Solving the Job Shop Scheduling Problem Using Parallel Genetic Algorithms

Athanasios Spanos[1], Stefan Wagner[2], Michael Affenzeller[2]

[1] Sector of Industrial Management and Operations Research
School of Mechanical Engineering
National Technical University of Athens, Greece
atspanos@mail.ntua.gr
[2] Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media - Hagenberg
Upper Austria University of Applied Sciences, Austria
{swagner,maffenze}@heuristiclab.com

Extended Abstract

The purpose of this paper is to present a thorough empirical analysis regarding the robustness, effectiveness and efficiency of four distinct genetic algorithm templates on the well known Job Shop Scheduling Problem (JSSP). Its popularity has maintained this combinatorial problem in the academic foray for the past five decades attracting researchers from the fields of combinatorial optimization, operations research and production management. The vast research can in a large part be attributed to classification of the JSSP as the most stubborn and intractable scheduling problem [11]. A typical JSSP instance of dimensionality $n \times m$ consists of n jobs that have to be processed on m machines in a pre-specified order and no operation of a job on a machine may be preempted [8]. As Lenstra and Rinnoy Kan [12] and Brizuela and Sannomiya [3] among others have proclaimed, for $m > 3$ the JSSP as a combinatorial optimization problem falls under the NP-hard category, meaning there is no efficient optimization algorithm known that can solve the general case for $m > 3$ in polynomial time. For proof regarding the NP-hardness of the JSSP the reader is referred to [5].

In light of the above, optimization algorithms such as branch and bound [4] and mathematical techniques based on mixed integer linear models [13] are applicable to small problem instances due to their sizeable computational overhead. For problems with more than 15 jobs and machines such techniques are rendered impractical since they may run for years on end on modern hardware and still not find the optimum solution. In the general case the solution space of the JSSP consists of $(n!)^m$ both feasible and infeasible solutions. Although quite a sizeable degree of this solution space is infeasible and as such useless, the remaining portion that needs to be traversed is still quite large and exhaustive enumeration techniques have stopped being competitive since the mid-90s [16].

The main reason for the diminished popularity of approximate techniques was the advent of metaheuristic approaches such as tabu search [15], simulated annealing [10] and genetic algorithms [14]. Of the three the first has been the

most effective thus far having numerous current state of the art algorithms to show for its remarkable performance: the iterated Tabu Seach Algorithm with Backtracking by Nowicki and Smutnicki [16], originally put forth in 2001, the hybrid tabu search simulated annealing template by Zhang et al. [21] and the hybrid constraint programming tabu search technique by Watson and Beck [18]. As it can be seen simulated annealing when hybridized with other metaheuristics can achieve state of the art performance, yet such a goal seems to be beyond the reach of genetic algorithms (GAs). As a matter of fact the most successful thus far has been Mattfeld's GA3 [14] which uses a hill climber based on the neighborhood operator of Dell'Amico and Trubian [6], a toroidal mesh on the population, thus forming fine grained parallelization, and a diffusion model. Despite strong performance on a variety of mainly square benchmark instances, the scalability of GA3 was quite poor.

In order to further assess performance of parallel GAs in terms of effectiveness and robustness the authors chose to apply some of the newer evolutionary paradigms such as SEGA (Segregative Genetic Algorithm) proposed by Affenzeller [1] and SASEGASA (Self Adaptive Segregative Genetic Algorithm with Simulated Annealing) developed by Affenzeller and Wagner [2]. SEGA is based on the concept of segregation and reunification which aims at significantly retarding premature convergence which plagues the vast majority of GA implementations [20]. The initial population of chromosomes is broken down to a fixed number of subpopulations that evolve independently from one another. When fitness stagnates reunification is performed to diversify the stagnated subpopulation and to stimulate the evolutionary search process again. Through a social metaphor, the segregation and reunification processes in SEGA can be viewed as numerous villages that step by step grow into towns while the latter grow further to form a large city by the final stages of the search. Population replacement is maintained by variable selection pressure, a concept derived from evolution strategies. In its SEGA adaptation, variable selection pressure guides the formation of a virtual population after crossover and mutation out of which the best solutions are selected to be promoted to the next generation. All the aforementioned characteristics formed the SEGA application to the TSP by Affenzeller [1] which was deemed successful especially when compared to COSA [19] and a standard GA.

SASEGASA builds on the initial success of SEGA for the TSP by embedding a new parameter termed success ratio to boost performance of the selection pressure model. The success ratio gives the quotient of the next population members that have to be generated by successful mating in relation to the total population size. The offspring has to be better in terms of fitness than its parents in order to be termed successful, a strategy which aligns SASEGASA with general simulated annealing techniques by providing a broader search in the beginning and a more constrained search towards the final generations. Synoptically, a portion of the next population, according to the success ratio, has to be filled by such offspring. The actual selection pressure adds this portion to the virtual population to maintain an adaptive nature. If at any point during the search, even

for maximal selection pressure, the successful portion of the population cannot be formulated, i.e. the offspring are not outperforming their parents, stagnation has occurred and reunification must be performed. On average effectiveness and efficiency was found to be better compared to the SEGA model and the standard GA.

