# The Complexity of Rummikub Problems

Jan N. van Rijn    Frank W. Takes    Jonathan K. Vis

LIACS, Leiden University

BNAIC 2015 — November 5, 2015

# Rummikub game

Universiteit Leiden

- Start with 14 tiles
- Pool of remaining tiles
- Two types of sets of at least three tiles:
  - Groups: same value, different suit
  - Runs: same suit, consecutive values
- Game ends when a player gets rid of all his tiles

# Groups and runs

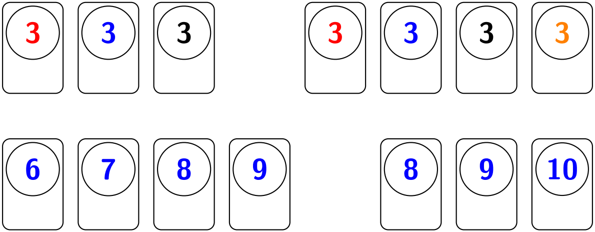



Figure : Two valid groups and two valid runs.

# Inspired by . . .

**Benelux Algorithm Programming Contest**

Various Rummikub assignments

- 2006 Main contest
  - 50 teams
  - 19 submissions
  - 3 correct solutions
- 2015 Preliminaries
  - The “easy” problem

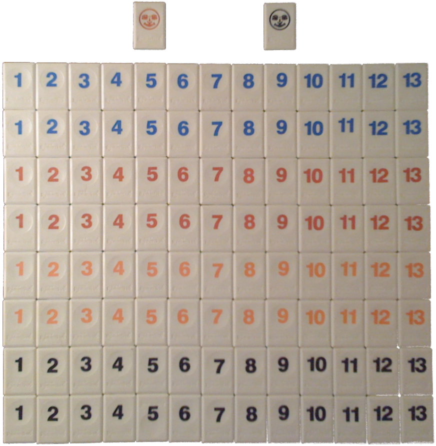
# Rummikub puzzle

Tileset parameters

- Numbers $n$ (default: 13)
- Suits/colors $k$ (default: 4)
- Duplicates $m$ (default: 2)

Set size constants

- Minimal set size (3)



# Problem statement

Universiteit Leiden

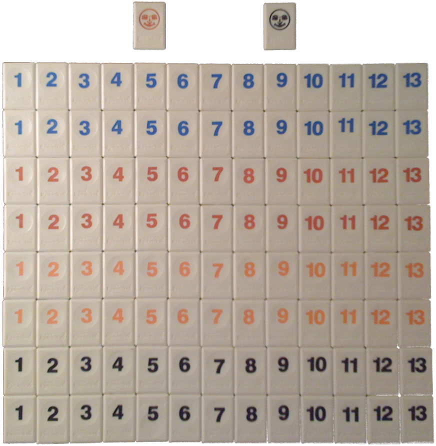
- **Rummikub puzzle**
  Given a subset of the Rummikub tile set of $n \times k \times m$ tiles with $n$ values, $k$ suits and $m$ copies of each tile, form valid sets of runs and groups such that the score (sum of used tile values) is maximized.

# Previous and related work

- Hand-making games
- ILP by den Hartog et al.
  - Mostly used for NP-hard problems
  - Complexity of Rummikub Puzzle undetermined
- Can we do something better?

# Algorithm

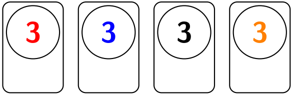
- Backtracking algorithm
- From low tile values to high tile values
- For value $v = 1$ to $v = n$
  - Partition tiles with value $v$ in runs and groups



# Algorithm

First forming groups and then forming runs

- Colors ($k$) and duplicates ($m$)
- $m = 1$: 6 options
- $m = 2$: 27 options
- Exponential increase
- Memory size



## Towards a polynomial algorithm

Universiteit Leiden

- Number of groups for a given number of suits $k$ and set size parameter $s$:

$$G(k,1) = 1 + \sum_{i=s}^{k} \binom{k}{i}$$

- Adding the number of duplicates $m$:

$$G(k,m) \leq G(k,1)^m$$

- Not polynomial ...

# Towards a polynomial algorithm

Universiteit Leiden

- Number of groups for a given number of suits $k$ and set size parameter $s$:

$$G(k,1) = 1 + \sum_{i=s}^{k} \binom{k}{i}$$

- Adding the number of duplicates $m$:

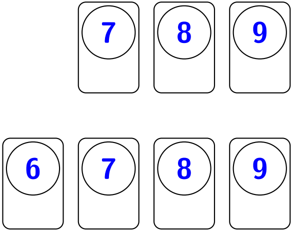
$$G(k,m) \leq G(k,1)^m$$

- Not polynomial ...
- First forming runs and then forming groups (focus of the remainder of this talk)

# Algorithm

Forming runs

- Decide for each tile which run to continue
- Try both and save the one with highest score



# State space

- Only runs of length 0, 1, 2 and 3 or longer have to be distinguished
- For $m = 2$, the number of different configurations of runs for a particular suit is 10: $\{0,0\}$, $\{0,1\}$, $\{0,2\}$, $\{0,3\}$, $\{1,1\}$, $\{1,2\}$, $\{1,3\}$, $\{2,2\}$, $\{2,3\}$, and $\{3,3\}$.
- For any $m$, we use the multinomial coefficient:

$$f(m,s) = \binom{s+1}{m}_m$$

- With $s = 3$, this is equal to the tetrahedral numbers:

$$f(m) = \binom{4}{m}_m = \frac{(m+1)\cdot(m+2)\cdot(m+3)}{6}$$

- State space is at most $n \times k \times f(m)$ (polynomial in $n$, $k$, and $m$).

