A Parsimonious Heuristic for the Discrete Network Design Problem

Inbal Haas

Department of Civil and Environmental Engineering,

Technion – Israel Institute of Technology

Haifa 32000, Israel

Corresponding author, Tel: (972) 4 829-3608 Email: ihaas@technion.ac.il

Shlomo Bekhor

Department of Civil and Environmental Engineering,

Technion – Israel Institute of Technology

Haifa 32000, Israel

Tel: (972) 4 829-2460 Fax: (972) 4 829-5708 Email: sbekhor@technion.ac.il

This research was conducted at the Technion - Israel Institute of Technology

ABSTRACT

The problem of selecting the optimal set of transportation projects out of a given set of projects, known as the Network Design Problem (NDP), has been researched for many years. Typical transportation projects are interdependent in their nature, which turns the problem into a very complex one. When a certain objective is sought, an exact solution of the problem can be derived only by enumerating each possible project combination.

Therefore, when a large set of possible combinations is involved an alternative approach must be taken. Meta-heuristic methods usually used for this purpose do not make use of the special properties of the given problem.

This paper proposes an alternative heuristic that simplifies significantly the solution process. The benefit of a certain combination of projects is inferred based on a subset (pairs or triplets) of projects. The proposed heuristic is tested on simple networks and applied for a real-size network. The paper also discusses the trade-offs between solution accuracy and computation time.

KEYWORDS: Network Design Problem; Project Selection; Project Interdependencies; Traffic Assignment.

INTRODUCTION

Transportation infrastructure projects typically involve substantial capital investments. Therefore, the process of selecting the most promising set of projects should be performed with great care. In order to select the optimal set of projects, the decision maker evaluates each project with respect to several criteria. When transportation infrastructure projects are considered, these criteria generally include the system time as a key performance measure. After all the criteria are evaluated, the selection of the most promising set of projects can take place.

Though simply described, the aforementioned process can be quite exhausting. Key factors that increase the problem complexity are the project definition and project interdependencies. This paper, as with several other studies on the subject, considers that the set of candidate projects is fixed. In transportation networks, it is reasonable to assume that regardless of the project definition, each of the examined projects delivers a benefit with respect to a certain criteria when implemented alone. However, when several projects are to be implemented in the same network, the overall benefit is affected by the interdependencies between the different projects. This issue alone bears a great significance when considering the optimal set of projects. In effect, in many cases, no simple method exists for determining the overall benefit, which also considers interdependence relations.

If the set of alternative projects is relatively small, it is possible to find the optimal combination by simply enumerating all possible project combinations individually. This procedure, which is of exponential complexity, turns the project selection process into a very challenging one, especially when a large set of potential projects is considered.

In this paper we present a parsimonious method for estimating the system time of different sets of projects. The proposed method obviates the need for calculating the system time of each combination of projects, and provides a fair evaluation of the system time of all possible combinations (including for the ones that were not explicitly calculated). The proposed method allows for fast computation of the system time for any combination of projects. Since this method only provides an approximate value for the system time of a specific combination of projects, there is a need to verify its accuracy. This is performed for typical network configurations. In addition, there is a need to test if the additional projects do not increase the system time, as in the well-known Braess paradox (Braess, 1968). Following previous papers on this subject, this paper presents a method to prune projects that are not beneficial in terms of system time savings.

The paper is organized as follows. The following section reviews previews studies concerning the problem of project selection, with emphasis on transportation projects. Next the proposed method is presented and tested using two case studies, one of them of a real-size network. The last section concludes and suggests some further research directions.

LITERATURE REVIEW

The problem of finding the optimal set of projects out of a large set of candidate projects is not new in operations research. A large body of research has been devoted to this topic, dating back to the late 1950's. The first versions of this problem concentrated on finding the optimal set of research and development (R & D) investments in firms (Mottely et al., 1959; Pound, 1964; Moore and Baker, 1969). The scope was extended to include also studies in the public sector in general (Vonortas and Hertzfeld, 1998; Chapman et al., 2006; Huang et al., 2008) and in transportation in particular (Avineri et al., 2000; Iniestra and Gutiérrez, 2009; Shang et al., 2004). The extensive number of studies on project selection regarding transportation investments is not surprising when considering the fact that transportation projects involve large capital expenditures, and are likely to have a great impact on road network users.

Different approaches have been proposed for solving the project selection problem, such as benefit measurement, mathematical programming, decision and game theory, simulation, heuristics, and cognitive emulation (Heidenberger and Stummer, 1999). Choosing the right technique is not an easy task since each approach has its own advantages and drawbacks (Iamratanakul et al., 2008).

One of the biggest challenges, common to all the aforementioned methods, is the selection of projects when the benefits of different project interact with each other. Namely, when the overall gained benefit from a mutual implementation of several projects differs from the sum of individual benefits that each project delivers. This interdependency can sometimes increase the overall benefit, and then the projects are said to be complementary; in other times, the interdependency decreases the benefit, and then the projects are considered substitutive (Teng and Tzeng, 1996). When either one of the cases occurs, the projects are considered interdependent.

Apart from benefit interdependency, some other types of interdependencies have also been recognized; resource interdependency exists when the overall cost of several projects differs from the sum of individual costs of each project. In addition, sometimes outcome interdependency may exist. In this case the execution of a certain project is required before another one can take place (Weingartner, 1966; Czajkowski and Jones, 1986). This paper focuses exclusively on benefit interdependency.

The problem of project selection that considers interdependencies has been introduced already in the 60's together with first attempts at addressing this problem. At first, the formulation of the problem and the proposed algorithms addressed only a second-degree dependency. Namely, the change of benefit which is caused by the insertion of pairs of projects (Weingartner, 1966; Czajkowski and Jones, 1986; Schmidt, 1993; Reiter, 1963). Only later were higher order interdependencies formulated. These formulations took into account also the dependency between larger sets of projects (triplets, quartets, etc.) (Santhanam and Kyparisis, 1996), highlighting the need to develop better techniques for addressing this issue.

Many times the evaluation of the interdependency itself is very difficult. In these cases, methods which are based on subjective judgment are usually applied like the Analytical Network Process (ANP) (Shang et al., 2004). In these cases, experts' opinion is used to evaluate the interdependencies. One of the main disadvantages of this method is that it is highly dependent on the knowledge of the chosen experts on the relevant topic (Ravi et al., 2005). In addition, some issues cannot be easily assessed, and demand the integration of analytical tools to assist in the decision making process. System time, which stands in the centre of the current study, is an example of such an issue.

