Scheduling trains with cross entropy optimisation

I. B. Wadhawan[1] P. J. Pudney[2] P. G. Howlett[3]
J. Piantadosi[4]

(Received 4 February 2010; revised 9 June 2010)

Abstract

The logistics of moving grain from silos to ports is constrained by the number of trains available and the capacity of loading and unloading facilities. We aim to schedule the available trains to move grain from silos to the port as quickly as possible. The sequence of trips that minimises the time required to complete all trips is a permutation of a basic sequence that has the required number of trips to each silo. We use the Cross Entropy Optimisation method to search for a permutation that minimises span. For small problems, where the optimal solution can be found by enumeration, the Cross Entropy Optimisation method achieves solutions within 5% of the optimum. It can also find good solutions for large problems.

http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/2594 gives this article, © Austral. Mathematical Soc. 2010. Published June 21, 2010. ISSN 1446-8735. (Print two pages per sheet of paper.)

Contents

1 Introduction

Trains are often used to transport grain from rural silos to ports. The logistics of moving grain from silos to ports is constrained by the number of trains available and the capacity of loading and unloading facilities. We aim to schedule the available trains to move grain from silos to the port as quickly as possible.

For our particular deterministic problem there are several silos and one port (see Figure 1). The port can unload only one train at a time; if another train arrives at the port while a train is unloading, the second train must wait in a queue for the port to become available. Similarly, each silo can load only one train at a time, and so trains may have to queue at silos. The loading time at each silo is fixed, as is the unloading time at the port.

The number of trips required to each silo is specified. When a train finishes unloading at the port, we send it to any silo requiring more trips. The

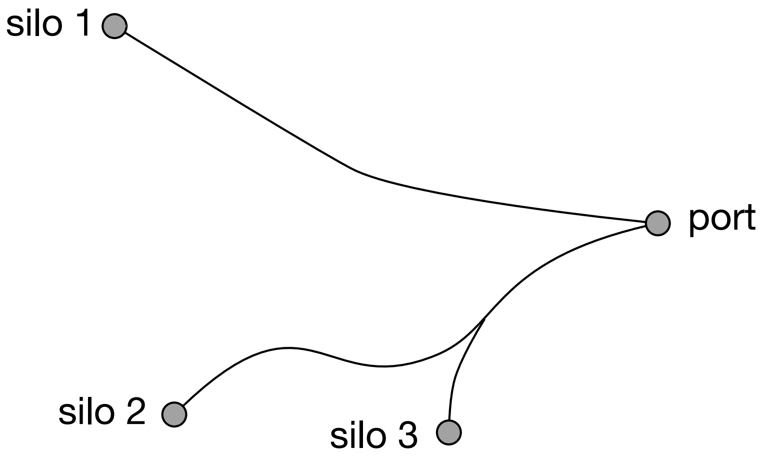


Figure 1: An example network connecting three silos and one port.

sequence of destinations is called a *trip sequence*. The number of times each silo appears in the trip sequence corresponds to the number of trips required to that silo.

The time taken to complete all trips depends on the trip sequence, because some sequences will result in longer queueing durations at silos and at the port. We aim to find the trip sequence that minimises the time required to complete all trips. For large problems there are too many possible trip sequences to evaluate, so we use Cross Entropy Optimisation to search for good solutions. The method is similar to that used to solve the Travelling Salesman Problem [2].

2 Problem formulation

We consider a problem with one port and many silos. For each silo j we know

- the time required for an empty train to drive from the port to silo j;

- the time required for a train to load at silo j;
- the time required for a full train to drive from silo j to the port.

We also know

- the time required to unload a train at the port;
- the number of trains;
- the time that each train is initially available to depart the port;
- the trip sequence.

We assume that the trains are identical, and that any train can service any silo.

Using this information, we simulate the movement of trains and calculate the time required to complete all trips. Each time a train is ready to depart the port, we send it to the next silo in the trip sequence. We use simulation rather than direct calculation because we cannot tell in advance the order in which trains will depart the port.

Figure 2 shows an example run with eight trips to silo 1 and six trips to silo 2. The horizontal axis represents time and the vertical axis represents location. There are two trains, dark and light.

The trip sequence is $[1, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 2]$. Notice that the first time the dark red train returns from silo 2, it is delayed at the port because the light blue train is unloading. There are also three occasions at silo 1 and two occasions at silo 2 where the light blue train arrives before the dark red train has finished loading, and so is delayed.

The requirement of eight trips to silo 1 and six trips to silo 2 is used to construct an initial sequence $[1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2]$. Our problem is now to find a permutation of this initial feasible sequence that minimises the time require to complete all trips.

