Fund Allocation in Complex Rehabilitation Programs

Tarek HEGAZY and Roozbeh RASHEDI
Department of Civil and Environmental Engineering, University of Waterloo
Waterloo, N2L3G1, Canada

ABSTRACT

Civil Infrastructure assets require continuous renewal actions to sustain their operability and safety. Allocating limited renewal funds amongst numerous building components, however, represents a large-scale optimization problem and earlier efforts utilized genetic algorithms (GAs) to optimize medium size problems yet exhibit steep performance degradation as problem size increases. In this research, after experimenting with various approaches of segmenting a large problem into multiple smaller sub-problems, clustered segmentation proved to be the most promising. The paper discusses the underlying life cycle analysis model, the various segmentation methods, and the optimization results using the improved GAs + clustered segmentation, which proved to be able to optimize asset renewals for 50,000 components with no noticeable performance degradation. The proposed method is simple and logical, and can be used on variety of asset types to improve infrastructure fund allocation. Future extension of this research is then highlighted.

Keywords: Capital Renewal, Asset Management, Computer Applications, Genetic Algorithms, Life Cycle Cost Analysis, Network-Level Decisions.

1. INTRODUCTION

Capital renewal is an essential asset management decision that is important to sustaining the serviceability of civil infrastructure assets (roads, bridges, water/sewer networks, educational buildings, etc.). In general, capital renewal involves two levels of decisions [1]: project-level decisions of the appropriate rehabilitation strategies to use for individual asset components (roof, windows, foundations, etc.); and network-level decisions of which asset components to repair in each year of the plan. Each level of decision, by itself, is a complex combinatorial problem that involves trying different combinations of actions to determine the best decision. For example, at the project level, consider the case of one bridge with 5 main components (deck, substructure, superstructure, joints, and finishing) with four repair alternatives for each component. One decision can be (1, 2, 0, 3, 1) which represents the indices to the repair types for the five components, respectively. As such, the number of possible decisions (solutions) is $4^5 = 1{,}024$. Expanding to the network level, on the other hand, consider a network of only 10 bridges that need to be repaired within a five-year planning period. The network-level decision involves deciding for each bridge *i* its year of repair (0, 1, 2, 3, 4, or 5). As such, for the ten-bridge network, the possible number of bridge selection combinations is 6^{10} = 60,466,176, which is extremely large. Network level decisions, therefore, are much larger in size than project level decisions. Due to large problem size, finding optimum decisions becomes a very difficult task that existing systems have not been able to adequately address.

In the literature, many researchers have proposed models for optimizing rehabilitation (renewal) decisions (e.g., [2]; [3]; [4]). These efforts provided interesting approaches to model life cycle cost analysis, however, none has proved to be able to handle very large-scale problems. Many commercial asset management systems also exist (Synergen, CityWorks, RIVA, etc.) but generally lack optimization capabilities and mostly use a simple ranking approach to prioritize assets for rehabilitation purposes [5]. Among the recent efforts that integrated both levels within an optimization framework is the Multiple Optimization and Segmentation Technique (MOST) developed by Hegazy and Elhakeem (2011). MOST (discussed briefly in the next section) handles a large-scale problem by first optimizing project-level sub-problems and using their results to formulate a network-level optimization (Fig. 1). Using this approach, MOST utilized the Genetic Algorithms (GAs) technique to handle network-level problems for up to 8000 components, simultaneously (one building has about 150 components). This paper builds upon the MOST technique and improves its performance for real-life problems to suit the organizations that own a large number of building assets (e.g., school boards, industrial facilities, etc.).

2. MULTIPE OPTIMIZATION AND SEGMENTAION TECHNIQUE (MOST)

MOST was introduced as an integrated life cycle cost analysis model to support asset renewal for school buildings administrated by Toronto District School Board (TDSB) in Canada. MOST uses a divide and conquer approach to segment the large optimization into a series of smaller optimizations, starting at the project level. Each individual optimization considers one building component and one possible repair year; and determines the best repair method and cost, for that component in that year. Within each small optimization, the formulation considers the component condition, deterioration behavior, and expected after-repair condition to determine the repair with the highest benefit-to-cost ratio. The results of all the individual optimizations (suboptimal at the project level) then are passed to a single network-level optimization.

