



Discrete Optimization

The Steiner travelling salesman problem with correlated costs

Adam N. Letchford^{a,*}, Saeideh D. Nasiri^b^a Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom^b Department of Mathematics and Statistics, Lancaster University, Lancaster, LA1 4YX, United Kingdom

ARTICLE INFO

Article history:

Received 20 May 2014

Accepted 22 February 2015

Available online 28 February 2015

Keywords:

Travelling salesman problem
 Mean-variance optimisation
 Mixed-integer nonlinear programming

ABSTRACT

The Steiner Travelling Salesman Problem (STSP) is a variant of the TSP that is suitable for instances defined on road networks. We consider an extension of the STSP in which the road traversal costs are both stochastic and correlated. This happens, for example, when vehicles are prone to delays due to rush hours, road works or accidents. Following the work of Markowitz on portfolio selection, we model this problem as a bi-objective mean-variance problem. Then, we show how to approximate the efficient frontier via integer programming. We also show how to exploit certain special structures in the correlation matrices. Computational results indicate that our approach is viable for instances with up to 100 nodes. It turns out that minimum variance tours can look rather different from minimum expected cost tours.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In the *Travelling Salesman Problem* (TSP), one is given a complete graph, and a cost for each edge of the graph, and one seeks a Hamiltonian circuit of minimum total cost. The TSP is a classical and much-studied problem in combinatorial optimisation, and several books have been written on it (e.g., Applegate, Bixby, Chvátal, & Cook, 2006; Gutin & Punnen, 2002; Lawler, Lenstra, Kan, & Shmoys, 1985). Somewhat less well known is the *Steiner TSP* (STSP), which is a variant of the TSP that is suitable for instances defined on road networks (see, e.g., Cornuéjols, Fonlupt, & Naddef, 1985; Fleischmann, 1985; Letchford, Nasiri, & Theis, 2013; Orloff, 1974). In the STSP, one is given a sparse graph, representing a road network, a cost for each edge of the graph, and a subset of nodes that represent customers. The task is to find a minimum-cost tour that passes through each customer node. Edges may be traversed more than once, and nodes visited more than once, if desired.

In real-life STSP applications, the edge-costs are likely to be stochastic rather than deterministic. (They are also likely to be time-varying rather than static, but we ignore that possibility in this paper.) For example, if the costs represent fuel consumption or travel times, then events such as rush hours, road works or accidents can cause the costs to be larger than expected. In addition, the costs are likely to be correlated, rather than statistically independent, since congestion on one road is likely to cause congestion on nearby roads.

These considerations lead us to consider an extension of the STSP, which we call the *Steiner TSP with Correlated Costs* (STSPCC). This is like the STSP, except that the costs follow a multi-variate distribution whose first and second moments are known. Following the classical Markowitz mean-variance approach to portfolio selection Markowitz (1952), we view the STSPCC as a *bi-objective* optimisation problem, in which there is a trade-off between minimising the expected cost of a tour and minimising the variance of the cost. This leads naturally to the problem of computing (or approximating) the efficient frontier.

Our approach to this problem is based on integer programming. We begin by taking the two best formulations of the standard STSP given in our earlier paper (Letchford et al., 2013), and adapting them to the problem of finding a single efficient solution to the STSPCC. This leads to four formulations, two quadratic and two linear. Next, we show how those formulations can be improved if the cost correlations have either of two special structures that may arise in practice. Finally, we take the best of the four formulations and solve it repeatedly, with different settings of a certain parameter, in order to construct an approximation of the efficient frontier. Our computational results show that, in this way, one can solve small- to medium-sized instances in reasonable times.

The paper is structured as follows. The relevant literature is reviewed in Section 2. In Section 3, we present our approach to the case in which the correlation matrix has no special structure. In Section 4, we show how to deal with the two special correlation structures mentioned. The computational results are presented in Section 5, and some concluding remarks appear in Section 6.

We use the following notation in the remainder of the paper. The road network is denoted by $G = (V, E)$, where V is the vertex set and E is the edge set. The cost of traversing the edge (i.e., road) $e \in E$ is

* Corresponding author. Tel.: +44 1524 594719.

E-mail addresses: A.N.Letchford@lancaster.ac.uk (A. N. Letchford), s.d.nasiri@lancaster.ac.uk (S. D. Nasiri).

denoted by c_e . The set of customer nodes, which is a subset of V , is denoted by C . It is assumed that G is undirected, i.e., that all edges $e \in E$ represent two-way streets. We do however make some remarks about the possibility of adapting our approach to the ‘asymmetric’ case, in which some or all of the roads are one-way (see Sections 2.1 and 3.1).

2. Literature review

We now review the relevant literature. We cover formulations of the STSP in Section 2.1, mean-variance and multi-objective optimisation in Section 2.2, and vehicle routing with correlated costs in Section 2.3.

2.1. Formulations of the STSP

In Cornuéjols et al. (1985) and Fleischmann (1985), it is shown how to formulate the STSP as an integer program with $|E|$ variables and an exponential number of constraints. In our own paper (Letchford et al., 2013), we presented some alternative formulations of polynomial size. Of those, the two which performed best computationally were the *single-commodity flow* (SCF) and *multi-commodity flow* (MCF) formulations, obtained by adapting formulations of the standard TSP due to Gavish and Graves (1978) and Claus (1984), respectively.

Before presenting the SCF and MCF formulations in detail, we need some additional notation. Let $\tilde{G} = (V, A)$ be a directed graph, where the set of directed arcs A is obtained from the edge set E by replacing each edge $\{i, j\}$ with two directed arcs (i, j) and (j, i) . For each arc $a \in A$, the cost c_a is set to the cost of the corresponding edge. For any $S \subset V$, let $\delta^+(S)$ denote the set of arcs in A whose tail is in S and whose head is in $V \setminus S$, and let $\delta^-(S)$ denote the set of arcs in A for which the reverse holds. We also assume, without loss of generality, that vertex 1 is a customer.

