Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Sugar beet transportation problem under growers' equity regulations: Metaheuristic approach

Dragana Drenovac[a*], Đorđe Stakić[b], Ana Anokić[c], Tatjana Davidović[d] and Milorad Vidović[a]

[a]*University of Belgrade, Faculty of Transport and Traffic Engineering Vojvode Stepe 305, 11000 Belgrade, Serbia*
[b]*University of Belgrade, Faculty of Economics and Business Kamenička 6, 11000 Belgrade, Serbia*
[c]*Academy of Technical and Art Applied Studies Belgrade, Department School of Information and Communication Technologies, Zdravka Čelara 16, 11000 Belgrade, Serbia*
[d]*Mathematical Institute of the Serbian Academy of Sciences and Arts, Kneza Mihaila 36, 11000 Belgrade, Serbia*

C H R O N I C L E

Article history:
Received March 15 2025
Received in Revised Format May 22 2025
Accepted June 23 2025
Available online June 23 2025

Keywords:
Sugar beet transportation
Growers' equity
Integer linear programming
Variable neighborhood search
Greedy randomized adaptive search procedure

A B S T R A C T

We consider the optimization problem related to the sugar beet transportation when supplying sugar mills in the sugar production. The sugar beet transportation comprises of loading the beet collected at storage piles and then delivering it to sugar mills. An essential prerequisite to guarantee a viability of the considered sugar mill, is to transport the required quantities of sugar beet while maximizing technological quality and minimizing transportation costs. Some growers may be privileged to conduct collection activities in days of a planning period when sugar beet is fresh and contains larger amount of sucrose, while others do not. This unfair collect scheduling plan should be avoided to provide equal treatment of growers. We propose an Integer Linear Programming (ILP) model with the aim of simultaneously maximizing the amount of collected sucrose during the planning period while minimizing the number of vehicles of a homogeneous vehicle fleet, including constraints that provide equal opportunities for growers in sugar beet collection. The problem is denoted by the Sugar Beet Transportation Problem under Growers' Equity Regulations (SBT-GER). By applying the weighted sum method, the two objective functions are combined to transform the bi-objective problem into a single-objective one. Equity regulations are expressed through the requirement that the minimum percentage of the quantity of sugar beet is guaranteed to be collected from each grower on the day of harvest. For real-sized instances, we propose two metaheuristic algorithms, based on Variable Neighborhood Search (VNS) and Greedy Randomized Adaptive Search Procedure (GRASP), respectively. The developed mathematical model and the proposed metaheuristic approaches are evaluated on a set of randomly generated test instances. The obtained results show that VNS outperforms exact solver and GRASP for the majority of examples.

1. Introduction

Sugar beet is an important culture from different aspects: economical, agricultural, and social, among others. When it is successfully cultivated and processed, it contains a large concentration of sucrose, representing the high-quality raw material in the sugar production process. There are two periods in sugar production during a year. The first period involves planting, growth, and maturation of the plant. It lasts from the beginning of March until August. This period is followed by the second phase, known as *campaign.* It includes: sugar beet harvesting, its transportation to the refinery, and processing of raw beet. The campaign may last up to four or even six months, depending on the yield of beet and weather conditions. Particularly in South-East Europe, it lasts two or three months, starting in September or October. Usually, sugar beet supply chains start from

* Corresponding author Tel: +381113091207
E-mail drenovac@sf.bg.ac.rs (D. Drenovac)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2025 Growing Science Ltd.
doi: 10.5267/j.ijiec.2025.6.011

beet growers. They can be individual growers or agricultural cooperatives, with different land areas planted, from several dozen to several hundred hectares. On average, between 10 and 16 tons per hectare are harvested in European countries. Besides planting and growing, growers are also responsible for harvesting ripe beets and storing them in areas suitable for loading vehicles, known as storage piles. The quantity of harvested sugar beet in a certain planning period is affected by the availability of machinery, maturity of beet, as well as by the intention of the supplying area to be treated equally. The storage piles protect the beet from heat, rotting and other technological changes that reduce the sucrose content in the roots. A loss of the sucrose content varies depending on weather conditions and characteristics of the beet. The usual content of sucrose in the root is 16% of its weight, while the processing of sugar beet containing under 14% of sucrose is regarded as unprofitable (FAO, 2009). However, the sucrose content of the beet in different geographic areas may vary, occasionally reaching 20% or even 22%. Sucrose losses amount to about 1% per day and are affected by both weather conditions and beet characteristics (Assadi, 2007). The first activity in the supply chain ends with the composing of storage piles.

The sugar beet supply chain continues with the transportation of collected beets to the sugar mill. This is usually completed by the company vehicle fleet, or by hiring the Third Party Logistics (3PL) providers. During the campaign, beet is continuously processed, which means that the daily requirement of sugar beet quantities must be transported to the supplying area of the sugar mill. Daily requirements are very demanding, they are 7,000 tons on average, however, they can reach up to 10,000 tons. This means that hundreds of tours should be performed each day, depending on vehicle capacity. The transportation plan is created for a multi-day planning period and depends on the locations of the storage piles, the amount of beet prepared at each location, the requirements for each individual day, the available fleet size, as well as the most favourable transport dynamics that ensures growers' equity.

Planning period for the beet collection and transportation during the campaign is observed in the sequences that usually last from two to three days, sometimes up to six days and rarely longer, from ten to fifteen days.

Harvested sugar beet should be immediately, or as soon as possible, delivered from the fields to sugar mills because a delay in transportation leads to the inevitable loss in sucrose content, which reduces its quality. Therefore, it is essential to create a transportation plan, which includes transporting beet with the highest possible percentage of sucrose while maximizing profit, i.e., reducing transportation costs. The number of used vehicles directly affects the amount of costs, so finding the smallest number of vehicles, sufficient to meet the requirements of the collection process, is essential to reduce costs.

From the growers' point of view, it is preferable that their sugar beet is transported as soon as a storage pile is created, because this particularly impacts their earnings. Beets that are delivered from the created storage piles immediately contain more sucrose, and therefore, they will be more profitable than beets being collected later. The collection plan is subject to processing capacity limitations, so the total amount of beets cannot be transported immediately. Certain growers could have an unfair advantage of time of beet collection over others, which causes complaints of the growers. Therefore, the question of fairness arises: is it possible to define a feasible beet collection plan during the planning period so that all growers have approximately equal participation conditions?

The concept of fairness has a long history in problems where a group of subjects faces the distribution of certain resources among themselves. Fairness and equity among growers in the sugar industry can be found in literature that deals with the sugar cane harvesting and transportation problems mainly related to Australia and Thailand (Grunow et al., 2007). To the best of our knowledge, there is a lack of studies dealing with sugar beet transportation and considering equity constraints, giving us the opportunity to make valuable contributions to solving this problem.

The focus of this study is to generate the optimal sugar beet transportation plan that respects the sugar mill requirements and growers' equity constraints, while minimizing the number of used vehicles and maximizing the quality of collected sugar beet. The problem is denoted by the Sugar Beet Transportation Problem under Growers' Equity Regulations (SBT-GER).

We modeled SBT-GER as an Integer Linear Program (ILP) with the aim of simultaneously maximizing the quality of the collected beet during the planning period while minimizing the number of engaged vehicles of a homogeneous vehicle fleet, with constraints on providing equal opportunities for growers in sugar beet collection. For measuring the sucrose content in sugar beet during a time we used the decay function based on the results presented by (Asadi, 2007). By applying the weighted sum method, the two conflicting functions are joined into a single objective function. Equity regulations are expressed through the minimum percentage of the quantity of sugar beet that is guaranteed to be collected from each grower on the day of harvest. The model is used within the framework of CPLEX solver to solve small-sized problem instances. For the considered problem, we implemented two metaheuristics: Variable Neighborhood Search (VNS) and Greedy Randomized Adaptive Search Procedure (GRASP). For the experimental evaluation of the proposed algorithms, we generated 40 small-sized and 40 large-sized instances in a random manner and used them for fair comparison of CPLEX exact solver, VNS, and GRASP.

