International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:7, No:7, 2013

Optimal Risk Reduction in the Railway Industry by Using Dynamic Programming

Michael Todinov and Eberechi Weli

***Abstract*—**The paper suggests for the first time the use of dynamic programming techniques for optimal risk reduction in the railway industry. It is shown that by using the concept '*amount of removed risk by a risk reduction option*', the problem related to optimal allocation of a fixed budget to achieve a maximum risk reduction in the railway industry can be reduced to an optimisation problem from dynamic programming. For *n* risk reduction options and size of the available risk reduction budget *B* (expressed as integer number), the worst-case running time of the proposed algorithm is O (n x (B+1)), which makes the proposed method a very efficient tool for solving the optimal risk reduction problem in the railway industry.

***Keywords*—**Optimisation, railway risk reduction, budget constraints, dynamic programming.

I. Introduction

THE railway operators and infrastructure owners are increasingly required to enhance services by introducing and implementing the best options for optimising risk reduction. In practice, the application of the "As Low As Reasonably Practicable" (ALARP) framework for risk reduction in the railway industryis a challenge, further compounded by decisions that must be made on a finite number of risk reduction options, within specified budgets. The current application of the cost-benefit technique as a decision support tool for determining the best options for risk reduction is inadequate[1] and there are advocates for alternative techniques [2].However, studies have exposed the inadequacies of applying basic economic theories in the transport industry[3]. A fuzzy-analytical hierarchy process has been proposed by [4]. The Analytical Hierarchy Process (AHP)requires the use of pair-wise comparison matrix and eigenvector to specify weights higher than a specified threshold [5], [6]. AHP does not adequately support the decision-maker in choosing alternatives that have higher weights than the threshold and are unsuitable for selecting more than one choice when multiple alternatives are present[7]. Other proponents of alternatives to the cost-benefit approach have demonstrated the application of different optimisation techniques in addressing risk reduction within budget constraints[8]–[15]. These studies apply multi-criteria methods such as AHP, *Simulated Annealing*, *Tabu Search*, *Genetic Algorithms*, *Expected Utility Theory* and combinations of these. The limitations of these approaches are well documented in [16]-[21]. A comprehensive analysis by [22], demonstrates that the optimal resource allocation problems are NP-hard problems. Studies undertakenon the suitability of optimisation techniques concluded that the optimal resource allocation is best addressed by using dynamic programming [23], [24].

In this paper, a case study of a railway line sectionhas been used to demonstrate the effectiveness and accuracy of the dynamic programming optimisation technique for a major renewal project. The accident data set has beenextracted from a 70km railway line with 34 stations, operating 33 - 35 trains daily.The railway line operates at an average speed of 60 to 70km/h line and 54 million journeys annually. The study focuses on the major accident risks on the line – Platform Train Interface (Platform-only accidents) and Collision between Trains. For the platform-only accidents, 20 available risk reduction options have been identified (Table I). The number of identified risk reduction options for the risk 'Collision Between Trains' was 81, of which only a small sample has been listed in Table II, due to space limitations. The risk reduction measures have been listed with the associated costs and risk reduction achieved.

Michael Todinov and Eberechi Weli are with the Department of Mechanical Engineering and Mathematical Sciences, Oxford Brookes University, Oxford, Wheatley, OX33 1HX (e-mail: mtodinov@brookes.ac.uk).

TABLE I
A SET OF RISK REDUCTION OPTIONS FOR THE RISK OF PLATFORM TRAIN INCIDENTS (PLATFORM-ONLY)

ID	Risk Reduction Option	Cost [x £10,000]	Removed Risk [x £ 10,000]
1	Emergency/incident management systems	100	530
2	Station defect reporting & corrective system	10	35
3	Emergency drills – station staff training	20	67
4	Crowd control procedures & systems	100	265
5	Slip, trip, fall toolkit	10	20
6	Station surface inspections/testing/renewals	100	220
7	Platform Edge Doors (half length)	800	1360
8	Audible warnings on platform	100	132
9	Access & egress from incident site	200	260
10	Support from platform supervisors	300	320
11	Painted line warnings/signage	50	530
12	Platform emergency plungers – train stops	400	3900
13	Gap fillers	200	180
14	One-person-operated CCTV systems	1200	6100
15	Stair-nose marking	50	350
16	Station supervisor/personnel training	100	660
17	Re-design/r-build platform	1000	2800
18	Platform lighting (incl. emergency lighting)	550	1300
19	Increased traffic – major events, peak times	1000	1200
20	Enhanced surfaces –platforms	350	410

