Executive summary

- Introduction (Situation)
    - Honestly, as pgm's gewerk het het ons al 20 jaar terug go gewen. Die potential is endless. Kyk waar dit klaar gebruik word, en imagine ons kan dit in die future kan gebruik. Use cases… kry potensial… hoekom is dit useful… as ons dit met 30% push het dit die potensiaal om te "tender" vir plekke waar CSPs gebruik word. Verder kan die graphing struktuur sterker gemaak word om te verseker dat meer application dit as `n drop-in solver gebruik.
    - Mobile robot estimate position, synthesize signals
    - Solve the problem of writing special-purpose programs for every domain or every query that needs to be answered
    - Extract solvers from one solution and apply it to others
    - Declarative representations model-based method for encoding our knowledge -> algorithms and structural stuff can then query the system. (Example is medical diagnostics). Simple and intuitive interpretation of the structures of probabilistic models can be used to motivate and design new models.
    - Graphical models provide additional insights into the properties of the model, including the conditional independence properties.
    - Probability theory calculus is used magnificently to answer these queries
    - Probabilistic Graphical Models for Fault Diagnosis in Complex Systems
    - Potensial: PGMs can make use of better graph structures to obtain more accurate results. -> cluster graphs -> purge-and-merge
    - Potensial: PGMs can make use of better discrete table representations in order to save on memory and complexity
    - Potensial: PGMs can make use of better inference techniques to better whatever
    - If we push PGMs with sparse tables & cluster graphs, we can use it as a drop-in replacement for some integrate problems.
    - If we add purge and merge, PGMs can incorporate CSP structures in their problem solving.
    - If we ensure additional inference logic a, b, c, and d, we can solve x, y, z
    - Original contributions
        - Paper 1 3D reconstruction using PGMs
        - Paper 2 cluster vs. factor
        - Paper 3 Land covering data
        - Paper 4 CSPs
        - LTRIP used by
            - Using probabilistic graphical models to detect dynamic objects for mobile robots *D Brink 2016*
            - Satellite image classification in *Practical Probabilistic Systems… McGregor 2020* in building higher-dimensional cluster graphs from Markov models
            - 3D structure building in *Triangular Mesh Mapping (paper and PhD.) Lombard* PGM approach using cluster graphs
            - Multiple Object Tracking *Multip… Louw 2018*
        - Sparse approach allowing larger CSP representations
        - Purge-and-merge solving larger CSP representations

        - Puzzle solving database...
- Complication (Before I came/ problem statement)
    - Why Simon vir PGM's (soos wat is die current situation van pgms, waar is die foute, wat is die improvements wat kort, waar moet ons research doen in general)
    - Nie `n lys van alle foute wat jy gaan solve nie. Bv, kan wees "ons moet die spoed van dit in N dimensies verbeter - en dan het jy bietjie gehelp daarmee of vir tot en met n=10 dit verbeter?" Bou `n paar verskillende plekke op waar daar foute is, en wys uit hoe jy iets daaromtrent gedoen het, mag iets kleins wees, die onus is nie op ons om al die probleme te gesolve het nie.
    - (Gewoonlik in lit study)
    - Practical PGMs structures are limited and hinder the speed etc.
    - Discrete table implementations aimed at PGMs are dense, this isn't efficient and causes memory issues. This also blocks the world of discrete sparse tables and implementations thereof like CSPs etc.
    - Loopy belief propagation and other loopy techniques are limited to heuristical usage -> cannot assure exact solution/outcome.
    - We want to come in and provide some solutions
        - Use a cluster graph algorithm to make better graphs
        - Define proper sparse table definitions to accommodate factor sizes currently out of reach
        - Provide methodology for modeling CSP problems in PGMs
        - Use a graph building and nudging approach to use current inference to achieve exact results. Less trade-off, more power to you
    - See the full literature study to read up on all of this
- Approach (this is me/ oplossing/ soos bitcoin-whitepaper)
    - Baie uit jou papers of wat jy create het.
    - Methodology -> product realisation
    - Horizontel view (per function) of vertical view (per product)
    - These are my products:
        - Cluster graphs and the RIP problem
        - Sparse tables
        - Factor clustering
        - Purge and merge
        - CSP inference model
        - Puzzle solvers
- Results (jou stats)
    - Hier is hoe baie produk A gehelp het
    - Dit is wat produk B kan doen
    - Clustergraphs vs. factorgraphs (way beter bewyse as laas)
    - Sudoku solving
    - Google OR-tools vs. us (they win)
    - ACE vs. us (we win)
    - Full puzzle dataset
- Conclusion / so what (hoekom is die results so wonderlik vir PGMs)
    - Nou is PGM's die mees betroubaarste vorm van masjien leer daarbuite. Your welcome. Nou met hierdie wen ons hierdie grond.

    - The RIP problem is solved, this enables people to get more accuracy out of their PGMs -> use less memory -> basically do cool stuff. This is evident in other research that used it their PGM structures
        - Cite papers that uses this algorithm
    - Additionally to the graph structure, we focus on sparse discrete tables, which allows modeling CSPs. As a result, we won ground in A, B, C.
    - The final contribution is the purge-and-merge inference approach. This allows work we successfully used these graph mutability to develop a further approach purge-and-merge
    - We can now reach exact inference in CSPs using "loopy graphs" advancing research in Discrete PGMs and constraint interactions.
    - We can use existing loopy implementations to achieve exact inference

Mirror Executive summary but with with a bunch of fluff

Literature study

- Volle verduidelikking vir iemand wat nie pgm ken nie/nie glo jy het geresearch nie
- PGMs
- CSP
- Their interaction
- Limitations
- Factor graph
- Cluster graph
    - Cite the hell out of everybody in Stellenbosch that uses LTRIP
    - Even explain their stuff in excruciating detail
- Factor operations
- Loopy belief update / propagation
- Graph settings / choices
- Junction tree
- Structure learning
- Purge and merge similarities