Infrastructure Work Order Planning Using Genetic Algorithms

E. William East
U.S. Army, Corps of Engineers, Construction Engineering Research Laboratories,
CECER-PL-E, 2902 Newmark Drive, P.O. Box 9005, Champaign, IL 61826-9005.
b-east@cecer.army.mil, 217-373-6710

ABSTRACT

Infrastructure management offices plan and complete several thousand small construction projects annually. Effective planning is vital if the public and private sectors are to maintain valuable infrastructure investments at the least cost to the taxpayer or shareholder. This paper presents the results an application of Genetic Algorithms (GA) in multi-project resource allocation to minimize the total cost of work order execution on realistically sized infrastructure management problems. In addition to direct crew costs indirect costs for set-up, idle time, and travel are included in this model. Results of test cases demonstrate the effectiveness of the approach when compared to several standard heuristics.

1. Introduction

The typical infrastructure management process begins with the submission of work orders from customers scattered across the facility. Work orders may also be created periodically for equipment and system maintenance. The set of projects to be accomplished is then distributed to shop foremen. The number, type, and location of these work orders are unknown, however, several thousand projects will be completed at a large facility each year.

Shop foremen are usually responsible for the specific allocation of crews and equipment to each project. Crews are composed of groups of workers with similar skills such as carpenters, electricians, and plumbers. There may be more than one crew assigned to important activities, while some other work may be accomplished by outside contract work crews. As a crew is assigned to complete a project, that crew must identify the scope of the work and obtain the materials and equipment necessary to complete the job.

Several times during the course of a typical business day, an infrastructure manager may need to re-prioritize projects based on the nature of incoming projects. Projects that threaten life-safety have first priority and require the immediate reallocation of crews, interrupting ongoing projects and delaying regularly scheduled jobs. Next highest priorities are those projects that return facilities to a normal operational state. These projects also require the ad-hoc reallocation of resources. Meeting scheduled project milestone dates is the next highest priority, followed by projects that allow coordination with other crews. If resources remain available, then additional jobs are assigned based on crew foreman's preference.

In most infrastructure management offices resource allocation decisions are made by individual shop foremen, without complete knowledge of the actions of other foremen. Anecdotal evidence, obtained by interviews, suggests that at least ten percent of the resources available to the infrastructure manager are wasted due to workers arriving to the job site when needed prior work has not been completed. The most common reason for this out-of-sequence work appears to be miscommunication among people responsible to allocate crews. Additional efficiencies could also be gained from sequencing multiple projects in the same, or related, physical locations to eliminate duplicated work or re-work required in a proportion of jobs.

The purpose of this paper is to document the results of an experiment conducted into the use of Genetic Algorithms (GA) to solve realistically sized, reduced complexity infrastructure management problems. The paper begins with a brief description of previous work. Design of the test program and processing operators are described. Comparison of theoretical values for GA parameters are compared to those used for test runs. The performance of the GA over a set of randomly created test cases is described. Finally, run results are compared to solutions developed using appropriate standard heuristics.

2. Previous Methods

Research into problems related to the infrastructure management domain has been ongoing for about fifty years. There are several basic themes in this large body of work. The first 'theme' is mathematical programming. Application of Integer Programming approaches to project management problems began the formal study the construction management [Brand 1964]. In terms of standard operations research problems the multi-project resource allocation problem addressed in this paper might be called a 'traveling job-shop scheduling problem'.

Unfortunately mathematical programming techniques have been proven to perform poorly for problems of modest size [Ullman 1976]. A more general criticism of mathematical programming is that these approaches attempt to develop single optimal decisions in situations where decisions are often anything but objective [Holloway 1979].

The second 'theme,' heuristics, is a popular alternative to mathematical programming [Davis 1975]

[Ozdamar 1995]. Heuristic programming attempts to find a reasonable answer through the application of 'rules-of-thumb'. Symbolic heuristics attempt to create 'knowledge-bases' by coding human experience. Typical heuristic applications often require prohibitively expensive subject matter expert and management input, tend to perform poorly for general problems, and do not support the variety of constraints needed to solve realistic problems [Vepsalainen 1987].

