1. Create design expert for time different factor for small-medium
2. Add graph for each factor that is significant
3. Describe each Anova test (for each of the tests explain what happened).
4. Add Anova tables to the description.
5. Explain why you think that the results are as they are.
6. Optimally solved problems
7. Create table for Guan vs. TBased Comparison
8. Create design expert for diff of gaps for big problems
9. Add Special cases chapter

**Experiments**

Problem generation process: For each problem to generate: define *number of jobs (,* to be generated; *number of parallel machines (*; define *maximum* *process time* , and for each job draw process time , from discrete uniform distribution ; define *maximum arrival time* , and for each job draw arrival time , from discrete uniform distribution ; define *maximum due date* , and for each job draw a due date , from discrete uniform distribution . A problem type is a specific set of predefined parameters, and it is denoted by , for example problem with 10 jobs, 10 machines, and max due date of 10 will be written as: m10j10d10. For each problem type 10 instances are being generated.

Time discretization process: Time discretization consists of choosing lower time resolution, and change time values accordingly. Parameter d is equals to . Time parameters are being changed in following manner: *Process time* is divided and then rounded up , *due date* is divided and then rounded down , *arrival time* is divided and then rounded up .

**Dataset 1:**

Parameters for this experiment are: . In this experiment we have 27 different variations and totally 270 generated problems.

Experiment 1

In this experiment Guan, and TBased2, formulations are tested. Goal of this experiment is to compare between two formulations by quality of received solution and time performance.

Analysis

Total time of solving all the problems for Guan took 27653 seconds and 30149 second for TBased2 formulation. Guan had 238 optimally proved solutions, TBased2 had 249. 19 problems TBased2 solution was able to proof optimality while Guan was not, on 8 problems it was the opposite. Number of problems where TBased2 and Guan had optimal solution is 230. Each formulation had one problem where it received a better solution value. Guan solved 222 problems better than our formulation, and only for 48 of the problems TBased2 formulation was aster time.

Figure 1

Figure 1 and 2 shows respectively distribution of time difference between all solutions and optimal solutions only. If time difference is less than 0 than TBased2 solved the problem faster, 0 is when exactly same time received for both of the formulations, and bigger than 0 means that Guan returned the solution faster.

Figure 2

Although for this data set Guan has small advantage over TBased2, there it is hard to decide to use always Guan.

Experiment 2

Purpose of this experiment is to analyze how time discretization affects time performance and solution quality. And compare this to the previous results of experiment 1.

On the same set of problems a discretization was made for TBased2 formulation. Time discretization factors that chosen are 2, 3, 5 and 10. With factor 2 total run time was 5977 seconds, all problems solved optimally. With factor 3 total run time was 2044 seconds, for one problem the solver was not able to proof optimality. For factor of 5 total run time was 649 seconds, all problems solved optimally. For factor of 10 total run time was 325 seconds and all problems were solved optimally. Figure 3 shows the change in percent of solution value as we decrease the time resolution.

Figure 3

Figure 4 and Figure 5 show time change in percent from the regular solution. Although in a more significant cases, where solve time is bigger than 30 seconds, time discretization shows good results, the variance from Guans formulation is very big.

Figure 4

Figure 5

Experiment 3

Purpose of this experiment is to compare constraint with limited time vs. received results from experiment 1, and to decide whether it is a good candidate for using it’s solution in practice.

We run a constraint programming (CP) formulation shown in previous section over the same set of problems. As CP is usually not able to proof its results, after 10 minutes, and it usually finds a very good solution already after 10 seconds, we run it for 10 seconds for each of the generated problems. CP has found the best known result (Guan or TBased2) for 139 of the problems. Figure 6 shows the number of solutions that received from CP, in each of the percentage error rate.

Figure 6

**Data Set 2**

Parameters for this experiment are: . In this experiment we have 30 different variations. For each variation 10 randomly problems are created, after this we change the resolution time for each problem, by multiplying it by constant.