The promising performance of SASEGASA and SEGA on the TSP is grounds for similar testing on the JSSP. In order to facilitate better comparisons with generic genetic algorithms a standard GA and an island model are also utilized. The island model uses different populations that evolve independently and exchange chromosomes through a migration operator at certain intervals during execution. All four algorithms utilize a job based representation [17] which can lead to infeasibility and requires an active schedule builder such as the Giffler-Thompson algorithm [7] to repair illegal offspring and create the initial populations. As far as crossover and the chosen representation go two distinct operators are tested: JOX [17] and SSX [9]. JOX maintains the ordering of jobs on machines with little regard for the precedence constraints, while SSX exchanges mere subsequences of operations on machines when they consist of the same job subset. Mutation is a simple shift operator that swaps the positions of two operations.

The experimental results along a sizeable number of benchmark problems, consisting of yet unsolved instances most of which adhere to the 2SETS principle (machines are grouped in two sets and all the jobs must first go through the first set before proceeding to the second) rending them especially difficult, showcase yet again the superior performance of SASEGASA over the remaining algorithms. SEGA also performs extremely well, but requires more time to achieve solution quality directly comparable to that of SASEGASA, whereas the island model has poor effectiveness and even poorer efficiency by having a very large computational overhead due to the inordinate number of evaluated solutions. In conclusion the new GA paradigms based on concepts of segregation and reunification can be competitive with other hybrid metaheuristics without the usage of hill climbers, or any other form of exploitation of problem specific knowledge. This sets the ground for future research directions such as the hybridization of SASEGASA or SEGA with tabu search templates to achieve even higher quality solutions by using one of the former to provide an elite solution set for the latter to operate upon.

References

1. M. Affenzeller. A New Approach to Evolutionary Computation: Segregative Genetic Algorithms (SEGA). *Connectionist Models of Neurons, Learning Procasses, and Artificial Intelligence*, *Lecture Notes in Computer Science*, vol. 2084, pp. 594–601. Springer, 2001.
2. M. Affenzeller and S. Wagner. SASEGASA: A New Generic Parallel Evolutionary Algorithm for Achieving Highest Quality Results. *Journal of Heuristics - Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems*, 10:239–263, 2004.

3. C.A. Brizuela and N. Sannomiya. From the Classical Job Shop to a Real Problem: A Genetic Algorithm Approach. *Proc. of the 39th IEEE Conference on Decision and Control*, 2000.
4. P. Brucker, B. Jurisch and B. Sievers. A Branch and Bound Algorithm for the Job-Shop Scheduling Problem. *Discrete Applied Mathematics*, 49:109–127, 1994.
5. P. Brucker. Scheduling Algorithms (5th ed.). Springer, 2007.
6. M. Dell'Amico and M. Trubian. Applying Tabu-Search to the Job Shop Scheduling Problem. *Annals of Operations Research*, 41:231–252, 1993.
7. B. Giffler and G.L. Thompson. Algorithms for solving production scheduling problems. *Operations Research*, 8:487–503, 1960.
8. A.S. Jain and S. Meeran. Deterministic Job Shop Scheduling: Past, Present and Future. *European Journal of Operations Research*, 113(2):390–434, 1999.
9. S. Kobayashi, I. Ono and M. Yamamura. An Efficient Genetic Algorithm for Job-Shop Scheduling Problems. *Proc. of the 6th International Conference on Genetic Algorithms*, pp. 506–511. Morgan Kaufmann, 1995.
10. M. Kolonko. Some New Results on Simulated Annealing Applied to the Job Shop Scheduling Problems. *European Journal of Operational Research*, 113:123–136, 1999.
11. E.L. Lawler. Recent Results in the Theory of Machine Scheduling. *Mathematical Programming: The State of the Art*, pp. 202–234. Springer, 1983.
12. J.K. Lenstra and A.H.G. Rinnooy Kan. Computational Complexity of Discrete Optimization Problems. *Annals of Discrete Mathematics*, 4:121–140, 1979.
13. A.S. Manne. On the Job-Shop Scheduling Problem. *Operations Research*, 8:219–223, 1960.
14. D.C. Mattfeld. Evolutionary Search and the Job Shop: Investigations on Genetic Algorithms for Production Scheduling. Physica, 1996.
15. E. Nowicki and C. Smutnicki. A Fast Taboo Search Algorithm for the Job Shop Problem. *Management Science*, 42:797–813, 1996.
16. E. Nowicki and C. Smutnicki. An Advanced Tabu Search Algorithm for the Job Shop Problem. *Journal of Scheduling*, 8:145–159, 2005.
17. I. Ono, M. Yamamura and S. Kobayashi. A Genetic Algorithm for Job-Shop Scheduling Problems Using Job-Based Order Crossover. *Proc. of the IEEE Int. Conference on Evolutionary Computation*, pp. 547–552, 1996.
18. J.P. Watson and J.C. Beck. A Hybrid Constraint Programming / Local Search Approach to the Job-Shop Scheduling Problem. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer, 2008.
19. O. Wendt and W. König. Cooperative Simulated Annealing: How Much Cooperation Is Enough. Working Paper 97-19, Institute of Information Systems, Goethe University Frankfurt, 1997.
20. G. Yang, Y. Lu, R. Li and J. Han. Adaptive Genetic Algorithms for the Job-Shop Scheduling Problems. *Proc. of the 7th World Congress on Intelligent Control and Automation*, pp. 4501–4505, 2008.
21. C.Y. Zhang, P.G. Li, Y.Q. Rao and Z.L. Guan. A Very Fast TS/SA Algorithm for the Job Shop Scheduling Problem. *Computers & Operations Research*, 35:282–294, 2008.