A third 'theme' attempts to learn an internal representation based on examples. Applications of one of these approaches, neural networks, related to construction planning include selection of equipment for specific types of construction [Alsugair 1994] and creation of detailed excavation plans [Chao 1998]. While successes have been noted in diagnostic/classification problems using machine learning techniques, the jury is still out on the ultimate impact of machine learning approaches to generation of new, innovative resource allocation plans.

The fourth 'theme' has been techniques that use selection and recombination to manipulate problem-sub structures to progressively find better solutions. Such Genetic Algorithms (GA) have proven successful in the related traveling salesman and job shop scheduling problems [Whitley 1998]. General single project allocation problems and related applications in other domains have been addressed by a number of authors [Cheng 1994] [Tanomaru 1995]. Multiply constrained resources for single projects has also been investigated [Ramat 1997]. Use of GA in the construction management domain has been demonstrated in applications of single project resource allocation problems [Chan 1996], simultaneous resource allocation and leveling problems [Grobler 1995], and the time-cost tradeoff problem [Feng 1997].

GA research has demonstrated the effectiveness of GA's to solve many canonical operations research problems and small construction management test projects. Results of GA research in construction management resource allocation problems have not included multiple projects, realistically sized data sets, or realistic constraint sets. Such is the 'meat' of the investigation currently being conducted by the author, the first course of which is provided below.

3. Test Program Design

The test program was developed using the Microsoft, Visual Basic (v 6.0 service pack 1) programming language. This programming language was chosen to take advantage of legacy source code. Programming was accomplished on a Pentium 166 MHZ lap top computer running Windows 95.

The test program includes a number of plug-in applications that handle the display of schedule, processing, and reporting data. For display of projects and activities in a bar chart format the AddSoft Gantt control was used. For iterative display of GA processing results Olectra Chart was employed. Report data was formatted using Seagate's Crystal Reports.

3.1 Chromosome Representation

While results from literature are insufficient to define a general 'best' representation for scheduling problems, order-based representations appear to capture an "essential" qualitative feature of scheduling domains [Mattfeld 1996]. An order-based chromosome is used to represent projects in the test program. Each chromosome represents one complete plan for scheduling all projects. Each locus in the chromosome represents a project. The alleles provide the relative order of that project compared to other projects.

Order-based representations often allow alleles to take on integer-only values. Such representations require specialized operators to ensure feasible solutions [Murata 1996]. Alleles in the test program take fractional values, or 'random keys' [Norman 1997]. The relative order of the allele value specifies the order of the projects.

3.2 Fitness Function

The objective of practicing infrastructure managers is to accomplish the most work within the available budget to the satisfaction of the majority of the office's clients. In this test program that objective was simplified to minimize cost of the projects to be scheduled. Every chromosome, or project plan, is evaluated to determine the cost of performing that plan.

The fitness function begins by decoding the project ordering data contained in the chromosome using a Critical Path Method (CPM) scheduling algorithm. This algorithm allocates the first available crew to activities, according to the activities' early start dates. Alternate allocation assignments based on late start date or total float dates are possible, however, on projects with crew-sequential activities, as is the general case in the infrastructure management domain, there is no difference between method of the activity allocation.

Once a project plan has been decoded into a resource allocation plan the direct and indirect cost may be developed and summed to determine the plan's total cost. The direct cost of the selected crew for an activity (with a specific work type) is found from a lookup table and multiplied by the number of days for each specific activity. It is assumed that the cost of material, labor, and equipment are included in the daily cost of the allocated crew. Activities requiring large equipment or material components are decomposed into separate parallel activities to represent each of the 'driving' resources.

The test program allows multiple crews to perform the same work type. Using such an arrangement the cost of individual crews may be modeled on the specific make up, experience, or historical performance of the crew with specific work types. The test program does not allow more than one crew to be allocated to a given crew. Shift work was not included in the test program.

Since the direct costs to perform a set of work with a single crew per work type will be the same regardless of the ordering of the projects, indirect costs must be considered to assess the relative value of project plans.

The first indirect cost considered is that associated with crew retooling and restocking. Such 'set-up' or mobilization costs are required when crews change work types. This cost represents the cost of restocking vehicles with needed materials and/or tools. The result of this penalty is that resource allocations that allow crews to focus on specific types of work for longer periods of time should be preferred to crews that daily change work types. It is assumed that restocking is accomplished at the beginning of each workday and does not impact the duration of given activities.

