# MULTI-DEME PARALLEL COMPUTING MODEL FOR OPTIMIZING LARGE-SCALE CONSTRUCTION PROJECTS

**Amr Kandil[1], and Khaled El-Rayes[2]**

## ABSTRACT

Available construction optimization models are capable generating optimal trade-offs between construction time and cost, however their application is still limited due to their high computational time requirements. In order to overcome this limitation, the present paper presents the development of a robust multi-deme parallel computing model for optimizing large-scale construction projects. The model implements an advanced multi-objective genetic algorithm that is capable of generating optimal trade-offs between construction duration and cost. The model also implements a multi-deme parallel computing framework to enable the optimization of large-scale construction projects. In this framework a number of genetic algorithm populations, called demes, are evolved in isolation on a number of parallel processors. These demes exchange good solutions occasionally through the process of migration, in order to collaborate in finding optimal/near optimal solutions. The model is implemented using a cluster of 50 Intel Xeon processors, and a number of experiments are performed in order to evaluate its efficiency and effectiveness in optimizing a number of large-scale construction projects. The results of these experiments demonstrate that the present model is capable of significantly reducing the computational time requirements for optimizing large-scale projects, and maintaining the quality of the solutions obtained.

## KEY WORDS

Optimization, Distributed computing, Decision making, Information technology, Construction.

## INTRODUCTION

The vast number of new construction technologies that have emerged in the past two decades led to creation of many alternative methods for performing construction activities. These new methods and technologies reduce construction project durations but they often also increase project costs. Examples of such technologies include tilt-up and lift slab construction methods that allow project acceleration but at the same time lead to an increase in construction costs (Allen and Iano 2004). A major challenge that faces construction planners

[1] Assistant Professor, Department of Civil Construction and Environmental Engineering, 494 Town Engineering Building, Iowa State University, Ames, IA 50011, Phone +1 515/294-8824, FAX 515/294-3845, kandil@iastate.edu

[2] Assistant Professor, Department of Civil and Environmental Engineering, University of Illinois at Urbana Champaign, 3219 NCEL 205 N. Mathews Ave., Urbana, IL 61801, Phone +1 217/265-0557, FAX +1 217/265-8039, elrayes@uiuc.edu

and decision makers is the need to measure and optimize the consequences of adopting these new methods and technologies in order to maintain their competitive edge. This challenge is further complicated with the emergence of a number of innovative contracting methods that consider both the project duration and cost in their evaluation of project bids (Jaraiedi et al. 1995, El-Rayes 2001).

In order to address this increase in complexity of construction project planning, a number of construction optimization and decision support models were developed to assist construction planners in selecting optimal construction plans from the vast number of available alternatives. These models addressed a number of important construction project objectives including: (1) construction cost minimization (Karim and Adeli 1999, Hegazy and Wassif 2001); (2) construction duration minimization (Hegazy 1999, Gomar et al. 2002); and (3) the simultaneous minimization of construction cost and duration (Burns et al. 1996, Feng et al. 2000). Although these models have been successful in addressing these challenging construction project objectives, their application was limited to small and medium size construction projects due to their high computational time requirements.

Parallel and distributed computing was successfully utilized to reduce the computational time requirements of decision support systems in a number of civil and environmental engineering disciplines, including: (1) transportation engineering (Agrawal and Mathew 2004, Girianna 2002); (2) structural engineering (De Santiago and Law 2000, Sziveri and Topping 2000); (3) water resources and hydrological engineering (Balla and Lingireddy 2000, Alonso et al. 2000); and (4) construction engineering and project management (El-Rayes and Kandil 2005, Kandil 2005). Although, these studies have demonstrated the ability of parallel and distributed computing frameworks to reduce the computational time requirements of decision support systems when applied to large-scale problems, further computational time reductions may be possible with the application of more efficient parallel computing paradigms (De Santiago and Law 2000). Therefore, the main objective of this paper is to develop an efficient parallel computing framework that follows the multi-deme parallel computing paradigm for optimizing the planning of large-scale construction projects. The model is developed in three main phases that: (1) formulate a multi-objective optimization algorithm capable of simultaneously minimizing construction duration and cost; (2) implement a multi-deme parallel computing framework that reduces the computational time requirements for optimizing large-scale construction projects; and (3) measure the performance of the developed model to determine both its effectiveness and efficiency.

