# Supplementary Material: Min-Max Propagation

**Christopher Srinivasa**
University of Toronto
Borealis AI
christopher.srinivasa
@gmail.com

**Inmar Givoni**
University of
Toronto
inmar.givoni
@gmail.com

**Siamak Ravanbakhsh**
University of
British
Columbia
siamakx@cs.ubc.ca

**Brendan J. Frey**
University of Toronto
Vector Institute
Deep Genomics
frey@psi.toronto.edu

## Abstract

This is the supplementary material for the paper titled "Min-Max Propagation".

## 1 Functions $f$ that are Easy to Minimize

Here we address the issues of minimizing $f(x)$ subject to a given set of constraints, fixing values of certain $x_i$s. There are several forms of $f$ for which this evaluation is easy. We explain some examples.

**Functions of the form $f(x) = \sum_i \alpha_i x_i$.** In the simplest case, where $\alpha_i = 1$, we are counting the number of variables set to one. Therefore, the minimum is obtained by setting all unconstrained variables to zero. The value obtained is equal to the number of constrained variables already fixed to one. With positive coefficients, again, the minimum is obtained by setting all the unconstrained variables to zero, and the value obtained is the weighted sum of the constrained variables fixed to one. Thus, generally, unconstrained variables are fixed to zero if their coefficient is positive, and to one if it is negative. The value obtained is the weighted sum of all coefficients whose variables are fixed to one, constrained and unconstrained.

**Functions of the form $f(x) = g\left(\sum_i \alpha_i x_i\right)$.** These can be solved if $g$ is either monotonically increasing or monotonically decreasing. In the former case, all unconstrained variables with positive coefficients are set to zero. All unconstrained variables with negative coefficients are set to one. The value is found by calculating the corresponding $g$ value. In the latter case, all unconstrained variables with positive coefficients are set to one, and all unconstrained variables with negative coefficients are set to zero.

## 2 Additional Results

### 2.1 Sum-Product Reduction vs. Min-Max Propagation

Fig. 1 shows the number of iterations for each of the head-to-head min-max results reported in the paper.

### 2.2 Additional Results for Identical Machines Problems

We compare min-max propagation (with different decimation procedures) against LPT on a set of benchmark experiments designed in (Gupta and Ruiz-Torres, 2001) for the identical machine version of the problem – *i.e.* a task has the same processing time on all machines ($p_n = p_{1n} = p_{2n} = .. = p_{Mn}$).

In total there are four families our benchmark experiments, as shown in Table 1, where for each family all combinations of specified number of jobs $N$, machines $M$, and distributions from which

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

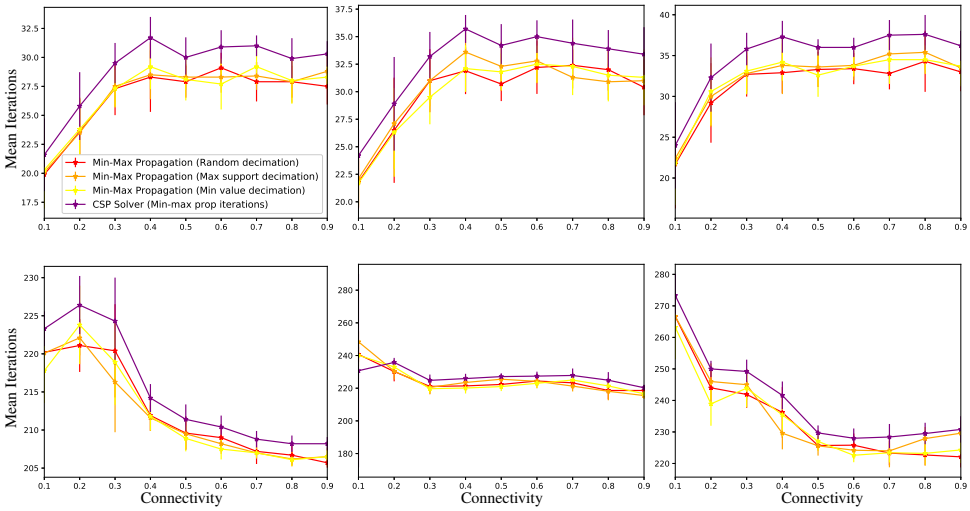


Figure 1: Performance of different methods on Erdos-Renyi graphs.Top: N=10, Bottom: N=100, Left: K=4, Middle: K=6, Right: K=8.