The total travel time, also referred as system time, is considered one of the key performance measures of every transportation system. When considering new enhancements of an existing transportation system or the development of new transportation system, like in the Network Design Problem, this is one of the most common performance measures to take into account, and hence it is also used in this paper.

The Network Design Problem (NDP) aims at finding the optimal set of links that should be improved in a road network in order to achieve a certain objective (minimize congestion, pollution, or energy consumption) (LeBlanc, 1975). A distinction is made between the discrete problem and the continuous one. In the discrete problem predefined set of projects designated for improving the network is considered; links are chosen for addition or improvement and their additional capacity is defined in advance. In the continuous problem on the other hand, a more general approach is taken, and a decision is made regarding the links that will serve as part of the network, and their appropriate capacities (Abdulaal and LeBlanc, 1979). Both problems have been extensively investigated in the literature (Magnanti and Wong, 1984; Yang and Bell, 1998; Farahani et al., 2013), and still serve as a fertile ground for research today (Xie, 2014; Liu and Wang, 2015).

As mentioned above, the projects themselves can take different forms. When discussing non-urban networks, the projects usually involve capacity expansion (either by lane addition, or section addition). In urban networks, where the available feasible resources are more limited, it is relevant to also take into consideration projects which alter the current configuration of lanes. Namely, in these projects a fixed number of lanes is considered, and reallocation of their direction according to the flow is executed. If for example one direction is highly congested with respect to the other one, more lanes can be allocated to the congested direction on the expense of the other. This is considered a relatively simple solution which requires neither large financial investment, nor the usage of more space. These types of projects have also been widely investigated in the literature, in the context of urban network improvement (Farahani et al., 2013).

Each improvement conducted on the links, will most certainly have an effect on the operation of the adjacent intersections. This fact brought to development of a series of studies concentrating on the reciprocal relation between the changes of link configuration and/or capacities and the effect these have on signal settings (Cantarella et al., 2006). This type of studies involves both discrete variables (used for the lane allocation configuration) and continuous ones (used for the junction signal settings) and therefore is referred to as the Mixed Network Design Problem. Others studies concentrate exclusively on the optimization of signal setting (Cascetta et al., 2006; Cantarella et al., 2012; Maher et al., 2013). In these cases, the dynamic aspects of the problem are usually taken in to account, as the route choice of the drivers is also determined according to the intersection delay. Thus, for this purpose, methods like dynamic stochastic assignment or Monte Carlo simulation are usually used. Still, many other studies choose not to focus their attention on the expected effects on the signal control (Lee and Yang, 1994; Gao et al., 2005). This approach is also taken in the current study, as we concentrate on the discrete NDP problem, solved using static traffic assignment, and leave the discussion on signal control for further research.

One of the reasons that led to the development of the continuous problem is the difficulty to solve its discrete version (Abdulaal and LeBlanc, 1979), since every solution of the discrete problem requires solving the traffic assignment problem. Thus, at first, when intending to solve the discrete problem explicitly, one had no other choice but to enumerate all potential different networks (Li et al., 2012), and hence as a result, only a small number of projects could be considered in these cases. Other attempts have been made using mathematical programming tools such as Branch and Bound method (LeBlanc, 1975), but those still suffered from great computational burden, once implemented on large networks (Farvaresh and Sepehri, 2013).

Further studies involved the formulation of the problem as a bi-level, where the upper level represents the system planners' prospective, seeking to minimize system cost or reduce externalities, and the lower level represents the users prospective, who seek to minimize their system time (LeBlanc and Boyce, 1986; Jiang and Szeto, 2015). For that purpose several algorithms were used including Branch and Bound (Farvaresh and Sepehri, 2013) and the Benders Decomposition method (Gao et al., 2005; Fontaine and Minner, 2014). In some other cases a multi-objective formulation of the problem was adopted (Xie, 2014). Luathep et al. (2011) transformed the problem into a mixed-integer linear programming one, and formulated the user equilibrium condition as a variational inequality problem. Then, the problem was solved using the cutting constraints method. Wang et al. (2013) developed relaxation based method, which integrates both the user optimum and the system optimum and provides the exact solution of the problem. Recently this method was further developed to account also for the optimal capacities of the newly added links (Wang et al., 2015).

It should be noted that one of the biggest advantage of the aforementioned methods is that they reach an optimal solution. Their main drawback is the cost incurred, in terms of computation complexity. This issue alone hinders the use of these methods once a solution for large networks is required, and contributed to the use of meta-heuristic methods for this purpose. Among these methods one can mention the genetic algorithm (Xiong and Schneider, 1995; Hsieh and Liu, 2004; Cantarella and Vitetta, 2006), simulated annealing (Lee and Yang, 1994), the ant colony (Poorzahedy and Abulghasemi, 2005; Vitins and Axhausen, 2008). In other cases, a combination of several heuristics was used (Bagloee et al., 2013).

The use of meta-heuristic methods, which have been successfully applied for solving the discrete NDP do not utilize the special characteristics of the problem, and in many cases do not eliminate the need to perform a large number of traffic assignments. The present paper addresses this problem, by proposing an alternative approach, which considers different aspects of the problem, both in terms of the simplicity of the solution procedure and computation time.

When discussing the discrete NDP problem and potential improvements of the road network, one of the main issues that arises is the need to account for the Braess Paradox (Braess, 1968). This paradox might come into effect, under certain conditions that are set by the link congestion function and the total demand for travels in the network (Pas and Principio, 1997; Penchina, 1997). The link congestion function of the different links must follow a certain structure for the paradox to take place. This fact alone decreases dramatically the probability of occurrence of the paradox.

Nonetheless, several strategies have already been suggested in order to avoid the paradox if it does occur. Some of them focused on eliminating the paradox in advance. For example by calculating the so called "reserve capacity" of the network with respect to the new project (Yang and Bell, 1998b). That is, evaluating the capacity of the network with respect to the given demand with and without the proposed project. A decrease in this capacity may suggest that by implementing the new project, the overall system time in the network is expected to increase. Abrams and Hagstrom (2011) proposed a method for detecting problematic links that is based on linear programming, while Park (2009) focused on finding the same links using stable dynamics model.

