

INTEGRATED GENETIC ALGORITHM AND ITS APPLICATIONS FOR CONSTRUCTION RESOURCE OPTIMIZATION

Jin-Lee Kim

California State University, Long Beach
1250 Bellflower Boulevard
Long Beach, CA 90840, USA

ABSTRACT

Construction project resource scheduling problems have been interesting and challenging subjects of extensive research for several decades in the optimization study area in order to put them in practical application. Recently, the integrated genetic algorithm rather than the stand-alone GA is being increasingly applied to solve the problems. An adaptive hybrid genetic algorithm search simulator (AHGASS) for resource scheduling problems has been developed in the previous stage of this research. Previous work outlined the strategies and practical procedures for the algorithm development, but did not deal with algorithm performance with regard to algorithm runtime, especially against runtime used in generating optimality. Since the major drawback of using GA is a great length of time required, it is meaningful to investigate the significance in algorithm runtime between AHGASS and optimality. To address this issue, this paper attempts to investigate the difference in algorithm performance with regard to algorithm runtime.

1 INTRODUCTION

A genetic algorithm (GA) has been continuously utilized as a powerful and applicable optimization tool in civil and construction management research domain to solve the resource-constrained project scheduling problem (RCPSp). The optimal solutions to the problem are of great value to project planners in resolving resource conflicts. Therefore, many researchers attempt to develop the best procedure to be used as a project management tool to allow for the complexities of real-world problems.

A recent trend in solving the RCPSp using GA is to develop an integrated meta-heuristic method by combining GA with other meta-heuristic method. As an effort, the author developed an adaptive hybrid genetic algorithm search simulator (AHGASS) to find an optimal solution to the problem by providing the strategies and practical procedures (Kim and Ellis 2009). The study presented intensive computational results to demonstrate that AHGASS provides a comparable and competitive performance. However, it does not deal with the algorithm performance with regard to algorithm runtime, especially against the runtime used in generating optimal solutions.

Since the major drawback of using GA is a great length of time required, in addition to the difficulty of choosing an encoding and fitness function, it is meaningful to investigate the significant difference in algorithm runtime between AHGASS and optimality. To address this issue, this paper attempts to investigate the difference in algorithm performance with regard to algorithm runtime.

The following section briefly introduces genetic algorithm for construction resource scheduling problems, followed by the strategies and practical procedures of the integrated GA approach for RCPSp. Next, the computational results and analysis section describes data and variables for obtaining algorithm performance and runtime data, immediately followed by statistical results and analysis. This paper then makes concluding remarks that summarize the findings and recommends the further analysis to reinforce the research outcomes.

2 GENETIC ALGORITHM FOR CONSTRUCTION RESOURCE SCHEDULING

Construction project resource scheduling problems have been interesting and challenging subjects of extensive research for several decades in the optimization study area in order to put them in practical application. The generalized model of the resource scheduling is valuable in that it can be incorporated into the advanced computational methods of commercial project management software for practical applications. The resource scheduling problems can be specialized by the objective function, renewable or nonrenewable resources, preemption or nonpreemption (Lee and Kim 1996). The problems have different versions, depending on the objective functions such as (a) the minimization of project duration which is used here, (b) the minimization of the total project cost, (c) the maximization of the net present value of cash flows, and (d) the resource leveling problem. The objective of the resource-constrained project scheduling problem (RCPS) is to allocate the available resources to activities in order to find the shortest duration (makespan, or fitness) of a project network within the constraints of precedence relationships.

Recently, many researchers used meta-heuristic methods such as GA, simulated annealing, tabu search, and ant colonies, to solve the RCPS to overcome the drawbacks of exact solution methods and priority rule-based heuristics. Among them, GA has been successfully used to solve construction resource scheduling problems (Chan et al. 1996; Hegazy 1999; Leu and Yang 1999; Toklu 2002; Hegazy and Kasab 2003; Liu et al. 2005; Kandil and El-Rayes 2006; Jaśkowski and Sobotka 2006; Kim and Ellis 2008, 2009). Kim et al. (2010) studied 87 peer-reviewed prestigious journal articles published over the past 20 years from 1995 to 2010 focusing on areas influenced by genetic algorithm application. The findings indicate that mathematical programming for cost and schedule optimization (40.23%, 35 articles) is the most widely applied area, followed by construction method and process (27.59%, 24 articles), design and layout (21.84%, 19 articles), and management (10.34%, 9 articles). A recent trend in solving the resource scheduling problems using GA is to develop an integrated meta-heuristic method by combining GA with other meta-heuristic method. An integrated GA approach has been successfully applied to many engineering optimization problems such as aerodynamic design, signal analysis, water resources planning and management, and others. Although the integrated GA using meta-heuristic methods are becoming popular in those areas, the approach to the construction resource scheduling problem is demanding more efforts for optimization.