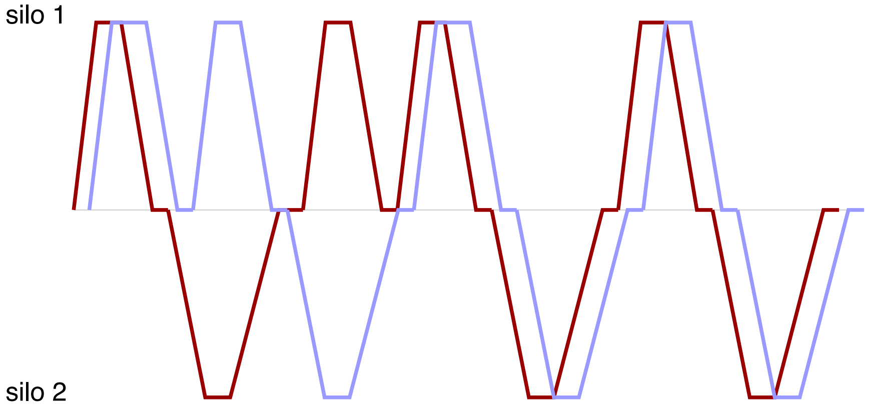


Figure 2: Train graph for trip sequence $[1, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 2]$ using two trains.

3 Cross entropy optimisation

The Cross Entropy method is a probabilistic search technique that can be used for rare event simulation and for optimisation [2, 3]. The key ideas are:

- represent the solution by parameterised probability density functions (PDF);
- generate some random candidate solutions; and
- use the best of the candidate solutions to update the PDF parameters so that the next iteration produces better candidate solutions.

The method we use to represent and solve our problem is based on the formulation of the Travelling Salesman Problem [2, Section 4.2].

We illustrate this method using a simple example with $m = 2$ silos. Suppose we require n_1 trips to silo 1, and n_2 trips to silo 2. Let $\mathcal{X}$ be the set of

Table 1: Probability generation for trips $(1,2)$.

trip sequence	probability	$S(x)$
$[1,1,1]$	$p_{11}p_{21}p_{31}$	∞
$[1,1,2]$	$p_{11}p_{21}p_{32}$	∞
$[1,2,1]$	$p_{11}p_{22}p_{31}$	∞
$[1,2,2]$	$p_{11}p_{22}p_{32}$	S_{122}
$[2,1,1]$	$p_{12}p_{21}p_{31}$	∞
$[2,2,1]$	$p_{12}p_{22}p_{31}$	S_{221}
$[2,1,2]$	$p_{12}p_{21}p_{32}$	S_{212}
$[2,2,2]$	$p_{12}p_{22}p_{32}$	∞

all possible trip sequences, and let $S(x)$ be the span associated with trip sequence $x \in \mathcal{X}$.

3.1 A relaxed problem using node placement

To make it easier to formulate our problem, we relax the constraint that each $x \in \mathcal{X}$ must have the correct number of trips to each silo, and set $S(x) = \infty$ for any infeasible trip sequence or the trip sequence which has incorrect number of trips to each silo. Let $\mathcal{X}^*$ be the set of all possible trip sequences with length $n = n_1 + n_2$.

We represent candidate solutions using an $(n \times m)$ matrix P where P_{ij} is the probability that trip i is to silo j. This representation is a variation of the *node placement method* that allows us to make more than one trip to each silo. If we require, for example, one trip to silo 1 and two trips to silo 2 then the probability of each possible trip sequence $x \in \mathcal{X}^*$ is shown in Table 1.

The trip sequences that do not have one trip to silo 1 and two trips to silo 2 have $S(x) = \infty$. The density function f used to generate candidate solutions is

$$\ln f(x;P) = \sum_{i=1}^{n}\sum_{j=1}^{m} I_{\{x_i=j\}} \ln p_{ij} \qquad (1)$$

where the indicator function

$$I_{\{x_i=j\}} = \begin{cases} 1, & \text{if trip } i \text{ is to silo } j, \\ 0, & \text{otherwise.} \end{cases}$$

For each trip sequence, we simulate the train movements to determine the corresponding span. We then select a set of *elite* sequences that have low span. These elite sequences are used to generate a new probability matrix Q that minimises the Kullback–Leibler distance D between

- $I_{\{S(X)\leqslant\gamma\}}f(x;P)$, the density of the elite trip sequences generated using probability matrix P, and
- $f(x;Q)$, the density we will use to generate the next trip sequences.