The remainder of this paper is organized as follows. Section 2 contains an overview of related work. The problem definition and the ILP mathematical formulation are given in Section 3. Section 4 contains the description of the VNS-based and the

GRASP algorithms for the considered problem. Test instances and computational results are presented and discussed in Section 5. Finally, the main conclusions are highlighted in Section 6.

2. Related work

In contrast to South American countries (Grunow et al., 2007) where the mills own and administer a large share of the haciendas (farms) and each farm is harvested completely in a single effort (single round), in Australia, farms are property of growers (Higgins, 1999). The sugar cane farms are divided into paddocks which are harvested at different times, in order to guarantee equity between the farmers whose profit depends on the sucrose content of the crops they deliver to the mills. A group of up to dozen farms are contracted for joint harvesting to ensure fair treatment of growers. Groups are harvested simultaneously during the whole harvesting period. The author developed a large-scale Integer Programming model to supply sugar mills in Australia with sugar cane while maximizing the total net revenue of growers and miller over a planning horizon of several years. Due to the timely distribution of raw cane during the planning period, it is necessary to make decisions on cane harvest terms, to determine the paddocks and quantities that meet production and transport constraints, under the equal treatment of growers. Sugar mill is supplied with cane from each farm on a rotational basis. In each round, each farm of a harvesting group should contribute to the mill requirements by an equal percent of its cane. Mill should be supplied with a similar percentage of cane from different harvesting groups. This approach provides that all harvesting groups supply the mill during the whole harvesting season. For the considered problem, a Local Search method in combination with the Tabu Search technique was developed.

In order to make a compromise between the farms within the harvesting group, Jiao et al. (2005) constructed a Linear Programming model that optimally allocates proportions of each farm to each harvest round to maximize the total sucrose content of all farms within the harvesting group. Each harvesting group as a whole, harvests a constant amount of cane throughout the harvest season. Sucrose content is modeled by a Second-order Polynomial Regression model while the estimated parameters are integrated into a Linear Optimization Program to better schedule harvest rounds to maximize gain in sucrose content. To assist growers in decision making, the application software tool (named SugarMax) was developed. The method is applied to three mill regions in Australia and showed potential gains in profitability from increased sucrose content.

Sethanan et al. (2014) considered a problem of Sugar Cane Harvest Scheduling to provide cane equity among growers in terms of both cane quantity and quality while maximizing the total sugar yield over the entire mill area. Fairness means that all groups of growers should have the same average sucrose content of cane per ton. The authors developed a Mixed Integer Programming model, as well as a heuristic algorithm called two-phase Cane Harvest Scheduling (CANEHAS). In Phase I, the CANEHAS algorithm maximizes the total sugar yield, while in Phase II, it considers the growers' equity, with respect to the cane quality. Tabu Search technique is used in both phases to improve solutions.

Thuankaewsing et al. (2015) developed a model for optimization of harvest schedule for a group of sugar cane growers in Thailand to maximize harvested sugar cane yield. Each grower owns several sugar cane fields with different yield patterns that depend on various cultivation factors. The growers are required to deliver sugar cane to the mill throughout the harvest season. The drawback of the previous harvest schedule was that it could lead to unequal opportunities for growers by allowing some of them to harvest cane in periods that provide higher sugar cane yields. An Artificial Neural Network was used to estimate the cane yield during the harvest season. Based on this estimation, an optimization model and a Tabu Search algorithm were developed. The objective was the maximization of sugar cane harvest yields with equal returns for all growers in the group under the fairness condition. The proposed optimization model could be used for solving only small-sized problem instances, while the heuristic algorithm was applied successfully to large-sized real-life instances. The results showed that the farm groups consisting of fields with widely distributed yield trends provided more opportunities to harvest cane in the best yield period for all growers, compared to the groups that consisted of fields with the same yield trend. Note that it is a different equity aspect compared to the concept proposed by Higgins (1999) where the equity among the growers in the group is ensured by an equal proportion of cane harvested in each round. The optimization model in the study of Thuankaewsing et al. (2015) determines a harvest plan in which the productivity of each grower is lower by an equal percentage than the maximum one. All of the growers in the group are assigned harvest time periods that ensure fairness with respect to both profits and sugar cane yields.

Jarumaneeroj et al. (2021) considered the supply chain of sugar cane, taking into account the three objectives of maximizing: sugar production, growers' equity in terms of a fair harvesting time-slot distribution, and the supply chain efficiency with respect to balancing resource usage during the season. The authors defined a Multi-Objective Sugarcane Harvesting Problem (MOSHP) that balances the tree objectives and proposed the Multi-Objective Evolutionary Genetic Algorithm (MOEGA) with Non-dominated Sorting Genetic Algorithm NSGA-III as a selection procedure to solve their problem. The implemented technique was tested on a real case in Thailand. It was shown that with a slight decrease in sugar production, grower equity and supply chain efficiency could be significantly improved. In addition, it was concluded that the greatest improvement can be achieved by applying a balanced yield plan, which requires a high degree of collaboration between all participants of the Thai sugar supply chain.

In the context of sugar production, there are limited papers related to the sugar beet supply chain and none of them deal with fairness and equity issues related to growers. Anokić et al. (2020) considered a variant of the Vehicle Scheduling Problem (VSP) that appears in the transportation of sugar beet. Each vehicle from a homogeneous fleet can be used multiple times during a day. Starting and ending in a sugar mill, each vehicle serves only one location in each tour with the obligation to serve urgent locations during the working day. It is not allowed that vehicles wait either at a storage pile location for loading or at the sugar mill area for unloading activities. The objective is to minimize the moment of time when all vehicles finish their last tours. The considered problem is first formulated as a Mixed Integer Quadratically Constrained Programming model and then a reformulation to a Mixed Integer Linear Program is proposed. The two models are compared using the Lingo 17 exact solver. As it turned out that Lingo can provide optimal solutions only for small-sized problem instances, two variants of the VNS metaheuristic are developed for the considered VSP.

Gvozdenović and Brcanov (2018) considered the problem of scheduling transport vehicles in a harvest season. They developed a mathematical model and designed a heuristic to synchronize the vehicle fleet and loading machines, keeping the quantities of agricultural products at the factory within the defined limits. The goal of optimization is to minimize the waiting time of transport vehicles. They considered real-world test instances related to sugar beet collection provided by the largest Serbian sugar producer (SUNOKO) with three sugar factories. The numerical experiments show that effective machine synchronization could significantly reduce vehicle waiting time and increase resource utilization. The same sugar producer was the source of real data used in study by Brcanov et al. (2025). The authors developed a mathematical model for minimizing the costs of sugar beet transportation from storage piles to sugar factories throughout the harvest campaign. They analyzed the current operating method and proposed two new scenarios that enable the application of the model to medium-sized instances and additionally reduce logistical costs. Brcanov and Gvozdenović (2024) considered the transportation process of collecting harvested sugar beet stored at storage piles to several factories. They developed a mathematical model for decision making at a strategic and tactical level in the sugar beet supply chain. The model determines an optimal subset of the existing storage piles for the supplying process to be assigned to the factories with an aim to minimize the transportation costs.

Drenovac et al. (2020) proposed a model for the optimal scheduling of deteriorating goods collection vehicles during a given planning period. In a collection process each vehicle from a homogeneous vehicle fleet performs multiple trips from a processing mill to intermediate storages where deteriorating goods are temporarily stored. The vehicle collects raw products and delivers it directly to the processing mill. The objective is to determine the optimal schedule of vehicles in such a way to maximize the quality of goods and to minimize the total vehicle idle time. The schedule should consider the fact that the quality of goods decays over time. The authors named the problem as the Multiple Trip Vehicle Scheduling Problem with Full Truckload Collection of Deteriorating Goods. The problem is first formulated as a Mixed Integer Nonlinear Program and then an alternative Set Partitioning formulation of the problem based on the Integer Linear Program is proposed. To solve the large-sized problem instances, the authors proposed the Simulated Annealing heuristic approach. The proposed approach is tested on a set of instances related to sugar beet collection with different initial quality levels, the number of storage locations, their distances from the depot, and percent of sucrose loss.

