Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

SELECTING EARTHMOVING EQUIPMENT FLEETS USING GENETIC ALGORITHMS

Mohamed Marzouk
Osama Moselhi

Department of Building, Civil, and Environmental Engineering
Concordia University, 1455 de Maisonneuve West
Montreal, Quebec H3G 1M8, CANADA

ABSTRACT

This paper presents an application of simulation optimization in construction utilizing genetic algorithms. The paper focuses on the use of genetic algorithms (GAs) as a tool for optimizing the total cost of earthmoving operations accounting for available equipment models to contractors and their corresponding quantities. The developed genetic algorithm has a powerful computational utility that increases its efficiency. The fitness of generated chromosomes is calculated utilizing a simulation engine dedicated for earthmoving operations which is dynamically linked to the developed genetic algorithm. The impact of the algorithm's control parameters on its conversion is also examined. A numerical example is presented to illustrate the capabilities of the developed algorithm in selecting near-optimum fleet configurations.

1 INTRODUCTION

Optimizing the total cost of construction operations is considered a major challenge for construction contractors. This cost influences contractors' decision to bid or not. Direct cost is typically estimated for the involved resources including labor, material, and equipment. In earthmoving operations, the direct cost is basically equipment cost. Therefore, assigning the right equipment fleet configuration is a key factor for the project success. GAs have been applied in different domains including equipment selection (Marzouk and Moselhi 2001-a); time-cost optimization (Li and Love 1997); water network rehabilitation (Halhal et al 1997); and cost optimization of composite floors (Kim and Adeli 2001).

This paper presents a methodology for optimizing total cost of earthmoving operations utilizing computer simulation and genetic algorithms (GAs). The paper first presents an overview of the simulation engine that is utilized to estimate the fitness of generated chromosomes. It then describes the different features and characteristics of the developed genetic algorithm. A numerical example is presented to illustrate the capabilities of the developed methodology and to study the convergence and performance of the developed genetic algorithm under different input parameters.

2 SIMULATION ENGINE

The recently developed simulation engine (*EMSP*) (Marzouk 2002, Marzouk and Moselhi 2000-a) is used for estimating the fitness of generated chromosomes in the proposed genetic algorithm. The engine has been developed utilizing discrete event simulation and object-oriented modeling and implemented in Microsoft C++ 6.0. It utilizes different features of object orientation such as classes, inheritance, and dynamic data structure.

The classes used in the design of *EMSP* are of two types: auxiliary and main (Marzouk and Moselhi 2000-b). Auxiliary classes are connected to the main classes through either association or aggregation relationships, whereas, the main classes are connected to each other through inheritance relationships. The main classes of *EMSP* capture different situations according to the activities involved. Therefore, they represent different combinations of earthmoving activities. Table 1 lists all main classes along with their corresponding activities.

Different methods (functions) are defined within these main classes to perform different tasks whether by themselves or by sending a message(s) to an object of a class that has an association or aggregation relationship with main classes. For instance, `Activity_Drive()` method is defined in the `OPY_Simulate` main class and overridden by methods that have the same name and defined in its subclasses. The method defined in `OPY_Simulate` class is responsible for calling four methods in the `OPY_Simulate` class. These methods are: 1) `Load_Drive()`; 2) `Haul_Drive()`; 3) `Dump_Drive()`; and 4) `Return_Drive()`. They perform different tasks including: 1) adding and removing activities from CAL (current activity list); 2) checking termination condition for simulation replica-

Table 1: *EMSP* Main Classes and Corresponding Activities

Class	Corresponding Activities
`OPY_Simulate`	Load, dump, haul, and return
`OPE_Simulate`	Piling, load, dump, haul, and return
`OSD_Simulate`	Load, dump, haul, return, and spreading
`OCT_Simulate`	Load, dump, haul, return, and compacting
`PS_Simulate`	Piling, load, dump, haul, return, and spreading
`PC_Simulate`	Piling, load, dump, haul, return, and compacting
`SC_Simulate`	Load, dump, haul, return, spreading, and compacting
`PSC_Simulate`	Piling, load, dump, haul, return, spreading, and compacting

tion; and 3) adding and removing haulers and loaders into their queues. Figure 1 illustrates how these functions are called within `Activity_Drive()` method till the termination of simulation replication is reached.