3 INTEGRATED GA APPROACH: LOCAL SEARCH INTEGRATION

GA is adopted in many applications such as education, engineering, government, industry, management, manufacturing, and others. The wide applicability of GA makes the integration possible in addition to its remarkable advantages such as capability and flexibility over exact solution methods and/or priority rule-based methods. It can be integrated with other techniques, for example, mathematical programming and meta-heuristic method in order to solve non-deterministic hard problems. This approach, regarded as the integrated GA, can definitely make a more realistic and promising search solution than stand-alone GA.

AHGASS, which was developed in the previous stage of this research, combines GA with local search using random walk algorithm (RWA) via Baldwinian evolution process. Figure 1 shows a flow chart for AHGASS algorithm. The initial population of possible solutions to the RCPS is created to apply the algorithm in the very first step of global search using GA. A fitness value (project duration) of an individual chromosome in an initial population is calculated, using the serial and/or parallel schedule generation scheme. Evaluation of local search is achieved before the move to the selection operator embedded in the GA. If local search is needed, it occurs following the selection operator. Otherwise, local search is not implemented. The selection of the parent individuals is made through the elitist roulette wheel selection operator for the next generation. The elitist roulette wheel selection operator is the combined operator using the elite selection and the roulette wheel selection. Using the parent individuals obtained from the selection operator, one-point crossover operator is performed by exchanging parent individual segments and then recombining them to produce two resulting offspring individuals. The uniform

mutation operator is performed to play the role of random local search, which searches much smaller portion than the random walk algorithm.

The local search using RWA is performed only when the performance of the global search using GA indicates the needs. The local search integration procedure performs an iterative process on a single schedule of activities. The process attains a better fitness value by taking tiny increments over a local search space. A random activity in an individual chromosome is selected and its flexibility of movement within the schedule is determined. If a required predecessor activity is directly before the activity, and a successor activity is directly after the activity, then it has no flexibility and another activity is chosen. Once chosen, the individual chromosome is adjusted iteratively to put the chosen activity in every one of its possible positions. For each adjustment, the overall fitness value of the individual is computed using schedule generation scheme. The location of the chosen activity that corresponds to the best overall fitness of the individual is chosen as that activity’s permanent location.

If the best location found is different from the activity’s permanent location, then an improvement on the overall fitness value is made. It is important to note that the objective of local search is not to search the best solution located in the neighborhood of the current point, but to provide more information that would better give a direction to the portion of the search space where the globally optimal solution is located. The RWA switches back to global search when the number of iterations exceeds a user-specified maximum value. The major characteristic of AHGASS algorithm is that it adapts in response to recent performance of the algorithm with the aid of local search as it converges to the solution.

