Statistical compiler tuning
Haneda, M.

Citation
Haneda, M. (2006, September 26). *Statistical compiler tuning. ASCI dissertation series*. Retrieved from https://hdl.handle.net/1887/4572

Propositions (Stellingen)
by Masayo Haneda, author of
Statistical Compiler Tuning

1. Non-parametric inferential statistics is suitable for analyzing the effect of compiler optimizations.

2. Default compiler settings use a number of compiler optimizations of which many are not essential.

3. Measuring the effect of each experimental factor is difficult. Comparing the measured effects is even more difficult. Explaining the reason for the differences is most difficult.

4. Random search is a very strong search algorithm to determine a suitable compiler setting for applications. Random search reaches sufficiently good results quickly compared with other search algorithms.

5. Current programming environments allow users too much freedom so that they can produce hundreds of variations of a code which are all semantically equivalent. This causes code optimization at compile time to be extremely hard. In fact, we are in the classical era of programming and we still need an abstract programming paradigm, which allows uniform representation whilst still being pragmatic.

6. We might say that a machine can think if it can automatically determine the correct use of articles "a" and "the".

7. Although architectures are constantly extended with new features to enhance the performance of software, software each time has to be adapted to maximally exploit these new features.

8. In some cases, it is nonsense to discuss the demand for a certain new technology since the technology itself can create demand. For example, the TV facility on the mobile telephone has been deemed unnecessary. However, after its introduction in Japan people started using it and this facility is now in high demand.

9. Ten times one rose is more than a bouquet of 10 roses.

10. "What do you think about Dutch food?" is the most embarrassing question for foreigners.