Thesis topic:

Heuristic generalized extremal optimization (GEO) can be successfully applied to task scheduling problem allowing find shortest execution time of specified program's graph.

SUMMARY

With fast development of high cost's computers and multiprocessors systems, it is necessary to optimal use of these resources. The scheduling problem concerns ordering of tasks to maximize use of all resources or execute all tasks in the shortest time. This M.Cs. focus on finding best ordering that minimize execution time.

Use of standard gradient algorithms is very difficult for this problem, because search space is huge, and these algorithms are in set of NP-complete problems, so they cannot be solved in polynomial time. The heuristic and meta-heuristic algorithms, based on behavior of nature, are helpful for this class of problems. The genetic algorithms and simulated annealing, which functioning is similar to natural processes like gene mutation, behavior of ants or metals annealing, limits (allow limit?) search space and allow get closer to optimal solution in shortest time. Some kind of these heuristic algorithms is generalized extremal optimization (GEO).

GEO is of easy implementation and can be applied with all kind of variables: district, integer and continuous. Algorithm allow to solve unconstrained and constrained problems, non-convex or disjoint design spaces. The main disadvantage of heuristic algorithms is that not always best solution can be returned.

In this paper focus on design and implement GEO algorithm solving scheduling problem. In effect it should allow find configuration with shortest execution time or close to it in short time, searching only part of solutions set.

In next chapters describe basics of scheduling problem and GEO algorithm. Propose and describe GEO algorithm for scheduling problem. In last sections illustrated results of conducted experiments on testing program graphs. Describe behavior of algorithm dependent of input parameters. Presents statistic chances to find optimal solution and necessary number of objective function executions.