

Throughput rate optimization in high multiplicity sequencing problems

Citation for published version (APA):

Grigoriev, A., & van de Klundert, J. (2001). Throughput rate optimization in high multiplicity sequencing problems. (METEOR research memorandum; No. 006). Maastricht: METEOR, Maastricht University School of Business and Economics.

Document status and date:

Published: 01/01/2001

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Throughput Rate Optimization in High Multiplicity Sequencing Problems

Alexander Grigoriev

A.Grigoriev@KE.unimaas.nl

Maastricht University, 6200 MD, Maastricht, The Netherlands

Joris van de Klundert

J.vandeKlundert@MATH.unimaas.nl

Maastricht University, 6200 MD, Maastricht, The Netherlands

October 12, 2001

Abstract

Mixed model assembly systems assemble products (parts) of different types in certain prespecified quantities. A minimal part set is a smallest possible set of product type quantities, to be called the multiplicities, in which the numbers of assembled products of the various types are in the desired ratios. It is common practice to repeatedly assemble minimal part sets, and in such a way that the products of each of the minimal part sets are assembled in the same sequence. Little is known however regarding the resulting throughput rate, in particular in comparison to the throughput rates attainable by other input strategies. This paper investigates throughput and balancing issues in repetitive manufacturing environments. It considers sequencing problems that occur in this setting and how the repetition strategy influences throughput. We model the problems as a generalization of the traveling salesman problem and derive our results in this general setting. Our analysis uses well known concepts from scheduling theory and combinatorial optimization.

Keywords: Manufacturing: performance, productivity; Production/scheduling: approximations, line balancing, sequencing; Networks/graphs: traveling salesman.

1 Introduction

Mass customization continues to place new demands on manufacturers. These new demands concern manufacturing technology, as well as planning and scheduling issues. The customization is often achieved by offering a choice of (optional) parts on a generic product. In manufacturing terms, this comes down to being able to customizing the generic product during the assembly stage, using various components. This concept has become known as assemble to order. Together with lean manufacturing requirements, assemble to order techniques pose challenging new planning and scheduling problems for manufacturers.

In a mass customization environment, demand for products is not entirely handled at the customer level. For several stage of the production process, demand is rather dealt with at the level of product variations, for instance because the figures are (partly) forecasts. The idea is that there is one product variation for each possible choice of components (or several closely resembling choices combine into one variation). In this paper we refer to the variations as types. Further, we deal with a situation where the assembly process is executed on an assembly line. As is customary in such settings, we assume that demand is specified at the level of product types: expected number of products per type for the next planning period, e.g. a month. The question that we would like to answer is: how to sequence all products of the various types through the assembly process?

Of course, this question has been studied before in several different contexts. First of all, there is a line of research dealing with so-called assembly line balancing problems (see e.g. Aigbedo and Monden (1997), Miltenburg (1989), Kubiak et al. (1997). This research originates from car assembly line scheduling problems at Toyota. Mostly, these papers assume that each assembly step takes a same, constant, amount of time. The task might now be to sequence the cars in such a way that the output levels are at any time proportional to the demand ratios. Alternatively, the sequencing problem might be to sequence the cars in such a way that the lines feeding the assembly line have a balanced work load.

Another line of research, having its roots in a variety of different manufacturing applications, also deals with related sequencing problem. These problems may again assume a linear production system, but the sequencing objective is now to optimize throughput rather than the the line balance. Related models can be found in maintenance and data broadcast scheduling Zhang and van der Veen (1996), Anily et al. (1998), Kenyon et al. (2000); Bar-Noy et al. (1998).

The problem under investigation can be further defined and specified as follows. We denote by $J = \{1, 2, \dots, N\}$ the set of product (job) types. For each product (job) type $j \in J$ there is a nonnegative integer d_j describing the demand of the j -th product for the next planning period, which specifies the number of type j jobs to be assembled. Further, we denote by $c_{i,j}$ the time interval required between inputting a type i job and a type j job, $i, j \in J$, into the system. These $c_{i,j}$'s can be considered as sequence dependent processing times in a sin-

gle machine scheduling problem, as sequence dependent switch over times, or as the waiting times between inputting parts into a permutation no-wait flowshop. Such a permutation no-wait flowshop system is often an adequate model for an assembly system. Notice that this problem specification is quite general. (For instance it contains scheduling problems in which the processing times are not sequence dependent as a special case.)

When $d_j = 1$ for all $j \in J$, and interpreting the $c_{i,j}$ as the distance between a city i and a city j , the sequencing problem that minimizes the sum of the sequence dependent $c_{i,j}$ is the classic traveling salesman problem (TSP). For reasons that will become clear soon, we refer to scheduling problems in which $d_j = 1$ for all job types as *single multiplicity* problems. In the classical, single multiplicity, TSP there is an explicit input parameter $c_{i,j}$ for each pair of cities. Hence, the length of the encoded input is $O(N^2 \times \max_{i,j \in \{J \times J\}} c_{i,j})$. In the sequencing problem that we investigate in this paper there are multiplicities d_j , $j \in J$, of jobs having the same processing requirements. Therefore, in any natural and compact encoding the input is of length $O(N^2 \times \max(\max_{j \in J} d_j, \max_{i,j \in \{J \times J\}} c_{i,j}))$. Scheduling (and sequencing) problems in which the input is specified this way, have become known as *high multiplicity scheduling* problems. Other high multiplicity scheduling problems can be found for instance in Hochbaum and Shamir (1991), and in Clifford and Posner (1996).

From a complexity perspective, high multiplicity scheduling problems have lead to several characteristic questions. Let us demonstrate this by considering high multiplicity sequencing problems with a makespan minimization objective. First of all, note that explicitly specifying a high multiplicity sequence of jobs requires an amount of space that is not polynomial in the input size. Therefore, it is not a priori clear that high multiplicity sequencing problem are in P-SPACE, let alone in NP. Hence, researchers have investigated whether optimal sequences can be compactly encoded. In particular, a question that has attracted attention are: Can polynomial algorithms for single multiplicity problem be extended so as to solve in polynomial time, the high multiplicity version (see e.g. Agnetis (1997))? Another question that has been researched is: for problems that are known to be NP-Complete in the single multiplicity version, is the high multiplicity version solvable in polynomial time if the number of types is fixed? This question is especially interesting since in many practical applications, the number of different types will be small compared to the numbers of multiplicities.

In practical settings in which the above high multiplicity scheduling problems arises, makespan optimization is important but not necessarily the dominant objective function. In addition, managers and planners think it to be desirable to have the output over time more or less in proportion to the multiplicities of the types. Among the advantages of such a policy are low inventories and a smooth work load. Consequently, it has become standard practice to repeatedly execute a short assembly sequence in which the jobs are produced in the required relative quantities. The concept of *minimal part sets* builds on this idea Hitz

(1979), Hitz (1980). A minimal part set is constructed as follows. Let d_0 be the greatest common divisor of the set of multiplicities $\{d_1, d_2, \dots, d_N\}$. Further let $k_j = d_j/d_0$, for $j \in J$. The minimal part set (MPS) is described by the vector $\{k_1, k_2, \dots, k_N\}$. A popular approach to finding a good sequence is now to find a sequence for the MPS, that when repeatedly executed, yields among all such sequences for MPSs the highest throughput rate. Alternatively, we shall formulate the problem in terms of finding an MPS sequence with shortest cycle time.

In assembly or production systems, the approach described above ideally results in low inventories, balanced workloads, and a simple and stable schedule, which is not too complicated to find. However, this approach may well yield sub-optimal throughput rates. In this paper we investigate high multiplicity sequencing problems with a cycle time minimization objective. In particular we are interested in the trade-off between long run cycle time versus the length of the sequence that is repeatedly executed. Indeed, a basic question that we will address is whether for a given instance there exists a finite repetitive cyclic policy that yields the optimal throughput rate.