The Kullback–Leibler Cross Entropy distance between these two density functions is

$$\begin{aligned} D\left[I_{\{S(X)\leqslant\gamma\}}f(x;P), f(x;Q)\right] &= \sum I_{\{S(X)\leqslant\gamma\}}f(x;P)\ln(I_{\{S(X)\leqslant\gamma\}}f(x;P)) \\ &\quad - \sum I_{\{S(X)\leqslant\gamma\}}f(x;P)\ln f(x;Q). \end{aligned}$$

Minimising D is equivalent to choosing Q such that $\sum I_{\{S(X)\leqslant\gamma\}}f(x;P)\ln f(x;Q)$ is maximised:

$$\max_{Q} E_P I_{\{S(X)\leqslant\gamma\}} \ln f(x;Q). \qquad (2)$$

We also require the elements in each row of Q to sum to one. We implement these constraints using a Lagrange multiplier μ_i for each row. The constrained optimisation problem is

$$\max_{Q}\min_{\mu}\left[E_P I_{\{S(X)\leqslant\gamma\}}\ln f(x;Q) + \sum_{i=1}^{n}\mu_i\left(\sum_{j=1}^{m} q_{ij} - 1\right)\right]$$

which has solution

$$q_{ij} = \frac{E_P I_{\{S(X)\leqslant\gamma\}} I_{\{x_i=j\}}}{E_P I_{\{S(X)\leqslant\gamma\}}} .$$

The corresponding estimator is

$$\hat{q}_{ij} = \frac{\sum_{k=1}^{N} I_{\{S(X_k)\leqslant\gamma\}} I_{\{x_i=j\}}}{\sum_{k=1}^{N} I_{\{S(X_k)\leqslant\gamma\}}} . \tag{3}$$

This equation has a straightforward interpretation: element q_{ij} is the proportion of elite trip sequences where trip i was to silo j.

The Cross Entropy Optimisation method uses smoothed updating of the probability matrix P

$$P_{k+1} = \alpha Q_k + (1-\alpha) P_k ,$$

where P_k is the initial probability matrix at step k, Q_k is the probability matrix calculated from the elite samples at step k, and α is a smoothing parameter. The parameter α is used to avoid converging too quickly to a local optimum.

3.2 Modified node placement

The node placement method used in the previous section generates candidate solutions that may not have the desired number of trips to each silo, and then assigns these infeasible trips sequences an infinite span. In practice, we speed up the method by not generating infeasible trips. As before, P_{ij} is the probability that trip i is to silo j. Algorithm 1 gives the modified method.

Table 2 shows the probability of generating various trip sequences using the unconstrained method and the modified method. The number of trips required is one trip to silo 1 and two trips to silo 2. We have $P_{ij} = 1/3$.

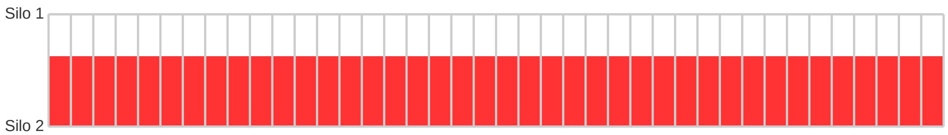
Consider the first four rows of the Table 2. The unconstrained method can generate any of these trip sequences, with the probabilities shown in the

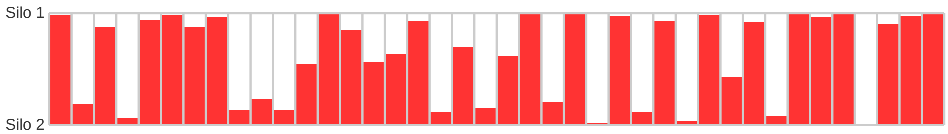
Algorithm 1 modified cross entropy method.

```
1. Use the first row of $P$ to select the destination for the first trip using roulette-wheel selection. Suppose the value is $j$.
2. If all the required trips to silo $j$ have been done, set $P_{ij} = 0$ for all remaining rows $i$, and renormalise each row.
3. Repeat the process with the remaining rows of $P$ to select the destinations for the remaining trips.
```

Table 2: Probability generation using unconstrained method and modified method.

