en otros capítulos, la distribución define los posibles valores en PH. Por ejemplo una distribución 13.0.0.0 tiene a lo sumo 10 PH, en cambio 12.1.0.0 llega a 14 PH. Esta información debería modelarse y utilizarse como restricción para no caer en combinaciones inexistentes.

Otro detalle a tener en cuenta es que el predicado labeling asigna por defecto las variables con valores del dominio en forma secuencial y ascendente. Eso significa que cada vez que se ejecuta el predicado con los mismos cantos, siempre retorna la misma configuración de mano (las mismas restricciones para SmartGen).
El primer parámetro del labeling [SIC 02] p. 445, permite controlar esta característica. Si por ejemplo ponemos la opción down la asignación se hará en forma descendente.
Hay varias opciones predefinidas, pero también podemos definir *predicados de usuario*. Es decir, podemos crear predicados con la heurística que queramos acerca de la selección de los valores del dominio. Incluso podemos definir el *orden* en que se asignan las variables. Con estas posibilidades que ofrece el lenguaje se puede definir un esquema de asignación random de valores, de forma tal de que los resultados de cada corrida sean distintos.

**Conclusiones**

Es muy prometedor el hecho de poder utilizar el remate como fuente de restricciones. De alguna manera provee un vínculo directo entre el juego y las restricciones, y es aplicable directamente al juego del bridge, como detallaremos posteriormente.

El esquema que proponemos fue desarrollado en forma experimental dando resultados alentadores.

El lenguaje a utilizar influye notablemente en la idea propuesta. Se puede decir que los lenguajes de CLP tienen la característica de que para resolver un problema no hay que escribir necesariamente un algoritmo, sino *describir* el problema adecuadamente, en términos del lenguaje. Esto para el remate resulta ser muy conveniente. Las técnicas del remate suelen ser en gran medida una colección de reglas, de ciertos tipos (con antecedentes dados por condiciones y consecuentes dados por decisiones), y estos lenguajes resultan muy aptos para escribirlas. Según [UEH 99] son unas 400 las reglas necesarias para describir el remate en el sistema Goren [GOR 85].
Capítulo 8

Cosecharás tu siembra.

Aplicaciones

En este capítulo detallamos algunas de las que creemos pueden ser las aplicaciones inmediatas de este trabajo así como de otros relacionados que pueden desarrollarse en el futuro.

Programas que juegan al Bridge

Los programas de Bridge representan una de las más claras aplicaciones. Estos programas pueden utilizar el generador tanto en la etapa de remate como en la de carteo. En el remate, generando manos usando los cantos efectuados hasta el momento y según las cartas encontradas, calcular el siguiente canto o bien optimizar la decisión que corresponda.

En el carteo, se puede usar mayor cantidad de restricciones: los cantos del remate, las cartas ya jugadas y las cartas del muerto (cartas preasignadas). Generar una mano teniendo en cuenta toda esta información implica saber las cartas que faltan jugar, o bien una muy buena aproximación a estas cartas. Tenemos la creencia de que de este modo se puede mejorar notablemente la performance de ciertos programas.

Jugadores

Desde el punto de vista de los jugadores humanos, hay aplicaciones concretas. Un jugador puede utilizar el generador para practicar distintos tipos de mano, tanto en remate como en carteo.

Enseñanza del Bridge

Esta es la aplicación más inmediata del generador. Un profesor de Bridge puede generar manos al azar a sus alumnos con las características del juego que está explicando. Irá variando o agregando restricciones en la medida en que sus alumnos avancen o progresen en el aprendizaje. Irá asimismo generando distintas manos con los mismos alumnos (para evitar así repetirse), y de ese modo acrecentar la práctica sobre los mismos tipos de manos.

También es una buena herramienta para un programa que enseña a jugar al Bridge. Con el generador se podrían conseguir programas interactivos más ricos y efectivos.

Torneos de Bridge

Como se dijo, una aplicación importante es la preparación de manos para el bridge duplicado en los torneos. Esta preparación se puede conseguir pues generando manos que se adapten a condiciones se desean se cumplan en el torneo. Esto se puede hacer en mayor o menor medida para todos los jugadores.

Se desea por ejemplo un torneo en donde haya un cierto porcentaje de manos que permitan el juego a nivel de game, otro con el juego a nivel de slam, otro con el juego del tipo sin triunfo, etc. Esto otorga la ventaja de que se
experimentarán distintas facetas del juego en el torneo, tendiendo a hacer por tanto más legítimos a los ganadores del mismo.

Para los casos comentados, resulta relativamente sencillo satisfacer estas demandas, a condición del armado de condiciones adecuadas para la generación de manos y el uso del generador varias veces pidiendo las condiciones deseadas y en los porcentajes indicados.

**Otros tipos de torneo**

Como aplicación adicional se pueden generar nuevos tipos de torneos de bridge, con características que detallamos a continuación.

Se establece como antes un cierto porcentaje de manos de diversos tipos (como los mencionados). Luego, se generan manos que satisfagan estas condiciones. Pero no se reparten las mismas manos a los jugadores de distintas mesas, sino sólo manos generadas con iguales condiciones.

Esto genera mayor variedad e incluso permitiría jugar en ambos roles (ataque y defensa) a los mismos equipos, sin riesgo de que el contenido de la mano sea conocido por ellos. Lo cual no es posible si se usan manos idénticas en distinta posición.

**Mejor comprensión de algunos problemas de Bridge**

Los estudiosos del Bridge encontrarán en el generador un aliado muy interesante a la hora de conseguir manos con condiciones difíciles o sobre situaciones de juego complejas.

Resultará interesante el poder comprobar ciertas correlaciones que se dan en algunas manos determinadas.

Por ejemplo, en el armado de algunos de los ejemplos que se usaron para probar SmartGen, se ha ido agregando restricciones en forma sucesiva. Al incluir ciertas restricciones, las manos generadas comenzaban a satisfacer ciertas condiciones no indicadas previamente.

A saber, y para ejemplificar con algo muy simple, al pedir manos sin restricciones se puede comprobar de algún modo cuáles son las distribuciones más comunes (suponiendo que el generador satsface parcialmente el ser uniformemente completo). Así como se puede conseguir esto, se podrá obtener información más compleja o difícil. Por ej. cómo son ciertas manos que satisfacen algunas condiciones dadas. Cuál es el número de honores esperado para un jugador. Cuál es el número de semifallos o dubletones esperado, etc. Otro caso: si se pide que N tenga 13 piques y E tenga 13 corazones, es obvio que el resto de las cartas estarán distribuidas entre S y O. Interesa por ej ver cuáles distribuciones son las más probables para los palos restantes. Así como estos ejemplos, hay innumerables facetas que podrían descubrirse -al menos en forma estocástica- en lo que respecta a este tipo de correlación, y con relativo poco esfuerzo. Esto debería ir acompañado de un análisis o justificación posterior, e incluso el uso de herramientas de aprendizaje inductivo podría ser una continuación natural para consolidar este conocimiento así adquirido.

**Otros juegos**

Es bueno aclarar que un generador con restricciones tiene sentido para casi cualquier juego de naipes. Nada impide utilizar un esquema similar en otros juegos. Es cuestión de estudiar las restricciones propias del juego y encontrar las heurísticas adecuadas, o bien aplicar las presentadas en esta Tesis cuando esto sea conveniente.

A saber, en los siguientes juegos conocidos:
- Tute, Belote, Piquete, Preference, Klaverjass (Klabiash) y otros varios juegos de bazas. Así como pasa con el Bridge, hay en estos juegos distintos tipos de mano, a veces iguales a las de Bridge, basadas en puntajes, palos largos y otras características relacionadas. Pero a diferencia de éste, no hay tanta literatura ni sistemas de juego tan conocidos.
- Poker. Podría tener interés un algoritmo de generación de manos para varios jugadores, a saber, que contengan ciertas combinaciones deseadas. Par, par doble, pierna, escalera, color, full, poker, escalera real. Pero a diferencia del
Bridge, aquí casi todo resultaría más sencillo, dado que las combinaciones son pocas y las cartas de una mano son menos, en este caso 5. Creemos que si el número de jugadores aumenta, entonces sí la generación revestirá una dificultad mayor.

Otros problemas

Es obvio que hay posibilidad de plantear el mismo esquema de solución, siguiendo alguna adaptación adecuada, a otros problemas en donde el esquema GAT sea aplicable, con el objeto de mejorar la performance de aquel esquema. Esto incluye por ejemplo la generación de conjuntos u otras estructuras que satisfagan restricciones determinadas para sus elementos o relaciones entre éstos.
Capítulo 9

Conclusiones

El juego del Bridge, dada su complejidad e interés en todo el mundo, reviste un campo abierto y apto para la aplicación de técnicas y procesos normalmente enmarcados en la inteligencia artificial, particularmente algoritmos de juego, búsqueda inteligente, resolución de problemas, muchas de estas cuestiones posibles de basarse en modelos de satisfacción de restricciones.

En este trabajo se ha hecho un análisis empírico sobre cómo utilizar las restricciones en la etapa de generación en un generador de manos de Bridge con restricciones varias, muchas específicas de este juego y otras más generales. Creemos que es la primera vez que se plantea un esquema de generación de este tipo.

La ventaja más obvia tiene que ver con la eficiencia para encontrar la respuesta en contraposición al método tradicional GAT, empleado por muchos generadores existentes que están actualmente siendo usados. Hemos mostrado una breve referencia del estado actual del arte con respecto a los generadores existentes. Vimos que algunos de los principales programas que juegan al Bridge utilizan generadores con esquemas GAT. Vimos también que algunos generadores admiten restricciones para uno o dos jugadores solamente o que algunos requieren de un programador para ejecutar una consulta complicada.

Planteamos los requisitos que un buen generador debe cumplir. Nos basamos en los generadores existentes, en el mismo juego de Bridge y en las aplicaciones que pudiera tener dicho generador.

Hemos visto que la generación de manos tiene diversos propósitos. Una aplicación importante la constituyen los programas que juegan Bridge. Intuimos fácilmente que un programa, a partir de sus propias cartas y usando conocimiento adquirido a lo largo de la mano (durante el remate fundamentalmente) puede de algún modo mejorar su juego mediante la generación 'a medida' de las manos de los otros jugadores. Esto bien puede lograrse mediante la generación automática de manos para los otros jugadores usando las restricciones obtenidas a lo largo del juego, incluyendo las cartas ya jugadas por cada uno de ellos. Este mecanismo se puede usar en combinación con los conocidos de búsqueda en ciertas etapas del carteo.

Otra de las aplicaciones precisas será el entrenamiento de jugadores reales, mediante distintos tipos de manos. Es sabido que los profesores y entrenadores de Bridge preparan manualmente manos específicas para 'probar' a sus alumnos. A veces resulta tedioso prepararlas, puede haber posibilidad de error e incluso los ejemplos a jugar suelen ser bastante restringidos.

Estudiamos y dejamos documentadas mejoras que podrán profundizarse e implementarse en el futuro, como el uso del remate como fuente de restricciones, que habilita al generador para ser usado por un programa que juega al Bridge en la etapa de remate.

También hablamos de heurísticas de ordenamiento de las restricciones, para aumentar la eficiencia sobre la convergencia de los resultados y deja planteado un esquema un tanto general como para que pueda ser usado en otros contextos o variantes.

Por último, hemos desarrollado una primera versión de SmartGen, un generador de manos que implementa algunas de las heurísticas descritas con el fin de probar y mostrar comparativamente que muchos de los resultados superan notablemente a los de los generadores tradicionales. Este prototipio, sin duda, podrá ser ampliado y mejorado en el futuro próximo.
Referencias

[ACB 03] American Contract Bridge League, URL: http://www.acbl.org


[BAR 99] Barták, R., Constraint Programming: In Pursuit of the Holy Grail, Charles University, Faculty of Mathematics and Physics, Department of Theoretical Computer Science, 1999


[BB 95] Bridge Baron, Great Games Productions Inc., 1995


[DEA 03] URL: http://thomason.best.vwh.net/bridge/deal/dealanswer.html


[DEL 98] Delphi version 4.0, Borland Corporation, URL: http://www.borland.com

[DIA 03] Diaz D., URL: http://pauillac.inria.fr/~diaz/gnu-prolog

[EAT 03] Eaton P., URL: http://www.cs.unh.edu/ccc/archive

[ECL 02] IC-Parc, URL: http://www-icparc.doc.ic.ac.uk/eclipse


[ILO 03] ILOG, URL: http://www.ilog.com/products/solver


[OKB 03] URL: http://www.okbridge.com


[PLA 03] URL: http://www.playbridge.com/pbeon_shuffle_project.asp


[SIC 02] Intelligent Systems Laboratory, SICStus Prolog User’s Manual (Release 3.10.0), Swedish Institute of Computer Science, 2002


[TCL 03] URL: http://www.scriptics.com


[UEH 94] Uehara T., Application of abduction to computer Bridge, Transactions of the institute of Electronics, Information and Communication Engineers (O-II), 177-01, No.II (1994-11) 2255-2264, 1994


Principiantes del Bridge

http://www.prairienet.org/bridge/learn.htm
http://www.acbl.org/Info/clubs.htm
http://www.acbl.org/educate/teachers.htm
http://www.prairienet.org/bridge/link-beg.htm
http://www.prairienet.org/bridge/learn2.htm
http://www.interlog.com/~dbd/bridge/bridge1.html

Para jugar por Internet

http://www.okbridge.com
http://www.bridgeplayer.com/~bridge/bptlive.html
http://www.zone.com/bridge.html
http://www.letsplaybridge.com

Servidores de Bridge

nsysu.edu.tw
    tslnet://okbridge@irc.nsysu.edu.tw 4321
    login: okbridge
password: okbridge

ttu.cc
telnet://bridge.bridge/0/zaphod.ttu.cc
login: bridge
password: bridge

Usenet

news.rec.games.bridge
news.rec.games.bridge.okbridge
news.aus.games.bridge
news.nl.sport.bridge

Clubs de Bridge

http://www.bridgeargentina.org.ar
http://mm.mbbhs.edu/awilstein/bridge.html
http://www.cris.com/~scaranza/brdc.html
http://www.bvax.net/~bridge
http://robotics.stanford.edu/users/sbenson/bridgeclub.html
http://student-www.uchicago.edu/RSO/bridgeclub
http://http.es.berkeley.edu/~pike/bridge
http://www.physics.utah.edu/~mjklubridge.html
http://www.umich.edu/~brdgame
http://ourworld.compuserve.com/homepages/tg_bridge
http://watserv1.uwaterloo.ca/~uwbc/uwbc www.html
http://regina.ism.ca/orgs/rdbc.htm
http://info.ox.ac.uk/~kebl0073/bridge.html
http://www.statslab.cam.ac.uk/~jmib/cnbx.html
http://www.es.warwick.ac.uk/~maux
http://www.ncl.ac.uk/~nbridge
http://www.cms.dmuc.ac.uk/~djh/public/Bridge.html
http://www.ace.ie/ace/soes/bridges/index.html
http://www.itse.co.uk/BCBC
http://www-star,qmw.ac.uk/~scg/ULUhtml
http://www.cq.ac.uk/~bridge
http://www0.escv.de/8000/utijsk/Desy_bridge/desy_bridge.html
http://www.ify.hut.fi/~oapitct/index.html
http://www2.e.kth.se/csckt/eng/bridge/KTHBK.html
http://www.csd.uu.se/~d94hje/ibvbxg.html
http://www.uddevall.scbridge
http://www.eks.hiaader.no/~kastrup/bridge.html
http://www.pi.net/~lpn/home.html
http://web.kyoto-inet.or.jp/people/umabou1/indexe.html
http://scsg@unige.ch/bcc/bcc.html

Revistas

http://www.ny-bridge.com/bt.html
http://sun1.tuwien.ac.at/~sh226218/bridge.html
http://mindlink.bc.ca/a10271
http://www.imp-bridge.nl
http://www.phoenix.net/~tbc
http://www.meadowlark
http://www.visual.co.uk/win-core.html
http://www.demon.co.uk/oxford-soft/bridge.html
http://mitigou.com/brumbbarclay
http://www.acbl.org/-acbl/acbl.html
http://www.cbf.ca/query
http://www.bridge.nl

Otras organizaciones

http://www.rtc-gatway.ie/bridge/bridge1.html
http://jeves.uwaterloo.ca/-csuircl/sh Borg/sh Borg.html
http://ecchunx.technion.ac.il/-herbst/bridge.html
http://www.dnai.com/-dealer

Programas, sistemas de juego, páginas personales,
links

http://www.bridgeworld.com
http://www.bridgebaron.com
http://www.q-plus.com
http://www.yahoo.com/Entertainment/Casino/Card_Games/Bridge
http://www.es.vu.nl/-sater/bridge/bridge.html
http://www.gate.net/-predlec
http://www.eslonett.nl/home/peulage/turneriger.html
http://abacus.bates.edu/-rshepard-bridge.html
http://drcce.centerline.com/people/thomaso-bridge.html
http://www.cs.ncl.ac.uk/external/nmorgans/bridge.html
http://www.jhi.hsr.uzw/sting/parsons/bridge.html
http://www.dai.ed.ac.uk/students/ranf/research.html
http://www.intersurf.com/-searamuz
http://www.spyglass.com/-bob
http://www.islandnet.com/-mimo/-warbridge.html
http://www.duct.net/users/H_van_ree
http://www.home.navisoft.com/dendarft/software.html
http://www.club.innet.be/-pub02163
http://www.cs.vu.nl/-sater
http://ftp.tamu.edu/8000/-hwe2233/bridge.html
http://www.bridge.fr.net
http://hamlet.tkp.uni-koei.n.de/-kuehn
http://www.cs.vu.nl/-sater/bridge/bridge-on-the-web.html
http://technion.technion.ac.il/-herbs/bridge.html
http://www.centerline.com/people/thomaso/bridge.html
http://www.rose.com/-acbl166/index.html
http://www.cis.ohio-state.edu/hypertext/fair/s.net-games/bridge-style-guide/faq.html
http://www.cbt.ca/query/Guil.html

ftp://arp.annu.edu.au/pub/Bridge-FAQ
ftp://ftp.cs.vu.nl/pub/-sater/bridge
ftp://ftp.cs.unc.edu/pub/Bridge
ftp://ftp.physik.uni-osnabrueck.de/pub/bridge
Torneos de Bridge

http://www.yahoo.com/Recreation/Games/Card_Games/Bridge/Tournaments
http://www.e-bridge.com
http://www.greatoak.com
http://www.bridgepro.net
http://www.aebf.org/tournaments/tourney_stm
http://denverbudge.com
http://www.bridgeprotour.com
http://www.bridgeplaza.com/tournaments/cavendish/cav98/index.html
http://www.math.auc.dk/~nwp/bridge/tour
http://downtown.aia.net/~beck33
http://www.topbridge.com/www.amich.edu/~bridgecom/tourn
Apéndices

Apéndice 1: Problemas de lógica de satisfacción de restricciones

La programación lógica con satisfacción de restricciones → Constraint Logic Programming (CLP) [BAR 99], [DEC 00] es un paradigma de programación declarativa (de alguna manera variante o mejora del paradigma de programación lógica). Involucra la solución efectiva de problemas basados fuertemente en restricciones variables, particularmente expresables en lógica de primer orden, sobre dominios y teorías varias –como números naturales, números reales, listas eficientes, programación lineal, mancjo de conjuntos, etc.–, así como problemas de propósitos generales, por ej. en áreas de planamiento, organización, selección, decisión, scheduling, control y en general optimización combinatoria. En general se combina el manejo de restricciones -externo- a la programación lógica subyacente.

Un Constraint Satisfaction Problem (CSP) consiste de:

- un número finito de variables
- un dominio de valores al que cada variable pertenece
- un conjunto de restricciones que condiciona el valor de las variables

Una solución al CSP es una asignación de valores del dominio en cada una de las variables en la que todas las restricciones son satisfactas.

La solución puede encontrarse ejecutando una búsqueda sistemática sobre las posibles asignaciones de valores. El algoritmo más común para resolver esto es el llamado backtracking. Existen otros algoritmos [FRO 97] que optimizan de una forma u otra el espacio de búsqueda: backjumping, backmarking, etc.

Algunos ejemplos de uso de CLP:

- coloreo de grafos
- puzzles: N-queens, criptoorimética (SEND+MORE=MONEY), etc
- planeamiento y scheduling

Algunos lenguajes disponibles:

- ECL/PS* [ECL 02]
- CHIP [ICI 90] [COS 02]
- ILOG Solver [ILO 03] es una biblioteca para C++ y Lisp
- SICStus Prolog con biblioteca para CLP sobre dominio finito [SIC 02], [CAR 97], [JAF 87], [HEN 89]
- GNU Prolog [DIA 03]

Para más información acerca de este tema, recomendamos leer [DEC 03], [BAR 03], [EAT 03].