## PHASE 1: MULTI-OBJECTIVE OPTIMIZATION ALGORITHM

The first phase of the implementation of the model aims to develop a multi-objective optimization algorithm capable of simultaneously minimizing project cost and duration. The proposed algorithm is developed as an advanced multi-objective genetic algorithm in the three main stages that: (1) analyze and identify all relevant decision variables; (2) formulate the objective functions; and (3) implement the algorithm.

## DECISION VARIABLES

The developed multi-objective optimization algorithm is designed to identify the optimal construction resource utilization options for each activity in the analyzed construction project. Construction resource utilization options combine a number of planning decisions pertaining to the construction of each activity, including: (1) construction methods, which describe the materials and construction technology used in each activity; (2) construction crews, which includes both the type and amount of labor and equipment utilized in each activity; and (3) crew overtime policies, which determine the length and number of work shifts of the jobsite, and also determines whether or not nighttime construction is utilized. The different possible combinations of these planning decisions are enumerated for all the activities in the project, and then aggregated into a number of discrete resource utilization options for each activity. These construction resource utilization options are then modeled as virtual DNA chromosomes required for implementing the present model as a multi-objective genetic algorithm (Goldberg 1989). These chromosomes contain locations for virtual genes that are used in the present model to represent the resource utilization options for each activity in the analyzed project. Each of these genes is represented as a binary number with varying number of digits depending on the available resource utilization options for the activity modeled by the gene.

## OBJECTIVE FUNCTIONS

The virtual chromosomes that model the aforementioned resource utilization options need to be evaluated to determine the relative merit of these options. In order to evaluate the relative merit of these resource utilization plans, the present model formulates the following two objective functions that measure project time and cost respectively.

$$\text{Minimize Project Time} = \sum_{i=1}^{I} T_i^n \tag{1}$$

Where, $T_i^n$ = duration of activity (i) on the critical path using resource utilization (n).

$$\text{Minimize Project Cost} = \sum_{i=1}^{I} [(M_i^n + D_i^n \times R_i^n) + (B_i^n)] \tag{2}$$

Where, $M_i^n$ = material cost of activity (i) using resource utilization (n); $D_i^n$ = duration of activity (i) using resource utilization (n); $R_i^n$ = daily cost rate in \$/day of resource utilization (n) in activity (i); $B_i^n$ = subcontractor lump sum cost for resource utilization (n) in activity i, if any.

These two objective functions are used to simultaneously minimize construction cost and duration and create an optimal tradeoff between these two objectives. In order to establish this trade-off, the following stage utilizes the above objective functions to implement an advanced multi-objective genetic algorithm.

## MULTI-OBJECTIVE GENETIC ALGORITHM

In order to establish the trade-off between project time and cost, the present multi-objective genetic algorithm is designed to perform three main functions: (1) population initialization;

(2) fitness evaluation; and (3) generation evolution (Deb 2001). The population initialization function generates an initial population of virtual chromosomes that model a group of feasible plans for constructing the analyzed project. This initial population is generated based on a number of parameters that are input in this function, including: (1) population size, which specifies the number of feasible solutions simultaneously evaluated by the genetic algorithm; (2) number of generations, which determines the number of times the genetic algorithm will iterate in order to find the optimal solutions; (3) crossover rate, which sets the probability of two virtual chromosomes crossing at a random point and exchanging their genes; and (4) mutation rate, which establishes the probability of genes in the virtual chromosomes randomly changing their values. The values of these different parameters are determined using a set of parametric equations developed by Reed et al. (2002).

The second function of the developed genetic algorithm establishes the relative merit of the different chromosomes in the genetic algorithm population. In order to perform this function each chromosome in the population is evaluated using the time and cost objective functions described above. These chromosomes are then ranked according to their non-domination. A solution that is identified to be non-dominated (e.g. solution P1 in Figure 1) when it is found to be better than all the other solutions in the genetic algorithm population in at least one objective. The obtained set of non-dominated solutions is called the Pareto optimal set of solutions to the problem. Once this set is identified by the algorithm, it is given rank one and then it removed from the genetic algorithm population. The remaining solutions are then re-examined and a second front is identified and is given rank two. This process is repeated until all solutions in the population are ranked. The fitness evaluation stage also measures a parameter called the solution crowding distance. This parameter is calculated by measuring the distance of each solution in the population from its neighboring solutions.