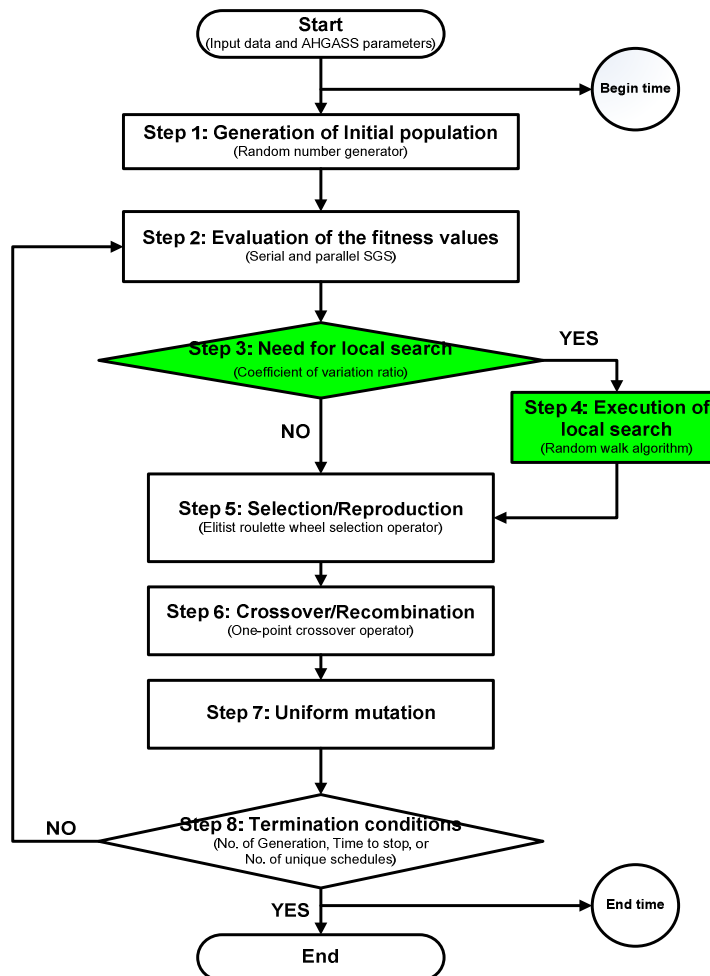


Figure 1: Flow chart for AHGASS algorithm

Figure 2 shows the Baldwinian evolution process for local search. The simple genetic algorithm (SGA) is performed as an initial search of solution space, identifying a limited set of possible solutions or solution regions. Using the solutions obtained from the SGA, the RWA is employed as a starting guess. In order to connect two methods, Baldwinian evolution (Baldwin 1896) is used because it helps avoid premature convergence by preserving only the direction of improvement. It is survival of the fittest following the direction of learning. Therefore, it is no doubt to make certain that only the improved fitness values are changed after local search occurs. However, its genotype information is not altered. The process from the SGA through Baldwinian evolution to the RWA is terminated if it converges to the single point with standard deviation of zero.

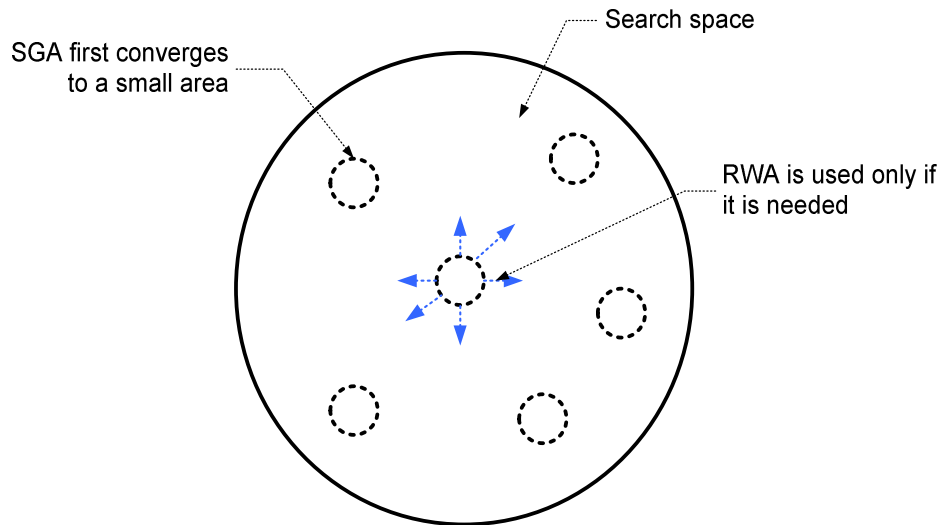


Figure 2: Baldwinian evolution process for local search

4 COMPUTATIONAL RESULTS AND ANALYSIS

4.1 DATA AND VARIABLES

To test algorithm performance with regard to algorithm runtime, especially against runtime used in generating optimality, a computational experiment is designed using standard project network problems available from the PSPLIB. The PSPLIB, Project Scheduling Problem LIBrary generated by Kolisch and Sprecher (1996), is the source that is now widely used in the resource scheduling optimization study area. The experiments consist of twofold: (1) the comparison test in project duration between AHGASS and optimality and (2) the comparison test in algorithm runtime between AHGASS and optimality available from Demeulemeester and Herroelen (1997).