Fikry et al. (2021) dealt with the integrated Sugar Beet Supply Chain problem. They developed an Integer Linear Programming model with the objective of minimizing the total transportation and storage costs, subject to crop rotation, harvesting, transportation, inventory and processing constraints. The proposed model considers planning decisions such as selecting the land to be planted each season, determining the harvested quantity each period, calculating the transported quantity from each farm per period and estimating the total stored quantity inside the processing facility. A realistic case from the Netherlands is used to test the proposed model and elaborate its complexity.

Inspired by many successful applications to the variety of optimization problems, we decided to develop a VNS-based algorithm for solving the considered SBT-GER. The VNS metaheuristic was introduced by Mladenović and Hansen (1997) and has already been successfully applied to similar problems related to the supply chain of sugar mills (Florentino et al., 2022, Anokić et al., 2019, 2020, 2021).

Florentino et al. (2022) considered the problem of choosing sugarcane varieties for planting and for determining the optimal planting and harvesting periods, with an aim to increase production in the sugar cane industry. The problem was formulated as ILP, used by CPLEX exact solver to find optimal solutions of small- and medium-sized instances. For large instances, metaheuristic approaches based on the genetic algorithm and the VNS were developed.

The Vehicle Scheduling Problem related to the transport of sugar beet in a sugar factory in Serbia was solved by Anokić et al. (2019, 2021), along with the study Anokić et al. (2020). The authors considered a homogeneous, as well as a heterogeneous fleet of vehicles. In the first case (Anokić et al., 2021), the problem was formulated as a Mixed Integer Linear Program, and solved by a CPLEX solver. In addition, VNS and GRASP were developed. The heterogeneous variant (Anokić et al., 2019) was modeled as a Mixed Integer Quadratic Program and solved with Lingo exact solver. A General VNS approach was proposed for large-sized instances.

The VNS metaheuristic has also been implemented in some of recent studies that belongs to transportation systems (Matijević, 2023), inventory management (Ren & Wang, 2024), and industrial applications (Wu et al., 2025). Matijević (2023) considered an Electric Vehicle Routing Problem with Soft Time Windows and Time-Dependent Speeds (TD-EVRP-STW), with an aim to minimize both the total distance travelled by all vehicles and the penalty cost for missing time windows. A Mixed Integer Linear Program formulation for TD-EVRP-STW is proposed and a General Variable Neighborhood Search metaheuristic for finding solutions to this problem is developed.

Ren and Wang (2024) solved the Batch Dispersion Problem (BDP) under the assembled bill of materials and proposed a Mixed Integer Linear Program as a planning model with the objective of minimizing the total dispersion of part batches. The authors proved the NP-completeness of the problem and developed a hybrid Genetic Algorithm with Variable Neighborhood Search (BDP-GAVNS). VNS was embedded into the improved GA to increase the Local Search capability. In addition, the memory bank preservation strategy is developed to enhance the crossover operation. Four types of neighborhood structures are created based on the switch operation, the reversal operation, the insertion operation, and the mutation operation, respectively. The effectiveness of BDP-GAVNS in solving the considered problem is verified by numerical experiments.

Wu et al. (2025) dealt with the Flexible Job-shop Scheduling Problem with Sequence-Dependent Setup Times (FJSP-SDST). To minimize the makespan, the authors developed a Constraint Programming (CP) model and evaluated it within CPLEX solver. Having in mind that the problem is NP-hard, a two-stage metaheuristic algorithm is developed that combines both Cooperative Variable Neighborhood Search (C-VNS) and the CP model (C-VNS-CP). C-VNS refers the two VNS algorithms staring from different initial solutions that cooperate trough FJSP-specific cross operators. The first stage involves C-VNS with eight neighborhood structures. The good solution obtained from C-VNS becomes the initial solution that is optimized by the CP model in the second stage. The efficiency of the proposed algorithms was experimentally evaluated using 20 test instances. In the experimental results, the authors first justified the superiority of C-VNS over VNS and then showed that C-VNS-CP outperforms GA, C-VNS and CP model.

The Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic is proposed by Feo and Resende (1995) and is used as a tool to solve the problems in recent studies (Almeida et al., 2022, Baykasoğlu and Madenoğlu, 2021). Almeida et al. (2022) considered the Multi-vehicle Prize Collecting Arc Routing for Connectivity Problem involving the determination of roads damaged by natural disasters that should be cleared to facilitate relief activities. Baykasoğlu and Madenoğlu (2021) considered the Dynamic FJSP integrated with preventive maintenance activities as a response to unexpected and unknown events which may occur in the real-life systems.

In the aforementioned literature considering fairness, all studies addressed the sugar cane supply chains. Our intention is to consider a sugar beet supply chain and fairness among growers in collection and transportation, without including harvesting process. Our aim is to develop an adequate mathematical model for the optimal assignment of collection vehicles during the operational management of sugar beet supply chains. Due to the complexity of the considered problem, we propose metaheuristic approaches. Our previous experience and the reviewed literature suggest that VNS and GRASP are adequate tools to deal with SBT-GER.

3. Problem definition

The considered SBT-GER problem is focused on planning the optimal collection of sugar beet for a single sugar mill. The supply area of an arbitrary sugar mill comprises a lot of locations for the storage piles of harvested beet. The process of collection and transportation of stored sugar beet is controlled or at least synchronized by the sugar mill, according to its processing capacity. Sugar beet can stay at the on-farm storage piles for two to three days, even about one week under favourable weather conditions, without considerable quality loss. These periods should be considered as typical for transportation planning. The goal is to determine the optimal vehicle assignment to a set of available storage piles in a given interval such that continuity of sugar mill processes is ensured with a minimum possible loss of sugar content while using the smallest number of vehicles.

The characteristics of SBT-GER can be summarized in the following way:

- the vehicle fleet is homogeneous
- the vehicles perform multiple trips (tours) during the day
- there is a single depot located at the mill
- each vehicle serves only one location in each tour, starting and ending it in the depot
- daily requirements of the mill must be satisfied during the planning period
- the level of deterioration of sugar beet is monitored at the end of the working day
- given is a predefined percentage of goods representing the minimum that should be transported from each grower on the day of storage;
- the objective is to maximize the quality of collected sugar beet, while simultaneously minimize the number of used vehicles.

3.1 Relevant notation and description of the problem

Supply area could be viewed as a radial transportation network (Fig.1), where each storage pile is connected only to a sugar mill. Let $W = \{0,1,2,\dots,|W|\}$ be the set of network nodes $w \in W$, where $|W|$ denotes the number of elements in set W, i.e., the cardinality of set W.