The resultant of all project-level optimizations is a pool of best repair scenarios and their corresponding costs and benefits. These are used as the input to a network-level optimization to decide on repair timing. The objective of network-level optimization is to minimize the overall network deterioration index (DI_N) while not exceeding the available repair budget. Rather than a one-shot optimization over the 5-year planning horizon, MOST uses a year-by-year optimization formulation, from the first year consecutively until the end of planning horizon (as indicated in Fig. 1). Using this formulation reduces the solution-space size and leads to better solution quality. In general, the overall parameters in the network-level optimization (variables, objective function, and constraints) are as follows:

$$\text{Decision Variables: } \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} \\ . & . & . & . & . \\ . & . & Y_{jk} & . & . \\ . & . & . & . & . \\ Y_{j1} & Y_{j2} & Y_{j3} & Y_{j4} & Y_{j5} \end{bmatrix} \quad (1)$$

where, Y_{jk} = 0 (no repair), Y_{jk} = 1 means component J is decided to be repaired in year k.

Objective function: minimize the network deterioration index (DI_N)

$$DI_N = \frac{\sum_j (Average\ DI_{jk} \times RIF_j) + \sum_j (IE_{jk} \times Y_{jk} \times RIF_j)}{\sum_j RIF_j} \quad (2)$$

$$\forall\, j \in Network \quad , \quad \forall\, k \in Planning\ Horizon$$

where, RIF_j is the relative importance factor of component j; DI_{jk} is the deterioration index of component j in year k; and IE_{jk} is the improvement effect of repairing component j in year k, which is equal to:

$$IE_{jk} = EP_{jk} - EP_{j0} \quad (3)$$

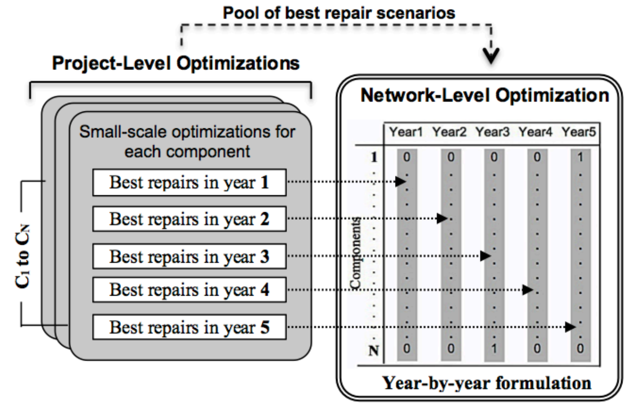

FIGURE 1 MOST

where EP_{jk} is the expected performance of instance j when repaired in year k and EP_{j0} is the initial performance of the component without repairs [6].

Constraint: Total repair cost for selected components in year $k \leq$ budget limit in year k.

Using these optimization parameters, and the results of the project-level optimizations, a life cycle cost analysis (LCCA) model was developed and implemented in an Excel spreadsheet, as shown in Fig. 2. Fig. 2 shows a partial list of asset components, where each row is a component. The highlighted component (fire alarm system), for example, has a relative importance factor (defined internally by experts at the TDSB) of 90. The current performance (deterioration) before repairs is also 72.49. The following columns then represent the cost and performance associated with repairs in year 1, 2, to 5 (results of the project-level optimizations). For example, if the component is repaired in year 2 (as highlighted), its deterioration will be reduced from 72.49 to 24.31, at a cost of $42,350. The LCCA model uses the data from project-level analysis to formulate the network-level model. As an example, the decision to repair the fire alarm system in year 2 is selected by assigning a value of 1 to the decision variable of year 2. Accordingly, the LCCA model reads values for RIF (90); expected performance after repair in year 2 (24.31); and repair cost ($42.350). The combination of component decisions determines the overall network deterioration index, using Eq. (1), Eq. (2) and Eq. (3). To handle the network-level optimization, MOST uses genetic algorithm (GA), which has been widely used by many researchers in various domains to solve combinatorial problems [7; 8; 9]. Experimenting on several network-level optimizations with different numbers of components, it was noticed that the performance of GAs steeply degrades as the problem size increases. At 8,000 components, it was noticed that GAs becomes incapable of reaching solutions that are better than simple ranking methods. To consider larger models GA + Segmentation approaches is proposed to improve the performance of MOST technique.