Figure 3 shows the graphical user interface of AHGASS implemented with JAVA programming language. It enables users to input the global and local search parameter values, while the numerous project network problems can be run simultaneously. The parameters and their values to run AHGASS are tabulated in Table 1. The more detail information on parameters can be found in Kim and Ellis (2009). AHGASS is terminated when it meets the number of 100 generations for the experiments.

A paired difference experiment, in which the sample data consist of paired observations randomly selected from a population of paired observations, is used to compare the (1) mean values for the project duration and (2) mean values for algorithm runtime in second through the run of AHGASS. The experiment method is appropriate because it provides more meaningful information about the difference between two means rather than independent sampling procedures since some sampling variability is removed by pairing the observations when possible (Meyer and Krueger 1998; Kuehl 2000).

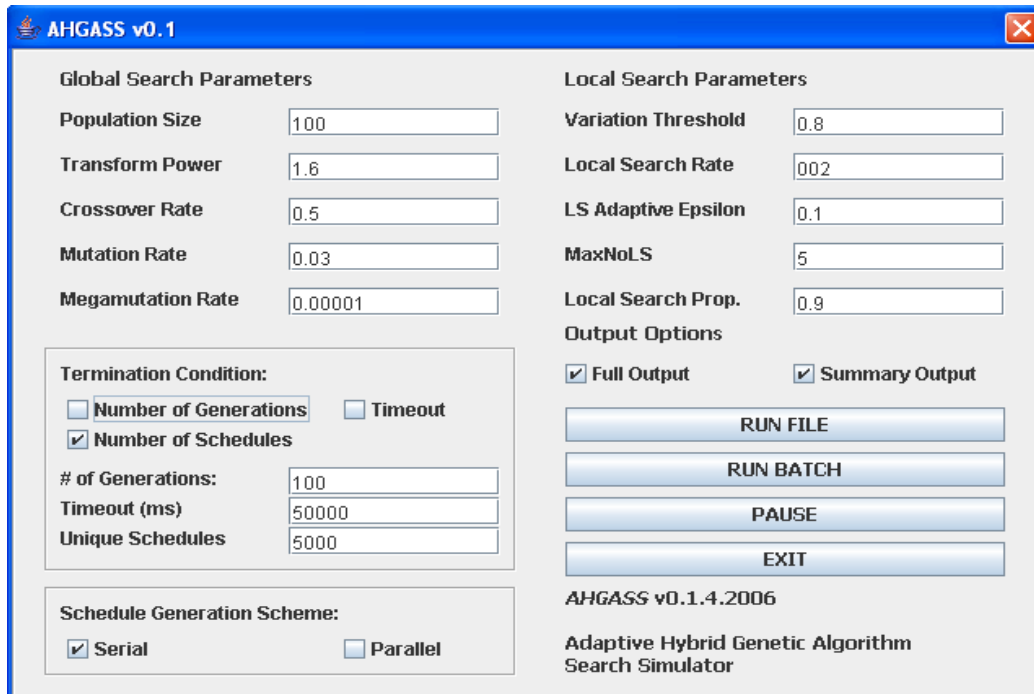


Figure 3: Graphical user interface for AHGASS (After Kim and Ellis 2009)

Table 1: AHGASS Parameters and Values

Global search (GS)	Symbol	Value	Local search (LS)	Symbol	Value
Population size	Pop_size	100	Variation Threshold	T	0.75
Transformation power	TP	1.6	LS probability	Q_o	0.2
Crossover probability	C_p	0.5	LS adaptive parameter	ε	0.5
Mutation probability	M_p	0.03	No. of LS iteration	LS	5
Termination condition	$No. of Gen$	100	LS proportion	LS_p	0.8

4.2 RESULTS AND ANALYSIS

First, the mean values for project durations produced by AHGASS using thirty project network problems are compared against those obtained from the optimality in order to show the robustness of the algorithm. AHGASS was run using thirty project network problems for the first test. Table 2 tabulates the comparison results in fitness values (i.e. project duration) between the optimal solutions obtained from Demeulemeester and Herroelen (1997) and those produced from AHGASS. Four out of thirty project networks revealed the deviation from optimality ranging from 0.02% to 0.09%. In other words, only four pairs resulted in different project durations. This result means that AHGASS produced about 86.67% of optimal solutions for all of thirty project networks.