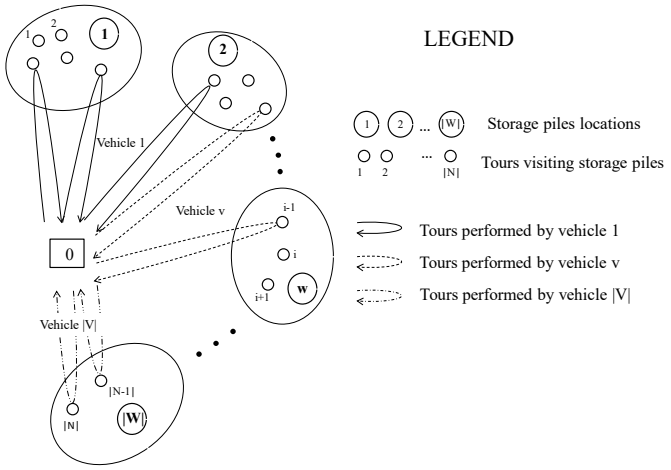

Fig. 1. The configuration of the transportation network in a sugar mill supply

The node $w = 0$ denotes the sugar mill, while the remaining nodes $w \in W \setminus \{0\}$ represent storage pile locations. The set of edges is denoted by $E = \{(0,w)|w \in W, w \neq 0\}$. As the quantity of sugar beet at each storage pile $w \in W$, Q_w is much larger than the capacity of vehicles C, collection process assumes multiple visits, i.e., multiple tours to the same network node. Therefore, graph can be transformed into another, in which each node $w \in W$ is replaced with the set N_w of virtual nodes $i \in N_w$ where $|N_w|$ represents the number of tours visiting the node $w \in W$. Denote by $N = \bigcup_{w \in W \setminus \{0\}} N_w$ the union of all subsets N_w, consisting of set of nodes $i \in N$ with single tour requests. Nodes $i \in N$ are numerated keeping the track about the location of the storage pile (and consequently the sugar beet grower). Note that index i is used also to represent the vehicle tour visiting node $i \in N$. Without loss of generality, we assumed that the vehicle fleet is homogeneous and that all vehicles have the same capacity C. Consequently, the number of virtual nodes $|N_w| = \left\lfloor \frac{Q_w}{C} \right\rfloor$. The collection process is based on the Full Truck Load (FTL) concept (Melchiori et al. 2023, Drenovac et al. 2020). The remaining quantities that are lower than the vehicle capacity could be collected by a sweep vehicle, and that part of collection is not considered in this research.

During its working hours, each vehicle $v \in V$ from the fleet $V = \{1,2,\dots,|V|\}$ performs one or more tours, collecting sugar beet from one or more storage piles and transporting it to the sugar mill. Each tour begins in the sugar mill and visits a sugar beet storage pile where the vehicle is loaded. Loaded vehicle returns to the sugar mill to be emptied into the storage area. The vehicle is ready for the next tour immediately after unloading. The total time needed for completing the tour to node $i \in N$, t_i includes traveling, loading - unloading, and weighing operations and it is known in advance. The order in which nodes are visited has no impact in the total traveling time of a vehicle. It is important to fit the tour sequence for each vehicle in the vehicle's working hours.

Optimizing the transportation of sugar beet requires the transportation process to be completed with a minimum number of vehicles to minimize the vehicle usage costs. The minimization of the number of vehicles could be achieved by packing tours into the smallest number of sequences.

Technological quality of sugar beet, i.e., root sucrose content rate, directly affects the economy of sugar mills. Therefore, the harvesting of matured sugar beet from a certain sugar beet field begins when the sucrose content is the highest. Accordingly, sugar beet collected from different storage pile locations, at different moments during the observed planning period, should be as fresh as possible, keeping sucrose content at the highest possible level. The typical daily sugar loss varies between 0.1% and 1.2% (Assadi, 2007). However, supply intensity should be synchronized with the processing capacity of the sugar mill. Our approach includes dynamic monitoring of the sucrose content on a daily level, as stored beet deteriorates during the storing period. Based on Assadi (2007), the amount of sucrose of sugar beet on an arbitrary day of the planning period, depends on the initial amount and the daily loss of sucrose content in stored sugar beet. It is expressed by equation (1) presented below.

Some growers could offer their entire beet quantities at the beginning of the planning period, in that way achieving higher profit through selling the beet with higher sucrose content, while others, who sell their own beet close to the end of the planning period, would get less profit because of lower sucrose content. Our approach assumes at least a predetermined number of collection tours to each grower, on the day of the planning period when sugar beet is stored at a storage pile. Due

to the fact that the sucrose content deteriorates on a daily basis, it is important to ensure that a certain portion of the stored beet quantity is collected from each grower on the harvesting day.

To provide fair and equal treatment of all sugar beet growers, the transportation plan should involve regulations on collection of sugar beet from different growers. It is obvious that omitting these requirements leads to grower inequity, resulting in decreasing of both sucrose content and profits for some growers.

3.2 Mathematical formulation

To formulate the problem, we use the following sets, parameters and variables.

Sets:

N set of network nodes with single tour requests $N = \{1,2, \dots, |N|\}$
D set of days in the planning period $D = \{1,2, \dots, |D|\}$
V set of vehicles available for collecting of sugar beet $V = \{1,2, \dots, |V|\}$ where, $|V| = \frac{|N|}{|D|}$
P set of sugar beet growers in the sugar mill area $P = \{1,2, \dots, |P|\}$
N_{pd} set of network nodes $i \in N_{pd} \subseteq N$ with sugar beet that grower $p \in P$ stored on day $d \in D$

Constants based on input data:

C vehicle capacity
R required daily quantity of beet necessary to be supplied to ensure the continuous production process in the sugar mill
t_i^E travel time of empty vehicle (from depot to node $i \in N$)
t_i^F the time from the moment when vehicle arrives at the node $i \in N$ until it starts the next tour from the depot. It includes the time needed for loading vehicle at the node $i \in N$, travel time of loaded vehicle to the depot, serving time at the depot, and the time a vehicle waits for the next tour
r_i the harvesting day for node $i \in N$ within the planning period D
α daily loss of sucrose content in stored beet
σ_i^0 the sucrose content of beet stored at node $i \in N$ on harvesting day r_i
σ_i^d the amount of sucrose on d-th day of the planning period of sugar beet, with initial amount of σ_i^0 units of sucrose, stored at node $i \in N$ on the harvesting day r_i. It can be expressed by equation (1)

$$\sigma_i^d = \begin{cases} \sigma_i^0 \cdot (1-\alpha)^{d-r_i}, & d \geq r_i \\ 0, & d < r_i \end{cases} \quad (1)$$

θ the minimum percentages of sugar beet amount that is needed to guarantee the equity constraints for each grower
E_{pd} minimum number of tours visiting nodes with sugar beet that grower $p \in P$ stored on day $d \in D$ required to ensure equity constraints, i.e., $E_{pd} = \lfloor \theta | N_{pd} | \rfloor$ for $d = r_i$. Note that $E_{pd} = 0$ for $d \neq r_i$.
T the number of working hours per day of vehicle $v \in V$
φ, ψ positive coefficients used in the objective function for normalization of the total collected amount of sucrose and number of used vehicles., i.e.,

$$\varphi = \frac{1}{\sum_{i=1}^{|N|} \sigma_i^0}, \qquad \psi = \frac{1}{|V|},$$

where $\sum_{i=1}^{|N|} \sigma_i^0$ is the maximal level of sucrose, i.e., the amount of sucrose collected from all storage piles at the moment of harvesting, while $|V|$ is the total number of available vehicles
λ nonnegative weight coefficient, $0 \leq \lambda \leq 1$

The binary decision variables:

$$y_{ivd} = \begin{cases} 1, & \text{if tour } i \in N \text{ is performed on day } d \in D \text{ by vehicle } v \in V \\ 0, & \text{otherwise} \end{cases}$$

$$x_{vd} = \begin{cases} 1, & \text{if vehicle } v \in V \text{ is used on day } d \in D \\ 0, & \text{otherwise} \end{cases}$$

$$z_v = \begin{cases} 1, & \text{if vehicle } v \in V \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

The considered SBT-GER includes two objectives: maximization of the quality of the collected sugar beet and minimization of the number of employed vehicles. These two opposing criteria are expressed by linear functions that are normalized (using the above defined constants φ and ψ) in order to transform bi-objective problem into a single-objective one. The normalized functions are multiplied by corresponding weights, λ and $1 - \lambda$.