# Algorithm

**input:** $\mathit{value}$, $\mathit{runs}[k \times f(m)]$ (the state)
**output:** maximum score (given the input state)

**if** $\mathit{value} > n$ **then**
    **return** $0$
**end if**
**if** $\mathit{score}[\mathit{value}, \mathit{runs}] > -\infty$ **then**
    **return** $\mathit{score}[\mathit{value}, \mathit{runs}]$
**end if**

**for** $\mathit{runs}', \mathit{runscores} \in \textsc{makeRuns}(\mathit{runs})$ **do**
    $\mathit{groupscores} \leftarrow \textsc{totalGroupSize}(\mathit{hand} \setminus \mathit{runs}') \cdot \mathit{value}$
    $\mathit{result} \leftarrow \mathit{groupscores} + \mathit{runscores} + \textsc{maxScore}(\mathit{value} + 1, \mathit{runs}')$
    $\mathit{score}[\mathit{value}, \mathit{runs}] \leftarrow \max(\mathit{result}, \mathit{score}[\mathit{value}, \mathit{runs}])$
**end for**

**return** $\mathit{score}[\mathit{value}, \mathit{runs}]$

# Complexity

Universiteit Leiden

- Traditional Rummikub puzzle with fixed $m$ and $k$: solvable in $O(n)$ time (polynomial in $n$).
- Rummikub state space is at most of size $O(n \cdot k \cdot m^4)$ (polynomial in all input parameters $n$, $m$ and $k$).
- Rummikub puzzle is solvable in $O(n \cdot m^4)$ (polynomial with input parameters $n$ and $m$).
- Our algorithm for solving Rummikub puzzle runs in $O(n \cdot (m^4)^k)$ time (not polynomial in all input parameters $n$, $m$ and $k$).

# From the puzzle to the Rummikub game

Universiteit Leiden

- Table constraint
  - Some tiles must be in the construction
- Jokers
  - Additional factor in memory
  - More options to make groups
- Multi-player aspect unaddressed

# Counting winning hands

- Aspects in human play
- Number of winning hands
- Decide when to get rid of tiles

# Counting winning hands

Game finishes in one move

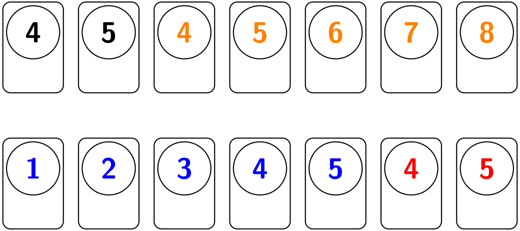
- Number of deals:

$$h(n, m, k, t) = \binom{n \cdot k}{t}_m$$

- Where $t$ is the number of tiles in hand
- For $m = 2$: Trinomial coefficient
- Real game ($t = 14$): 37,418,772,170,780
- How many of these games are winning?

# Counting winning hands

Example:



# Counting winning hands

- Roughly 10,000,000 hands are winning
- 0.0000273%

# Counting winning hands

Universiteit Leiden

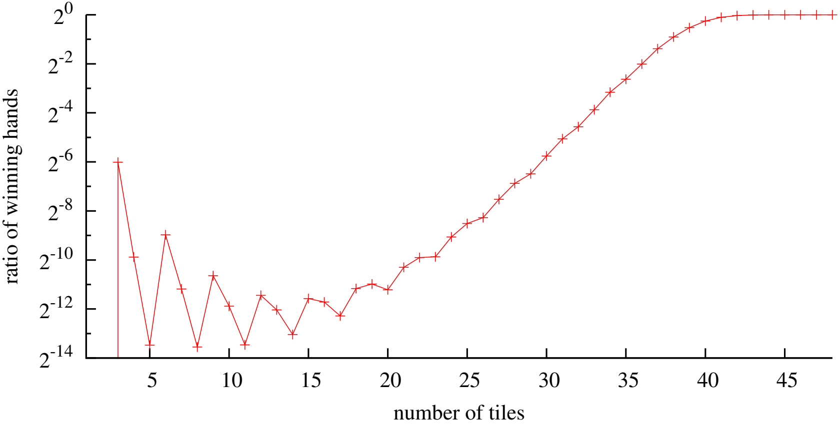
- Roughly 10,000,000 hands are winning
- 0.0000273%
- Hand of 15 tiles:
- Chances increase to 0.0000509%

# Counting winning hands

- Roughly 10,000,000 hands are winning
- 0.0000273%
- Hand of 15 tiles:
- Chances increase to 0.0000509%
- Hand of 16 tiles: 0.0000302%
- Hand of 17 tiles: 0.0000137%
- Hand of 18 tiles: 0.0000198%
- Period of 3

# Counting winning hands

Small instance of the game:



# Conclusion

- Solving the standard Rummikub puzzle is easier than sorting an array.
- We proposed an algorithm for the Rummikub puzzle that is polynomial in the numbers $n$ and duplicates $m$.
- Generalizing over the number of suits $k$ may make the problem harder.
- Complexity of the generalized Rummikub puzzle with parameters $n$, $m$ and $k$ remains an open problem.
- Future work: multi-player game

# Thank you for your attention!

Questions?