The permutation no-wait flowshop scheduling problem $F|no - wait|C_{\max}$ in the representation scheme of Graham et al. (1979), is an important application for the research presented in this paper. We therefore quickly review its (high multiplicity) complexity status. It is well known, see e.g. Wismer (1972), that the permutation no-wait flowshop scheduling problem can be modeled as a traveling salesman problem (TSP). For the case of two machines, the resulting TSP instances have a distance matrix with a special structure, and, in the single multiplicity case, the cycle time minimization problem can be solved in polynomial time by the algorithm of Gilmore & Gomory Gilmore and Gomory (1964). Agnetis in Agnetis (1997) shows that the high multiplicity version of the two machine no-wait flowshop scheduling problem can also be solved in polynomial time, by an extension of the Gilmore & Gomory algorithm. From Röck (1981) we know that for three and more machines the decision versions of permutation no-wait flowshop scheduling problems are NP-complete in the strong sense as well as the traveling salesman problem in general.

The idea of finding an optimal sequence for an MPS is encountered in various other settings, such as the literature on assembly line balancing, or the literature on maintenance scheduling. Another previously investigated problem that fits in this framework is the the high multiplicity TSP (HMTSP), which is sometimes referred to as "the traveling salesman problem with many visits to few cities". We derive several results that can be applied to HMTSP. In this respect, our work extends earlier work on the HMTSP, as it can be found in Cosmadakis and Papadimitriou (1984), Papadimitriou and Kanellakis (1980), Psaraftis (1980), Zhang and van der Veen (1996), Van de Klundert (1995).

2 Problem Statement and Formulations

In this paper we consider a general high multiplicity scheduling problem with sequence dependent processing times. In fact, we will not be interested so much in the actual scheduling of tasks, i.e. specifying starting times, but rather in sequencing jobs, i.e. determining the order in which they are to be processed. More specifically, we will be interested in what we call the high multiplicity traveling salesman problem, since it is general enough to serve as a model for a variety of scheduling applications (see e.g. Remark 1 or Cosmadakis and Papadimitriou (1984); Zhang and van der Veen (1996)).

Definition 2.1 *The high multiplicity traveling salesman problem (HMTSP):*

$$\begin{aligned}
 P : \quad & \min_x \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j} & (i) \\
 \text{subject to} \quad & \sum_{i \in J} x_{i,j} = k_j, & j \in J; & (ii) \\
 & \sum_{j \in J} x_{i,j} = k_i, & i \in J; & (iii) \\
 & \sum_{i \in J'} \sum_{j \in J \setminus J'} x_{i,j} \geq 1, & \forall J' \subset J \text{ such that } J' \neq \emptyset; & (iv) \\
 & x_{i,j} \in \mathbf{Z}_+, & i \in J, j \in J. & (v)
 \end{aligned}$$

For a generic instance we will refer to this formulation as P , and to the value of its optimal solution as $F(P)$.

Remark 1. *Consider the high multiplicity formulation of the permutation no-wait flowshop for a single MPS with cycle time minimization objective. For each ordered pair of types $(i, j) \in J \times J$ it is possible to compute a number $c_{i,j}$ which is the time that must elapse between consecutively sequenced products (jobs, parts) of type i and type j respectively. That is*

$$c_{i,j} = p_{1,i} + \max_{m=1, \dots, M-1} \left(0, \sum_{n=1}^m (p_{n,j} - p_{n+1,i}) \right).$$

It is well known (see e.g. Wismer (1972)) that using the parameters $c_{i,j}$, $i \in J$, $j \in J$, the classical, single multiplicity, no-wait flowshop problem can be straightforwardly formulated as TSP. Indeed, letting the jobs correspond to cities, and letting $c_{i,j}$, $i \in J$, $j \in J$, be the distance between cities i and j , the task is to find a closed walk through all cities such that visiting every city exactly once and total length of which is minimal overall such walks.

The HMTSP is also known as the traveling salesman problem with many visits to few cities. From its definition as an integer linear program we conclude

that the decision version of the HMTSP, is in NP (despite the exponential number of constraints). Moreover, using Lenstra's algorithm Lenstra (1983) it can be solved in polynomial time when the number of cities (types) is fixed. In fact combinatorial, strongly polynomial, algorithms for the case where the number of cities is bounded from above by a constant have been developed by Cosmadakis, Kanelakis, Papadimitriou and Van de Klundert in Cosmadakis and Papadimitriou (1984); Papadimitriou and Kanellakis (1980); Van de Klundert (1995).

We now explain how an (optimal) *sequence* can be constructed from an (optimal) solution x^0 to P (see for instance Zhang and van der Veen (1996); Eiselt et al. (1995); Van de Klundert (1995)). Let $[x_{i,j}^0]_{N \times N}$ be an (optimal) solution to (P) . We are going to describe a procedure *ConvertToSequence* that can be repeatedly executed to convert the solution and matrix $[x_{i,j}^0]_{N \times N}$ into a closed walk that traverses arc (i, j) , exactly $x_{i,j}$ times $(i, j \in J)$, and which represents a high multiplicity sequence in which to visit the cities.

We denote by $G(J, A)$ the multigraph in which the nodes correspond to the cities in the HMTSP, and each arc (i, j) occurs with multiplicity $x_{i,j}^0$. In the procedure *ConvertToSequence* we use the phrase "simple cycle", which is to be understood as a simple cycle in G . More precisely, we define a cycle $c = ((i_1, i_2), (i_2, i_3), \dots, (i_t, i_1))$, $i_k \in J$, $k \in 1, 2, \dots, t$ to be a *simple cycle* if it doesn't contain any subcycle, i.e. if $i_q \neq i_r$ for $q, r \in 1, \dots, t$ and $q \neq r$.

Definition 2.2 *Procedure ConvertToSequence*

Input: $[x_{i,j}^0]_{N \times N}$.

Output: A closed walk, represented by a collection C of pairs (i, c) , where i is an integer and c a simple cycle.

1. $C := \emptyset, n = 1$.
2. Find a cycle $c_n = ((i_1, i_2), (i_2, i_3), \dots, (i_t, i_1))$, $i_k \in J$, $k \in 1, 2, \dots, t$, such that $x_{i_k, i_{k+1}}^{n-1} > 0$, $k \in \{1, 2, \dots, t-1\}$ and $x_{i_t, i_1}^{n-1} > 0$.
3. Let $m_n = \min\{x_{i,j}^{n-1} | (i, j) \in c_n\}$, $C := C \cup \{m_n, c_n\}$ and $x_{i,j}^n := x_{i,j}^{n-1} - m_n$ for $(i, j) \in c_n$ and $x_{i,j}^n := x_{i,j}^{n-1}$ otherwise.
4. If $[x_{i,j}^n]_{N \times N} = [0]_{N \times N}$, output: C and stop, else set $n := n+1$, and goto step 2.

The reader may verify that this algorithm can be read as an high multiplicity extension of the text book algorithm to find a eulerian trail in a eulerian graph in the following way. We construct a eulerian walk by traversing the first edge i_1, i_2 of the first simple cycle c_1 . Then, we check whether city i_2 is the first city in any other cycle c_l . If this is the case we continue by recursively traversing c_l . Otherwise, or after having recursively traversed c_l , we continue by traversing the second edge of c_1 , i.e. (i_2, i_3) . For city i_3 , again we look whether there is some simple, and yet untraversed cycle c_k that starts in i_3 . If so, we start traversing c_k , and otherwise, or after having recursively traversed c_k , we continue traversing c_1 . When c_1 is completely traversed, i.e. after having traversed the arc (i_t, i_1) , and recursively all cycles starting at i_1 , we finish by traversing c_1 $m_1 - 1$ times without making recursive traversals. In general, when during the construction of the eulerian walk some simple cycle c_k is completely traversed, we traverse it $m_k - 1$ times, without making the recursive traversals.

The graph $G(J, A)$ is a eulerian graph, and therefore must contain a eulerian walk. For the correctness of the procedure *ConvertToSequence* we refer to Eiselt et al. (1995)). A few words on its complexity are in order here. First of all, it is not hard to verify that step 2 can be executed in $O(N)$ time. Further, $x_{i,j}^n$ contains at least one more zero element than $x_{i,j}^{n-1}$, by definition of step 3. Hence steps 2 to 4 are executed $O(N^2)$ times. This leaves the overall time complexity of *ConvertToSequence* to be $O(N^3)$. In addition, this reasoning implies that the collection C consists of at most $O(N^2)$ pairs (i, c) . Since i and c are also polynomially bounded in the input size, the output of it *ConvertToSequence* is a polynomial encoding of an optimal solution. Hence we have two compact encoding schemes of optimal solutions: C , and $x_{i,j}^0$. In subsequent sections these compact encoding schemes will be contrasted with not necessarily polynomial encoding schemes in which solutions are more explicitly specified. An example of such an encoding scheme is to disregard the multiplicities and completely specify the sequence in which the cities are to be visited.