In the SCF formulation, we imagine that the salesman departs from vertex 1 with $|C| - 1$ units of a commodity, and delivers one unit of that commodity to each vertex in $C \setminus \{1\}$. There are two variables for each arc $a \in A$. The binary variable x_a takes the value 1 if and only if arc a is traversed, and the continuous variable f_a represents the amount of the commodity passing through arc a . The formulation is then:

$$\min \sum_{a \in A} c_a x_a \tag{1}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} x_a \geq 1 \quad (\forall i \in C) \tag{2}$$

$$\sum_{a \in \delta^+(i)} x_a = \sum_{a \in \delta^-(i)} x_a \quad (\forall i \in V \setminus \{1\}) \tag{3}$$

$$\sum_{a \in \delta^-(i)} f_a - \sum_{a \in \delta^+(i)} f_a = 1 \quad (\forall i \in C \setminus \{1\}) \tag{4}$$

$$\sum_{a \in \delta^-(i)} f_a - \sum_{a \in \delta^+(i)} f_a = 0 \quad (\forall i \in V \setminus C) \tag{5}$$

$$0 \leq f_a \leq (|C| - 1)x_a \quad (\forall a \in A) \tag{6}$$

$$x_a \in \{0, 1\} \quad (\forall a \in A). \tag{7}$$

Note that the SCF formulation is a mixed 0–1 linear program (mixed 0–1 LP) with $\mathcal{O}(|E|)$ variables and constraints.

In the MCF formulation, we imagine that the salesman departs from vertex 1 with a distinct commodity for each vertex in $C \setminus \{1\}$. Accordingly, for all $a \in A$ and all $k \in C \setminus \{1\}$, we define the continuous variable f_a^k , representing the proportion of commodity k passing through arc a . (One could declare f_a^k to be binary, but there is no

need to.) We then minimise (1) subject to (3), (7) and the following constraints:

$$\sum_{a \in \delta^-(i)} f_a^k - \sum_{a \in \delta^+(i)} f_a^k = 0 \quad (\forall i \in V \setminus \{1\}; k \in C \setminus \{1, i\}) \tag{8}$$

$$\sum_{a \in \delta^-(k)} f_a^k - \sum_{a \in \delta^+(k)} f_a^k = 1 \quad (\forall k \in C \setminus \{1\}) \tag{9}$$

$$0 \leq f_a^k \leq x_a \quad (\forall a \in A, k \in C \setminus \{1\}). \tag{10}$$

Note that the MCF formulation is a mixed 0–1 LP with $\mathcal{O}(|C||E|)$ variables and constraints.

We remark that the x variables are binary in both formulations, because there always exists an optimal STSP solution such that no arc is traversed more than once. This is not true in the case of asymmetric costs, so if one wished to adapt the formulations to the asymmetric STSP, one would have to change the x variables from binary to general-integer.

2.2. Mean-variance and multi-objective optimisation

Mean-variance optimisation was first presented by Markowitz (1952), in the context of portfolio selection. In his model, there are n stocks, and we are given a vector $r \in \mathbb{R}^n$ of expected returns and a covariance matrix $Q \in \mathbb{R}^{n \times n}$. For $i = 1, \dots, n$, the continuous variable x_i represents the proportion of money invested in stock i . Then, for a given solution x , the mean (expected) return is $r^T x$, and the variance of return is $x^T Q x$.

It is desirable to have a high mean, but a low variance. Hence, the mean-variance problem is a *bi-objective* problem. Markowitz calls a solution x *efficient* if all solutions with higher mean than x have higher variance, and all solutions with smaller variance have smaller mean. He then defines the *efficient frontier* as the curve in \mathbb{R}_+^2 described by all mean-variance pairs that are achieved by at least one efficient solution.

Computing the efficient frontier is rather demanding. On the other hand, computing a single efficient solution is relatively easy. One simple way to do it is to solve the following problem

$$\min \left\{ \alpha(x^T Q x) - (1 - \alpha)(r^T x) : \sum_{i=1}^n x_i = 1, x \geq 0 \right\}$$

for some $0 < \alpha < 1$. This is a convex quadratic program (CQP), since the covariance matrix Q is positive semidefinite. It can be solved, e.g., with simplex methods (Beale, 1959; Wolfe, 1959). Another way to compute an efficient solution is to solve the CQP

$$\min \left\{ x^T Q x : \sum_{i=1}^n x_i = 1, r^T x \geq \epsilon, x \geq 0 \right\}$$

for some $\epsilon > 0$. These two approaches can be viewed as applications of the *weighting* and ϵ -*constraint* methods for multi-objective optimisation (Miettinen, 1999).

In this paper, we wish to apply the mean-variance approach to a combinatorial optimisation problem. Although there does not seem to be a literature on this specific topic, there is a considerable literature on multi-objective combinatorial optimisation problems with *linear* objectives (e.g., Ehrgott & Gandibleux, 2007; Mavrotas & Diakoulaki, 1998; Sourd & Spanjaard, 2008; Ulungu & Teghem, 1995). Even for that apparently simpler case, computing the entire efficient set is extremely challenging. Moreover, two complications arise that are not present in the continuous case: (i) there may exist efficient solutions that cannot be found by the weighting method, and (ii) the ϵ -constraint method may return solutions that are not efficient. Nevertheless, each efficient solution can be obtained with the ϵ -constraint method for a suitable value of ϵ (see, e.g., Ehrgott & Gandibleux, 2007; Ulungu & Teghem, 1995).

2.3. Vehicle routing with correlated costs

For an overview of stochastic vehicle routing, see Gendreau, Laporte, and Seguin (1996). Here, we concentrate on problems with stochastic travel times and/or costs.

Three papers are concerned with TSPs with stochastic travel times. Kao (1978) proposed a dynamic programming heuristic for the variant in which the objective is to maximise the probability of finishing the route by a deadline. Jula, Dessouky, and Ioannu (2006) applied dynamic programming to a different variant, in which the goal is to minimise expected cost, subject to a lower bound on the service level. Teng, Ong, and Huang (2004) used 2-stage stochastic integer programming to solve a third variant, in which a recourse action is available.