Now we can express our problem in the form of ILP:

$$max\left(\lambda\left(\varphi\sum_{d=1}^{|D|}\sum_{v=1}^{|V|}\sum_{i=1}^{|N|}\sigma_i^d y_{ivd}\right)+(1-\lambda)\left(1-\psi\sum_{v=1}^{|V|}z_v\right)\right) \tag{2}$$

Subject to

$$\sum_{d=r_i}^{|D|}\sum_{v=1}^{|V|}y_{ivd}=1,\quad \forall i\in N \tag{3}$$

$$\sum_{d=1}^{r_i-1}\sum_{v=1}^{|V|}y_{ivd}=0,\quad \forall i\in N,\ r_i>1 \tag{4}$$

$$\sum_{i=1}^{|N|}y_{ivd}\,(t_i^E+t_i^F)\le T\cdot x_{vd},\qquad \forall v\in V,\qquad \forall d\in D \tag{5}$$

$$\sum_{d=1}^{|D|}x_{vd}\le |D|z_v,\quad \forall v\in V \tag{6}$$

$$C\sum_{i=1}^{|N|}\sum_{v=1}^{|V|}y_{ivd}\ge R,\quad \forall d\in D \tag{7}$$

$$\sum_{i\in N_{pd}}\sum_{v=1}^{|V|}y_{ivd}\ge E_{pd},\qquad \forall d\in D,\ \ \forall p\in P,\qquad N_{pd}\neq\emptyset,\ \ E_{pd}\neq 0 \tag{8}$$

$$y_{ivd},x_{vd},z_v\in\{0,1\},\quad \forall i\in N,\ \ \forall v\in V,\ \ \forall d\in D \tag{9}$$

The objective function (2) represents the trade-off between the total quality of beet collected during the whole planning period and the number of used vehicles. The first term corresponds to the total quality of the collected sugar beet from all visited network nodes, during the planning period D, while the second term refers to the number of vehicles that are engaged in transportation within the available working hours. The goal is to maximize both terms, as the second one represents the weighted percentage of unused vehicles. Constraints (3) provide that each node should be visited exactly once during the entire planning period but only after beet is harvested and stored. Constraints (4) prevent visiting a network node before sugar beet is ready for transport. Constraints (5) ensure that the vehicle must visit all assigned nodes within the available working hours. The relation between variables x and z is expressed by constraints (6), i.e., the vehicle is considered used only if it is used on some day. Constraints (7) provide that the amount of beet which will be delivered on any day of the planning period must be at least the required daily quantity. Constraints (8) ensure that the predetermined lowest number of tours is completed for each grower on the day of the planning period when sugar beet is harvested and stored at the storage pile. Constraints (9) define the domains of variables. The proposed model is used within CPLEX solver to provide solutions on a set of generated instances of different sizes. For large-size instances, CPLEX usually provides a feasible solution and the corresponding Upper Bound (UB) on the objective function value. The gap between the objective function value of the obtained solution and the corresponding UB increases with an increase in the size of instance. Therefore, the metaheuristic method seems to be an adequate approach for the considered problem.

4. The proposed metaheursitic approaches to SBT-GER

In order to address large-sized test instances of the considered SBT-GER problem, we have designed two algorithms based on the VNS and the GRASP metaheuristic, respectively. In the following subsections all aspects of the proposed algorithms are explained in detail.

4.1 The proposed VNS approach to SBT-GER

The structure of the Basic VNS (Hansen & Mladenović, 2018, Hansen et al., 2010, Hansen et al., 2019) where the objective function should be maximized is shown by Algorithm 1.

Algorithm 1 Basic VNS
procedure VNS($Problem\ Data; rmax$)
Generate initial solution S;
repeat
  $r \leftarrow 1$;
  while ($r \leq rmax$) **do**
    $S' \leftarrow Shake(S; r)$; //Shaking phase
    $S'' \leftarrow LocalSearch(S')$; //Local search
    if ($f(S'') > f(S)$) **then** //Neighborhood change
      $S \leftarrow S''$;
      $r \leftarrow 1$;
    else
      $r \leftarrow r + 1$;
    end if
  end while
until ($The\ termination\ criterion\ is\ satisfied$)
return S;

VNS is a metaheuristic based on exploring the different neighborhoods of a single solution. The two main steps of this method are Shaking and Local Search (LS). The efficiency of the method depends on properly defined problem-specific neighborhood structures $N_r,\ r = 1, \dots, r_{max}$. The parameter r_{max} indicates the maximum number of neighborhoods. Multiple neighborhoods are usually required for an efficient exploration of the search space.

VNS is based on the following facts:

1. A local optimum with respect to one neighborhood structure is not necessarily a local optimum for another neighborhood structure.
2. A global optimum is a local optimum with respect to all possible neighborhood structures.
3. For many problems, local optima with respect to one or several neighborhoods are relatively close to each other.

After generating an initial solution it is being perturbed by certain moves within the current neighborhood in Shaking step to provide the diversification of the search. In the Shaking phase, a solution is selected randomly in the current neighborhood structure. The solution obtained after Shaking step is locally improved. If the LS step improves the current best solution, the search continues from the new best solution by applying the first neighborhood in Shaking step, otherwise the neighborhood structure for Shaking is performed in the next neighborhood.

The adapted elements of the VNS implementation for the considered problem are described in detail as follows.

4.1.1 Solution representation

The solution is represented as a three-dimensional matrix S, indexed by the ordinal number of the day (d), the vehicle (k) used in day d, and the ordinal number of the tour (i) assigned to vehicle k. The element $S(d, k, i)$ of the matrix S is equal to the ordinal number of storage pile to which a vehicle k is directed in its tour i on day d. As the vehicles may not perform the same number of tours per day, we introduce empty tours concept to keep all elements of matrix S defined. The entries corresponding to empty tours are set to zero. The maximum daily number of tours G for each vehicle k is determined as the floor of the quotient of the vehicle's working hours and the duration of the shortest tour, i.e., $G = \left\lfloor T / \min_{i \in N} \{t_i^E + t_i^F\} \right\rfloor$. The maximum number of vehicles $|K|$ engaged per day is calculated as the quotient of the number of nodes (tours to be performed) $|N|$ and the length of the planning period $|D|$, $|K| = |N|/|D|$. The dimension of matrix S is $|D| \times |K| \times G$.

4.1.2 Generating initial solution

The solution matrix of the initial solution is filled for each day considering locations (storage piles) first and vehicles next, taking into account the equity constraints and limits for working hours. The locations are sorted in the non-increasing order of their distance from the depot. In addition, it is necessary to keep the track about the remaining quantities of sugar beet on each location. The vehicles are assigned tours according to the Best Fit Decreasing (BFD) principle (Johnson, 1974). To illustrate the procedure for generating an initial solution, we use a small example. The data in this instance correspond to the first of tested instances in Section 5, denoted by 1000_1, with a daily requirement of a sugar mill of 1000 tons on average, 5 storage piles owned by 3 growers, and the minimum percentage of the sugar beet amount to be transported for each grower on a day when it is stored θ=20%. It is assumed that vehicle capacity is C=25 t and that the planning period lasts 3 days. The detailed data related to each storage pile location are provided in Table 1. The third row contains the amounts of sugar beet collected at each storage pile (Q_w). The number of nodes (tours) for each storage pile ($|N_w|$) is calculated based on vehicle capacity of 25t and it is provided in row four. The last two rows contain the index of the day when sugar beet is collected (r_i) and the duration of tour ($t_i^E + t_i^F$) visiting node i, respectively.

Table 1
Data for instance 1000_1

	Storage piles (corresponding grower)				
w	1 (grower 1)	2 (grower 1)	3 (grower 2)	4 (grower 2)	5 (grower 3)
Q_w (t)	400	575	600	725	700
$\|N_w\|$	16	23	24	29	28
r_i	1	1	1	1	1
$(t_i^E + t_i^F)$ (h)	7.08	6.76	6.10	4.75	6.60

The initial collection plan for the instance (1000_1) is given in Table 2. The daily number of vehicles for this instance is 40. However, as it can be seen from Table 2, only 14 vehicles is engaged to satisfy all problem specific constraints. The number of tours for each vehicle and each day is estimated as G=5.