Since infrastructure projects occur in physical space the travel time between jobs is another indirect cost included in the test program. The result of such a travel penalty is that resource allocations that allow workers to complete several projects in one work area before moving to a different site should be preferred over allocations that have crews move to different work areas every day. As with the retooling fee, travel time in the test program does not delay the completion of a task but simply adds indirect cost to a work plan. Such an assumption is valid for relatively compact geographical regions such as a single college campus or military base.

One management consideration applied to each project plan measures the idle time. The result of the idle time penalty is that crew allocations that are 'level' are preferred over crew allocations that contain work that starts and stops. Consistent resource allocations allow managers to assign crews with idle time to other special projects and may provide clues about the crews that may be outsourced.

3.3 Selection Mechanism

To ensure that a consistent selection pressure is applied throughout a test program run, a pair-wise tournament selection procedure was used in the test program. Processing in the test program proceeds through each generation by (1) evaluating population fitness and calculating descriptive statistics, (2) performing elitist selection, (3) creating the remainder of the next generation, (3) performing cross over on the new generation, and (4) performing mutation on the new generation.

3.4 Population Seeding

In order for the GA to take advantage of existing heuristic methods users of the test program may elect to seed the initial population with heuristic solutions. Heuristics included in the test program were the First-In-First-Out, Last-In-Last-Out, Maximum Resource Utilization and Minimum Resource Utilization heuristics. The inclusion of other heuristics was not appropriate since data on which to generation solutions, such as project due dates, was not included in the test sets. For example weighted tardiness heuristics could not be included since the relative priority of projects, and subsequent due date assignment, is not a component of the current test program.

Restricting the list to the four selected heuristics also has some value since the first two heuristics (along with user-defined priority assignment) are the heuristics used in commercial infrastructure management software.

3.5 Elitism

GA's are stochastic processes that may, on occasion, ignore good solutions. To combat this problem methods that allow for overlapping populations have been developed. In these methods, a percentage of the best chromosomes from the current generation are copied, as is, to the next generation. The test program provides a single, the best, member of the current population to be copied to the next generation. Such a single-individual procedure is referred to as an 'elitist' selection strategy.

In the test program, the use of the elitist strategy is an option that the user may apply if desired. In the test program, the elitist selection does not protect the best of generation from cross over or mutation in the current generation.

3.6 Cross Over and Mutation Operators

The test program allows the user to select from one of four types of cross over and between two mutation operators. The cross over operators implemented in the test program are: left- and right-handed one-way cross over, inside and outside two-way cross over. A random selection of one of the four cross over operators is also an option. Good building block of shorter length are best preserved with one-way cross over, those of longer length best preserved with two-way cross over. Since there may be multiple levels of building blocks being identified and refined during GA operations on realistically sized problems one cannot determine *apriori* the best cross over operator to be used.

The test program implements a random allele swapping and adjacent allele swapping procedures for mutation operators. As with the cross-over option the user may also select to randomly choose either the random or adjacent swapping procedure.

4. Test Case Description

Test cases were developed by refining a set of representative test projects from data provided by the University of Illinois Operations and Maintenance Department and the Department of Public Works from the U.S. Army base at Fort Gordon, GA. Both data sets contained two types of projects service order projects and work order projects. Service order projects were typically single crew, single day activities. Work order projects contained multiple, single crew, activities that were to be completed in sequence. Nine work order-only cases and nine mixed work order and service order cases were developed by randomly selecting projects from the appropriate set of generic test projects.

Projects within test cases were distributed among one of three types of geographical layouts each containing five nodes. The 'linear' procedure assigned jobs to work areas along a straight line. The 'cross' procedure assigned jobs to work areas in an 'x' layout. The 'grid' procedure

assigned jobs to an 'x' layout then connected the adjacent nodes of the 'x'.

The procedure of assigning jobs to locations also considered if the distribution of the projects should be evenly distributed among the nodes, centrally distributed, or distributed more to the outside points of the locations.

Eighteen test cases of 50 projects each were used to evaluate the performance of the Resource Manager. The 50 project set was selected be similar to the number of projects that need to be concurrently scheduled under actual conditions. Each test set of had a total of $3*10^{64}$ possible orderings. Test cases characteristics are provided in the six, left-most columns of Table 1.