Para nuestro problema, la generación de manos con restricciones, la formulación como CSP resultó no práctica y un tanto tediosa. Si bien el problema involucrado puede ser formulado como CSP distinguiendo dominios, valores y relaciones, esto no parece conducir a una buena manera de analizar las posibles soluciones, más teniendo en cuenta que se desean muchas soluciones al azar y no precisamente una ni tampoco todas, y además el número de nodos y ejes involucrados resultaba extremadamente grande. Por otro lado se podría dar lugar a nodos diferentes provenientes de la representación de manos en donde el orden de las cartas cambia pero la mano no cambia -problema básico en la representación de conjuntos-, lo cual hace aparentemente impráctico el planteo como CSP.

El hecho de haber analizado el problema de otro modo -específico para conjuntos de cartas, sus características y condiciones- nos condujo a otro enfoque distinto para el cual era más apta la formulación -y resolución- usando programación lógica e imperativa.
Resultó práctica la formulación como CLP, en el uso del remate como fuente de restricciones. Aquí el lenguaje es adecuado, no solo para generar las restricciones, sino también para escribir las reglas del sistema de remate: aperturas, respuestas, reaperturas, etc.
Apéndice 2: La PC de testeo y desarrollo

Las pruebas de los generadores estudiados y el desarrollo de SmartGen fueron efectuados en una computadora PC compatible con procesador Pentium II de 500 MH con 64 MB de RAM.

Los sistemas operativos en donde se corrieron los distintos generadores fueron:

- Windows Millennium Edition
- DOS
Apéndice 3: Pruebas de otros generadores

A continuación mostramos algunas corridas de ejemplo hechas con los generadores evaluados en el Capítulo 2. Solo incluimos los generadores considerados más interesantes.

Las tres pruebas que realizamos son:

Test 1: PH(\(N\)) > 30

Test 2: PH(\(N\)) > 35

Test 3: encontrar una mano en la que cada jugador tenga 10 PH y Norte tenga 5 Picas y Oeste 5 Corazones

Y los resultados con SmartGen fueron (sobre 1000 corridas):

<table>
<thead>
<tr>
<th></th>
<th>Promedio de intentos</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>2.31</td>
<td>2.72</td>
</tr>
<tr>
<td>Test 2</td>
<td>13.59</td>
<td>13.08</td>
</tr>
<tr>
<td>Test 3</td>
<td>181.03</td>
<td>180.10</td>
</tr>
</tbody>
</table>

Deal 3

En éste generador, el tiempo de cada consulta puede dividirse en dos etapas:

1) La interpretación del script TCL
2) Encontrar la mano pedida

Cuando el script es complejo el tiempo (1) se hace notorio. Digamos que es un tiempo fljo dependiente de la consulta.

En cambio el tiempo (2) es dependiente de la corrida y de la consulta. Es decir, depende de lo que el motor GAT genere en cada intento.

Las 4 corridas del siguiente test dieron tiempos (2) muy distintos. El (1) en estos ejemplos es despreciable ya que la consulta es simple –en términos del script a interpretar-. Más que tiempos, lo que evaluamos son intentos, es decir cuantas veces se generaron manos hasta que salió la pedida (el programa tiene la capacidad de indicar esto):

\( Test 1: PH(\(N\)) > 30 \)

```c
main {
    if { [hcp north] > 30} { accept }
}
```

Corrida 1: 303153 intentos
Corrida 2: 832362 intentos
Corrida 3: 3640256 intentos
Corrida 4: 1629213 intentos

\( Test 2: PH(\(N\)) > 35 \)

```c
main {
    if { [hcp north] > 35} { accept }
}
```
Generación automática de manos de Bridge con restricciones

Corrida 1: la paramos a los 58997000 intentos. Entendemos que no hay solución en un tiempo considerable (esperamos como 20 minutos)

Test 3: encontrar una mano en la que cada jugador tenga 10 PH y Norte tenga 5 Picas y Oeste 5 Corazones

```c
main {
    if { [hcp north] == 10 && [hcp west] == 10 && [hcp south] == 10 &&
        [hcp east] == 10 && [north spades] == 5 && [west hearts] == 5 }
    {accept}
}
Corrida 1: 45782000 intentos. Tardó varios minutos, pero la encontró.
```

Dealer

Los siguientes son los resultados de las pruebas efectuadas. También incluimos las restricciones ingresadas:

Test 1: \( PH(N) > 30 \)

```c
condition
    hcp(north)>30
produce
    1
Resultado:
```
Corrida 1: 5.44 seg (334347 intentos)
Corrida 2: 12.19 seg (674740 intentos)
Corrida 3: 22.79 seg (2003549 intentos)
Corrida 4: 27.03 seg (2387098 intentos)

Test 2: \( PH(N) > 35 \):

```c
condition
    hcp(north)>35
produce
    1
Resultado:
```
Corrida 1: 10.000.000 intentos y no encontró solución (115.17 seg )

Test 3: retornar una mano en la que cada jugador tenga 10 puntos de honor, Norte tenga 5 pica y Oeste 5 corazones:

```c
condition
    hcp(north)==10 and hcp(west)==10 and hcp(south)==10 and hcp(east)==10
    and spades(north)==5 and hearts(west)==5
produce
    1
Resultado:
```

Pag. 70
Corrida 1: 29628 intentos
Corrida 2: 57983 intentos
Corrida 3: 134 intentos
Apéndice 4: SmartGen

SmartGen es un generador de manos de Bridge que surge del presente estudio y análisis. Implementa las heurísticas explicadas en los Capítulos 4 y 5. Fue escrito en lenguaje Delphi [DEL 98], es decir, Pascal Orientado a Objetos en entorno MS Windows. Elegimos Delphi porque es un lenguaje de propósito general, que permite una codificación elegante y que facilita la creación de aplicaciones con una interfaz de usuario elaborada (con la posibilidad de reuso de componentes) y a la vez simple de usar. Cabe aclarar que podríamos haber elegido otro ya que los algoritmos pueden ser escritos en cualquier lenguaje imperativo casi sin cambios. De alguna manera, hemos elegido un lenguaje imperativo dado la naturaleza de los algoritmos usados (en particular muchas de las rutinas usadas), ya que no se han descripto estos en forma declarativa sino procedural con el fin de tener el control de todos los mecanismos involucrados en detalle.

SmartGen tiene dos modos de funcionamiento:

- Smart
- GAT

El único propósito del modo GAT es comparar resultados contra el modo Smart.
En el modo Smart o Dirigido, se ejecutan las heurísticas de distribución y puntos de honor

Las restricciones que se pueden definir para cada jugador, incluyen tres grupos:

- Restricciones sobre la distribución
- Restricciones sobre los puntos de honor
- Cartas preasignadas

Entre las restricciones sobre distribución, se pueden especificar:

- Mano libre (sin restricción sobre la distribución)
- Mano balanceada
- Mano entre un rango dado de puntos de distribución
- Mano que cumple cierto patrón. Se incluyen fallos, semifallos, Dobletones, tricolor, etc.
- Mano con rango de cantidad de cartas por palo

Entre las restricciones sobre puntos de honor, tenemos:

- Mano libre (sin restricción sobre PH)
- Mano entre un rango dado de PH
- Mano con rango de PH para cada palo

En cuanto a la generación de manos, se puede especificar dos formas:

- Una sola mano
- Sin limite: el programa genera manos hasta que se le indique lo contrario

Además de mostrar las cartas generadas, distribución, PH por palo, etc, SmartGen calcula el número de intentos promedio que le llevó encontrar la mano. Con ello es muy fácil conmutar entre modo GAT y Smart y comparar los resultados.

Existe una tercer forma de generación, llamada estadística, en la que se le indica un número máximo de corridas tanto para el modo GAT como Smart. Aquí no muestra las cartas encontradas, sino información comparativa entre ambos modos: número de intentos promedio, desviación standard de los intentos, cantidad de éxitos (manos encontradas que cumplen las restricciones), cantidad de fracasos (manos sin encontrar) y tiempo (ms) de ejecución del test.

En el siguiente Apéndice incluimos varias corridas a modo de ejemplo.
El código fuente está incluido en el diskette adjunto. Los archivos principales son:

- **SmartGen.dpr**: proyecto Delphi
- **SGMain.pas/SGMain.dfm**: diálogo principal, armado de restricciones, y cálculo de estadística.
- **Gen.pas**: heurísticas, procedimientos de generación (Smart y GAT) y verificación de restricciones.

Los fuentes están comentados y hacen referencia a los algoritmos citados por su letra-número.

**Requerimiento**

SmartGen corre en PC compatible con MS Windows 95 o posterior.

**Instalación**

Para correr el programa, solo es necesario ejecutar el archivo SmartGen.exe.
Apéndice 5: Pruebas de SmartGen

Los siguientes son ejemplos de corridas de SmartGen comparadas con un generador GAT. Algunas corridas, además de tener valor demostrativo, tienen valor desde el punto de vista del juego.

Por cada corrida mostramos las manos de los 4 jugadores, indicando:

- Cartas por palo (en el orden pica, corazón, diamante y trébol)
- Distribución (en el mismo orden)
- Puntos de Honor totales y por palo
- Puntos de distribución
- Bajas perdedoras (losing tricks)

Además informamos el número de intentos en el que surgió la mano

Cuando mostramos información comparativa con GAT, se menciona:

- Cantidad de corridas
- Promedio de intentos
- Cantidad de éxitos, es decir, veces en que se encontró una mano
- Cantidad de fracasos, es decir, si al cabo de 30,000 intentos no se encontró mano, se cuenta un fracaso.
- Desviación standard del número de intentos, que demuestra qué tanto se converge a la solución
- Tiempo en que tardó en generar todas las corridas, en milisegundos.

La cantidad de corridas depende del ejemplo. En general las hicimos con 1000, pero en los casos más pesados (casos lentos en que se tarda en resolver), las bajamos a 100.
1. Ejemplo: Un jugador con muchos puntos de honor.

Descripción
Sur con PM1 > 25

Restricciones
NORTE: Dist. libre & PH libre
ESTE: Dist. libre & PH libre
SUR: Dist. libre & PH: 25-37
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 1 (Smart) ---
--- Norte --- | --- Este --- | --- Sur --- | --- Oeste ---
♦ T94   ♠ Q53   ♠ AKJ   ♠ 8762
♥ J63   ♥ T96542 ♦ AKQ   ♥ 9873
♦ K732   ♠ 654   ♠ AQ39   ♦ T8
♦ Q9654 ♠ 2     ♠ AK7   ♠ T83
Dist: 3 0 4 6    Dist: 3 6 3 1  | Dist: 3 3 4 3  | Dist: 4 4 2 3
PH: 6 (0 0 3 3)  PH: 2 (2 0 0 0)  | PH: 31 (8 9 7 7) PH: 1 (0 1 0 0)
PD: 3           PD: 2           PD: 0           PD: 1
LT: 7           LT: 9           LT: 3           LT: 11

--- #2 - Intento 1 (Smart) ---
--- Norte --- | --- Este --- | --- Sur --- | --- Oeste ---
♦ K8532 ♠ J63   ♠ AQ    ♠ T9764
♠ 886   ♠ QT8532 ♠ A4    ♠ K97
♠ 32     ♠ JT754 ♠ AKJ07 ♠ T43
Dist: 6 2 3 2    Dist: 0 6 2 5  | Dist: 2 2 5 4  | Dist: 5 3 3 2
PH: 7 (4 2 1 0)  PH: 3 (0 2 0 1)  | PH: 27 (6 4 8 9) PH: 3 (0 3 0 0)
PD: 2           PD: 4           PD: 2           PD: 1
LT: 8           LT: 7           LT: 3           LT: 10

--- #3 - Intento 2 (Smart) ---
--- Norte --- | --- Este --- | --- Sur --- | --- Oeste ---
♦ 5     ♠ T3     ♠ AKJ6   ♠ Q97842
♥ Q97432 ♠ J16     ♠ AK    ♠ 85
♦ Q87653 ♠ 94     ♠ AKJ3 ♠ 12
Dist: 1 6 6 0    Dist: 2 3 2 6  | Dist: 4 2 3 4  | Dist: 6 2 2 3
PH: 4 (0 2 2 0)  PH: 1 (0 1 0 0)  | PH: 30 (8 7 8 7) PH: 5 (2 0 0 3)
PD: 5           PD: 2           PD: 1           PD: 2
LT: 5           LT: 10          LT: 3           LT: 8

--- #4 - Intento 3 (Smart) ---
--- Norte --- | --- Este --- | --- Sur --- | --- Oeste ---
♦ JT9865432 ♠  -     ♠ AKQ    ♠ 9
♥ 3     ♠ T987    ♠ AKQ    ♠ J6542
♦ 942   ♠ K76     ♠ AQ8    ♠ 7653
Dist: 9 1 0 3    Dist: 0 4 5 4  | Dist: 3 3 4 3  | Dist: 1 5 4 3
PH: 1 (1 0 0 0)  PH: 5 (0 0 1 4)  | PH: 31 (9 9 4 4) PH: 3 (0 1 0 2)
PD: 5           PD: 3           PD: 0           PD: 2
LT: 7           LT: 8           LT: 2           LT: 9

Comparación

*** Smart ***
Corridas: 1000
Promedio Intentos: 1.28
Exitos: 1000
Sin mano: 0
Desviación std: 0.58
Tiempo (ms): 190

*** GAT ***
Corridas: 1000
Promedio Intentos: 1050.84
Exitos: 1000
Sin mano: 0
Desviación std: 1036.07
Tiempo (ms): 40905
2. Ejemplo: Dos jugadores con muchos puntos de honor

Descripción
Norte con PH >= 15 y Este con PH >= 15

Restricciones
NORTE: Dist. libre & PH: 15-37
ESTE: Dist. libre & PH: 15-37
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 12 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Su</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQ74</td>
<td>♠ J632</td>
<td>♣ 9</td>
<td>♠ 85</td>
</tr>
<tr>
<td>♥ K75</td>
<td>♥ A6</td>
<td>♥ Q382</td>
<td>♥ T943</td>
</tr>
<tr>
<td>♦ A2</td>
<td>♦ KQ754</td>
<td>♦ T9</td>
<td>♦ 863</td>
</tr>
<tr>
<td>♣ Q</td>
<td>♣ AK</td>
<td>♣ J97643</td>
<td>♣ T852</td>
</tr>
<tr>
<td>Dist: 6 3 3 1</td>
<td>Dist: 4 2 5 2</td>
<td>Dist: 1 4 2 6</td>
<td>Dist: 2 4 3 4</td>
</tr>
<tr>
<td>PH: 19 (9 5 3 2)</td>
<td>PH: 17 (1 4 5 7)</td>
<td>PH: 4 (0 3 0 1)</td>
<td>PH: 0 (0 0 0 0)</td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 2</td>
<td>PD: 3</td>
<td>PD: 1</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 5</td>
<td>LT: 8</td>
<td>LT: 11</td>
</tr>
</tbody>
</table>

--- #2 - Intento 2 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Su</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AQ97</td>
<td>♠ K87632</td>
<td>♣ -</td>
<td>♠ T54</td>
</tr>
<tr>
<td>♥ AQ74</td>
<td>♥ K</td>
<td>♥ 52</td>
<td>♥ T9863</td>
</tr>
<tr>
<td>♦ K962</td>
<td>♦ A4</td>
<td>♦ 3</td>
<td>♦ Q8753</td>
</tr>
<tr>
<td>Dist: 4 5 0 4</td>
<td>Dist: 6 1 3 3</td>
<td>Dist: 0 2 10 1</td>
<td>Dist: 3 5 0 5</td>
</tr>
<tr>
<td>PH: 17 (7 0 3 2)</td>
<td>PH: 15 (3 3 5 4)</td>
<td>PH: 6 (0 0 5 1)</td>
<td>PH: 2 (0 0 0 2)</td>
</tr>
<tr>
<td>PD: 3</td>
<td>PD: 2</td>
<td>PD: 6</td>
<td>PD: 3</td>
</tr>
<tr>
<td>LT: 4</td>
<td>LT: 6</td>
<td>LT: 5</td>
<td>LT: 8</td>
</tr>
</tbody>
</table>

--- #3 - Intento 2 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Su</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AK</td>
<td>♠ Q652</td>
<td>♣ 743</td>
<td>♠ J98</td>
</tr>
<tr>
<td>♥ T9852</td>
<td>♥ AK39</td>
<td>♥ 7643</td>
<td>♥ -</td>
</tr>
<tr>
<td>♦ AK33</td>
<td>♦ 87</td>
<td>♦ T642</td>
<td>♦ Q85</td>
</tr>
<tr>
<td>♣ A3</td>
<td>♣ KQ5</td>
<td>♣ 82</td>
<td>♣ T97643</td>
</tr>
<tr>
<td>Dist: 2 5 4 2</td>
<td>Dist: 4 4 2 3</td>
<td>Dist: 3 4 4 2</td>
<td>Dist: 4 0 3 6</td>
</tr>
<tr>
<td>PH: 20 (7 0 8 3)</td>
<td>PH: 17 (2 1 0 5)</td>
<td>PH: 0 (0 0 0 0)</td>
<td>PH: 3 (1 0 2 0)</td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 1</td>
<td>PD: 1</td>
<td>PD: 3</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 5</td>
<td>LT: 11</td>
<td>LT: 8</td>
</tr>
</tbody>
</table>

--- #4 - Intento 7 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Su</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AK4</td>
<td>♠ Q</td>
<td>♣ I86</td>
<td>♠ T9752</td>
</tr>
<tr>
<td>♥ A18</td>
<td>♥ KQ35</td>
<td>♥ 965</td>
<td>♥ T74</td>
</tr>
<tr>
<td>♦ A97</td>
<td>♦ KQT862</td>
<td>♦ 543</td>
<td>♦ 3</td>
</tr>
<tr>
<td>♣ AK7</td>
<td>♣ Q3</td>
<td>♣ 9853</td>
<td>♣ T642</td>
</tr>
<tr>
<td>Dist: 4 3 3 3</td>
<td>Dist: 1 4 6 2</td>
<td>Dist: 1 3 3 4</td>
<td>Dist: 5 3 1 4</td>
</tr>
<tr>
<td>PH: 23 (7 5 4 7)</td>
<td>PH: 15 (2 5 5 3)</td>
<td>PH: 1 (1 0 0 0)</td>
<td>PH: 1 (0 0 1 0)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 3</td>
<td>PD: 0</td>
<td>PD: 2</td>
</tr>
<tr>
<td>LT: 6</td>
<td>LT: 5</td>
<td>LT: 12</td>
<td>LT: 10</td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 1000
Promedio Intentos: 3.76
Exitos: 1000
Sin mano: 0
Desviación std: 3.21
Tiempo (ms): 470

*** GAT ***
Corridas: 1000
Promedio Intentos: 271.93
Exitos: 1000
Sin mano: 0
Desviación std: 260.03
Tiempo (ms): 12200

Pag. 76
3. Ejemplo: Un jugador con muchos puntos de distribución.