In spite of their effectiveness, the aforementioned methods might not be suitable for large-scale networks, due to their computational complexity. For this reason, a heuristic method was developed, that can also be applied on large-scale networks (Bagloee et al., 2014; Sun et al., 2015). This method involved two stages. At first, links whose inclusion in the network increased the overall system time were identified. Then, using the genetic algorithm, the system time was minimized by removing links of the network, out of the links previously found.

Similarly to Bagloee et al. (2014) the heuristic that we propose in this paper also involves an early identification of problematic single projects that cause the Braess Paradox, but in addition we also identify problematic pairs of projects that might cause the paradox. In the next section, the heuristic is introduced, and then demonstrated using two case studies.

METHODOLOGY

Problem definition

As described in the sections above, the central problem in this study is to find the selected set of projects to maximize the benefit of a given network, given a fixed set of infrastructure projects and a fixed budget. In the previous section it was indicated that the system time is a generally used performance measure of a road network, and is calculated as:

$$T = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} \cdot t_{ij} \qquad (1)$$

where, T is the system time, f_{ij} is the flow on each link (i, j) and t_{ij} is the travel time on each link, and there is a total of n links in the network.

By ignoring the interdependencies between different projects, the problem can be formulated as the classic knapsack problem (Dantzig, 1957). However, consideration of the interdependencies between the benefits of different projects leads to the following formulation, similar to Santhanam and Kyparisis (1996):

$$max \sum_{i=1}^{N} x_i \cdot b_i + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} b_{ij}x_ix_j + \sum_{i=1}^{N-2}\sum_{j=i+1}^{N-1}\sum_{k=j+1}^{N} b_{ijk}x_ix_jx_k + \cdots + b_{1..N}\prod_{i=1}^{N} x_i \qquad (2)$$

s.t.

$$\sum_{i=1}^{N} x_i \cdot c_i \leq B \qquad (3)$$

$$x_i \in \{0,1\}, \mathrm{int}$$

where, x_i is the decision variable, indicating whether the i*th* project of N projects is selected for implementation; b is the benefit obtained from the implementation of a set of projects. Namely, when a certain set is implemented each possible combination of projects of this set is likely to deliver a certain benefit, and the above formulation considers every possible benefit of each of the possible combinations; c_i stands for the costs of every project, and B for the overall budget. This formulation also explains why the complexity of the total benefit computation increases so rapidly as more and more projects are inserted into the list of potential projects.

Unlike the formulation proposed by Santhanam and Kyparisis (1996), here it is assumed for simplicity that interaction exists only among the benefits themselves, and that the there is no interaction between the costs of different projects.

The Genetic Algorithm (Holland, 1975) is a commonly used method for solving combinatorial optimization problems. However, when discussing real-size problems with especially large solution space, the Genetic Algorithm (GA) demands increasing amount of resources. One of the main reasons for that is the population size. A too small population size leads to a premature convergence, while a too large population size increases significantly the required running time (Goldberg et al., 1991). Nonetheless, when discussing real-size problems, a large population size is inevitable in order to avoid premature convergence. Therefore, the challenge of reaching an accurate solution, while exploiting as little resources as possible, remains relevant. In contrast to the GA method, the present heuristic puts emphasis on decreasing the running time needed to reach a solution, by exploring the problem characteristics.

In this paper, the benefit is defined as the difference between the system time when no projects are implemented and the system time of a certain combination. Since the system time when no projects are implemented is fixed, the attractiveness of a certain combination increases as the system time decreases. Hence, for convenience, from here on the discussion focuses on the system time of a certain combination, without relating to the reference value of the system time when no projects are implemented.

The system time for a given network is calculated by solving the traffic assignment problem. In this paper, we assume for simplicity the static deterministic user equilibrium problem, which has a unique solution for monotonically increasing cost functions. More complex traffic assignment models can be used, which might produce different system times and consequently a different set of projects comprising the optimal combination. However, for the purposes of this paper we will assume the well-known static assignment, to allow for easy interpretation of the interdependencies of each possible combination of projects as presented above.

The proposed method

The number of possible combinations of N projects is $2^{N} - 1$ (assuming that at least one project is chosen). When N is large, computing the system time of every possible project combination is not a viable option because of the need to perform a very large number of traffic assignment runs. Therefore the goal is to develop a method that obviates the need for computing the traffic assignment for each possible project combination, but at the same time can provide an adequate level of accuracy.

Several fundamental principles are used for the development of the proposed heuristic. First, since the problem concentrates on improving the performance of the road network, each project is supposed to deliver a positive benefit. Namely, implementation of any of the candidate projects should decrease the overall system time of the network (for the discourse regarding this assumption, see section 0).

Moreover, implementation of more projects in a network, where several projects are already implemented, should further decrease the overall system time. Hence, theoretically it can be argued that the minimal system time can be achieved by implementing all the potential projects on the network (when no budget restrictions exist).

According to the assumptions above, the system time is bounded by two limits: its highest value is obtained when no projects are implemented in the network (which is equivalent to the base case), and its lowest value when all the projects are implemented. Each project combination that is implemented in the network decreases the system time, but it still stays in the range defined above. It is important to note that we also assume that the cost of a single project does not exceed the fixed budget.

In order to further confine the range of system time values, it is worth considering the different possible combinations of projects by inspecting the single contribution of each project as follows. For a certain combination, each single project delivers a certain benefit when implemented alone in the network (i.e. decreases the system time). Therefore, given a certain combination, it is possible to find the project that delivers the maximal benefit, when implemented alone in the network. Thus, when implementing all the projects of a certain combination, the overall benefit is at least as high as the benefit achieved by implementing the most promising project of the combination. As a result, the range of possible system times, as previously defined, can be tightened even further.

To illustrate the concept above, consider a case with 7 candidate projects, of which 3 projects are considered for implementation: 1, 4, and 5. Given that project no. 4 yields the lowest system time, the range of possible system times when all 3 projects are implemented is as shown in Figure 1.

It is still needed to determine how much each additional project contributes to the decrease of the system time, in addition to the decrease in the system time obtained as a result of implementing the most promising project. If there is no additional information about the network, then it is assumed that all additional projects contribute equally to the reduction of the system time. Therefore, the only factor that dictates the system time, apart from the contribution of the most promising project of the combination, is the number of projects included in the combination. Based on the concepts presented, the system time can be calculated as follows:

$$T\_single_c = \min_{i\in c} T_i - \frac{\mathrm{y}-1}{\mathrm{N}-1}\cdot\left(\min_{i\in c} T_i - T_N\right) \qquad (4)$$

where, y is the number of projects in the current combination, T_single_c is the system time when combination c is implemented, T_i is the system time when only project i is implemented and T_N is the system time when all N projects are implemented.