| | M | N | Dist |
|---|---|---|---|
| E1 | 3,4,5 | 2M,3M,5M | U(1,20), U(20,50) |
| E2 | 2,3,4,6,8,10 | 10,30,50,100 | U(100,800) |
| E3 | 3,5,8,10 | 3M+1,3M+2,4M+1, 4M+2, and 5M+2 | U(1,100),U(100,200) |
| E4 | 2<br>3 | 9<br>10 | U(1,20), U(20,50), U(50,100), U(100,200), U(100,800) |

Table 1: Set of benchmark experiments for identical machines makespan minimization.

the jobs are drawn are tested. The only exception to this is $E4$ where the $M = 2, N = 9$ combination is tested separately from the $M = 3, N = 10$. For all experiments the processing times are drawn from a uniform distribution with varying bounds.

The experiments are designed to set different difficulties of the problem. For example, $E1$ and $E4$ are designed to test small problem instances. In contrast, $E2$ and $E3$ are setup such as to test larger instance. For all experiments, the performance number reported is the ratio of the makespan achieved by the method being tested to a lower bound $LB$ computed as:

$$LB = \max\left(\max_{1\leq i\leq n} p_n, \sum_{n=1}^{N} p_n/M\right). \tag{1}$$

Table 2 shows the scenario where min-max propagation performs best against the LPT algorithm. We see that this scenario involves large instances. From this table, we also see that max-support decimation almost always outperforms the other decimation schemes. From the additional results in Tables 3 to 7, we see that our framework does not perform as well on small instances.

### 2.3 Additional Results for Unrelated Machines Problem

We test the min-max propagation with max-support decimation against a more difficult version of the problem: the unrelated machine model, where each job has a different processing time on each machine. Specifically, we compare our method against that of (Vinyals et al., 2013) which also uses distributive law for min-max inference to solve a load balancing problem. However, that paper studies a sparsified version of the unrelated machines problem where tasks are restricted to a subset of machines (*i.e.* they have infinite processing time for particular machines). Nevertheless, we can

| M | N | LPT | Min-Max Prop (Random Dec.) | Min-Max Prop (Max-Support Dec.) | Min-Max Prop (Min-Value Dec.) |
|---|---|---|---|---|---|
| 3 | 10 | 1.127 | 1.124 | 1.078 | **1.071** |
| | 11 | **1.067** | 1.106 | 1.083 | 1.076 |
| | 13 | 1.098 | 1.112 | **1.031** | 1.044 |
| | 14 | **1.052** | 1.054 | 1.069 | 1.056 |
| | 16 | 1.083 | **1.041** | 1.048 | 1.047 |
| | 17 | 1.040 | 1.057 | 1.048 | **1.032** |
| 5 | 16 | 1.162 | 1.173 | **1.098** | 1.099 |
| | 17 | 1.123 | 1.178 | 1.097 | **1.082** |
| | 21 | 1.113 | 1.108 | **1.055** | 1.062 |
| | 22 | 1.092 | 1.116 | 1.084 | **1.057** |
| | 26 | 1.102 | 1.070 | 1.062 | **1.053** |
| | 27 | 1.073 | 1.075 | **1.044** | 1.048 |
| 8 | 25 | 1.178 | 1.183 | **1.091** | 1.128 |
| | 26 | 1.144 | 1.167 | **1.079** | 1.112 |
| | 33 | 1.135 | 1.144 | **1.081** | 1.093 |
| | 34 | 1.117 | 1.132 | **1.071** | 1.086 |
| | 41 | 1.112 | 1.117 | **1.055** | 1.077 |
| | 42 | 1.094 | 1.109 | 1.079 | **1.074** |
| 10 | 31 | 1.184 | 1.168 | 1.110 | **1.105** |
| | 32 | 1.165 | 1.186 | **1.109** | 1.111 |
| | 41 | 1.138 | 1.183 | **1.077** | 1.088 |
| | 42 | 1.124 | 1.126 | **1.074** | 1.090 |
| | 51 | 1.112 | 1.131 | **1.077** | 1.081 |
| | 52 | 1.102 | 1.100 | **1.051** | 1.076 |