We now continue by considering some relaxations and extensions of Definition 2.1. For a generic instance we define T to be the problem that results from P by relaxing the subtour elimination constraints (iv):

Definition 2.3

$$\begin{aligned}
T : \quad & \min_x \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j} \\
\text{subject to} \quad & \sum_{i \in J} x_{i,j} = k_j, \quad j \in J; \\
& \sum_{j \in J} x_{i,j} = k_i, \quad i \in J; \\
& x_{ij} \in \mathbf{Z}_+, \quad i \in J, j \in J.
\end{aligned}$$

We denote the value of the optimal solution of T by $F(T)$. Problem T is a transportation problem, and therefore always has an integral optimal solution that can be found in polynomial time.

Finally, we introduce a more general problem formulation of the cycle time minimization problem. Since the research aims to find input sequences that yield maximum throughput, we propose a formulation that allows more general input sequences, not just input sequences for MPSs:

Definition 2.4

$$\begin{aligned}
P(l) : \quad & \min_{x,l} \frac{1}{l} \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j} \\
\text{subject to} \quad & \sum_{i \in J} x_{i,j} = l \times k_j, \quad j \in J; \\
& \sum_{j \in J} x_{i,j} = l \times k_i, \quad i \in J; \\
& \sum_{i \in J'} \sum_{j \in J \setminus J'} x_{i,j} \geq 1, \quad \forall J' \subset J \text{ such that } J' \neq \emptyset; \\
& x_{i,j} \in \mathbf{Z}_+, \quad i \in J, j \in J; \\
& l \in \mathbf{Z}_{++}.
\end{aligned}$$

Problem $P(l)$ contains the decision variable l . Let (l^*, x^*) be some optimal solution, then it encodes a sequence in which each city $j \in J$ is visited $\sum_{i \in J} x_{i,j}^* = l^* \times k_j$ times. Hence the resulting sequence contains l^* MPSs. Since the aim is to maximize throughput rate, this formulation minimizes the minimum cycle time achievable by a sequence for the jobs in l MPSs, over all natural numbers l . Thus, in a scheduling context, and letting l be infinity, the optimal solution value specifies the maximum throughput rate attainable while producing all types in the prespecified ratios.

We shall again refer to the objective function of $P(l)$ by $F(P(l))$, or $F(l)$ for short. Notice that $P(l)$ is not a linear program. However, for fixed l it is, and as before $P(l)$ can be solved in polynomial time when both l and N are fixed Lenstra (1983).

As before, we define $T(l)$ to be the problem that results from $P(l)$ after relaxing the subtour elimination constraints:

Definition 2.5

$$\begin{aligned}
F(T(l)) &= \frac{1}{l} \min_x \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j} \\
\text{subject to} \quad & \sum_{i \in J} x_{i,j} = l \times k_j, & j \in J; \\
& \sum_{j \in J} x_{i,j} = l \times k_i, & i \in J; \\
& x_{i,j} \in \mathbf{Z}_+, & i \in J, j \in J.
\end{aligned}$$

Notice that by defining $y_{i,j} = x_{i,j}/l$, $T(l)$ can be rewritten as

$$\begin{aligned}
F(T(l)) &= \min_x \sum_{i \in J} \sum_{j \in J} c_{i,j} y_{i,j} \\
\text{subject to} \quad & \sum_{i \in J} y_{i,j} = k_j, & j \in J; \\
& \sum_{j \in J} y_{i,j} = k_i, & i \in J; \\
& y_{i,j} \in \mathbf{Z}_+, & i \in J, j \in J.
\end{aligned}$$

Hence, we conclude that $T(l)$ and T are identical transportation problems. By consequence for any natural number l the value $F(T)$ yields a lower bound for $F(l)$.

3 General Properties of Optimal Solutions

In this section we derive some basic properties of the problem $P(l)$. We derive some results on how the optimal value $F(l)$ decreases when l increases. All of these results are based on the following inequality:

Theorem 3.1 *For any natural number l , the following inequality holds*

$$F(l+1) \leq \frac{l}{l+1} F(l) + \frac{1}{l+1} F(T). \quad (1)$$

Proof. Let x^{l+1} and $x^l, l \in \mathbf{Z}_{++}$, denote optimal solutions for $P(l+1)$ and $P(l)$ respectively, and let x^T denote an optimal solution for T . Now, notice first that the vector $x^l + x^T$ is a feasible solution for $P(l+1)$. Thus, for any number $l \in \mathbf{Z}_{++}$

we have

$$\begin{aligned}
F(l+1) &= \frac{1}{l+1} \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j}^{l+1} \\
&\leq \frac{1}{l+1} \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j}^l + \frac{1}{l+1} \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j}^T \\
&= \frac{l}{l+1} \left(\frac{1}{l} \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j}^l \right) + \frac{1}{l+1} \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j}^T \\
&= \frac{l}{l+1} F(l) + \frac{1}{l+1} F(T)
\end{aligned}$$

as required. \square

Corollary 3.1 $F(l) \geq F(l+1)$ holds for any $l \in \mathbf{Z}_{++}$.

Proof. Using Theorem 3.1 we straightforwardly derive,

$$\begin{aligned}
F(l+1) &\leq \frac{l}{l+1} F(l) + \frac{1}{l+1} F(T) \\
&= \frac{l}{l+1} F(l) + \frac{1}{l+1} F(T(l)) \\
&\leq \frac{l}{l+1} F(l) + \frac{1}{l+1} F(l) \\
&= F(l),
\end{aligned}$$

where the second inequality follows from the fact that $T(l)$ is obtained from $P(l)$ by relaxation of the sub-tour elimination constraints. \square

Theorem 3.2 $F(l) > F(l+1)$ if and only if $F(l) > F(T)$.

Proof. If $F(l) > F(T)$, then it must hold that

$$\begin{aligned}
F(l) &= \frac{l}{l+1} F(l) + \frac{1}{l+1} F(l) \\
&> \frac{l}{l+1} F(l) + \frac{1}{l+1} F(T) \\
&\geq F(l+1)
\end{aligned}$$

by 3.1. Conversely, if $F(l) = F(T)$, it must hold that

$$\begin{aligned}
F(l+1) &\geq F(T(l+1)) \\
&= F(T) \\
&= F(l).
\end{aligned}$$

Combined with 3.1 this yields that $F(l+1) = F(l)$. \square

By consequence we also have the following corollary:

Corollary 3.2 *If there exists l^0 such that $F(l^0) = F(T)$ then $F(l) = F(T)$ for any number $l \geq l^0$.*

Hence, the question rises whether for a given instance there exists l^0 such that $F(l^0) = F(T)$? Moreover, in case the answer is affirmative, one would like to efficiently compute or at least estimate a smallest number l^0 such that the equality holds. These questions will be discussed and answered in the Sections 4 and 5. In case the answer is negative, there might still be some long run stabilization worthy of characterization (which in turn might be efficiently computable).

Pursuing this direction, we first consider the following theorem, which is an extension of Theorem 3.1.

Theorem 3.3 *For any fixed natural number l^* and any natural number $l \geq l^*$ the following inequality holds*

$$F(l) \leq \frac{l^*}{l} F(l^*) + \frac{l - l^*}{l} F(T). \quad (2)$$

Proof. We shall prove by induction on l from the basis l^* . For any natural number l^* it holds that $F(l^*) \leq l^* \times F(l^*)/l^* + (l^* - l^*) \times F(T)/l^* = F(l^*)$, and hence 2 holds for $l = l^*$.

Now, suppose that for some number $l \geq l^*$ property 2 holds. We prove that based on this induction hypothesis the validity of this property for $l+1$ can be derived. From 3.1 we obtain, for every natural number l , $F(l+1) \leq l \times F(l)/(l+1) + F(T)/(l+1)$. Hence, by induction we have that

$$\begin{aligned} F(l+1) &\leq \frac{l}{l+1} F(l) + \frac{1}{l+1} F(T) \\ &\leq \frac{l}{l+1} \left(\frac{l^*}{l} F(l^*) + \frac{l - l^*}{l} F(T) \right) + \frac{1}{l+1} F(T) \\ &= \frac{l^*}{l+1} F(l^*) + \frac{l+1 - l^*}{l+1} F(T) \end{aligned}$$

and the proof is complete. \square

Finally, we prove that the value $F(l)$ converges to $F(T)$ when l goes to infinity.