Four other papers address problems with more than one vehicle. Laporte, Louveaux, and Mercure (1992) applied branch-and-cut to a variant in which vehicles incur a penalty for tardiness. Lambert, Laporte, and Louveaux (1993) presented a savings-based heuristic for a variant in which a subset of arcs are prone to congestion. Kenyon and Morton (2003) applied branch-and-cut to a third variant, in which the objective is to minimise the expected total completion time. Li, Tian, and Leung (2010) described a tabu-search heuristic for some variants with time windows.

Another variant is the *dynamic* TSP with stochastic costs/times, in which the vehicle can be re-routed in real-time. Secomandi (2003) and Toriello, Haskell, and Poremba (2014) propose “rollout” and approximate dynamic programming heuristics, respectively, for this variant.

A different though related stream of literature is concerned with modelling spatial and/or temporal correlations between travel times on real-life road networks. Fan, Kalaba, and Moore (2005), Samaranyake, Blandin, and Bayen (2012) and Dong, Li, Vo, and Vu (2012) argue that, in rural areas, spatial correlations satisfies a certain Markovian property, so that only the conditional probability distributions of travel times on adjacent pairs of links need to be modelled. Chen, Lam, Sumalee, and Li (2012) examine real travel time data from an urban area in Hong Kong with low traffic flow, and find that spatial correlations decrease from 0.29 to 0.04 as one moves from adjacent links to links that are up to 4 links apart. They conclude that it is enough to model correlations only for pairs of roads that are spatially very close. On the other hand, Nasiri (2014) analysed both temporal and spatial correlations in real travel time data from a highly congested road network in the City of London, and found that spatial correlations between roads that are up to 20 links apart are still significant. The ways in which we handle these different correlation structures are explained in Section 4.

3. An approach to the general case

In this section and the next, we present our approach. This section deals with the general case, in which the cost correlation matrix has no special structure. In the next section, special correlation structures are considered.

From now on, we use the following notation. For all $e \in E$, we let \bar{c}_e denote the expected cost of edge e . For all pairs $e, f \in E$, we let Q_{ef} denote the covariances between the costs on edges e and f . When considering \bar{G} , the directed graph derived from G (see Section 2.1), we let \bar{c}_a denote the expected cost of the edge in E that corresponds to the arc a , and let Q_{ab} denote the covariance between the costs of the edges in E that corresponds to the arcs $a, b \in A$.

3.1. Finding individual solutions

In order to find a single (hopefully efficient) solution, one can use the ϵ -constraint method mentioned in Section 2.2. To do this, one can take either of the two formulations of the STSP mentioned in

Section 2.1 (i.e., the SCF and MCF formulations), change the objective function to

$$\sum_{a \in A} \sum_{b \in A} Q_{ab} x_a x_b,$$

and add the constraint

$$\sum_{a \in A} \bar{c}_a x_a \leq \epsilon. \quad (11)$$

Note that the resulting formulations, having a quadratic objective function, are mixed 0–1 CQPs. We will call them “SCF- ϵ ” and “MCF- ϵ ”.

It is possible nowadays to solve mixed 0–1 CQPs directly, with commercial software packages. Nevertheless, software for mixed 0–1 LPs is much more sophisticated. For this reason, we thought it desirable also to derive and test mixed 0–1 LP formulations. Fortunately, using a standard trick (Fortet, 1959), one can linearise either of the formulations, as follows:

- For all pairs $a, b \in A$, create a new continuous variable, say y_{ab} , representing the product $x_a x_b$. (One could declare y_{ab} to be binary, but there is no need to.)
- Replace each quadratic objective term, say $Q_{ab} x_a x_b$, with the linear term $Q_{ab} y_{ab}$.
- For all pairs $a, b \in A$, add the constraints $y_{ab} \geq 0, y_{ab} \leq x_a, y_{ab} \leq x_b$ and $y_{ab} \geq x_a + x_b - 1$ to the formulation.

The resulting two mixed 0–1 LP formulations are valid, but tend to have a weak continuous relaxation. In order to strengthen them, we propose to add the following valid linear constraints:

$$\sum_{b \in \delta^+(i)} y_{ab} = \sum_{b \in \delta^-(i)} y_{ab} \quad (i \in V \setminus \{1\}, a \in A) \quad (12)$$

$$\sum_{b \in A} \bar{c}_b y_{ab} \leq \epsilon x_a \quad (a \in A) \quad (13)$$

$$\sum_{b \in A} \bar{c}_b (x_b - y_{ab}) \leq \epsilon (1 - x_a) \quad (a \in A). \quad (14)$$

The equations (12) are derived by multiplying the equations (3) by individual x variables, and then linearising them by replacing products of x variables with the corresponding y variables. The inequalities (13) and (14) are obtained by multiplying the inequality (11) by each x variable and its complement in turn, and then linearising as before.

The idea of multiplying linear constraints by binary variables to derive new constraints was first proposed in Adams and Sherali (1986), and forms the basis of the well-known *Reformulation-Linearisation Technique* (RLT) for global optimisation (Sherali & Adams, 1998). Accordingly, we call our mixed 0–1 LP formulations “RLT-SCF- ϵ ” and “RLT-MCF- ϵ ”. We remark, however, that a full application of the RLT to the mixed 0–1 CQP formulations would lead to mixed 0–1 LP formulations with even more variables and constraints. Specifically, there would be additional variables representing products of x and f variables, and additional constraints obtained by multiplying the flow conservation constraints by x variables. There is no need to do that here, since neither SCF- ϵ nor MCF- ϵ has quadratic terms involving f variables.

Observe that each of the four formulations presented in this section has $\mathcal{O}(|E|)$ binary variables, $\mathcal{O}(|E|^2)$ continuous variables and $\mathcal{O}(|E|^2)$ constraints. Provided the graph G is sparse, as is usually the case in practice, this is likely to be manageable.

We remark that the version of the RLT used in this section cannot be applied to the asymmetric version of the STSPCC, since it relies on the assumption that the x variables are binary.