Table 2: Makespan minimization experiment 3 with U(100,200). Min-max ratio to a lower bound (lower is better) obtained by LPT with 4/3-approximation guarantee versus min-max propagation using different decimation procedures. $N$ is the number of jobs and $M$ is the number of machines. In this setting, all jobs have the same run-time across all machines.

| M | N | LPT | Min-Max Prop (Random Dec.) | Min-Max Prop (Max-Support Dec.) | Min-Max Prop (Min-Value Dec.) |
|---|---|---|---|---|---|
| 3 | 6 | **1.115** | 1.197 | 1.132 | 1.140 |
| | 9 | **1.052** | 1.098 | 1.069 | 1.090 |
| | 15 | **1.015** | 1.057 | 1.035 | 1.050 |
| 4 | 8 | **1.083** | 1.242 | 1.128 | 1.178 |
| | 12 | **1.048** | 1.116 | 1.072 | 1.089 |
| | 20 | **1.024** | 1.052 | 1.036 | 1.049 |
| 5 | 10 | **1.122** | 1.230 | 1.144 | 1.186 |
| | 15 | **1.051** | 1.127 | 1.089 | 1.102 |
| | 25 | **1.019** | 1.061 | 1.029 | 1.053 |

Table 3: Makespan minimization experiment 1 with U(1,20).

| M | N | LPT | Min-Max Prop (Random Dec.) | Min-Max Prop (Max-Support Dec.) | Min-Max Prop (Min-Value Dec.) |
|---|---|---|---|---|---|
| 3 | 6 | **1.048** | 1.111 | 1.082 | 1.087 |
| | 9 | **1.038** | 1.063 | 1.057 | 1.069 |
| | 15 | **1.012** | 1.126 | 1.042 | 1.044 |
| 4 | 8 | **1.055** | 1.191 | 1.103 | 1.111 |
| | 12 | **1.024** | 1.135 | 1.071 | 1.074 |
| | 20 | **1.006** | 1.079 | 1.050 | 1.048 |
| 5 | 10 | **1.049** | 1.134 | 1.091 | 1.116 |
| | 15 | **1.021** | 1.142 | 1.080 | 1.112 |
| | 25 | **1.007** | 1.064 | 1.053 | 1.053 |

Table 4: Makespan minimization experiment 1 with U(20,50).

| N | M | LPT | Min-Max Prop (Random Dec.) | Min-Max Prop (Max-Support Dec.) | Min-Max Prop (Min-Value Dec.) |
|---|---|---|---|---|---|
| 10 | 2 | **1.011** | 1.018 | 1.022 | 1.022 |
| | 3 | 1.045 | 1.053 | 1.047 | **1.035** |
| | 4 | **1.089** | 1.145 | 1.108 | 1.092 |
| | 6 | **1.103** | 1.252 | 1.132 | 1.182 |
| | 8 | **1.027** | 1.135 | 1.043 | 1.046 |
| | 10 | **1.000** | **1.000** | **1.000** | **1.000** |
| 30 | 2 | **1.001** | 1.010 | 1.006 | 1.006 |
| | 3 | **1.004** | 1.046 | 1.014 | 1.015 |
| | 4 | **1.017** | 1.067 | 1.016 | 1.024 |
| | 6 | **1.015** | 1.073 | 1.032 | 1.062 |
| | 8 | **1.042** | 1.155 | 1.054 | 1.098 |
| | 10 | **1.041** | 1.223 | 1.073 | 1.128 |
| 50 | 2 | **1.000** | 1.004 | 1.004 | 1.004 |
| | 3 | **1.006** | 1.038 | 1.007 | 1.010 |
| | 4 | **1.010** | 1.038 | 1.011 | 1.014 |
| | 6 | **1.019** | 1.066 | 1.029 | 1.041 |
| | 8 | **1.022** | 1.091 | 1.044 | 1.047 |
| | 10 | **1.017** | 1.157 | 1.073 | 1.067 |
| 100 | 2 | **1.000** | 1.003 | 1.001 | 1.001 |
| | 3 | **1.004** | 1.007 | 1.004 | 1.003 |
| | 4 | **1.000** | 1.022 | 1.008 | 1.007 |
| | 6 | **1.005** | 1.057 | 1.024 | 1.016 |
| | 8 | **1.010** | 1.070 | 1.065 | 1.022 |
| | 10 | **1.004** | 1.063 | 1.107 | 1.038 |