The 'Project Statistics' columns in Table 1 identify the number of activities in each of the sets of 50 projects. Mixed work order and service order test cases have on average 2/3 fewer activities than those test cases that were solely comprised of work order projects. The connectivity of activities only slightly increased between the two types of project sets since the order among work order projects is typically sequential

Table 1. Test Cases, Description & Result.

Run	Project Set Description			Project Statistics		Results		
	Job Type	Layout	Distribution	Activities	Priors	Time (h:m)	Difference	Percent
1	Mixed	Linear	Distributed	137	88	0:48	$23,049	16%
3	Mixed	Linear	Dispersed	125	79	0:46	$21,303	18%
4	Mixed	Linear	Central	140	92	1:45	$25,228	19%
6	Mixed	Cross	Central	127	82	0:46	$11,493	10%
8	Mixed	Cross	Distributed	117	72	0:42	$13,930	13%
10	Mixed	Cross	Dispersed	137	89	0:50	$21,351	16%
11	Mixed	Grid	Central	126	82	0:46	$18,805	18%
13	Mixed	Grid	Distributed	130	82	0:46	$20,805	17%
15	Mixed	Grid	Dispersed	148	104	0:52	$31,765	22%
2	Work Orders	Linear	Central	216	170	2:18	$29,778	15%
5	Work Orders	Linear	Distributed	220	176	2:22	$47,821	22%
7	Work Orders	Linear	Dispersed	228	184	2:25	$37,419	15%
9	Work Orders	Cross	Central	215	171	2:25	$37,438	17%
12	Work Orders	Cross	Distributed	199	152	1:27	$30,017	15%
14	Work Orders	Cross	Dispersed	229	185	2:38	$43,864	18%
16	Work Orders	Grid	Central	230	186	2:44	$66,575	23%
17	Work Orders	Grid	Distributed	207	161	1:13	$28,118	15%

4.1 Population Sizing and Parameters

Each test was run for 50 generations. The length of the run was set to provide a consistent comparison between runs. Typically, convergence occurred within the first thirty generations. The population size was initially set to 100 members to provide data on initial population distribution and signal difference from which the theoretical population sizing model could be evaluated.

Theoretical results have provided a 'signal to noise' formulation for population sizing shown in Equation 1 [Harik 1996]. In applying the population-sizing model, a number of the terms must be estimated or derived from data. The first term to be estimated based on general knowledge about the project is that the size of building blocks. Crews will, in general, be assigned to 5 projects per week since the typical duration for tasks is one day. If this domain knowledge is used to set the building blocks, then the size of the building blocks, *k*, is 5. For a chromosome composed of 50 projects the number of building blocks in the population, *m*, is 10. The *m'* term is one less than the total number of building blocks.

Equation 1. Population Sizing Model.

$$n = -2^{k-1}\ln(\alpha)\frac{\sigma_{BB}\sqrt{\pi m'}}{d}$$

The α term is the chance of error in the model. For a 95% chance of success, α is set to 0.05. Results from the first generation of the 18 test cases allow us to estimate the signal difference, *d* and the standard deviation of building block fitness, σ_{BB}. The average fitness difference in the 18 test cases was 4,702. The standard deviation of building blocks may be estimated by taking the square root of the population variance divided by the number of building blocks, *m*. Data from the first generation of all 18 test cases was again used to develop the initial population variance resulting in a σ_{BB} of 5766.

Applying these estimated values to the population sizing model yields a suggested population size of 117. The theoretical estimate of population sizing is very close to the 100 member population size used in the 18 test cases documented above.

Experiments with population sizes of 200 and 500 were conducted but did not yield results that appeared to be significantly different from those results provided by the 100 member population. As a result, runs using these large populations were not documented in this paper.

The upper limit for cross over rate is that which ensures that good building blocks, over the population as a whole, are kept in the population. This boundary shown in Equation 2, derived from the Schema Theorem assumes that we want to not have every cross disrupt each of building block we want to keep [Goldberg 1998]. Application of the upper bound cross over operator with pair-wise tournament selection, and single member elitist selection yields a selection pressure, *S*, of 3. The upper bound on cross over should, therefore, be 0.6.