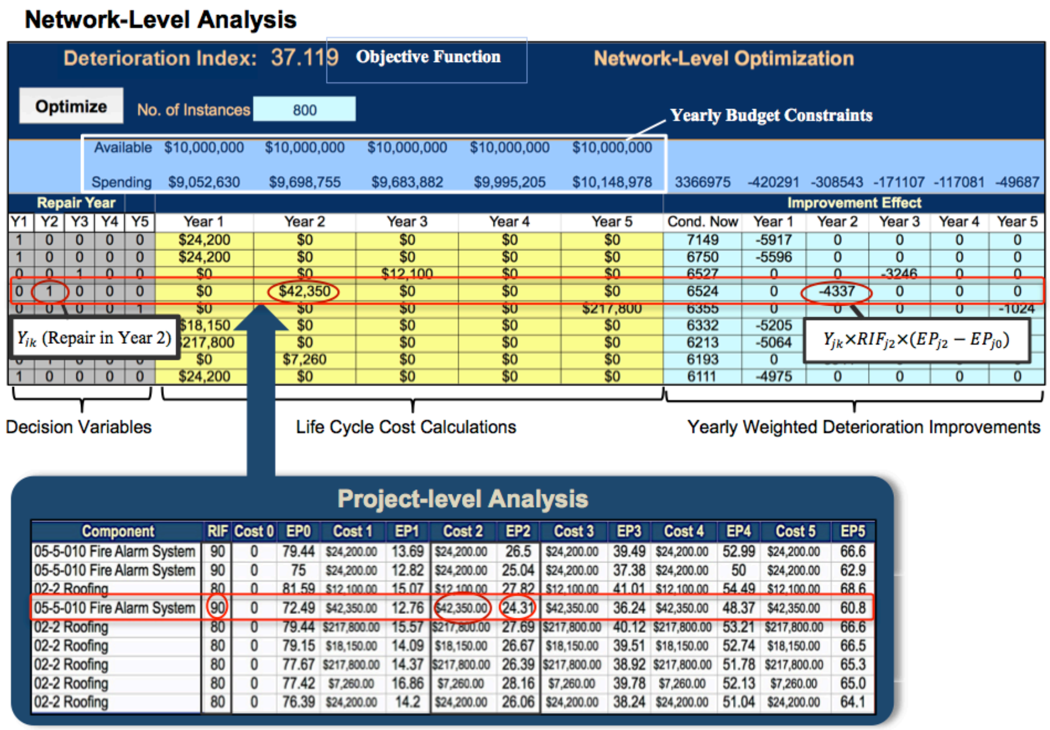

FIGURE 2 Network-level life cycle cost analysis model

3. PROPOSED GA + SEGMENTATION

This study proposes a segmentation approach at the network level to improve the solution quality and to handle very large-scale problems (as indicated in Fig. 1). While the original MOST technique utilizes segmentation at the project level by solving multiple small-scale optimizations, the proposed "GA + Segmentation" approach is applied at the large-scale network-level optimization. The main idea is to decompose a large problem into sub-problems that are optimized separately and their results are then merged to assemble the final solution. The main idea is to decompose the large network-level optimization model into several sub-problems with reduced solution-space. Next, the generated sub-problems that are easier to handle by GA, are optimized separately and their results are merged to assemble the final solution. Various segmentation methods have been investigated and tested on different size problems (Fig. 3). Initial experiments involved two methods: Random Segmentation; and Data Compression. The clustered segmentation approach, which is developed based on previous approaches, uses different mechanisms for generating segments, allocating budget limit to each segment, and redistributing leftover moneys in the best manner. The description of the various methods and their results are discussed in the following.

Random Segmentation

Random segmentation is a simple procedure for generating segments, which can be used as a starting point to evaluate the effectiveness of using segmentation with GAs. In this approach, the large list of components is divided into segments with equal number of randomly selected components (Fig. 3). To distribute the total yearly budget among the segments, the budget for each segment is calculated by dividing the total yearly budget by the number of segments. For example, if four segments are created, each will be allocated 25% of the total yearly budget. After optimizing all segments individually, results are combined to give the solution to the original model. The results of several experiments show improved solution quality when the number of components is less than 6,000; however, it still suffers from performance degradation. Accordingly, similarity-based segmentation methods (i.e., data compression and clustering) are introduced and investigated in the next sections.