Theorem 3.4 *If there exists a (finite) natural number l^* such that $F(l^*) < \infty$ then*

$$\lim_{l \rightarrow \infty} F(l) = F(T). \quad (3)$$

Proof. Using Theorem 3.3, and letting $l > l^*$ we derive

$$\begin{aligned}
F(T) &\leq F(l) \\
&\leq \frac{l^*}{l}F(l^*) + \frac{l-l^*}{l}F(T) \\
&= F(T) + \frac{l^*}{l}F(l^*) - \frac{l^*}{l}F(T).
\end{aligned}$$

Hence,

$$\begin{aligned}
F(T) &= \lim_{l \rightarrow \infty} F(l) \\
&\leq \lim_{l \rightarrow \infty} \{F(T) + \frac{l^*}{l}F(l^*) - \frac{l^*}{l}F(T)\} \\
&= F(T).
\end{aligned}$$

which yields the desired result. \square

In Section 5 we obtain a much stronger result which improves Theorems 3.1 and 3.3. Namely, we shall find that for a very general class of instances, and for all $l \geq N - 1$, it holds that $F(l) = F(N - 1) \times (N - 1)/l + F(T) \times (l - N + 1)/l$.

We finish this section by demonstrating how optimal solutions and objective function values change with l .

Example 3.1 *Consider the following instance:*

Let $J = \{1, 2, 3\}$;
 $k_1 = 1, k_2 = 1, k_3 = 1$;
 $c_{1,3} = c_{3,1} = a, (a > 1)$;
 $c_{i,j} = 1$ for $(i, j) \notin \{(1, 3), (3, 1)\}$.

The reader is encouraged to verify that

- *For the transportation problem T it holds that $F(T) = 3$, $x_{i,j}^T = 0$ for $i \neq j$ and $x_{i,j}^T = 1$ for $i = j$. The collection C as output by `ConvertToSequencein` this case is $C = \{(1, (1, 1)), (1, (2, 2)), (1, (3, 3))\}$.*
- *For the problem $P(1)$ we have $F(P) = a + 2$. An optimal solution for this problem is $x_{i,j}^1 = 1$ for $(i, j) \in \{(1, 2), (2, 3), (3, 1)\}$ and $x_{i,j}^1 = 0$ for any other arcs. In this case is : $C = (1, ((1, 2), (2, 3), (3, 1)))$.*
- *For any natural number $l \geq 2$ we get $F(l) = 3$. An optimal solution is specified by $x_{1,2}^l = x_{2,3}^l = x_{3,2}^l = x_{2,1}^l = 1$; $x_{1,1}^l = x_{3,3}^l = l - 1$; $x_{3,1}^l = x_{1,3}^l = 0$ and $x_{2,2}^l = l - 2$. Further, $C = ((1, (1, 2)(2, 3), (3, 2), (2, 1)), ((l - 1), (1, 1)), ((l - 2), (2, 2)), ((l - 1), (3, 3)))$.*

In this example it holds that $F(l) = F(T)$, for $l \geq 2$. This example also demonstrates that the ratio between $F(P)$ and $F(l)$ can be extremely bad, namely $F(P)/F(l) = (a + 2)/3$, where a may be chosen arbitrarily.

Example 3.2 Now, consider another instance which is identical to the one described in Example 3.1, but for the following arc lengths: $c_{1,2} = c_{2,1} = b$, where $1 < b < a$.

It can be checked that in all cases discussed in Example 3.1 the same solutions are optimal ones, but the objective function values are different. More specifically,

- For the transportation problem T it holds that $F(T) = 3$, $x_{i,j}^T = 0$ for $i \neq j$ and $x_{i,j}^T = 1$ for $i = j$. The collection C as output by *ConvertToSequence* in this case is $C = \{(1, (1, 1)), (1, (2, 2)), (1, (3, 3))\}$.
- For the problem $P(1)$ we have $F(P) = a + b + 1$. An optimal solution for this problem is $x_{i,j}^1 = 1$ for $(i, j) \in \{(1, 2), (2, 3), (3, 1)\}$ and $x_{i,j}^1 = 0$ for any other arcs. In this case is : $C = ((1, (1, 2)), (1, (2, 3)), (1, (3, 1)))$.
- For any natural number $l \geq 2$ we get $F(l) = (3l + 2b - 2)/l = 3 + (2b - 2)/l$. An optimal solution is again specified by $x_{1,2}^l = x_{2,3}^l = x_{3,2}^l = x_{2,1}^l = 1$; $x_{1,1}^l = x_{3,3}^l = l - 1$; $x_{3,1}^l = x_{1,3}^l = 0$ and $x_{2,2}^l = l - 2$. Further, $C = ((1, (1, 2)(2, 3), (3, 2), (2, 1)), ((l - 1), (1, 1)), ((l - 2), (2, 2)), ((l - 1), (3, 3)))$.

In this example, $F(l)$ strictly decreases when l increases, and indeed there doesn't exist l^0 such that $F(l^0) = F(T)$. Nevertheless, the optimal solution displays the same (stable) behavior as in the previous example.

4 Optimal Sequences of Bounded Length in the Stable Case

Definition 4.1 Consider an instance for which there exist a number $l^0 \in \mathbf{Z}_{++}$ such that $F(l) > F(T)$ for any $l < l^0$ and $F(l) = F(T)$ for any $l \geq l^0$. Such an instance is called *stable* and l^0 is called the *stabilization number*.

Theorem 4.1 For every stable instance, the stabilization number $l^0 \leq (N + 1)^2/4$.

Proof. Consider a stable instance and corresponding stabilization number l^0 . Let $M = (J, A)$ be a complete directed multigraph on the set of nodes J . In M let us consider a closed walk $W^0 = ((i_1, i_2), (i_2, i_3), \dots, (i_t, i_1))$, $i_k \in J$, $k \in \{1, \dots, t\}$, which represents an optimal solution for the problem $P(l^0)$. Call to mind that any closed walk which represents a feasible solution of the problem $P(l)$ passes through all cities $j \in J$ exactly $l \times k_j$ times.

In addition, let $S = (s_1, s_2, \dots, s_w)$, $s_k \in A$, $k \in \{1, \dots, w\}$ be any closed walk in the graph M . Recall that by definition, for a closed walk it holds that there exists i, j, k such that $s_1 = (i, j)$ and $s_w = (k, i)$. We say closed walk S is *complete* if it passes through all vertices in G . More formally, we define:

Definition 4.2 *Given a multigraph $M(J, A)$ and a closed walk $S = (s_1, s_2, \dots, s_w)$, $s_k \in A$, $k \in \{1, \dots, w\}$ in M . We say S is complete if for every $j \in J$ there exists $k \in \{1, \dots, w\}$ and $i \in J$ for which $(i, j) = s_k$.*

Definition 4.3 *Given a multigraph $M(J, A)$, closed walk $S = (s_1, s_2, \dots, s_w)$, $s_k \in A$, $k \in \{1, \dots, w\}$ in M , let $S' = (s_i, s_{i+1}, \dots, s_{j-1}, s_j)$, where $1 \leq i < j \leq w$ be a closed walk which is contained as a subsequence in S . We say S is a minimal complete walk if, for every $S' = (s_i, s_{i+1}, \dots, s_{j-1}, s_j)$, where $1 \leq i < j \leq w$, the closed walk $S \setminus S' = (s_1, s_2, \dots, s_{i-1}, s_{j+1}, \dots, s_w)$ is not complete.*

If closed walk S' is contained as a subsequence in S and $S \setminus S'$ is complete, we call S' ommissable.

Consider a minimal complete walk $W' = ((j_1, j_2), (j_2, j_3), \dots, (j_t, j_{t+1}))$ where $j_1 = j_{t+1}$ which is obtained from the complete closed walk W^0 by removing ommissable closed walks from it as described above. We are going to prove that W' contains at most $(N+1)^2/4$ arcs. To see this, let f be the number of vertices that are passed in W' exactly once. More formally, f is the number of vertices j for which there is exactly one arc $(j_k, j_{k+1}), k \in 1, \dots, t$ in W' such that $j_k = j$.