Equation 2. Cross Over Upper Bound.

$$p_c \leq \frac{S-1}{S}$$

The cross over rate was set to 0.6 based on an evaluation of the theoretical upper bound on the cross over rate. The theoretical lower bound for cross over was not calculated since the upper bound value was used. The cross over type was randomly selected from the four cross over operators implemented in the test program.

The mutation rate was set to an average of two mutations per 100-member generation. Such a mutation rate is less than that which disrupts good building blocks while adding some alternative solutions. Given that there is no information to choose between the adjacent and non-adjacent mutation operators, mutation operators were randomly selected during the run.

4.2 Experimental Results

A complete program trace file and 'GA Results Per Generation' report were produced for each of the 18 test cases. The results of the runs, extracted from the trace files, are provided in Table 1 and Table 2. The GA was able to better all initial solutions an average of 17% over the initial solutions for both the mixed and work order test sets.

The time of processing columns in Table 1 shows the time required to process each of the test runs. Two different machines were used for the test runs. Both machines were Intel Pentium II, single processor systems, running Windows NT 4.0 (service pack 3). Runs 1, 3, 6, 8, 10, 11, 13, 15, 12, 17, and 18 were processed on a machine with a 450MHZ processor only running the test cases. Runs 2, 4, 5, 7, 9, 14, and16 were executed on a 400MHZ processor simultaneously running a variety of business and computer programming applications.

The data provided in Table 1 demonstrates that the GA is 'innovative' on the initially random solutions set. These results are of little value to the project manager unless the solutions are much better than standard, and immediate, heuristic solutions.

The costs of project plans from each of the four heuristics are shown in Table 2 along with the resulting GA solution. The difference between the best heuristic, shown in italic typeface, and the final GA solution after 50 generations is shown in dollars and percentage difference. The GA solution cost, on average, about 1/4 less than the least costly heuristic solution. In terms of costs the GA solution costs, on average, $52,000 less than the best heuristic solution.

Using the initially randomized GA population alone, the savings over heuristic method averaged to over $24,000 per project plan. During the first third of the GA runs, after which users would normally have stopped the run, there was an average saving of $1,000 per minute of GA processing.

5. Critique and Discussion

There are a variety of outstanding issues and possible critiques that could be raised by the results described in the previous section. These issues include: (1) inability of the current work to allocate resources for multiple large construction projects, (2) inadequately representation of realistic infrastructure management domains, (3) unrealistic test cases, (4) uninformed GA processing, and (5) inadequate heuristics operators used for comparison. A discussion and proposed resolution of each of these critiques is provided in the following paragraphs.

While the results provided in this paper can provide hope for infrastructure managers these results may not be applicable to more general construction management scheduling problems. In the infrastructure management domain the ratio of activities to projects is no greater than 10. The average value may be closer to 5 activities per project. In the general contractor multi-project resource problem the number of activities per project may exceed 1000 activities per project.

The next critique of the current work is that the current GA model does not adequately represent the infrastructure management domain. Since the results presented in this paper appear positive extensions to the current work are planned. First, a means to allocate multiple crews to individual activities will need to be considered. This is important to address production 'bottlenecks' or idle work 'sinks'. Shift work may also be considered as part of this first enhancement. Given that multiple crews will be assigned, planed activity duration will need to be dynamically assigned. In addition, the 'grain-size' of task duration will be reduced to the typical one-hour unit of crew charge time.

Table 3. Comparison of Heuristic and GA Results.

