CONOP Optimization on HANA

Orlando Ding

August 23rd, 2022

Agenda

- **Research background**
  - Ordinal timeline of fossils
- **Algorithm model**
  - CONOP
  - Complexity analysis
- **Performance evaluation and optimization**
  - Optimization for sequential version
  - Optimization via parallelization
  - Optimization results
- **Conclusion**
  - HANA-CONOP application
  - HANA-CONOP extension
- **Appendix**

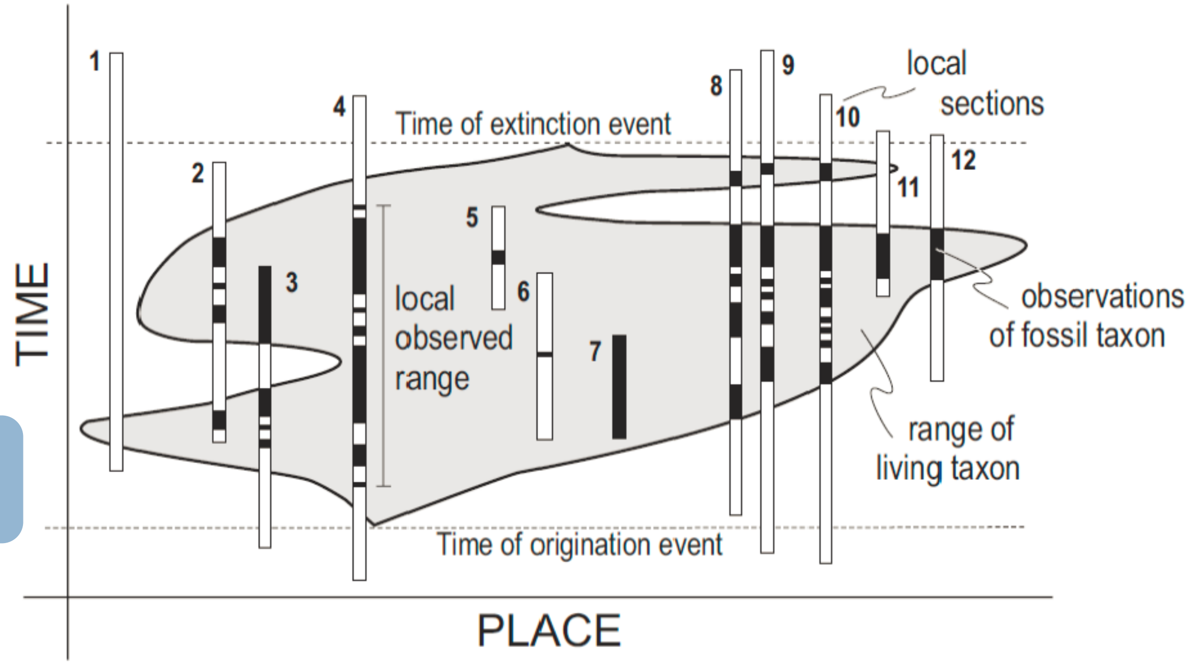
Research background

Domain background
Biostratigraphy

- What is Biostratigraphy

❖ Biostratigraphy the branch of stratigraphy which focuses on correlating and assigning relative ages of rock strata by using the fossil assemblages contained within them.[1] The primary objective of biostratigraphy is correlation, demonstrating that a particular horizon in one geological section represents the same period of time as another horizon at a different section.

Fossil events sequence → Ordinal timeline of species → Correlation of geological sections

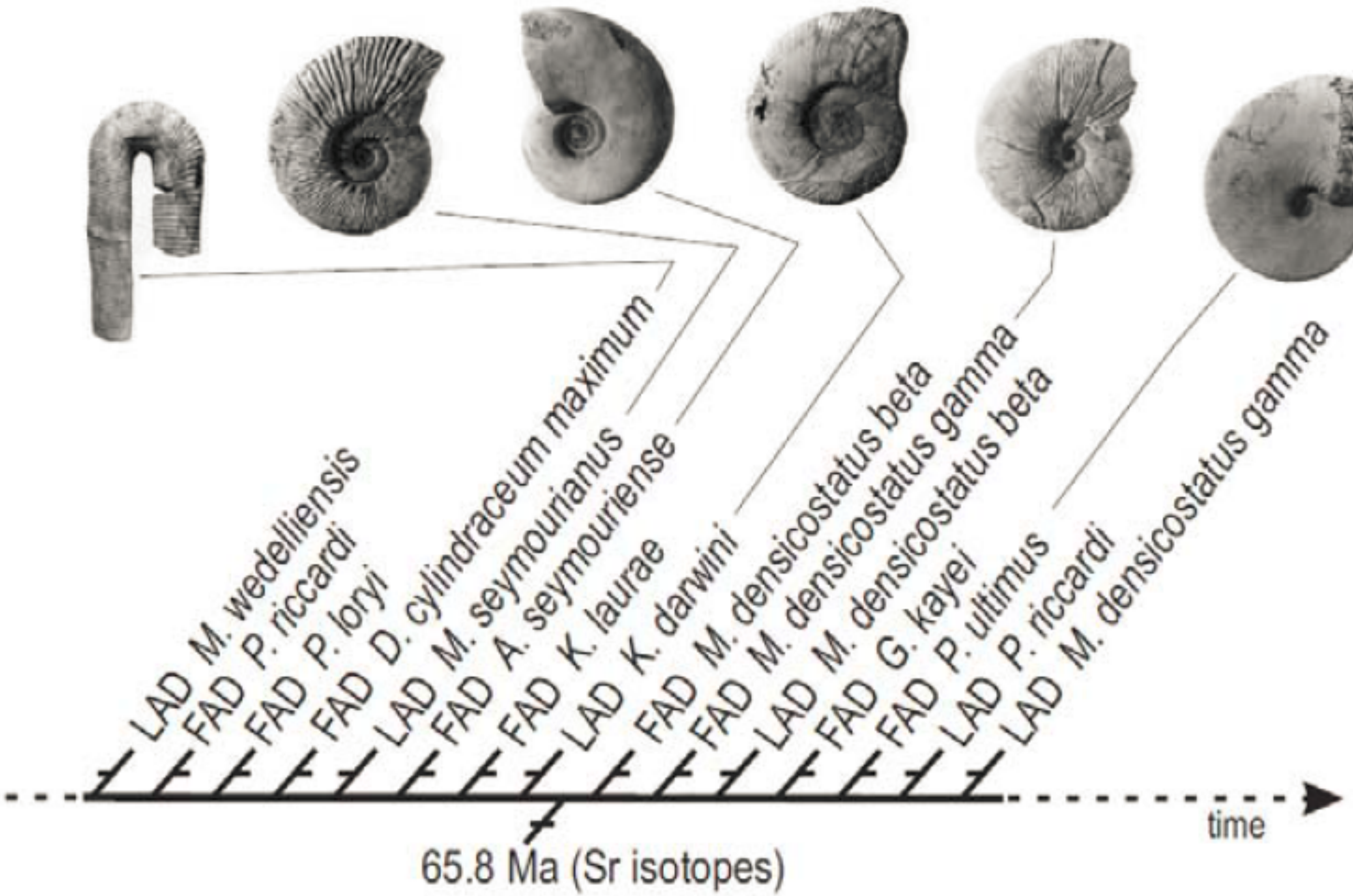


- Values of Biostratigraphy

❖ Produce fossil event sequence and relevant ordinal timeline

❖ Reflect the evolution history of the Earth and provide time measures for other relevant geological research