3 GENETIC ALGORITHM

A genetic algorithm (*EM_GA*) has been developed to search for a near-optimum fleet configuration that reduces project total cost. It is dynamically linked to the *EMSP* to perform pilot simulation runs (see Figure 2). The algorithm considers a set of qualitative and quantitative variables that influence the production of earthmoving operations. Qualitative variables represent the models of equipment used in each fleet scenario, whereas, quantitative variables represent the number of these equipment models in each scenario (Marzouk and Moselhi 2001-a).

3.1 Population Structure

According to the case at hand, a number of sub-populations, chromosomes and genes are established

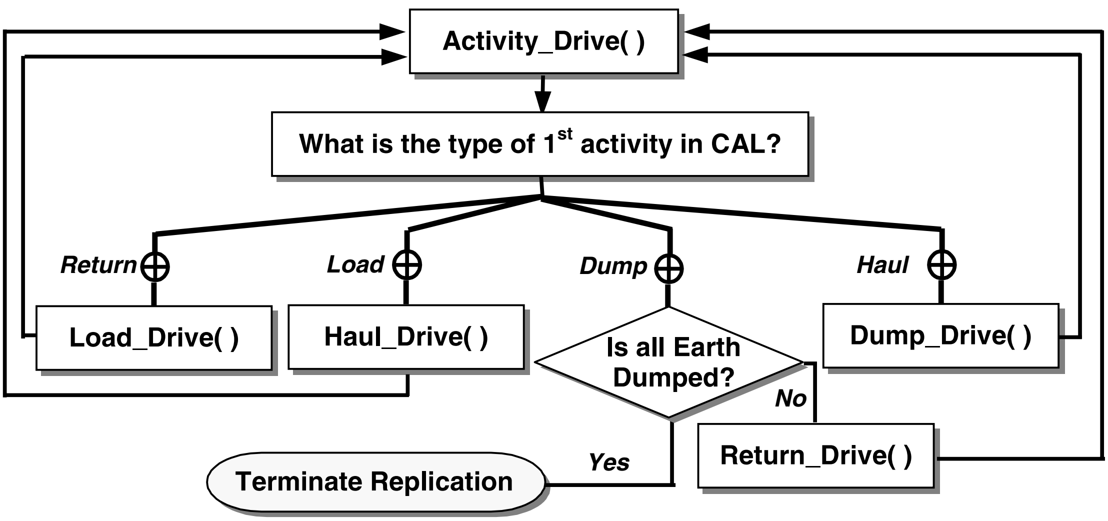

Figure 1: Calling OPY_Simulate Functions within Activity_Drive ( )

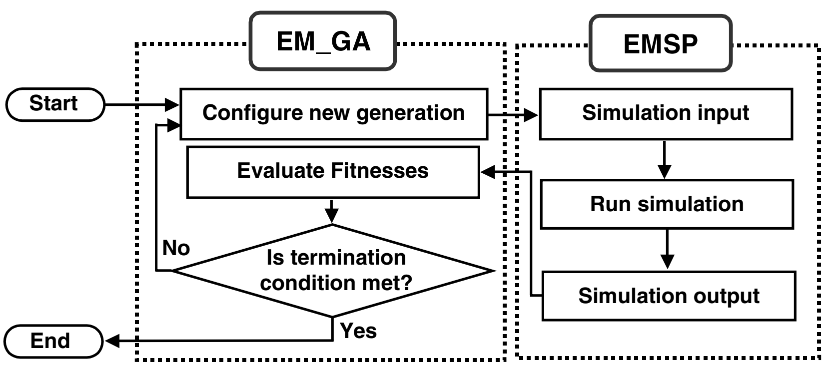

Figure 2: Interaction between *EM_GA* and *EMSP*

within their population. For example, for a case that consists of two fleet scenarios a population composing two sub-populations is generated. Each chromosome defined within each sub-population represents a fleet configuration, whereas, the genes in that chromosome represent the numbers of each equipment type (model) used. As such, chromosome's length (i.e. number of genes) is dynamically adjusted to suit the construction task to be performed by the equipment fleet. Figure 3 illustrates the representation of configured fleets within *EM_GA*.