Segmentation Using Data Compression

Data compression segments the large list of components based on similarities between components. Components of the same system (e.g., mechanical, electrical, architectural, etc.) and very close initial conditions (DI_0) are grouped in a separate segment. Next, to compress the

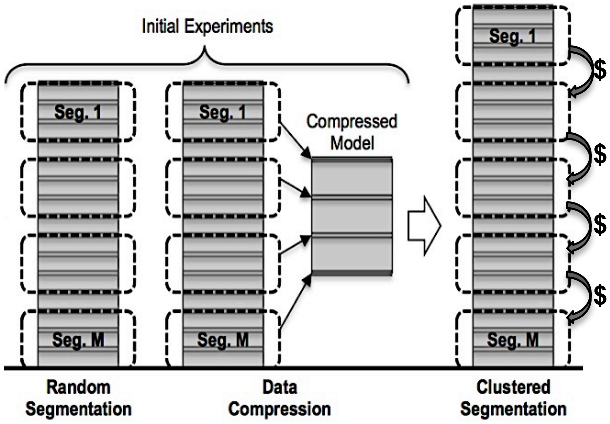

FIGURE 3 Segmentation approaches

large-scale data, each consistent segment is replaced by a single representative component (Fig. 3), which has the combined characteristic of all of the internal components. The compressed model, which contains only the representative components, is then optimized instead of the original large-scale model and solutions obtained for each representative component is reassigned to all the components in the segment that it represents. A model with 800 components from TDSB asset inventory is used for primary experimentation. The initial model consists of 541 architectural, 210 mechanical, and 49 electrical components. After grouping components based on their building system, then the components with similar RIF and very close initial condition (i.e., 1% variation) are fitted into segments.

Following this procedure, 29, 22, and 21 segments are generated, containing architectural, mechanical, and electrical components, respectively. One representative component of each segment is then generated from its individual components, with the following characteristics:

- Expected performance in each year = average of EPs for the individual components;
- Relative importance factor = average of RIFs of all individual components; and
- Repair cost = sum of repair costs of those for the individual components.

With the representative components, the initial model with 800 components is compressed to a model with only 72 representative ones. The compressed model has the same objective function as the base model and also the same budget limits. After optimizing the compressed model and reassigning the representative solution to corresponding components, a network deterioration index of 41.04 was obtained. Although, data compression results are better than simple ranking ($DI_N = 44.89$), comparing to solutions obtained by using GA with random segmentation, data compression is found to have lower quality solution. The poor optimality can be attributed to the approximation used to determine the representative components.

Segmentation Using Clustering

Clustering is a similarity-based method that considers all the components, without compression. It involves four steps that contribute to improving the efficiency of very large-scale network-level optimization: (1) Similarity analysis of input data: used to generate segments with similar internal data and to assign specific characteristics to segments to allocated budget accordingly; (2) Determining optimum segment size: used to achieve better quality solutions; (3) Segment ordering: used to prioritize budget allocation; and (4) Budget allocation and redistribution: used to allocate budget in the most efficient way. These steps are discussed as follows:

1) Similarity Analysis of Input Data: Clustering is mainly based on pattern analysis and similarities between datasets with respect to different parameters. Considering similarity during the segmentation can result in creating segments from components having close characteristics (e.g., very close RIF, initial condition, deterioration behavior, etc.). One of the widely used similarity measures is the Euclidian Distance [10]. Considering two sets of data, $x = \{x_1, x_2, \dots, x_n\}$ and $y = \{y_1, y_2, \dots, y_n\}$, as two points in an n-dimensional Euclidian space, the similarity (Euclidian Distance) between the two datasets is defined as :

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{4}$$

With N datasets, an (N × N) similarity matrix can be created to indicate the level of similarity among all data elements. As such, for the network-level LCCA, similarity between components is determined by calculating the Euclidian Distances, considering the deterioration behaviors (variations in DIs), and relative importance factor (RIF) as follow:

$$d\left(component_i, component_j\right) = \sqrt{\left(Average(DI_j)_k - Average(DI_i)_k\right)^2 + \left(RIF_j - RIF_i\right)^2} \tag{5}$$

$$\forall\, j \in Network \quad , \quad \forall\, k \in Planning\ Horizon$$

where, $Average(DI_j)_k$ is the average of deterioration indices during the planning horizon for component j, and RIF_j is the relative importance factor for that component. Using Eq. 5, a similarity matrix was created and the values were color coded, then the components were sorted based on the similarity values.

2) Determining Optimum Segment Size: Segment size refers to the number of components within each segment. Determining segment size is based on the capability of optimization tools and model formulations to achieve high quality solutions. Based on the results of many experiments, segment sizes from 50 to 100 have the highest solution quality and are suggested to use for the TDSB asset renewal problem.