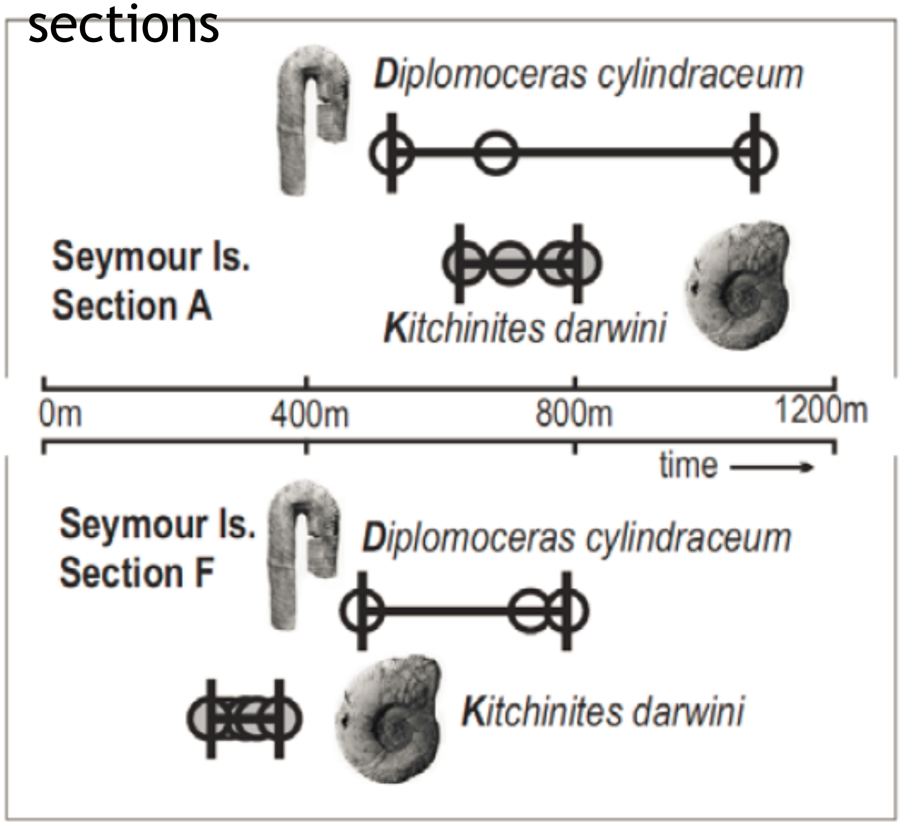
Domain background
Ordinal timeline of fossils



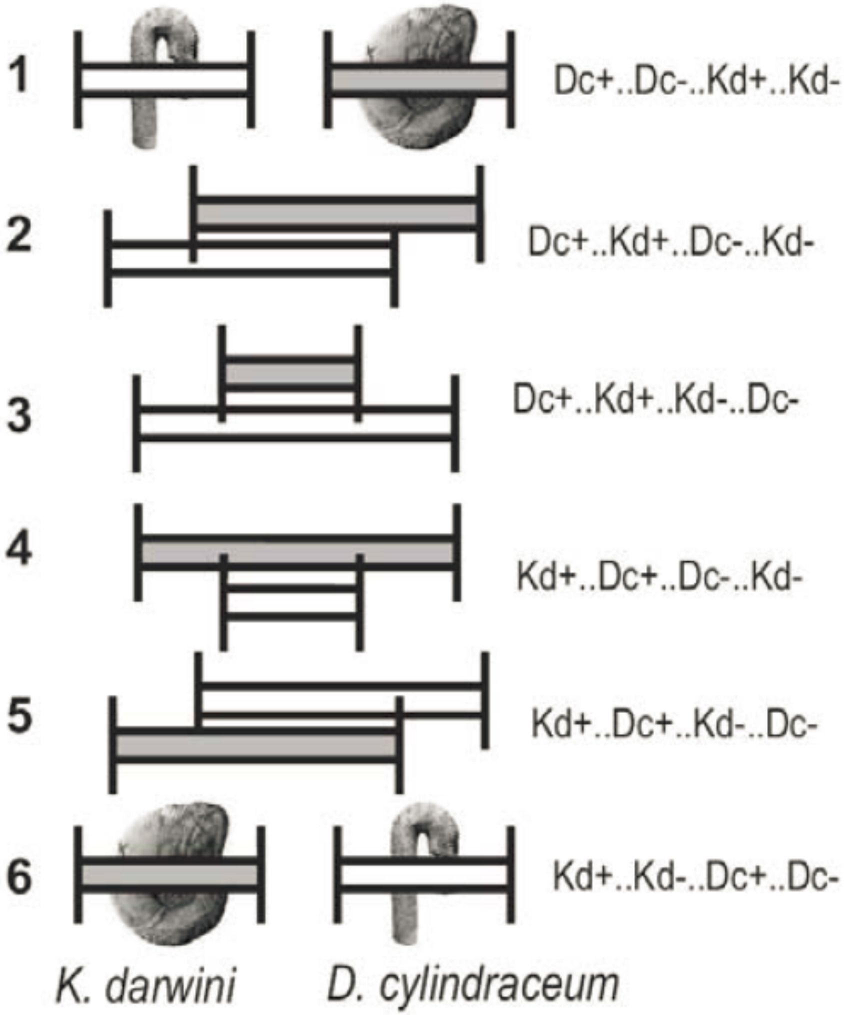
Ordinal timeline with ammonite range-end events and dated events

Domain background
Fossil Serialization

- Fossil samples in different geological sections



- Range charts for two shared ammonite taxa A and F in two sections from Seymour Island, Antarctic.



Kd: Kitchinites darwini (Species Name)
Dc: Diplomoceras cylindraceum (Species Name);
"+": The first appearance time of species
"-": The last appearance time of species

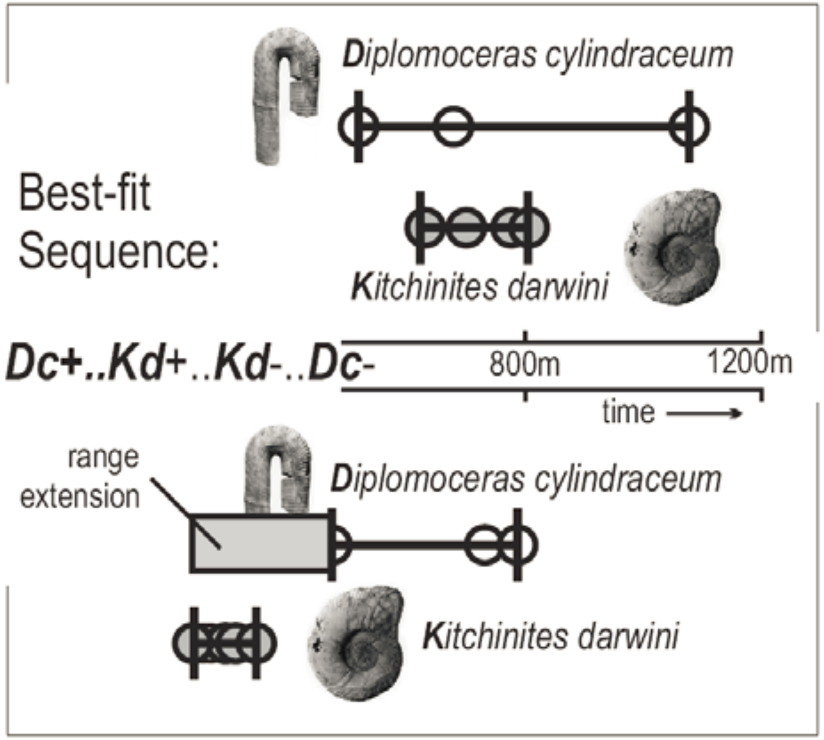
The picture above indicates 6 possible permutations of the 2 taxa **P(4)/(C(2)*C(2))**

Domain background
Fossil Serialization

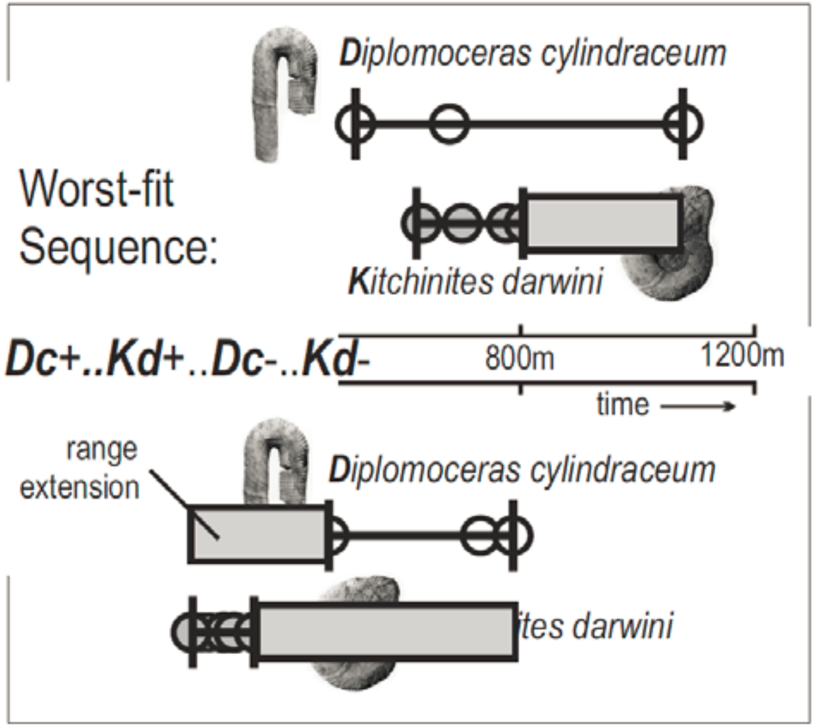
- Sequence estimate after event adjustment