Table 2 contains the 2D representation of S matrix for the initial solution. The elements of this matrix contain indices of storage piles that each vehicle serves in its tour sequence for each day of the planning period. The basic variant of BFD algorithm does not include checking of equity constraints, and therefore, the resulting transportation plan is usually unfeasible for the considered problem. It is illustrated in the left part of Table 2. As the second phase of initial solution generation process, the reparation is performed in order to provide feasible regulated solution with respect to equity and working hours constraints. The resulting initial feasible solution is presented in the right part of Table 2. To analyze the regulation of the collection plan with respect to equity, both the plan before regulation and the regulated plan are represented.

Table 2
Unregulated and regulated collection plan of the instance 1000_1

No. of vehicle	**Ordinal number of storage pile (unregulated collection plan)**															**Ordinal number of storage pile (regulated collection plan for $\theta = 20\%$)**													
	Day 1			Day 2						Day 3						Day 1						Day 2						Day 3			
1	**1**	**1**	**1**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0	**3**	**3**	**3**	0	0	**1**	**1**	**1**	0	0	**3**	**3**	**3**	**4**	0
2	**1**	**1**	**1**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0	**4**	**3**	**3**	0	0	**1**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0
3	**1**	**1**	**1**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0	**3**	**4**	**3**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0
4	**1**	**1**	**1**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0	**3**	**5**	**5**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**3**	**4**	0
5	**1**	**1**	**1**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**4**	**4**	0	**5**	**5**	**1**	0	0	**5**	**5**	**5**	0	0	**3**	**3**	**4**	**4**	0
6	**1**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**	**1**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**
7	**2**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**	**2**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**
8	**2**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**	**2**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**
9	**2**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**	**2**	**2**	**2**	0	0	**5**	**5**	**5**	0	0	**4**	**4**	**4**	**4**	**4**
10	**2**	**2**	**2**	0	0	**3**	**3**	**3**	**4**	0	0	0	0	0	0	**2**	**2**	**2**	0	0	**1**	**1**	**1**	0	0	0	0	0	0	0
11	**2**	**2**	**2**	0	0	**3**	**3**	**3**	**4**	0	0	0	0	0	0	**2**	**2**	**2**	0	0	**1**	**1**	**1**	0	0	0	0	0	0	0
12	**2**	**2**	**2**	0	0	**3**	**3**	**3**	**4**	0	0	0	0	0	0	**2**	**2**	**2**	0	0	**1**	**1**	**3**	0	0	0	0	0	0	0
13	**2**	**2**	**2**	0	0	**3**	0	0	0	0	0	0	0	0	0	**2**	**2**	**2**	0	0	**3**	**1**	**1**	0	0	0	0	0	0	0
14	**5**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**5**	0	0	0	0	**4**	0	0	0	0	0	0	0	0	0

Having in mind the value θ =20% and the data from Table 1, it is necessary to include at least 7 tours to storage piles 1 and 2 in total on the first day of the planning period, as they correspond to grower 1. On the same day, at least 10 tours both to storage piles 3 and 4 must be performed to satisfy the equity condition for grower 2. It is also required to perform at least 5 tours to storage pile 5 on the same day. In unregulated plan, these conditions are not satisfied, as for example storage pile 5 is visited only once on the first day, without a single visit to the storage piles 3 and 4. Regulation of this plan to satisfy both equity and working hours constraints, yields the feasible initial solution of instance 1000_1.

4.1.3 Shaking in the proposed VNS algorithm

After the initial solution is generated, the Shaking phase of the VNS algorithm is applied, using two types of neighborhood, named N_1 and N_2.

Neighborhood N_1

A random move in neighborhood N_1 represents the exchange of two randomly selected tours between two randomly selected vehicles in two randomly selected days. To be able to perform this exchange, the following conditions must be satisfied. Both tours should be non-empty, visiting different storage piles and sugar beet should be available (stored) on both piles on the corresponding days. The whole procedure is repeated several times, and the number of moves is the order of the neighborhood. The order changes from 1 to r_{max_1}, which is the first parameter of the proposed VNS.

Neighborhood N_2

The neighborhood N_2 is described by the following random move. For a randomly selected day, two different vehicles are picked at random to exchange one randomly selected tour. In addition, if a vehicle has less tours than the maximum number, it is allowed to choose an "empty" tour corresponding to this vehicle. Exchanging an empty tour with the non-empty one is important because it allows to change the number of tours per vehicle. Shaking in N_2 is repeated the defined number of times, representing the order of the neighborhood. The order changes from 1 to r_{max_2}. The variable r_{max_2} is another VNS parameter,

related to the neighborhood N_2. When the defined number of moves is applied in the neighborhood N_1 or N_2, it may happen that feasibility of solution is violated in terms of working hours, and after moves in the neighborhood N_1 the obtained plan can become unregulated. Therefore, the repairing procedures should be applied to obtain a feasible solution. The procedure Repair includes equity regulations and constraints related to working hours after moves in the neighborhood N_1. After r moves within neighborhood N_2, only the working hours constraints can be violated and feasibility is restored by procedure RepairW, if necessary. Functions for Shaking in neighborhoods N_1 and N_2 are presented in Algorithms 2 and 3, respectively. The required number of moves in the corresponding neighborhood is represented by an argument r.

Algorithm 2 Shaking in neighborhood N_1
function $Shaking1(S, r)$
for $(ii \leftarrow 1; ii \leq r; ii++)$ **do**
  $(d_1, d_2) \leftarrow selectDd(D)$;
  $k_1 \leftarrow selectT(S, d_1)$;
  $i_1 \leftarrow selectt(S, d_1, k_1)$;
  $k_2 \leftarrow selectT(S, d_2)$;
  $i_2 \leftarrow selectt(S, d_2, k_2)$;
  if $(S(d_1, k_1, i_1) == S(d_2, k_2, i_2))$ **then**
    continue;
  end if
  $SwapIF(S, d_1, k_1, i_1, d_2, k_2, i_2)$;
end for
if $(S\ is\ unfeasible)$ **then**
  $Repair(S)$;
end if
return S;

Algorithm 3 Shaking in neighborhood N_2
function $Shaking2(S, r)$
for $(ii \leftarrow 1;\ ii \leq r; ii++)$ **do**
  $d_1 \leftarrow selectD(D)$;
  $(k_1, k_2) \leftarrow selectTd(S, d_1)$;
  $i_1 \leftarrow selectte(S, d_1, k_1)$;
  $i_2 \leftarrow selectte(S, d_1, k_2)$;
  $Swape(S, d_1, k_1, i_1, k_2, i_2)$;
end for
if $(S\ is\ unfeasible)$ **then**
  $RepairW(S)$;
end if
return S;

We list all functions and procedures used in Algorithms 2 and 3, required to realize the defined moves in each of neighborhoods N_1 and N_2:
$selectDd(D)$ - randomly selects two different days from set D
$selectD(D)$ - randomly selects a day from set D
$selectT(S, d_1)$ - randomly selects a vehicle from day d_1 in solution S
$selectTd(S, d_1)$ - randomly selects two different vehicles from day d_1 in solution S
$selectt(S, d_1, k_1)$ - randomly selects a tour of vehicle k_1 from day d_1 in solution S
$selectte(S, d_1, k_1)$ - randomly selects a tour of vehicle k_1 from day d_1 in solution S including an "empty" tour
$SwapIF(S, d_1, k_1, i_1, d_2, k_2, i_2)$ - exchanges two locations which are visited by tour i_1 of vehicle k_1 on day d_1 and tour i_2 of vehicle k_2 on day d_2 in solution S, if possible, i.e., if sugar beet is available at the locations
$Swape(S, d_1, k_1, i_1, k_2, i_2)$ - exchanges two locations visited on day d_1: tour i_1 of vehicle k_1 and tour i_2 of vehicle k_2; if one of the tours is "empty", the exchange corresponds to the transferring a tour to that vehicle
$Repair(S)$ - repairs the solution with respect to equity and working hours constraints
$RepairW(S)$ - repairs the solution with respect working hours constraints

4.1.4 Local Search in the proposed VNS algorithm

To perform LS step on the shaken solution, two new neighborhoods N_3 and N_4, described in more details below, are used.