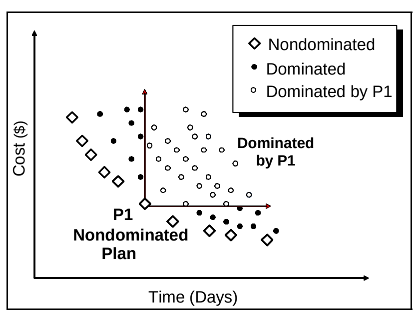


Figure 1: Nondominated Solutions.

Finally the population generation function produces new genetic algorithm populations using the selection, crossover, and mutation processes. The function starts with a parent population of solutions that has been evaluated in the fitness evaluation function. The selection process identifies the chromosomes that would be used to generate the new solutions in a child population based on both the non-dominated rank, and crowding distance. This process gives solutions with higher ranks and larger crowding distances a higher probability of selection to increase the number of non-dominated solutions and avoid the selection of very closely spaced solutions, respectively. The selected solutions then exchange their genes at a randomly selected point in the crossover process. The selected individuals could also undergo a random change in one of their genes in the mutation process. The generated child population is then combined with the parent population. The combined population is then passed back to the fitness evaluation function to enable the sorting of its solutions based on their non-dominated ranking and crowding distances. The top solutions of the sorted combined population form the parent population of the following generation. The iterative execution of fitness evaluation and the generation evolution functions continues until the specified number of generations is completed (Deb 2001).

## PHASE 2: MULTI-DEME PARALLEL COMPUTING FRAMEWORK

The second phase of the development of the present model implements a multi-deme parallel computing framework to reduce the computational time required for optimizing large-scale construction projects. This framework is implemented in two main stages: (1) the framework design stage; and (2) the framework implementation stage.

### FRAMEWORK DESIGN

The multi-deme framework is designed using the coarse-grained paradigm of parallel and distributed computing (Cantú-Paz 2000). In this paradigm the genetic algorithm population is divided into a number of sub-populations called demes. Each of these demes is optimized independently by one of the processors utilized in the computation, in order to reduce the computational time required to optimize the project. These processors implement all the functions of the multi-objective genetic algorithm in a nonhierarchical collaborative fashion. The processors involved in the computation are fully connected to each other, as shown in Figure 2, and collaborate with each other by exchanging their best found solutions in a process called the migration process. The migration process requires the determination of two parameters called the migration rate, and the migration interval. The migration rate determines the number of solutions exchanged in each migration, while the migration interval determines how often the migration process is executed.

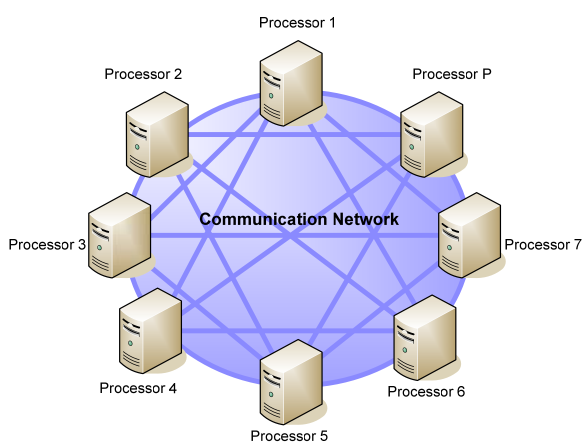


Figure 2: Multi-Deme Parallel Framework Design

### FRAMEWORK IMPLEMENTATION

The present multi-deme parallel computing framework was implemented using 50 processors on the Tungsten cluster which is located at the National Center for Supercomputing Applications. This cluster is composed of 640 Dell PowerEdge 1750 servers, each with two Intel Xeon 3.2 GHz processors, 1.5 MB of cache memory, and a total of 3 GB of SDRAM. The cluster uses the Red Hat Linux operating system and the servers are connected using Myricom's Myrinet low latency cluster interconnect network. The cluster has a peak performance of 6.4 Gflops which makes it the 10th fastest supercomputing cluster in the world at the time of publication of this paper (NCSA 2006). The communications in the framework were implemented using the message passing interface (MPI), which is a standard library that offers a myriad of communication functions that enable the implementation of parallel and distributed programs on a wide range of computing systems including supercomputing clusters and networks of personal computers (Snir et al. 1998; Gropp et al. 1999). This present implementation using MPI is envisioned to facilitate the planned migration of the present framework in a future phase of this study to a network of personal computers that is typically available in construction engineering and management offices.

The implementation of the present model was tested and evaluated using two parallel and distributed computing performance metrics to determine the effect of the size of the optimized project, the number of processors utilized, and the migration process on both the efficiency and effectiveness of the model. The main performance metrics along with the main results of the evaluation are discussed in the following section.