Main Class	No. of Genes	Chromosomes						
OPY_Simulate	3	I	L	H1	H2			
OPE_Simulate	4	I	P	L	H1	H2		
OSD_Simulate	4	I	L	H1	H2	S		
OCT_Simulate	4	I	L	H1	H2	C		
PS_Simulate	5	I	P	L	H1	H2	S	
PC_Simulate	5	I	P	L	H1	H2	C	
SC_Simulate	5	I	L	H1	H2	S	C	
PSC_Simulate	6	I	P	L	H1	H2	S	C

I : Index gene
P : No. of piling Equipment
H1: No. of haulers (Model 1)
H2: No. of haulers (Model 2)
S : No. spreaders
L : No. of loaders
C : No. compactors

Figure 3: Representation of Fleets

3.2 Input Parameters

The input data to *EM_GA* includes: 1) sub-population size; 2) number of generations; 3) non-improvement limit; 4) crossover and mutation probabilities; 5) activating elitism; 6) including extreme fleet configurations; 7) user-defined fleet configurations; 8) indirect cost; and 9) ranges of equipment being considered. Figure 4 illustrates the dialog box used to define load equipment along with its auxiliary dialog boxes. It should be noted that in the case of user-defined fleet, the dedicated dialog box is activated. In addition, upon activating any of the items marked by the numbers shown in the square boxes of the upper part of Figure 4, *EM_GA* generates automatically the screens marked with the corresponding numbers shown in circles (see the lower part of Figure 4).

Upon defining the input data for the algorithm, the first population is generated by creating a randomly selected set of chromosomes. This population might contain the extreme fleet configurations (if applicable) and user-defined fleet configurations (if applicable). Subsequently, the fitnesses of these chromosomes are estimated with the assistance of *EMSP*. These fitnesses are essentially the inverse of the estimated total cost of the equipment fleets represented by these chromosomes. Detailed description of the cost components of the fitness function can be found elsewhere (Marzouk 2002, Marzouk and Moselhi 2001-a).

3.3 Chromosomes Selection

In the process of generating new populations, chromosomes are selected in pairs and moved to the new generation. This process is carried out for each sub-population separately utilizing *Roulette Wheel* selection procedure (Coley 1999, Holland 1992, Goldberg 1989). In this procedure, the chance of selecting an individual chromosome is proportional to its fitness (maximization problem). A modification was considered in that procedure to account for minimizing the total cost in two stages. First, chromosomes fitnesses are inverted [$1/F(C_i)$]. Second, theses inverted fitnesses are normalized in a way that their summation equals to 1.0. Subsequently, the typical *Roulette Wheel* selection procedure is carried out for the inverted normalized fitnesses. Figure 5 illustrates the process of chromosomes selections.

3.4 Genetic Operators

After selecting chromosomes in pairs, each pair is either subject to crossover or moved directly to the new generation. Crossover process takes place when a generated random number is less than the pre-specified threshold value for crossover (P_c). The crossover process is the fundamental mechanism of GAs that makes them imitate biological genes (Holland 1992). In the proposed algorithm, linear interpolation crossover is utilized to ensure that genes contents receive new values in the new generations (Kim and Adeli 2001). The process of linear interpolation is carried out according to the code shown in Figure 6.

On the other hand, mutation process takes place, for all genes of generated chromosomes except the sub-population index. Mutation is carried out if the generated random number is less than the pre-specified threshold value for mutation (P_m), otherwise the gene is skipped.

The value of the mutated gene is altered within its predefined range. The mutation process is preformed to avoid local minima and to ensure that newly generated populations are not uniform and incapable of further evolution (Holland 1992).