Table 5: Makespan minimization experiment 2.

| M | N | LPT | Min-Max Prop (Random Dec.) | Min-Max Prop (Max-Support Dec.) | Min-Max Prop (Min-Value Dec.) |
|---|---|---|---|---|---|
| 3 | 10 | **1.040** | 1.065 | 1.059 | 1.063 |
| | 11 | **1.038** | 1.079 | 1.040 | 1.041 |
| | 13 | **1.023** | 1.045 | 1.039 | 1.050 |
| | 14 | **1.021** | 1.062 | 1.059 | 1.044 |
| | 16 | **1.016** | 1.050 | 1.029 | 1.033 |
| | 17 | **1.013** | 1.034 | 1.024 | 1.030 |
| 5 | 16 | **1.031** | 1.129 | 1.075 | 1.104 |
| | 17 | **1.038** | 1.116 | 1.066 | 1.112 |
| | 21 | **1.032** | 1.122 | 1.078 | 1.078 |
| | 22 | **1.028** | 1.106 | 1.035 | 1.059 |
| | 26 | **1.017** | 1.101 | 1.065 | 1.038 |
| | 27 | **1.018** | 1.093 | 1.049 | 1.050 |
| 8 | 25 | **1.036** | 1.141 | 1.078 | 1.121 |
| | 26 | **1.045** | 1.202 | 1.082 | 1.155 |
| | 33 | **1.029** | 1.126 | 1.049 | 1.097 |
| | 34 | **1.038** | 1.087 | 1.061 | 1.088 |
| | 41 | **1.020** | 1.081 | 1.068 | 1.071 |
| | 42 | **1.021** | 1.130 | 1.049 | 1.069 |
| 10 | 31 | **1.057** | 1.211 | 1.116 | 1.154 |
| | 32 | **1.050** | 1.144 | 1.107 | 1.148 |
| | 41 | **1.028** | 1.133 | 1.103 | 1.102 |
| | 42 | **1.026** | 1.164 | 1.073 | 1.114 |
| | 51 | **1.020** | 1.119 | 1.035 | 1.070 |
| | 52 | **1.013** | 1.122 | 1.042 | 1.075 |

Table 6: Makespan minimization experiment 3 with U(1,100).

| M,N | U(,) | LPT | Min-Max Prop (Random Dec.) | Min-Max Prop (Max-Support Dec.) | Min-Max Prop (Min-Value Dec.) |
|---|---|---|---|---|---|
| 2,9 | U(1,20) | **1.016** | 1.036 | 1.026 | 1.026 |
| | U(20,50) | 1.062 | 1.048 | **1.028** | **1.028** |
| | U(1,100) | 1.017 | **1.016** | 1.034 | 1.034 |
| | U(50,100) | 1.072 | 1.055 | **1.042** | **1.042** |
| | U(100,200) | 1.073 | **1.051** | 1.055 | 1.055 |
| | U(100,800) | 1.028 | 1.027 | **1.026** | **1.026** |
| 3,10 | U(1,20) | **1.031** | 1.127 | 1.057 | 1.076 |
| | U(20,50) | 1.103 | 1.095 | **1.052** | 1.075 |
| | U(1,100) | **1.033** | 1.096 | 1.039 | 1.048 |
| | U(50,100) | 1.142 | 1.120 | **1.083** | 1.068 |
| | U(100,200) | 1.127 | 1.133 | **1.083** | 1.094 |
| | U(100,800) | **1.049** | 1.056 | 1.061 | 1.078 |