Basic adjustments → Adjustments in CONOP

Comparison penalty/loss after distance adjustment



Generate fossil events sequence from observed sample, **whose measure of distance adjustment is the least among all 6 possible sequences - Best-fit sequence.**



Generate fossil events sequence from observed sample, **whose measure of distance adjustment is the largest among all 6 possible sequences - Worst-fit sequence.**

Domain background
CONOP performance

- Nowadays scientists still can't construct a comprehensive timeline including all fossil first appearance and last disappearance events, due to the following three reasons:
  1. Data volume, esp. the size of geological sections and relevant fossil records
  2. Algorithm complexity of CONOP
  3. Application complexity of CONOP that leads to no-convex restriction in algorithm
- CONOP performance:

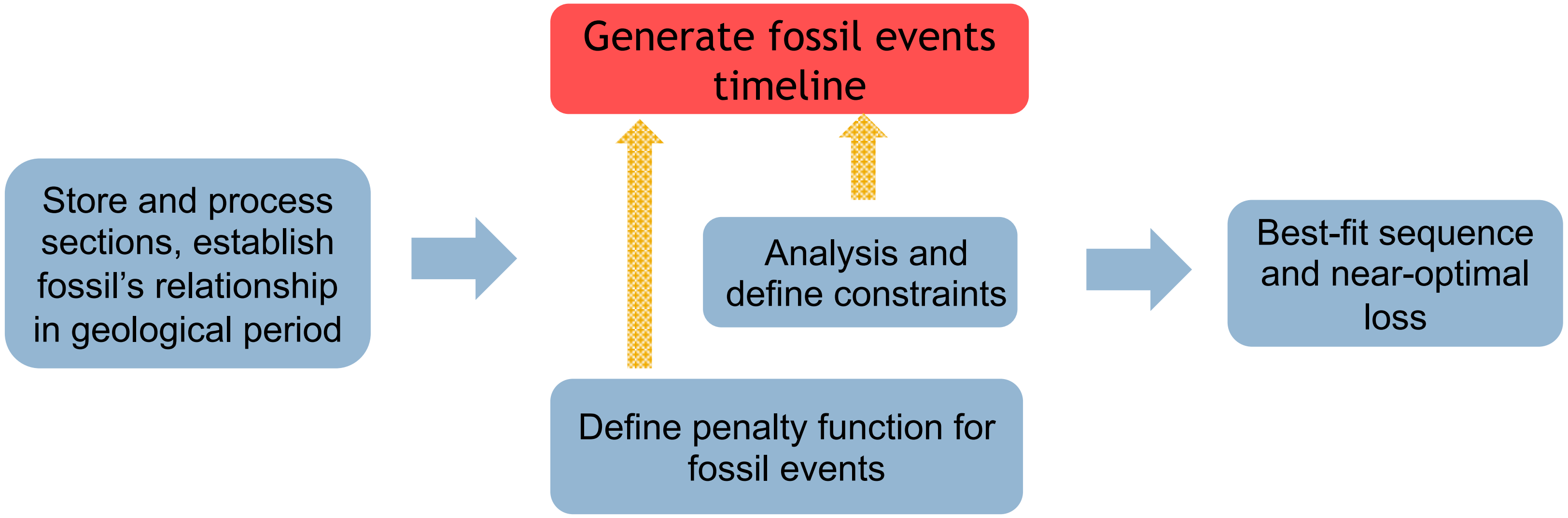
Data volume	Time
Small-size dataset(7 sections, 62 species, 402 fossil records)	7 seconds
Middle-size dataset(195 sections,1365 species,12,212 fossil records)	3 hours
Large-size dataset(287 sections, 7000+ species, 1,000,000+fossil records)	6+ days

Algorithm model

Algorithm model
Abstraction

- CONOP: a program is used to generate a near-optimal fossil events timeline based on geological section samples, which is optimized by a penalty function given **biostratigraphy** and non-**biostratigraphy** restrictions. Meanwhile, it also supports different calculation and validation.

- CONOP deals with:
  - Store and process information of geological sections and establish their correlation
  - Generate and adjust fossil events timeline based on sections' information
  - Discover constraints based on fossil records and non-paleobiologic events
  - Define penalty function for specific fossil events sequence or parts of the sequence



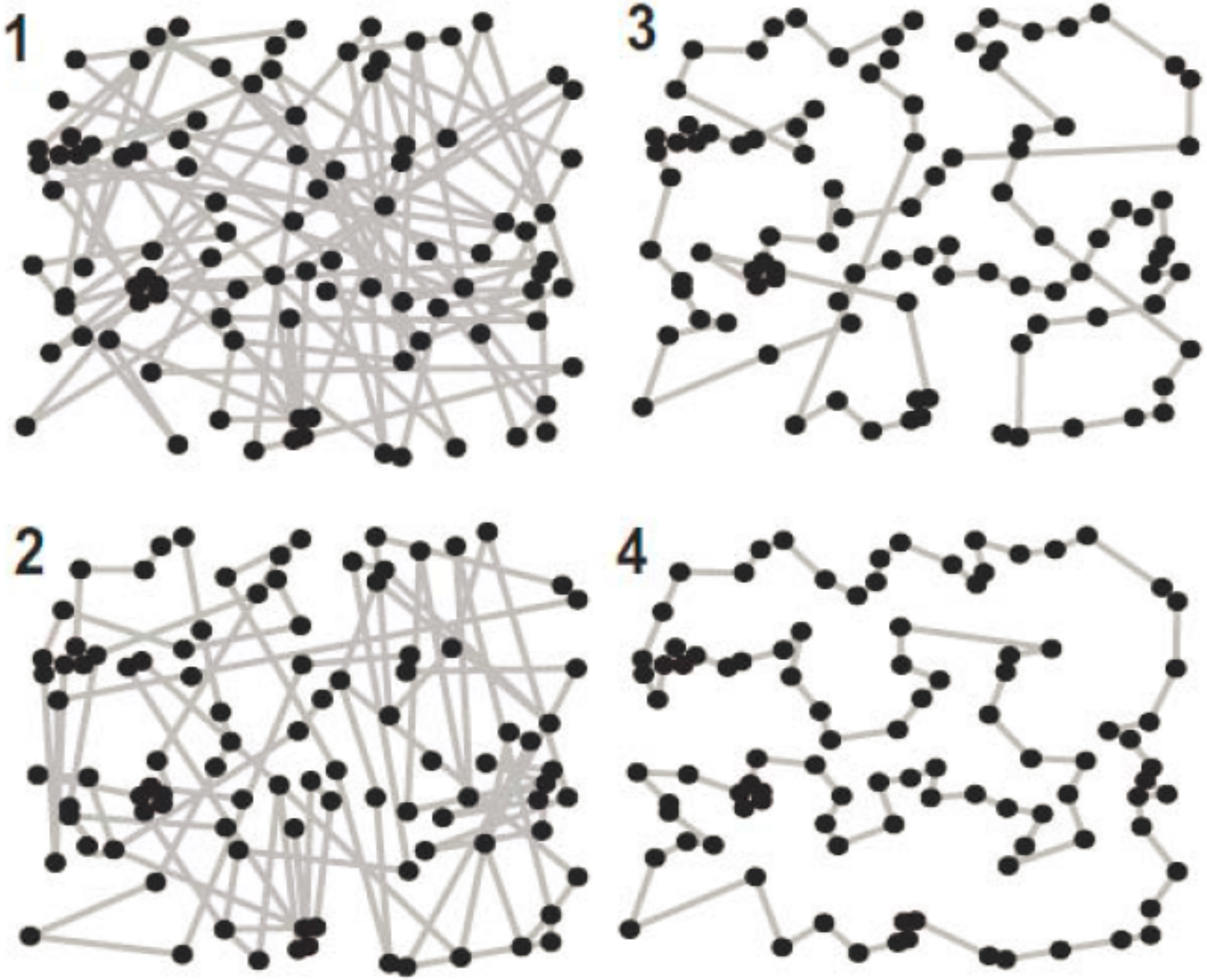
Algorithm model
CONOP-Travelling Salesman Problem