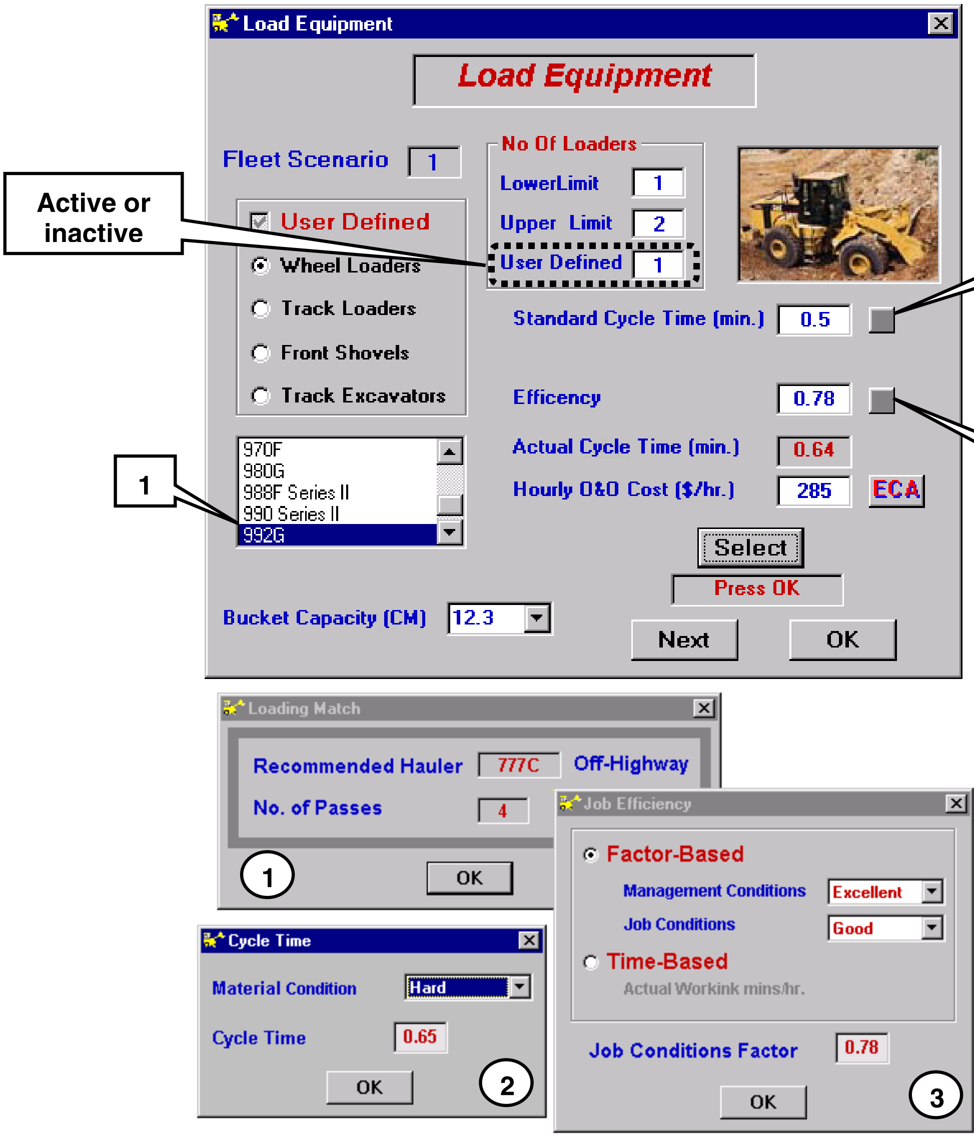

Figure 4: Qualitative and Quantitative Variables

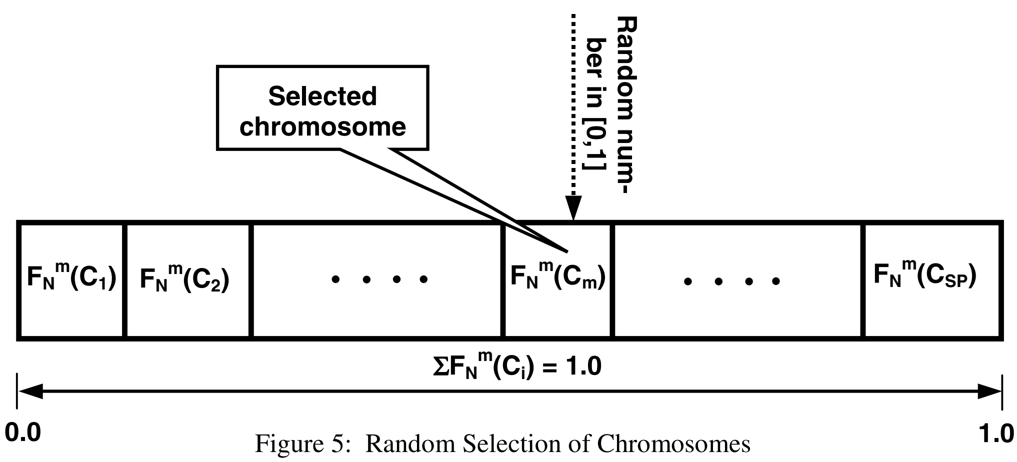

Figure 5: Random Selection of Chromosomes

```
For i=1:N_sub
Action 1   Pick two parents A and B according to
    the modified Roulette Wheel selection.
Action 2   Randomly pick a number R1 between 0 and
    1.
Condition 1    R1 ≤ P_c
Action 2.1   Randomly pick a number R2 between 0
    and 1 to calculate the crossover position
    (C_pos).
Action 2.2   Randomly pick a number R3 between 0
    and 1 to perform linear interpolation.
C_pos = (int)((N_Gen-2)*R2+1);
For j=1: C_pos
    G_i,j = G_A,j
    G_i+1,j = G_B,j
For j= C_pos+1:N_Gen
    G_i,j = Round_Number (G_A,j*R3+ G_B,j*(1-R3))
    G_i+1,j = Round_Number (G_A,j*(1-R3)+G_B,j*R3)
Condition 2    R1 > P_c
For j= 1:N_Gen
    G_i,j = G_A,j
    G_i+1,j = G_B,j
```