## PHASE 3: PERFORMANCE EVALUATION

The performance of the present model was evaluated using: (1) three different project sizes which are composed of 180, 360, and 720 activities; (2) different numbers of processors ranging from 1 to 50 processors; (3) three different migration rates that transfer 25%, 50%, and 75% of the populations of each deme; and (4) three different migration intervals that allow the migration process to occur every one, two, or four generations. The performance of the model was evaluated using two performance metrics: (1) Elapsed time, which evaluates the efficiency of the model by measuring the computational time it requires; (2) Quality of solutions, which evaluates the effectiveness of the model by measuring the number of optimal solutions it obtains.

### ELAPSED TIME

The elapsed time of the present model was measured for the different combinations of the aforementioned parameters using the high precision MPI_Time function, as shown in Figure 3. The results of this evaluation demonstrated that as the number of processors utilized in the computation increased the total elapsed time for optimizing the different project sizes decreased. The percentage of this decrease was also noted to be larger for larger projects reaching a maximum of a 98% decrease in the 720 activity project optimized using 50 processors, 25% migration rate, and a four generation migration interval. This decrease in elapsed time was however noted to be less for experiments with higher migration rates and intervals. For example, the percent decrease in elapsed time for the 720 activity project using a migration rate of 25%, and a migration interval of one generation was found to be 97%. The results also demonstrated that the impact of larger migration rates and intervals was higher for projects with a smaller size. For example, the elapsed time using 50 processors and 25% migration rate increased by 46% for the 180 activity project, as compared to an increase of only 28% for the 360 activity project, when the migration rate increased from once every 4 generations to once every generation (see Figure 3). These results, therefore, clearly demonstrate the impact of deme size (and hence number of processors), migration rates, and migration intervals on elapsed time of the multi-deme parallel computing model.

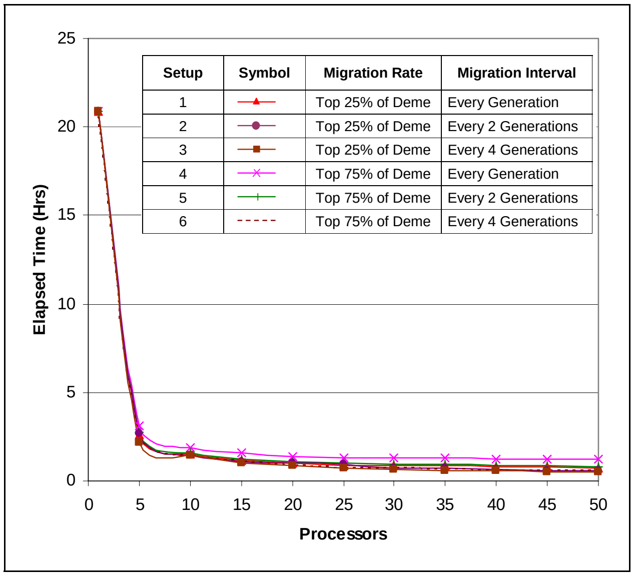


Figure 3: Sample Elapsed Time Results for 360 Activity Project

### QUALITY OF SOULUTIONS

The quality of solutions performance indicator was also measured for the different combinations of numbers of project sizes, numbers of processors, migration rates, and migration intervals (see Figure 4). The results demonstrated that the number of non-dominated optimal solutions decreased as the number of processors involved in the computation increased. For example, the number of non-dominated solutions for the 360 activity project decreased from 232 solutions using a single processor to about 98 solutions using 50 processors. This decrease can be attributed to the decrease in deme size as a result of this increase in the number of processors, which decreases the explorative and exploitive abilities of the genetic algorithm that allow it to use existing solutions to find better ones. The results also demonstrated that as the migration rates and intervals increased the number of non-dominated solutions obtained increased. For example, the number of non-dominated solutions for the 360 activity project increased from 98 solutions for the 75% migration rate and the every four generations migration interval to 112 solutions using the same migration rate and the every generation migration interval. This increase in the non-dominated solutions can be attributed to the increase in the collaboration between the different demes in the model by increasing the number and frequency of exchanged top solutions.

