

The network maintenance problem on an arc with uncapacitated repair

P. Charkhgard ^a, T. Kalinowski ^b and H. Waterer ^a

^a*School of Mathematical and Physical Sciences, The University of Newcastle, Australia*

^b*School of Science and Technology, University of New England, Australia*

Email: parisa.charkhgard@uon.edu.au

Abstract: The network maintenance problem is motivated by the need to maintain infrastructure networks over time. We consider networks in which a commodity is transported between origin-destination pairs, and at the same time the infrastructure assets need to be maintained by resources moving in the network. In order to perform maintenance the assets have to be shut down thus reducing the system capacity. The objective is to maximise the total throughput by aligning the maintenance activities efficiently. In this paper, we study a special case of the network maintenance problem where the network consists of a single arc connecting an origin to a destination. Furthermore, there is no restriction on the amount of repair if the resource is to perform maintenance in a time period.

This problem is of interest for the following reasons. Firstly, it generalises variants of the lot-sizing problem and the warehouse problem, both of which have been well-studied in the literature. Secondly, we hope that understanding this special case will be useful in tackling more general variants of the network maintenance problem.

In this paper, we present a mixed integer linear programming formulation. We then show that a special class of feasible solutions, called *Maximum Flow Order Up* (MFOU) solutions, contains at least one optimal solution. Based on this result, we introduce an alternative integer linear programming formulation with only five decision variables. As a consequence, the optimal objective function value for any instance of the problem can be obtained in polynomial time.

Keywords: *Network maintenance, mixed integer linear programming, scheduling, polynomial time*

1 INTRODUCTION

The network maintenance problem, introduced by Charkhgard et al. (2017), is motivated by real-life applications in infrastructure networks, where the infrastructure assets are maintained over time. In this paper, we study a special case of the problem in which (1) the network consists of a single arc connecting an origin to a destination, and (2) there is no restriction on the amount of repair if we decide to do maintenance in a time period. Note that this can be a reasonable assumption if (i) the time periods are always longer than the maximum amount of maintenance required, and/or (ii) renewing the asset is the only option available for maintenance. Other special cases of the network maintenance problem are studied in Charkhgard (2019).

The formal definition of the problem is as follows. Consider an arc with capacity u that is connecting an origin node to a destination node. This arc is used to send flow between the two nodes. We are interested in scheduling maintenance activities on the arc so as to maximise the total flow over a time horizon $[T] = \{1, \dots, T\}$. The arc has an inventory which represents its condition. The higher the level of inventory, the better the condition of the arc. The initial inventory of the arc is σ and at its best/new condition the arc has inventory C . For each unit of flow that is sent along the arc, the inventory is decreased by one. The inventory can be increased to any amount no larger than C if the arc is under repair for one time period. When the inventory is zero, the arc is broken, implying that no flow can be sent through the arc. Furthermore, maintenance shuts down the arc, making its capacity zero in that period.

We define binary variable y_t that takes value one if the arc is under repair in period $t \in [T]$ and zero otherwise. We also define x_t to be the amount of repair and θ_t to be the amount of flow in period $t \in [T]$. Variable s_t denotes the amount of inventory at the end of period $t \in [0, T]$. Then the problem can be formulated as