Figure 6: Linear Interpolation Code

The developed genetic algorithm has a powerful computational utility that increases its computational efficiency. This is accomplish by employing its elitism function and by storing the fitness of its chromosomes in a built-up database to avoid duplication of fitness calculations, should these chromosomes appear in future generations. In elitism process, the chromosomes with the best fitness in each sub-population are retrieved and used to replace randomly selected chromosomes in new generations. This process overcomes the problem of losing the best chromosome, in each sub-population, due to the random nature of selection and the effect of crossover and mutation. Figure 7 illustrates the implementation of these two features in the developed *EM_GA*.

3.5 Algorithm Output

EM_GA is coded using Microsoft C++ 6.0 and its user interfaces are implemented utilizing Microsoft Visual Basic 6.0 to facilitate data entry. The output of the developed algorithm is exported to Microsoft Excel file and then automatically generated using VBA. *EM_GA* provides its output in text and graphical formats. Figure 8 depicts the dialog box for the main menu dedicated for reporting *EM_GA* results. The text reports simply list the fitnesses of chromosomes in the different generations, whereas, the graphical reports provide charts that are designated to show average and best fitness, crossover and mutation rates, and number of calculated and retrieved fitnesses in each generation.

4 NUMERICAL EXAMPLE

The case study presented by Marzouk and Moselhi (2001-b) is considered here to illustrate the capabilities of the proposed algorithm. This example involves moving approximately 855,000 m^3 of moraine (clay). The example considers two secondary activities (spreading and compacting) in addition to the four main activities. The defineranges for the equipment used in *Fleet 1* and *Fleet 2* are listed in Table 2. The characteristic of the equipment util-

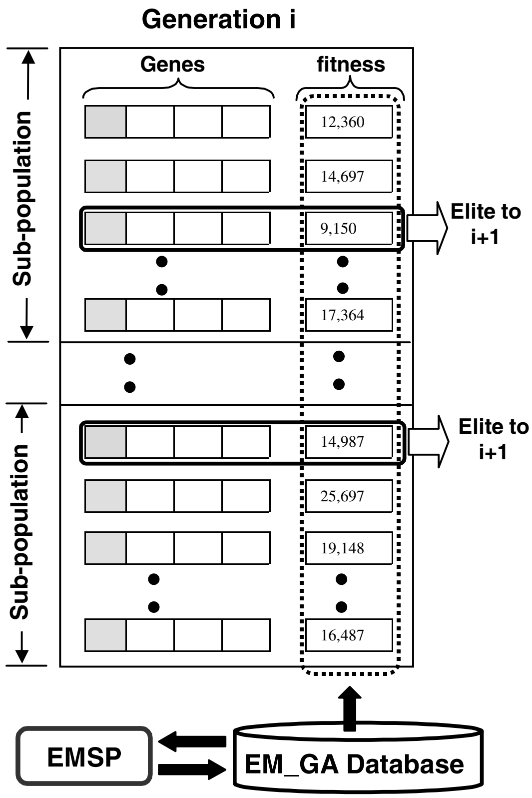

Figure 7: Elitism and Fitness Retrieval

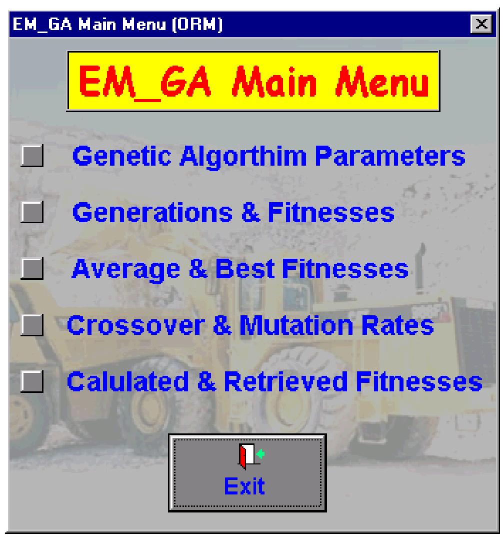

Figure 8: Dialog Box for *EM_GA* Output

ized (in the two fleets) for the secondary activities are listed in Table 3. After entering the input parameters (listed in Table 4), the algorithm was activated and a sensitivity analysis was carried out for different threshold values of crossover and mutation. The output of the sensitivity analysis is shown in Table 5.