The presented method makes use of the traffic assignment model only for the calculation of the system time when each of the candidate projects is implemented alone in the network, and when all the projects are implemented. Thus, when referring to the time complexity of the problem, there is a substantial decrease from $o(2^{\mathrm{N}})$ to $o(\mathrm{N})$.

A further refinement of the above formulation is to perform traffic assignments for every possible pair of projects. This will contribute to the accuracy of the overall estimation of the system time, while causing a moderate increase of the time complexity (moving from $o(\mathrm{N})$ to $o(\mathrm{N}^2)$). Hence, in the updated estimation, the upper bound of the system time is calculated based on lowest system time of all the pairs comprising a combination. Returning to the example presented earlier, given that we have 7 candidate projects, from which projects no. 1, 4, and 5 are considered for implementation, the minimal system time based on each possible pair of projects dictates the upper bound of them all, as shown in Figure 2.

The updated formulation is similar to the previous one:

$$T\_pair_c = \min_{i\in c} T_i - \frac{\mathrm{y}-2}{\mathrm{N}-2}\cdot\left(\min_{i\in c} T_i - T_N\right) \qquad (5)$$

where, T_i is the system time when implementing the most promising pair of projects in the combination C out of the projects considered. It is possible to further tighten the suggested bounds, by calculating all possible triplets, quarters and so on, and to update Equation 5 accordingly, as shown in Equation 6.

$$T\_multiple_c = \min_{i \in c} T_i - \frac{\mathrm{y} - \mathrm{p}}{\mathrm{N} - \mathrm{p}} \cdot \left(\min_{i \in c} T_i - T_N \right) \qquad (6)$$

where, p stands for the number of projects used to determine the upper bound (referred from here on as the base combination), and T_i stands for its system time. As the number of projects included in the base combination increases, a further increase of the overall computation time is expected, since more traffic assignments will be required for estimating the system time of every combination. Therefore, one should find the right balance between the affordable computation time and the required accuracy of the estimation.

After computing all the necessary traffic assignments, obtaining the optimal set of projects is quite straightforward - that is, simply calculate the system time for every possible combination of projects, and check whether the selected set meets the budget constraint.

A closer look reveals that a much simpler approach can be taken in order to find the optimal combination. Since the cost of each project is known in advance, the maximum number of projects that can be implemented (k) can be easily calculated, by sorting the projects according to their costs, and choosing the least expensive ones, until the budget is exhausted. Since each project further decreases the overall system time, it is advantageous to insert as many projects as possible, to get a lower system time.

Using Equation 6, it is possible to calculate the system time of every possible size of combination. Going from the minimum system time found upwards, a search of the combination that yields the lowest system time should be performed. This combination should, of course, include the set of projects which is responsible for the generation of the relevant system time. The iterative process of finding the optimal combination of projects, for the case where the base combination includes 2 projects is demonstrated in Figure 3. In each iteration there is a need to define the composition of the base combination which will be used to determine the overall system time, and to define the examined combination.

Considering the Braess Paradox

As indicated at the beginning of section 0, one of the assumptions of the proposed heuristic is that each candidate project contributes to a further decrease of the overall system time. However, it might not always be the case; some projects might increase the overall system time, as in the well-known Braess paradox (Braess, 1968). In order to prevent this from happening, several examinations based on traffic assignments are proposed. The system time for each case should be lower than system time of the base case, where no projects are implemented. If one of the examinations fails, i.e. the Braess Paradox was detected, the project or pair of projects are assumed to be out of the candidate set. The examination process includes performing the following traffic assignments:

- Implementation of every single potential project (N traffic assignments)
- Implementation of all but a single project (N traffic assignments)
- Implementation of every pair of potential projects ($\binom{N}{2}$ traffic assignments)
- Implementation of all but a pair of projects ($\binom{N}{2}$ traffic assignments)

Passing these examinations still cannot ensure a complete avoidance of the Braess Paradox, since the paradox may come into effect only for specific sizes of combinations which are not examined (more than 2 and less than N-2). However, to the best of the authors' knowledge, no practical method exists for detecting combinations of projects that cause the Braess Paradox for large networks. In one of the recent studies conducted regarding this subject (Bagloee et al., 2014; Sun et al., 2015), identification of problematic projects or links was done only for the first order (single links only), while here we also consider second order combinations (pairs of links).

Solution accuracy

At first, the proposed heuristic is examined using three typical network configurations: hub-and-spoke, ring and grid, determined similarly to Xie and Levinson (2007). For each network, projects are defined by improving a certain link in two different ways, either by increasing its capacity or by decreasing its free-flow travel time. This means that a maximum of 2 projects can be defined for each directional link. The networks and the defined projects are shown in Figure 4.

For these simple networks, the system time can be calculated for every possible combination of projects, by performing traffic assignments for every combination. The convergence criterion for each assignment was set to 0.001 difference of system time. Then, using the proposed heuristic, the system time is calculated for the same set of scenarios for each network. This is performed while increasing gradually the size of the base combination. At first, the base combination includes a single project, then 2 projects and finally 3 projects. It should be noted that all projects were also tested for detection of the Braess Paradox according to the process described in section 0, and no problematic project combination was found. The information regarding each network together with a comparison of the results of the proposed heuristic with those obtained from the traffic assignment are presented in Table 1.

As expected, the accuracy of the results increases with the size of the base combination, i.e. as more projects are used to determine the upper bound of the system time. However, the increase in the accuracy is quite moderate. In addition, all the deviations from the full solution of the system time, as obtained from the traffic assignment, are lower than 10%. This shows the relatively good accuracy of the proposed heuristic.

When closely examining the proposed heuristic it becomes clear that the interdependencies are only taken into account by the upper and lower bound. Furthermore, by any addition of a project to a combination an implicit assumption is made, that each additional project beyond those already used for the determination of the upper bound, is independent of the other, and contributes to the decrease of the system time equally. Thus, it was important to assess how much this implicit assumption reduces the accuracy of the solution, once the examined set of projects is highly interdependent.

Note that for the examples above, all projects are highly interdependent, because the networks are relatively small. That is, the projects are not topologically far from each other. Therefore, it is expected that proposed heuristic would function poorly by giving far-off system time estimations, compared with the times obtained using the traffic assignment algorithm. The fact that the results based on the heuristic are less than 10% away from the real values of the system time is very promising, when considering the use of the proposed heuristic on real-size networks.