$$\text{Maximise } \sum_{t \in [T]} \theta_t \tag{1}$$

$$\text{subject to: } \theta_t \leq u(1 - y_t) \tag{2} \quad t \in [T],$$

$$s_t = s_{t-1} - \theta_t + x_t \tag{3} \quad t \in [T],$$

$$x_t \leq C y_t \tag{4} \quad t \in [T],$$

$$s_t \leq C \tag{5} \quad t \in [T],$$

$$s_0 = \sigma \tag{6}$$

$$x_t, \theta_t \geq 0 \tag{7} \quad t \in [T],$$

$$s_t \geq 0 \tag{8} \quad t \in [0, T],$$

$$y_t \in \{0, 1\} \tag{9} \quad t \in [T].$$

Constraint (2) guarantees that the flow in time period $t \in [T]$ is a positive value only if the arc is not under repair in that time period. Moreover, it ensures that the flow in time period $t \in [T]$ does not exceed the flow capacity u . Constraint (3) is the inventory balance equation. Constraint (4) ensures that $x_t > 0$ only if $y_t = 1$. Constraint (5) guarantees that the inventory does not exceed the inventory upper bound. Constraint (6) captures the initial inventory. Finally, Constraints (7)–(9) specify the non-negativity and integrality of the decision variables.

Boland et al. (2014a) introduce an abstract network optimisation problem in which a set of arc maintenance tasks needs to be scheduled such that the total flow in the network over time is maximised. A simplified version of the problem in which all tasks have unit processing times is studied by Boland et al. (2014b). The authors establish the complexity of the problem taking into account certain instance characteristics, such as special network structures and restrictions on the set of tasks. Boland et al. (2016) extend the problem by adding a constraint to ensure that the number of tasks scheduled in any time period does not exceed a given number. These problems are different from the network maintenance problem as they assume that a set of maintenance tasks with their corresponding time windows are given a priori.

There are some similarities between the problem studied in this paper and the warehouse problem which was first introduced by Cahn (1948). The warehouse problem is to optimally decide on purchasing (or production), storage and sales quantities for a product with a fixed warehouse capacity and a given initial stock. If we consider the arc in our problem to be the warehouse in the warehouse problem, then increasing the inventory of the arc by performing maintenance is equivalent to increasing the inventory of the warehouse through purchasing. In addition, sending flow on the arc decreases its inventory which is equivalent to decreasing the inventory of the warehouse by selling product. However, the two problem are somewhat different. First,

there is no upper bound on sales in the warehouse problem whereas there is a flow capacity u in the network maintenance problem. Second, unlike the network maintenance problem, production and sales may occur in the same period in a typical warehouse problem. The reader is referred to Bellman (1956), and Secomandi (2010) for more details on warehouse problems.

Wolsey and Yaman (2018) extend the warehouse problem by including per unit inventory holding costs, production and selling fixed costs, and a constraint that does not allow production and sales in the same period. Their research is the first study on strong formulations of the warehouse problem in the presence of fixed costs. They provide convex hull descriptions and tight compact extended formulations. To the best of our knowledge, there is no study on the warehouse problem in the presence of an upper bound on sales, as well as a restriction on production and sales occurring in the same period.

We can also compare our problem to the lot-sizing problem first introduced by Wagner and Whitin (1958). The simplest variant of the lot-sizing problem is to optimally decide on production and storage quantities for an item to satisfy a known demand in each time period starting with a given initial stock. If we consider the arc in our problem to be an item in the lot-sizing problem, then increasing the inventory of the arc by performing maintenance is equivalent to increasing the inventory of the item by production. In addition, sending flow on the arc decreases its inventory which is equivalent to decreasing the inventory of the item by satisfying the demand. Lot-sizing problems have been extensively studied in the literature, and the reader is referred to Pochet and Wolsey (2006) for more details on these problems. In most of the models in the literature, demand is assumed to be known, and the common objective is to minimise total cost. Loparic et al. (2001) examine a variant of the uncapacitated lot-sizing problem involving sales instead of fixed demands, and lower bounds on stocks. The objective in their model is profit maximisation. They present two extended formulations, a dynamic programming algorithm and a complete description of the convex hull of solutions. Di Summa and Wolsey (2011) present a linear-inequality description for the convex hull of the feasible region of the single-item discrete lot-sizing problem with sales. Furthermore, they show that the single-item discrete lot-sizing problem with sales can be solved in polynomial time. In most studies on the lot-sizing problem inventory is assumed to be unbounded, however Atamtürk and Küçükyavuz (2005) investigate the polyhedral structure of the lot-sizing problem with inventory bounds. Wolsey (2017) presents a tight extended shortest path formulation of the convex hull of solutions for the uncapacitated lot-sizing problem with stock upper bounds, stock fixed costs, stock overload and backlogging.