The near-optimum fleet is obtained in the 16th run for 0.85 and 0.20 threshold values for crossover and mutations, respectively. That fleet is configured from *Fleet 2* and consists of 3 loaders, 50 haulers, 4 spreaders and 8 compactors. It should be noted that the number of populations generated in this example was 36. The best fitnesses, for *Fleet 1* and *Fleet 2* were obtained after 21 and 17 generations, respectively (see Figure 9). They were initially estimated to be $9,113,640 and $6,147,860 (CND), respectivly. These fitnesses were determined, based on the results obtained from the pilot simulation runs. It is interesting to note that the run with highest threshold values of crossover and mutation (no. 16) yielded the least cost fleet configuration. These results suggest that further analysis should be performed using incrementally higher threshold values for both crossover and mutation. Subsequently, the simulation analysis was performed for the recommended fleet configuration using *EMSP* (see Figure 10). The analysis results indicated the project total duration and total cost to be 332 hrs. and $5,731,400, respectively.

Table 2: Maximum and Minimum Number of Equipment

Fleet No.	Loaders	Haulers	Spreaders	Compactors
1	1-10	10-60	1-8	1-10
2	1-8	10-50	1-8	1-10

Table 3: Characteristics of Spread and Compact Equipment

Spread Equipment (Dozers)	
Type :	CAT D8R
Cycle Production (m3) :	27
Hourly O&O Cost ($/hr) :	150
Duration :	T(2.5, 2.6, 3)
Soil Compactor	
Type :	CAT CS-583C
Cycle Production (m3) :	19.1
Hourly O&O Cost ($/hr) :	90
Duration :	T(1.8, 1.9, 2.3)
T(N1, N2, N3); T : Triangle Distribution, N1: Lower Limit, N2: Mode and N3: Upper Limit	

Table 4: *EM_GA* Input Parameters

Parameter	Value
Sub-population Size :	20
No. of generations :	100
Non-improvement limit :	20
Elitism :	Applicable
Extreme fleet configurations :	Applicable
Scheduled daily hours :	8
No. of working days per month :	22
Time-related indirect cost ($/month) :	250,000
Time-independent indirect cost($) :	1,000,000

Table 5: Result of Sensitivity Analysis

Run No.	Crossover	Mutation	No. of Generations	¥ Recommended Configuration	Cost ($CND)
1	0.55	0.05	21	(2, 5, 42, 5, 10)	6,139,470
2	0.65	0.05	74	(2, 3, 50, 4, 6)	5,798,790
3	0.75	0.05	36	(2, 3, 46, 4, 8)	5,858,030
4	0.85	0.05	89	(2, 3, 50, 4, 8)	5,797,060
5	0.55	0.10	37	(2, 4, 45, 4, 9)	5,918,840
6	0.65	0.10	33	(2, 4, 46, 4, 6)	5,918,080
7	0.75	0.10	44	(2, 3, 49, 4, 8)	5,796,130
8	0.85	0.10	50	(2, 4, 48, 4, 9)	5,851,920
9	0.55	0.15	70	(2, 3, 49, 4, 8)	5,804,520
10	0.65	0.15	68	(2, 3, 50, 4, 10)	5,783,380
11	0.75	0.15	50	(2, 3, 49, 4, 9)	5,785,570
12	0.85	0.15	44	(2, 3, 48, 4, 9)	5,822,950
13	0.55	0.20	53	(2, 3, 47, 4, 8)	5,842,910
14	0.65	0.20	33	(2, 3, 49, 4, 10)	5,803,520
15	0.75	0.20	41	(2, 3, 49, 4, 9)	5,785,570
16	0.85	0.20	36	(2, 3, 50, 4, 8)	5,780,680
¥(N1, N2, N3, N4, N5); N1: fleet scenario, N2: number of haulers, N3: number of loaders, N4: number of spreaders and N5: number of compactors					