3.2. Approximating the efficient frontier

Now we turn our attention to the problem of computing the efficient frontier. Since the number of efficient solutions to the STSPCC is potentially exponentially large, and computing even one efficient solution is clearly \mathcal{NP} -hard in the strong sense, it is clear that computing the entire frontier exactly is out of the question for instances of realistic size. We therefore content ourselves with a crude approximation of the frontier, obtained with the following algorithm:

1. Let $k \geq 2$ be an integer parameter.
2. Solve a standard STSP, in which the objective is to minimise the total expected cost. Record the resulting tour and its expected cost and variance. Let the expected cost be ℓ .
3. Solve another modified STSP instance, in which the objective is to minimise the variance of the cost. Record the resulting tour and its expected cost and variance. Let the expected cost be u .
4. For $i = 1, \dots, k - 1$, solve one of the four formulations described in the previous section with ϵ set to $\ell + i(u - \ell)/k$. Record the resulting tours and their expected cost and variance.
5. Check the collection of $k + 1$ tours generated and discard any that are dominated by other tours in the collection.

We will show in Section 5 that this method can be used successfully for instances with up to around 50 nodes. (For larger instances, one could perhaps interrupt the branch-and-bound solver whenever a certain time limit is exceeded, and record the lower and upper bounds obtained at that point in time. In this way, one could construct a lower and upper approximation to the frontier.)

4. Exploiting special correlation structures

An instance of the STSPCC is given by a graph $G = (V, E)$, an expected cost vector $c \in \mathbb{R}^{|E|}$ and a covariance matrix $Q \in \mathbb{R}^{|E| \times |E|}$. Up to now, we have assumed that Q is simply given. In real-life applications, however, Q will be unknown initially, and will have to be estimated, based on empirical measurements of road costs. Realistically speaking, it is unlikely that enough measurements will have been made to estimate all of the entries of Q with any accuracy. In such a situation, it may be acceptable (or even essential) to work with a rough approximation of Q .

In our opinion, there are two possible approximations of Q that are likely to be acceptable in practice. The first is a low-rank approximation, based on principal components analysis. The other is a sparse approximation, in which it is assumed that the covariance will be zero whenever two roads are far from each other. In the following two sections, we explain these two approximation strategies in detail, and show how they can be used to derive improved formulations of the STSPCC.

4.1. Covariance matrices with low rank

We now recall some well-known facts from linear algebra (see, e.g., Strang, 2009, chap. 6). Given a covariance matrix $Q \in \mathbb{R}^{n \times n}$, let $\lambda_1, \dots, \lambda_n \in \mathbb{R}_+$ be its eigenvalues, sorted in non-increasing order of value, and let $v^1, \dots, v^n \in \mathbb{R}^n$ be the corresponding (normalised) eigenvectors. Then Q can be decomposed as:

$$\sum_{i=1}^n \lambda_i v^i (v^i)^T. \tag{15}$$

The rank of Q is the least integer k such that $\lambda_i > 0$ for $i \leq k$, but $\lambda_i = 0$ for $i > k$. If we let $M \in \mathbb{R}^{k \times n}$ be the matrix whose i th row is $\sqrt{\lambda_i} v^i$, we have $Q = M^T M$.

In our context, Q is a matrix of order $|E|$, where E is the set of roads in a road network. Given raw data on the road costs, there

are basically two ways to compute a low-rank approximation of Q . One simple way is to perform a principal components analysis (see, e.g., Anderson, 2011). This means computing the sample covariance matrix, computing the decomposition (15), and then selecting a value k such that the sum of the first k eigenvalues is large compared to the sum of the remaining $|E| - k$ eigenvalues. Then, the desired low-rank approximation can be obtained by summing only the first k terms in (15). Another way, popular in the finance literature (e.g., Chamberlain & Rothschild, 1983), is to assume *a priori* that Q has rank k , for some small integer k , and then estimate the matrix M directly.

In any case, suppose that we are given an instance of the STSPCC such that Q has rank k , where k is a small integer. We can improve the mixed 0-1 CQP formulations presented in Section 3.1 in the following way. Let $M \in \mathbb{R}^{k \times |E|}$ be such that $Q = M^T M$. Add k new continuous variables z_1, \dots, z_k to the formulation, along with the following k linear equations:

$$z_r = \sum_{e=\{i,j\} \in E} M_{re}(x_{ij} + x_{ji}) \quad (r = 1, \dots, k).$$

Then, in the objective function, replace the expression

$$\sum_{a \in A} \sum_{b \in A} Q_{ab} x_a x_b$$

with the simpler expression $\sum_{r=1}^k z_r^2$.

The above modification of the mixed 0-1 CQP formulations does not affect the lower bound obtained by solving the continuous relaxation, but it dramatically decreases the number of non-zero terms in the objective function, at the cost of introducing only k continuous variables and k linear constraints. If k is small compared to n , this may reduce the amount of work done at each node of the branch-and-bound tree.

We remark that it is not possible to modify the mixed 0-1 LP formulations presented in Section 3.1 in an analogous way. This is because the y variables are continuous, rather than binary.

4.2. Covariances only between nearby links

Another way to simplify the covariance matrix Q is to assume that the covariances are significant only for pairs of roads that are ‘close’ in the road network. To be more precise, let us define the distance between edges e and f to be the least integer p such that there exists a path in G , from one of the end-vertices of e to one of the end-vertices of f , that uses p intermediate edges. Then, the assumption under consideration in this section is that there exists a small non-negative integer k such that, for all pairs $e, f \in E$, the covariance Q_{ef} is close to zero whenever the distance between e and f exceeds k .

As mentioned in Section 2.3, there is empirical evidence that the above-mentioned property holds for small values of k in road networks that have only mild congestion. Moreover, there now exist efficient computational methods for estimating sparse covariance matrices with specified sparsity patterns (see, e.g., Chaudhuri, Drton, & Richardson, 2007). Such methods can be used to compute good approximations of Q for any specified value of k .

So, suppose that we are given an instance of the STSPCC such that Q has a zero entry whenever the distance between the corresponding pair of edges is larger than k , for some small non-negative integer k . If we consider the mixed 0-1 CQP formulations presented in Section 3.1, we see that the only benefit is that the objective function is sparse. On the other hand, if we consider the mixed 0-1 LP formulations, a much more significant benefit arises: for all pairs $a, b \in A$ such that the distance between the corresponding edges $e, f \in E$ exceeds k , we can delete the variable y_{ab} from the formulation, along with all constraints in which it appears. If k is small, this will result in a dramatic decrease in the number of variables and constraints.