Descripción
Norte con PD >= 7

Restricciones
NORTE: PD: 7-9 & PH libre
ESTE: Dist. libre & PH libre
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 1 (Smart) ---
  === Norte ===  === Este ===  === Sur ===  === Oeste ===
  ♠ -  ♠ J983  ♠ AKt6  ♠ Q7542
  ♥ AK9T98765432  ♥ -  ♥ -  ♥ -
  ♦ -  ♦ K9752  ♦ Q364  ♦ AT83
  □ -  □ A384  □ K9632  □ QT75
  Dist: 01300  Dist: 4054  Dist: 4045  Dist: 5044
  PH: 10  (01000) PH: 9  (1035) PH: 13  (7033) PH: 8  (2042)
  PD: 9  PD: 3  PD: 3  PD: 3
  LT: 0  LT: 7  LT: 5  LT: 6

--- #2 - Intento 1 (Smart) ---
  === Norte ===  === Este ===  === Sur ===  === Oeste ===
  ♠ J  ♠ Q3  ♠ AK9T9876542  ♠ -
  ♥  ♥ A6432  ♥ Q35  ♥ K987
  ♦ -  ♦ T87654  ♦ -  ♦ AK9J32
  Dist: 11200  Dist: 2056  Dist: 10030  Dist: 0157
  PH: 11  (10000) PH: 6  (2040) PH: 10  (7030) PH: 13  (00310)
  PD: 8  PD: 4  PD: 6  PD: 5
  LT: 1  LT: 7  LT: 3  LT: 3

--- #3 - Intento 1 (Smart) ---
  === Norte ===  === Este ===  === Sur ===  === Oeste ===
  ♠ -  ♠ QT3  ♠ KJ7642  ♠ A985
  ♥ AK9T98765432  ♥ -  ♥ -  ♥ -
  ♦ -  ♦ A98765  ♦ K794  ♦ J32
  Dist: 01300  Dist: 3064  Dist: 6043  Dist: 4036
  PH: 10  (01000) PH: 12  (2064) PH: 11  (4034) PH: 7  (4012)
  PD: 9  PD: 3  PD: 3  PD: 3
  LT: 0  LT: 5  LT: 6  LT: 7

--- #4 - Intento 1 (Smart) ---
  === Norte ===  === Este ===  === Sur ===  === Oeste ===
  ♠ -  ♠ AK98752  ♠ 64  ♠ QT3
  ♥ -  ♥ JT7654  ♥ A98  ♥ KQ32
  ♦ AK9T98765432  ♦ -  ♦ -  ♦ -
  Dist: 00310  Dist: 7600  Dist: 2308  Dist: 4405
  PH: 10  (00100) PH: 8  (7100) PH: 7  (0403) PH: 15  (3507)
  PD: 9  PD: 6  PD: 4  PD: 3
  LT: 0  LT: 4  LT: 6  LT: 4

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1.00
Exitos: 100
Sin mano: 0
Desviación std: 0.00
Tiempo (ms): 30

*** GAT ***
Corridas: 100
Promedio Intentos: 84264.30
Exitos: 30
Sin mano: 70 (sobre 30000 intentos por corrida)
Desviación std: 8795.73
Tiempo (ms): 74865
4. Ejemplo: Dos jugadores con muchos puntos de distribución.

Descripción

Norte con PD >= 5 y Este con PD >= 5

Restricciones

NORTE: PD: 5-9 & PH libre
ESTE: PD: 5-9 & PH libre
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados

--- #1 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>---</th>
<th>Este</th>
<th>---</th>
<th>Sur</th>
<th>---</th>
<th>Oeste</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ A</td>
<td>♠ 7</td>
<td>♠ KJT98654</td>
<td>♠ 32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♥ KJ987642</td>
<td>♥ A</td>
<td>♥ -</td>
<td>♥ 53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♦ Q</td>
<td>♦ -</td>
<td>♦ KJT98765432</td>
<td>♦ K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♣ 3</td>
<td>♣ -</td>
<td>♣ KJT9862</td>
<td>♣ AQ3754</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dist: 1 10 11</td>
<td>Dist: 1 11 0</td>
<td>Dist: 8 0 5</td>
<td>Dist: 3 2 1 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH: 12 (4 6 2)</td>
<td>PH: 9 (0 4 5 0)</td>
<td>PH: 7 (4 0 0 3)</td>
<td>PH: 12 (2 0 3 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD: 6</td>
<td>PD: 7</td>
<td>PD: 6</td>
<td>PD: 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 3</td>
<td>LT: 3</td>
<td>LT: 4</td>
<td>LT: 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--- #2 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>---</th>
<th>Este</th>
<th>---</th>
<th>Sur</th>
<th>---</th>
<th>Oeste</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQJT98765432</td>
<td>♠ -</td>
<td>♠ -</td>
<td>♠ -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♥ -</td>
<td>♥ A54</td>
<td>♥ KJT762</td>
<td>♥ QJ983</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♦ -</td>
<td>♦ 6</td>
<td>♦ AKJT</td>
<td>♦ Q9875432</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♣ -</td>
<td>♣ AKJT85432</td>
<td>♣ Q976</td>
<td>♣ -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dist: 13 0 0 0</td>
<td>Dist: 0 3 1 9</td>
<td>Dist: 0 5 4 4</td>
<td>Dist: 0 5 8 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH: 10 (0 0 0 0)</td>
<td>PH: 12 (0 4 0 8)</td>
<td>PH: 13 (0 3 8 2)</td>
<td>PH: 5 (0 3 2 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD: 9</td>
<td>PD: 5</td>
<td>PD: 3</td>
<td>PD: 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 0</td>
<td>LT: 4</td>
<td>LT: 5</td>
<td>LT: 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--- #3 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>---</th>
<th>Este</th>
<th>---</th>
<th>Sur</th>
<th>---</th>
<th>Oeste</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>♣ 5</td>
<td>♣ KJT96432</td>
<td>♣ JT</td>
<td>♣ AQ8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♥ 8</td>
<td>♥ -</td>
<td>♥ AKQJT96452</td>
<td>♥ J73</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♠ 9</td>
<td>♠ 6</td>
<td>♠ AKT432</td>
<td>♠ 95</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♦ AKQJT987532</td>
<td>♦ -</td>
<td>♦ Q9876</td>
<td>♦ 64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dist: 1 10 11</td>
<td>Dist: 7 0 6 0</td>
<td>Dist: 2 9 2 0</td>
<td>Dist: 3 3 5 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH: 10 (0 0 0 0)</td>
<td>PH: 10 (3 0 7 0)</td>
<td>PH: 10 (1 9 0 0)</td>
<td>PH: 10 (6 1 3 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD: 7</td>
<td>PD: 6</td>
<td>PD: 5</td>
<td>PD: 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 2</td>
<td>LT: 3</td>
<td>LT: 4</td>
<td>LT: 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--- #4 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>---</th>
<th>Este</th>
<th>---</th>
<th>Sur</th>
<th>---</th>
<th>Oeste</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ QJT9765432</td>
<td>♠ -</td>
<td>♠ -</td>
<td>♠ AK8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♥ A</td>
<td>♥ J</td>
<td>♥ KQT96432</td>
<td>♥ T964</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♦ 3</td>
<td>♦ -</td>
<td>♦ AKQ9832</td>
<td>♦ QT7564</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>♣ K</td>
<td>♣ AQJT97865432</td>
<td>♣ -</td>
<td>♣ -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dist: 10 1 1</td>
<td>Dist: 0 1 0 12</td>
<td>Dist: 0 7 6 0</td>
<td>Dist: 3 4 6 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH: 11 (3 4 1 3)</td>
<td>PH: 8 (0 1 0 7)</td>
<td>PH: 12 (0 5 7 0)</td>
<td>PH: 9 (7 0 2 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD: 6</td>
<td>PD: 8</td>
<td>PD: 6</td>
<td>PD: 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 4</td>
<td>LT: 2</td>
<td>LT: 2</td>
<td>LT: 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1.36
Exitos: 100
Sín mano: 0
Desviación std: 0.67
Tiempo (ms): 35

*** GAT ***
Corridas: 100
Promedio Intentos: 34191.08
Exitos: 59
Sín mano: 41 (sobre 30000 intentos por corrida)
Desviación std: 10521.94
Tiempo (ms): 60340
5. Ejemplo: Un jugador con suit largo y otro con PH alto

Descripción
Palo largo en Norte y PH alto en Este

Restricciones
NORTE: Patrón 7.x.x.x & PH libre
ESTE: Dist. libre & PH: 15-37
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 1 (Smart) ---
  Norte ---  Este ---  Sur ---  Oeste ---
  ♠ J9765  ♠ K8  ♠ Aq  ♠ 432
  ♥ K97632  ♥ AJ  ♥ T  ♥ 854
  ♦ -  ♦ AKJ5  ♦ Q96432  ♦ 7
  ♣ -  ♣ Q963  ♣ 854  ♣ AKJ72

  Dist: 6 7 0 0  Dist: 2 2 5 4  Dist: 2 1 7 3  Dist: 3 3 1 6
  PH: 6 (1 5 0 0)  PH: 18 (3 5 8 2)  PH: 8 (6 0 2 0)  PH: 8 (0 0 0 8)
  PD: 6  PD: 2  PD: 3  PD: 2
  LT: 4  LT: 5  LT: 7  LT: 8

--- #2 - Intento 2 (Smart) ---
  Norte ---  Este ---  Sur ---  Oeste ---
  ♠ A2  ♠ -  ♠ KQ1974  ♠ 8653
  ♥ AQ87542  ♥ KJ963  ♥ -  ♥ -
  ♦ T83  ♦ AKJ  ♦ Q2  ♦ 97654
  ♣ -  ♣ Aq17  ♣ 8532  ♣ K964

  Dist: 2 7 3 1  Dist: 0 6 3 4  Dist: 7 0 2 4  Dist: 4 0 5 4
  PH: 10 (4 6 0 0)  PH: 19 (0 4 8 7)  PH: 8 (6 0 2 0)  PH: 3 (0 0 0 3)
  PD: 3  PD: 4  PD: 4  PD: 3
  LT: 6  LT: 4  LT: 6  LT: 8

--- #3 - Intento 3 (Smart) ---
  Norte ---  Este ---  Sur ---  Oeste ---
  ♠ 2  ♠ A8643  ♠ Q  ♠ KJ9745
  ♥ 754  ♥ T62  ♥ KJ83  ♥ AQ9
  ♦ K7  ♦ A  ♦ J9842  ♦ Q653
  ♣ KT86532  ♣ AQJ9  ♣ 74  ♣ -

  Dist: 1 3 2 7  Dist: 3 1 4  Dist: 1 4 6 2  Dist: 6 3 4 0
  PH: 6 (0 0 3 3)  PH: 15 (4 0 4 7)  PH: 7 (2 4 1 0)  PH: 12 (4 6 2 0)
  PD: 3  PD: 2  PD: 3  PD: 3
  LT: 7  LT: 6  LT: 8  LT: 5

--- #4 - Intento 1 (Smart) ---
  Norte ---  Este ---  Sur ---  Oeste ---
  ♠ 7642  ♠ KQ  ♠ 1953  ♠ AT8
  ♥ AK97642  ♥ QT53  ♥ J8  ♥ -
  ♦ -  ♦ AQ  ♦ 93  ♦ KJ976542
  ♣ QT  ♣ K7642  ♣ A853  ♣ 9

  Dist: 4 7 0 2  Dist: 2 4 2 5  Dist: 4 2 2 5  Dist: 3 0 9 1
  PH: 9 (0 7 0 2)  PH: 16 (5 2 6 3)  PH: 7 (1 1 0 5)  PH: 8 (4 0 4 0)
  PD: 4  PD: 2  PD: 2  PD: 5
  LT: 6  LT: 6  LT: 9  LT: 5

Comparación

*** Smart ***
Corridas: 1000
Promedio Intentos: 1.50
Exitos: 1000
Sin mano: 0
Desviación std: 0.85
Tiempo (ms): 300

*** GAT ***
Corridas: 1000
Promedio Intentos: 92.63
Exitos: 1000
Sin mano: 0
Desviación std: 91.22
Tiempo (ms): 3365
Generación automática de manos de Bridge con restricciones

6. Ejemplo: Dos jugadores balanceados y los restantes con fallos

Descripción
Manos S1 para Norte y Este. Manos con fallos en los restantes jugadores
Norte y Este balanceados con PH >= 13. Sur y Oeste con PD >= 6

Restricciones
NORTE: Dist. balanceada & PH: 13-37
ESTE: Dist. balanceada & PH: 13-37
SUR : PD: 6-9 & PH libre
OESTE: PD: 6-9 & PH libre

Resultados
--- #1 - Intento 981 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
♠ A16    ♠ K95    ♠ T987432  ♠ -
♥ K76    ♥ AQ5    ♥ KT8432    ♥ -
♦ Q985    ♦ AK7    ♦ -    ♦ JT6432
♣ AKQ    ♣ T83    ♣ -    ♣ J976542
Dist: 3 3 4 3    Dist: 3 4 3 3    Dist: 7 6 0 0    Dist: 0 0 6 7
PM: 17 (5 1 2 9)    PH: 18 (5 6 7 0)    PH: 3 (0 3 0 0)    PH: 2 (0 0 1 1)
P D: 0    P D: 0    P D: 6    P D: 6
L T: 7    L T: 6    L T: 5    L T: 6
--- #2 - Intento 976 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
♠ Q76    ♠ AKJ7    ♠ 985432    ♠ -
♥ 852    ♥ KQ3    ♥ -    ♥ A9T9764
♦ AKQ    ♦ 965    ♦ JT87432    ♦ -
♣ A152    ♣ K83    ♣ -    ♣ Q97964
Dist: 3 3 4 3    Dist: 4 3 3 3    Dist: 6 0 7 0    Dist: 0 7 0 6
PM: 16 (2 0 9 5)    PH: 16 (5 0 5 3)    PH: 1 (0 0 1 0)    PH: 7 (0 5 0 2)
P D: 0    P D: 0    P D: 6    P D: 6
L T: 7    L T: 7    L T: 6    L T: 4
--- #3 - Intento 202 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
♠ AKJ2    ♠ 75    ♠ QT9643    ♠ -
♥ Q6    ♥ AKJ    ♥ -    ♥ T9875432
♦ Q954    ♦ AKJ    ♦ JT8762    ♦ -
♣ AK7    ♣ T8653    ♣ -    ♣ Q9742
Dist: 4 2 4 3    Dist: 2 3 3 5    Dist: 7 0 6 0    Dist: 0 8 0 5
PM: 20 (4 2 8 6)    PH: 15 (4 0 7 0)    PH: 3 (2 0 1 0)    PH: 2 (0 0 0 2)
P D: 1    P D: 1    P D: 6    P D: 6
L T: 6    L T: 7    L T: 5    L T: 5
--- #4 - Intento 1001 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
♠ AKJ    ♠ T84    ♠ -    ♠ QT96533
♥ AKJ    ♥ QT3    ♥ -    ♥ T98654
♦ 974    ♦ AKJ    ♦ QT86532    ♦ -
♣ AQ6    ♣ KT3    ♣ 987542    ♣ -
Dist: 3 3 4 3    Dist: 3 4 3 3    Dist: 0 0 7 6    Dist: 7 6 0 0
PM: 23 (8 8 0 7)    PH: 13 (0 2 8 3)    PH: 2 (0 0 2 0)    PH: 2 (0 0 0 0)
P D: 0    P D: 0    P D: 6    P D: 6
L T: 6    L T: 8    L T: 5    L T: 5

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1985.76
Exitos: 100
Sin mano: 0
Desviación std: 1706.34
Tiempo (ms): 33955

*** GAT ***
Corridas: 100
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)

GAT no generó manos
7. Ejemplo: Dos jugadores con muchas cartas del mismo palo

Descripción
Norte y Este con muchas cartas de corazón.
Norte con cartas altas de corazón menos el As. Este con As de corazones.

Restricciones
NORTE: P:1r & C:5-13 & D:5r & T:5r & PH:1r & PHC:6-10 & PHT:5r & PHD:2r
ESTE: P:1r & C:5-13 & D:5r & T:5r & PH libre & { AC }
 SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

<table>
<thead>
<tr>
<th>Resultados</th>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A - A</td>
<td>A A2</td>
<td>A 9763</td>
<td>A KQJ854</td>
</tr>
<tr>
<td></td>
<td>♥ KQJ94</td>
<td>♥ AT876532</td>
<td>♥ 74</td>
<td>♥ 74</td>
</tr>
<tr>
<td></td>
<td>♥ AKQ</td>
<td>♥ 2</td>
<td>♥ 1987654</td>
<td>♥ T3</td>
</tr>
<tr>
<td></td>
<td>♥ KQJ83</td>
<td>♥ A5</td>
<td>♥ 74</td>
<td>♥ 962</td>
</tr>
<tr>
<td>Dist: 0535</td>
<td>Dist: 2812</td>
<td>Dist: 4072</td>
<td>Dist: 7024</td>
<td></td>
</tr>
<tr>
<td>PH: 20(0695)</td>
<td>PH: 12(4404)</td>
<td>PH: 1(0010)</td>
<td>PH: 7(6001)</td>
<td></td>
</tr>
<tr>
<td>PD: 3</td>
<td>PD: 4</td>
<td>PD: 4</td>
<td>PD: 4</td>
<td></td>
</tr>
<tr>
<td>LT: 2</td>
<td>LT: 5</td>
<td>LT: 8</td>
<td>LT: 6</td>
<td></td>
</tr>
<tr>
<td>--- #2 - Intento 1 (Smart) ---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norte</td>
<td>Este</td>
<td>Sur</td>
<td>Oeste</td>
<td></td>
</tr>
<tr>
<td></td>
<td>J7</td>
<td>-</td>
<td>A1798</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ KQJT24</td>
<td>♥ A8763</td>
<td>♥ 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ AQ</td>
<td>♥ JT94</td>
<td>♥ 872</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ AKQ</td>
<td>♥ T1754</td>
<td>♥ 98632</td>
<td></td>
</tr>
<tr>
<td>Dist: 2524</td>
<td>Dist: 0544</td>
<td>Dist: 4135</td>
<td>Dist: 7240</td>
<td></td>
</tr>
<tr>
<td>PH: 23(16610)</td>
<td>PH: 5(4100)</td>
<td>PH: 4(4000)</td>
<td>PH: 8(5030)</td>
<td></td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 3</td>
<td>PD: 2</td>
<td>PD: 4</td>
<td></td>
</tr>
<tr>
<td>LT: 4</td>
<td>LT: 8</td>
<td>LT: 9</td>
<td>LT: 5</td>
<td></td>
</tr>
<tr>
<td>--- #3 - Intento 15 (Smart) ---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norte</td>
<td>Este</td>
<td>Sur</td>
<td>Oeste</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A9</td>
<td>J7</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ KQ853</td>
<td>♥ AT97642</td>
<td>♥ QT42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ J63</td>
<td>♥ A87</td>
<td>♥ K95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ A3</td>
<td>♥ T</td>
<td>♥ KQ86532</td>
<td>♥ 974</td>
</tr>
<tr>
<td>Dist: 2632</td>
<td>Dist: 2731</td>
<td>Dist: 7047</td>
<td>Dist: 7033</td>
<td></td>
</tr>
<tr>
<td>PH: 10(7615)</td>
<td>PH: 9(1440)</td>
<td>PH: 7(025)</td>
<td>PH: 5(2030)</td>
<td></td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 3</td>
<td>PD: 4</td>
<td>PD: 3</td>
<td></td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 7</td>
<td>LT: 5</td>
<td>LT: 7</td>
<td></td>
</tr>
<tr>
<td>--- #4 - Intento 6 (Smart) ---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norte</td>
<td>Este</td>
<td>Sur</td>
<td>Oeste</td>
<td></td>
</tr>
<tr>
<td></td>
<td>K98</td>
<td>96</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ KQJT3653</td>
<td>♥ AT8742</td>
<td>♥ 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ AK3</td>
<td>♥ QT5</td>
<td>♥ T86432</td>
<td></td>
</tr>
<tr>
<td></td>
<td>♥ -</td>
<td>♥ T7</td>
<td>♥ AK532</td>
<td></td>
</tr>
<tr>
<td>Dist: 3730</td>
<td>Dist: 2632</td>
<td>Dist: 7065</td>
<td>Dist: 6016</td>
<td></td>
</tr>
<tr>
<td>PH: 19(5680)</td>
<td>PH: 6(0420)</td>
<td>PH: 7(007)</td>
<td>PH: 8(5003)</td>
<td></td>
</tr>
<tr>
<td>PD: 3</td>
<td>PD: 2</td>
<td>PD: 4</td>
<td>PD: 5</td>
<td></td>
</tr>
<tr>
<td>LT: 3</td>
<td>LT: 8</td>
<td>LT: 6</td>
<td>LT: 5</td>
<td></td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 1000
Promedio Intentos: 7.83
Exitos: 1000
Sin mano: 0
Desviación std: 7.22
Tiempo (ms): 740

*** GAT ***
Corridas: 1000
Promedio Intentos: 123.31
Exitos: 1000
Sin mano: 0
Desviación std: 126.90
Tiempo (ms): 4185
8. Ejemplo: Un palo muy largo en cada jugador

Descripción

Restricciones
  NORTE: Patrón 8.x.x.x & PH libre
  ESTE: Patrón 8.x.x.x & PH libre
  SUR: Patrón 8.x.x.x & PH libre
  OESTE: Patrón 8.x.x.x & PH libre

Resultados

--- #1 - Intento 30 (Smart) ---

  NORTE ==  KQ98542  ==  KJT7653  ==  AQ85  ==  KT4  ==  7  ==  AT63
  ESTE ==  Q2  ==  KT987654  ==  42  ==  J76  ==  2  ==  3