3) Segment Ordering: To help in identifying the budget limit to apply to each segment, a measure of a segment's relative criticality is calculated based on the relative importance factors of its components. First, the Criticality Index SCr_j of segment j is defined as the average of the criticalities of its components and the segment size SS_j (number of components in the segment), as follows:

$$SCr_j = \frac{\sum_{i=1}^{SS} CrI_i}{SS_j} \tag{6}$$

where, CrI_i is the Criticality index (from 0 to 1) of component *i*, which is calculated as follows:

$$CrI_i = \frac{RIF_i}{100} \times \frac{Average(EP_i)_k}{100} \tag{7}$$

$$\forall\ k\ \in Planning\ Horizon\ [1, 2, 3, 4, 5]$$

Afterwards, the Relative criticality (RC_j) of a segment is then calculated as follow:

$$RC_j = \frac{SCr_j}{\sum_{i=1}^{NS} SCr_i} \tag{8}$$

NS = Number of Segments

After defining the relative criticality of all segments, segments are ordered from low relative criticality to high relative criticality to facilitate budget allocation in the following step.

4) Budget Allocation and Redistribution: After ordering segments based on criticality values, available yearly budget is allocated to segments based on their relative criticality, as follows:

$$B_{kj} = TB_k \times RC_j \tag{9}$$

where, B_{kj} is the allocated budget to segment *j* in year *k*, and TB_k is the total available budget in year *k*, and RC_j is the relative criticality of segment *j*.

Based on various fund allocation experiments, it was found that increasing the available budget for high criticality segments improves the final solution. Since each segment will be optimized separately and the budget constraint cannot be met to the exact dollar, a small fraction of the budget will be remained unallocated. This leftover money will accumulate from many segments to become a considerable amount in a large-scale problem with many segments. One way to redistribute the leftover money from segment *j* is to add it to available budget for segment *j*+1. Using this approach during the optimization process results in more allocation of budget to segments with higher relative criticality (sorted at the bottom), as shown in Fig. 4. Combining this budget redistribution with budget allocation based on relative criticality leads to the following revised budget allocation function:

$$B_{kj} = TB_k \times RC_j + \left(B_{K(j-1)} - SRC_{k(j-1)}\right) \tag{10}$$

where, $B_{k(j-1)}$ is the allocated budget to segment (*j*-1) in year *k* and $SRC_{k(j-1)}$ is total cost of repairs in segment (*j*-1) in year *k*.

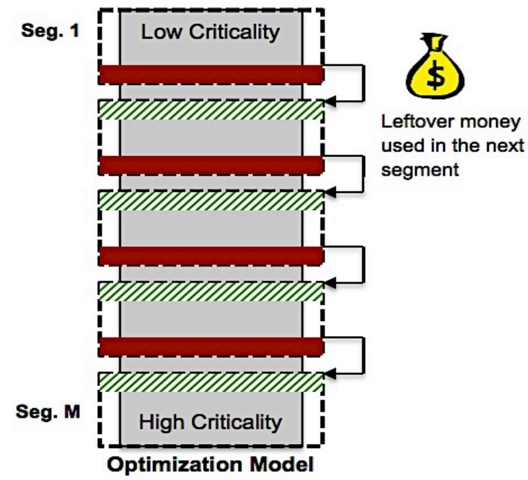

FIGURE 4 Sequential budget redistribution

4. RESULTS AND DISCUTION

For comparison purposes, the best result of the previous work of Hegazy and Elhakeem (2011), which achieved a DI_N of 44.8 for the 8000 components of the TDSB, was used as a baseline. The proposed "GA + Segmentation" mechanism was then applied to networks of different sizes, created by copying the 800 components multiple times. The benefit of this approach is that the results should be multiples of the best results obtained from the 800-component case. Fig. 5 shows an overall comparison of results obtained from the optimization with clustered segments, in comparison to the simple ranking approach typically used by asset managers, and the previous research of the MOST technique. As shown in Fig. 5, at 8000 components, the optimization achieved a network deterioration index of 32.8, which is a huge improvement in optimization performance compared to the baseline case. As shown in the figure, experimenting with even larger number of components the proposed approach shows almost no degradation in the optimization performance. All experiments were conducted using variations of the base LCCA model in Fig. 2 and a laptop computer with 2.4 GHz speed and 4 GB of memory.