5. Computational experiments

In this section, we report on some computational experiments. We explain how we created our test instances in Section 5.1, compare the integer programming formulations in Section 5.2, and report on approximate frontier construction in Section 5.3.

5.1. Construction of test instances

We began by taking eight STSP instances that were described in our earlier paper (Letchford et al., 2013). These were constructed as follows. For $V \in \{25, 50, 75, 100\}$, a random planar graph was created that was designed to resemble a real road network. For each of these four graphs, a random node was selected to be the depot, and the cost c_e of traversing an edge was then set to be proportional to the Euclidean distance between the end-nodes. Then, for each of the four graphs, two STSP instances were created. In one, each non-depot node was selected to be a customer with probability $1/3$. In the other, the probability was $2/3$. Table 1 shows the number of edges and customers in each of the eight sparse graphs.

Next, we converted each of the eight STSP instances into STSPCC instances with no special covariance structure. The expected cost vector \bar{c} was just set to c . The covariance matrix Q was created as follows:

- Generate $|E| \times |E|$ non-negative random numbers from the standard normal distribution.
- Scale each of the $|E|$ vector coordinates to have length one. This results in having $|E|$ uniform random vectors in the non-negative orthant of the unit sphere in $|E|$ dimensions (Marsaglia, 1972; Muller, 1959).
- Multiply each of the $|E|$ vectors by a uniform random number in range $[0, 10]$ representing the standard deviation of the cost of that edge and round the components of every vector to the nearest integer.
- Let $M \in \mathbb{Z}_+^{|E| \times |E|}$ be the matrix whose columns are the $|E|$ dimensional vectors, in any order.
- Set Q to $M^T M$.

By construction, the entries in Q are in the range $[0, 100]$ for all eight instances. We chose to create covariance matrices with non-negative entries since negative covariances seem unlikely in real life (see, e.g. Nasiri, 2014).

Next, in order to explore the potential of the formulations that we gave in Section 4.1, we created another eight STSPCC instances, with a *low-rank* covariance structure. In fact, we considered the situation that ought to be most favourable to those formulations: the case in which the rank of Q is only one. To create Q in this case, we just set Q to vv^T , where $v \in \mathbb{Z}$ is a vector in which each entry is a random integer between 0 and 10. The entries in Q are then again between 0 and 100.

Finally, in order to explore the potential of the formulations that we gave in Section 4.2, we created another eight STSPCC instances in

Table 1
Number of nodes, edges and customers for the 8 instances.

Instance	$ V $	$ E $	$ C $
25-1/3	25	31	8
25-2/3			16
50-1/3	50	69	16
50-2/3			32
75-1/3	75	105	25
75-2/3			50
100-1/3	100	139	33
100-2/3			66

which covariances are non-zero only for links that are close in the road network. Again, we considered the situation that ought to be most favourable to those formulations: the case in which the covariance between two edges is non-zero only if those edges are adjacent in G . To create Q in this case, we set Q to MM^T , where $M \in \mathbb{Z}_+^{|E| \times |V|}$ is created as follows. For a given $e \in E$ and a given $i \in |V|$, we set M_{ei} to:

- A random integer between 1 and 10 if edge e has node i as an end-node;
- zero otherwise.

For this case, Q will be a sparse matrix with entries between 0 and 200.

5.2. Comparing the integer programming formulations

A program was written in Microsoft Visual C that reads an instance from a file, and then outputs four integer programs, corresponding to the formulations SCF- ϵ , MCF- ϵ , RLT-SCF- ϵ and RLT-MCF- ϵ . Each of the resulting $96 (4 \times 3 \times 8)$ integer programs was fed into the branch-and-bound solver of IBM CPLEX version 12.3. The computer used was a PC with a 1.6 gigahertz Intel Core i7 processor, with 8 gigabytes of RAM, operating under Windows 7. Default CPLEX settings were used. The parameter ϵ was set to half-way between the values ℓ and u mentioned in Section 3.2.

For each integer program, a time limit of 5000 seconds was imposed. We were able to find the optimal solutions to 23 out of the 24 instances within the time limit. The one that was not solved was the instance with 100 nodes where $2/3$ of the nodes are required and the covariance matrix has non-zero elements only for adjacent links.

Table 2 shows, for each of the twenty three instances and each of the four formulations, the time taken to solve the continuous relaxation. Indices 'G', 'L' and 'N' refer to the general, low rank and nearby covariance structures, respectively. We see that, for the G and L instances, the relaxations of the mixed 0–1 CQP formulation are solved much more quickly than those of the RLT formulations. This is not the case for the N instances, presumably because the RLT formulations have fewer variables and constraints in the case of the N instances. For a similar reason, the relaxations of the SCF formulations can be solved more quickly than those of the MCF formulations, especially when the number of customers is large.

Table 2
Time taken to solve continuous relaxation (in seconds).

Instance	SCF- ϵ	MCF- ϵ	RLT-SCF- ϵ	RLT-MCF- ϵ
25-1/3-G	0.06	0.13	0.41	0.86
50-1/3-G	0.08	0.30	6.33	6.91
75-1/3-G	0.16	1.12	24.77	30.72
100-1/3-G	0.16	2.59	96.03	86.24
25-2/3-G	0.08	0.30	0.67	0.78
50-2/3-G	0.11	0.83	5.23	7.69
75-2/3-G	0.22	3.12	32.18	27.85
100-2/3-G	0.16	8.74	106.78	111.15
25-1/3-L	0.05	0.08	0.36	0.83
50-1/3-L	0.06	0.25	5.74	8.47
75-1/3-L	0.09	0.75	19.00	43.46
100-1/3-L	0.13	1.78	62.96	88.39
25-2/3-L	0.09	0.19	0.73	0.89
50-2/3-L	0.06	0.80	5.88	7.88
75-2/3-L	0.13	2.43	21.34	35.41
100-2/3-L	0.11	5.46	54.24	72.70
25-1/3-N	0.05	0.13	0.03	0.09
50-1/3-N	0.09	0.27	0.05	0.58
75-1/3-N	0.14	1.00	0.08	2.82
100-1/3-N	0.13	2.00	0.06	6.10
25-2/3-N	0.08	0.22	0.03	0.11
50-2/3-N	0.08	0.87	0.05	1.06
75-2/3-N	0.09	3.46	0.06	2.95

Table 3
Percentage integrality gaps.