First, we make clear that $f \geq 2$. Consider an arbitrary cycle C that is contained as a subsequence in W' . If no such cycle C exists, we are done since W' is complete and hence $f = N$. Thus let us assume C exists. Then C (and also $W' \setminus C$) consists of at least two arcs, since otherwise it forms a loop, and therefore is ommissable, which contradicts the minimality of W' .

Now, let C_1 be a cycle that is contained as a subsequence in C with minimum number of arcs and let C_2 be defined analogously with respect to $W' \setminus C$. There are two cases for C_1 .

Assume first that all vertices passed in the cycle C_1 are represented in W' more than once. If all vertices passed in C_1 are passed also in $W' \setminus C_1$, C_1 is ommissable contradicting the minimality of W' . Hence at least one vertex must be passed more than once in C_1 . Thus C_1 contains a cycle. Again we have reached a contradiction since by definition of C_1 it doesn't contain any cycles contained in it as a subsequence. Thus, we conclude that there is at least one vertex passed in C_1 which is passed in W' only once. Applying the same reasoning to C_2 , we conclude that $f \geq 2$.

Now let i and j be two vertices that are passed once in W' and such that the part Q of the closed walk from i to j doesn't contain any other vertex that is

passed once in W' . (We will say that i and j are neighboring.) It can be seen as follows that in Q no vertex is passed more than once. If there exist a vertex that is passed twice in Q we have found a cycle. By definition of i, j and Q , this cycle doesn't contain vertices that are passed once in W' . Hence, the cycle must be ommissable contradicting the minimality of W' . Thus all vertices that are passed in Q are passed once in Q .

Now we know that in the walk there are $f \geq 2$ vertices which are passed exactly once and that between pairs of neighboring such vertices, no other vertices occur more than once. This allows us to bound the total number of vertices in W' . Indeed, it must hold that the total number of vertices and hence arcs in W' does not exceed $f + f(N - f)$. Maximizing $f + f(N - f)$ over $f = 2, \dots, N$ yields $f_{\max} = (N + 1)/2$. It then follows that $f_{\max} + f_{\max}(N - f_{\max}) = (N + 1)^2/4$ and therefore W' has length at most $(N + 1)^2/4$.

Using that the length t of W' is less than or equal to $(N + 1)^2/4$, we now finish the proof by constructing a solution to $P(t)$ such that $F(t) = F(T)$.

Since we consider a stable instance, by definition $F(l^0) = F(T)$. Therefore the walk W^0 is also an optimal solution for the problem $T(l^0)$. Let $x_{i,j}^0$ be the number of times arc (i, j) is traveled in W^0 , for all $i, j \in J$. Then, $x_{i,j}^0/l^0$ is a fractional optimal solution for problem T . Since the transportation problem has a totally unimodular constraint matrix, the polyhedron of problem T is integral. By consequence, any fractional optimal solution can be represented as a convex combination of integer optimal solutions. Hence, we have that for every arc (i, j) contained in the walk W^0 , there exists an integer optimal solution for T for which $x_{i,j} > 0$.

Now let us consider the walk W^0 and a corresponding minimal complete closed walk W' . Recall that for every arc $(i, j) \in W'$ there exists a solution $x^{i,j}$ of the problem T which contains this arc. Let x' be obtained by summing all $x^{i,j}$, $(i, j) \in W'$. Since W' is a complete walk, x' satisfies the subtour elimination constraints, and hence it is an optimal solution to $P(t)$.

Since x' is obtained by summing t solutions for problem T , x' is a solution for $P(t)$. Therefore $F(t) = F(T)$ and thus $t \leq l^0$. It then follows that $l^0 \leq t \leq (N + 1)^2/4$ as required. \square

With Theorem 4.1 at hand, we are now able to address a computational complexity issue. Despite the fact that even for fixed l the decision version of $P(l)$ is a generalization of the TSP and therefore NP -complete, we have the following theorem:

Theorem 4.2 *The problem of deciding whether an instance is stable can be solved in polynomial time.*

Proof. Firstly, we are going to check for each arc $(i', j') \in J \times J$ whether there exists an optimal solution of T that contains this arc. This can be achieved by

solving a modification of T , where the modification consists of decreasing $k_{i'}$ by one in the outflow constraint (iii) for city i' , and decreasing $k_{j'}$ by one in the inflow constraint (ii) for city j' . Let us call this modified problem $T_{i',j'}$. Obviously, $T_{i',j'}$ is in turn a transportation problem that can be easily solved in polynomial time. The aforementioned optimal solution exists if and only if $F(T) = F(T_{i',j'}) + c_{i'j'}$.

We subsequently construct a directed multigraph $D = (J, A)$ as follows. For all (i', j') such that $F(T) = F(T_{i',j'}) + c_{i'j'}$, let $x^{i'j'}$ be obtained by slightly modifying an optimal solution to $T_{i',j'}$. The modification consists of increasing $x_{i'j'}^{i'j'}$ by one. Now, the arc set A of D is obtained as follows. For all (i, j) , the multiplicity of arc (i, j) is defined to be the sum of the $x_{ij}^{i'j'}$ over all (i', j') for which $F(T) = F(T_{i',j'}) + c_{i'j'}$.

Now, if D is strongly connected then it contains a Eulerian tour (see e.g. Eiselt et al. (1995)), i.e. a closed walk visiting every arc in this directed multigraph. Clearly this tour visits every vertex as well and hence satisfies the subtour elimination constraints. But then this Eulerian tour contains a solution to $P(l)$ with value $l \times F(T)/l = F(T)$.

Conversely, suppose that D is not strongly connected. Thus, there exists a subset $J' \subset J, J' \neq \emptyset$, such that D doesn't contain any arc (i, j) such that $i \in J', j \in J \setminus J'$. Now, for contradiction, assume that there is some number l such that $F(l) = F(T)$ and let x^l be an optimal solution to $P(l)$. Define $y_{i,j}^l = x_{i,j}^l/l, (i, j) \in J \times J$. Note, that y^l is an optimal solution for T . Notice also, that x^l is a feasible solution to $P(l)$ and hence satisfies the subtour elimination constraints. Then for J' there must exist an arc (i, j) such that $i \in J', j \in J \setminus J'$, and $y_{i,j}^l > 0$. Then, in some optimal solution z^l of the transportation problem $T(l)$ it holds that $z_{ij}^l > 0$. By total unimodularity, there must exist an optimal solution z^0 for T in which $z_{i,j}^0$ is a positive integer. But then, by definition of A , it holds that $(i, j) \in A$. Since arc (i, j) connects J' and $J \setminus J'$ we have arrived at a contradiction.

Thus, we conclude that an instance I is stable if and only if the directed multigraph D is strongly connected. As described above, D can be constructed in polynomial time. Further, it is easy to check in polynomial time, whether D is strongly connected (see e.g. Ahuja and Orlin (1994)), which completes the proof. \square

We finish this section by giving a tight example for the bound derived in Theorem 4.1.

Example 4.1 Consider the following instance. $J = \{1, 2, 3, 4, 5, 6\}$, and

$$C = \begin{pmatrix} a & a & a & +1 & a & a \\ a & a & a & a & +1 & a \\ a & a & a & a & a & +1 \\ a & a & +1 & +1 & +1 & +1 \\ +1 & a & a & +1 & +1 & +1 \\ a & +1 & a & +1 & +1 & +1 \end{pmatrix}$$

where $a > 1$, $k^T = (1, 1, 1, 3, 3, 3)$. This instance has minimal complete closed walk and also optimal sequence $[1, 4, 5, 6, 2, 5, 6, 4, 3, 6, 4, 5]$:

It is straightforward to generalize example 4, in particular the distance matrix C to the case where $J = \{1, \dots, n\}$ and a corresponding minimal complete closed walk W' of length $(n+1)^2/4$ exists that entails the optimum solution value. let it be noted however that this doesn't mean that every optimal solution necessarily results in a minimal closed walk of length $(n+1)^2/4$. Indeed we shall see in the next Section, that a better bound can be obtained when using linear programming techniques. Nevertheless, if we only encode solutions as sequences and consider resulting closed walks, Theorem 4.1 is tight. Hence the section title. Indeed, the next section goes to show that it pays to consider and encode solutions for this high multiplicity sequencing problem in a different way then by explicit sequences.