TABLE II
A REPRESENTATIVE SAMPLE SET FROM 81 RISK REDUCTION OPTIONS FOR THE RISK OF COLLISION BETWEEN TRAINS ACCIDENTS

ID	Risk Reduction Option	Cost [x £100,000]	Removed Risk [x £ 100,000]
1	Train stops	70	160
2	Speed restrictions – compromised overlaps	50	170
3	On-board sanding	20	60
4	In-cab CCTV	300	130
5	Driver training – Signal Passed at Danger	30	180

II. ALGORITHM FOR SOLVING THE PROBLEM OF OPTIMAL BUDGET ALLOCATION IN THE RAILWAY INDUSTRY

Let *S* be the set of all available *n* risk reduction options i=1,2,...,n, for a particular major risk in the railway industry. As a measure of the effectiveness of each risk reduction option, we postulate the measure *amount of removed risk*. The amount of removed risk is *the expected cost of prevented accidents, delays, fatalities, injuries etc. expressed in monetary terms*. Each risk reduction measure *i*, (i=1,2,...,n) is characterised by the amount of risk rr_i it removes after its implementation. Each risk reduction measure *i*, (i=1,2,...,n) is also characterised by its cost of implementation C_i.

Each risk reduction option cannot be selected more than once. As a result, each risk reduction option from the set *S* of all available risk reduction options can either be accepted or rejected.

The task of optimal allocation of the fixed budget reduces to determining the optimal subset $P \subseteq S$ of risk reduction options, whose total sum of removed risks $max \sum_{k\in P} rr_k$ is maximum and whose total cost of implementation does not exceed the available risk reduction budget *B*.

$$max \sum_{k\in P} rr_k; \sum_{k\in P} C_k \leq B \quad (1)$$

Considering the magnitude of the implementation costs for the risk reduction optionsin the railway industry and the magnitude of removed risks, it can be assumed that the costs and the amount of removed risk can always be expressed integer numbers. These express the removed risk and the cost of implementation in thousands, tens of thousands or hundreds of thousands of pounds sterling.It is also assumed that the available budget can also be specified by an integer number.As a result, the problem of optimal allocation of a risk reduction budget in the railway industry is reduced to a combinatorial optimisation problem involving integersonly. This problem can be solved by using dynamic programming techniques [25],[26].Although the dynamic programming techniques have been known for a long time, to the best of our knowledge,in this paper,these methods have been applied for the first time to solve a problem of optimal risk reduction in the railway industry.

The advantage of the dynamic programming [23],[25],[26] consists of the fact that it finds solutions to sub-problems increasing in size, stores them in the memory and describes the solution of each sub-problem in terms of already solved and previously stored solutions of smaller sub-problems. As a result, sub-problems are solved only once, which makes the dynamic programming significantly more efficient than a brute-force method based on the enumeration of all possible subsets in the set of available risk reduction options*S*. The number of possible subsets in the set *S* is 2^n and the computational time of a brute-force method based on scanning all possible subsets increases dramatically with increasing the number *n*of risk reduction options.

The description of the algorithm in pseudo-code is presented next.

A. Algorithm 1:Building the Dynamic Risk Reduction Table

Initialising array x[][] with zeroes in the row with index '0' and in the column with index '0'.

```
for i=1 to n do
for  j=1 to B do
  {
cur_budget=j;
if (c[i]>cur_budget) then  { x[i][j]=x[i-1][j];
trac[i][j]=0; }
        else
        {
rem = cur_budget-c[i];
tmp = rr[i]+x[i-1][rem];
if(x[i-1][cur_budget] >tmp) then {
  x [i][j] = x[i-1][j]; trac [i][j]=0;
                                }
```

```
else{
   x [i][j]=tmp; trac [i][j]=1;
       }
   }
 }
```

The algorithm works as follows. The solutions of the sub-problems are kept in the arrayx[][], where the rowscorrespond to the risk reduction options and the columns correspond to the available budget. The information necessary to restore the optimal solution is kept in the arraytrac[][]. The size of the x[][] array is (n+1) x (B+1) elements. The row with index '0' of the array x[][] corresponds to zero number of selected risk reduction options in the optimal set P; the column with index '0' of the array x[][] corresponds to zero budget.