- CONOP Algorithm category: Travelling Salesman Problem (TSP) with restrictions, a kind of NP-complete problem

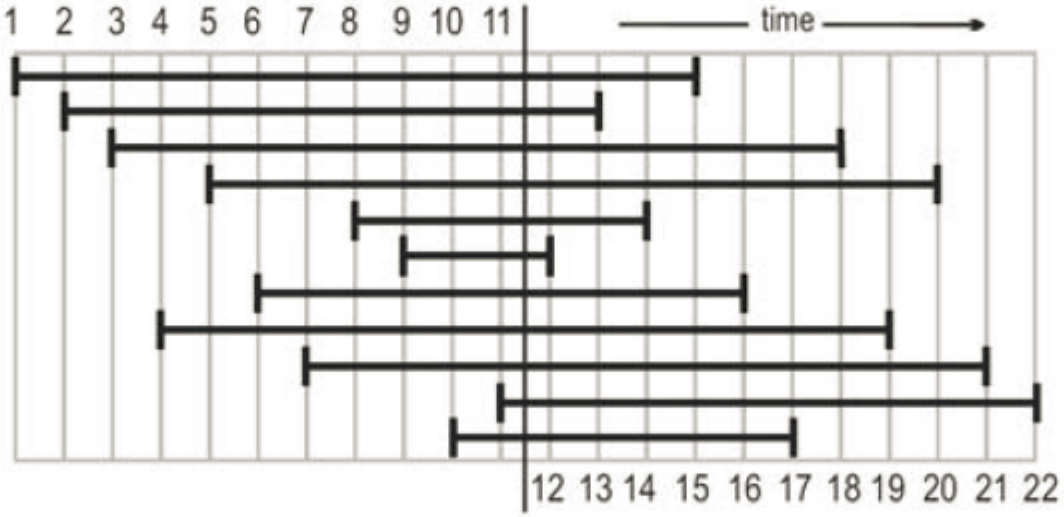
- The traveling salesman problem (TSP) asks the following question: given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

- Paleobiologic time-line problem as TSP
  - Range-end events → Cities
  - Net range adjustment → Travel distance

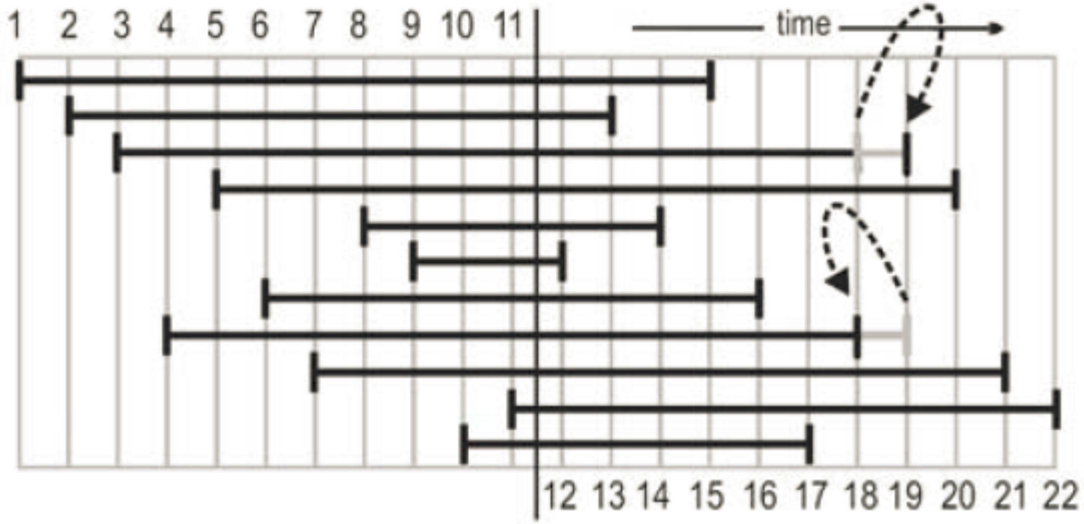
- Solution: choose a random seed of fossil serialization, then use heuristic strategy to optimize events based on current penality/loss, which is **an adjustive sorting model**(Compared with **the generative sorting model**, CONOP can't resort to branch-pruning restrictions, such as $\alpha, \beta, A*$ pruning)

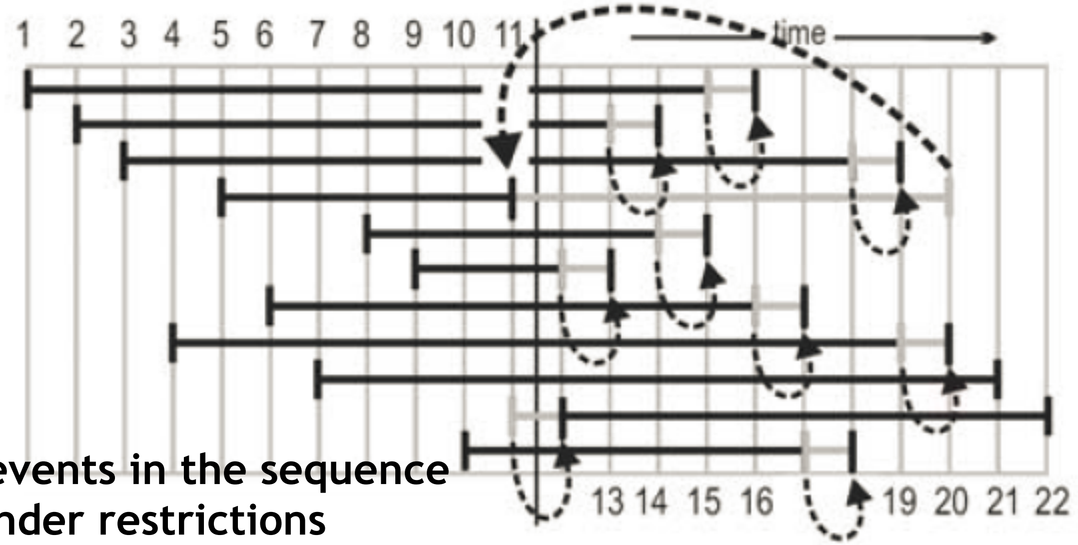


Algorithm model
CONOP - How to figure out a better solution



- Initialize sequence





Adjust events in the sequence
under restrictions

Algorithm model
CONOP-Simulated Annealing

- Simulated Annealing (SA) is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space.