As mentioned above, one of the main challenges of the current algorithms in use is the running time issue, especially when considering a large number of potential projects. When using traffic assignment, the duration of each assignment is fixed, and therefore an increase in the number of potential projects will only increase the total number of assignments needed, and not the duration of each assignment. That is why here the total running time will be dictated by the number of projects comprising the base combination, which will determine the upper bound of the system time.

After demonstrating the proposed heuristic on small networks, in the next section a case study using the Sioux-Falls network is presented.

FIRST CASE STUDY - SIOUX-FALLS NETWORK

In order to examine the accuracy of the proposed heuristic the well-known Sioux-Falls network is chosen with 12 potential projects. This network includes 24 nodes, 76 links and the total demand is 360,600 trips. The 12 most congested links in the network are identified, and each one of them is considered as a candidate for expansion. The network with the potential projects is shown in Figure 5.

A total of 4096 traffic assignments are performed for every possible combination of projects. The longest system time of all scenarios is obtained for the base case, where no projects are implemented, and the shortest system time is obtained for the case where all the projects are implemented. Thus, the Braess Paradox does not come into effect in this network configuration, with this given set of projects. The proposed heuristic is performed based on single projects and on pairs of projects and triplets. The deviation from the traffic assignment results are shown in Table 2. As can be noticed, the deviation of the results obtained using the proposed heuristic from the traffic assignment results are quite moderate, and do not change dramatically depending on the size of the base combination used. The deviation even increases slightly when moving from single projects to pairs and then to triplets. However, an examination of the maximum deviation shows that as expected the maximum deviation decreases with the increase of size of the base combination.

A graphic comparison of the results obtained using the proposed heuristic based on the use of pairs of projects and the results of the traffic assignment is presented in Figure 6. As can be inferred from this figure the system time evaluations are ordered according to the size of the combination used for the estimation. Here we used pairs of projects to determine the system time of combinations of projects with an increasing size from 3 to 11. That also explains the pattern that can be seen in the figure, where the dots are ordered on a bundle of semi-lines, and each line represents different size of combination. The trend line differentiates between two groups of combinations:

- Combinations whose estimated system time using the proposed heuristic is higher with respect to system time using traffic assignment (lie to the left of the trend line)
- Combinations whose estimated system time using the heuristic is lower with respect to the system time using traffic assignment (lie to the right of the trend line)

SECOND CASE STUDY - WINNIPEG NETWORK

Next, we want to test our heuristic on a real-size network. In addition to the system time estimation, we also find the optimal project combination given a budget constraint. For this purpose we use the road network of the city of Winnipeg, which is provided in the EMME/2 software and used in several papers (e.g. Bekhor et al., 2008). This network includes 154 centroids, 903 nodes, 2,231 links, and the total demand is 54,459 trips. The demand is fixed, and for simplicity, the projects are carried out at a single time point. Therefore, the costs of the projects are not capitalized. In addition, for simplicity, a single mode is considered and the traffic assignment equilibrium is not calculated for public transportation.

At first, in order to decide on the set of candidate projects, a "base case" traffic assignment was performed, just as for the Sioux-Falls network, the 30 most congested links were identified and served as candidates for expansion. Each candidate project is defined as the duplication of its current capacity. In addition, in order to take into account also cases where an expansion of the road is not feasible, 3 additional projects were defined. These projects considered the reallocation of existing lanes; that is, adding more capacity to one direction on the expense of the other one. For this purpose, sections with highly uneven flows for each direction were identified. Sections in which these unbalanced flows were not accommodated by an adequate distribution of lanes, were added as candidate projects, and their capacities were changed accordingly. Description of the lane configuration with and without projects for these 3 sections is shown in Table 3. Note, that in case several options for reallocation of the lanes exist, as in project no. 32, the configuration that yields the lowest system time was chosen.

The distribution of the projects in the network is shown in Figure 7. In order to verify that the Braess Paradox does not occur, traffic assignment was computed according to the procedure described in section 0. Based on the traffic assignments performed, no problematic combinations yielding longer system time than the base case were detected.

The traffic assignment was also computed for every triplet of projects, and for the case where all the 33 projects are implemented in the network. As before, in order to test the proposed heuristic, the system time obtained by the traffic assignment of each pair of projects and every triplet of projects was compared with the result of Equation 4. The average deviation of the traffic assignment results from the estimated ones was 0.15% with standard deviation of 0.08% and maximum deviation of 0.82%.

Next, the results of the traffic assignment of every triplet are compared with the results of Equation 5, when using the same projects. The expectation was that the deviation between the traffic assignment results and the estimation would decrease. Indeed, this expectation was confirmed ,as the average deviation from the computed system time using traffic assignment is 0.10%, with standard deviation of 0.03% and maximum deviation of 0.30%.

At the next stage, the proposed heuristic was used to find the best combination of all 33 projects, given a budget constraint. For this purpose, the cost of each project was estimated. In this paper, it was assumed an average expansion cost of 10 million dollars per kilometre road. The reallocation of existing lanes incurs relatively low costs, especially when compared with the costs of lane expansion. Therefore the costs of all 3 additional projects which involved the reallocation of existing lanes were disregarded, and they were assumed not to incur any costs. In addition, a budget constraint of 50 million dollars was set, which corresponds roughly to the median of the 33 projects.

The estimation was performed according to the algorithm outlined in the previous section, and the best combinations based on pairs of projects and on single project as the base combination were found, together with their corresponding system time values. When using pair-based combinations, 787 different combinations consisting of 23 projects each, obtained the best system time value which equals to 14,962.8 vehicle hours. All these combinations were based on the same pair of projects (no. 7 and 27) functioning as the base combination. When estimating the system time using single-based combinations, the best system time was 14,980.8, where 47 different combinations yield this value, each made out of 24 projects. The base combination in this case was project no. 7.

The reason that all best combinations share the same system time values is that once the most promising project or pair of projects in a combination is found, the only thing that dictates the overall system time is the total number of projects in the combination. That is why it is possible to generate many different combinations with the same number of projects which yield same system time value, as long as the budget constraint is not violated.