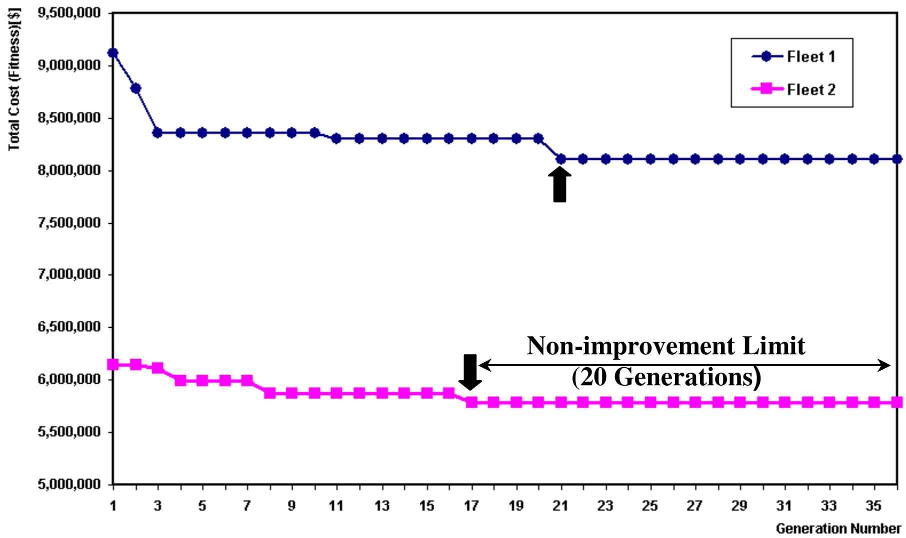

Figure 9: Best Fitnesses (Run No. 16)