Second, the mean values for project durations and algorithm runtime produced by AHGASS using 100 project network problems are compared against those obtained from the optimality in order to test the difference in algorithm performance. T-statistic is used to make an inference on the mean of the population differences with the assumption that the distribution of differences is approximately normal. The point estimate of the mean is the mean of the sample difference. The result indicates that, for most pairs of the average fitness values, those produced from AHGASS had same values as those obtained from the optimality. The average deviation from the optimality is 0.43%. Also, the result presents a lot of variation in the average fitness values of the problems. This variation in the values is accounted for by the matched pairs design. A 95% confidence interval for the difference between mean fitness values is constructed.

The point estimate of the true mean difference in the average fitness values is 0.26. With 95% accuracy, the mean difference in the values is between -3.39 and 3.91. The mean difference means that the values produced from AHGASS have the average values from -3.39 to 3.91, which includes the value of zero.

Table 2: Comparison Results for Fitness Values against Optimality

Project ID	Optimality (A)	Fitness by AHGASS (B)	Deviation (%) (B-A)/A x 100	Project ID	Optimality (A)	Fitness by AHGASS (B)	Deviation (%) (B-A)/A x 100
1	43	43	0.00	16	47	47	0.00
2	47	51	0.09	17	47	47	0.00
3	47	47	0.00	18	54	54	0.00
4	62	62	0.00	19	54	54	0.00
5	39	39	0.00	20	43	43	0.00
6	48	49	0.02	21	72	72	0.00
7	60	60	0.00	22	40	40	0.00
8	53	54	0.02	23	57	57	0.00
9	49	50	0.02	24	98	98	0.00
10	45	45	0.00	25	53	53	0.00
11	38	38	0.00	26	54	54	0.00
12	51	51	0.00	27	48	48	0.00
13	43	43	0.00	28	54	54	0.00
14	43	43	0.00	29	65	65	0.00
15	51	51	0.00	30	59	59	0.00

Table 3 tabulates the statistical results for the average fitness values and total algorithm runtime in second as a result of scheduling 100-project networks with 30 activities. The hypotheses to test whether the average fitness value ($\mu_{\text{optimality}}$) obtained from the optimality differs from that (μ_{AHGASS}) obtained from AHGASS are $H_0: \mu_{\text{optimality}} - \mu_{\text{AHGASS}} = 0$ and $H_a: \mu_{\text{optimality}} - \mu_{\text{AHGASS}} \neq 0$. Given that the average project duration for 100 project networks is 54.63 from the optimality, AHGASS produced 54.89 average fitness values, as they converge to a single point across the number of generations of 100. For the test of the average fitness values, we do not reject the null hypothesis because the observed significance level or p-value of 0.888 is greater than $\alpha = 0.05$. Therefore, we do not have sufficient evidence to conclude that the mean values are different between AHGASS and optimality. In other words, AHGASS produces optimal and/or near optimal solutions to the project networks employed in this study.

Table 3: Statistical Analysis for Algorithm Performance

Measurement	Optimality (Demeulemeester and Herroelen 1997)	AHGASS (This paper)	t score (p-value)
No. of project networks	100	100	-
Average project duration	54.63	54.89	0.14 (0.888)
Total algorithm runtime (sec)	1050.23	438.00	1.34 (0.092)

The hypotheses to test whether the average CPU runtime in second ($\mu_{\text{optimality}}$) obtained from the optimality differs from that (μ_{AHGASS}) obtained from AHGASS are $H_0: \mu_{\text{optimality}} - \mu_{\text{AHGASS}} = 0$ and $H_a: \mu_{\text{optimality}} - \mu_{\text{AHGASS}} \neq 0$. Given that the average CPU runtime for 100 project networks is 10.50 second per project network from the optimality, AHGASS produced average CPU runtime of 4.38 second per project network, as they converge to a single point across the number of generations of 100. For the test of the average CPU runtime, we do not reject the null hypothesis because the observed significance level or p-value of 0.092 is greater than $\alpha = 0.05$. Therefore, we do not have sufficient evidence to conclude that AHGASS, integrated GA, takes more time to solve the project network problems than the optimality.