The sub-problems are defined by the size of the current budget which varies from 1 to B units. The cost of the ith risk reduction option is compared with the value of the current budget and if it is greater than the current budget, the ith risk reduction option is not included in the optimal set*P*, which is reflected by placing zero in the tracarray (trac[i][j]=0). In the case where the current budget is greater than the cost of the ith risk reduction option, a decision is taken whether to include the ith risk reduction option or not.

Initially, the statement 'rem = cur_budget-c[i];' determines the remaining budget if the ithrisk reduction option is included in the optimal set P. The sub-problem marked by x[i-1][rem] however has already been solved and its solution has been recorded in the x[][] array. The entry x[i-1][rem] gives the maximum amount of removed risk within budget equal to 'rem' and for i-1 available risk reduction options. Consequently, the solution of the sub-problem does not need to be determined again; it can simply be read out from the x[][] array. The amount of risk removed by the ithrisk reduction option is rr[i]. Consequently, the maximum amount of removed risk for budget cur_budget=j,if the ith risk reduction option is included, is given by 'tmp = rr[i]+x[i-1][rem];'. If the ith option is not included in the optimal set *P*, the maximum amount of removed risk within the budget cur_budget is given by x[i-1][cur_budget], (cur_budget=j). Consequently, the decision whether to include the ith risk reduction option in the optimal set or not, depends on the outcome of the comparison made in the statement'if(x[i-1][cur_budget]>tmp)' wheretmp = rr[i]+x[i-1][rem].

If 'x[i-1][cur_budget] >tmp', not including the ithrisk reduction option yields greater amount of removed risk and the entry'trac[i][j]=0'in the track[][] array is set to zero, which indicates that the ith risk reduction option has not been included in the optimum set of options P. The maximum amount of removed risk is equal to the maximum amount of removed risk within the current budget 'j', for i-1 total number of available options. This maximum however, has been computed and is already in the array x[][]; this is the entry x[i-1][j].

If 'x[i-1][cur_budget] <tmp', including the ith option yields greater amount of removed risk and the entry in the trac-array is set to one(trac[i][j]=1;), which indicates that the ith risk reduction option has been included in the optimal set P. The maximum amount of removed risk is equal to x[i][j]= rr[i]+x[i-1][rem].

In words, the maximum amount of removed risk is equal to the removed risk from including the ith risk-reduction option plus the maximum amount of removed risk for i-1 available options within the remaining budget 'rem'.

The optimal set of risk reduction options is restored by the next algorithm in pseudo-code.

B. Algorithm 2: Restoring the Optimal Set of Risk Reduction Options from the Dynamic Tables

```
Initialise all entries of the solution[] array with zeroes.
    cur_bud=B;
    cur_opt=n;
    tmp=trac[cur_opt][cur_bud];
    while(cur_opt> =1 ) do
          {
    if(trac[cur_opt][cur_bud]=1)  then  {
     solution[cur_opt] = 1;
                cur_bud=cur_bud - c[cur_opt];
                cur_opt = cur_opt - 1;
                  }
       elsecur_opt=cur_opt-1;
     }
```

The algorithm starts with the entry trac[n][B] of the track[][] array, which corresponds to a full budget *B* and all *n* available risk reduction options. If the n-th option has been included in the optimal set P, this will be indicated by a non-zero entry in the trac array (trac[n][B]=1). In this case, the solution array 'solution[]' marks the n-th option as 'included' in the optimal set P, by the statement 'solution[n]=1'. The current budget is then reduced by the statement 'cur_bud=cur_bud-c[cur_opt]' with the cost of the current (n-th) option. The current option to be considered should now be the n-1st option. This is ensured by the statement 'cur_opt=cur_opt-1'.

If the n-th option has not been included in the optimal set, this will be indicated by a zero entry in the trac-array (trac[n][B]=0). In this case, the current budget is not reduced because no cost has been incurred for implementing the n-th risk reduction option.