❖ More efficient than exhaustive enumeration for NP problems

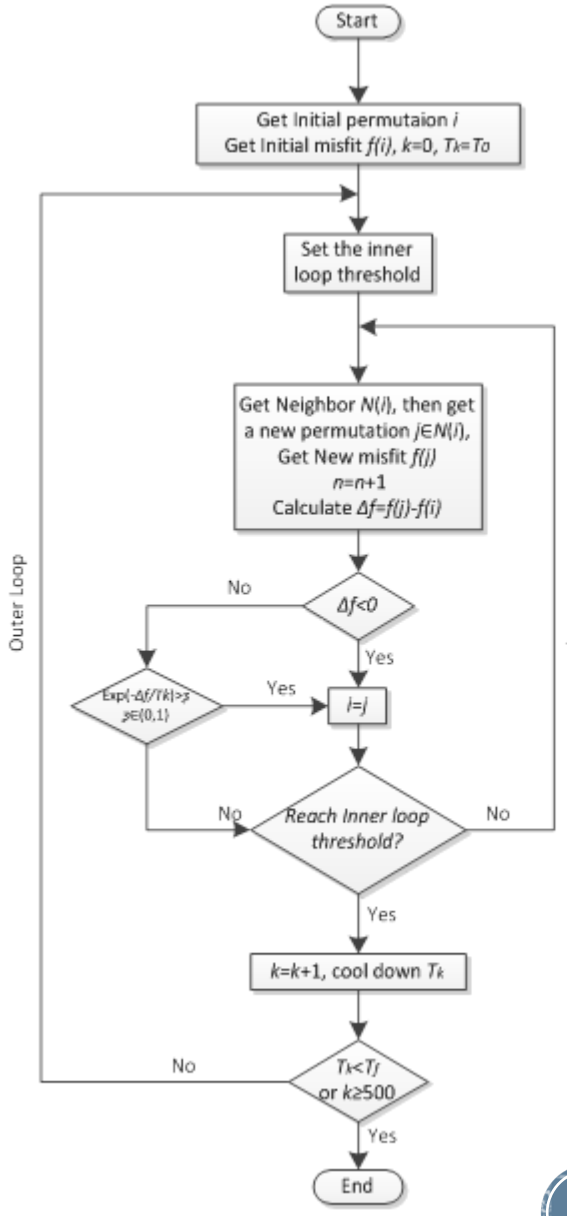
❖ Avoid steep steps to search global optimal

❖ Imposes almost no limits on the mathematical properties of the fitness formulations and constraints

- As a general algorithm to find out near-optimal solutions for NP/NPC problems, it is applicable for almost every area:

❖ Resource Allocation Plan

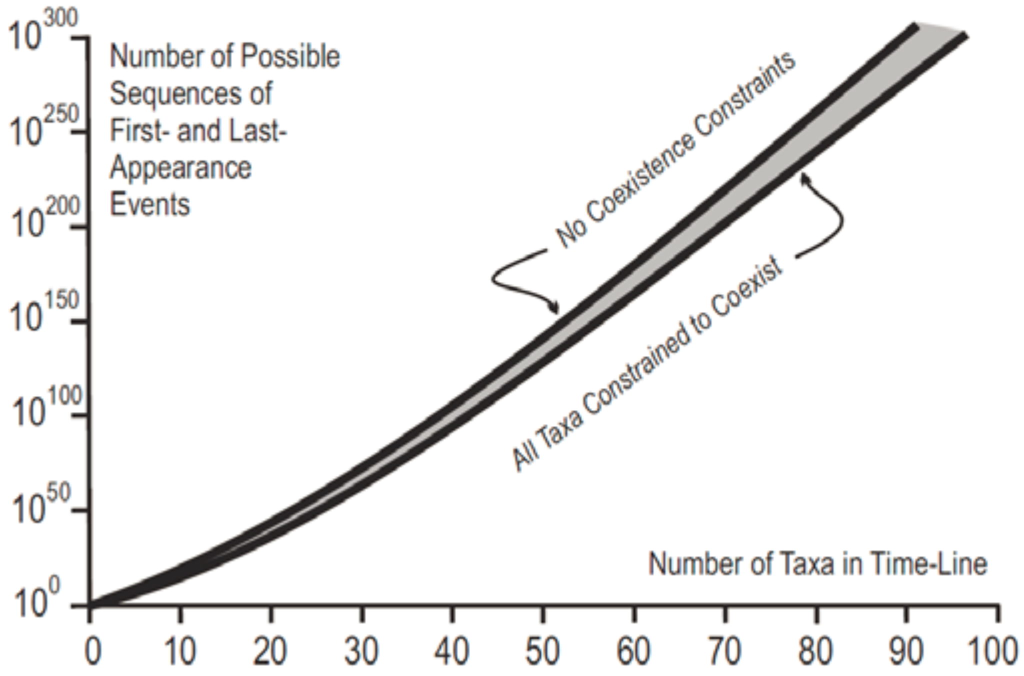
❖ Investment Portfolio Design



Algorithm model
CONOP computational complexity

Number of Taxa:	1	2	3	4
Number of Possible Time-line Sequences:	1	6	90	2,520

5	6	7
113,400	7,484,400	681,080,400

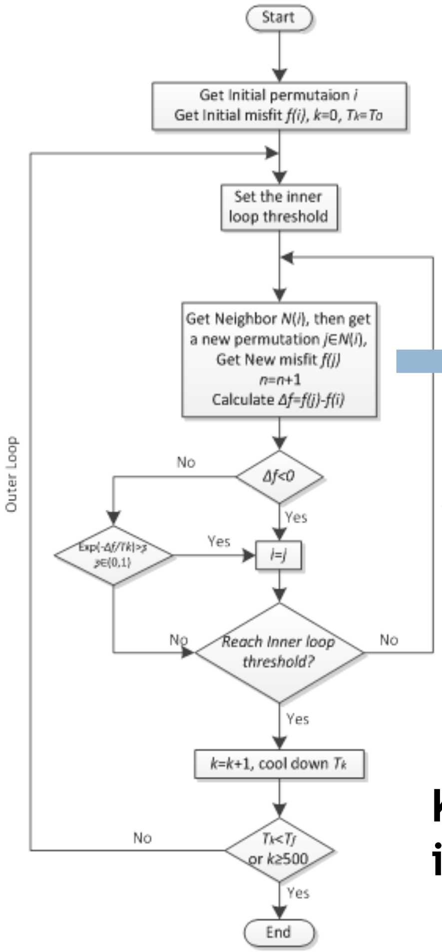


- Computational complexity under co-existence constraints: $\mathbf{O}(\mathbf{2n-1})!$
- n: number of taxa(fossil records)

Performance evaluation and optimization

Algorithm optimization
Optimization for sequential version

- **Comparison between theorical estimate and actual performance**
  - Theorical analysis



Neighbor N(i) - O(n^2)

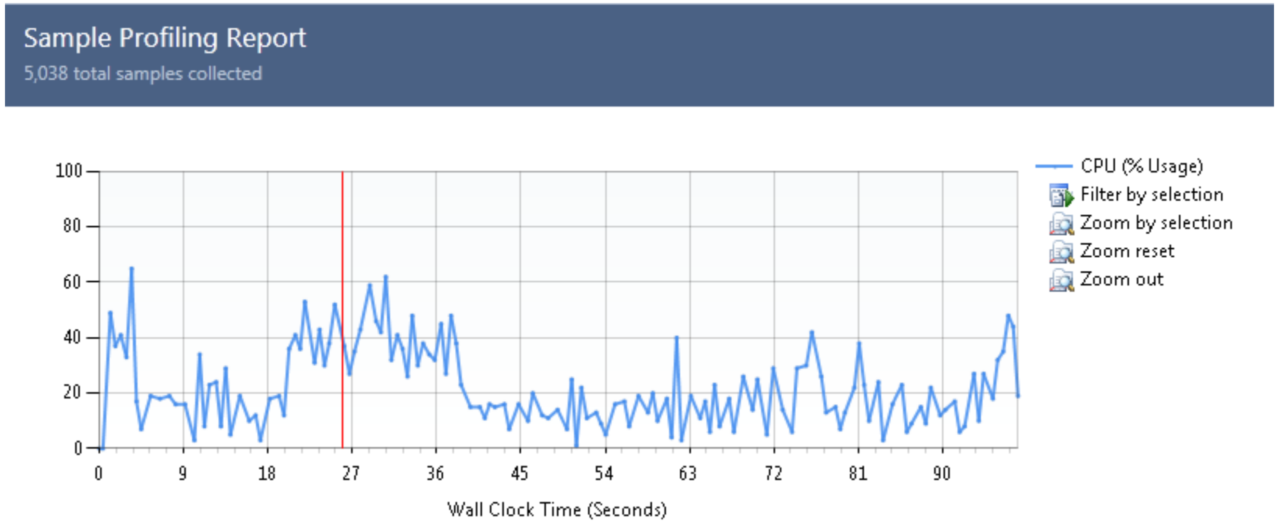
New permutation - O(n)

New misfit - O(n^3)

Calculate Δf and adjustment - O(n^2)