The best single-based and pair-based combinations were quite similar, apart from the fact that project 27 which was included in each best pair-based combination was not included in any single-based combination. The reason for that is as follows. Project no. 7 was included in each of the best single-based combinations, consisting of 24 projects each. Projects no. 7 and 27 were included in the best pair-based combinations consisting of 23 projects each. Had projects no. 7 and 27 been included in the best single-based combinations including 24 projects, then the best combination size of pair-based combinations should have been also 24 projects; hence, contradicting the fact that the best pair-based combination size is 23 projects only.

The traffic assignment for each of the 10 best pair-based combinations was performed. This helped to ensure that the Braess Paradox does not occur. It also further allowed for a comparison of the results obtained using the proposed heuristic (based on single projects and pairs of projects) to take place. The deviation of the system time between the proposed heuristic and the traffic assignment of the 10 best pair-based combinations is quite small also in this case and ranges from 0.07% to 0.12%.

The system time for the best combination was normalized to 1.00. The results of the normalized system time of the 10 best combinations, together with the number of projects in each of the chosen combinations, and the deviations from the proposed heuristic are also presented in Table 4. Each coloured cell represents a project that was selected for implementation. The cells in black comprise the base combination, which was used to determine the system time for the whole combination. In this respect, it is important to note that although it might seem that only a pair projects is responsible for the system time of the combination, it is not the case. That is because the most promising pair of projects was selected with respect to all the projects in the combination, as explained in the methodology section.

Running times

Since running time issues are of upmost interest when discussing a problem as this, it is necessary to check the performance of the proposed heuristic with respect to other methods within this context. Such an examination is performed using a growing number of single projects for implementation (5 to 20), where full enumeration and GA were compared to the proposed heuristic when applied using a base combination that included at first pairs of projects, and later on triplets of projects. Each assignment run of the Winnipeg network took approximately 1 second on an Intel i5 CPU with 8Gb RAM. The population size of the GA algorithm is set to the number of candidate projects in each problem considered, in order to decrease the chances of premature convergence of the algorithm to a non-optimal solution. The results of this comparison between the different running times of all methods, presented in logarithmic scale, along with their trend lines (which in most cases converge with the original curves) are shown in Figure 8.

Figure 8 shows that as expected, running times increase together with the increase in the number of single projects considered for implementation. Moreover, the differences in the running times between the three methods (i.e. full enumeration, GA and the proposed heuristic) are also evident, where the proposed algorithm, when used based on single projects, requires the lowest running time, and is of the lowest complexity. Figure 8 also demonstrates the increase of time complexity when shifting from one version of the proposed heuristic to another, starting with linear complexity (for single projects) and ending with complexity of $O(\mathrm{n}^3)$ (for triplets of projects). The GA method, which in this case seems to follow an exponential complexity, lies between the running time estimations of the proposed heuristic based on pairs and triplets of projects.

When comparing the running times of the proposed heuristic in its different versions, based on single, pairs or triplets of projects as the base combination, the advantage of working with single projects is clear with respect to the decrease in the required running time. However, as discussed in the previous sections, higher accuracy can be gained by using pairs of projects, while still maintaining reasonable running times even when compared with the GA method.

SUMMARY AND CONCLUSIONS

In this study a heuristic was proposed for the solution of the discrete NDP problem. The proposed heuristic makes use of the properties of the problem as other methods recently developed (Wang et al., 2013), instead of using common meta-heuristic methods. Unlike other methods, the major advantage of the proposed method is that it can be applied to large scale networks quite easily, with a large set of potential projects and in a reasonable time. Increasing the number of projects is usually a problematic issue both for methods which rely on the capability to solve linear programming problems and for meta-heuristic methods. For the first type of methods because of the sharp increase in the number of variables involved, and for the latter due to the dramatically growth in running time requirements with the rise in problem size.

The proposed method is based on a different approach that do not require solving integer programming problems, and thus it can also be applied when a large set of potential projects is involved. It also does not rely on the convergence of meta-heuristic based methods, which also contributed to the relatively short running time it requires.

The main disadvantage of the proposed heuristic is that it does not reach an accurate solution, but only an approximation. However, according to the tests performed, this inaccuracy is not substantial when compared with the results obtained using traffic assignment.

The proposed heuristic was tested using three typical network configurations and also the Sioux-Falls network, and applied in a case study of the Winnipeg network.

Using the suggested heuristic, the process of project selection of infrastructure related projects, when interdependencies are issued can be significantly simplified, and therefore can be of great value in the hands of decision makers.

This paper used the static traffic assignment with fixed demand. The method can be extended to accommodate public transport projects, by changing the model that performs traffic assignment to a combined split and assignment model, or more complex transportation models. This can be performed, provided that these models can produce the necessary metric (system time or other variable) for every single project in a systematic way.

The proposed heuristic can be further developed to include also cost interdependency, address the issue of signal settings and to try and answer problems of a more complex kind. For example, expanding the problem to include multi-objectives, where the benefit of each project is not positive with respect to each criteria. Another research direction is the development of this method for a case where not all projects are to be implemented on the same timeframe.

REFERENCES

Abdulaal, M. and LeBlanc, L.J. (1979) Continuous equilibrium network design models. Transportation Research Part B: Methodological, Vol. 13, No. 1, pp. 19-32.

Abrams, R.A. and Hagstrom, J.N. (2006) Improving traffic flows at no cost. In: Lawphongpanich, S., Hearn, D.W., Smith, M.J. (Eds.), Mathematical and Computational Models for Congestion Charging. Springer, New York, NY, pp. 1–22.

Avineri, E., Prashker, J. and Ceder, A. (2000) Transportation projects selection process using fuzzy sets theory. Fuzzy sets and systems, Vol. 116, No. 1, pp. 35-47.

Bagloee, S.A., Tavana, M. and Ceder, A. (2013) A hybrid meta-heuristic algorithm for solving real-life transportation network design problems. International Journal of Logistics Management, Vol. 16, No. 1, pp. 41-66.

Bagloee, S.A., Ceder, A., Tavana, M. and Bozic, C. (2014) A heuristic methodology to tackle the Braess Paradox detecting problem tailored for real road networks. Transportmetrica A, Vol. 10, No. 5, pp. 437-456.

Bekhor, S., Toledo, T. and Prashker, J. N. (2008) Effects of choice set size and route choice models on path-based traffic assignment. Transportmetrica, 4(2), 117-133.

Braess, D. (1968) Über ein Paradoxen aus der Verkehrsplannung. Unternehmensforschung, Vol. 12, No. 1, pp. 258-268.