The problem studied in this paper is similar to a lot-sizing problem that simultaneously considers all the following extensions: (1) sales with no demand, (2) upper bound on inventory, (3) upper bound on sales, and a (4) restriction on production and sales occurring in the same period. To the best of our knowledge, there is no study in the literature of the lot-sizing problem which contains all these extensions together.

This paper is organised as follows. In Section 2 we define a class of feasible solutions to Formulation (1)–(9), *Maximum Flow Order Up* (MFOU) solutions, and prove that it always contains an optimal solution for the problem. In Section 3 we reformulate the problem as an integer linear programming with a fixed number of variables and consequently the optimal objective value of the problem can be computed in polynomial time. Finally, conclusions and future work are discussed in Section 4.

2 MAXIMUM FLOW ORDER UP (MFOU) SOLUTIONS

We begin this section with some definitions. Let $X = X(C, u, \sigma, T)$ be the feasible set in Formulation (1)–(9), that is,

$$X = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}, \boldsymbol{\theta}) \in \mathbb{R}^{4T+1} : (2)–(9)\},$$

and let $f = f(C, u, \sigma, T)$ be the optimal objective value in Formulation (1)–(9), that is,

$$f = \max \left\{ \sum_{t \in [T]} \theta_t : (\mathbf{x}, \mathbf{y}, \mathbf{s}, \boldsymbol{\theta}) \in X(C, u, \sigma, T) \right\}.$$

Period $t \in [T]$ is said to be a *repair period* with respect to a solution $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \boldsymbol{\theta}) \in X$ if $y_t = 1$, and a *flow period* otherwise, so that we obtain a partition $[T] = \mathcal{R} \cup \mathcal{F}$, where $\mathcal{R} = \mathcal{R}(\mathbf{y})$ is the set of repair periods and $\mathcal{F} = \mathcal{F}(\mathbf{y})$ is the set of flow periods. In this section we prove that there is always an optimal solution satisfying the conditions in the following definition.

Definition 1. A solution $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \boldsymbol{\theta}) \in X$ is said to be a *Maximum Flow Order Up* (MFOU) solution if the following conditions are satisfied for all $t \in [T]$:

1. $y_t = 1 \implies s_t = C$,
2. $s_{t-1} = 0 \implies y_t = 1$,
3. $y_t = 0 \implies \theta_t = \min\{u, s_{t-1}\}$,
4. $s_{t-1} \geq u \implies y_t = 0$.

Condition 1 implies that in every repair period $t \in [T]$, we repair as much as possible so that the inventory is C at the end of period t . Condition 2 requires that $t \in [T]$ is a repair period if the inventory at the end of period $t - 1$ is 0. Condition 3 implies that if $t \in [T]$ is a flow period then we send as much flow as possible. Finally, Condition 4 requires that period $t \in [T]$ is a flow period if the inventory at period $t - 1$ is at least u . Note that Conditions 1 and 3 indicate that MFOU solution $(x, y, s, \theta) \in X$ is uniquely determined by its set of repair periods \mathcal{R} . Next we present two observations and four lemmas that enable us to prove that there is always an optimal solution in X which is MFOU.

Observation 1. *If Condition 1 in Definition 1 is violated in period $t \in [T]$ then $y_t = 1$, and therefore Conditions 2 and 3 are satisfied in period t .*

Observation 2. *If Condition 4 in Definition 1 is violated in period $t \in [T]$ then $y_t = 1$, and therefore Conditions 2 and 3 are satisfied in period t .*

To prove each of the following lemmas we need to define a solution (x', y', s', θ') and show that this solution is (a) feasible, (b) optimal, and (c) satisfies Conditions 1–4 in periods $t \leq t_0$ for any $t_0 \in [T]$. The details of the proofs can be found in Charkhgard (2019).