--- #2 - Intento 2 (Smart) ---

  NORTE ==  8743  ==  AQJT9652  ==  743  ==  3
  ESTE ==  Q2  ==  KT987654  ==  42  ==  J76  ==  2  ==  3

--- #3 - Intento 1 (Smart) ---

  NORTE ==  KQ98762  ==  AJS4  ==  J786  ==  KT4  ==  5
  ESTE ==  9763  ==  AKQJ7542  ==  5  ==  2

--- #4 - Intento 1 (Smart) ---

  NORTE ==  AT  ==  KQ987632  ==  9  ==  54
  ESTE ==  AKQ98754  ==  Q2  ==  T6  ==  3

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 17.79
Éxitos: 100
Sin mano: 0
Desviación std: 18.89
Tiempo (ms): 300

*** GAT ***
Corridas: 100
Promedio Intentos: 80711.94
Éxitos: 31
Sin mano: 69 (sobre 30000 intentos por corrida)
Desviación std: 8930.48
Tiempo (ms): 94580
9. Ejemplo: Doblo informativo

Descripción
Norte con PH > 13 y palo alto cantable (1 pica). Este con fallo en el palo alto cantado y los otros tres palos altos con cantidad pareja de cartas y PHD ≥ 13

Restricciones
ESTE: P:10 & C:4-13 & D:4-13 & T:4-13 & PH: 10-37
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados

--- #1 - Intento 4 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQT96543</td>
<td>♠ -</td>
<td>♠ -</td>
<td>♠ 3872</td>
</tr>
<tr>
<td>♥ Q</td>
<td>♥ AKJ2</td>
<td>♥ T97643</td>
<td>♥ 85</td>
</tr>
<tr>
<td>♦ K76</td>
<td>♦ AQJ3</td>
<td>♦ T95</td>
<td>♦ 842</td>
</tr>
<tr>
<td>♠ -</td>
<td>♠ T9754</td>
<td>♠ QJ32</td>
<td>♠ AK86</td>
</tr>
<tr>
<td>Dist: 9 1 3 0</td>
<td>Dist: 0 4 4 5</td>
<td>Dist: 0 6 3 4</td>
<td>Dist: 4 2 3 4</td>
</tr>
<tr>
<td>PH: 14 (9 2 3 0)</td>
<td>PH: 15 (0 8 7 0)</td>
<td>PH: 3 (0 0 0 3)</td>
<td>PH: 8 (1 0 0 7)</td>
</tr>
<tr>
<td>PD: 5</td>
<td>PD: 3</td>
<td>PD: 3</td>
<td>PD: 1</td>
</tr>
<tr>
<td>LT: 3</td>
<td>LT: 5</td>
<td>LT: 8</td>
<td>LT: 9</td>
</tr>
</tbody>
</table>

--- #2 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQT432</td>
<td>♠ -</td>
<td>♠ J85</td>
<td>♠ T96</td>
</tr>
<tr>
<td>♥ 985</td>
<td>♥ AKQJ</td>
<td>♥ T32</td>
<td>♥ 764</td>
</tr>
<tr>
<td>♦ AQ</td>
<td>♦ J863</td>
<td>♦ T742</td>
<td>♦ K95</td>
</tr>
<tr>
<td>♠ 8</td>
<td>♠ A9655</td>
<td>♠ 732</td>
<td>♠ KQT4</td>
</tr>
<tr>
<td>Dist: 7 3 2 1</td>
<td>Dist: 0 4 4 5</td>
<td>Dist: 3 3 4 3</td>
<td>Dist: 3 3 3 4</td>
</tr>
<tr>
<td>PH: 15 (9 0 6 0)</td>
<td>PH: 16 (0 1 0 5)</td>
<td>PH: 1 (0 0 0 0)</td>
<td>PH: 8 (0 0 3 5)</td>
</tr>
<tr>
<td>PD: 3</td>
<td>PD: 3</td>
<td>PD: 0</td>
<td>PD: 0</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 5</td>
<td>LT: 12</td>
<td>LT: 9</td>
</tr>
</tbody>
</table>

--- #3 - Intento 6 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQ9765</td>
<td>♠ -</td>
<td>♠ J3</td>
<td>♠ 42</td>
</tr>
<tr>
<td>♥ J9</td>
<td>♥ AK632</td>
<td>♥ 4</td>
<td>♥ Q1B75</td>
</tr>
<tr>
<td>♦ K</td>
<td>♦ A875</td>
<td>♦ QJT62</td>
<td>♦ 943</td>
</tr>
<tr>
<td>♦ AQ</td>
<td>♦ J979</td>
<td>♦ K164</td>
<td>♦ 852</td>
</tr>
<tr>
<td>Dist: 8 2 1 2</td>
<td>Dist: 0 5 4 4</td>
<td>Dist: 3 1 5 4</td>
<td>Dist: 2 5 3 3</td>
</tr>
<tr>
<td>PH: 19 (9 1 3 6)</td>
<td>PH: 12 (0 7 4 1)</td>
<td>PH: 7 (1 0 3 3)</td>
<td>PH: 2 (0 2 0 0)</td>
</tr>
<tr>
<td>PD: 4</td>
<td>PD: 3</td>
<td>PD: 2</td>
<td>PD: 1</td>
</tr>
<tr>
<td>LT: 4</td>
<td>LT: 6</td>
<td>LT: 8</td>
<td>LT: 10</td>
</tr>
</tbody>
</table>

--- #4 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ K865432</td>
<td>♠ -</td>
<td>♠ AQJT7</td>
<td>♠ 9</td>
</tr>
<tr>
<td>♥ AQJT</td>
<td>♥ 9874</td>
<td>♥ K3</td>
<td>♥ 652</td>
</tr>
<tr>
<td>♠ J</td>
<td>♠ K9683</td>
<td>♠ QT</td>
<td>♠ A7542</td>
</tr>
<tr>
<td>♠ Q</td>
<td>♠ AKJ3</td>
<td>♠ T982</td>
<td>♠ 7654</td>
</tr>
<tr>
<td>Dist: 7 4 1 1</td>
<td>Dist: 0 4 5 4</td>
<td>Dist: 5 2 2 4</td>
<td>Dist: 1 3 5 4</td>
</tr>
<tr>
<td>PH: 13 (3 7 1 2)</td>
<td>PH: 11 (0 0 3 8)</td>
<td>PH: 12 (7 3 2 0)</td>
<td>PH: 4 (0 0 4 0)</td>
</tr>
<tr>
<td>PD: 4</td>
<td>PD: 3</td>
<td>PD: 2</td>
<td>PD: 2</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 6</td>
<td>LT: 7</td>
<td>LT: 9</td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1.34
Exitos: 100
Sin mano: 0
Desviación std: 0.70
Tiempo (ms): 50

*** GAT ***
Corridas: 100
Promedio Intentos: 14357.67
Exitos: 90
Sin mano: 10 (sobre 30000 intentos por corrida)
Desviación std: 9075.65
Tiempo (ms): 72410
10. Ejemplo: Apertura de obstrucción

Descripción
Jugador con palo bueno de 7 cartas y poco PH

Restricciones
NORTE: Patrón 7,x,x,x & PH: 6-10
ESTE: Dist. libre & PH libre
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ Q865432</td>
<td>♠ J</td>
<td>♠ -</td>
<td>♠ AK97</td>
</tr>
<tr>
<td>♥ A</td>
<td>♥ 853</td>
<td>♥ QT976</td>
<td>♥ K34</td>
</tr>
<tr>
<td>♦ Q9874</td>
<td>♦ J532</td>
<td>♦ T</td>
<td>♦ AK6</td>
</tr>
<tr>
<td>♦ -</td>
<td>♦ K1982</td>
<td>♦ AQ7643</td>
<td>♦ 5</td>
</tr>
<tr>
<td>Dist: 7150</td>
<td>Dist: 1345</td>
<td>Dist: 0517</td>
<td>Dist: 5431</td>
</tr>
<tr>
<td>PH: 8 (2420)</td>
<td>PH: 6 (1014)</td>
<td>PH: 8 (0206)</td>
<td>PH: 18 (7470)</td>
</tr>
<tr>
<td>PD: 5</td>
<td>PD: 2</td>
<td>PD: 5</td>
<td>PD: 2</td>
</tr>
<tr>
<td>LT: 4</td>
<td>LT: 9</td>
<td>LT: 4</td>
<td>LT: 5</td>
</tr>
</tbody>
</table>

--- #2 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ QX</td>
<td>♠ 74</td>
<td>♠ J196532</td>
<td>♠ A8</td>
</tr>
<tr>
<td>♥ A976542</td>
<td>♥ K13</td>
<td>♥ QT8</td>
<td>♥ -</td>
</tr>
<tr>
<td>♦ 5</td>
<td>♦ QT86</td>
<td>♦ 93</td>
<td>♦ AK3742</td>
</tr>
<tr>
<td>♣ 632</td>
<td>♣ AK108</td>
<td>♣ 9</td>
<td>♣ KJ354</td>
</tr>
<tr>
<td>Dist: 2713</td>
<td>Dist: 2344</td>
<td>Dist: 7321</td>
<td>Dist: 2065</td>
</tr>
<tr>
<td>PH: 9 (5400)</td>
<td>PH: 12 (0426)</td>
<td>PH: 3 (1200)</td>
<td>PH: 16 (4084)</td>
</tr>
<tr>
<td>PD: 3</td>
<td>PD: 1</td>
<td>PD: 3</td>
<td>PD: 4</td>
</tr>
<tr>
<td>LT: 7</td>
<td>LT: 7</td>
<td>LT: 8</td>
<td>LT: 4</td>
</tr>
</tbody>
</table>

--- #3 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ J187542</td>
<td>♠ AK93</td>
<td>♠ -</td>
<td>♠ Q6</td>
</tr>
<tr>
<td>♥ AQ17</td>
<td>♥ K8</td>
<td>♥ 9653</td>
<td>♥ T42</td>
</tr>
<tr>
<td>♦ 52</td>
<td>♦ AKQ386</td>
<td>♦ T</td>
<td>♦ 9743</td>
</tr>
<tr>
<td>♣ 5</td>
<td>♣ AK18976</td>
<td>♣ 9</td>
<td>♣ Q343</td>
</tr>
<tr>
<td>Dist: 7420</td>
<td>Dist: 4261</td>
<td>Dist: 0418</td>
<td>Dist: 2344</td>
</tr>
<tr>
<td>PD: 4</td>
<td>PD: 3</td>
<td>PD: 5</td>
<td>PD: 1</td>
</tr>
<tr>
<td>LT: 6</td>
<td>LT: 3</td>
<td>LT: 5</td>
<td>LT: 10</td>
</tr>
</tbody>
</table>

--- #4 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ -</td>
<td>♠ AQ13</td>
<td>♠ 964</td>
<td>♠ K38752</td>
</tr>
<tr>
<td>♥ K</td>
<td>♥ 983</td>
<td>♥ AQ2</td>
<td>♥ T7654</td>
</tr>
<tr>
<td>♦ 87632</td>
<td>♦ 9</td>
<td>♦ AKQ34</td>
<td>♦ T5</td>
</tr>
<tr>
<td>♣ AQ9873</td>
<td>♣ K652</td>
<td>♣ T4</td>
<td>♣ -</td>
</tr>
<tr>
<td>Dist: 0157</td>
<td>Dist: 4414</td>
<td>Dist: 3352</td>
<td>Dist: 6520</td>
</tr>
<tr>
<td>PH: 10 (0307)</td>
<td>PH: 10 (6103)</td>
<td>PH: 16 (0610)</td>
<td>PH: 4 (4000)</td>
</tr>
<tr>
<td>PD: 5</td>
<td>PD: 2</td>
<td>PD: 1</td>
<td>PD: 4</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 7</td>
<td>LT: 6</td>
<td>LT: 7</td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 1000
Promedio Intentos: 1.40
Exitos: 1000
Sin mano: 0
Desviación std: 0.76
Tiempo (ms): 275

*** GAT ***
Corridas: 1000
Promedio Intentos: 22.81
Exitos: 1000
Sin mano: 0
Desviación std: 23.11
Tiempo (ms): 970

Pag. 84
11. Ejemplo: Única apertura artificial de Mayor Quinto (2 tréboles)

Descripción
Mano balanceada con PH >= 23

Restricciones
NORTE: Dist. balanceada & PH: 23-37
ESTE: Dist. libre & PH libre
SUR: Dist. libre & PH libre
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>A QO</td>
<td>A 98765432</td>
<td>A -</td>
<td>A KT</td>
</tr>
<tr>
<td>♥ AQ152</td>
<td>♥ -</td>
<td>♥ K198643</td>
<td>♥ 7</td>
</tr>
<tr>
<td>♦ A4</td>
<td>♦ J165</td>
<td>♦ 932</td>
<td>♦ KQ87</td>
</tr>
<tr>
<td>♠ AKQ</td>
<td>♠ 7</td>
<td>♠ J82</td>
<td>♠ T96543</td>
</tr>
<tr>
<td>Dist: 3 5 2 3</td>
<td>Dist: 8 0 4 1</td>
<td>Dist: 0 7 3 3</td>
<td>Dist: 2 1 4 6</td>
</tr>
<tr>
<td>PH: 22 (7 7 4 9)</td>
<td>PH: 1 (0 0 1 0)</td>
<td>PH: 4 (0 3 0 1)</td>
<td>PH: 8 (3 0 5 0)</td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 5</td>
<td>PD: 3</td>
<td>PD: 3</td>
</tr>
<tr>
<td>LT: 3</td>
<td>LT: 7</td>
<td>LT: 8</td>
<td>LT: 6</td>
</tr>
</tbody>
</table>

--- #2 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 128</td>
<td>A 975</td>
<td>A 643</td>
<td>A KQT2</td>
</tr>
<tr>
<td>♥ AKJ</td>
<td>♥ 97652</td>
<td>♥ T4</td>
<td>♥ Q83</td>
</tr>
<tr>
<td>♦ A543</td>
<td>♦ KQ872</td>
<td>♦ J196</td>
<td>♦ -</td>
</tr>
<tr>
<td>♠ AKQ</td>
<td>♠ -</td>
<td>♠ 9863</td>
<td>♠ J17542</td>
</tr>
<tr>
<td>Dist: 3 3 4 3</td>
<td>Dist: 3 5 5 0</td>
<td>Dist: 3 2 4 4</td>
<td>Dist: 4 3 0 6</td>
</tr>
<tr>
<td>PH: 26 (5 8 4 9)</td>
<td>PH: 5 (0 0 5 0)</td>
<td>PH: 1 (0 0 1 0)</td>
<td>PH: 8 (5 2 0 1)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 3</td>
<td>PD: 1</td>
<td>PD: 3</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 7</td>
<td>LT: 11</td>
<td>LT: 6</td>
</tr>
</tbody>
</table>

--- #3 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>A AKJ</td>
<td>A T9532</td>
<td>A 874</td>
<td>A 66</td>
</tr>
<tr>
<td>♥ AKJ</td>
<td>♥ Q</td>
<td>♥ 32</td>
<td>♥ T987654</td>
</tr>
<tr>
<td>♦ AKQ</td>
<td>♦ KJ7642</td>
<td>♦ 3</td>
<td>♦ 9875</td>
</tr>
<tr>
<td>♠ AKQ7</td>
<td>♠ 8</td>
<td>♠ 965432</td>
<td>♠ -</td>
</tr>
<tr>
<td>Dist: 3 3 2 5</td>
<td>Dist: 5 1 6 1</td>
<td>Dist: 3 2 1 7</td>
<td>Dist: 2 7 4 0</td>
</tr>
<tr>
<td>PH: 31 (8 8 6 9)</td>
<td>PH: 6 (0 2 4 0)</td>
<td>PH: 1 (0 0 0 1)</td>
<td>PH: 2 (2 0 0 0)</td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 4</td>
<td>PD: 3</td>
<td>PD: 4</td>
</tr>
<tr>
<td>LT: 3</td>
<td>LT: 7</td>
<td>LT: 9</td>
<td>LT: 8</td>
</tr>
</tbody>
</table>

--- #4 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>A AKJT</td>
<td>A 842</td>
<td>A 53</td>
<td>A 8976</td>
</tr>
<tr>
<td>♥ KQ8</td>
<td>♥ J32</td>
<td>♥ A9854</td>
<td>♥ T7</td>
</tr>
<tr>
<td>♦ AQ</td>
<td>♦ 9532</td>
<td>♦ T4</td>
<td>♦ 8976</td>
</tr>
<tr>
<td>♠ AKQ</td>
<td>♠ T72</td>
<td>♠ 8654</td>
<td>♠ 93</td>
</tr>
<tr>
<td>Dist: 4 3 3 3</td>
<td>Dist: 3 3 4 3</td>
<td>Dist: 2 5 2 4</td>
<td>Dist: 4 2 4 3</td>
</tr>
<tr>
<td>PH: 29 (8 5 7 9)</td>
<td>PH: 1 (0 1 0 0)</td>
<td>PH: 4 (0 4 0 0)</td>
<td>PH: 6 (2 0 3 1)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 2</td>
<td>PD: 1</td>
</tr>
<tr>
<td>LT: 3</td>
<td>LT: 12</td>
<td>LT: 9</td>
<td>LT: 9</td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1.01
Exitos: 100
Sin mano: 0
Desviación std: 0.10
Tiempo (ms): 35

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 105610
12. Ejemplo: Convención Stayman

Descripción
Mano balanceada para Norte. Para Sur 8 o más PH y 4 o más cartas en los palaivos altos

Restricciones

NORTE: Dist. balanceada & PH: 15-18
ESTE: Dist. libre & PH libre
SUR: P:4-5 & C:4-5 & D:5r & T:5r & PH: 8-10
OESTE: Dist. libre & PH Libre

Resultados

--- #1 - Intento 1 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ A4</td>
<td>♠ JT32</td>
<td>♠ J9875</td>
<td>♠ 6</td>
</tr>
<tr>
<td>♥ K7</td>
<td>♥ -</td>
<td>♥ Q9764</td>
<td>♥ A9853</td>
</tr>
<tr>
<td>♦ 864</td>
<td>♦ 975</td>
<td>♦ J</td>
<td>♦ AKQ3</td>
</tr>
<tr>
<td>♠ KQ93</td>
<td>♠ JT7652</td>
<td>♠ A8</td>
<td>♠ 4</td>
</tr>
</tbody>
</table>

Dist: 3 3 3 4 Dist: 4 0 3 6 Dist: 5 5 1 2 Dist: 1 5 6 1
PH: 15 (7 3 0 5) PH: 3 (2 0 0 1) PH: 8 (1 2 1 4) PH: 14 (0 5 9 0)
PD: 0 PD: 3 PD: 3 PD: 4
LT: 7 LT: 8 LT: 7 LT: 4

--- #2 - Intento 2 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ K96</td>
<td>♠ JT52</td>
<td>♠ AQ843</td>
<td>♠ 7</td>
</tr>
<tr>
<td>♥ 76532</td>
<td>♥ J</td>
<td>♥ Q1984</td>
<td>♥ AK</td>
</tr>
<tr>
<td>♦ K7</td>
<td>♦ T8</td>
<td>♦ J63</td>
<td>♦ AKQ54</td>
</tr>
<tr>
<td>♠ AKQ</td>
<td>♠ T98652</td>
<td>♠ -</td>
<td>♠ JT43</td>
</tr>
</tbody>
</table>

Dist: 3 5 2 3 Dist: 4 1 2 6 Dist: 5 5 3 0 Dist: 1 2 6 4
PH: 15 (3 0 3 9) PH: 2 (1 1 0 0) PH: 9 (6 2 1 0) PH: 14 (0 7 6 1)
PD: 1 PD: 3 PD: 3 PD: 3
LT: 6 LT: 9 LT: 6 LT: 5

--- #3 - Intento 3 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AK154</td>
<td>♠ -</td>
<td>♠ Q8763</td>
<td>♠ T92</td>
</tr>
<tr>
<td>♥ 383</td>
<td>♥ K5</td>
<td>♥ AQ87</td>
<td>♥ KT64</td>
</tr>
<tr>
<td>♦ K2</td>
<td>♦ K987</td>
<td>♦ 63</td>
<td>♦ AT4</td>
</tr>
<tr>
<td>♠ AQ9</td>
<td>♠ K17654</td>
<td>♠ T</td>
<td>♠ 832</td>
</tr>
</tbody>
</table>

Dist: 5 3 2 3 Dist: 0 2 5 6 Dist: 5 4 3 1 Dist: 3 4 3 3
PH: 17 (8 1 2 6) PH: 11 (0 3 4 4) PH: 8 (2 6 0 0) PH: 4 (0 0 4 0)
PD: 1 PD: 4 PD: 2 PD: 0
LT: 7 LT: 5 LT: 7 LT: 11

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 5.51
Exitos: 100
Sin mano: 0
Desviación std: 5.17
Tiempo (ms): 100

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 116440

GAT no generó manos
13. Ejemplo

Descripción

Tres jugadores con nro. fijo de PH (un jugador 12 puntos, otro 0, otro 2) y el otro con muchos piques (6 ó más).