Experiment

Purpose of this experiment is to compare between the two formulations Guan and TBased2 and their behavior when time resolution is changed, while the solution value remains the same. Additionally for each formulation to derive the parameters that has the most impact on solution time.

Analysis

Totally 1500 problems were solved TBased2 solved all of them optimally while Guan had only 1129 optimal solutions. Out of 421 not optimally solved problem, only 25 of them did not had the optimal solution as proved by TBased2

Figure 7 shows clearly that as time resolution increases the average solution time for Guan stays the same, while for TBased2 we can see an exponential increase in time. Figure 7, includes only problems where . Figure 8, shows the same graph when . A similar result is achieved, but average solution time for Guan is higher than for TBased2, meaning despite the increase in time we would still prefer to use TBased2 formulation.

Figure 7

Figure 8

Problems with have high rate of not optimal solutions for Guan thus, despite the graph will look similar as almost straight line for Guan, and an exponentially increasing line for TBased2 it is left out.

A lot of not optimal values in Guan formulation for, indicates that Guan is a lot more affected by number of jobs than TBased2 formulation. Figure 9 depicts this. Trend lines ending with “Optimal”, mean that only solutions for which Guan had optimal solution had been taken into account.

Figure 9

**Big problems data set**

**Data Set Big 1**

Parameters for this experiment are: . In this experiment we have 4 different variations. There are total of 40 problems, TBased was able to solve it optimally 4, while Guan solved optimally 12. Guan had better solution value for 35 problems, and for 5 problems there is equality in solution value. Average gap value for Guan is 15.9%, and for TBased2 is 74.6%. Average build time for Guan is 0.4 seconds, while for TBased2 is 411.66 seconds. Average solution time for guan is 486.7 seconds, and for TBased it is 1035.6 seconds.

**Data Set Big 1**

Parameters for this experiment are: . In this experiment we have 4 different variations. There are total of 40 problems, TBased was able to solve it optimally 23, while Guan solved optimally 14. Guan had 8 better solution values while TBased had 6, and equality on other 26 problems. Average gap value for Guan is 12.6%, and for TBased2 is 4.8%. Average build time for Guan is 0.23 seconds, while for TBased2 is 85.75 seconds. When taking problems that have only p=20 the build time for TBased is 72 seconds, and for p=25 it is 99.56 seconds, Guan’s average build time remains the same. Average solution time for guan is 435.23 seconds, and for TBased it is 447 seconds. When taking problems that have only p=20 the solution time for TBased is 386.55 seconds, and when p=25 the solution time is 508.55 seconds, for Guan it remains similar all the time. Solution time includes the build phase and the solution search phase.

**Dataset 2:**

Parameters for this experiment are: . In this experiment we have 32 different variations and totally 320 generated problems.

Experiment 1

In this experiment Guan, and TBased2, formulations are compared on small to medium problems. Goal of this experiment is to compare between two formulations by quality of received solution and time performance.