Instance	SCF- ϵ	MCF- ϵ	RLT-SCF- ϵ	RLT-MCF- ϵ
25-1/3-G	53.0	0.0	72.5	57.6
50-1/3-G	51.9	0.0	88.4	78.8
75-1/3-G	63.8	0.4	95.5	92.4
100-1/3-G	68.0	0.0	100.0	98.4
25-2/3-G	40.4	0.0	62.6	54.5
50-2/3-G	42.1	2.2	79.5	68.4
75-2/3-G	55.6	0.0	95.2	93.6
100-2/3-G	65.1	2.0	100.0	100.0
25-1/3-L	56.6	0.0	72.2	23.1
50-1/3-L	49.8	0.0	77.2	44.9
75-1/3-L	44.0	0.0	83.2	59.8
100-1/3-L	55.4	1.2	98.7	85.5
25-2/3-L	34.1	0.0	53.2	28.8
50-2/3-L	29.1	0.7	61.1	51.1
75-2/3-L	32.5	0.3	81.2	68.7
100-2/3-L	35.2	0.0	91.5	84.6
25-1/3-N	33.7	0.9	68.5	67.9
50-1/3-N	37.4	11.3	82.2	81.4
75-1/3-N	43.4	4.7	94.8	92.9
100-1/3-N	46.9	11.6	100.0	100.0
25-2/3-N	21.7	0.9	69.8	69.5
50-2/3-N	27.6	7.3	83.2	78.1
75-2/3-N	32.2	6.2	95.8	94.6

Table 4
Number of branch-and-bound nodes.

Instance	SCF- ϵ	MCF- ϵ	RLT-SCF- ϵ	RLT-MCF- ϵ
25-1/3-G	156	0	842	44
50-1/3-G	50913	8	-	-
75-1/3-G	-	264	-	-
100-1/3-G	-	4	-	-
25-2/3-G	52	0	170	0
50-2/3-G	43005	32	-	-
75-2/3-G	-	19	-	-
100-2/3-G	-	78	-	-
25-1/3-L	50	0	201	10
50-1/3-L	3545	0	-	-
75-1/3-L	39417	0	-	-
100-1/3-L	-	57	-	-
25-2/3-L	72	0	240	11
50-2/3-L	82813	0	-	-
75-2/3-L	-	0	-	-
100-2/3-L	-	9	-	-
25-1/3-N	160	40	56	3
50-1/3-N	143175	-	3295	277
75-1/3-N	-	-	137587	682
100-1/3-N	-	-	-	16180
25-2/3-N	151	18	44	9
50-2/3-N	1333279	4590	45444	1120
75-2/3-N	-	-	-	7171

Table 3 shows, for the 23 instances that were solved to optimality, and for the same four formulations, the *percentage integrality gap*, i.e., the difference between the lower bound given by the LP solution and the cost of the true integer optimum, expressed as a percentage of the latter. We see that the gaps are far smaller for MCF- ϵ than for any of the other formulations. The superiority of the MCF formulation over the SCF formulation is explained in Letchford et al. (2013), but the reason for the severe weakness of the RLT bounds is not clear. We also remark that for the SCF formulations the gaps decrease as the proportion of required nodes increases, and for the RLT formulations the gaps increase with n . The reason for these two phenomena is not clear.

Table 4 shows, for each of the 23 instances solved, the number of branch-and-bound nodes needed to solve the integer programs to proven optimality. A dash (–) indicates that the given formulation could not be solved within the given time limit. We see that MCF- ϵ usually gives the true integer optimal solution using very few branch-and-bound nodes. In fact, for L instances with smaller values of n , no branching is required at all. Moreover, the mixed 0-1 CQP formulations need significantly fewer branch-and-bound than the RLT formulations in most cases. An important exception is the N instances, where the RLT formulations prove to be superior.

It should be noted that, in a few cases, the number of branch-and-bound nodes is zero even though the integrality gap is positive. The explanation for this anomalous behaviour is that, in some cases, CPLEX automatically appended some of its own internal cutting planes to the continuous relaxation. If one were to switch off all internal cutting plane generation in CPLEX, then branching would become necessary whenever there is a positive integrality gap.

Finally, Table 5 shows the total time, in seconds, taken by the branch-and-bound procedure. We see that despite having more variables and constraints, the MCF formulations perform better than the SCF ones. This is probably due to the relative tightness of their continuous relaxations. We see that formulation MCF- ϵ performs very well in most cases, except for the N instances, where RLT-MCF- ϵ does better. This is probably because the RLT formulations contain relatively few variables in the case of the N instances.

All things considered, the MCF formulations outperform the SCF ones. We recommend using MCF- ϵ for instances with general or low

Table 5
Branch-and-bound time (in seconds).

Instance	SCF- ϵ	MCF- ϵ	RLT-SCF- ϵ	RLT-MCF- ϵ
25-1/3-G	0.30	0.20	39.51	21.50
50-1/3-G	34.94	1.64	-	-
75-1/3-G	-	29.48	-	-
100-1/3-G	-	10.59	-	-
25-2/3-G	0.42	0.37	151.66	54.41
50-2/3-G	26.39	7.07	-	-
75-2/3-G	-	16.08	-	-
100-2/3-G	-	1713.62	-	-
25-1/3-L	0.26	0.09	41.28	13.07
50-1/3-L	2.48	0.50	-	-
75-1/3-L	29.19	1.22	-	-
100-1/3-L	-	738.35	-	-
25-2/3-L	0.28	0.20	66.72	7.33
50-2/3-L	47.99	10.92	-	-
75-2/3-L	-	26.18	-	-
100-2/3-L	-	90.81	-	-
25-1/3-N	0.22	0.36	0.34	0.42
50-1/3-N	70.11	-	5.35	5.97
75-1/3-N	-	-	573.38	66.71
100-1/3-N	-	-	-	4633.06
25-2/3-N	0.31	0.53	0.28	0.56
50-2/3-N	2078.12	163.32	63.85	46.66
75-2/3-N	-	-	-	749.10

rank covariance structures, butss RLT-MCF- ϵ for instances with significant covariances only between nearby links.