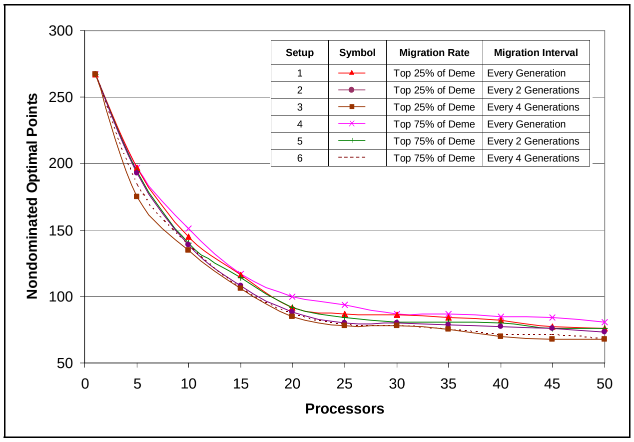


Figure 4: Nondominated Solutions of 360 Activity Project

The results of both performance measures illustrated that the developed multi-deme parallel computing model is capable of significantly reducing the time required to optimize large-scale construction projects. The results also demonstrated that there is a tradeoff between the effectiveness and efficiency of the developed multi-deme parallel computing model. This tradeoff is illustrated in Figure 5 which shows the relation between elapsed time and the number of non-dominated optimal solutions obtained in one of the examined example projects. The tradeoff demonstrated that although a higher number of non-dominated solutions requires a larger elapsed time, a reasonable number of non-dominated solutions can be obtained at a significantly smaller elapsed time using a small number of processors. For example in Figure 5, it can be seen that 303 non-dominated solutions are obtained using a single processor in approximately 137 hours of processing. For the same project it only takes 12 hours to obtain 291 non-dominated solutions using 5 processors.

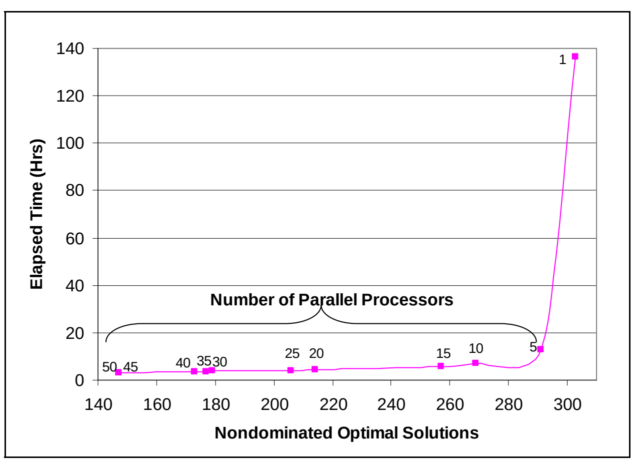


Figure 5: Tradeoff between model efficiency and effectiveness.

## CONCLUSIONS

This paper presented the development of a multi-deme parallel computing model for optimizing the planning of large-scale construction projects. The model was developed in three main phases that: (1) developed an advanced multi-objective genetic algorithm that simultaneously minimized both construction duration and cost; (2) implemented a multi-deme parallel computing framework; and (3) evaluated the performance of the model using two performance measures. The model was implemented using 50 processors on the Tungsten cluster located at the National Center for Super Computing Application. The performance of the model was tested using three projects containing 180, 360, and 720 activities, respectively. The results of this evaluation demonstrated that model was capable of reducing the computational time requirements for all three projects, and that the time reduction was greater for the largest project that included 720 activities. This indicates that the model is more effective in reducing the computational requirements of larger projects, and shows the potential of more computational time reductions in larger sized projects. The results also demonstrated that the model was effective in finding non-dominated optimal plans for the optimized projects and that a tradeoff exists between the efficiency and effectiveness of the model. The analysis of this tradeoff demonstrated that the marginal loss in the effectiveness of the model was small compared to the efficiency gains achieved. This analysis also demonstrated to potential users of the model the computational time required to attain different levels of effectiveness, which enables them to select the level of efficiency they can afford or the level of effectiveness they desire.

## ACKNOWLEDGMENTS

The financial support provided by the National Science Foundation for this research project under the NSF CAREER Award CMS-0238470 is gratefully acknowledged. The authors also acknowledge the technical support provided by the department of Computational Science and Engineering at the University of Illinois at Urbana-Champaign, and the National Center for Supercomputing Applications.

## REFERENCES

Agrawal, J., and Mathew, T, (2004). “Transit Network Design Using Parallel Genetic Algorithm” *J. Comp. Civ. Engrg.*, ASCE, 18(3), 248-256.

Allen, E., and Iano, J. (2004). *Fundamentals of Building Construction*, 4th Ed., John Willey & Sons, LTD, New York, N.Y.

