A Fast Genetic Algorithm Based Static Heuristic

For Scheduling Independent Tasks on Heterogeneous Systems

Gaurav Menghani

Department of Computer Engineering,

Thadomal Shahani Engineering College,

University of Mumbai

Outline

- Scheduling
- Scheduling Heuristics
- Sympathy & Segmented Sympathy
- Modified Genetic Algorithm
- Simulation Procedure
- Results

Scheduling

- Is important in systems like Grids, where heterogeneous resources might be connected and heterogeneous tasks need to be scheduled on them.

- Finding the optimal time for running all the tasks on a set of heterogeneous machines is an **NP-Complete** problem.

Scheduling Heuristics

- Online Mode Heuristic – Task is mapped to a machine, as soon as it arrives.

- Batch Mode Heuristic – Task mapping happens at certain scheduled mapping events. A set of unexecuted tasks, termed as a *Metatask*, is considered for mapping.

Scheduling Heuristics

- Minimum Completion Time (MCT) – Map the task to the machine with the best completion time for that task.

- Min-Min – The task with the overall minimum completion time in the meta-task, is assigned to the machine where it achieves it, and removed from the meta-task.

- Sufferage – For every task, its sufferage value (the difference between its second best completion time and best completion time) is computed. The task with the highest sufferage value is assigned to the machine with its minimum completion time, and removed from the metatask.

Scheduling Heuristics

- Segmented Min-Min (SMM) – SMM sorts the tasks according to the minimum/maximum/average expected completion time of each task with respect to each machine. The list is then partitioned into a fixed number of segments. Subsequently, Min-Min is applied individually applied on each segment, in order.

- Genetic Algorithm (GA) – Is used to search large search spaces. It starts with a randomly generated population of chromosomes (task mappings), optionally seeded with mappings generated from other heuristics.

Sympathy Heuristic

- Uses the Sympathy Metric.

$$s_i = E(c_i) \times V(c_i)$$

here,

s_i is the Sympathy metric for task *i*.

$E(c_i)$ is the mean completion time for task *i* on all machines.

$V(c_i)$ is the variance of completion times for task *i* on all machines.

Sympathy Heuristic

1. start

2. do
   2.1 Compute s_i for all tasks in M,
   2.2 Find task i with the highest value of s_i,
   2.3 Map task i to machine j with the lowest value of c_{ij},
   2.4 Remove i from M,
   2.5 Recompute completion times for all tasks

3. while M is not empty.

4. end.

Segmented Sympathy Heuristic

1. start
2. Sort each task by their respective Sympathy metrics.
3. Partition into a fixed number of Segments.
4. Starting with the segment with the highest sympathy metric, apply Min-Min individually on all segments.
5. end.

Modified Genetic Algorithm

Genetic Algorithm described in Braun et. al (referred to as Braun GA henceforth), was found to be inefficient. The following improvements were made.

- **Seeding**: Seeded with Min-Min, Segmented Min-Min, and Segmented Sympathy. Only seeded iterations were retained, since unseeded iterations were not producing good quality mappings.

- **Population & Elitism**: The population size was kept to 100. Top 5% chromosomes in each generation survived.

Modified Genetic Algorithm

- **Swap Based Crossover**: A one-bit swap operation with 40% probability, swaps the machine mapping of a single task between two chromosomes.

- **Fitness Based Crossover**: Is crossover, which is based upon fitness with 20% probability, takes place. In this, two parents produce a single offspring. If one of the parents is more fit than the other, the offspring is more likely to have the assignments according to that parent.

Modified Genetic Algorithm

- **Stopping Condition**: If the elite chromosome does not change for 100 generations, the algorithm stops.

- **Local Search**: To improve the final solution, a local search is performed on the elite chromosome.

Simulation Procedure

- **Heuristics Chosen**: MCT, Min-Min, Sufferage, SMM Average, Sympathy, Segmented Sympathy, Braun GA and GA.

- Three basic types of ETC matrices, Consistent, Semi-Consistent and Inconsistent were used. Each matrix has four types of heterogeneities with respect to tasks and machines. Number of tasks in each matrix is, $t = 16$ and machines is, $m = 512$. In all, twelve types of ETC matrices were used to compare the heuristics.

- For each type of ETC matrix, ten meta-tasks were provided.

Simulation Procedure

- **Heuristics Chosen**: MCT, Min-Min, Sufferage, SMM Average, Sympathy, Segmented Sympathy, Braun GA and GA.

- Three basic types of ETC matrices, Consistent, Semi-Consistent and Inconsistent were used. Each matrix has four types of heterogeneities with respect to tasks and machines. Number of tasks in each matrix is, $t = 16$ and machines is, $m = 512$. In all, twelve types of ETC matrices were used to compare the heuristics.

- For each type of ETC matrix, ten meta-tasks were provided.

Simulation Procedure

- After a heuristic provided a mapping, its makespan was obtained, and a penalty was assigned to the heuristic based upon its makespan and the maximum of the makespans provided by all the heuristics in H, the set of heuristics.

$$P_{it} = \texttt{(makespan of heuristic } i \texttt{ for metatask } t\texttt{) / (maximum makespan amongst all heuristics in } H \texttt{ for } t\texttt{)}$$

- Thus for each metatask a heuristic can receive a score in the range (0, 1]

Results

- MCT was comparable to Min-Min in the consistent cases, but otherwise its makespans were amongst the worst.

- Sufferage also did not perform well.

- Sympathy performed better than Sufferage. It matched Min-Min in all the consistent cases, and was better in High-Low Consistent and High-High Consistent cases.

Results

- Segmented Sympathy was better than Min-Min in seven out of twelve cases, which includes all the consistent cases. Its makespans in the consistent cases were upto 9.1% better than Min-Min. It was also better than Braun GA in all the consistent cases.

- Braun GA was better than Min-Min by about 3.6% in the Low-Low cases. Its best result was 6.4% better than Min-Min in the High-Low Semi Consistent cases.

- Segmented Min-Min Average was the second best heuristic. It was better than Min-Min in ten of the twelve cases, and matched it in the other two cases. In the Low-High Consistent and High-Low Consistent cases, it was better than Min-Min by

Results

- The proposed GA turned out to be the best heuristic. It did better than Min-Min in all the cases. In the High-Low consistent case, it was better than Min-Min by 14.42%.

- Its makespans were also better than Braun GA. In the High-High Consistent and High-Low Consistent cases it got 6.48% and 8.34% smaller makespans. It was also observed that the GA outperforms Braun GA most in the consistent cases (around 5%). On an average, GA gave makespans, which were 3.42% better than Braun GA.

- In one meta-task, the GA was 14 times better than Braun GA. On an average, it was 2.60 times faster.

Thank You

gaurav.menghani@gmail.com

@Gaurav_Menghani on Twitter

+91-932-303-9463