Lemma 1. *Let (x, y, s, θ) be an optimal solution which satisfies Conditions 1–4 in Definition 1 in periods $t < t_0$, and violates Condition 4 in period $t = t_0$. There exists an optimal solution (x', y', s', θ') which satisfies Conditions 1–4 in periods $t \leq t_0$.*

Solution (x', y', s', θ') in Lemma 1 can be defined as follows:

$$\begin{aligned}
 x'_t &= x_t, y'_t = y_t, s'_t = s_t, \theta'_t = \theta_t & t \in [t_0 - 1] \\
 y'_{t_0} &= 0, x'_{t_0} = 0, \theta'_{t_0} = u, s'_{t_0} = s_{t_0-1} - u \\
 y'_{t_0+1} &= 1, x'_{t_0+1} = C - s'_{t_0}, \theta'_{t_0+1} = 0, s'_{t_0+1} = C \\
 y'_t &= y_t, \theta'_t = \theta_t & t \in [t_0 + 2, T] \\
 x'_t &= \min\{x_t, C - s'_{t-1}\} & t \in [t_0 + 2, T] \\
 s'_t &= s'_{t-1} + x'_t - \theta'_t & t \in [t_0 + 2, T]
 \end{aligned}$$

Lemma 2. *Let (x, y, s, θ) be an optimal solution which satisfies Conditions 1–4 in Definition 1 in periods $t < t_0$, and violates Condition 1 but not Condition 4 in period $t = t_0$. There exists an optimal solution (x', y', s', θ') which satisfies Conditions 1–4 in periods $t \leq t_0$.*

Solution (x', y', s', θ') in Lemma 2 can be defined as follows:

$$\begin{aligned}
 x'_t &= x_t, y'_t = y_t, s'_t = s_t, \theta'_t = \theta_t & t \in [t_0 - 1] \\
 y'_{t_0} &= 1, s'_{t_0} = C, x'_{t_0} = C - s_{t_0-1}, \theta'_{t_0} = 0 \\
 y'_t &= y_t, \theta'_t = \theta_t & t \in [t_0 + 1, T] \\
 x'_t &= \min\{x_t, C - s'_{t-1}\} & t \in [t_0 + 1, T] \\
 s'_t &= s'_{t-1} + x'_t - \theta'_t & t \in [t_0 + 1, T]
 \end{aligned}$$

Lemma 3. *Let (x, y, s, θ) be an optimal solution which satisfies Conditions 1–4 in Definition 1 in periods $t < t_0$, and violates Condition 2 in period $t = t_0$. There exists an optimal solution (x', y', s', θ') which satisfies Conditions 1–4 in periods $t \leq t_0$.*

Solution (x', y', s', θ') in Lemma 3 can be defined as follows:

$$\begin{aligned}
 x'_t &= x_t, y'_t = y_t, s'_t = s_t, \theta'_t = \theta_t & t \in [t_0 - 1] \\
 y'_{t_0} &= 1, s'_{t_0} = C, x'_{t_0} = C, \theta'_{t_0} = 0 \\
 y'_t &= y_t, \theta'_t = \theta_t & t \in [t_0 + 1, T] \\
 x'_t &= \min\{x_t, C - s'_{t-1}\} & t \in [t_0 + 1, T] \\
 s'_t &= s'_{t-1} + x'_t - \theta'_t & t \in [t_0 + 1, T]
 \end{aligned}$$

Lemma 4. *Let (x, y, s, θ) be an optimal solution which satisfies Conditions 1–4 in Definition 1 i periods $t < t_0$, and violates Condition 3 but not Condition 2 in period $t = t_0$. There exists an optimal solution (x', y', s', θ') which satisfies Conditions 1–4 in periods $t \leq t_0$.*