5 CONCLUDING REMARKS

This paper presented the results using the paired difference experiment to compare the mean values for the project duration and mean values for algorithm runtime in second. An adaptive hybrid genetic algorithm search simulator (AHGASS) for resource scheduling problem is used for the comparison study. This paper deals with algorithm performance with regard to algorithm runtime, especially against runtime used in generating optimality. The algorithm performance measures such as project duration and algorithm runtime were compared against the optimality. The results support that the integrated GA approach provides optimal solutions to the RCPSP and that it does not require significant amount of time to solve the problem as compared to the runtimes used in generating the optimality. The intensive statistical analysis is necessary to prove that the integrated GA approach can definitely make a more realistic and promising search solution than stand-alone GA.

REFERENCES

- Baldwin, J. M. 1896. *A new factor in evolution*. American Naturalist, 30: 441-451.
- Chan, W., D. K. H. Chua, and G. Kannan. 1996. Construction resource scheduling with genetic algorithms. *Journal of Construction Engineering and Management*. 122(2), 125-132.
- Demeulemeester, E. L., and W. S. Herroelen. 1997. New benchmark results for the resource-constrained project scheduling problem. *Management Science*, 43(11): 1485-1492.
- Hegazy, T. 1999. Optimization of resource allocation and leveling using genetic algorithms. *Journal of Construction Engineering and Management*. 125(3), 167-175.
- Hegazy, T., and M. Kassab. 2003. Resource optimization using combined simulation and genetic algorithms. *Journal of Construction Engineering and Management*. 129(6), 698-705.
- Jaśkowski, P., and A. Sobotka. 2006. Scheduling construction projects using evolutionary algorithm. *Journal of Construction Engineering and Management*. 132(8), 861-870.
- Kandil, A., and K. El-Rayes. 2006. Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects. *Journal of Construction Engineering and Management*. 132(5), 491-498.
- Kim, J.-L., E. Noornejad, and A. Koofigar. 2010. Integrated genetic algorithm application in construction research. *Unpublished manuscript*, Department of Civil Engineering and Construction Engineering Management, California State University, Long Beach, CA.
- Kim, J.-L., and R. D. Ellis. 2008. Permutation based elitist genetic algorithm for multiple resource constrained project scheduling problem. *Journal of Construction Engineering and Management*. 134(11): 904-913.
- Kim, J.-L., and R. D. Ellis. 2009. Robust global and local search approach to resource-constrained project scheduling. *Canadian Journal of Civil Engineering*. 36(3): 375-388.
- Kolisch, R., and A. Sprecher. 1996. PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, 96: 205-216.
- Kuehl, R. O. 2000. *Design of experiment: Statistical principles of research design and analysis*. 2nd ed., Duxbury Press, Pacific Grove, CA.
- Lee, J.-K., and Y.-D. Kim. 1996. Search heuristics for resource-constrained project scheduling. *The Journal of Operations Research Society*, 47(5), 678-689.
- Leu, S., and C. Yang. 1999. GA-based multicriteria optimal model for construction scheduling. *Journal of Construction Engineering and Management*. 125(6), 420-427.
- Liu, Y., S.-L. Zhao, X.-K. Du, and S.-Q. Li. 2005. Optimization of resource allocation in construction using genetic algorithms. *Proc. 2005 Int. Conf. on Machine Learning and Cybernetics*. 6(18-21), 3428-3432.
- Meyer, R., and D. Krueger. 1998. *A Minitab guide to statistics*. Prentice Hall, Upper Saddle River, NJ.
- Toklu, Y. C. 2002. Application of genetic algorithms to construction scheduling with or without resource constraints. *Canadian Journal of Civil Engineering*. 29, 421-429.

AUTHOR BIOGRAPHIES

JIN-LEE KIM, Ph.D., P.E., is an assistant professor of Dept. of Civil Engineering & Construction Engineering Management at California State University, Long Beach. He received his BE and first ME degrees in Architectural Engineering from Chungbuk National University, Korea and his second ME and PhD in Civil Engineering from University of Florida. He is a registered professional engineer in Florida. His research interests include information technology in construction, simulation-based resource scheduling, optimization techniques, building information modeling, and sustainable design and construction. He is a member of ASCE. His email address is jin5176@gmail.com.