5 Optimal Solutions of Bounded Length

For further investigations it will be convenient to define for the given vector $\delta \in \mathbf{Z}_+^N$ the following transportation problem $T(l, \delta)$

$$\begin{aligned} F(T(l, \delta)) &= \min_x \sum_{i \in J} \sum_{j \in J} c_{i,j} x_{i,j} \\ \text{subject to} \quad & \sum_{i \in J} x_{i,j} = l \times k_j - \delta_j, & j \in J; \\ & \sum_{j \in J} x_{i,j} = l \times k_i - \delta_i, & i \in J; \\ & x_{i,j} \in \mathbf{Z}_+, & i \in J, j \in J. \end{aligned}$$

Theorem 5.1 Consider an instance of the problem such that the following holds simultaneously

- $F(N-1)$ is finite;

- The problem $T(N-1, \alpha)$ has a finite optimal solution where the i -th component from the vector $\alpha \in \mathbf{Z}_+^N$ represents the number of times city i is passed in a minimal complete closed walk obtained from an optimal solution of the problem $P(N-1)$ (see theorem 4.1).

Then, for any natural number $l \geq N-1$ the following holds:

$$F(l+1) = \frac{l}{l+1}F(l) + \frac{1}{l+1}F(T) \quad (4)$$

and

$$F(l) = F(N-1)\frac{N-1}{l} + F(T)\frac{l-N+1}{l} \quad (5)$$

and the solution $x^l = x^{N-1} + (l-N+1)x^T$ is an optimal solution of the problem $P(l)$.

Proof. To prove the theorem we have to show that equation 4 holds. Equation 5 then follows by induction as in Theorem 3.3.

To prove equation 4, we show that $(l+1)F(l+1) \geq lF(l) + F(T)$ for any natural number $l \geq N-1$. Together with Theorem 3.1 this yields the desired result.

Consider an optimal solution x^0 of the problem $P(l+1)$, $l \geq N-1$. As before, let W^0 be a corresponding closed walk, and let W' be a minimal complete closed walk obtained from W^0 by deleting ommissable cycles. Let α be the vector such that α_i is the number of times city i is passed in W' . It can be seen that since W' is minimal and complete, $1 \leq \alpha_i \leq N-1$ for all $i \in J$.

Now consider a solution x^l of $P(l)$ which is constructed as follows: $x^l = x^{W'} + x^{T(l, \alpha)}$ where $x_{ij}^{W'}$ ($i \in J, j \in J$) equals the number of times arc (i, j) occurs in W' and $x^{T(l, \alpha)}$ is a solution of the problem $T(l, \alpha)$. Since $l \times k_i \geq (N-1) \times k_i \geq N-1 \geq \alpha_i$ for any index $i \in J$, we may conclude that W' is a feasible complete tour for the problem $P(l)$. Thus, the solution x^l is feasible for $P(l)$ and the transportation problem $T(l, \alpha)$ is correctly defined with strictly positive right hand-side coefficients $l \times k - \alpha$. Since x^l is a feasible solution of $P(l)$, we have

$$\begin{aligned} & l \times F(l) + F(T) \\ & \leq \sum_{(i,j) \in W'} c_{i,j} x_{i,j} + l \times F(T(l, \alpha)) + F(T) \\ & = \sum_{(i,j) \in W'} c_{i,j} x_{i,j} + (l+1) \times F(T(l+1, \alpha)) + l \times F(T(l, \alpha)) + F(T) \\ & \quad - (l+1) \times F(T(l+1, \alpha)) \\ & = F(l+1) + l \times F(T(l, \alpha)) + F(T) - (l+1) \times F(T(l+1, \alpha)). \end{aligned}$$

Hence, we can prove equation 4 by proving

$$l \times F(T(l, \alpha)) + F(T) = (l + 1) \times F(T(l + 1, \alpha)) \text{ for any } l \geq N - 1 \text{ (*)}$$

Since the polyhedron of the transportation problem is integer, we can assume without loss of generality that $F(T)$, $F(T(l, \alpha))$ and $F(T(l + 1, \alpha))$ are linear programs. Further, changing notation from x to x/l yields that instead of proving that $l \times F(T(l, \alpha)) + F(T) = (l + 1) \times F(T(l + 1, \alpha))$ it suffices to prove that $l \times F(T(1, \alpha/l)) + F(T) = (l + 1) \times F(T(1, \alpha/(l + 1)))$. The reader is encouraged to verify that the three linear programs $T(1, \alpha/l)$, T , $T(1, \alpha/(l + 1))$ only differ in the right hand sides. Let it be noted also that T is by definition identical to $T(1, 0)$, and hence $F(T)$ equals $F(T(1, 0))$.

Let us denote the dual of $T(1, \alpha/l)$ as $DT(1, \alpha/l)$. It can be described as follows:

$$\begin{aligned} DT(1, \frac{\alpha}{l}) : \max_{u, v} & \left(\sum_{i \in J} (k_i - \frac{\alpha_i}{l}) u_i + \sum_{j \in J} (k_j - \frac{\alpha_j}{l}) v_j \right) \\ \text{subject to} & \\ u_i + v_j & \leq c_{i,j}, \quad i \in J, j \in J. \end{aligned}$$

We simply write DT for $DT(1, 0)$, and indeed DT is the dual of T . Since the three problems $T(1, \alpha/l)$, T , $T(1, \alpha/(l + 1))$ only differ in the right hand sides, the three problems $DT(1, \alpha/l)$, DT , $DT(1, \alpha/(l + 1))$ only differ in the coefficients of the objective function.

Now, suppose that despite the differences in the coefficients of the objective function, some optimal solution (u^*, v^*) of DT is also optimal for $DT(1, \alpha/l)$ and $DT(1, \alpha/(l + 1))$. Then, by strong duality, we have

$$\begin{aligned} & (l + 1) \times F(T(1, \frac{\alpha}{l + 1})) \\ = & (l + 1) \times \left(\sum_{i \in J} (k_i - \frac{\alpha_i}{l + 1}) u_i^* + \sum_{j \in J} (k_j - \frac{\alpha_j}{l + 1}) v_j^* \right) \\ = & \sum_{i \in J} ((l + 1) \times k_i - \alpha_i) u_i^* + \sum_{j \in J} ((l + 1) \times k_j - \alpha_j) v_j^* \\ = & + \sum_{i \in J} (l \times k_i - \alpha_i) u_i^* + \sum_{j \in J} (l \times k_j - \alpha_j) v_j^* + \sum_{i \in J} k_i u_i^* + \sum_{j \in J} k_j v_j^* \\ = & l \times \left(\sum_{i \in J} (k_i - \frac{\alpha_i}{l}) u_i^* + \sum_{j \in J} (k_j - \frac{\alpha_j}{l}) v_j^* \right) + \left(\sum_{i \in J} k_i u_i^* + \sum_{j \in J} k_j v_j^* \right) \\ = & l \times F(T(1, \frac{\alpha}{l})) + F(T) \end{aligned}$$

which proves equation (*) and hence equation 4 as required to prove the theorem.

Thus, intuitively, we have to show that for l large enough, or more specifically $l \geq N - 1$, the differences in the coefficients of the objective function between the problems DT , $DT(1, \alpha/l)$ and $DT(1, \alpha/(l+1))$. are small enough to yield a solution (u^*, v^*) that is optimal for all three of DT , $DT(1, \alpha/l)$ and $DT(1, \alpha/(l+1))$. Continuing in this intuitive fashion, such a solution (u^*, v^*) can be found as follows. Consider the polyhedron of DT . Let us make some small modifications to the objective function, and hence to the objective hyperplane. Let's say these modifications are obtained by adjusting the orthogonal vector by a small vector σ . It is clear that if optimal face of the polyhedron is bounded in the σ -direction, then we preserve at least one optimal point of this face. By the second condition of the theorem, $F(T(N-1, \alpha))$ is finite. It then follows that in the direction α the optimal face is bounded. Therefore with a small adjustment of the orthogonal vector in the α -direction, we keep at least one dual optimal solution (u^*, v^*) of T for the problems $DT(1, \alpha/l)$ for any sufficiently big number $l \in \mathbf{Z}_+$.