Solution (x', y', s', θ') in Lemma 4 can be defined as follows:

$$\begin{aligned}
 x'_t &= x_t, y'_t = y_t, s'_t = s_t, \theta'_t = \theta_t & t \in [t_0 - 1] \\
 y'_{t_0} &= 0, x'_{t_0} = 0, \theta'_{t_0} = \min\{u, s_{t_0-1}\}, s'_{t_0} = s_{t_0-1} - \theta'_{t_0} \\
 y'_t &= y_t, x'_t = x_t & t \in [t_0 + 1, T] \\
 \theta'_t &= \min\{\theta_t, s'_{t-1}\} & t \in [t_0 + 1, T] \\
 s'_t &= s'_{t-1} + x_t - \theta'_t & t \in [t_0 + 1, T]
 \end{aligned}$$

The following proposition follows from Observations 1–2 and Lemmas 1–4.

Proposition 1. *There always exists an optimal solution in $X(m, \rho, C, u, \sigma, T)$ which is MFOU.*

Proof. Assume every optimal solution violates at least one of the conditions. We pick the optimal solution (x, y, s, θ) in which the first violation occurs as late as possible, say at time period t_0 . Using Lemmas 1–4, it is always possible to transform this optimal solution to another optimal solution (x', y', s', θ') in which the violation occurs later than t_0 . This contradicts our initial assumption. \square

3 PROBLEM REFORMULATION

Let $C = p_0u + r_0$ and $\sigma = qu + r$ where $r, r_0 \in [0, u - 1]$, $p_0 \in \mathbb{Z}_{>0}$ and $q \in \mathbb{Z}_{\geq 0}$. The problem instances with $T \leq q + 1$ have trivial optimal solutions. For these instances, the MFOU solution with the set of repair periods $\mathcal{R} = \emptyset$ (i.e., the set of flow periods $\mathcal{F} = [T]$) is an optimal solution. We now focus on the instances with $T \geq q + 2$. We will show that these instances can be modelled as a budget constrained longest walk problem on a digraph H with node set $V = \{\sigma, C, D\}$ and arc set A (described below) which is illustrated in Figure 1.

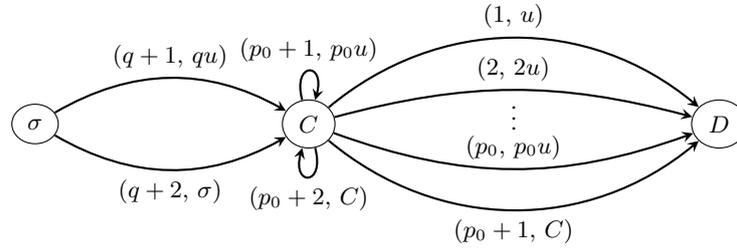


Figure 1: The digraph H corresponding (C, u, σ) . The arcs are labelled by (cost, utility).

There are two functions associated with H : a cost function $h: A \rightarrow \mathbb{R}$ and a utility function $g: A \rightarrow \mathbb{R}$. Node σ represents the initial state, node C represents the state that the inventory is equal to C , and in an MFOU solution this state is reached after every repair period, and node D represents the end of the time horizon. The digraph has arc set A consisting of the following $p_0 + 5$ arcs:

- The first arc from σ to C corresponds to q flow periods followed by one repair period. This arc has cost $q + 1$ and utility qu .
- The second arc from σ to C corresponds to $q + 1$ flow periods followed by one repair period. This arc has cost $q + 2$ and utility σ .
- The first loop on node C corresponds to p_0 flow periods followed by one repair period. It has cost $p_0 + 1$ and utility p_0u .
- The second loop on node C corresponds to $p_0 + 1$ flow periods followed by one repair period. It has cost $p_0 + 2$ and utility C .

- For every $i \in [p_0]$, there is an arc from C to D which represents i flow periods. It has cost i and utility iu .
- An arc from C to D which corresponds to $p_0 + 1$ flow periods. It has cost $p_0 + 1$ and utility C .

