Efficient Priority Optimization in Complex Distributed Embedded Systems through Search Space Adaptation

Arne Hamann, Rolf Ernst
Institute of Computer and Communication Network Engineering
Technical University of Braunschweig, Germany

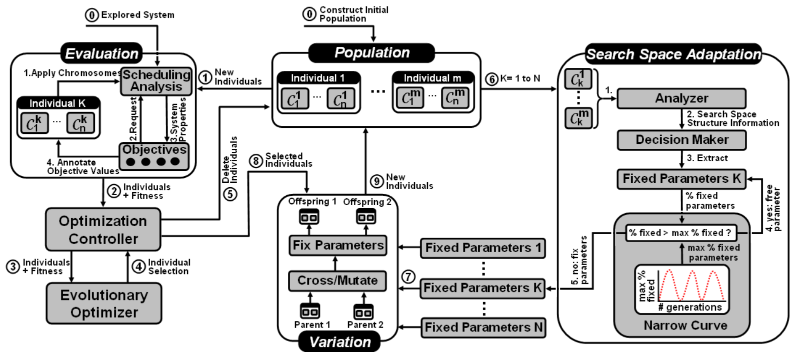

Figure 1. Design Space Exploration Framework with Dynamic Search Space Adaptation

ABSTRACT

In this paper we present a framework for dynamic search space adaptation during evolutionary design space exploration. Compared to previous approaches our framework is capable of adapting the search space dynamically during exploration leading to better search space exploitation in the same exploration time. The application of our framework to priority optimization in complex distributed embedded systems shows that dynamic search space adaptation can significantly increase exploration efficiency, both in terms of exploration time and quality of achieved results.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Performance Attributes

General Terms: Algorithms, Performance

1. SEARCH SPACE ADAPTATION FRAMEWORK

The functioning of the dynamic search space adaptation is visualized in Figure 1. It is clinged into the exploration process between the deletion of individuals (step 5) and the creation of new individuals through variation (steps 8 and 9). Note that the exploration framework adapts the search space exactly once per processed generation.

A search space adaptation strategy is chromosome dependent. For instance, heuristic strategies finding interesting priority assignments for priority scheduled systems are fundamentally different from those assigning clever time slots in time-triggered domains. Therefore, the search space adaptation (step 6) is applied separately for each chromosome included into the exploration process. More precisely, if the population consists of the individuals $[i_1,\dots,i_m]$, where individual i_k, $1 <= k <= m$, consists of n chromosomes instances $[c^k_1,\dots,c^k_n]$, then step 6 is performed once for each set of chromosome instances $[c^1_k,\dots,c^m_k]$, $1 <= k <= n$.

In order to implement a search space adaptation strategy for a specific chromosome, three parts need to be implemented: the *Analyzer*, the *Decision Maker*, and the *Narrow Curve*.

The *Analyzer* analyses the structure of the search space covered by the population at arbitrary times during exploration (step 6.1). It extracts information, which is useful to evaluate the quality of specific system parameter values. As mentioned above, this search space structure analysis is performed chromosome-wise on the population.

The *Decision Maker* takes as input the search space structure information extracted by the *Analyzer* (step 6.2). It is responsible for interpreting this information and makes concrete decisions on promising search space adaptations (step 6.3).

Narrow Curves are a substantial part of heuristic search space adaptation strategies in our framework. A *Narrow Curve* bounds the maximum percentage by which the search space may be restricted at arbitrary times during exploration. Consequently, they define when and to what extend the adaptation strategies are allowed to intervene into the exploration process. It is important to bound the impact of search space adaptation, since heavy search space restrictions early in the exploration process destroy diversity and might easily lead to the convergence of the search towards local optima.

In case that the *Decision Maker* restricts the search space too heavily, the *Narrow Curve* frees parameters until the maximum percentage criterion is satisfied (step 6.4). Afterwards, the search space adaptation decisions are applied (step 6.5), and will be enforced during the subsequent variation process (step 7).

Note that the search space adaptation decisions are only taken into consideration during the variation process of a single generation. For the subsequent generation new search space adaptations are calculated.

In [1] we introduce concrete *Analyzer*, *Decision Maker*, and *Narrow Curve* algorithms for the dynamic search space adaptation of priority scheduled resources in heterogeneous embedded systems. There we show that the dynamic search space adaptation can considerably improve exploration speed and quality of obtained results in comparison to evolutionary algorithms using standard search techniques.

2. References

[1] A. Hamann, R. Ernst. Dynamic Search Space Adaptation in Complex Distributed Embedded System, *Technical Report TR-IDA-2007-01*, Institute of Computer and Communication Network Engineering, Technical University of Braunschweig, Germany.

GECCO'07, July 7–11, 2007, London, England, United Kingdom
ACM 978-1-59593-697-4/07/0007.