More formally, the equation (*) can be proved using the following theorem from sensitivity analysis:

Theorem 5.2 Partition Perturbation Theorem (see, e.g. Greenberg (1997)). *Consider linear program $LP(b) = \min\{cx | x \geq 0, Ax \geq b\}$ and its dual $DLP(b) = \max\{\pi b | \pi \geq 0, \pi A \leq c\}$. Suppose (x^0, π^0) is a pair of strictly complementary solutions for $LP(b)$ and $DLP(b)$ respectively, and γ is an admissible direction, i.e. $LP(b + \theta\gamma)$ has an optimal solution for some $\theta > 0$. Define the differential linear program: $\Delta(b) = \max\{\pi\gamma | \pi \text{ is optimal solution of } DLP(b)\}$. Then there exists $\theta^* > 0$ such that the following holds true for $\theta \in [0, \theta^*]$: the optimal partition for $LP(b + \theta\gamma)$ or, equivalently, $DLP(b + \theta\gamma)$ is the same as the optimal partition for $\Delta(b)$.*

By the conditions of Theorem 5.1 $F(T(N-1, \alpha))$ and hence $F(T(1, \alpha/(N-1)))$ are finite and therefore α, α is an admissible direction.

Let (u^*, v^*) be the optimal solution of DT that is also optimal for $\Delta(b) = \max\{(u, v)^T, (\alpha, \alpha) | (u, v) \text{ is an optimal solution of } DT\}$. Then, by the Partition Perturbation Theorem, there exists $\theta^* > 0$ such that for any $\theta \in [0, \theta^*]$ the optimal partition of $T(1, \theta\alpha)$ (and hence $DT(1, \theta\alpha)$) is the same as the optimal partition for T (and DT). This in turn implies that the optimal dual solution (u^*, v^*) of DT is also optimal for $DT(1, \theta\alpha)$ for any $\theta \in [0, \theta^*]$. By consequence, we can complete the proof by showing that $\theta^* \geq 1/l$.

To estimate θ^* for a general linear program LP , the following approach is proposed in Monteiro and Mehrotra (1996). Consider the system of linear equations $A_B z = \gamma$ where A_B denotes the submatrix of A whose columns correspond to the indices from $B = \{n | x_n > 0\}$. In Monteiro and Mehrotra (1996), the authors list the following two mutually exclusive cases:

- If the system has no solution, that is γ is not in the range of A_B , then there is not a number $\theta > 0$ that preserves the optimal partition.
- Assume that the system has a solution, say z , then $\theta^* = +\infty$ if $z \leq 0$ and $\theta^* \leq \min\{x_n/z_n | n \in B, z_n > 0\}$ otherwise.

Since we have chosen the conditions of the theorem such that an admissible direction exists, we are clearly in the second case. Thus, there exist a solution z^* of the system $Bz = \alpha^2$ where $\alpha^2 \in \mathbf{Z}^{2N}$ is the vector (α, α) . Letting x^* be an optimal solution to T , the value $\min\{x_{i,j}^*/z_{i,j}^* | x_{i,j}^* > 0; z_{i,j}^* > 0\}$ bounds θ^* from above. Since the polyhedron of P is integral $x_{i,j}^* \geq 1$ if $x_{i,j}^* > 0$. Hence, $\theta^* \geq 1/Z^*$, where $Z^* = \max\{z_{i,j}^* | z_{i,j}^* > 0\}$. Thus, it remains to bound $\max\{z_{i,j}^* | z_{i,j}^* > 0\}$.

Let us denote by B the base matrix of the linear program T , which is obtained from the constraint matrix of T by elimination of all columns corresponding to variables x_{ij}^* for which $x_{ij}^* = 0$. The problem $Bz = \alpha^2$ can be represented as the following problem on the graph. Given a graph $G = (J, E)$ where $E = \{(i, j) \in J \times J | x_{i,j}^* > 0\}$. The problem consists in finding a circulation flow such that every city $j \in J$ has inflow and outflow equal to α_j times. Negative components of the vector z are allowed, hence we may disregard the orientation of the $x_{i,j}$ in the edges. Consider the circulation flow z^* . Since every city $j \in J$ of the graph D has inflow and outflow α_j -times, $z_{i,j} \leq \alpha_j$ for all i, j . Hence, $z_{i,j} \leq \alpha_j \leq N - 1$ for all $i, j \in J$, and thus $Z^* \leq \max_{j \in J} \alpha_j$. But then, $\theta \geq 1/\max_{j \in J} \alpha_j \geq 1/(N - 1)$, and hence for $l \geq N - 1$ as stated in the theorem, $\theta^* \geq 1/l$ and the proof is complete. \square

Theorem 5.3 *For every instance of $P(l)$ with finite input data*

$$F(l) = F(N - 1) \frac{N - 1}{l} + F(T) \frac{l - N + 1}{l}$$

and the solution $x^l = x^{N-1} + (l - N + 1)x^T$ is optimal for $P(l)$.

Proof. All conditions of the theorem 5.1 are satisfied when the input data are finite. Indeed, if the objective coefficients $c_{i,j}$, $i \in J$, $j \in J$, and multiplicity coefficients k_j , $j \in J$, are finite, then the values $F(N - 1)$ and $F(T(N - 1, \alpha))$ are finite. \square

Theorem 5.4 *If an instance is stable then the stabilization number l^0 is such that $l^0 \leq N - 1$, and this bound is tight.*

Proof. According to Theorem 5.1, it holds for $l \geq N$ that $l \times F(l) = (l - 1) \times F(l - 1) + F(T)$ or equivalently $F(l - 1) = (l \times F(l) - F(T))/(l - 1)$. By stability, we have that there exists l such that $F(l) = F(T)$. Substituting $F(T)$

for $F(L)$ in the above equality we get $F(l-1) = F(T)$. Since by Theorem 5.1 the equality holds for any $l \geq N$ we have $F(N-1) = F(T)$ and immediately $l^0 \leq N-1$.

It remains to show that there exists instances for which $l^0 = N-1$. Consider the following instance, which is a straightforward extension of Example 3.1.

Example 5.1

Let $J = \{1, 2, \dots, N\}$, $k_i = 1$ for any $i \in J$ and the distance matrix is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & a & a & \dots & a & a \\ 1 & a & 1 & a & \dots & a & a \\ \vdots & & & \ddots & & & \\ 1 & a & \dots & a & 1 & a & a \\ 1 & a & \dots & a & a & 1 & a \\ 1 & a & \dots & a & a & a & 1 \end{pmatrix}$$

where $a > 1$.

- For the transportation problem $F(T) = N$, $x_{i,j}^T = 0$ for $i \neq j$ and $x_{i,j}^T = 1$ for $i = j$. An optimal collection $C = \{(1, (1, 1)), (1, (2, 2)), \dots, (1, (N, N))\}$.
- For $l = 1$, an optimal solution is given by $x_{i,i+1} = 1$ for $i = 1, \dots, N-1$, $x_{N,0} = 1$, and $x_{i,j} = 0$, $i, j \in J$ otherwise. Thus, the value of an optimal solution is in this case $F(1) = 1 + (N-2)a + 1 = 2 + (N-2)a$.
- For $l = 2$, an optimal solution is given by $x_{1,2} = x_{2,2} = x_{2,1} = 1$, $x_{1,3} = 1$, $x_{i,i+1} = 1$ for $i = 3, \dots, N-1$, and $x_{N,0} = 1$, $x_{i,i} = 1$, $i = 3, \dots, N$, and $x_{i,j} = 0$, $i, j \in J$ otherwise. Thus, in this case the value of an optimal solution equals $F(2) = (3 + 1 + (N-3)a + 1 + (N-2))/2 = ((N-3)a + N + 3)/2$,
- For any $1 < l < N-1$ an optimal solution $x_{i,j}^l$ ($i \in J$, $j \in J$) is

$$\begin{pmatrix} 1 & 1 & \dots & 1 & 1^* & 0 & \dots & 0 & 0 \\ 1 & l-1 & & 0 & 0^* & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots & & \dots & & \vdots \\ 1 & 0 & & l-1 & 0^* & 0 & \dots & 0 & 0 \\ 0^* & 0^* & \dots & 0^* & l-1^* & 1^* & & 0^* & 0^* \\ 0 & 0 & \dots & 0 & 0^* & l-1 & \ddots & 0 & 0 \\ \vdots & & \dots & & \vdots & & \ddots & \ddots & \\ 0 & 0 & \dots & 0 & 0^* & 0 & & l-1 & 1 \\ 1 & 0 & \dots & 0 & 0^* & 0 & \dots & 0 & l-1 \end{pmatrix}$$

where the ** superscripts the l -th row and l -th column of the matrix. The value of this optimal solution

$$\begin{aligned} F(l) &= \frac{(l-1)2 + (l-1)(l-1) + 1 + (N-(l+1))a + 1 + (N-l)(l-1)}{l} \\ &= \frac{(l-1)(N+1) + 2 + (N-l-1)a}{l} \end{aligned}$$

(where the equality can be derived by adding all $(l-1)$ terms and slightly rewriting the remainder).