Alonso, J., Alvarriuz, F., Guerrero, D., Hernández, V., Ruiz, P., Vidal, A., Martinez, F., Vercher, J., and Ulanicki, B. (2000). “Parallel Computing in Water Network Analysis and Leakage Minimization” *J. Water Resour. Plng. and Mgm.t*, ASCE, 126(4), 251-260.

Balla, M., and Lingireddy, S. (2000) “Distributed Genetic Algorithm Model on Network of Personal Computers” *J. Comp. Civ. Engrg.*, ASCE, 14(3), 199-205.

Burns, S., Liu, L., and Feng, C. (1996) "The LP/IP hybrid Method for Construction Time-Cost Trade-Off Analaysis," *J. Constr. Mgmt. Econ.*, 14, 265-276.

Cantú-Paz, E. (2000). *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publishers, Boston, MA.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, LTD, New York, NY.

De Santiago, E., and Law, K. H. (2000) "A Distributed Implementation of an Adaptive Finite Element Method for Fluid Problems," *Computers and Structures*, 74(1), 97-119.

El-Rayes K., and Kandil, A. (2005) "Time-Cost-Quality Trade off Analysis for Highway Construction," *J. Constr. Engrg. Mgmt*, ASCE, 131(4) 477-486.

El-Rayes, K. (2001) "Optimum Planning of Highway Construction Under the A + B Bidding Method," *J. Constr. Engrg. Mgmt.*, ASCE, 127(4), 261-269.

Feng, C., Liu, L. and Burns, S. (2000) "Stochastic Construction Time-Cost Trade-Off Analysis," *J. Comp. Civ. Engrg.*, ASCE, 14(2), 117-126.

Girianna, M. (2002). *Dynamic Signal Coordination Model for a Network with Oversaturated Intersections*, Ph.D. Diss, Civil & Envir. Engrg., Univ. of Illinois at Urbana Champaign, Urbana, IL.

Goldberg, D E (1989). *Genetic Algorithms in search, optimization and machine learning*, Addison Wesley Publishing Company, New York N.Y.

Gomar, J., Haas, C. and Morton, D. (2002) "Assignment and Allocation Optimization of Partially Multiskilled Workforce," *J. Constr. Engrg. Mgmt.*, ASCE, 128(2), 103-109.

Gropp, W., Lusk, E., and Thakur, R. (1999). U*sing MPI-2: Advanced Features of the Message Passing Interface*, Massachusetts Institute of Technology, Cambridge, MA.

Hegazy, T. (1999). "Optimization of Resource Allocation and Leveling Using Genetic Algorithms." *J. Constr. Engrg. Mgmt.*, ASCE, 125(3), 167-175.

Hegazy, T. and Wassef, N. (2001) "Cost Optimization in Projects with Repetitive Nonserial Activities," *J. Constr. Engrg. Mgmt.*, ASCE, 127(3), 183-191.

Jaraiedi, M., Plummer, R., and Aber, M. (1995) "Incentive/Disincentive Guidelines for Highway Construction Contracts" *J. Constr. Engrg. Mgmt.*, ASCE, 121(1), 112-120.

Kandil, A., and El-Rayes, K. (2005) "Parallel Computing Framework for Optimizing Construction Planning in Large Scale Projects" *J. Comp. Civ. Engrg.*, ASCE,19(3), 304-312.

Kandil, A. (2005). *Multi-Objective Optimization for Large-Scale Highway Construction Projects*, Ph.D. Diss, Civil & Envir. Engrg., Univ. of Illinois at Urbana Champaign, Urbana, IL.

Karim, A. and Adeli, H. (1999) "OO Information Model for Construction Project Management," *J. Constr. Engrg. Mgmt.*, ASCE, 125(5), 361-367.

Reed, P., Minsker B. S., and Goldberg, D. E. (2003). "Simplifying Multiobjective Optimization: An Automated Design Methodology for the Non-dominated Sorted Genetic Algorithm-II." *Water Resources Research*, American Geophysical Union, 39(7).

Snir, M., Otto, S., Huss-Lederman, S., Walker, D., and Dongarra, J. (1998). *MPI: The Complete Reference, Volume1, The MPI Core*, Massachusetts Institute of Technology, Cambridge, MA.

Sziveri, J., and Topping, B. (2000) "Transient Dynamic Nonlinear Analysis Using MIMD Computer Architectures." *J. Comp. Civ. Engrg.*, ASCE, 14(2), 79-91.