Empirical Performance Evaluation of a Parameter-Free GA for JSSP

Shouichi Matsui, Isamu Watanabe, and Ken-ichi Tokoro

Central Research Institute of Electric Power Industry (CRIEPI)
2-11-1 Iwado-kita, Komae-shi, Tokyo 201-8511, JAPAN
{matsui,isamu,tokoro}@criepi.denken.or.jp

1 Introduction

The job-shop scheduling problem (JSSP) is a well known difficult NP-hard problem. Genetic Algorithms (GAs) for solving the JSSP have been proposed, and they perform well compared with other approaches [1]. However, the tuning of genetic parameters has to be performed by trial and error. To address this problem, Sawai et al. have proposed the Parameter-free GA (PfGA), for which no control parameters for genetic operation need to be set in advance [3].

We proposed an extension of the PfGA, a real-coded PfGA, for JSSP [2], and reported that the GA performed well without tedious parameter-tuning. This paper reports the performance of the GA to a wider range of problem instances. The simulation results show that the GA performs well for many problem instances, and the performance can be improved greatly by increasing the number of subpopulations in the parallel distributed version.

2 Computational Results

The GA is tested by the benchmark problems from ORLib [4]. We tested a wider range of instances, namely 10 tough problems, ORB01–ORB10, SWV01–SWV20, and TA01–TA30, but only the results for ORB01–ORB10 and SWV01–SWV20 are shown in Table 1 and Table 2. The GA was run for each problem instance using 50 different random seeds. The maximum number of fitness evaluation was set to 1,000,000 for all cases.

– The GA can always find the optimal makespan when $N \geq 1$ in ORB01, ORB03, ORB04, and ORB07–ORB10, when $N \geq 2$ in ORB10, when $N \geq 32$ in ORB09, and when $N = 64$ in ORB08.
– As we increase the number of subpopulations, the average makespan is reduced, but the best makespan does not always shorten.
– The relative error of SWV instances to the best upper bound is larger than that of ORB cases. The size of the problems is a reason of the difference. The size of ORB problems is 10×10 and it is 20×10, 20×15, or 50×20 in SWV problems. The length of the chromosomes used in the GA is proportional to the size of problems, $2 \times n$ (jobs) $\times m$ (machines), therefore the length (L) is 200 in ORB problems, and 400, 600, or 2000 in SWV problems.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 1318–1319, 2004.

Table 1. Relative error to the optimal makespan (%) (ORB problems)

		$N = 1$		$N = 2$		$N = 4$		$N = 8$		$N = 16$		$N = 32$		$N = 64$	
Prob.	Opt[1]	best	μ	best	μ	best	μ	best	μ	best	μ	best	μ	best	μ
ORB01	1059	**0.0**	2.6	**0.0**	2.1	**0.0**	1.6	**0.0**	1.0	**0.0**	0.7	**0.0**	0.3	**0.0**	0.1
ORB02	888	0.1	0.5	0.1	0.5	**0.0**	0.3	**0.0**	0.1	**0.0**	0.1	**0.0**	0.1	**0.0**	0.1
ORB03	1005	**0.0**	2.3	**0.0**	2.2	**0.0**	1.7	**0.0**	1.1	**0.0**	1.0	**0.0**	0.6	**0.0**	0.2
ORB04	1005	**0.0**	1.2	0.6	1.4	0.6	1.1	**0.0**	0.8	**0.0**	0.6	**0.0**	0.5	**0.0**	0.3
ORB05	887	0.2	0.7	0.2	0.5	0.2	0.4	**0.0**	0.2	**0.0**	0.2	**0.0**	0.2	**0.0**	0.2
ORB06	1010	0.2	1.9	0.2	1.6	0.2	1.4	**0.0**	1.1	**0.0**	0.9	**0.0**	0.7	**0.0**	0.4
ORB07	397	**0.0**	1.1	**0.0**	1.0	**0.0**	0.8	**0.0**	0.6	**0.0**	0.2	**0.0**	0.0	**0.0**	**0.0**
ORB08	899	**0.0**	2.4	**0.0**	2.4	**0.0**	1.9	**0.0**	1.5	**0.0**	0.8	**0.0**	0.4	**0.0**	0.3
ORB09	934	**0.0**	0.6	**0.0**	0.8	**0.0**	0.3	**0.0**	0.3	**0.0**	0.1	**0.0**	**0.0**	**0.0**	**0.0**
ORB10	944	**0.0**	0.1	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