Run	Heuristic Results (best is bold)				GA Soln		
	LIFO	FIFO	MinRes	MaxRes	Value	Difference	%Better
1	$168,444	$176,465	$175,874	**$162,589**	$121,450	$41,139	25%
3	$158,667	$149,842	**$136,037**	$144,801	$98,371	$37,666	28%
4	$162,875	$172,889	$165,807	**$158,950**	$109,948	$49,002	31%
6	**$114,445**	$138,474	$147,856	$136,086	$102,952	$11,493	10%
8	$154,356	$135,080	**$115,861**	$127,915	$91,320	$24,541	21%
10	$191,454	$161,224	$172,896	**$156,884**	$116,063	$40,821	26%
11	**$140,806**	$147,581	$146,116	$126,346	$84,731	$56,075	40%
13	$149,572	$159,972	$151,517	**$147,787**	$101,732	$46,055	31%
15	$167,134	$218,184	$180,249	**$165,879**	$110,479	$55,400	33%
2	$249,994	$291,322	**$234,247**	$265,953	$175,379	$58,868	25%
5	$299,108	$301,475	**$223,635**	$245,568	$170,112	$53,523	24%
7	$342,589	$447,836	$341,544	**$282,059**	$217,890	$64,169	23%
9	**$220,311**	$409,248	$238,502	$301,470	$182,873	$37,438	17%
12	**$225,055**	$290,297	$245,050	$300,291	$172,803	$52,252	23%
14	**$286,098**	$351,344	$327,629	$316,445	$205,316	$80,782	28%
16	$335,888	$514,993	**$310,430**	$369,658	$227,817	$82,613	27%
17	$200,003	$211,162	$198,599	**$191,509**	$155,394	$36,115	19%
18	$329,648	$414,039	**$313,718**	$380,724	$227,393	$86,325	28%

Addition of heuristics to consider multiple crew allocations is possible. Such a hybrid GA-heuristic system may choose, for example, to allocation multiple crews if additional crews are idle for longer than the cost of travel to the job. Other rules may determine if 'outside', more expensive crews should be used to clear bottlenecks. The decision of which rules to apply may be a feature of a revised GA representation or a user-specified user specified value.

The second change needed to provide a better reflection of the actual infrastructure management domain is that of project priority. Priority is needed to distinguish between emergency and non-emergency projects. Use of project priorities is expected to simplify GA processing since priority sets will partition the solution space. Other changes planned are to add budgetary constraint to show the cut-off point for planned work and penalties for starting or finishing work outside required milestone dates.

The next critique of the work presented in this paper is that the test cases used are unrealistic. Since the test projects and activities were developed from actual infrastructure management databases such a critique could only be based on the random assignment of the projects to work areas. In a realistic infrastructure management setting it is likely that projects will not be created in a random order but have some underlying patterns. Since GA's are able to exploit such underlying patterns the randomly generated cases used in the tests should be more difficult than the project sets presented in an actual infrastructure management setting. Consideration of the 'unrealistic test case' critique asserts that if the problem were easier, the GA would produce better results.

The fourth critique of the GA design is that the GA processing procedure could be improved. If heuristic operators were used to group projects according to similarity of work type and location, then better results should be expected. Such a heuristic seeding procedure is planned for the next version of the test program. Alternative formulations to improve the linkage will also be included in the next prototype. Such a formulation will more tightly couple, or link, groups of projects that should be completed in a relative sequence to one another. A temperature-based method for varying mutation rates will also be implemented.

Finally, additional heuristics should be used to test the results of this work. The addition of new heuristics, once the test program has sufficient depth to accommodate such heuristics, should assist in validating the planned future changes.

6. Summary

The objective of this paper was to document the results of an experiment conducted into the use of Genetic Algorithms (GA) to solve a realistically large but simplified subset of the infrastructure management problem. The paper began with a brief description of previous work. Design of the GA and processing operators were described. The performance of the GA over a set of 18 randomly generated test cases was described. Theoretical bounding values for adequate population sizing compared favorably with the values used during test runs. Finally, run results were compared to a

small set of heuristic solutions and the results of the work were critiqued.

7. Conclusions

Based on the results of the experiment presented in this paper, GA's have been shown to provide good solutions to realistically sized but simplified multi-project resource allocation problems in the infrastructure management domain. Processing times are such that infrastructure managers may develop daily plans that take into account new projects and changed resources. Proposed enhancements promise to provide resource allocation tool that may be applied to realistic infrastructure management problems.

Acknowledgements

The author would like to acknowledge the contributions of Professors, Liu, Melin, Palmore, Hall, and Goldberg for their suggestions and contributions to this ongoing project. The author would also like to acknowledge the donation of computer time from of the U.S. Army, Construction Engineering Research Laboratories without whom the experiments described in this paper could not have been completed.