The next proposition implies that the optimal objective value for the original problem, $f(C, u, \sigma, T)$, is the maximum utility of a walk from σ to D in Digraph H with total cost T .

Proposition 2. *Any maximum utility walk from σ to D with total cost T corresponds to an optimal MFOU solution, and the optimal objective value is equal to the total utility of the walk.*

The details of the proof of Proposition 2 can be found in Charkhgard (2019). In the proof we show that (a) for every walk from σ to D with cost T , there is an MFOU solution in X with the objective value equal to the total utility of the walk, and (b) for every MFOU solution $(x, y, s, \theta) \in X$, there is a walk from σ to D with utility $\sum_{t \in [T]} \theta_t$ and cost T .

Our next step is to find a maximum utility walk from σ to D with total cost T . Let ξ_1 and ξ_2 denote the number of times loops with labels $(p_0 + 1, p_0 u)$ and $(p_0 + 2, C)$ respectively are used in a walk from σ to D . We also define binary variable η that takes value zero if the arc with label $(q + 1, qu)$ is used in the walk and one if arc $(q + 2, \sigma)$ is used. To capture the arcs between nodes C and D we define variables κ and μ . The binary variable κ takes value one if the arc between C to D with cost $p_0 + 1$ and utility C is chosen. The integer variable μ takes positive values $i \in [p_0]$. The following integer linear program finds a walk with maximum utility and total cost T .

$$\text{Maximise } (C - r_0)\xi_1 + C\xi_2 + r\eta + u\mu + r_0\kappa + (\sigma - r) \quad (10)$$

$$\text{subject to: } \eta + (p_0 + 1)\xi_1 + (p_0 + 2)\xi_2 + \mu + \kappa = T - q - 1 \quad (11)$$

$$1 \leq \mu \leq p_0 \quad (12)$$

$$\xi_1, \xi_2, \mu \in \mathbb{Z}_{\geq 0} \quad (13)$$

$$\eta, \kappa \in \{0, 1\} \quad (14)$$

Note that since $u > r_0$, $\kappa = 1$ only if $\mu = p_0$. Let $(\xi_1, \xi_2, \mu, \eta, \kappa)$ be an optimal solution for Formulation (10)–(14), then the corresponding optimal MFOU solution can be specified by the following set of repair periods:

$$\mathcal{R} = \{q + \eta + 1 + i(p_0 + 1) : i \in [0, \xi_1]\} \cup \{q + \eta + 1 + \xi_1(p_0 + 1) + i(p_0 + 2) : i \in [\xi_2]\}.$$

To prove that the optimal objective value of the problem can be computed in polynomial time we use the following result by Lokshtanov (2009).

Theorem 1. *An integer linear programming problem with p variables can be solved using $O(p^{2.5p+o(p)} \cdot L \cdot \log(MN))$ arithmetic operations and space polynomial in L . Here L is the number of bits in the input, N is the maximum of the absolute values any variable can take, and M is an upper bound on the absolute value of the maximum taken by the objective function.*

Proposition 3. *Given (C, u, σ, T) , the optimal objective value of the problem, $f(C, u, \sigma, T)$, can be computed by an algorithm whose run-time is bounded by a polynomial in $\log(CT)$.*

Proof. According to Proposition 2 the optimal objective value for the instance with parameters (C, u, σ, T) is equal to the optimal objective value of the Formulation (10)–(14). This formulation is an integer program with five variables and three constraints that can be solved in time $O(L^2)$, where $L = O(\log(CT))$ is the number of bits required to write down the integer program. To see this we apply Theorem 1 with $p = 5$, $N = T$ and $M = CT$. Then the run-time is bounded by $O(L \log(MN)) = O(L(\log M + \log N)) = O(L^2)$. \square

4 CONCLUSIONS

In this paper, we studied a special case of the network maintenance problem where the network is a single arc and in a repair period it is always possible to increase the inventory to its upper bound C . We presented a mixed integer linear programming formulation for the problem and then showed that a special class of feasible solutions, MFOU solutions, contains at least one optimal solution. This result enabled us to reformulate the problem as a budget constrained longest walk problem on a special digraph with 3 nodes and $p_0 + 5$ arcs. We showed that for any instance, an integer linear programming formulation of the longest walk problem has 5

P. Charkhgard et al., The network maintenance problem on an arc with uncapacitated repair.

variables and 3 constraints, thus the optimal objective value of the problem can be computed by an algorithm whose run-time is bounded by a polynomial in $\log(CT)$.