Restricciones

NORTE: Dist. libre & PH: 12-12
ESTE: Dist. libre & PH: 0-0
SUR: Dist. libre & PH: 2-2
OESTE: P:6-13 & C:isr & D:isr & T:isr & PH libre

Resultados

--- #1 - Intento 30 (smart) ---
== Norte ==   == Este ==   == Sur ==   == Oeste ==
♠ -            ♠ 43         ♥ T7652      ♠ AKJQ98
♥ QT9432      ♥ 875        ♥ J6          ♥ AK
♦ AKJ         ♦ T82        ♦ 9734        ♦ Q63
♠ Q753        ♠ 98642      ♠ JT          ♠ AK
Dist: 0 6 3 4  Dist: 2 3 3 5  Dist: 6 2 4 2  Dist: 6 2 3 2
PH: 12 (0 2 8 2)  PH: 0 (0 0 0 0)  PH: 2 (0 1 0 1)  PH: 26 (10 7 2 7)
PD: 3           PD: 1         PD: 2          PD: 2
LT: 5          LT: 11        LT: 10         LT: 2
--- #2 - Intento 355 (smart) ---
== Norte ==   == Este ==   == Sur ==   == Oeste ==
♣ -            ♣ 8           ♣ J6543       ♣ AKJ9762
♦ K9872        ♦ T6543      ♦ J            ♦ AQ
♥ K3T5         ♥ 9874        ♥ 632         ♥ AQ
♠ A173         ♠ T92         ♠ 8654        ♠ KQ
Dist: 0 5 4 4  Dist: 1 5 4 3  Dist: 5 1 3 4  Dist: 7 2 2 2
PH: 12 (0 3 4 5)  PH: 0 (0 0 0 0)  PH: 2 (1 1 0 0)  PH: 26 (9 6 6 5)
PD: 3           PD: 2         PD: 2          PD: 3
LT: 6          LT: 10        LT: 10         LT: 7
--- #3 - Intento 5112 (smart) ---
== Norte ==   == Este ==   == Sur ==   == Oeste ==
♠ -            ♠ T976        ♠ J83          ♠ AKJ542
♥ AKJ952      ♥ T743        ♥ 986          ♥ 6
♦ Q987432      ♦ T           ♦ AKJ5         ♥ AKJ
♣ -            ♣ 9652        ♣ JT8743        ♣ AKQ
Dist: 0 6 7 0  Dist: 4 4 1 4  Dist: 3 3 1 6  Dist: 6 0 4 3
PH: 12 (0 1 0 2 0)  PH: 0 (0 0 0 0)  PH: 2 (1 0 0 1)  PH: 26 (9 0 8 9)
PD: 6           PD: 2         PD: 2          PD: 3
LT: 2          LT: 10        LT: 10         LT: 7
--- #4 - Intento 47 (smart) ---
== Norte ==   == Este ==   == Sur ==   == Oeste ==
♣ -            ♣ JT64        ♣ 8752        ♣ AKJ765
♥ QTR42       ♥ 753          ♥ 96          ♠ AKJ
♠ AKJ6         ♠ 94          ♠ JT8732        ♠ 5
Dist: 0 4 5 4  Dist: 4 4 3 2  Dist: 3 2 2 6  Dist: 6 3 3 1
PH: 12 (0 1 2 9)  PH: 0 (0 0 0 0)  PH: 2 (1 0 0 1)  PH: 26 (9 9 8 0)
PD: 3           PD: 1         PD: 2          PD: 2
LT: 5          LT: 11        LT: 10         LT: 2

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 664.67
Exitos: 100
Sin mano: 0
Desviación std: 588.71
Tiempo $(ms)$: 6905

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)

GAT no generó manos
14. Ejemplo

Descripción
Dos jugadores con manos balanceadas y los otros dos con manos altamente desbalanceadas

Restricciones
NORTE: Dist. balanceada & PH libre
ESTE: Dist. balanceada & PH libre
SUR: Patrón 0.x.x.x & PH libre
OESTE: Patrón 0.x.x.x & PH libre

Resultados
--- #1 - Intento 7 (Smart) ---
<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ 63</td>
<td>♠ Q65</td>
<td>♠ AKJ974</td>
<td>♠ -</td>
</tr>
<tr>
<td>♦ T82</td>
<td>♦ KQ9</td>
<td>♦ 4</td>
<td>♦ A7653</td>
</tr>
<tr>
<td>♢ J7</td>
<td>♢ 632</td>
<td>♢ AKQ85</td>
<td>♢ 94</td>
</tr>
<tr>
<td>Dist: 2353</td>
<td>Dist: 4423</td>
<td>Dist: 7105</td>
<td>Dist: 0562</td>
</tr>
<tr>
<td>PH: 1 (0001)</td>
<td>PH: 10 (2620)</td>
<td>PH: 17 (8009)</td>
<td>PH: 12 (0480)</td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 1</td>
<td>PD: 5</td>
<td>PD: 4</td>
</tr>
<tr>
<td>LT: 11</td>
<td>LT: 8</td>
<td>LT: 2</td>
<td>LT: 5</td>
</tr>
</tbody>
</table>

--- #2 - Intento 3 (Smart) ---
<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ 9872</td>
<td>♠ A543</td>
<td>♠ -</td>
<td>♠ KQJT6</td>
</tr>
<tr>
<td>♦ Q2</td>
<td>♦ J5</td>
<td>♦ AKT987543</td>
<td>♦ 7632</td>
</tr>
<tr>
<td>♠ K19</td>
<td>♠ AT85</td>
<td>♠ Q4</td>
<td>♠ QT72</td>
</tr>
<tr>
<td>Dist: 4234</td>
<td>Dist: 4243</td>
<td>Dist: 0922</td>
<td>Dist: 5044</td>
</tr>
<tr>
<td>PH: 14 (0248)</td>
<td>PH: 9 (4140)</td>
<td>PH: 9 (0720)</td>
<td>PH: 8 (6002)</td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 1</td>
<td>PD: 5</td>
<td>PD: 3</td>
</tr>
<tr>
<td>LT: 8</td>
<td>LT: 9</td>
<td>LT: 5</td>
<td>LT: 6</td>
</tr>
</tbody>
</table>

--- #3 - Intento 7 (Smart) ---
<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ K32</td>
<td>♠ QT96</td>
<td>♠ A3J5</td>
<td>♠ 8</td>
</tr>
<tr>
<td>♦ K75</td>
<td>♦ 43</td>
<td>♦ -</td>
<td>♦ AQJT9862</td>
</tr>
<tr>
<td>♦ Q62</td>
<td>♦ A83</td>
<td>♦ T75</td>
<td>♦ K394</td>
</tr>
<tr>
<td>♦ J7</td>
<td>♦ 9542</td>
<td>♦ AKQ863</td>
<td>♦ -</td>
</tr>
<tr>
<td>Dist: 4333</td>
<td>Dist: 4234</td>
<td>Dist: 4036</td>
<td>Dist: 1840</td>
</tr>
<tr>
<td>PH: 9 (3321)</td>
<td>PH: 6 (2040)</td>
<td>PH: 14 (5009)</td>
<td>PH: 11 (0740)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 1</td>
<td>PD: 3</td>
<td>PD: 5</td>
</tr>
<tr>
<td>LT: 9</td>
<td>LT: 9</td>
<td>LT: 5</td>
<td>LT: 4</td>
</tr>
</tbody>
</table>

--- #4 - Intento 10 (Smart) ---
<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ T52</td>
<td>♠ A984</td>
<td>♠ KQJT</td>
<td>♠ 7</td>
</tr>
<tr>
<td>♦ 52</td>
<td>♦ T8</td>
<td>♦ KQ3</td>
<td>♦ A9764</td>
</tr>
<tr>
<td>♦ KQJT</td>
<td>♦ A92</td>
<td>♦ -</td>
<td>♦ 876543</td>
</tr>
<tr>
<td>♦ K362</td>
<td>♦ A84</td>
<td>♦ Q9753</td>
<td>♦ T</td>
</tr>
<tr>
<td>Dist: 3244</td>
<td>Dist: 5233</td>
<td>Dist: 5305</td>
<td>Dist: 0661</td>
</tr>
<tr>
<td>PH: 10 (0064)</td>
<td>PH: 13 (5044)</td>
<td>PH: 12 (5502)</td>
<td>PH: 5 (0500)</td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 1</td>
<td>PD: 3</td>
<td>PD: 5</td>
</tr>
<tr>
<td>LT: 8</td>
<td>LT: 8</td>
<td>LT: 4</td>
<td>LT: 6</td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 6.40
Exitos: 100
Sin mano: 0
Desviacion std: 6.31
Tiempo (ms): 150

*** GAT ***
Corridas: 100
Promedio Intentos: 180.69
Exitos: 100
Sin mano: 0
Desviacion std: 165.53
Tiempo (ms): 855
15. Ejemplo

Descripción
Entre Norte y Sur todos los PH (N con 37, S con 3)

Restricciones
- **NORTE**: Patrón 3.x.x.x & PH: 37-37  
- **ESTE**: Dist. libre & PH libre  
- **SUR**: Dist. libre & PH: 3-3  
- **OESTE**: Dist. libre & PH libre

Resultados
--- #1 - Intento 110 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQ</td>
<td>♠ T7643</td>
<td>♠ -</td>
<td>♠ 9852</td>
</tr>
<tr>
<td>♥ AKQ</td>
<td>♥ 2</td>
<td>♥ J974</td>
<td>♥ T8653</td>
</tr>
<tr>
<td>♦ AKQ</td>
<td>♦ 65</td>
<td>♦ J97432</td>
<td>♦ T</td>
</tr>
<tr>
<td>♣ AKQ</td>
<td>♣ 9742</td>
<td>♣ J13</td>
<td>♣ 865</td>
</tr>
</tbody>
</table>

Dist: 4 3 3 3  Dist: 5 1 3 4  Dist: 0 4 6 3  Dist: 4 5 1 3
PH: 37 (10 9 9 9) PH: 0 (0 0 0 0) PH: 3 (0 1 1 1) PH: 0 (0 0 0 0)
Pd: 0  Pd: 2  Pd: 3  Pd: 2
Lt: 0  Lt: 10  Lt: 9  Lt: 10

--- #2 - Intento 120 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQ</td>
<td>♠ 5</td>
<td>♠ 796432</td>
<td>♠ 7</td>
</tr>
<tr>
<td>♥ AKQ</td>
<td>♥ T7432</td>
<td>♥ J</td>
<td>♥ 9865</td>
</tr>
<tr>
<td>♦ AKQ</td>
<td>♦ 32</td>
<td>♦ J765</td>
<td>♦ 9874</td>
</tr>
<tr>
<td>♣ AKQ</td>
<td>♣ 97632</td>
<td>♣ J</td>
<td>♣ T854</td>
</tr>
</tbody>
</table>

Dist: 4 3 3 3  Dist: 1 5 2 5  Dist: 7 1 4 1  Dist: 1 4 4 4
PH: 37 (10 9 9 9) PH: 0 (0 0 0 0) PH: 3 (0 1 1 1) PH: 0 (0 0 0 0)
Pd: 0  Pd: 3  Pd: 4  Pd: 2
Lt: 0  Lt: 9  Lt: 8  Lt: 10

--- #3 - Intento 129 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQ</td>
<td>♠ 963</td>
<td>♠ -</td>
<td>♠ T87542</td>
</tr>
<tr>
<td>♥ AKQ</td>
<td>♥ T</td>
<td>♥ J8643</td>
<td>♥ 9752</td>
</tr>
<tr>
<td>♦ AKQ</td>
<td>♦ 8542</td>
<td>♦ J796</td>
<td>♦ 73</td>
</tr>
<tr>
<td>♣ AKQ</td>
<td>♣ 9862</td>
<td>♣ J743</td>
<td>♣ 5</td>
</tr>
</tbody>
</table>

Dist: 4 3 3 3  Dist: 3 1 4 5  Dist: 0 5 4 4  Dist: 6 4 2 1
PH: 37 (10 9 9 9) PH: 0 (0 0 0 0) PH: 3 (0 1 1 1) PH: 0 (0 0 0 0)
Pd: 0  Pd: 2  Pd: 3  Pd: 3
Lt: 0  Lt: 10  Lt: 9  Lt: 9

--- #4 - Intento 274 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKQ</td>
<td>♠ 53</td>
<td>♠ J7942</td>
<td>♠ 86</td>
</tr>
<tr>
<td>♥ AKQ</td>
<td>♥ 9532</td>
<td>♥ J87</td>
<td>♥ T64</td>
</tr>
<tr>
<td>♦ AKQ</td>
<td>♦ 97652</td>
<td>♦ J</td>
<td>♦ T843</td>
</tr>
<tr>
<td>♣ AKQ</td>
<td>♣ 82</td>
<td>♣ T73</td>
<td>♣ 9654</td>
</tr>
</tbody>
</table>

Dist: 3 3 3 4  Dist: 2 4 5 2  Dist: 6 3 1 3  Dist: 2 3 4 4
PH: 37 (9 9 9 10) PH: 0 (0 0 0 0) PH: 3 (1 1 1 0) PH: 0 (0 0 0 0)
Pd: 0  Pd: 2  Pd: 2  Pd: 1
Lt: 0  Lt: 10  Lt: 10  Lt: 11

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 119.05
Exitos: 100
Sin mano: 0
Desviación std: 137.92
Tiempo (ms): 4460

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)

GAT no generó manos
16. Ejemplo

Descripción
Entre Norte y Sur todos los PH (N con 20, S con 20)

Restricciones
**NORTE**: Dist. libre & PH: 20-20  
**ESTE**: Dist. libre & PH libre  
**SUR**: Dist. libre & PH: 20-20  
**OESTE**: Dist. libre & PH libre

Resultados

<table>
<thead>
<tr>
<th>#1 - Intento 85 (Smart)</th>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q65</td>
<td>T984</td>
<td>AK7</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>KQ</td>
<td>T743</td>
<td>A38</td>
<td>9652</td>
<td></td>
</tr>
<tr>
<td>KQ176</td>
<td>T52</td>
<td>A983</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>AQ2</td>
<td>76</td>
<td>K8</td>
<td>T95432</td>
<td></td>
</tr>
<tr>
<td>Dist: 3253</td>
<td>Dist: 4432</td>
<td>Dist: 2416</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH: 20 (2567)</td>
<td>PH: 0 (0000)</td>
<td>PH: 20 (8543)</td>
<td>PH: 0 (0000)</td>
<td></td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 1</td>
<td>PD: 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#2 - Intento 11 (Smart)</th>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>KQ12</td>
<td>984</td>
<td>AT</td>
<td>7653</td>
<td></td>
</tr>
<tr>
<td>KQ</td>
<td>875</td>
<td>AJT92</td>
<td>643</td>
<td></td>
</tr>
<tr>
<td>Q9</td>
<td>742</td>
<td>AK15</td>
<td>T863</td>
<td></td>
</tr>
<tr>
<td>AQ938</td>
<td>7632</td>
<td>K4</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>Dist: 4225</td>
<td>Dist: 3334</td>
<td>Dist: 2542</td>
<td>Dist: 4342</td>
<td></td>
</tr>
<tr>
<td>PH: 20 (6527)</td>
<td>PH: 0 (0000)</td>
<td>PH: 20 (4583)</td>
<td>PH: 0 (0000)</td>
<td></td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 0</td>
<td>PD: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 12</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#3 - Intento 92 (Smart)</th>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>K6</td>
<td>T87542</td>
<td>AQ3</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>AJ12</td>
<td>7654</td>
<td>KQ</td>
<td>983</td>
<td></td>
</tr>
<tr>
<td>AJ87</td>
<td>92</td>
<td>KQ</td>
<td>T8543</td>
<td></td>
</tr>
<tr>
<td>AQ2</td>
<td>T</td>
<td>KB5432</td>
<td>976</td>
<td></td>
</tr>
<tr>
<td>Dist: 2443</td>
<td>Dist: 6421</td>
<td>Dist: 3226</td>
<td>Dist: 2353</td>
<td></td>
</tr>
<tr>
<td>PH: 20 (3557)</td>
<td>PH: 0 (0000)</td>
<td>PH: 20 (7553)</td>
<td>PH: 0 (0000)</td>
<td></td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 3</td>
<td>PD: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 6</td>
<td>LT: 9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#4 - Intento 166 (Smart)</th>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>A965</td>
<td>T2</td>
<td>KQ3</td>
<td>8743</td>
<td></td>
</tr>
<tr>
<td>AKJ</td>
<td>98532</td>
<td>Q</td>
<td>T764</td>
<td></td>
</tr>
<tr>
<td>Q6</td>
<td>9852</td>
<td>AKJ3</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>AQ64</td>
<td>K93</td>
<td>875</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dist: 4324</td>
<td>Dist: 2542</td>
<td>Dist: 3154</td>
<td>Dist: 4423</td>
<td></td>
</tr>
<tr>
<td>PH: 20 (4826)</td>
<td>PH: 0 (0000)</td>
<td>PH: 20 (6284)</td>
<td>PH: 0 (0000)</td>
<td></td>
</tr>
<tr>
<td>PD: 1</td>
<td>PD: 2</td>
<td>PD: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT: 6</td>
<td>LT: 10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***
Corridas: 100  
Promedio Intentos: 98.31  
Exitos: 100  
Sin manco: 0  
Desviación std: 103.73  
Tiempo (ms): 1099

*** GAT ***
Corridas: 100  
Promedio Intentos: 0  
Exitos: 0  
Sin manco: 100 (sobre 30000 intentos por corrida)

GAT no generó manos
17. Ejemplo

Descripción

Entre Norte y Sur todos los P11 (idem anterior pero especificando N con 20 ó más, S con 20 ó más)

Restricciones

NORTE: Dist. libre & PH: 20-37
ESTE: Dist. libre & PH libre
SUR: Dist. libre & PH 20-37
OESTE: Dist. libre & PH libre

Resultados

--- #1 - Intento 21 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AKJ</td>
<td>♠ 987632</td>
</tr>
<tr>
<td>♥ K3</td>
<td>♥ 96</td>
</tr>
<tr>
<td>♦ KQ42</td>
<td>♦ 7</td>
</tr>
<tr>
<td>♣ K32</td>
<td>♣ T875</td>
</tr>
</tbody>
</table>

Disc: 3 2 5 3
Dist: 6 2 1 4
PH: 20 (8 4 5 3) PH: 0 (0 0 0 0)
PD: 1
LT: 5
LT: 9

--- #2 - Intento 12 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ Q3</td>
<td>♠ 765</td>
</tr>
<tr>
<td>♥ Q5</td>
<td>♥ 73</td>
</tr>
<tr>
<td>♦ AKJ</td>
<td>♦ T7543</td>
</tr>
<tr>
<td>♣ KQ42</td>
<td>♣ 86</td>
</tr>
</tbody>
</table>

Disc: 3 2 3 5
Dist: 3 2 5 3
Dist: 4 3 2 4
PH: 20 (5 2 8 5) PH: 0 (0 0 0 0)
PD: 1
LT: 5
LT: 11

--- #3 - Intento 14 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AQJ</td>
<td>♠ 87532</td>
</tr>
<tr>
<td>♥ A4</td>
<td>♥ 83</td>
</tr>
<tr>
<td>♦ Q74</td>
<td>♦ T653</td>
</tr>
<tr>
<td>♣ AQJ86</td>
<td>♣ 74</td>
</tr>
</tbody>
</table>

Disc: 3 2 3 5
Dist: 5 2 4 2
Dist: 3 3 3 4
PH: 20 (7 4 2 7) PH: 0 (0 0 0 0)
PD: 1
LT: 5
LT: 10

--- #4 - Intento 12 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AT9</td>
<td>♠ 8764</td>
</tr>
<tr>
<td>♥ AQJ</td>
<td>♥ 74</td>
</tr>
<tr>
<td>♦ AJ53</td>
<td>♦ T974</td>
</tr>
<tr>
<td>♣ A9</td>
<td>♣ 87542</td>
</tr>
</tbody>
</table>

Disc: 3 3 5 2
Dist: 4 0 4 5
Dist: 4 3 2 4
PH: 20 (4 7 5 4) PH: 0 (0 0 0 0)
PD: 1
LT: 6
LT: 9

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 86.07
Exitos: 100
Sin mano: 0
Desviación std: 75.57
Tiempo (ms): 1020

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)

GAT no generó manos
18. Ejemplo

Descripción
Entre Norte y Sur un pequeño o gran slam con posible obstrucción de Este (N con 16 ó más PH, S con 16 ó más PH, E con un palo de 9)