Key: adjust implementation to make it consistent with theorical estimate

- Actual performance



Hot Path

The most expensive call path based on sample counts

Function Name	Inclusive Samples %	Exclusive Samples %
AFLNjFossil::TestNjFossil::test2(void)	99.86	0.00
AFLNjFossil::testSAMain(void)	99.86	0.00
AFLNjFossil::SimulateAnneal_Sequence(struct AFLNjFossil::TPL_SA_GLB_PARA &,st...	99.56	0.14
AFLNjFossil::getNwPen(int,int,class ltt::vector<int> const &,struct AFLNjFossil::...	90.41	2.12
AFLNjFossil::getSctPenDPV2(int,int,class ltt::vector<int> &,struct AFLNjF...	**81.84**	**68.60**

Related Views: Call Tree Functions

Algorithm optimization
Optimization for sequential version

- HANA-CONOP: optimization for **input data and auxiliary data structures**
- HANA-CONOP: **optimization for memory and CPU cache**
  - Adjust and optimize memory-accessing approaches[multi-dimensional array, pointer array, etc.]
  - Analyze and optimize CPU cache-hitting rate
- **Mathematical model: optimization for the incremental adjustment given continual non-convex functions**
  - Extract shared $O(n^3)$ factors to avoid duplicate calculations
  - Estimate the result of $O(n^3)$ functor to prune branch in advance

Algorithm optimization
Heuristic speedup

- When to start parallelization strategy - **Heuristic speedup by parallelization**
  - Comparison between parallelization speedup and synchronization delay during runtime – If and only if the former is larger than the latter, we will trigger parallelization

  - Heuristic speedup by parallelization

    - Assumption: given specific hardware and HANA parallelization settings, the characteristic of **speed-up curve via parallelization can keep stable.** Therefore, it’s possible to learn relationship between speedup and input data(species, sections), then utilize such approximate functor to determine if parallelization option is needed to switch on

    - Prototype implementation

      1. Acquire speedup curve’s key control parameters in pre-processing

      2. In HANA-CONOP implementation, estimate payoff between speedup benefit and synchronization delay and decide whether to switch on parallelization option

Algorithm optimization
Parallelization strategy

- **Parallelization version consistent with sequential version**
  - Analysis of speedup vs synchronization cost

Pseudo-code of sequential version

```
for(int i=0; i<OUTER_LOOP_COUNTER; i++){
    for(int j=0; j< INNER_LOOP_COUNTER; j++){
        independent_context  = independent_context_generation();
        for(int k=0; k < sizeof(independent_context); k++){
            independent_calculor(independent_context[k]);
        }
    } //inner loop
} //outer loop
```

- Adjustment of sequential version based on HANA parallelization Job API

Pseudo-code of parallelization version

```
Execution::JobContextHandle jch = initialize_job_context();
Execution::JobNodeHandle *hjobGroup = initialize_job_group(jch);
for(int i=0; i<OUTER_LOOP_COUNTER; i++){
    for(int j=0; j< INNER_LOOP_COUNTER; j++){
        independent_context  = independent_context_generation();
        for(int k=0; k < sizeof(independent_context); k++)
            add_into_jobNode(hjobGroup[k], independent_context[k]);
        jch->startExecution();
        jch->wait();
    } //inner loop
} //outer loop
```

Parallelization payload(inversely proportional with speedup)

Parallelization synchronization payload

Algorithm optimization
Parallelization strategy1

- Consistent Model via Multiple Markov Chains

❖ Algorithm model

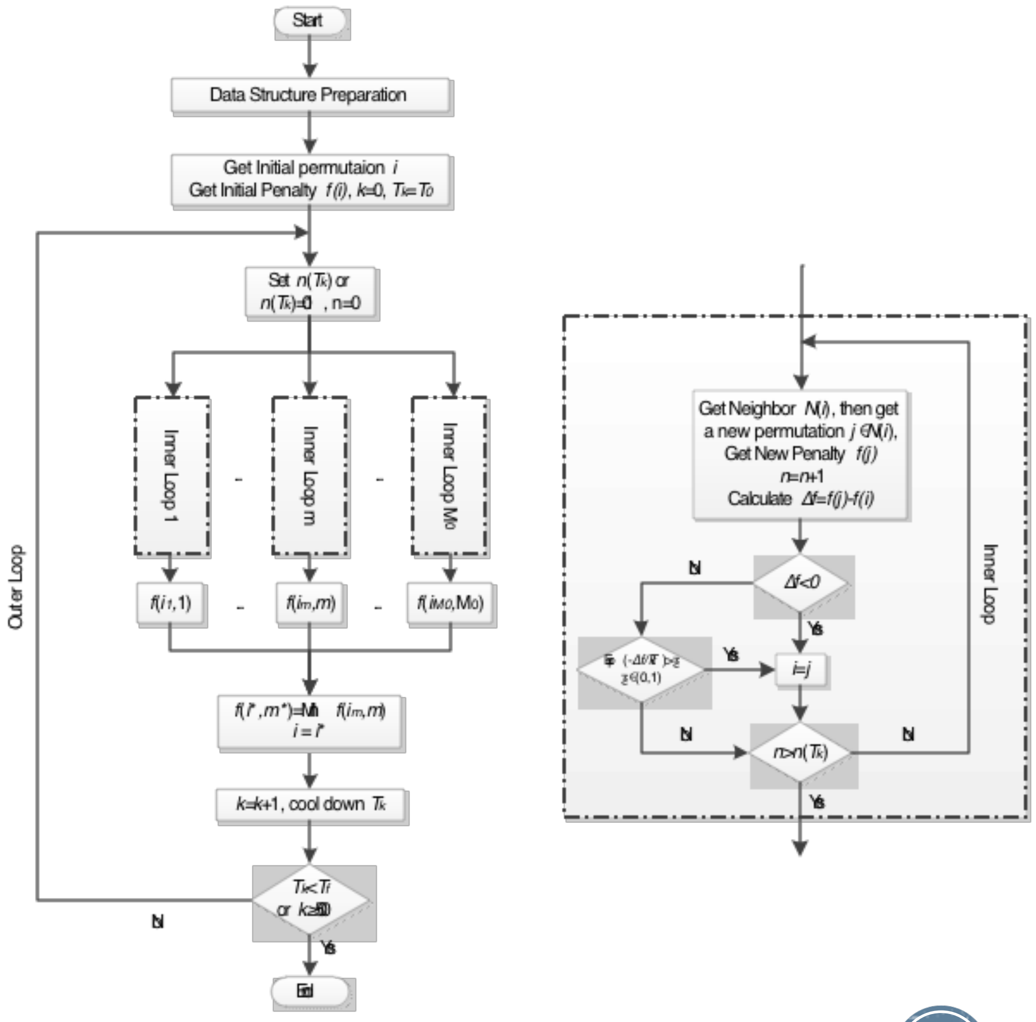
➢ Goal: trigger parallelization calculation for the functor in inner loop **under the fix evaluation context**, then acquire the best events sequence with the minimal penalty

➢ Implementation: add control logic for CONOP on HANA with reasonable parallelization thread number. This approach can guarantee the equivalent result as the sequential version

❖ Advantage and experimental result

➢ Fully utilized CPU and multi-threading on HANA platform

➢ The speedup ratio is proportional with the size of input data and the thread number (For the case of species number equal with 409, speed-up ratio is about 65)