Neighborhood N_3

The neighborhood N_3 in the LS step is applied only if the part related to sugar beet quality is included in the objective function, i.e., $\lambda \neq 0$. It aims to increase the total amount of sucrose in the solution. The move that determines this neighborhood is

based on the random selection of two different days (d_1, d_2) such that $d_1 \neq d_2$. For day d_1, both a vehicle and its tour to some storage pile are found corresponding to the greatest amount of sucrose when transferred to day d_2 (if such a tour exists). Analogously, for day d_2 a vehicle and its tour that result in the smallest amount of sucrose if transferred to day d_1 are identified. If exchanging the identified tours increases the total amount of sucrose in the solution, the move is accepted. The random search through N_3 is repeated until an improvement move can be found. If the resulting solution violates the equity regulations or working hours constraints, it is improved to feasibility by procedure Repair. Having in mind the quadratic complexity of N_3 with respect to the number of virtual nodes $|N|$ and that its search would be time consuming, we decided to reduce the exploration of this neighborhood using the described strategy of identifying a single tour from day d_1, and another tour from d_2 to be exchanged. In such a way, the complexity of N_3 exploration becomes $O(|N|)$.

Neighborhood N_4

The neighborhood N_4 in the LS step is applied only if the part related to the number of vehicles is included in the objective function, i.e., $\lambda \neq 1$. Its role is to reduce the number of used vehicles. For each day, vehicles are sorted in increasing order with respect to their total working hours, and within one vehicle, tours are also sorted in increasing order of their serving times. Starting with the first vehicle, an attempt is made to transfer each of its tours to some other vehicle, beginning with the last one, i.e., the vehicle with the largest value of total working hours. If transfer is not possible due to exceeding the working hours, an attempt is made to exchange tours between vehicles in such a way to decrease the working hours of the vehicle that is tried to be excluded from the current solution. This is performed by procedure named as TryEmpty. As it may happen that the number of vehicles cannot be reduced (due to exchange moves), the search in N_4 is repeated $niter$ times, $niter$ representing parameter of the proposed VNS algorithm. The structure of the Local Search step of the proposed VNS approach is represented in Algorithm 4.

Algorithm 4 Local Search phase for the proposed VNS algorithm

function $LocalSearch(S, niter)$
$S_1 \leftarrow S$;
// Exploring neighborhood N_3
if $(\lambda! = 0)$ **then**
  $p \leftarrow$ true;
  while (p) **do**
    $(d_1, d_2) \leftarrow selectDd(D)$;
    $(k_1, i_1, p) \leftarrow selectMax(S_1, d_1, d_2)$;
    if $(p ==$ false$)$ **then**
      break;
    end if
    $(k_2, i_2, p) \leftarrow selectMin(S_1, d_2, d_1)$;
    if $(p ==$ false$)$ **then**
      break;
    end if
    $p \leftarrow swapIFB(S_1, d_1, k_1, i_1, d_2, k_2, i_2)$;
    $S \leftarrow S_1$;
  end while
  if $(S\ is\ unfeasible)$ **then**
    $Repair\ (S)$;
  end if
end if
// Exploring neighborhood N_4
if $(\lambda! = 1)$ **then**
  $S_1 \leftarrow S$;
  for $(iter \leftarrow 1;\ iter \leq niter;\ iter++)$ **do**
    for $(d \leftarrow 1; d \leq |D|; d++)$ **do**
      $Sortt(S_1, d)$;
      for $(k \leftarrow 1; k \leq |K|; k++)$ **do**
        $TryEmpty(S_1, d, k)$;
      end for
    end for
  end for
end if
$S \leftarrow S_1$;
return S;

We list all functions and procedures used in Algorithm 4, along with $selectDd(D)$ already mentioned in Shaking phase, for performing defined moves in each of neighborhoods N_3 and N_4:
$selectMax(S, d_1, d_2)$ - returns the three values: index of vehicle k_1 in day d_1, index of its tour i_1 and the value of logical variable p, that is true if the largest amount of collected sucrose is obtained when transferring i_1 to day d_2. If it fails to find a vehicle and its tour that can be transferred to day d_2, the returned value for p is false.
$selectMin(S, d_2, d_1)$ - returns the three values: index of vehicle k_2 in day d_2, index of its tour i_2, and the logical variable p, that has value *true* if the smallest amount of collected sucrose is obtained when transferring i_2 to day d_1. If it fails to find a vehicle and its tour that can be transferred to day d_1, the returned value for p is *false*.
$swapIFB(S, d_1, k_1, i_1, d_2, k_2, i_2)$ - exchanges tour i_1 of vehicle k_1 on day d_1 and tour i_2 of vehicle k_2 on day d_2 in solution S, if a solution with a larger amount of collected sucrose is obtained. The returned value (p) is *true* if an exchange is made and *false* otherwise.
$Sortt(S, d)$ - sorts vehicles on day d of solution S in increasing order with respect to their total working hours; for each vehicle, tours are sorted in the same order regarding their serving times.
$TryEmpty(S, d, k)$ - a procedure that attempts to transfer all tours of a non-empty vehicle k on day d of solution S to some other vehicles. It starts from the last (largest) tour of vehicle k and tries to transfer to the last vehicle. If transfer is not possible, it tries to replace it with some shorter tour of the last vehicle. If the move is successful, the next tour of vehicle k is considered, otherwise an attempt is made to transfer the current tour to the previous vehicle in the sorted list. In the case of unsuccessful move involving current tour and vehicle k+1, the next tour of vehicle k is considered. The search continues until all tours of vehicle k are examined.

4.1.5 The structure of the proposed VNS algorithm

After exploring both neighborhoods N_3 and N_4 a Local Search is completed. It returns the best obtained solution that is used to realize the Neighborhood Change step of the VNS algorithm. A detailed pseudocode of the proposed VNS algorithm for SBT-GER is shown in Algorithm 5.

Algorithm 5 The proposed VNS algorithm
procedure VNS ($Problem\ Data;\ r_{max_1};\ r_{max_2}; niter$)
Generate initial solution S;
repeat
  $kVNS \leftarrow 1$;
  $r \leftarrow 1$;
  repeat
    $S' \leftarrow S$;
    if ($kVNS == 1$) **then**
      $S' \leftarrow Shaking1(S', r)$;
    else
      $S' \leftarrow Shaking2(S', r)$;
    end if
    $S' \leftarrow LocalSearch(S')$
    $r++$;
    if ($f(S') > f(S)$) **then**
      $S \leftarrow S'$;
      $r \leftarrow 1$;
      $kVNS \leftarrow 1$;
    else if ($f(S') == f(S)$ && $nF(S') > nF(S)$) **then**
      $S \leftarrow S'$;
    end if
    if ($kVNS == 1$ && $r > r_{max_1}$) **then**
      $kVNS++$;
      $r \leftarrow 1$;
    end if
  until ($\left(kVNS == 2 \text{ \&\& } r > r_{max_2}\right) \parallel The\ termination\ criterion\ is\ satisfied$)
until ($The\ termination\ criterion\ is\ satisfied$)
return S;

As it can be seen from the beginning of Algorithm 5, Shaking step is performed by repeating a random move in neighborhood N_1 r times, starting from r=1 followed by the Local Search step and the increased of value for r. If the newly obtained solution has better quality then the current best one, the new solution replaces the current best one and the next Shaking starts from neighborhood N_1 and r=1. In the case when Local Search provides the solution with the same objective function value but with increased number of "full" vehicles, denoted by $nF(S)$ for arbitrary solution S, we adopt the new solution as the current

best one. The vehicle is considered to be "full" if the difference between its working hours and the maximum working hours (T) is less than 1 hour. If r exceeds maximum value r_{max_1}, the neighborhood N_2 with r=1 is considered for Shaking. Shaking in N_2 is performed until the current best solution is improved or r reaches r_{max_2}. The main steps of VNS are performed until the termination criterion is satisfied.