Restricciones
NORTE: Dist. libre & PH: 16-37
ESTE: Patrón 9.x.x.x & PH libre
SUR: Dist. libre & PH 16-37
OESTE: Dist. libre & PH libre

Resultados
--- #1 - Intento 31 (Smart) ---
                    Norte       Este       Sur        Oeste
       ♠ A7632  ♠ Q198       ♠ K        ♠ 54
       ♥ AQJ   ♥ KT9765432  ♥ -        ♥ 8
       ♣ A3    ♣ -         ♣ KQJT65    ♣ 98742
       ♦ J6    ♦ -         ♦ AKQTJ43    ♦ T9852
Dist: 6 3 2 2  Dist: 4 9 0 0  Dist: 1 0 6 6  Dist: 2 1 5 5
PH: 16 (4 7 4 1) PH: 6 (3 3 0 0) PH: 18 (3 0 6 9) PH: 0 (0 0 0 0)
PD: 2          PD: 6        PD: 5        PD: 3
LT: 6          LT: 4        LT: 2        LT: 9

--- #2 - Intento 47 (Smart) ---
                    Norte       Este       Sur        Oeste
       ♠ A7    ♠ KQT986543  ♠ J2         ♠ -
       ♥ AT    ♥ -          ♥ KQ        ♥ J98765432
       ♣ A323   ♣ -          ♣ KQ654      ♣ T987
       ♦ KQ64   ♦ 9763       ♦ AJ72       ♦ -
Dist: 2 2 4 5  Dist: 9 9 0 4  Dist: 2 2 5 4  Dist: 0 9 4 0
PH: 18 (4 4 5 5) PH: 5 (5 0 0 0) PH: 16 (1 5 5 5) PH: 1 (0 1 0 0)
PD: 2          PD: 6        PD: 2        PD: 6
LT: 5          LT: 4        LT: 6        LT: 6

--- #3 - Intento 78 (Smart) ---
                    Norte       Este       Sur        Oeste
       ♠ KQJ943  ♠ T8      ♠ A7652    ♠ -
       ♥ Q      ♥ JT9865432  ♥ K         ♥ A7
       ♣ J5     ♣ 63         ♣ AK        ♣ T87542
       ♦ K6843  ♦ 98643     ♦ J1972      ♦ -
Dist: 6 1 3 3  Dist: 2 9 2 0  Dist: 5 1 2 5  Dist: 0 2 6 5
PH: 17 (6 2 3 6) PH: 1 (0 1 0 0) PH: 17 (4 3 7 3) PH: 5 (0 4 0 1)
PD: 2          PD: 5        PD: 3        PD: 4
LT: 5          LT: 7        LT: 5        LT: 7

--- #4 - Intento 1 (Smart) ---
                    Norte       Este       Sur        Oeste
       ♠ A     ♠ QT9875432  ♠ KJ6       ♠ -
       ♥ KB6   ♥ QT933      ♥ A8754     ♥ -
       ♣ KT964  ♣ -          ♣ AQ8       ♣ J7532
       ♦ AQ97  ♦ -          ♦ J          ♦ KT865432
Dist: 1 3 5 4  Dist: 9 4 0 0  Dist: 3 6 3 1  Dist: 0 0 5 8
PH: 16 (4 3 3 6) PH: 4 (2 2 0 0) PH: 16 (4 5 6 1) PH: 4 (0 0 1 3)
PD: 2          PD: 6        PD: 2        PD: 6
LT: 5          LT: 4        LT: 6        LT: 5

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 54.68
Exitos: 100
Sin mano: 0
Desviación std: 63.76
Tiempo (ms): 905

*** GAT ***
Corridas: 100
Promedio Intentos: 54.68
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)

GAT no generó manos
19. Ejemplo

Descripción
Norte muy desbalanceado y 0 PH, Sur muchos PH para pequeño o gran slam, y Este y Oeste con casi todos los corazones

Restricciones
NORTE: PD: 5-9 & PH: 0-0
ESTE: P:SR & C:5-13 & D:SR & T:SR & PH libre
SUR: Dist. libre & PH: 32-37

Resultados
--- #1 - Intento 3516 (Smart) ---
- Norte == = Este == = Sur == = Oeste ==
  ♠ T865 ♠ 432 ♠ AKQ ♠ 97
  ♥ 7 ♦ KQT6 ♥ A ♥ J985432
  ♦ 4 ♥ J6542 ♥ AKQ3 ♦ T98
  Dist: 4 0 9 0 Dist: 3 5 0 5 Dist: 4 1 3 5 Dist: 2 7 1 3
  PH: 0 (0 0 0 0) PH: 6 (0 5 0 1) PH: 32 (10 4 9 9) PH: 2 (0 1 1 0)
  PD: 6 PD: 3 PD: 2 PD: 3
  LT: 6 LT: 7 LT: 0 LT: 9
--- #2 - Intento 2601 (Smart) ---
- Norte == = Este == = Sur == = Oeste ==
  ♠ T9632 ♠ 854 ♠ AKQ ♠ 37
  ♥ 7 ♦ T7642 ♥ AK ♥ Q9853
  ♦ 4 ♥ T9864 ♥ AKQ3 ♦ 732
  Dist: 5 0 0 8 Dist: 3 5 4 1 Dist: 3 2 6 2 Dist: 2 6 3 2
  PH: 0 (0 0 0 0) PH: 0 (0 0 0 0) PH: 33 (9 7 10 7) PH: 7 (1 3 0 3)
  PD: 6 PD: 2 PD: 2 PD: 2
  LT: 6 LT: 10 LT: 0 LT: 9
--- #3 - Intento 16384 (Smart) ---
- Norte == = Este == = Sur == = Oeste ==
  ♠ T9865432 ♠ - ♠ AKQJ ♠ -
  ♥ 7 ♦ T7542 ♥ AQ ♥ KJ9863
  ♦ 3 ♦ J543 ♦ AKQ ♦ 98762
  Dist: 9 0 1 3 Dist: 0 5 4 4 Dist: 4 2 3 4 Dist: 0 6 5 2
  PH: 0 (0 0 0 0) PH: 1 (0 0 1 0) PH: 33 (10 6 9 8) PH: 6 (0 4 0 2)
  PD: 5 PD: 3 PD: 1 PD: 4
  LT: 7 LT: 9 LT: 2 LT: 7
--- #4 - Intento 8866 (Smart) ---
- Norte == = Este == = Sur == = Oeste ==
  ♠ - ♣ T8743 ♠ AKQJ2 ♠ 965
  ♥ 7 ♦ Q965 ♦ AK ♦ T87432
  ♦ 3 ♦ J62 ♦ AKQ ♦ 8
  Dist: 0 0 6 7 Dist: 5 5 3 0 Dist: 5 2 3 3 Dist: 3 6 1 3
  PH: 0 (0 0 0 0) PH: 4 (0 3 1 0) PH: 33 (10 7 9 7) PH: 3 (0 0 0 3)
  PD: 6 PD: 3 PD: 1 PD: 2
  LT: 6 LT: 8 LT: 1 LT: 9

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 9307.61
Exitos: 97
Sin mano: 3 (sobre 30000 intentos por corrida)
Desviación std: 8017.41
Tiempo (ms): 295640

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 151935

GAT no generó manos
20. Ejemplo

Descripción

Entre Norte y Este casi todos los piques (de 5 a 7), entre Sur y Oeste casi todos los corazones (de 5 a 7).

Norte y Sur con PH en todos los palos

Restricciones

NORTE: P:5-7 & C:5-7 & D:5 & T:5 & PH:1-10 & PHC:1-10 & PHD:1-10 & PHY:1-10
ESTE: P:5-7 & C:5-7 & D:5 & T:5 & PH:1-10 & PHC:1-10 & PHD:1-10 & PHY:1-10
OESTE: P:5 & C:5-7 & D:5 & T:5 & PH:1-10 & PHC:1-10 & PHD:1-10 & PHY:1-10

Resultados

--- #1 - Intento 2081 (Smart) ---

Norte == Este == Sur == Oeste ==

♣ K7952 ♠ Q8643 ♦ A3 ♣ T95432
♥ K ♦ J ♦ AQ786 ♦ T4
♣ AK ♦ T7654 ♦ Q3 ♣ 932
♣ AKQ ♦ T4 ♦ J65 ♣ 9873
Dist: 6 1 2 4 Dist: 5 1 5 2 Dist: 2 5 3 3 Dist: 0 6 3 4
PH: 22 (3 3 7 9) PH: 3 (2 1 0 4) PH: 13 (5 6 3 1) PH: 0 (0 0 0 0)
PD: 3 PD: 3 PD: 1 PD: 3
LT: 3 LT: 8 LT: 7 LT: 9

--- #2 - Intento 2362 (Smart) ---

Norte == Este == Sur == Oeste ==

♣ K7632 ♠ AQ9854 ♦ J ♣ J
♥ AKJ ♦ QA754 ♦ QT9542 ♣ 98632
♣ A ♦ KQ ♦ QT9542 ♣ 8763
♣ ATK ♦ KQJ ♦ J ♣ 9652
Dist: 5 3 1 4 Dist: 7 0 2 4 Dist: 1 5 6 1 Dist: 0 5 4 4
PH: 19 (3 8 4 4) PH: 16 (6 0 5 5) PH: 5 (1 2 1 1) PH: 0 (0 0 0 0)
PD: 2 PD: 4 PD: 4 PD: 3
LT: 5 LT: 3 LT: 7 LT: 9

--- #3 - Intento 3499 (Smart) ---

Norte == Este == Sur == Oeste ==

♣ Q87643 ♠ AJT952 ♦ K ♣ K
♥ K ♦ JT953 ♦ A986542
♣ AKJ ♦ 32 ♦ QT952 ♣ 864
♣ A ♦ 98765 ♦ KT952 ♣ KQ3
Dist: 6 1 3 3 Dist: 6 0 2 5 Dist: 1 5 5 2 Dist: 0 7 3 3
PH: 14 (2 3 8 1) PH: 5 (5 0 0 0) PH: 12 (3 3 2 4) PH: 9 (0 4 0 5)
PD: 7 PD: 4 PD: 3 PD: 3
LT: 7 LT: 7 LT: 6 LT: 6

--- #4 - Intento 8502 (Smart) ---

Norte == Este == Sur == Oeste ==

♣ AK9763 ♠ T8542 ♦ Q ♣ K
♥ K ♦ AQJ6542 ♦ T9873
♣ A ♦ T9875 ♦ KQ3 ♣ 362
♣ AKJ ♦ 84 ♦ Q6 ♣ T9753
Dist: 7 1 4 4 Dist: 5 0 6 2 Dist: 1 7 3 2 Dist: 0 5 3 5
PH: 23 (8 3 4 8) PH: 0 (0 0 0 0) PH: 16 (2 7 5 2) PH: 1 (0 0 1 0)
PD: 4 PD: 4 PD: 3 PD: 3
LT: 3 LT: 8 LT: 5 LT: 9

Comparación

*** Smart ***
Corrías: 100
Promedio Intentos: 2150.79
Exitos: 100
Sin mano: 0
Desviación std: 1943.03
Tiempo (ms): 28105

*** GAT ***
Corrías: 100
Promedio Intentos: 33055.08
Exitos: 60
Sin mano: 40 (sobre 3000 intentos por corrida)
Desviación std: 10234.95
Tiempo (ms): 106135

Pag. 94
21. Ejemplo

Descripción
Norte y Sur con 1 PH c/u. Este con mano de paso y todos los jugadores con distribución 5.3.3.2

Restricciones
NORTE: Patrón 5.3.3.2 & PH: 1-1
ESTE : Patrón 5.3.3.2 & PH: 0-12
SUR : Patrón 5.3.3.2 & PH: 1-1
OESTE: Patrón 5.3.3.2 & PH Libre

Resultados
--- #1 - Intento 111 (Smart) ---
  Norte: J8762, T7, 862, T63
  Este: 954, Q9542, KQ5, A5
  Sur: T3, 863, J39743, 972
  Oeste: AKQ, AKJ, AT, KQ384
  Dist: 5 233, Dist: 3 532, Dist: 2 353, Dist: 3 325
  PH: 1 (1 0 0 0), PH: 11 (0 2 5 4), PH: 1 (0 0 1 0), PH: 27 (9 8 4 6)
  PD: 1, PD: 1, PD: 1, PD: 1
  LT: 11, LT: 7, LT: 11, LT: 3

--- #2 - Intento 3 (Smart) ---
  Norte: J9872, T6, 732, D43
  Este: A65, Q94, Q1, KT872
  Sur: T43, 87532, 964, AQ9
  Oeste: KQ, AKJ, AKJ385
  Dist: 5 233, Dist: 3 325, Dist: 2 353, Dist: 3 325
  PH: 1 (1 0 0 0), PH: 11 (4 2 2 3), PH: 1 (0 0 0 1), PH: 27 (5 8 8 6)
  PD: 1, PD: 1, PD: 1, PD: 1
  LT: 11, LT: 8, LT: 11, LT: 4

--- #3 - Intento 143 (Smart) ---
  Norte: 36, 854, T53, 87642
  Este: KQ9, 93, Q9364, QT2
  Sur: T8732, 762, 872, AKS
  Oeste: A54, AKQT, AK
  Dist: 2 353, Dist: 2 353, Dist: 2 353, Dist: 3 253
  PH: 1 (1 0 0 0), PH: 10 (5 0 3 2), PH: 1 (0 0 0 1), PH: 28 (4 10 7 7)
  PD: 1, PD: 1, PD: 1, PD: 1
  LT: 11, LT: 7, LT: 11, LT: 3

--- #4 - Intento 281 (Smart) ---
  Norte: 86, T9542, T53, J73
  Este: KQT53, Q83, 87, Q82
  Sur: 942, 78, 362, T9654
  Oeste: A37, AKJ, AKQ94
  Dist: 2 353, Dist: 2 353, Dist: 3 235, Dist: 3 352
  PH: 1 (0 0 0 1), PH: 9 (5 2 0 2), PH: 1 (0 0 1 0), PH: 29 (5 8 9 7)
  PD: 1, PD: 1, PD: 1, PD: 1
  LT: 11, LT: 7, LT: 11, LT: 3

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 126.14
Exitos: 100
Sin mano: 0
Desviación std: 125.72
Tiempo (ms): 44445

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 151935

GAT no generó manos
22. Ejemplo

Descripción
Todos los jugadores con 3 ó 4 PD, Norte con 4 ases y mínimo PH en pique. Este con 4 reyes y mínimo PH en corazón, Sur con 4 damas y mínimo PH en diamante, Oeste con 4 jacks y mínimo PH en trébol.

Restricciones
ESTE: PD: 3-4 & PH:3:s & PHC:0-3 & PHD:0-3 & PHF:0-3

Resultados
--- #1 - Intento 100 (Smart) ---
== Norte ==    == Este ==    == Sur ==    == Oeste ==
* A2          * KT87543    * Q           * J96
* A9876543    * KT          * Q           * J2
* A3          * K2          * Q9765        * JT84
* A            * K8          * Q98652       * JT73
Dist: 2 8 2 1  Dist: 7 2 2 2  Dist: 1 1 5 6  Dist: 2 1 3 3
PH: 10 (4 4 4 4) PH: 12 (3 3 3 3) PH: 8 (2 2 2 2) PH: 4 (1 1 1 1)
PD: 4          PD: 3          PD: 4          PD: 3
LT: 4          LT: 5          LT: 6          LT: 9
--- #2 - Intento 5 (Smart) ---
== Norte ==    == Este ==    == Sur ==    == Oeste ==
* A5          * K97642      * QT3          * J8
* A8          * K6542       * QT93         * J7
* A876432     * K            * Q95          * JT
* A6          * K            * QT3          * J198753
Dist: 2 2 7 2  Dist: 6 5 1 1  Dist: 2 3 2 2  Dist: 2 2 2 7
PH: 16 (4 4 4 4) PH: 12 (3 3 3 3) PH: 8 (2 2 2 2) PH: 4 (1 1 1 1)
PD: 3          PD: 4          PD: 3          PD: 3
LT: 5          LT: 6          LT: 8          LT: 9
--- #3 - Intento 56 (Smart) ---
== Norte ==    == Este ==    == Sur ==    == Oeste ==
* A            * KT654       * Q9832        * J7
* AT87642     * K            * QT93         * J3
* A64         * KT987       * Q2           * J53
* A            * K5          * Q9           * JT87643
Dist: 1 7 3 2  Dist: 5 1 5 2  Dist: 4 3 1 1  Dist: 2 1 3 7
PH: 16 (4 4 4 4) PH: 12 (3 3 3 3) PH: 8 (2 2 2 2) PH: 4 (1 1 1 1)
PD: 3          PD: 4          PD: 3          PD: 3
LT: 5          LT: 6          LT: 8          LT: 9
--- #4 - Intento 7 (Smart) ---
== Norte ==    == Este ==    == Sur ==    == Oeste ==
* A5          * K            * Q2           * JT987643
* AT976       * K843         * Q2           * J35
* A            * K54         * QT87632      * J39
* A9432       * K8765       * QT           * J1
Dist: 2 5 1 5  Dist: 0 3 2 4  Dist: 2 2 7 2  Dist: 8 2 2 1
PH: 16 (4 4 4 4) PH: 12 (3 3 3 3) PH: 8 (2 2 2 2) PH: 4 (1 1 1 1)
PD: 3          PD: 4          PD: 3          PD: 4
LT: 5          LT: 5          LT: 8          LT: 8

Comparación
*** Smart ***
Corridas: 100
Promedio Intentos: 43.59
Exitos: 100
Sin man: 0
Desviación std: 41.19
Tiempo (ms): 635

*** GAT ***
Corridas: 100
Promedio Intentos: 2692.49
Exitos: 100
Sin man: 0
Desviación std: 2500.48
Tiempo (ms): 20920

Pag. 96
23. Ejemplo

Descripción
Norte con 1P, 2C, 3D, Este con 4P, 2C, 6D, Sur con 2P, 3C, 4D

Restricciones

NORTE: P:1 & C:2 & D:3 & T:SP & PH libre
ESTE: P:4 & C:2 & D:6 & T:SP & PH libre
SUR: P:2 & C:3 & D:4 & T:SP & PH libre
OESTE: Dist. libre & PH libre

Resultados

--- #1 - Intento 1 (Smart) ---

--- Norte --- | Este --- | Sur --- | Oeste ---
| 3 | QJT9 | AK | 876542 |
| 97 | Q6 | KT4 | A38532 |
| 952 | K38743 | AQ7T6 | - |
| KJT9753 | 2 | AQ64 | 8 |

Dist: 1 13 7 | Dist: 4 2 6 1 | Dist: 2 3 4 4 | Dist: 6 6 0 1 |

PH: 4 (0 0 0 4) | PH: 9 (3 2 4 0) | PH: 22 (7 3 6 6) | PH: 5 (0 5 0 0)

PD: 3 | PD: 3 | PD: 1 | PD: 5

LT: 8 | LT: 7 | LT: 4 | LT: 6

--- #2 - Intento 1 (Smart) ---

--- Norte --- | Este --- | Sur --- | Oeste ---
| 5 | KJT8 | 97 | AQ6432 |
| 34 | QT | K97 | A86352 |
| 963 | QT754 | AK82 | - |
| AQ10872 | 4 | KT63 | 5 |

Dist: 12 3 7 | Dist: 4 2 6 1 | Dist: 2 3 4 4 | Dist: 6 6 0 1 |

PH: 8 (0 1 0 7) | PH: 9 (4 2 3 0) | PH: 13 (0 3 7 3) | PH: 10 (6 4 0 0)

PD: 3 | PD: 3 | PD: 1 | PD: 5

LT: 7 | LT: 7 | LT: 7 | LT: 4

--- #3 - Intento 1 (Smart) ---

--- Norte --- | Este --- | Sur --- | Oeste ---
| 6 | A875 | 32 | KQ3T94 |
| QT | 72 | AK6 | J98543 |
| 363 | QT754 | AK92 | - |
| A1097653 | K | QT42 | 8 |

Dist: 12 3 7 | Dist: 4 2 6 1 | Dist: 2 3 4 4 | Dist: 6 6 0 1 |

PH: 8 (0 2 1 5) | PH: 9 (4 0 2 3) | PH: 16 (0 7 7 2) | PH: 7 (6 1 0 0)

PD: 3 | PD: 3 | PD: 1 | PD: 5

LT: 8 | LT: 7 | LT: 6 | LT: 5

--- #4 - Intento 1 (Smart) ---

--- Norte --- | Este --- | Sur --- | Oeste ---
| A | Q762 | J5 | K19843 |
| A7 | 35 | QT2 | J98643 |
| 54 | KT9876 | QT32 | - |
| KQT8653 | 7 | AQ42 | 3 |