Formulation (10)–(14) has only 5 variables and 3 constraints. One future direction is to characterise the optimal solutions for this formulation as we envisage that this will lead to a combinatorial algorithm for the problem that runs in polynomial time. Another future direction is to study other special cases of the network maintenance problem. See Charkhgard (2019) for some progress in this direction.

ACKNOWLEDGEMENT

This research was supported by the ARC Linkage Grant LP140101000. We thank Aurizon and the Australian Research Council for their joint funding under the ARC Linkage Grant LP140101000.

REFERENCES

- Atamtürk, A. and S. Küçükyavuz (2005). Lot sizing with inventory bounds and fixed costs: Polyhedral study and computation. *Operations Research* 53(4), 711–730.
- Bellman, R. (1956). On the theory of dynamic programming—A warehousing problem. *Management Science* 2(3), 272–275.
- Boland, N., T. Kalinowski, and S. Kaur (2016). Scheduling arc shut downs in a network to maximise flow over time with a bounded number of jobs per time period. *Journal of Combinatorial Optimization* 32(3), 885–905.
- Boland, N., T. Kalinowski, H. Waterer, and L. Zheng (2014a). Scheduling arc maintenance jobs in a network to maximise total flow over time. *Discrete Applied Mathematics* 163, 34–52.
- Boland, N., R. Kapoor, S. Kaur, and T. Kalinowski (2014b). Scheduling unit time arc shutdowns to maximise network flow over time: Complexity results. *Networks* 63(2), 196–202.
- Cahn, A. (1948). The warehouse problem. In *Bulletin of the American Mathematical Society*, Volume 54, pp. 1073–1073.
- Charkhgard, P. (2019). *The network maintenance problem*. Ph. D. thesis, The University of Newcastle, Australia. In preparation.
- Charkhgard, P., T. Kalinowski, and H. Waterer (2017, December). The network maintenance problem. In G. Syme, D. Hatton MacDonald, B. Fulton, and J. Piantadosi (Eds.), *MODSIM2017, 22nd International Congress on Modelling and Simulation*, pp. 1302–1308. Modelling and Simulation Society of Australia and New Zealand. <https://www.mssanz.org.au/modsim2017/J5/charkhgard.pdf>.
- Di Summa, M. and L. A. Wolsey (2011). Mixing sets linked by bidirected paths. *SIAM Journal on Optimization* 21(4), 1594–1613.
- Lokshantov, D. (2009). *New methods in parameterized algorithms and complexity*. Ph. D. thesis, University of Bergen, Norway.
- Loparic, M., Y. Pochet, and L. A. Wolsey (2001). The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming* 89(3), 487–504.
- Pochet, Y. and L. A. Wolsey (2006). *Production planning by mixed integer programming*. Springer Science & Business Media.
- Secomandi, N. (2010). Optimal commodity trading with a capacitated storage asset. *Management Science* 56(3), 449–467.
- Wagner, H. M. and T. M. Whitin (1958). Dynamic version of the economic lot size model. *Management Science* 5(1), 89–96.
- Wolsey, L. A. (2017). Erratum: A tight formulation for uncapacitated lot-sizing with stock upper bounds. *Mathematical Programming* 161(1), 603–609.
- Wolsey, L. A. and H. Yaman (2018). Convex hull results for the warehouse problem. *Discrete Optimization* 30(11), 108–120.