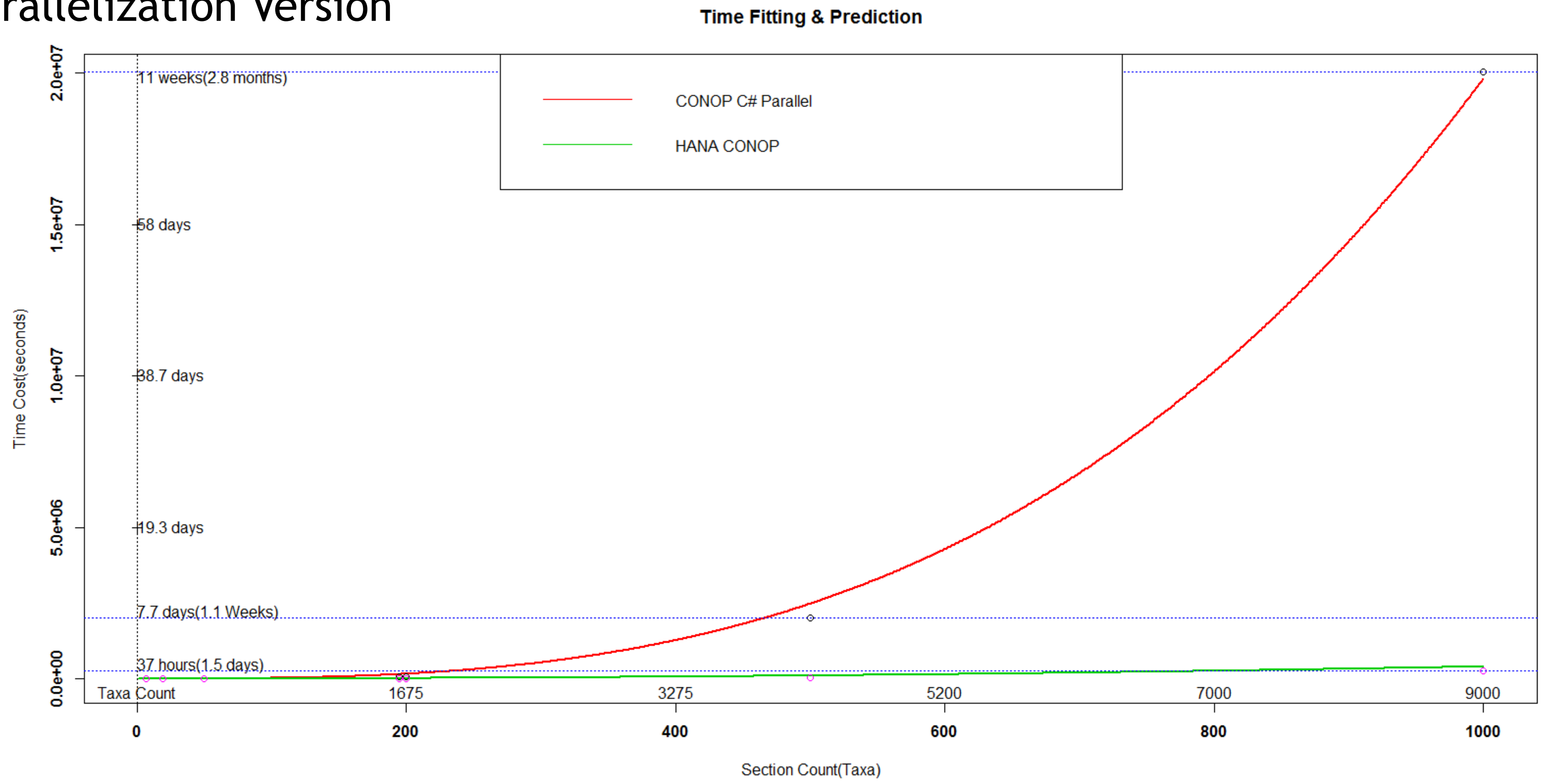


Algorithm optimization
Optimization result

❖ Performance comparison

CONOP C# Parallelization Version

HANA CONOP



Testing environment:

- 3 workstations
- 1 z820 server

Conclusion

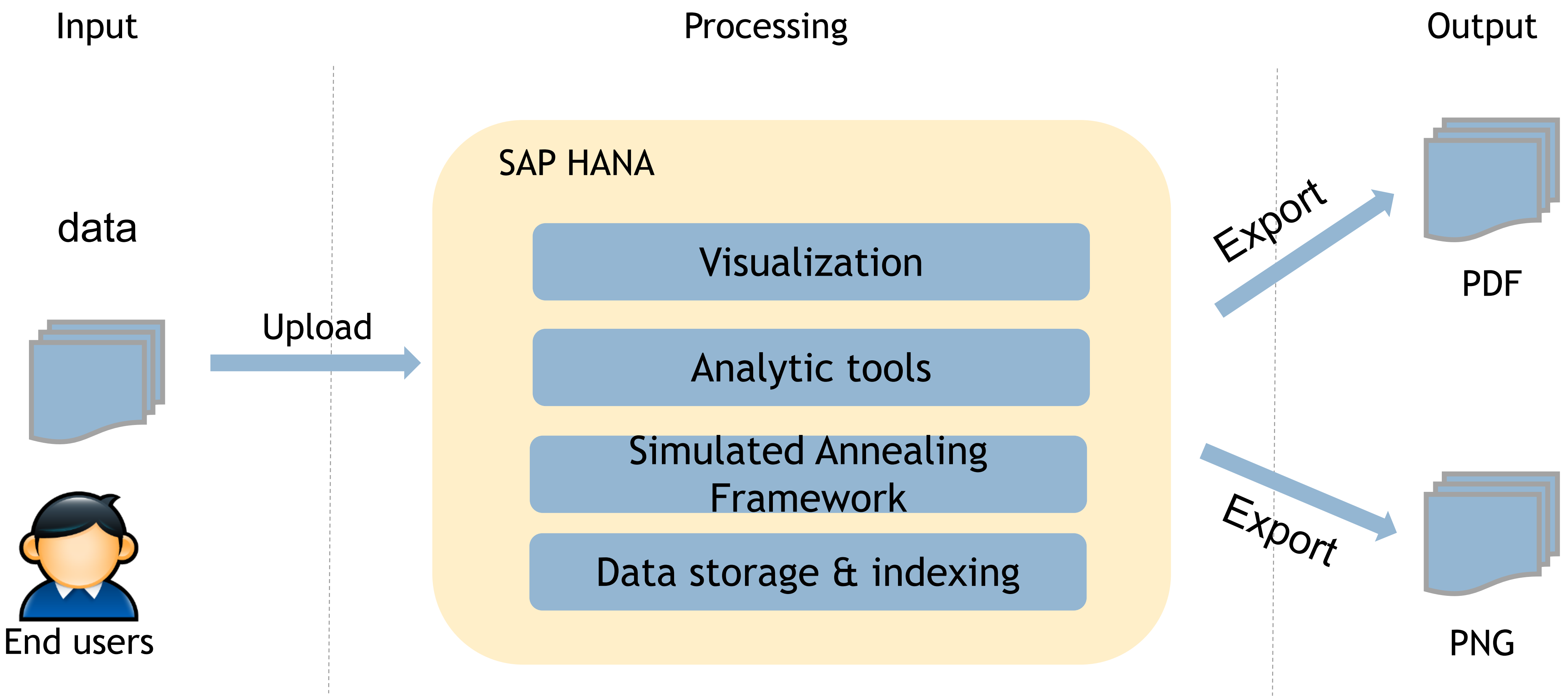
Conclusion
HANA-CONOP

- HANA-CONOP: an application fusing CONOP logic, algorithm optimization as well as heuristic parallelized simulated annealing framework