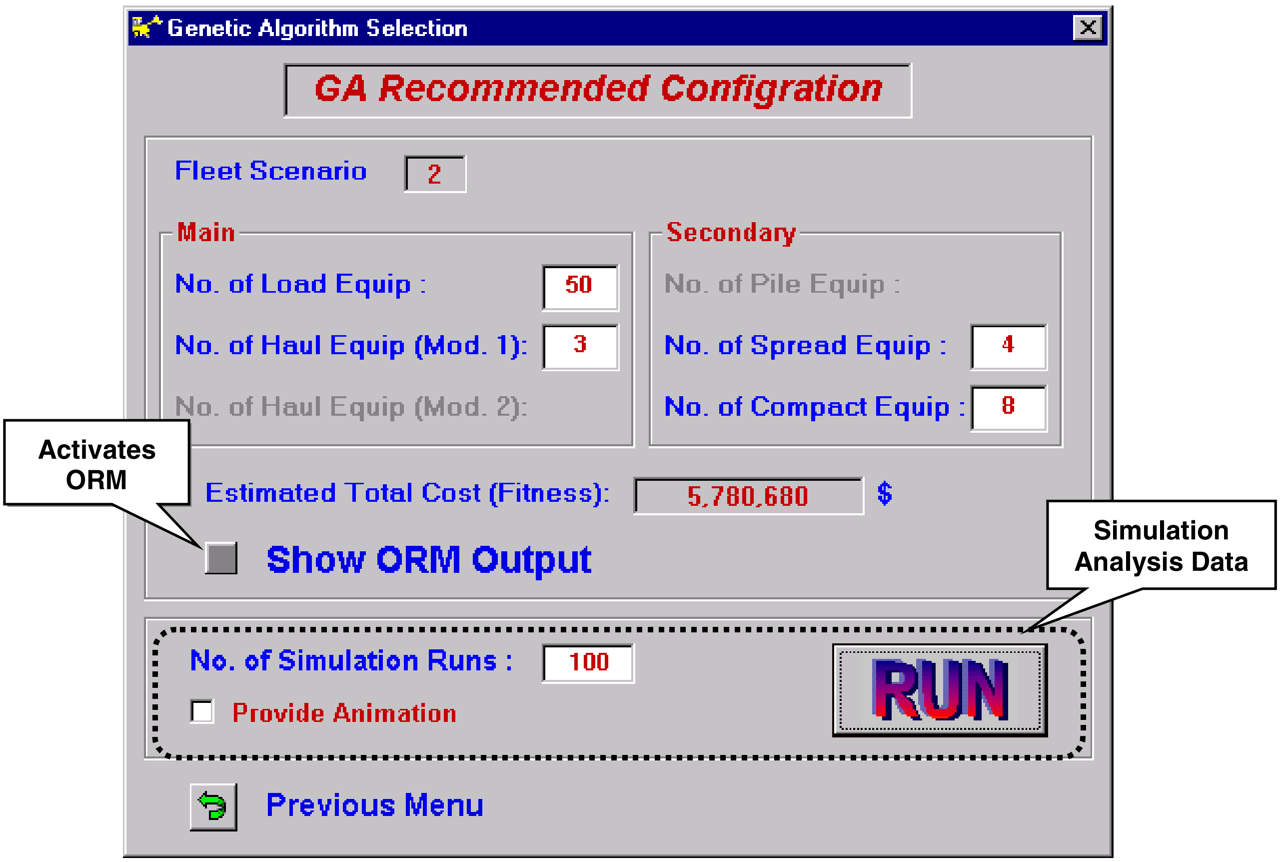

Figure 10: Recommended Fleet Configuration Dialog Box

5 SUMMARY AND CONCLUSIONS

A methodology for optimizing earthmoving operations utilizing computer simulation and genetic algorithms has been presented. The paper provided an overview of a recently developed simulation engine (*EMSP*) that is dedicated for earthmoving operations. *EMSP* is a discrete event simulation engine that has been designed utilizing object-oriented simulation. The paper focused on the optimization aspects and the developments made in a genetic algorithm

(*EM_GA*) that is dynamically linked to *EMSP*. Different features of the developed *EM_GA* were presented including population structure, input parameters, chromosomes selection and genetic operators. The impact of the algorithm's control parameters (crossover and mutation) was studied. The results indicated that the optimization process is sensitive to these parameters. A numerical example was presented to demonstrate the practical use of the developed algorithm.

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support provided by the *National Sciences and Engineering Council of Canada.*

REFERENCES

Coley, D.A. 1999. *An introduction to genetic algorithms for scientists and engineers*. World Scientific, NJ.

Goldberg, D.E. 1989. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley.

Halhal, D., G. A. Walters, D. Ouazar, and D. A. Savic. 1997. Water network rehabilitation with structured messy genetic algorithm. *Journal of Water Resources Planning and Management, ASCE* 123(3): 137-146.

Holland, J. H. 1992. Genetic algorithms. *Scientific American,* July 1992: 66-72.

Kim, H., and H. Adeli. 2001. Discrete cost optimization of composite floors using a floating-point genetic algorithm. *Journal of Engineering Optimization* 33(4): 485-501.

Li, H., and P. Love. 1997. Using improved genetic algorithms to facilitate time-cost optimization. *Journal of Construction Engineering and Management, ASCE* 123(3): 233-237.

Marzouk, M. 2002. Optimizing earthmoving operations using computer simulation. Doctoral dissertation, Department of Building, Civil, and Environmental Engineering, Concordia University, Montreal, Canada.

Marzouk, M., and O. Moselhi. 2000-a. Optimizing earthmoving operations using object-oriented simulation. In *Proceeding of the 2000 Winter Simulation Conference*, ed., J.A. Joines, R.R. Barton, K. Kang and P.A. Fishwick, 1926-1932. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Marzouk, M., and O. Moselhi. 2000-b. An object-oriented simulation model for earthmoving operations. *Journal of Construction Engineering and Management, ASCE* (in press).

Marzouk, M., and O. Moselhi. 2001-a. Simulation optimization for earthmoving operations using genetic algorithms. *Construction Management and Economics Journal* (in press).

Marzouk, M., and O. Moselhi. 2001-b. On the use of fuzzy clustering in construction simulation. In *Proceeding of the 2001 Winter Simulation Conference*, ed., B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1547-1555. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

AUTHOR BIOGRAPHIES

MOHAMED MARZOUK is a Postdoctoral Fellow in Department of Building, Civil, and Environmental Engineering at Concordia University. He received his B.Sc. and M.Sc. in Civil Engineering from Cairo University in 1995 and 1997, respectively. He received his Ph.D. form Concordia University in 2002. His research interest includes simulation and optimization of construction processes, O-O simulation, fuzzy logic and its applications in construction, and decision analysis. His e-mail and web addresses are <marzouk@cbs-engr.concordia.ca> and <http://alcor.concordia.ca/~marzouk/>

OSAMA MOSELHI is Professor and Chair of the Department of Building, Civil and Environmental Engineering at Concordia University. He held several industrial and academic posts in Canada and abroad, in a wide spectrum of the engineering profession, ranging from structural analysis and design to construction engineering and management, on building and heavy civil engineering projects including bridges, offshore and harbor facilities, and nuclear power plants. He is a professional engineer, a Fellow of ASCE and CSCE, and a member and director of a number of professional associations. He authored and co-authored over 150 scientific publications. His research interest includes IT applications in construction, impact of change orders and weather on construction productivity, and risk management. His e-mail and web addresses are <moselhi@cbs-engr.concordia.ca> and <http://www.encs.concordia.ca/bce/professors/Moselhi/moselhihomenew.html>.