Dist: 12 3 7 | Dist: 4 2 6 1 | Dist: 2 3 4 4 | Dist: 6 6 0 1 |

PH: 17 (4 4 4 5) | PH: 6 (2 1 3 0) | PH: 10 (1 2 3 4) | PH: 7 (3 3 0 1)

PD: 3 | PD: 3 | PD: 1 | PD: 5

LT: 4 | LT: 7 | LT: 8 | LT: 5

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1.00
Exitos: 100
Sin mano: 0
Desviación std: 0.00
Tiempo (ms): 55

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 144395

GAT no generó manos
### 24. Ejemplo

**Descripción**
Norte y Este con mano tricolor, Sur y Oeste con palo séptimo

**Restricciones**
- NORTE: Patrón 4.4.4.1 & PH libre
- ESTE: Patrón 4.4.4.1 & PH libre
- SUR: Patrón 7.8.8.x & PH libre
- OESTE: Patrón 7.8.8.x & PH libre

**Resultados**

--- #1 - Intento 3 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 9642</td>
<td>4 J</td>
<td>3 A</td>
<td>AKQT875</td>
</tr>
<tr>
<td>♠ A874</td>
<td>♥ JT95</td>
<td>♥ KQ632</td>
<td>♥ -</td>
</tr>
<tr>
<td>♠ 62</td>
<td>♥ K863</td>
<td>♥ AQ9T754</td>
<td>♥ 3</td>
</tr>
<tr>
<td>♠ J742</td>
<td>♥ AK78</td>
<td>♥ -</td>
<td>♥ Q963</td>
</tr>
<tr>
<td>Dist: 4 4 1 4</td>
<td>Dist: 1 4 4 4</td>
<td>Dist: 1 5 7 0</td>
<td>Dist: 7 0 1 5</td>
</tr>
<tr>
<td>PH: 5 (0 4 0 1)</td>
<td>PH: 12 (1 3 1 7)</td>
<td>PH: 11 (0 5 6 0)</td>
<td>PH: 12 (9 0 1 2)</td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 2</td>
<td>PD: 5</td>
<td>PD: 5</td>
</tr>
<tr>
<td>LT: 9</td>
<td>LT: 7</td>
<td>LT: 3</td>
<td>LT: 3</td>
</tr>
</tbody>
</table>

--- #2 - Intento 10 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>KQ5</td>
<td>4</td>
<td>J</td>
<td>A987632</td>
</tr>
<tr>
<td>♥ AK75</td>
<td>9874</td>
<td>Q</td>
<td>J632</td>
</tr>
<tr>
<td>♥ K754</td>
<td>AT86</td>
<td>Q32</td>
<td>9</td>
</tr>
<tr>
<td>♠ 7</td>
<td>♥ T865</td>
<td>♥ AKQ943</td>
<td>♥ 2</td>
</tr>
<tr>
<td>Dist: 4 4 1</td>
<td>Dist: 1 4 4 4</td>
<td>Dist: 1 4 7</td>
<td>Dist: 7 4 1 1</td>
</tr>
<tr>
<td>PH: 15 (5 3 0)</td>
<td>PH: 4 (0 0 4)</td>
<td>PH: 16 (1 2 3 10)</td>
<td>PH: 5 (4 1 0 0)</td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 2</td>
<td>PD: 4</td>
<td>PD: 4</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 9</td>
<td>LT: 4</td>
<td>LT: 7</td>
</tr>
</tbody>
</table>

--- #3 - Intento 9 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>KT53</td>
<td>J</td>
<td>AQ87642</td>
</tr>
<tr>
<td>♥ A535</td>
<td>♥ KQ73</td>
<td>Q</td>
<td>T9642</td>
</tr>
<tr>
<td>♥ KQ37</td>
<td>A843</td>
<td>♠ T9852</td>
<td>♠ -</td>
</tr>
<tr>
<td>♠ K98</td>
<td>♠ 6</td>
<td>♠ AT75432</td>
<td>♠ 3</td>
</tr>
<tr>
<td>Dist: 1 4 4 4</td>
<td>Dist: 1 4 4 1</td>
<td>Dist: 1 0 5 7</td>
<td>Dist: 7 5 0 1</td>
</tr>
<tr>
<td>PH: 16 (5 6 5)</td>
<td>PH: 12 (3 5 4 0)</td>
<td>PH: 5 (1 0 0 4)</td>
<td>PH: 7 (6 0 0 1)</td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 2</td>
<td>PD: 5</td>
<td>PD: 5</td>
</tr>
<tr>
<td>LT: 5</td>
<td>LT: 6</td>
<td>LT: 6</td>
<td>LT: 5</td>
</tr>
</tbody>
</table>

--- #4 - Intento 2 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>K765</td>
<td>2</td>
<td>AQ9T793</td>
</tr>
<tr>
<td>♥ K652</td>
<td>♥ JT85</td>
<td>7</td>
<td>♠ AQ97642</td>
</tr>
<tr>
<td>♥ 585</td>
<td>T</td>
<td>♠ JT82</td>
<td>♠ Q</td>
</tr>
<tr>
<td>♠ 9753</td>
<td>♥ AX44</td>
<td>4 1 4 1</td>
<td>4 1 4 1</td>
</tr>
<tr>
<td>Dist: 1 4 4 4</td>
<td>Dist: 4 1 4 4</td>
<td>Dist: 1 7 4</td>
<td>Dist: 7 4 1 1</td>
</tr>
<tr>
<td>PH: 4 (0 3 1 0)</td>
<td>PH: 11 (3 1 0 7)</td>
<td>PH: 7 (0 0 6 1)</td>
<td>PH: 18 (7 6 3 2)</td>
</tr>
<tr>
<td>PD: 2</td>
<td>PD: 2</td>
<td>PD: 4</td>
<td>PD: 4</td>
</tr>
<tr>
<td>LT: 9</td>
<td>LT: 7</td>
<td>LT: 6</td>
<td>LT: 4</td>
</tr>
</tbody>
</table>

**Comparación**

*** Smart ***
- Corridas: 100
- Promedio Intentos: 23.64
- Exitos: 100
- Sin mano: 0
- Desviación estándar: 24.26
- Tiempo (ms): 560

*** GAT ***
- Corridas: 100
- Promedio Intentos: 281436.00
- Exitos: 10
- Sin mano: 90 (sobre 30000 intentos por corrida)
- Desviación estándar: 6078.76
- Tiempo (ms): 211640
25. Ejemplo

Descripción
Norte y Sur con buen palo de diamantes y PH para game
Este y Oeste con manos balanceadas

Restricciones
Norte: P: sr & C: sr & D: 4-10 & T: sr & PH: 13-16
Este: Dist. balanceada & PH libre
Oeste: Dist. balanceada & PH libre

Resultados

--- #1 - Intento 608 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>T974</td>
<td>AK65</td>
<td>3832</td>
</tr>
<tr>
<td>♥ Q42</td>
<td>♥ A75</td>
<td>♥ KJ83</td>
<td>♥ T96</td>
</tr>
<tr>
<td>♦ AKQ</td>
<td>♦ T95</td>
<td>♦ 8643</td>
<td>♦ 72</td>
</tr>
<tr>
<td>♣ 87653</td>
<td>♣ T94</td>
<td>♣ K</td>
<td>♣ AQJ2</td>
</tr>
</tbody>
</table>

Dist: 1345 Dist: 4333 Dist: 4441 Dist: 4324
PH: 14 (2 2 10 0) PH: 4 (0 4 0 0) PH: 14 (7 4 0 3) PH: 8 (1 0 0 7)
PD: 2   PD: 0   PD: 2   PD: 1
LT: 6   LT: 11  LT: 7   LT: 9

--- #2 - Intento 3196 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>763</td>
<td>AK5T54</td>
<td>Q82</td>
</tr>
<tr>
<td>♥ T9543</td>
<td>♥ 8762</td>
<td>♥ KQ</td>
<td>♥ AJ</td>
</tr>
<tr>
<td>♦ A874</td>
<td>♦ K3</td>
<td>♦ QT52</td>
<td>♦ J96</td>
</tr>
<tr>
<td>♣ AKQ</td>
<td>♣ T853</td>
<td>♣ J</td>
<td>♣ 97642</td>
</tr>
</tbody>
</table>

Dist: 1543 Dist: 3424 Dist: 6241 Dist: 3235
PH: 13 (0 0 4 9) PH: 3 (0 0 3 0) PH: 16 (8 5 2 1) PH: 8 (2 5 1 0)
PD: 2   PD: 1   PD: 3   PD: 1
LT: 6   LT: 10  LT: 5   LT: 9

--- #3 - Intento 635 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q6</td>
<td>J82</td>
<td>AT974</td>
<td>K53</td>
</tr>
<tr>
<td>♥ JT</td>
<td>♥ 7642</td>
<td>♥ AK5</td>
<td>♥ Q983</td>
</tr>
<tr>
<td>♦ QT62</td>
<td>♦ A3</td>
<td>♦ K853</td>
<td>♦ 94</td>
</tr>
<tr>
<td>♣ AK37</td>
<td>♣ 9652</td>
<td>♣ 8</td>
<td>♣ QT43</td>
</tr>
</tbody>
</table>

Dist: 2544 Dist: 3424 Dist: 5341 Dist: 3424
PH: 13 (2 1 2 8) PH: 6 (1 0 5 0) PH: 14 (4 7 3 0) PH: 7 (3 2 0 2)
PD: 2   PD: 1   PD: 2   PD: 1
LT: 7   LT: 10  LT: 6   LT: 8

--- #4 - Intento 3454 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>AK98</td>
<td>J65</td>
<td>72</td>
<td>T43</td>
</tr>
<tr>
<td>♥ 5</td>
<td>♥ T32</td>
<td>♥ KQ9764</td>
<td>♥ A18</td>
</tr>
<tr>
<td>♦ 96542</td>
<td>♦ 83</td>
<td>♦ AQ17</td>
<td>♦ K3</td>
</tr>
<tr>
<td>♣ AT</td>
<td>♣ J974</td>
<td>♣ K</td>
<td>♣ 98532</td>
</tr>
</tbody>
</table>

Dist: 5152 Dist: 3325 Dist: 2641 Dist: 3325
PH: 13 (9 0 0 4) PH: 4 (1 0 0 3) PH: 14 (0 5 6 3) PH: 9 (0 5 4 0)
PD: 3   PD: 1   PD: 3   PD: 1
LT: 5   LT: 10  LT: 5   LT: 9

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1354.16
Éxitos: 100
Sin mano: 0
Desviación std: 1174.47
Tiempo (ms): 26075

*** GAT ***
Corridas: 100
Promedio Intentos: 10139.81
Éxitos: 95
Sin mano: 5 (sobre 30000 intentos por corrida)
Desviación std: 8109.31
Tiempo (ms): 52868
26. Ejemplo

Descripción
Norte con 3 fallos, Este con A de pique, Sur con 25 PH, Oeste con más de 8 corazones

Restricciones
NORTE: PD: 9-9 & PH libre
ESTE : Dist. libre & PH libre & { AP }
SUR : Dist. libre & PH: 25-25
OESTE: P:5r & C:1-13 & Dist & Tisr & PH libre

Resultados
--- #1 - Intento 4590 (Smart) ---
== Norte ==     == Este ==     == Sur ==     == Oeste ==
♦ ♦          ♦ AT987          ♦ KQJ42          ♦ 653
♥ ♦          ♥ -              ♥ AKQJ          ♥ T98765432
♥ ♦          ♥ JT976543      ♥ AKQ8          ♥ 2
♣ ♦          ♣ AKQJT9876543  ♣ -              ♣ -
Dist: 0 0 10 13 Dist: 4 0 8 0 Dist: 5 4 4 0 Dist: 3 9 1 0
PH: 10 (0 0 0 10) PH: 5 (4 0 1 0) PH: 25 (6 10 9 0) PH: 0 (0 0 0 0)
PD: 9          PD: 6              PD: 3          PD: 5
LT: 0          LT: 5              LT: 1          LT: 7
--- #2 - Intento 4487 (Smart) ---
== Norte ==     == Este ==     == Sur ==     == Oeste ==
♦ ♦          ♦ AT986          ♦ KQ532          ♦ 74
♥ ♦          ♥ -              ♥ AKQ1          ♥ T98765432
♥ ♦          ♥ JT98765432    ♥ -              ♥ -
Dist: 0 0 13 0 Dist: 5 0 0 7 Dist: 5 4 0 4 Dist: 2 9 0 2
PH: 10 (0 0 10 0) PH: 5 (5 0 0 0) PH: 25 (5 10 0 10) PH: 0 (0 0 0 0)
PD: 9          PD: 6              PD: 3          PD: 5
LT: 0          LT: 5              LT: 1          LT: 7
--- #3 - Intento 22612 (Smart) ---
== Norte ==     == Este ==     == Sur ==     == Oeste ==
♦ ♦          ♦ AT87653        ♦ KQ94          ♦ 32
♥ ♦          ♥ -              ♥ AKQJ          ♥ T98765432
♥ ♦          ♥ JT98765432    ♥ -              ♥ -
Dist: 0 0 13 0 Dist: 6 0 0 6 Dist: 4 4 0 5 Dist: 2 9 0 2
PH: 10 (0 0 10 0) PH: 4 (4 0 0 0) PH: 25 (5 10 0 10) PH: 1 (1 0 0 0)
PD: 9          PD: 6              PD: 3          PD: 5
LT: 0          LT: 5              LT: 1          LT: 7
--- #4 - Intento 4719 (Smart) ---
== Norte ==     == Este ==     == Sur ==     == Oeste ==
♦ ♦          ♦ AT87642        ♦ KQ9          ♦ 953
♥ ♦          ♥ -              ♥ AKQJ          ♥ T98765432
♥ ♦          ♥ JT86543        ♥ -              ♥ -
Dist: 0 0 13 0 Dist: 6 0 6 0 Dist: 3 4 6 0 Dist: 3 9 1 0
PH: 10 (0 0 10 0) PH: 5 (4 0 1 0) PH: 25 (6 10 9 0) PH: 0 (0 0 0 0)
PD: 9          PD: 6              PD: 3          PD: 5
LT: 0          LT: 5              LT: 1          LT: 7

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 22574.70
Exitos: 73
Sin mano: 27 (sobre 30000 intentos por corrida)
Desviación std: 10916.64
Tiempo (ms): 227240

*** GAT ***
Corridas: 100
Promedio Intentos: 0
Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 151935

GAT no generó manos
27. Ejemplo

Descripción

Ningún jugador con PD, Norte, Este y Sur con PH en todos los palos

Restricciones

- **NORTE**: PD: 0-0 & PH:1-10 & PHC:1-10 & PHD:1-10 & PHT:1-10
- **ESTE**: PD: 0-0 & PH:1-10 & PHC:1-10 & PHD:1-10 & PHT:1-10
- **SUR**: PD: 0-0 & PH:1-10 & PHC:1-10 & PHD:1-10 & PHT:1-10
- **OSTE**: PD: 0-0 & PH Libre

Resultados

--- #1 - Intento 2132 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ K743</td>
<td>♠ AT9</td>
<td>♠ J82</td>
<td>♠ Q65</td>
</tr>
<tr>
<td>♥ Q72</td>
<td>♥ K96</td>
<td>♥ A33</td>
<td>♥ T854</td>
</tr>
<tr>
<td>♦ AJ8</td>
<td>♦ K764</td>
<td>♦ QT5</td>
<td>♦ 932</td>
</tr>
<tr>
<td>♣ A75</td>
<td>♣ KQT</td>
<td>♣ J986</td>
<td>♣ 432</td>
</tr>
</tbody>
</table>

Dist: 4333 Dist: 3343 Dist: 3343 Dist: 3433

PH: 14 (3 2 5 4) PH: 15 (4 3 3 5) PH: 9 (1 5 2 1) PH: 2 (2 0 0 0)

PO: 0 PO: 0 PO: 0 PO: 0

LT: 8 LT: 7 LT: 10 LT: 11

--- #2 - Intento 297 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ KQT5</td>
<td>♠ A86</td>
<td>♠ J94</td>
<td>♠ 732</td>
</tr>
<tr>
<td>♥ J97</td>
<td>♥ AK43</td>
<td>♥ Q62</td>
<td>♥ 935</td>
</tr>
<tr>
<td>♦ AT9</td>
<td>♦ Q74</td>
<td>♦ J863</td>
<td>♦ K52</td>
</tr>
<tr>
<td>♣ Q43</td>
<td>♣ A38</td>
<td>♣ KT9</td>
<td>♣ 7562</td>
</tr>
</tbody>
</table>

Dist: 4333 Dist: 3433 Dist: 3433 Dist: 3334

PH: 12 (5 1 4 2) PH: 18 (4 7 2 5) PH: 7 (1 2 1 3) PH: 3 (0 0 3 0)

PO: 0 PO: 0 PO: 0 PO: 0

LT: 8 LT: 7 LT: 10 LT: 11

--- #3 - Intento 61 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ Q98</td>
<td>♠ KT4</td>
<td>♠ AJ35</td>
<td>♠ 632</td>
</tr>
<tr>
<td>♥ J18</td>
<td>♥ A065</td>
<td>♥ K92</td>
<td>♥ 743</td>
</tr>
<tr>
<td>♦ Q32</td>
<td>♦ A86</td>
<td>♦ K97</td>
<td>♦ 543</td>
</tr>
<tr>
<td>♣ K63</td>
<td>♣ 384</td>
<td>♣ QT5</td>
<td>♣ 6972</td>
</tr>
</tbody>
</table>

Dist: 3343 Dist: 3433 Dist: 4333 Dist: 3334

PH: 9 (2 1 3 3) PH: 14 (3 6 4 1) PH: 13 (5 3 3 2) PH: 4 (0 0 0 4)

PO: 0 PO: 0 PO: 0 PO: 0

LT: 9 LT: 8 LT: 8 LT: 11

--- #4 - Intento 412 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ AK6</td>
<td>♠ J9S2</td>
<td>♠ Q76</td>
<td>♠ T43</td>
</tr>
<tr>
<td>♥ AK9</td>
<td>♥ Q64</td>
<td>♥ JS2</td>
<td>♥ 873</td>
</tr>
<tr>
<td>♦ K865</td>
<td>♦ AJ4</td>
<td>♦ QT7</td>
<td>♦ 932</td>
</tr>
<tr>
<td>♣ A92</td>
<td>♣ Q87</td>
<td>♣ KT3</td>
<td>♣ 654</td>
</tr>
</tbody>
</table>

Dist: 3343 Dist: 4333 Dist: 3343 Dist: 3433

PH: 21 (7 7 3 4) PH: 10 (1 2 5 2) PH: 9 (2 1 2 4) PH: 0 (0 0 0 0)

PO: 0 PO: 0 PO: 0 PO: 0

LT: 6 LT: 9 LT: 9 LT: 12

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1537.03
Exitos: 100
Sin mano: 0
Desviación std: 1613.54
Tiempo (ms): 31470

*** CAT ***
Corridas: 100
Promedio Intentos: 1475863.00
Exitos: 2
Sin mano: 98 (sobre 30000 intentos por corrida)
Desviación std: 3466.45
Tiempo (ms): 158475
28. Ejemplo

Descripción
Todos los jugadores con dist. 4.3.3.3 y todos con alguna figura o As en cada palo

Restricciones


--- #1 - Intento 1231 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ KT2</td>
<td>♠ J94</td>
<td>♠ Q853</td>
<td>♠ A76</td>
</tr>
<tr>
<td>♥ J97</td>
<td>♥ A863</td>
<td>♥ KT4</td>
<td>♥ Q52</td>
</tr>
<tr>
<td>♦ J176</td>
<td>♦ A42</td>
<td>♦ Q83</td>
<td>♦ K95</td>
</tr>
<tr>
<td>♣ A72</td>
<td>♣ K84</td>
<td>♣ QT3</td>
<td>♣ J963</td>
</tr>
<tr>
<td>Dist: 334</td>
<td>Dist: 343</td>
<td>Dist: 433</td>
<td>Dist: 334</td>
</tr>
<tr>
<td>PH: 9 (31114)</td>
<td>PH: 12 (1443)</td>
<td>PH: 9 (2322)</td>
<td>PH: 10 (4231)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
</tr>
<tr>
<td>LT: 19</td>
<td>LT: 9</td>
<td>LT: 8</td>
<td>LT: 9</td>
</tr>
</tbody>
</table>