trip sequences	unconstrained probability	modified probability
$[1,1,1]$	$\frac{1}{3}\frac{1}{3}\frac{1}{3} = \frac{1}{27}$	-
$[1,1,2]$	$\frac{1}{3}\frac{1}{3}\frac{2}{3} = \frac{2}{27}$	-
$[1,2,1]$	$\frac{1}{3}\frac{2}{3}\frac{1}{3} = \frac{2}{27}$	-
$[1,2,2]$	$\frac{1}{3}\frac{2}{3}\frac{2}{3} = \frac{4}{27}$	$\frac{1}{3}11 = \frac{9}{27}$
$[2,1,1]$	$\frac{2}{3}\frac{1}{3}\frac{1}{3} = \frac{2}{27}$	-
$[2,1,2]$	$\frac{2}{3}\frac{1}{3}\frac{2}{3} = \frac{4}{27}$	$\frac{2}{3}\frac{1}{3}1 = \frac{6}{27}$
$[2,2,1]$	$\frac{2}{3}\frac{2}{3}\frac{1}{3} = \frac{4}{27}$	$\frac{2}{3}\frac{2}{3}1 = \frac{12}{27}$
$[2,2,2]$	$\frac{2}{3}\frac{2}{3}\frac{2}{3} = \frac{8}{27}$	-

(a)


(b)


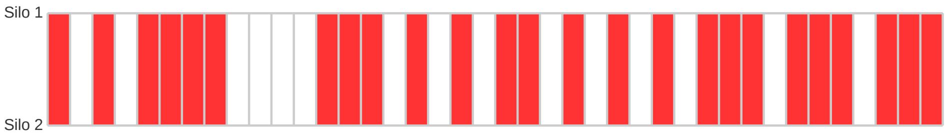
(c)


Figure 3: Convergence at iteration 0, 10 and 20.

table. With the modified method, once we have generated the first trip to silo 1, the two remaining trips must be to silo 2.

The graphs in Figure 3 illustrate the convergence of the method at iterations 0, 10 and 20. In this example we require twenty five trips to silo 1 and fifteen trips to silo 2, with four trains. Each vertical red bar corresponds to a trip i, and indicates $P_{i,2}$, the probability that trip i is to silo 2.

The train graph in Figure 4 shows the train movements.

4 Cross entropy optimisation versus enumeration

For sufficiently small problems, the optimal solution can be found by calculating time span for every possible trip sequence. The problems shown in Table 3 are calculated using the following problem data:

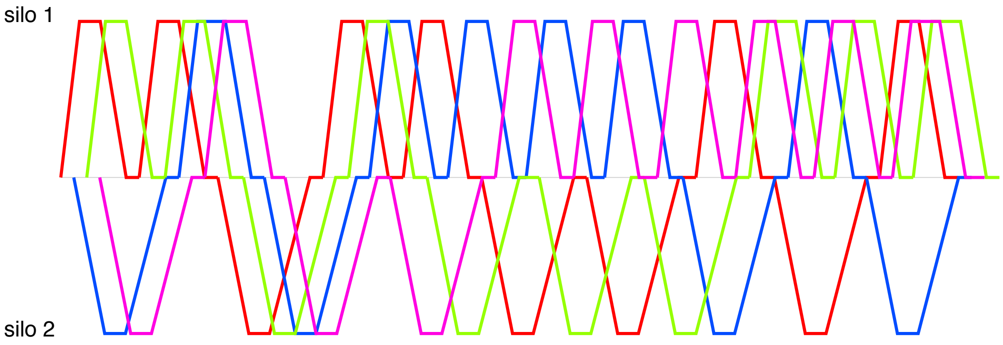


Figure 4: Train graph for $(25, 15)$ trips using four trains.

- time taken to travel from the port to silo 1 is 5.74;
- time taken to travel from silo 1 to the port is 7.97;
- time taken to travel from from the port to silo 2 is 9.52;
- time taken to travel from silo 2 to the port is 12.46;
- loading duration at silo 1 is 6.4;
- loading duration at silo 2 is 6.4;
- unloading duration at the port is 4.

Table 3 shows some results.

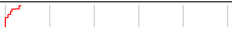
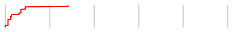
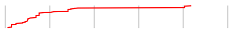
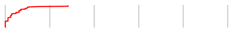
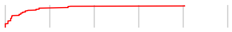
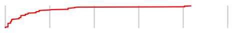
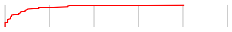
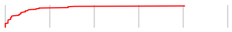
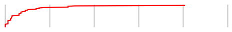
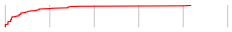
For small problems, complete enumeration is faster than the Cross Entropy Optimisation method. However, the Cross Entropy Optimisation method is able to find good solutions for problems that are too large for complete enumeration.