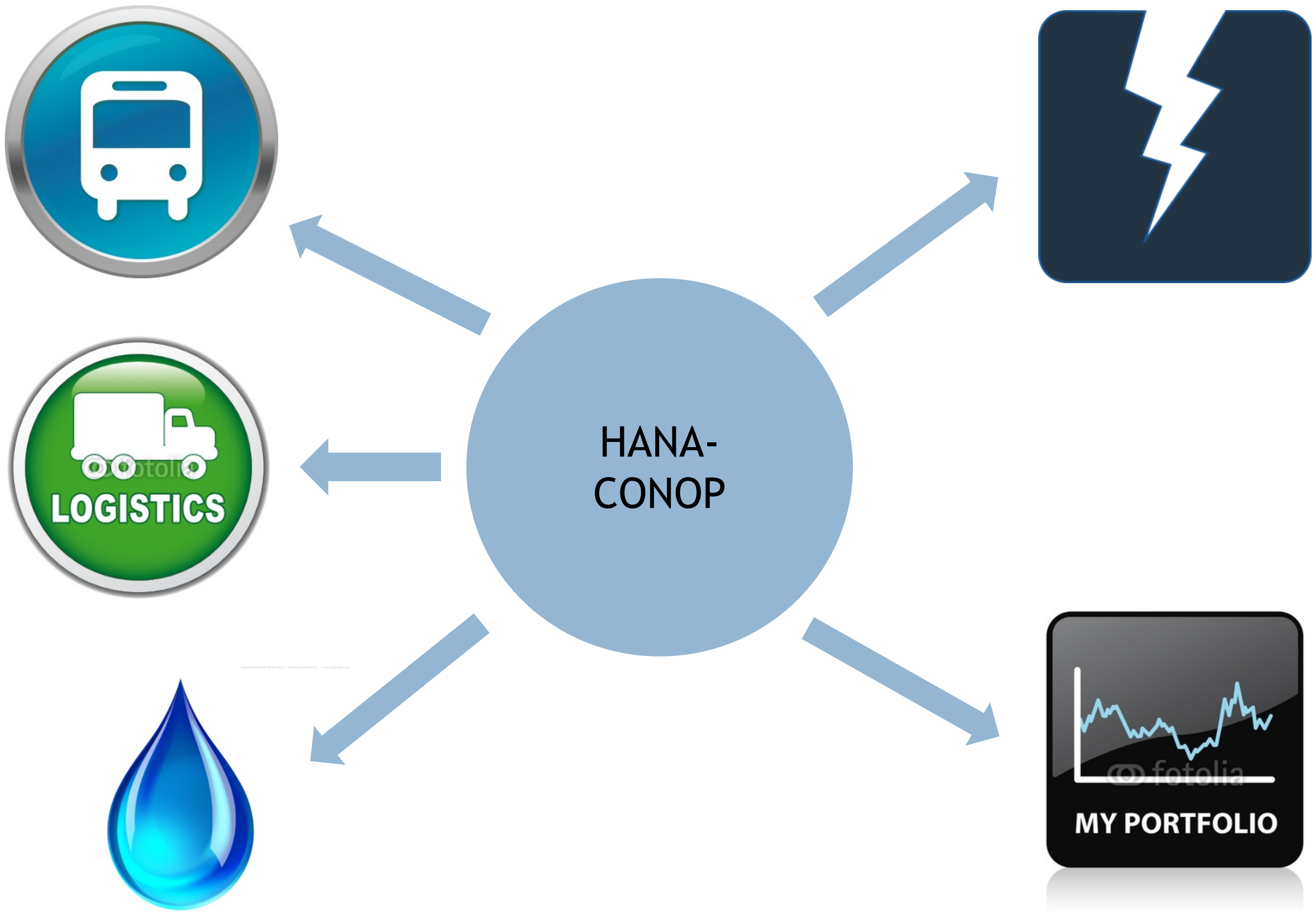
- HANA-CONOP fully leverages platform advantages
  1. Storage and analysis of fossil records
  2. Optimized data structure well adapted with in-memory computation
  3. Optimized for the simulated annealing algorithm based on CONOP case

- HANA-CONOP can help scientists
  1. Build up a more comprehensive fossil events sequence
  2. Support diversity research in the Earth science and paleontology
  3. Recognize effective bio-geological signals and filter “noisy” information
  4. Greatly improve the accuracy of geological period timeline, extending the confidence time duration to about 500, 000 years that well cover the whole phanerozoic

HANA-CONOP
A scientific research platform for paleontology



HANA-CONOP extension
Innovation vision



QA

Appendix

Domain background

CONstrained Optimization for events sequence(CONOP)

- Constraints: the most reliable, incontrovertible observations, such as co-existence.
  - Co-existence
  - The first appearance date(FAD) is always before the last appearance(LAD)
  - Non-paleobiologic events

- Penalty functions: all of the others, which are subject to adjustment, may be incorporated into measures of misfit.
  - Interval
  - Level
  - Eventual
  - ……

Domain background
CONOP implementation

- Current CONOP(CONOP9) implementation, besides simulated annealing framework, has already added more control and optimization options, in order to support more complex calculation pattern and more flexible validations:

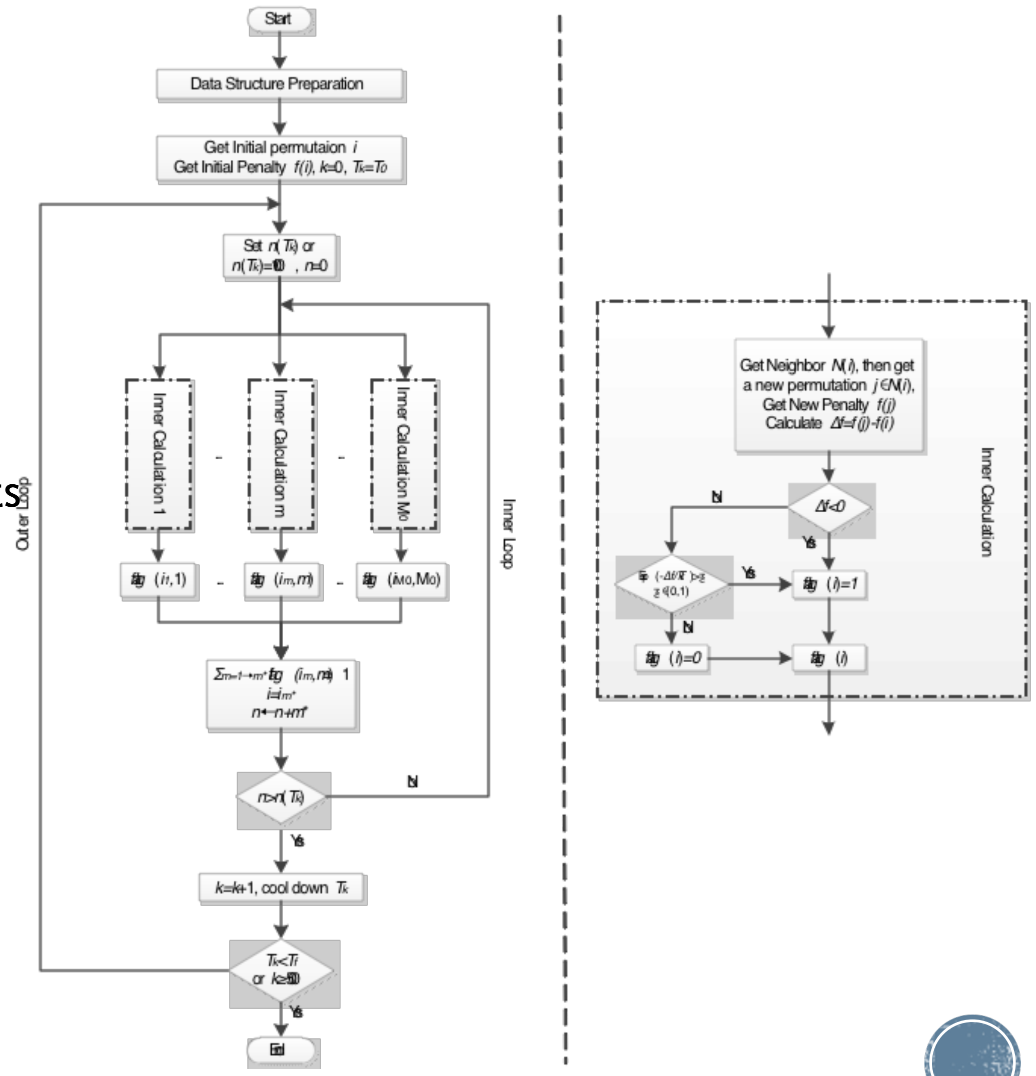
- Three mutation options of the timeline for faster search of near-optimal solutions
- Several significantly different options for measuring the misfit between the timeline and the data
- Adding Composite Timelines to the CONMAN9 database as New Sections for a better validation
- CONOP RUN-CONFIGURATION FILE (CONOP9.CFG): 74 configuration items, which increases algorithm's flexibility as well as complexity

Algorithm optimization
Parallelization strategy2

- Inconsistent parallelization model based on Multiple Random Trials

Algorithm model

- Goal: trigger parallelization for external loop and screen out candidate seed in nearby evaluation context
- Idea: consider a sub-procedure, includes getting a neighbor and calculating the new penalty, as a random trial, execute a bunch of sub-procedures in parallel, then synchronize the results such that its penalty is $\varepsilon - equivalent$ to the Sequential Simulated Annealing



Advantage and theoretical estimate

- Fully utilized CPU and multi-threading on HANA platform
- Speedup is proportional with the acceptance rate of random trials, synchronization cost of sub-procedure and threading number(The Boundary estimate is still on the way)

Thank you

- Contact:

- Lei Ding
- lding25@ucsc.edu