Cantarella, G.E. and Vitetta, A. (2006) The multi-criteria road network design problem in an urban area. Transportation, Vol. 33, No. 6, pp. 567-588.

Cantarella, G.E., Pavone, G. and Vitetta, A. (2006) Heuristics for urban network design: Lane layout and signal settings. European Journal of Operational Research, Vol. 175, pp. 1682-1695.

Cantarella, B.E., Velonà, P. and Vitetta, A. (2012) Signal setting with demand assignment: global optimization with day-to-day dynamic stability constraints. Journal of Advanced Transportation, Vol. 46, pp. 254-268.

Cascetta, E., Gallo, M., and Montella, B. (2006) Models and algorithms for the optimization of signal settings on urban networks with stochastic assignment. Annals of Operations Research, Vol. 144 No.1, pp. 301–328.

Chapman, C.B., Ward ,S.C. and Klein, J.H. (2006). An optimised multiple test framework for project selection in the public sector, with a nuclear waste disposal case-based example. International journal of project management, Vol. 24, No. 5, pp. 373-384.

Czajkowski, A.F. and Jones,.S. (1986) Selecting interrelated R & D projects in space technology planning. IEEE Transactions on Engineering Management, Vol. 33, No. 1, pp. 17-24.

Dantzig, G. (1957) Discrete-variable extremum problems. Operations Research, Vol. 5, No. 2, pp. 266-277.

Emme/2 User's Manual: Release 9.2. INRO Consultants, Montreal, 1999.

Farahani, R.Z., Miandoabchi, E., Szeto, W.Y. and Rashidi, H. (2013) A review of urban transportation network design problems. European Journal of Operational Research, Vol. 229, pp. 281-302.

Farvaresh, H. and Sepehri, M.M. (2013) A branch and bound algorithm for bi-level discrete network design problem. Network and Spatial Economics, Vol. 13, No. 1, pp. 67-106.

Fontaine, P. and Minner S. (2014) Benders Decompositon for discrete-continuous linear bilevel problems with applicaton to traffic network design. Transportation Research Part B: Methodological, Vol 70, pp. 163-172.

Gao, Z., Wu, J. and Sun, H (2005) Solution algorithm fo the bi-level discrete network design problem. Transportation Research Part B: Methodological, Vol 39, pp. 479-495.

Goldberg, D.E., Deb, K., and Clark J.H. (1991) Genetic algorithms, noise and the sizing of populations. Complex Systems, Vol. 6, pp. 333-362.

Heidenberger, K. and Stummer, C.(1999) Research and development project selection and resource allocation: a review of quantitative modelling approaches. International Journal of Management Review, Vol. 1, No. 2, pp. 197–224.

Holland, G.H. (1975) Adaptation in natural and artificial systems, MIT press, Cambridge, MA.

Hsieh, T.Y. and Liu, H.L. (2004) Genetic algorithm for optimization of infrastructure investment under time-resource constraints. Computer-Aided Civil and Infrastructure Engineering, Vol. 19, No. 3, pp. 203-212.

Huang, C.C., Chu, P.Y. and Chiang ,Y.H. (2008) A fuzzy AHP application in government-sponsored R&D project selection. Omega, Vol. 36, No. 6, 2008, pp. 1038-1052.

Iamratanakul, S., Patanakul, P. and Milosevic, D. (2008) Project portfolio selection: from past to present. 4th IEEE International Conference on Management of Innovation and Technology, ICMIT 2008.

Iniestra, J.G., and Gutiérrez, J.G. (2009) Multicriteria decisions on interdependent infrastructure transportation projects using an evolutionary-based framework. Applied soft computing, Vol. 9, No. 2, pp. 512-526.

Jiang Y. and Szeto W.Y. (2015) Time-dependent transportation network design that considers health cost. Transportmetrica A, Vol. 11, No. 1, pp.74–101.

LeBlanc, L.J. (1975) An algorithm for the discrete network design problem. Transportation Science, Vol. 9, No. 3, pp. 183-199.

LeBlanc, L.J. and Boyce, D.E. (1986) A bilevel programming algorithm for exact solution of the network design problem with user-optimal flows. Transportation Research Part B: Methodological, Vol. 20, No. 3, pp. 259-265.

Lee, C.K. and Yang, K.L. (1994) Network design of one-way streets with simulated annealing. Papers in Regional Science, Vol. 73, No. 2, pp. 119-134.

Li, Z., Kaul, H., Kapoor, S., Veliou, E., Zhou, B., and Lee, S.H. (2012) New methodology for transportation investment decisions with consideration of project interdependencies. Transportation Research Record, 2285, pp. 36-46.

Liu, H. and Wang, Z.W. (2015) Global optimization method for network design problem with stochastic user equilibrium. Transportation Research Part B: Methodological, Vol 72, pp. 20-39.

Luathep, P., Sumalee, A., Lam, W.H.K., Li, Z.C., Lo, H.K. (2011) Global optimization methods for mixed transportation network design problem: a mixed-integer linear programming approach. Transportation Research Part B: Methodological, Vol 45, pp. 808-827.

Magnanti, T.L. and Wong R.T. (1984) Network design and transportation planning: models and algorithms. Transportation Science, Vol. 18, No. 1, pp.1-55.

Maher, M., Liu, R. and Ngoduy, D. (2013) Signal optimization using the cross entropy method. Transportation Research Part C, Vol, 27, pp. 76-88.

Moore, J.R. and Baker N.R. (1969). An analytical approach to scoring model design - application to research and development project selection. IEEE Transactions on Engineering Management, Vol. 16, No.3, pp. 90-98.

Mottely, C.M. and Newton, R.D. (1959) The selection of projects for industrial research. Operations Research, Vol. 7, No. 6, 1959, pp. 740-751.

Park, K. (2011) Detecting Braess Paradox based on stable dynamics in general congested transportation networks. Networks and Spatial Economics, Vol. 11, No. 2, pp. 207-232.

Pas, E.I., and Principio, S.L. (1997) Braess' Paradox: some new insights. Transportation Research Part B: Methodological, Vol 31, No. 3, pp. 265-276.

Penchina, C.M. (1997) Braess paradox: maximum penalty in a minimal critical network. Transportation Research Part A: Policy and Practice, Vol 31, No. 5, pp. 379-388.

Poorzahedy, H. and Abulghasemi, F. (2005) Application of ant system to network design problem. Transportation, Vol. 32, No. 3, pp. 251-273.