The convergence of the Cross Entropy Optimisation method, and the solution converged to, depend in part on the selection of three parameters: the number of samples N generated at each iteration, the proportion ρ of the

Table 3: Cross Entropy Optimisation Versus Enumeration results. Columns (left to right): the required number of trips to each silo; the number of possible trip sequences; the minimum span found by complete enumeration; the CPU time required by complete enumeration; the minimum span found by the Cross Entropy Optimisation method and the CPU time required by the Cross Entropy Optimisation method.

		Enumeration		Cross Entropy Optimisation	
	Trip	Minimum	CPU Time	Minimum	CPU Time
(n_1, n_2)	sequences	span	(seconds)	span	(seconds)
(10, 10)	184756	6.3927	24	6.3927	298
(8, 13)	203490	6.9927	37	6.9927	253
(11, 10)	352716	6.7219	56	6.7219	385
(7, 18)	480700	8.4319	104	8.4319	330
(11, 11)	705432	7.0953	183	7.0953	269
(12, 12)	2704156	*7.5652*	2697	*7.5693*	255
(50, 50)	10^{29}	—	—	30.0708	1110

Table 4: Cumulative distribution for three parameters for six examples.

N	α	ρ	distribution
500	0.45, 0.55, 0.65	0.005, 0.01, 0.03, 0.05	
1000	0.45, 0.55, 0.65	0.005, 0.01, 0.03, 0.05	
1500	0.45, 0.55, 0.65	0.005, 0.01, 0.03, 0.05	
500, 1000, 1500	0.45	0.005, 0.01, 0.03, 0.05	
500, 1000, 1500	0.55	0.005, 0.01, 0.03, 0.05	
500, 1000, 1500	0.55	0.005, 0.01, 0.03, 0.05	
500, 1000, 1500	0.45, 0.55, 0.65	0.005	
500, 1000, 1500	0.45, 0.55, 0.65	0.01	
500, 1000, 1500	0.45, 0.55, 0.65	0.03	
500, 1000, 1500	0.45, 0.55, 0.65	0.05	

samples in the elite set, and the smoothing parameter α. Table 4 shows results of experiments where we used Cross Entropy Optimisation to find solutions to six different problems using every combination of parameters $N \in \{500, 1000, 1500\}$, $\alpha \in \{0.45, 0.55, 0.65\}$ and $\rho \in \{0.005, 0.01, 0.03, 0.05\}$.

Each row of the Table 4 shows the cumulative distribution of results when one of these parameters was held constant. The horizontal axis in each graph represents the value of the solution relative to the optimal solution (found by enumeration). The vertical lines correspond to relative values of 1.00, 1.01, ..., 1.05. For each set of parameters, all best solutions found were within 5% of the optimal solution. In general, the ideal parameter values for a given problem have to be determined experimentally.

5 Conclusions and future work

Transportation problems that seek an optimal trip sequence can be difficult to solve, particularly when there are many possible sequences. The Cross

Entropy Optimisation method is easy to implement and is effective at finding good solutions to our problem.

We have started investigating the use of the Cross Entropy Optimisation method to solve multi-objective versions of our problem. Multi-objective versions of the method are discussed by Unveren et al. [1]. However, it is not straightforward because many different sequence map onto the same objective function value. For example, a sample problem with ten trips to each of two silos gave 184756 trip sequences, 28703 unique objective values, but only 23 Pareto sequences and 18 Pareto points.

References

[1] Unveren A. and A. Acan. Multi-objective optimisation with cross entropy method: Stochastic learning with clustered Pareto fronts. *IEEE Congress on Evolutionary Computations*, pages 3065–3071, 2007. C345

[2] Boer P. T., Kroese D. P., Mannor S., and Rubinstein R. Y. *A tutorial on the cross-entropy method.* http://www.springerlink.com/index/KPW596202975755N.pdf. C334, C336

[3] Rubinstein R. Y. and D. P. Kroese. *The cross entropy method: A unified approach to combinatorial optimization, monte-carlo simulation, and machine learning.* Springer–Verlag, New York, 2004. C336

Author addresses

1. **I. B. Wadhawan**, School of Mathematics and Statistics, University of South Australia, Australia.
mailto:wadib001@postgrads.unisa.edu.au

2. **P. J. Pudney**, School of Mathematics and Statistics, University of South Australia, Australia.
mailto:peter.pudney@unisa.edu.au

3. **P. G. Howlett**, School of Mathematics and Statistics, University of South Australia, Australia.
mailto:phil.howlett@unisa.edu.au

4. **J. Piantadosi**, School of Mathematics and Statistics, University of South Australia, Australia.
mailto:julia.piantadosi@unisa.edu.au