--- #2 - Intento 399 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ J743</td>
<td>♠ AT2</td>
<td>♠ Q95</td>
<td>♠ K86</td>
</tr>
<tr>
<td>♥ K96</td>
<td>♥ AT53</td>
<td>♥ Q74</td>
<td>♥ J82</td>
</tr>
<tr>
<td>♦ Q54</td>
<td>♦ A63</td>
<td>♦ KT8</td>
<td>♦ J972</td>
</tr>
<tr>
<td>♣ K85</td>
<td>♣ A97</td>
<td>♣ Q832</td>
<td>♣ J74</td>
</tr>
<tr>
<td>Dist: 433</td>
<td>Dist: 343</td>
<td>Dist: 334</td>
<td>Dist: 334</td>
</tr>
<tr>
<td>PH: 9 (1323)</td>
<td>PH: 16 (4444)</td>
<td>PH: 9 (2232)</td>
<td>PH: 6 (3111)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
</tr>
<tr>
<td>LT: 9</td>
<td>LT: 8</td>
<td>LT: 8</td>
<td>LT: 11</td>
</tr>
</tbody>
</table>

--- #3 - Intento 226 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ A65</td>
<td>♠ JT8</td>
<td>♠ K92</td>
<td>♠ Q743</td>
</tr>
<tr>
<td>♥ QT9</td>
<td>♥ J862</td>
<td>♥ K53</td>
<td>♥ A74</td>
</tr>
<tr>
<td>♦ K853</td>
<td>♦ Q94</td>
<td>♦ AT6</td>
<td>♦ J72</td>
</tr>
<tr>
<td>♣ Q43</td>
<td>♣ K96</td>
<td>♣ A872</td>
<td>♣ J75</td>
</tr>
<tr>
<td>Dist: 343</td>
<td>Dist: 343</td>
<td>Dist: 334</td>
<td>Dist: 433</td>
</tr>
<tr>
<td>PH: 11 (4232)</td>
<td>PH: 7 (1123)</td>
<td>PH: 14 (3344)</td>
<td>PH: 8 (2411)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
</tr>
<tr>
<td>LT: 8</td>
<td>LT: 10</td>
<td>LT: 8</td>
<td>LT: 10</td>
</tr>
</tbody>
</table>

--- #4 - Intento 547 (Smart) ---

<table>
<thead>
<tr>
<th>Norte</th>
<th>Este</th>
<th>Sur</th>
<th>Oeste</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠ A63</td>
<td>♠ K52</td>
<td>♠ QT87</td>
<td>♠ J94</td>
</tr>
<tr>
<td>♥ A73</td>
<td>♥ K42</td>
<td>♥ J96</td>
<td>♥ QT8</td>
</tr>
<tr>
<td>♦ Q98</td>
<td>♦ A63</td>
<td>♦ J72</td>
<td>♦ K754</td>
</tr>
<tr>
<td>♣ Q864</td>
<td>♣ A53</td>
<td>♣ J72</td>
<td>♣ KT9</td>
</tr>
<tr>
<td>Dist: 334</td>
<td>Dist: 343</td>
<td>Dist: 433</td>
<td>Dist: 334</td>
</tr>
<tr>
<td>PH: 12 (4422)</td>
<td>PH: 14 (3344)</td>
<td>PH: 5 (2111)</td>
<td>PH: 9 (1323)</td>
</tr>
<tr>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
<td>PD: 0</td>
</tr>
<tr>
<td>LT: 8</td>
<td>LT: 8</td>
<td>LT: 11</td>
<td>LT: 9</td>
</tr>
</tbody>
</table>

Comparación

*** Smart ***

Corridas: 100
Promedio Intentos: 870.37
 Exitos: 100
Sin mano: 0
Desviación std: 735.70
Tiempo (ms): 18320

*** GAT ***

Corridas: 100
Promedio Intentos: 0
 Exitos: 0
Sin mano: 100 (sobre 30000 intentos por corrida)
Tiempo (ms): 151935

GAT no generó manos
29. Ejemplo

Descripción
Todos los jugadores de 1 a 6 PH en cada palo.

Restricciones

Resultados
--- #1 - Intento 3229 (smart) ---
- Norte
  - K
  - J86
  - TQ43
- Este
  - A
  - Q76
  - AT942
- Sur
  - 9
  - 5
  - 3
- Oeste
  - 2
  - 1
  - 0

--- #2 - Intento 18207 (smart) ---
- Norte
  - K
  - Q
  - A
  - 2
  - 1
  - 0
- Este
  - J
  - 8
  - 7
- Sur
  - 9
  - 6
  - 4
- Oeste
  - 8
  - 7
  - 6

--- #3 - Intento 42 (smart) ---
- Norte
  - A
  - J
  - 8
- Este
  - A
  - 7
  - 6
- Sur
  - 5
  - 4
  - 3
- Oeste
  - 2
  - 1
  - 0

--- #4 - Intento 5484 (smart) ---
- Norte
  - A
  - 8
  - 7
- Este
  - 6
  - 5
  - 4
- Sur
  - 3
  - 2
  - 1
- Oeste
  - 0
  - 1
  - 2

Comparación
*** Smart ***
Corridas: 100
Promedio Intentos: 6885.34
Exitos: 100
Sin manio: 0
Desviación std: 6219.23
Tiempo (ms): 118215

*** GAT ***
Corridas: 100
Promedio Intentos: 25479.55
Exitos: 69
Sin manio: 31 (sobre 30000 intentos por corrida)
Desviación std: 10190.42
Tiempo (ms): 198190
30. Ejemplo

Descripción
Todos los jugadores con un As y de 1 a 6 PH en cada palo.

Restricciones

Resultados
--- #1 Intento 3329 (Smart) ---
  Norte =  [K96 , J83 , Q76 , A]  
  Este =  [A7543 , K76 , A92 , ]  
  Sur =  [J82 , A542 , ]  
  Oeste =  [Q , ]  
  Dist: 4 4 1  
  PH: 10 (3 1 2 4)  
  PD: 2  
  LT: 7  
--- #2 Intento 18207 (Smart) ---
  Norte =  [K87652 , Q43 , K , J]  
  Este =  [A9 , Q76 , ]  
  Sur =  [A9 , ]  
  Oeste =  [J , ]  
  Dist: 6 1 3 3  
  PH: 11 (3 3 1 4)  
  PD: 2  
  LT: 8  
--- #3 Intento 42 (Smart) ---
  Norte =  [K86 , J5 , J853 , J54]  
  Este =  [Q , KT92 , ]  
  Sur =  [A7432 , Q , ]  
  Oeste =  [J95 , ]  
  Dist: 3 2 4 4  
  PH: 6 (3 1 1 1)  
  PD: 1  
  LT: 10  
--- #4 Intento 5484 (Smart) ---
  Norte =  [K84 , Q98 , QT976 , J6]  
  Este =  [AT , K432 , ]  
  Sur =  [Q963 , ]  
  Oeste =  [J752 , ]  
  Dist: 3 3 5 2  
  PH: 8 (3 2 2 1)  
  PD: 1  
  LT: 8  

Comparación
*** Smart ***
Corridas: 100
Promedio Intentos: 6396.47
Exitos: 98
Sin mano: 2 (sobre 30000 intentos por corrida)
Desviación std: 6829.04
Tiempo (ms): 80365

*** GAT ***
Corridas: 100
Promedio Intentos: 29069.31
Exitos: 64
Sin mano: 36 (sobre 30000 intentos por corrida)
Desviación std: 10643.26
Tiempo (ms): 92265
31. Ejemplo

Descripción
Todos los jugadores con un As y de 1 a 4 PH en cada palo.

Restricciones
NORTE: Dist. libre & PhP:1-4 & PhC:1-4 & PhD:1-4 & PhT:1-4 & { AP }
ESTE: Dist. libre & PhP:1-4 & PhC:1-4 & PhD:1-4 & PhT:1-4 & { AC }
SUR: Dist. libre & PhP:1-4 & PhC:1-4 & PhD:1-4 & PhT:1-4 & { AD }

Resultados
--- #1 - Intento 5312 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
◆ A1B3 ♠ K7652 ♠ J94 ♠ Q
♥ QT63 ♥ A ♥ J92 ♥ KB754
♦ K95 ♦ Q7 ♦ A86432 ♦ JT
♣ JT ♣ K7532 ♣ Q ♣ A9864
Dist: 4 4 3 2 Dist: 5 0 2 5 Dist: 3 3 6 1 Dist: 1 5 2 3
PH: 10 (4 2 3 1) PH: 12 (3 4 2 3) PH: 8 (1 4 2 7) PH: 10 (2 3 1 4)
PD: 1 PD: 4 PD: 2 PD: 3
LT: 8 LT: 6 LT: 9 LT: 7
--- #2 - Intento 3599 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
◆ A183 ♠ J954 ♠ Q72 ♠ KT6
♥ J864 ♥ A7 ♥ K192 ♥ Q53
♦ K753 ♦ Q8 ♦ A6 ♦ JT942
♣ JT ♣ QT982 ♣ K543 ♣ A7
Dist: 4 4 2 Dist: 4 2 2 5 Dist: 3 4 2 4 Dist: 3 3 5 2
PH: 8 (4 1 3 1) PH: 9 (1 4 2 2) PH: 12 (2 3 4 3) PH: 10 (3 2 1 4)
PD: 2 PD: 2 PD: 1 PD: 1
LT: 8 LT: 7 LT: 7 LT: 8
--- #3 - Intento 5374 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
◆ A178 ♠ K96543 ♠ Q2 ♠ J7
♥ J96 ♥ A ♥ J543 ♥ K1872
♦ J7 ♦ KT96 ♦ A8642 ♦ J3
♣ QT963 ♣ J8 ♣ K4 ♣ A752
Dist: 3 3 2 5 Dist: 6 0 4 2 Dist: 2 4 5 2 Dist: 2 5 2 4
PH: 10 (4 2 2 7) PH: 11 (3 4 3 1) PH: 10 (2 1 4 3) PH: 9 (1 3 1 4)
PD: 1 PD: 4 PD: 2 PD: 2
LT: 8 LT: 6 LT: 8 LT: 8
--- #4 - Intento 1840 (Smart) ---
--- Norte --- --- Este --- --- Sur --- --- Oeste ---
◆ A682 ♠ J753 ♠ K19 ♠ Q4
♥ K742 ♥ A5 ♥ J786 ♥ Q93
♦ J6 ♦ K632 ♦ A97 ♦ QT854
♣ Q ♣ KT9853 ♣ J72 ♣ A64
Dist: 4 4 4 3 Dist: 4 1 1 6 Dist: 3 4 3 3 Dist: 2 3 5 3
PH: 12 (4 3 3 2) PH: 9 (1 4 1 3) PH: 9 (3 1 4 1) PH: 10 (2 2 2 4)
PD: 2 PD: 4 PD: 0 PD: 1
LT: 7 LT: 6 LT: 10 LT: 8

Comparación

*** Smart ***
Corridas: 100
Promedio Intentos: 1800.24
Exitos: 100
Sin mano: 0
Desviación std: 2000.42
Tiempo (ms): 17140

*** CAT ***
Corridas: 100
Promedio Intentos: 13927.83
Exitos: 84
Sin mano: 16 (sobre 30000 intentos por corrida)
Desviación std: 10433.06
Tiempo (ms): 63930
Apéndice 6: Glosario de Bridge

Incluimos acá un glosario general sobre el juego de Bridge, con términos empleados en la presente tesis, incluyendo aquellos que son más comúnmente usados por la comunidad de practicantes de este juego.

- Abridor: La persona que declara primero en el remate un canto distinto de paso

- Afirmación (de un palo): El poseer buenas cartas de un determinado palo que le permite ganar (al jugador o al bando) todas las bazas restantes de ese palo

- Apoyo: Declaración a un palo previamente mencionado por el compañero. También, posesión de un número suficiente de cartas a un palo declarado por el compañero como para efectuar una declaración de apoyo.

- Asistir: Jugar un jugador una carta del mismo palo de la carta jugada por el que jugó la primera de la baza (lo cual es obligación hacer en caso de tenerla).

- Baza: Secuencia de 4 cartas jugadas una por cada jugador a su turno (el carteo consta en total de 13 bazas)

- Baza extra o sobrebaza: Cada baza ganada por el bando declarante en exceso del contrato.

- Baza de menos, baja, multa o down: cada baza que le faltó ganar al bando declarante para cumplir su contrato.

- Baza subastada: Cada baza que se compromete a ganar el bando declarante en exceso de seis. Se indican con el número en el contrato.

- Blackwood: Empleo convencional del canto de 4 ST para preguntar al compañero cuántos ases posee, como precaución para declarar un slam sin que falte más de un as.

- Bridge duplicado: Bridge de competición en sus distintas modalidades. Todas ellas tienen en común que cada mano se juega de idéntica forma en como mínimo 6 mesas (juego por equipos), o más (juego individual o por parejas). El objetivo es obtener mejores puntuajes que otras parejas con las mismas cartas.

- Canto, voz o declaración: cualquier jugada (oral) de la subasta: un número del 1 al 7 seguido de un palo o de sin triunfo, doble, redoble o paso.

- Carteo: Fase de la partida durante la cual se juegan las cartas, luego del remate, y en la cual el declarante debe intentar cumplir el contrato mientras que la defensa debe evitarlo.

- Contrato: Compromiso asumido por el bando declarante de ganar por lo menos la cantidad de bazas especificada en la declaración final del remate. También el valor comprometido (de 1 a 7) y la denominación en la que se jugará (cierto palo o sin triunfo).

- Convención: Canto o jugada que por acuerdo entre compañeros sirve para transmitir una información acerca de la mano del que la emplea; puede ser artificial.

- Cue-bid: Declaración imprescindible en una denominación a la que muy posiblemente no se desea jugar, como por ejemplo, al palo de los adversarios, para mostrar una mano fuerte o una distribución determinada.

- Dador: Jugador que reparte y debe efectuar el primer canto en la subasta.

- Deal o reparto: El conjunto de las 4 manos, que serán jugadas.
- Declaración: Compromiso de ganar como mínimo la cantidad de bazas indicada en una denominación especificada (a un palo dado, o a sin triunfo). Una proposición para un contrato, que debe ser más alta que las declaraciones precedentes.

- Declarante: Jugador que nombró primero el palo al que se juega el contrato. Juega sus cartas y las del muerto durante el corteo. También, bando de este jugador.

- Defensor: Oponente del bando declarante.

- Denominación: Palo o sin triunfo, especificado en una declaración.

- Descarte: Jugar una carta de otro palo porque no se tiene una del palo de la primera carta jugada de la baza.

- Doblo: Canto, sobre la declaración de un oponente, que aumenta (a aproximadamente el doble) la puntuación de los contratos cumplidos así como de las multas.

- Doubleton: Tener sólo dos cartas de un determinado palo.

- Fallo: No tener cartas en determinado palo. También el jugar de una carta de triunfo en una baza iniciada con una carta de otro palo.

- Fallar: El jugar de una carta de triunfo en una baza iniciada con una carta de otro palo.

- Finesse: Jugar la carta más baja de un grupo no secuencial de honores (A-Q ó K-J, por ej.) esperando (si la suerte acompaña) que la carta faltante esté en la mano del siguiente jugador (oponente).

- Fit: Poseer con su compañero al menos 8 cartas de un cierto palo (número generalmente aceptado como apto para un posible palo de triunfo).

- Game, manga o juego: Contrato para ganar 100 puntos o más en premios por bazas (los cantos mínimos posibles para esto son 4 picas, 4 corazones, 5 diamantes, 5 tréboles, y 3 sin triunfo)

- Gran slam: contrato para ganar 13 bazas (cualquier canto a la altura de 7)

- Honor: Cada una de las 5 cartas más altas de cada palo: As, K, Q, J, 10.

- Honor mayor: Cualquiera de las cartas As, K, Q.

- Intervención: Cualquier declaración distinta de paso (incluso doblo) del bando que no ha abierto la subasta.

- Inverso: Redeclaración al nivel de dos o más de un palo de rango superior al declarado previamente.

- Mano: Cartas originalmente repartidas a un jugador. También, conjunto de subasta y corteo que constituyen un juego completo.

- Marca parcial o puntaje parcial: Todo contrato de menos de 100 puntos en premios por bazas.

- Muerto: Compañero del declarante. También, mano o cartas de dicho jugador, una vez expuestas sobre la mesa tras la salida inicial.

- Multa: castigo que resta puntos al bando declarante por bazas declaradas y no cumplidas.

- Muerto: El jugador quien coloca sus cartas sobre la mesa antes del comienzo del corteo y posteriormente no toma parte en el juego.

- No Vulnerable El "estado" de un equipo donde aún no se ha ganado un game; con lo cual los puntajes por game y slam son más pequeños.
- Obstrucción: Declaración, generalmente con salto, efectuada más con el propósito de quitar espacio de subasta a los adversarios que esperando cumplir un contrato.

- Palo: Cada una de las 4 familias de 13 cartas en el mazo: Tréboles, Diamantes, Corazones, Picas (o Piques).

- Palos mayores, nobles o ricos: Los palos de Corazones y Picas.

- Palos menores o pobres: Los palos de Tréboles y Diamantes.

- Parada: Carta o combinación de la cual se espera que interrumpa el desfile (afirmación) de un palo por parte de los adversarios.

- Pareja: La asociación de 2 jugadores uno enfrente del otro.

- Partida libre: Partida que no está incluida en el bridge duplicado o de competición. La juegan cuatro o más jugadores (por turno) en una sola mesa y el objetivo es ganar puntos, que pueden representar un valor para obtener un premio. Las dos modalidades principales son el rubber y el Chicago.

- Paso: Canto que deja la subasta tal como está en ese momento. Tres pasos consecutivos finalizan la subasta.

- Pequeño slam: Contrato para ganar 12 bazas (cualquier canto a la altura de 6)

- Premios por bazas: Puntos ganados por el bando declarante por cumplir el contrato.

- Premios especiales: Puntos ganados por cumplir un contrato, además de los premios por bazas.

- Promover: Afirmar un honor menor eliminando (mediante jugadas previas) los honores más altos de los oponentes.

- Puntos de match: Unidad de puntuación del bridge duplicado asignada a través de la comparación con otros resultados de jugadores que recibieron idénticas manos.

- Reabrir: Efectuar una declaración o un doblo cuando la subasta de los adversarios se ha detenido a un nivel bajo.

- Redeclaración: Canto sucesivo de un jugador que ha abierto la subasta o que a respondido a una apertura de su compañero.

- Redoble: Canto sobre el doble de un oponente que aumenta la puntuación de los contratos cumplidos o multados. (a aproximadamente el cuádruple del puntaje original).

- Respondedor: El compañero del abridor.

- Respuesta: Primer canto del compañero del abridor tras la apertura.

- Rubber: Conjunto de dos games ganadas por un bando sobre un total de tres.

- Salida o salida inicial: Durante el corteo, primera carta jugada en una baza (lo que corresponde al equipo de la defensa).

- Secuencia: Una serie de al menos 3 cartas en secuencia; por ejemplo: A K Q, K Q J, Q J 10, J 10 9, 7 6 5 4 3 2, etc.

- Semifallo o singleton: El tener una sola carta de un palo determinado.

- Sin triunfo (ST): modalidad de juego en donde no hay un palo de triunfo. También, canto en la subasta en donde se dicen estas dos palabras explicitamente, proponiendo esta modalidad de juego

- Sistema de subasta (o de remate): Conjunto de pactos y convenciones acordados con el compañero para el desarrollo de la subasta. Hay sistemas de subasta naturales, como el Mayor Quinto, el Acol y otros, y sistemas artificiales, como el Trébol de Precisión, el Trébol Napolitano y otros.
- **Slam**: Contrato que exige ganar todas las bazas menos una (pequeño slam, son 12 bazas), o todas las bazas (gran slam, son 13 bazas).

- **Stayman**: Convención artificial muy conocida que tiene por objeto preguntar al compañero por un posible fit a un palo mayor, tras una apertura a sin triunfo.

- **Subasta o remate**: Proceso para determinar el contrato, para uno de los bandos, por medio de cantos sucesivos de todos los jugadores. También, la secuencia de los cantos efectuados.

- **Texas**: Convención para que el abridor a sin triunfo, y no su compañero, juegue el game a un palo mayor.

- **Torneo duplicado**: Competición de Bridge donde todos los jugadores tienen la oportunidad de jugar con las mismas manos, preparadas previamente.

- **Triunfar**: Jugar una carta del palo de triunfo

- **Triunfo**: Un palo escogido (del último canto de la subasta, distinto de paso) cuyas cartas poseen durante el carteo más valor que las cartas de cualquier otro palo.

- **Vulnerable**: Situación en la que se encuentra un bando si ya ganó un game durante el transcurso del rubber, lo cual le incrementa el valor de games y slams posteriores así como también las multas posteriores.