Pound, W.H. (1964) Research project selection: testing a model in the field. IEEE Transactions on Engineering Management, Vol. 11, No. 1, pp. 16-22.

Ravi, V., Shankar, R. and Tiwari, M.K. (2005) Analyzing alternatives in reverse logistics for end-of-life computers: ANP and balanced scorecard approach. Computers & Industrial Engineering, Vol. 48, No. 2, pp. 327-356.

Reiter, S. (1963) Choosing an investment program among interdependent projects. The Review of Economic Studies, Vol. 30, No. 1, 1963, pp. 32-36.

Santhanam, R. and Kyparisis, G.J. (1996) A decision model for interdependent information system project selection. European Journal of Operational Research, Vol. 89, No. 2, pp. 380-399.

Schmidt, R.L.(1993) A model for R&D project selection with combined benefit, outcome and resource interactions. IEEE transactions on engineering management, Vol. 40, No. 4, pp. 403-410.

Shang, J.S., Tjader, Y. and Ding, Y.A. (2004) A unified framework for multicriteria evaluation of transportation projects. IEEE Transactions on Engineering Management, Vol. 51, No. 3, pp. 300-313.

Sun, L., Liu, L., Xu, Z., Jie, Y., Wei, D. and Wang, P. (2015) Locating inefficient links in a large-scale transportation netowrk. Physica A, Vol. 419, pp. 537-545.

Teng, J. and Tzeng, G. (1996). A multiobjective programming approach for selecting non-independent transportation investment alternatives. Transportation research Part B: Methodological,Vol. 30, No. 4, pp. 291-307.

Vitins, B.J. and Axhausen, K.W. (2008) Optimization of large transport networks using the ant colony heuristic. Computer-Aided Civil and Infrastructure Engineering, Vol. 24, No. 1, pp. 1-14.

Vonortas, N. S. and Hertzfeld H. R. (1998). Research and development project selection in the public sector. Journal of Policy Analysis and Management, Vol. 17, No. 4, 1998, pp. 621–638.

Wang, S., Meng, Q., Yang, H. (2013) Global optimization methods for the discrete network design problem. Transportation Research Part B: Methodological, Vol 50, pp. 42-60.

Wang D. Z.W, Liu H and Szeto W. Y. (2015) A novel discrete network design problem formulation and its global optimization solution algorithm, Transportation Research Part E, Vol. 79, pp. 213-230.

Weingartner, H.M. (1966) Capital budgeting of interrelated projects: survey and synthesis. Management Science, Vol. 12, No. 7, pp. 485-516.

Xiong, Y. and Schneider, J.B. (1995) Transportation network design using a cumulative genetic algorithm and neural network. Transportation Research Record, 1364, pp. 37–44.

Xie, C. (2014) Bicriterion discrete equilibrium network design problem. Networks, Vol. 63, pp. 286-305.

Xie, F. and Levinson, D. (2007) Measuring the structure of road networks. Geographical Analysis, Vol. 39, No. 3, pp. 336-356.

Yang, H. and Bell, M.G.H. (1998) Models and algorithms for road network design: a review and some new developments. Transport Reviews, Vol. 18, pp. 257-278.

Yang, H. and Bell, M.G.H. (1998b) A capacity paradox in network design and how to avoid it. Transportation Research Part A: Policy and Practice, Vol. 32, No. 7, pp. 539-545.

TABLES

TABLE 1 General Information and Comparison Results of the Three Network Types and the Full Solution

General Information	Hub-and-spoke	Ring	Grid
No. of Elements (nodes / links)	(6,5)	(4,8)	(16,24)
No. of Projects	10	12	12
No. of Assignments - Full Enumeration	$2^{10} = 1024$	$2^{12} = 4096$	$2^{12} = 4096$
Method Used to Determine Upper Bound	No. of Assignments - The Proposed Heuristic		
Single Projects	10+1=11	12+1=13	12+1=13
Project Pairs	$\binom{10}{2} + 1 = 46$	$\binom{12}{2} + 1 = 67$	$\binom{12}{2} + 1 = 67$
Project Triplets	$\binom{10}{3} + 1 = 121$	$\binom{12}{3} + 1 = 221$	$\binom{12}{3} + 1 = 221$
Method Used to Determine Upper Bound	Deviation of the System Time with respect to the Full Solution (Average, SD)		
Single Projects	(7.37%, 5.19%)	(7.58%, 4.82%)	(4.19%, 2.95%)
Project Pairs	(6.07%, 4.34%)	(7.36%, 4.33%)	(4.01%, 2.54%)
Project Triplets	(5.79%, 3.93%)	(6.29%, 4.01%)	(3.75%, 2.20%)

TABLE 2 the Sioux-Falls network - deviation of the system time with respect to the Full Solution (Average, SD, Max deviation)

Single Projects	(1.62%, 1.17%, 6.56%)
Project Pairs	(1.64%, 1.12%, 5.78%)
Project Triplets	(1.71%, 1.06%, 5.70%)

TABLE 3 lane configuration for the 3 reallocation-of-lanes projects in the Winnipeg network

project number	number of lanes without project		number of lanes with project	
	A to B direction	B to A direction	A to B direction	B to A direction
31	2	2	1	3
32	3	4	2	5
33	2	2	3	1

TABLE 4 Normalized System Time for the 10 Best Project Combinations

solution number										
possible projects	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
System Time – traffic assignment	1.0000	1.0000	1.0000	1.0000	1.0001	1.0001	1.0004	1.0004	1.0004	1.0005
System Time - the proposed heuristic	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993	0.9993
Deviation from traffic assignment	0.0007	0.0007	0.0007	0.0008	0.0008	0.0011	0.0011	0.0011	0.0012	0.0007

FIGURES

FIGURE 1 Upper and lower bounds of the system time using the "best" single project as an upper bound

FIGURE 2 The upper and lower bounds of the system time using the "best" pairs of projects as an upper bound

FIGURE 3 A flow chart describing the steps for finding the optimal combination and the corresponding system time

FIGURE 4 The three typical network configurations used and the distribution of projects in them

FIGURE 5 The Sioux-Falls network with the potential projects

FIGURE 6 Comparison between the traffic assignment system times and the system times based on the proposed heuristic for each possible project combination in the Sioux-Falls network

FIGURE 7 Distribution of projects in the Winnipeg network

FIGURE 8 Running time results of all 5 methods: full enumeration, Genetic Algorithm, traffic assignment using single project, pairs of projects and triplets of projects