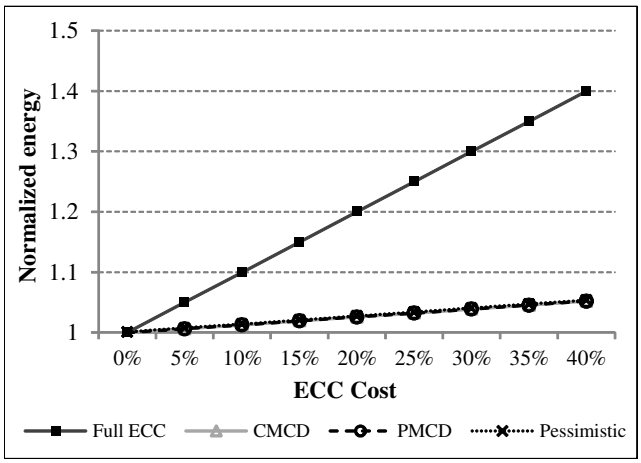

Figure 2: Energy consumption for different ECCs.

likely to hold for the CMCD because it performs a search lookahead, which can be an expensive operation, to correct a node's h-value. Nevertheless, it is reasonable to assume the ECC's coding and decoding operations to be similar to the overhead incurred by the lightweight Pessimistic and PMCD methods with respect to energy consumption.

We present the results in terms of normalized energy, defined as $NE = Y \times X$ + (1 - Y). Here, Y is the fraction of times that the search algorithm accesses ECC-protected memory during search, and X is the energy cost associated with the ECC scheme employed. For Full ECC $Y = 1$ as all memory is ECC protected, resulting in $NE = X$. For our approaches Y is measured empirically using the GEM5 simulator (**?**). We simulate an IDA* search in a core and a memory hierarchy of a typical desktop processor, with 32 KB of data and instruction caches, 256 KB of L2 cache per core and a shared 8 MB L3 cache.

Figure 2 presents the obtained results for a representative instance of topspin (the NE-values are similar in all domains tested), assuming an FPNE of 10^{-3} (the results were nearly identical for other FPNE rates). As explained, the Full ECC energy cost grows at the same pace of the assumed ECC overheads. The other three curves use the proposed detection and correction techniques, and apply ECC only to non-PDB regions. The three correction strategies presented similar results. It becomes clear that the overall energy consumption is much lower when using our techniques. For example, assuming an ECC cost of 12.5% (typical of Double Error Detection codes), the energy overhead of the proposed approach is only 1.6%, while the Full ECC approach must consume 12.5% more energy for all accesses.

Related Work

Finocchi et al. (**?**) present a survey on reliable algorithms for unreliable memory. There are works describing resilient sorting algorithms and resilient data structures such as search trees (**?**), priority queues (**?**), and dictionaries (**?**). In contrast with our work which assumes an implicit representation of the state space, these works assume explicit representations of the data structures and of the elements stored in them. Moreover, previous works on reliable algorithms assume an upper bound on the number of errors that occur during the algorithm's execution. Such an assumption is too strong for state-space search problems. This is because one usually does not know a priori how long the search will take (**?**). Our correction methods do not assume an upper bound on the number of errors that might occur during search.

Wagstaff and Bornstein (**?**) studied the effects of memory unreliability caused by radiation on the k-means algorithm (**?**). They discovered that implementations of k-means using kd-trees (**?**) tended to perform poorly under radiation. Later, Gieseke et al. (**?**) developed a version of the kd-tree which is resilient to errors caused by radiation.

Bidirectional pathmax (BPMX) (**?**) (and also pathmax (**?**)) is a technique for dealing with heuristic imprecisions that occur in inconsistent heuristics. BPMX propagates the largest f-value encountered during search while keeping the heuristic admissible. If applied to the approximate computing setting, BPMX would propagate corrupted h-values to different parts of the tree. While BPMX always tries to increase a node's f-value, our methods try to fix a node's h-value. BPMX is applied to inconsistent but admissible heuristics, and our methods to corrupted (and thus potentially inadmissible) heuristics.

Concluding Remarks

In this paper we presented algorithmic-level bounded-suboptimal algorithms for correcting memory errors in memory-based heuristics such as PDBs. IDA* using memory-based heuristics and our correction algorithms are guaranteed to find a solution if one exists and the solutions are guaranteed to cost no more than $3 \cdot C^*$. Our correction algorithms do not make any assumptions on the number of corruptions that occur during search. We showed empirically that if IDA* does not use any correction technique, the solutions it finds might be arbitrarily suboptimal. By contrast, IDA* using our methods found near-optimal solutions in all problem instances it was able to solve within the time limit. We also showed empirically the advantages of our methods over traditional methods for dealing with memory errors. Namely, our methods can be much more energy and memory efficient than ECC schemes.

Acknowledgements: The authors gratefully acknowledge funding from TODO.

Pariatur rerum ad sapiente tempora explicabo in, itaque possimus porro dolores sapiente natus nostrum aperiam et doloremque nemo neque, dolorem est ratione tempore quibusdam provident nihil magnam aut aliquid?Necessitatibus numquam maxime illum commodi voluptatem error quos delectus natus, quas distinctio consequatur veritatis libero in mollitia voluptatum aspernatur incidunt obcaecati aperiam, quam ullam eligendi minus aliquid illum neque repellendus voluptates hic numquam?Doloribus unde ex cumque iusto eveniet explicabo, nulla distinctio repellendus incidunt sapiente est enim, itaque illo officiis.Quos odio vel, sapiente illum velit in-

cidunt tenetur pariatur labore vero numquam esse nulla nihil?