5.3. Approximating the efficient frontier

Now we explore the computation of approximate frontiers, using the method described in Section 3.2. For brevity, we concentrate on a single STSPCC instance, the one with $|V| = 50$, $|E| = 69$ and $|C| = 32$ and general correlation structure (“50-2/3-G” in the preceding tables). We set the parameter k to 20, and solved formulation MCF- ϵ for the corresponding values of ϵ . This meant that 21 integer programs had to be solved in total, each one taking around 10 seconds. Surprisingly, we found that we obtained only eight distinct solutions. The expected costs and variances of these solutions are plotted in Fig. 1. We remark

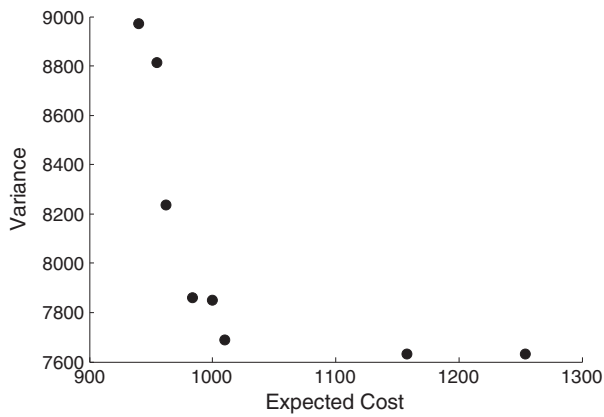


Fig. 1. Approximate frontier for instance with $|V| = 50$, $|E| = 69$ and $|C| = 32$.

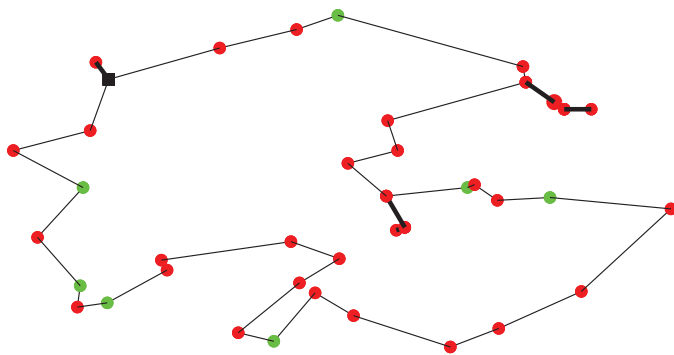


Fig. 2. Minimum expected cost tour.

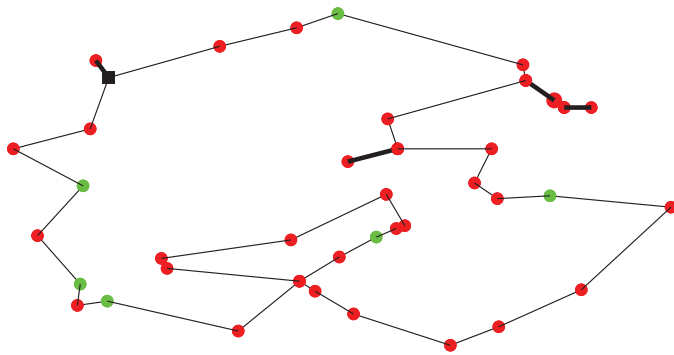


Fig. 3. Tour corresponding to fourth point in Fig. 1.

that two of the eight points in the figure (the second and fifth from the left) lie in the interior of the convex hull of the points. The solutions that correspond to these two points would not have been found if we had used the weighting method rather than the ϵ -constraint method (see Section 2.2).

In Figs. 2–4, we present three of the tours found by the algorithm. These correspond to the first, fourth and last of the points displayed in Fig. 1. In the colour version of this paper, the depot is represented by a black square, required nodes are displayed as red dots, visited non-required nodes are displayed as green dots, and non-visited non-required nodes are not displayed at all. Edges traversed once are drawn as narrow lines, and edges traversed twice (once in each direction) are drawn as thick lines. We see that, although the three tours are similar in shape, there are some noticeable differences. In particular, the latter two tours take some rather unexpected ‘diversions’ that are not present in the minimum expected cost tour.

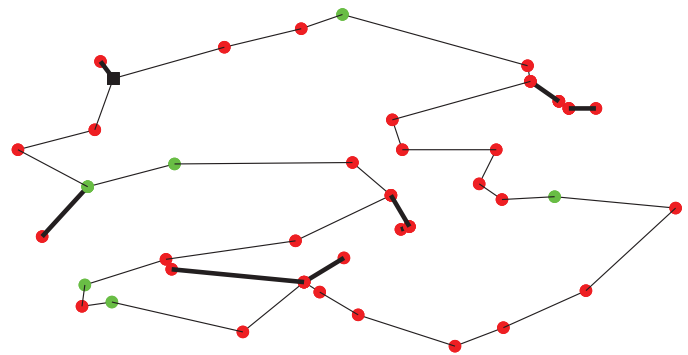


Fig. 4. Minimum variance tour.

6. Concluding remarks

In real-life road networks, the costs or travel times for individual roads are stochastic, since they may be affected by rush hours, or unexpected events such as road works or accidents. Moreover, as mentioned in Section 2.3, there is empirical evidence that the costs or times for pairs of roads can be correlated. Accordingly, we have proposed the Steiner TSP with correlated costs (STSPCC) as a new and potentially useful VRP. We presented four different integer programming formulations for finding efficient tours, along with an algorithm for approximating the efficient frontier. We also showed how to exploit special structures in the covariance matrix. Our experimental results have shown that instances with up to 100 nodes can be tackled with a standard branch-and-bound solver.