Analysis

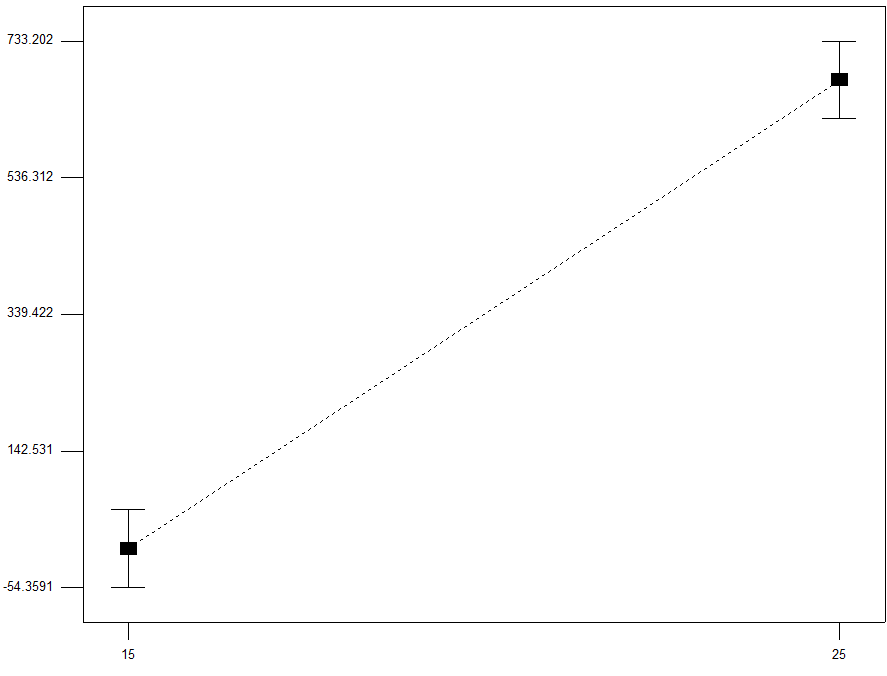
Total time of solving all the problems for Guan took 108615 seconds and 84304 second for TBased2 formulation. Guan had 227 optimally proved solutions, TBased2 had 300. 37 problems TBased2 solution was able to proof optimality while Guan was not, on 14 problems it was the opposite. Number of problems where TBased2 and Guan had optimal solution is 263. TBased2 had twice better solution value than Guan, while Guan had 10 better solutions. Guan solved 210 problems faster than TBased2, and 104 of the problems TBased2 formulation was faster time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **p** | **guan\_time** | **tbased2\_time** | **guan\_median** | **tbased2\_median** |
| **5** | 311.3007372 | 31.59303751 | 6.642 | 4.457 |
| **10** | 568.1528265 | 64.06026299 | 13.757 | 15.319 |
| **20** | 168.7783255 | 309.9214136 | 6.179 | 63.762 |
| **40** | 309.4541251 | 648.2167989 | 3.117 | 3.117 |

Figure <Id1> shows that the process time of guan is not being increased as the maximum process time increases, while TBased2 time is being increased exponentially.

Figure <Id1>

Anova test showed, that due date has no impact on the solution time for any of the models. Guan is mostly being affected by number of jobs



For TBased2, all parameters except due date significant were significant to the process time value.

**Dataset Big**

Parameters for this experiment are: . In this experiment we have 24 different variations and totally 240 generated problems.

Experiment 1

In this experiment Guan, and TBased2, formulations are compared on big problems. Goal of this experiment is to compare between two formulations by quality of received solution and time performance for big problems.

Analysis

Total time of solving all the problems for Guan took 406124 seconds and 230914 second for TBased2 formulation. Guan had 20 optimally proved solutions, TBased2 had 144. 130 problems TBased2 solution was able to proof optimality while Guan was not, on 6 problems it was the opposite. Number of problems where TBased2 and Guan had optimal solution is 14. TBased2 had 94 better solutions value than Guan, while Guan had 58 better solutions. Guan solved 14 problems faster than TBased2, and 133 of the problems TBased2 formulation was faster time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **p** | **guan\_time** | **tbasedw\_time** | **guan\_median** | **tbased2\_median** |
| **5** | 1781.526366 | 199.3056508 | 1800 | 42.893 |
| **10** | 1772.180449 | 696.2042634 | 1800 | 274.22 |
| **20** | 1585.8147 | 1240.709871 | 1800 | 1753.732 |
| **40** | 1629.198816 | 1712.361517 | 1800 | 1800 |

Optimally solved problems

On the 14 problem that were optimally solved in both formulations, guan had total run time of 8154 seconds and TBased2 run for 7745 seconds. All the problems are where number of jobs is 30. In general Guan does not have optimal solution where the number of jobs is bigger than 30.

Not optimally solved problems

There are totally 90 problems that were not solved optimally both by Guan and TBased2. For this problems average gap value is 0.338 for guan model, and 0.439 for TBased2 model. In 55 problems TBased2 had worse solution than Guan, and in 20 problems it was the opposite.