- For $l = N - 1$ we get $F(N - 1) = N = F(T)$. An optimal solution is

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & l-1 & & \\ \vdots & & \ddots & \\ 1 & & & l-1 \end{pmatrix}$$

Thus, for this instance $l^0 \leq N - 1$. We finish the proof by showing that $l^0 \geq N - 1$. Indeed, for $l = N - 2$, which implies that $N \geq 3$,

$$\begin{aligned} F(N-2) &= \frac{(N-3)(2N-2) + 2 + (N-(N-2)-1)a}{N-2} \\ &= \frac{2N^2 - 8N + 6 + 2 + a}{N-2} \\ &= \frac{2(N-2)^2 + a}{N-2} \\ &= 2(N-2) + \frac{a}{N-2} \\ &\geq 2(N-2) + \frac{1}{N-2} \\ &\geq N + N - 4 + \frac{1}{N-2} \\ &\geq N \end{aligned}$$

where the last inequality holds since $N \geq 3$, which yields $l^0 > N - 2$ as required. \square

The previous instance also proves a final theorem of this section:

Theorem 5.5 *The bound of Theorem 5.1 is tight, i.e. there exist instances for which all conditions hold and $F(N - 2) > F(N - 1)$.*

6 Summary and Conclusions

This paper deals with a very general high multiplicity sequencing problem, namely the high multiplicity traveling salesman problem (HMTSP). The research is motivated by the applicability of the HMTSP to a variety of scheduling problems. Lately, high multiplicity scheduling problems have been widely studied, but very little is known about the impact of scheduling the jobs as specified by a minimal part set, as opposed to the more general scheduling problem of scheduling jobs in non-minimal, optimal, part sets. This paper investigates how the cycle time, or length of a HMTSP tour varies with the multiplicities in the (minimal) part set. In particular we investigate the behavior of the optimal solution, and the optimal solution value, when all quantities in the minimal part set are multiplied by a factor l . We show that the optimal solution value decreases when l increases. Moreover, we show that the ratio between the value of an optimal solution for $l = 1$ and for arbitrary l can grow proportional to l .

Further, we have investigated whether there exist finite l for which the optimal value cannot be improved upon. We have given a polynomial procedure that decides this problem in Section 4. In the same section we show that, if there exists finite l for which the optimal value is minimal, then this l can be bounded from above by $(N+1)^2/4$. In fact, we have shown that, if we restrict our encoding schemes to sequences which explicitly specify an order in which the cities are to be visited, then $(N+1)^2/4$ is a tight bound. However, in Section 5 we show that if the optimal solution is given in terms of a solution to an integer program, this bound can be improved to be $N-1$. Section 5 also shows that even if there is no finite l for which the optimal solution value is lowest attainable, the differences between the lengths of cyclic optimal schedules for l and $l+1$ are identical for all $l \geq N-1$.

The bounds on l establish the importance of encoding schemes for solutions, especially in the case of sequencing problems. Indeed, the results make clear that if the optimal solution is given in terms of an explicit sequence, then the tightest bound on l that can be derived is $O(N^2)$, whereas this bound can be improved upon to be $O(N)$ when using a more compact encoding scheme, which is based upon an integer programming formulation.

The linear bound obtained in this paper applies to all high multiplicity scheduling problems that can be formulated as a HMTSP. It will be interesting to see whether this result, as well as the ratio analysis, can be improved upon for specific problems.

Finally, we conjecture that the bound of Theorem 5.1 can be improved to be $(N-1)/\min_i k_i$.

7 Acknowledgements

The authors are grateful to Yves Crama and Nadia Brauner for useful discussions on high multiplicity issues. The authors thank Jos Sturm for providing references on sensitivity analysis.

References

- Agnetis, A. 1997. No-wait flowshop scheduling with large lot size. *Annals of Operations Research*. 70, 415–438.
- Ahuja, T., Magnanti, F., and Orlin, D. 1994. *Network Flows*. Wiley Interscience, San Francisco.
- Aigbedo, H., and Monden, Y. 1997. A parametric procedure for multicriterion sequence scheduling for just-in-time mixed-model assembly lines. *International Journal of Production Research*. 35, 2543–2564.
- Anily, S., Glass, C.A., and Hassin, R. 1998. The scheduling of maintenance service. *Discrete Applied Mathematics*. 82, 27–42.
- Bar-Noy, A., Bhatia, R., Naor, J.S., and Schiber, B. 1998. Minimizing service and operation costs of periodic scheduling. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 11–20.
- Clifford, J.J., and Posner, M.E. 1996. Parallel machine scheduling with high multiplicity. Working Paper 1995-006. The Ohio State University, Department of Industrial, Welding and Systems Engineering.
- Cosmadakis, S., and Papadimitriou, C. 1984. The traveling salesman problem with many visits to few cities. *SIAM Journal of Computations*. 13, 99–108.
- Eiselt, H.A., Gendreau, M., and Laporte, G. 1995. Arc routing problems, part i: The chinese postman problem. *Operations Research*. 43, 231–243.
- Gilmore, P.C., and Gomory, R.E. 1964. Sequencing a one-state variable machine: A solvable case of the traveling salesman problem. *Journal of Operations Research Society of America*. 12, 655–679.
- Graham, R.L., Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Operations Research*. 5, 287–326.
- Greenberg, H.J. 1997. RIM Sensitivity Analysis from an Interior Solution. World Wide Web, <http://www-math.cudenver.edu/hgreenbe>.

- Hitz, K.L. 1979. Scheduling of flexible flow shops. ii. Report LIDS-TH-1063. Laboratory of Information and Decision Systems, MIT, Cambridge, Mass.
- Hitz, K.L. 1980. Scheduling of scheduling of flow shops. ii. Report LIDS-R-879. Laboratory of Information and Decision Systems, MIT, Cambridge, Mass.
- Hochbaum, D.S., and Shamir, R. 1991. Strongly polynomial algorithms for the high multiplicity scheduling problem. *Operations Research*. 39, 648–653.
- Kenyon, C., Schabanel, N., and Young, N. 2000. Polynomial-time approximation scheme for data broadcast. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (STOC)*, Portland, Oregon, 21-23.
- Kubiak, W., Steiner, G., and Yeomans, J.S. 1997. Optimal level schedules for mixed-model, multi-level just-in-time assembly systems. *Annals of Operations Research*. 69, 241–259.
- Lenstra, H.W., Jr. 1983. Integer programming with a fixed number of variables. *Mathematics of Operations Research*. 8, 538–547.
- Miltenburg, J. 1989. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*. 35, 192–207.
- Monteiro, R.D.C., and Mehrotra, S. 1996. A general parametric analysis approach and its implications to sensitivity analysis in interior point methods. *Mathematical Programming*. 72, 65–82.
- Papadimitriou, C.H., and Kanellakis, P. 1980. Flow-shop scheduling with limited temporary storage. *Journal of the ACM*. 27, 533–549.
- Psaraftis, H.N. 1980. A dynamic programming approach for sequencing groups of identical jobs. *Operations Research*. 28, 1347–1359.
- Röck, H. 1981. The three-machine no-wait flow shop is np-complete. *Journal of the Association for Computing Machinery*. 31, 336–345.
- Van de Klundert, J. 1995. A note on the high multiplicity traveling salesman problem. Working paper. University of Maastricht, Faculty of Economics and Business Administration, Department of Quantitative Economy.
- Wisner, D.A. 1972. Solution of the flow shop scheduling problem with no intermediate queues. *Operations Research*. 20, 689–697.
- Zhang, S., and Van der Veen, J.A.A. 1996. Low-complexity algorithms for sequencing jobs with a fixed number of job-classes. *Computers And Operations Research*. 23, 1059–1067.