Table 2. Relative error to the best upper bound (%) (hard SWV problems)

		$N = 1$		$N = 2$		$N = 4$		$N = 8$		$N = 16$		$N = 32$		$N = 64$	
Prob.	bub[1]	best	μ	best	μ	best	μ	best	μ	best	μ	best	μ	best	μ
SWV01	1407	5.8	9.8	5.5	8.5	5.1	7.5	4.5	7.1	3.5	5.9	2.5	5.1	3.7	5.2
SWV02	1475	5.0	7.7	4.3	7.1	4.2	6.4	3.5	5.6	1.6	5.2	1.4	4.6	1.9	4.3
SWV03	1398	5.9	8.9	6.1	8.8	4.4	7.2	3.7	6.5	2.5	5.5	2.7	5.1	2.5	4.8
SWV04	1483	5.1	8.0	3.6	7.3	3.0	6.5	2.9	5.8	1.8	4.9	2.4	4.2	1.8	3.5
SWV05	1424	7.6	10.7	7.0	10.0	6.3	8.9	4.1	8.2	3.6	7.0	3.2	6.3	3.2	5.8
SWV06	1678	8.3	12.0	8.5	11.4	6.7	10.6	6.9	9.5	6.2	8.8	6.0	7.8	4.4	7.1
SWV07	1620	7.2	10.4	6.9	10.3	5.2	9.6	5.0	8.7	5.6	7.5	4.2	6.9	4.1	6.2
SWV08	1763	10.0	13.8	9.6	12.8	5.8	11.5	6.7	10.7	5.8	9.7	6.7	8.9	5.0	8.0
SWV09	1663	8.8	12.6	7.2	12.1	7.8	11.1	6.0	10.0	6.0	9.0	4.7	8.0	4.8	7.3
SWV10	1767	7.6	9.8	6.1	8.8	4.4	8.2	5.3	7.5	4.6	6.7	3.6	5.8	2.4	5.4
SWV11	2991	11.2	16.0	10.9	15.6	9.8	14.8	10.6	13.5	8.3	12.2	8.4	11.1	7.4	10.3
SWV12	3003	14.2	16.9	12.5	16.0	10.7	15.0	11.1	14.6	10.9	13.1	9.8	12.2	9.0	11.5
SWV13	3104	11.5	14.9	10.1	14.1	8.8	12.6	9.0	12.1	7.9	11.2	8.7	10.3	6.6	9.1
SWV14	2968	10.3	14.1	9.8	13.7	7.4	11.7	8.4	11.2	6.5	9.8	6.1	9.1	5.0	7.4
SWV15	2904	14.6	18.0	13.8	17.7	12.6	16.3	11.6	15.4	10.5	14.3	9.8	12.8	9.8	12.1

References

1. Jain, A.S., and Meeran, S.: Deterministic job-shop scheduling: past, present and future, *European Journal of Operational Research*, vol.113, pp.390–434, 1999.
2. Matsui, S., Watanabe, I., and Tokoro, K.: Real-coded parameter-free genetic algorithm for job-shop scheduling problems, *Proc. Seventh Parallel Problem Solving from Nature – PPSN VII*, pp.800–810, 2002.
3. Sawai, H., Kizu, S.: Parameter-free genetic algorithm inspired by "disparity theory of evolution", *Proc. Fifth Parallel Problem Solving from Nature – PPSN V*, pp.702–711, 1998.
4. Vaessens, R.J.M.: Operations Research Library of Problems, Management School, Imperial College London, ftp://mscmga.ms.ic.ac.uk/pub/jobshop1.txt, 1996.