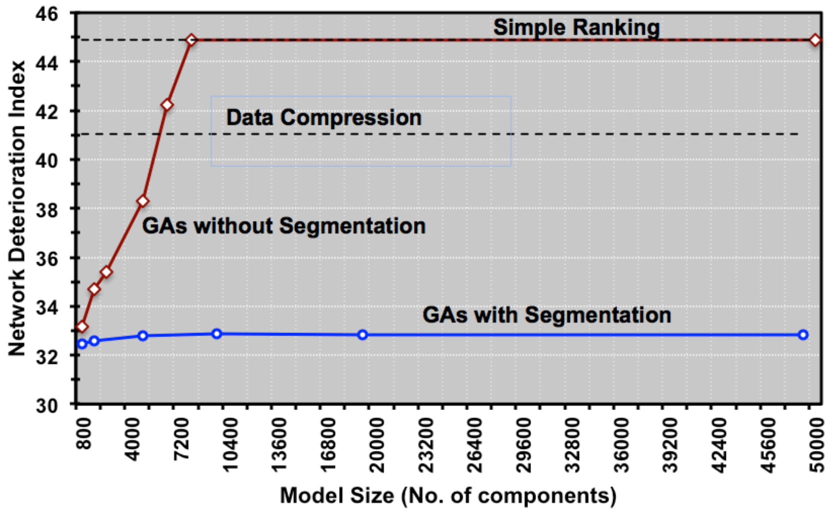

FIGURE 5 Network-level optimization using GA + segmentation

5. CONCLUSIONS

This paper builds upon an earlier work that developed an integrated life cycle cost analysis model to support asset renewal for school buildings administrated by Toronto District School Board (TDSB) in Canada. This prior work used Genetic Algorithms (GAs) to optimize asset renewal decisions for up to 8,000 components simultaneously. In the paper, three segmentation approaches: Random Segmentation; Data Compression; and Clustered Segmentation have been investigated to improve the performance of GAs in optimizing problems with larger size. After extensive experimentation, GA + Clustered segmentation proved to be the most promising. In this approach, segments are generated based on similarities among the components in terms of relative importance and deterioration behaviors. Optimum segment size was then determined based on experimentation to be 100 components. Segments were then ordered, in an ascending order, based on segment relative critically and leftover money was redistributed in a sequential manner from low criticality segments to higher criticality segments. The proposed mechanism was applied on data obtained from the Toronto District School Board and proved to be able to optimize asset renewals for 50,000 components and more with no noticeable performance degradation as the number of components increases. The proposed optimization mechanism with segmentation has the potential to be applied to data from other types of complex infrastructures systems such as bridges, highways, and water/sewer networks.

6. REFERENCES

[1] Hudson, W.R., Haas, R., and Uddin, W. **Infrastructure Management**, McGraw-Hill, New York, USA, 1997.

[2] Liu, C., Hammmad, A., and Itoh, Y., "Maintenance Strategy Optimization of Bridge Decks using Genetic Algorithm," **Journal of Transportation Engineering,** ASCE, 123(2), 1997, pp. 91-100.

[3] Thompson, P. D., T. Merlo, B. kerr, A. Cheethan, and R. Ellis, "The New Ontario Bridge Management System," **Transportation Research Circular 498**, **TRB**, 2000, pp. F-6/1-F-6/15.

[4] Macke M. and Higuchi S., "Optimizing Maintenance Interventions for Deteriorating Structures Using Cost-Benefit", **Journal of Structural Engineering**, Vol.133, 2007, 925-934.

[5] Halfawy, M., Newton, L., and Vanier, D., "Municipal infrastructure asset management systems: State-of-the-art review," **National Research Council of Canada.** Report No: 48339, 2005.

[6] Hegazy T., Elhakem A., "Multiple optimization and segmentation technique (MOST) for large-scale bilevel life cycle optimization", **Canadian Journal of Civil Engineering**, 38(3), 2011, pp. 263-271

[7] Cheung, S., Tong, T. K., & Tam, C., "Site pre-cast yard layout arrangement through genetic algorithms," **Automation in Construction**, 11(1), 2002, pp. 35-46.

[8] Osman, H., Georgy, M., and Ibrahim, M., "A Hybrid CAD-Based Construction Site Layout Planning System Using Genetic Algorithms", **Automation in Construction**, 12 (6), 2003, pp. 749-764.

[9] Heagay, T., Elbeltagi. E., El-Behairy, H., "Bridge Deck Management with Integrated Life-Cycle Cost Optimization", **TRB**, Vo. 1866, 2004, pp. 44-50.

[10] Steinbach M., Karypis G., Kumar V., **A Comparison of Document Clustering Techniques**, University of Minnesota, Technical Report #00-034, 2000.