Bibliography

Alsugair, Abdullah, Chang, Davide Y., (1994) “An Artificial Neural Network Approach to Allocating Construction Resources,” 1st Congress on Computing in Civil Engineering, American Society of Civil Engineers, pp. 950-957.

Brand, J.D., Meyer, W.L. and Shaffer, L.R. (1964) “The Resource Scheduling problem in Construction,” Construction Research Series (2), Department of Civil Engineering, University of Illinois.

Chan, Weng-Tat, Chua, David K.H., Kannan, Govindan (1996) "Construction Resource Scheduling with Genetic Algorithms," American Society of Civil Engineers, Journal of Computing in Civil Engineering, 122(2), pp. 125-132.

Chao, Li-Chung (1998) "Estimating Excavation Duration: OOP Plus NN Approach," Proceedings of the 1998 International Computing Congress, American Society of Civil Engineers, pp. 644-651.

Cheng, Runwei, Gen, Mitsuo (1994) "Evolution Program for Resource Constrained Project Scheduling Problem," Proceedings IEEE World Congress on Computational Intelligence, v. 2, pp. 741-736.

Davis, E.W., Patterson, J.H., (1975) “A Comparison of Heuristics and Optimum Solutions in Resource Constrained Project Scheduling,” *Management Science*, v. 21, pp. 994-955.

Feng, Chung-Wei, Liu, Liang, Burns, Scott A. (1997) “Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems,” Journal of Computing in Civil Engineering, American Association of Civil Engineers, 11(3), pp. 184-189.

Goldberg, David (1998) "The Race, the Hurdle, and the Sweet Spot: Lessons From Genetic Algorithms for the Automation of Design Innovation and Creativity," Report Number 98007, Illinois Genetic Algorithms Laboratory, University of Illinois, pg. 6.

Grobler, Francois, Kannan, G., Subick, C., Kargupta, H. (1995) “Optimization of Uncertian Resource Plans with Genetic Algorithms,” Proceedings of the 2nd International Congress on Computing in Civil Engineering, American Society of Civil Engineers, Atlanta, GA.

Harik, Georges, Cantu-Paz, Erick, Goldberg, David E., Miller, Brad L. (1996) “The Gambler’s Ruin Problem, Genetic Algorithms, and the Sizing of Populations,” Report Number 90004, Illinois Genetic Algorithms Laboratory, University of Illinois, pg. 6.

Mattfeld, Dirk (1996) Evolutionary Search and the Job Shop: Investigations on Genetic Algorithms for Production Scheduling, Physica-Verlag, pg. 74.

Murata, Tadahiko, Ishibuchi, Hisao (1996) "Positive and Negative Combination Effects of Crossover and Mutation Operators in Sequencing Problems," Proceedings of the Interational Conference on Evolutionary Computing.

Norman, Bryan, A., Bean, James C. (1997) "Random Keys Genetic Algorithm for Job-Shop Scheduling," Engineering Design and Automation, 3(2), pp. 145-156.

Ozdamar, Linet, Ulusoy, Gunduz (1995) “A Survey on the Resource-Constrained Project Scheduling Problem,” IIE Transactions, v. 27, pp. 574-586.

[Ramat 1997] Ramat, E., Venturini, G., Slimane, M. (1997) "Solving the Multiple Resource Constrained Project Scheduling problme with a Hybrid Genetic Algorithm," Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kaufman, San Fransisco, CA, pp.489-496.

Tanomaru, Julio (1995) "Staff Scheduling by a Genetic Algorithm with Heuristic Operators," Proceedings IEEE International Conference on Evolutionary Computing, pp. 456-461.

Ullman, J. D. (1976) “Complexity of Sequencing Problems” in Coffman, E. G. ed. Computer and Job-Shop Scheduling Theory, Wiley & Sons, New York, pp. 139-164.

Holloway, Charles (1979) Decision Making Under Uncertainty: Models and Choices, Prentice Hall, pg. 19.

Vepsalainen, Ari Pl J., Morton Thomas E. (1987) "Priority Rules for Job Shops with Weighted Tardiness Costs," Management Science, 33(8), August, pg. 1046.

Witley, Darrell, Startkweather, Timothy, Fuquay, D'Ann (1989) "Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator," Proceedings of the International Conference on Genetic Algorithms, pp.133-140.