Table 7: Makespan minimization experiment 4.

| Mode | Dist |
|---|---|
| Uncorrelated (Mode 0) | $N(100, 10)$ |
| Machine correlated (Mode 1) | $N(\alpha_i, 10)$ where $\alpha_i \sim U(50, 100)$ and $\alpha_i$ is different for each machine |
| Machine/Task correlated (Mode 2) | $N(\beta_k + \alpha_i, 10)$ where $\alpha_i, \beta_k \sim U(50, 100)$, $\alpha_i$ is different for each machine, and $\beta_i$ is different for each job |

Table 8: Processing time generation for unrelated machine makespan minimization.

| Mode | N/M | (Vinyals et al., 2013) | Min-Max Prop |
|---|---|---|---|
| 0 | 10 | **0.94(0.01)** | 0.92(0.01) |
| | 15 | **0.93(0.00)** | 0.90(0.02) |
| 1 | 10 | **0.92(0.00)** | 0.87(0.05) |
| | 15 | **0.89(0.00)** | 0.82(0.02) |
| 2 | 10 | **0.87(0.00)** | 0.85(0.05) |
| | 15 | **0.85(0.00)** | **0.85(0.04)** |

Table 9: Makespan minimization with M=40. Min-max ratio (LP relaxation to that) of min-max propagation versus same for the method of (Vinyals et al., 2013) (higher is better). Mode 0, 1 and 2 corresponds to uncorrelated, machine correlated and machine-task correlated respectively.

still compare their results to what we can achieve using min-max propagation using infinite-time constraints.

We use the same problem setup with three different ways of generating the processing times (uncorrelated, machine correlated, and machine/task correlated) and compare our answers to IBM's CPLEX solver exactly as the authors do in that paper (where a high ratio is better). Table 8 shows a summary of the processing time generation for each of the different ways (note that all normal distributions are bounded at the low end to have a minimum of 10).

Tables 9 to 11 show the results. Here again, min-max propagation works best for large instances.

| Mode | N/M | (Vinyals et al., 2013) | Min-Max Prop |
|---|---|---|---|
| 0 | 10 | **0.94(0.01)** | 0.93(0.00) |
| | 15 | **0.94(0.00)** | 0.89(0.03) |
| 1 | 10 | **0.90(0.00)** | 0.87(0.01) |
| | 15 | **0.87(0.01)** | 0.85(0.02) |
| 2 | 10 | 0.87(0.01) | **0.89(0.03)** |
| | 15 | **0.87(0.01)** | 0.79(0.09) |

Table 10: Makespan minimization with M=60.

| Mode | N/M | (Vinyals et al., 2013) | Min-Max Prop |
|---|---|---|---|
| 0 | 5 | 0.93(0.03) | **0.95(0.01)** |
| | 10 | **0.94(0.01)** | 0.93(0.01) |
| | 15 | **0.94(0.00)** | 0.90(0.01) |
| 1 | 5 | **0.90(0.01)** | 0.86(0.07) |
| | 10 | **0.90(0.00)** | 0.88(0.00) |
| | 15 | **0.87(0.01)** | 0.73(0.03) |
| 2 | 5 | 0.81(0.01) | **0.89(0.01)** |
| | 10 | 0.81(0.01) | **0.89(0.01)** |
| | 15 | 0.78(0.01) | **0.86(0.01)** |

Table 11: Makespan minimization with M=80.

## References


J. N. D. Gupta and A. J. Ruiz-Torres. A listfit heuristic for minimizing makespan on identical parallel machines. *Production Planning & Control*, 12(1):28–36, 2001.

M. Vinyals, K. S. Macarthur, A. Farinelli, S. D. Ramchurn, and N. R. Jennings. A message-passing approach to decentralized parallel machine scheduling. *The Computer Journal*, 2013.