The process of considering the options in reverse order continues, until the first option is reached. At this point, the entries of the solution array will contain '1' for options which have been included in the optimal set P and '0' for options which have not been included in the optimal set P.

The running time of Algorithm 1building the dynamic table, is determined by the two nested loops:'**for**i=1 **to** n **do**'and '**for** j=1 **to** B **do**', which contain a set of operations that are executed in constant time. The maximum number of steps, after which Algorithm 1 will terminate,is $n \times B$. The maximum number of steps performed by Algorithm 2 is *n*, because after each iteration of the while-do loop, the number of options is reduced by 1. As a result, after at most *n* steps, Algorithm 2 will terminate. The total number of steps of the optimisation algorithm is therefore $\times B + n = n \times (B + 1)$. The worst-case running time of the algorithm for optimal allocation of a risk reduction budget is$O(n \times (B + 1))$ . .

The algorithm has been tested on standard data sets with known solutions. For each of the data sets the algorithm returned the correct solution.

Now consider the risk 'platform train incident' with 20 available risk reduction options (Table I), whose removed risk and cost have been given as a multiple of £10000. For different specified budgets, the optimal set of risk reduction options are according to Table III.

TABLE III
OPTIMAL SETS OF RISK REDUCTION OPTIONS FOR THE RISK OF PLATFORMTRAIN INCIDENTS (PLATFORM-ONLY)

Budget [x £10,000]	Optimal set of options	Cost of option [x £10,000]	Removed Risk [x £ 10,000]
2900	1,11,12, 14,15,16,17	2900	14870
3300	1,4,6,9,11, 12,14,15,16,17	3300	15615
3500	1,2,3,5,11, 12,14,15,16,17,18	3490	16292
4000	1,2,3,4,5,6,8,9,11, 12,14,15,16,17,18	3990	17169

For the risk 'train collision' (Table II presents a representativesample data set) from 81 available risk reduction options, whose costs and associated removed risk have been given as a multiple of £100,000. For a specified budget of £110 million, the optimal set of risk reduction options is according to Table IV.

TABLE IV
OPTIMAL SETS OF RISK REDUCTION OPTIONS FOR THE RISK OF TRAIN COLLISION ACCIDENT

Budget [x £10,000]	Optimal set of options	Cost of option [x £100,000]	Removed Risk [x £ 100,000]
1100	1-3;5,6, 16,20, 26,30, 38,40-42, 44-46, 48-50, 53, 60, 62-66, 68, 69, 73-75, 77-80	1100	7446

The largest running time of the budget allocation algorithm, on a computer with processor *Intel(R) Core(TM) 2 Duo CPU T9900 @ 3.06 GHz*, was 0.015s!

III. CONCLUSIONS

1. By using the concept '*amount of removed risk by a risk reduction option*',the problem of optimal allocation of a fixed budget, among a finite number of risk reduction options in the railways industry, can be reduced to an optimisation problem from dynamic programming.
2. For a risk reduction budget *B*and*n* risk reduction options, the running time of the optimal allocation algorithm is O(*n* x (*B*+1)) (where *B* is the size of the budget).
3. The optimal solution for 81 available risk reduction options and various fixed budgets has been achieved within a very short time, which makes the developed algorithm a very efficient decision support tool for the railway industry.