A possible extension of this work could be to devise methods for computing the entire efficient frontier for the STSPCC, or at least approximations of it. It would also be interesting to see if larger instances could be tackled by incorporating strong cutting planes into the formulation and solution algorithm. Another possible topic for future research would be to incorporate correlations in costs or travel times into other vehicle routing problems, or indeed into models of traffic and/or transportation in general.

References

- Adams, W. P., & Sherali, H. D. (1986). A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32, 1274–1290.
- Anderson, T. W. (2011). *An introduction to multivariate statistical analysis* (3rd ed.). Wiley series in probability and statistics. New York: Wiley.
- Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2006). *The traveling salesman problem: A computational study*. Princeton, NJ: Princeton University Press.
- Beale, E. M. L. (1959). On quadratic programming. *Naval Research Logistics Quarterly*, 6, 227–243.
- Chamberlain, G., & Rothschild, M. (1983). Arbitrage, factor structure and mean-variance analysis in large asset markets. *Econometrica*, 51, 1305–1324.
- Chaudhuri, S., Drton, M., & Richardson, T. S. (2007). Estimation of a covariance matrix with zeros. *Biometrika*, 94, 199–216.
- Chen, B. Y., Lam, W. H. K., Sumalee, A., & Li, Z. L. (2012). Reliable shortest path finding in stochastic networks with spatial correlated link travel times. *International Journal of Geographical Information Science*, 26, 365–386.
- Claus, A. (1984). A new formulation for the travelling salesman problem. *SIAM Journal on Algebraic and Discrete Methods*, 5, 21–25.
- Cornuéjols, G., Fonlupt, J., & Naddef, D. (1985). The travelling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33, 1–27.
- Dong, W., Li, M., Vo, Q. B., & Vu, H. L. (2012). Reliability in stochastic time-dependent traffic networks with correlated link travel times. *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 16–19.
- Ehrgott, M., & Gandibleux, X. (2007). Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research*, 34, 2674–2694.
- Fan, Y. Y., Kalaba, R. E., & Moore, J. E. (2005). Shortest paths in stochastic networks with correlated link costs. *Computers & Mathematics with Applications*, 49, 1549–1564.
- Fleischmann, B. (1985). A cutting plane procedure for the traveling salesman problem on a road network. *European Journal of Operational Research*, 21, 307–317.
- Fortet, R. (1959). L’algèbre de boole et ses applications en recherche opérationnelle. *Cahiers du Centre d’Études de Recherche Opérationnelle*, 4, 5–36.
- Gavish, B., & Graves, S. C. (1978). *The travelling salesman problem and related problems. (Working paper)*. Operations Research Centre, Massachusetts Institute of Technology.

- Gendreau, M., Laporte, G., & Seguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, 88, 3–12.
- Gutin, G., & Punnen, A. (Eds.). (2002). *The traveling salesman problem and its variations*. Dordrecht: Kluwer.
- Jula, H., Dessouky, M. M., & Ioannu, P. (2006). Truck route planning in non-stationary stochastic networks with time-windows at customer locations. *IEEE Transactions on Intelligent Transportation Systems*, 37, 51–63.
- Kao, E. P. C. (1978). A preference order dynamic program for a stochastic traveling salesman problem. *Operations Research*, 26, 1033–1045.
- Kenyon, A. S., & Morton, D. P. (2003). Stochastic vehicle routing with random travel times. *Transportation Science*, 37, 69–82.
- Lambert, V., Laporte, G., & Louveaux, F. (1993). Designing collection routes through bank branches. *Computers & Operations Research*, 20, 783–791.
- Laporte, G., Louveaux, F. V., & Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science*, 26, 161–169.
- Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R., & Shmoys, D. B. (1985). *The traveling salesman problem: A guided tour of combinatorial optimization*. Chichester: Wiley.
- Letchford, A. N., Nasiri, S. D., & Theis, D. O. (2013). Compact formulations for the Steiner travelling salesman problem and related problems. *European Journal of Operational Research*, 228, 83–92.
- Li, X., Tiana, P., & Leung, S. C. H. (2010). Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. *International Journal of Production Economics*, 125, 137–145.
- Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7, 77–91.
- Marsaglia, G. (1972). Choosing a point from the surface of a sphere. *Annals of Mathematical Statistics*, 43, 645–646.
- Mavrotas, G., & Diakoulaki, D. (1998). A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research*, 107, 530–541.
- Miettinen, K. (1999). Nonlinear multiobjective optimization. Vol. 12: *International series in operations research & management science*. New York: Springer.
- Muller, M. E. (1959). A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2, 19–20.
- Nasiri, S. D. (2014). Correlation between speeds on a congested road network in the city of London (submitted for publication).
- Orloff, C. S. (1974). A fundamental problem in vehicle routing. *Networks*, 4, 35–64.
- Samaranayake, S., Blandin, S., & Bayen, A. (2012). A tractable class of algorithms for reliable routing in stochastic networks. *Transportation Research C*, 20, 199–217.
- Secomandi, N. (2003). Analysis of a rollout approach to sequencing problems with stochastic routing applications. *Journal of Heuristics*, 9, 321–352.
- Sherali, H. D., & Adams, W. P. (1998). *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Dordrecht: Kluwer.
- Sourd, F., & Spanjaard, O. (2008). A multiobjective branch-and-bound framework: application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20, 472–484.
- Strang, G. (2009). *Introduction to linear algebra* (4th ed.). Wellesley MA: Wellesley-Cambridge Press.
- Teng, S. Y., Ong, H. L., & Huang, H. C. (2004). An integer L-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times. *Asia-Pacific Journal of Operational Research*, 21, 241–257.
- Toriello, A., Haskell, W. B., & Poremba, M. (2014). A dynamic travelling salesman problem with stochastic arc costs. *Operations Research*, 62, 1107–1125.
- Ulungu, E. L., & Teghem, J. (1995). The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing & Decision Sciences*, 20, 149–165.
- Wolfe, P. (1959). The simplex method for quadratic programming. *Econometrica*, 27, 382–398.