REFERENCES

[1] Elvik, R. (2001). Cost–benefit analysis of road safety measures: applicability and controversies. Accident Analysis and Prevention. vol 33 (2001) pp. 9–17.
[2] J. Li, S. Pollard, G. Kendall, E. Soane, G. Davies. "Optimising risk reduction: An expected utility approach for marginal risk reduction during regulatory decision-making". Reliability Engineering and System Safety. Elsevier Ltd, (2009).
[3] B. Flyvbjerg, M. Holm, K. Skamris, S.L. Buhl. "How common and how large are Cost Overruns in Transport Infrastructure Projects", Transport Reviews, volume 23 (1): 71-88, (2003).
[4] An M., Chen, Y., Baker, C.J.(2011). A fuzzy reasoning and fuzzy-analytical hierarchy process based approach to the process of railway risk information: A railway risk management system. Information Sciences 181 (2011) 3946–3966. Elsevier Inc.
[5] Ramanathan, R., Ganesh, L.S. (1995). Using AHP for resource allocation problems. European. Journal of Operational Research, vol. 80, 417.
[6] Saaty, T. (1988). The Analytic Hierarchy Process, McGraw-Hill, New York,
[7] Ghazinoory, S., Aliahmadi A., Namdarzangeneh, S., Ghodsypour, S.H. (2007). "Using AHP and L.P. for choosing the best alternatives based the gap analysis". Applied Mathematics and Computation vol. 184, pp. 316–321.
[8] Rashid, M., Hayes, D.F. (2011). "Needs-based sewerage prioritization: Alternative to conventional cost-benefit analysis" Md. Journal of Environmental Management, vol. 92, 2427-2440. Elsevier Ltd
[9] Cagno E., Di Giulio, A., Trucco, P. (2001). "An algorithm for the implementation of safety improvement programs". Safety Science, vol. 37, pp. 59-75. Elsevier Science Ltd.
[10] Persaud, B., Kazakov, A. (1994). "A procedure for allocating a safety improvement budget among treatment types". Accident Analysis and Prevention. Vol. 26, No. 1, pp. 121-126. Pergamon Press Ltd.
[11] Khisty, C.J., Mohammadi, J., (2001). Fundamentals of System Engineering, with Economics, Probability and Statistics. Prentice Hall, Inc., Upper Saddle River, N.J,pp. 1-57.
[12] Lindhe, A., Rose´n, L., Norberg, T., Bergstedt, O., Pettersson, T.J.R. (2011). "Cost-effectiveness analysis of risk-reduction measures to reach water safety targets". Water Research 45, pp. 241-253. Elsevier Ltd.
[13] Ozkir, V., Demirel, T. (2012). A fuzzy assessment framework to select among transportation projects in Turkey. Expert Systems with Applications, vol. 39 pp. 74–80. Elsevier Ltd.
[14] Sato, Y. (2012). "Optimal budget planning for investment in safety measures of a chemical company". International Journal of Production Economics, vol. 140, pp. 579-585. Elsevier B.V.
[15] Caputo, A.C.,Pelagagge, P.M., Palumbo, M. (2011). Economic optimization of industrial safety measures using genetic algorithms.. Journal of Loss Prevention in the Process Industries, vol. 24 pp. 541-551. Elsevier Ltd.
[16] Pirlot, M. (1996). "General local search methods". European Journal of Operational Research, vol. 92, pp. 493-511.
[17] Olson, D.L. (1988). "Opportunities and Limitations of AHP in Multi-objective Programming". Math Computing Modelling, Vol. 11, pp. 206-209
[18] Hey, J.D. (1995). "Experimental investigations of errors in decision making under risk". European Economic Review, vol. 39, pp. 633-640. Elsevier Science B.V.
[19] Van Laarhoven, P.J.M., Aarts, E.H.L., Lenstra, J.K. (1992). "Jobshop scheduling by simulated annealing". Operations Research, vol. 40, pp. 113-125.
[20] Aven, T., Kørte, J. (2003). "On the use of risk and decision analysis to support decision-making". Reliability Engineering and System Safety, vol. 79 pp. 289–299. Elsevier Science Ltd.
[21] Fukuba, Y., Ito, K. (1984). "The so-called expected utility theory is inadequate". Mathematical Social Sciences, vol. 7, pp.1-12. Elsevier Science Publishers B.V.
[22] Basso, A., Peccati, L.A. (2001). "Optimal resource allocation with minimum activation levels and fixed costs – Theory and methodology". European Journal of Operational Research, vol. 131 pp. 536-549
[23] Horowitz, E., Sahni, S. (1974). "Computing partitions with applications to the Knapsack Problem", Journal of the ACM, vol.21 pp. 277-292.
[24] Bjorndal, M.H. Caprara, A., Cowling, P.I., Croce, F.D., Lourenco, H., Malucelli, F., Orman, A.J., Pisinger, D., Rego, C., Salazar, J.J.

(1995).“Some thoughts on combinatorial optimization”. European Journal of Operational Research, vol. 83, pp. 253-270.
[25] Dasgupta, S., Papadimitriou, C., Vazirani, U. (2008). “Algorithms”. McGraw Hill, Boston, 2008.
[26] Bellman R. (1957). “Dynamic programming”. Princeton, N. J., Princeton University Press.