4.2 The proposed GRASP approach to SBT-GER

To evaluate our VNS approach, we develop another metaheuristic algorithm for the considered problem following the structure of the basic Greedy Randomized Adaptive Search Procedure (Resende & Ribeiro, 2016, 2019, Festa & Resende, 2018). GRASP is an iterative method consisting of two phases: Greedy Randomized Construction (GRC) and LS that alternate until the termination criterion is satisfied. The structure of the proposed GRASP algorithm is presented in Algorithm 6.

Algorithm 6 Basic GRASP
procedure GRASP ($Problem\ Data;\ \beta$)
$L \leftarrow List\ of\ storage\ piles$;
$first \leftarrow$ true;
repeat
 $S \leftarrow$ GRC($L,\ \beta$); //Solution construction
 if ($S\ is\ not\ feasible$) **then**
  $S \leftarrow Repair(S)$;
 end if
 if ($first$) **then**
  $S^* \leftarrow S$;
  $first \leftarrow$ false;
 end if
 $S' \leftarrow LocalSearch(S, niter)$; //Local Search
 if ($f(S') > f(S^*)$) **then**
  $S^* \leftarrow S'$;
 end if
until ($The\ termination\ criterion\ is\ satisfied$)
return $\boldsymbol{S}$;

The solution of the proposed GRASP algorithm depends on the input data and the value of a single parameter β used in its GRC phase. The list L contains indices of storage piles and it is used in the procedure GRC to generate initial solution of each GRASP iteration. If GRC provides an infeasible solution with respect to the equity constraints, the reparation procedure (*Repair*) is invoked to provide its feasibility. An indicator denoted by $first$ is used to mark the first GRASP iteration when the corresponding solution S constructed by GRC phase should be recorded as the current best S^*. In the next phase the same *LocalSearch* procedure used in our VNS algorithm (see Algorithm 5) is applied to the solution S. If an improvement is obtained, the current best solution S^* is updated and one GRASP iteration is completed. Iterations are repeated until the termination criterion is satisfied. The GRC phase of GRASP is a stochastic construction procedure presented in Algorithm 7.

Algorithm 7 Greedy randomized construction
procedure GRC (L; β)
$cl \leftarrow 0$;
for (each $k \in L$) **do**
 Calculate $ic(k)$;
 if ($ic(k)! = 100$) **then**
  $cl = cl \cup \{k\}$;
 end if
end for
Initialize S;
$i \leftarrow 0$; $j \leftarrow 1$; $d \leftarrow 1$;
repeat
 determine ic_{min} among elements of cl;
 determine ic_{max} among elements of cl;
 $rcl \leftarrow 0$;
 for (each $k \in cl$) **do**
  if $\left(ic_{min} \le ic(k) \le ic_{min} + \beta(ic_{max} - ic_{min})\right)$ **then**
   $rcl = rcl \cup \{k\}$;
  end if
 end for

randomly select k from rcl;
if (vehicle j can be assigned to k at day d) **then**
$i \leftarrow i+1$; $S(d,j,i)=k$;
else
$j \leftarrow j+1$; $i \leftarrow 1$; $S(d,j,i)=k$;
end if
decrease number of tours for storage pile k;
if (no tours to k left) **then**
$cl \leftarrow cl \backslash \{k\}$;
end if
Update(ic);
if (daily requirements are satisfied) **then**
$i \leftarrow 0$; $j \leftarrow 1$; $d \leftarrow d+1$;
$cl \leftarrow 0$;
for (each $k \in L$) **do**
Calculate $ic(k)$;
if $(ic(k)! = 100)$ **then**
$cl = cl \cup \{k\}$;
end if
end for
end if
until $(d > |D|)$
return S;

For each storage pile $k \in L$, a real value denoted by *incremental cost,* $ic(k)$, is calculated according to its distance from the sugar mill and its urgency. Storage piles that need to be visited on the considered day in order to provide equity constraints are referred to as urgent. The storage piles that are prepared before day d are considered as non-urgent. The unoccupied storage piles (either emptied or prepared after day d) are considered as unavailable on day d. For each urgent storage pile k, an incremental cost is calculated as the reciprocal value of its serving time, i.e., $ic(k) = \frac{1}{t_k^E + t_k^F}$. For each of the non-urgent storage piles k the value is computed as $ic(k) = \frac{1}{t_k^E + t_k^F} + 1$. To skip tours visiting unavailable storage piles, their incremental costs are set to 100. Urgent storage piles can change their status, i.e., they become non-urgent when equity constraints are satisfied and their incremental costs are updated accordingly. On the other hand, storage piles that are emptied become unavailable with incremental costs equal to 100. The tours are included in the solution in a random manner that prioritize non-decreasing order of incremental costs. In this way, the priority is given to urgent storage piles with large distances from sugar mill. When all urgent storage piles are exhausted, non-urgent ones are considered following the same order with respect to their incremental costs. Before the main loop of GRC, all entries of the solution matrix S are initialized by zeroes.

GRC is based on adding tours to the partial solution. Storage piles are selected randomly from the restricted candidate list rcl that is a subset from candidate list cl. rcl consists of storage piles whose incremental costs are within the defined range that depends on the extreme incremental costs values (ic_{min}, ic_{max}) and the value of parameter β. The aim of GRC is to favour the urgent tours and those more distant from the sugar mill. This strategy is used to provide feasible solution in terms of working hours, as a more challenging task than improving the objective function value, while equity constraints should be additionally verified by calling *Repair* procedure in the main GRASP loop. The GRC procedure iterates over tours, vehicles, and days and it ends when all tours are assigned to vehicles.

5. Experimental Analysis

All experiments are conducted on Intel Xeon CPU E5-2620 v3, 2.40 GHz with 32GB RAM memory, under Linux operating system. Mathematical model is tested using CPLEX invoked from Python, while VNS and GRASP are implemented in C programming language.

5.1 Test Instances

The performances of the developed mathematical model and the proposed metaheuristic approaches are evaluated on the set of randomly generated instances. In each instance we considered planning period of three days. A daily requirements of the mill are set to 1000, 3000, 5000 and 7000 tons, respectively, where the largest one corresponds to the typical daily production of the most of European sugar mills. Fleet size is homogeneous with the vehicle capacity of 25 tons and working hours of 24h. All instances include set of sugar beet storage piles and one mill, similar to the network shown in the Fig. 1. Distances between the mill and the storage piles are defined in the time domain, representing the total time of vehicle tour realization, and they are randomly generated according to the uniform distribution U [4, 10] hours. The number of storage piles in the small-sized instances (with daily production rate of 1000 or 3000 tons) is 5, while the large-sized instances (with daily

production rate of 5000 or 7000 tons) include 10 and 15 storage piles, respectively. Quantities of sugar beet at storage piles are determined in accordance with the required daily mill supply during the whole planning period, i.e., $Q_w = |D| \cdot R/|N_w|$. To provide different instances of the same size, the average quantities per storage pile are modified by random relocation of $25 \cdot H$ tons from one storage pile to another. The variable H is random variable with discrete uniform distribution defined by formula (10).

$$H \sim \begin{cases} U[-12, 12], H \in Z & \text{for the small-sized storage piles,} \\ U[-24, 24], H \in Z & \text{for the large-sized storage piles.} \end{cases} \quad (10)$$

The percentage of sucrose is randomly generated according to the discrete uniform distribution U [15, 22]%, while the daily losses of the sucrose are assumed to be $\alpha = 1.2\%$. The day of the planning period of the length $|D|$ when beet is available at the storage pile w is defined by a random variable R_w with integer values in the range [1, $|D|$]. Inspired by the well-known Logit model (Greene, 2018), one of widely accepted discrete choice model, we select the distribution law (11), which determines the probability that beet is available at the storage pile w on a day r_w.

$$p_{r_w} = \frac{e^{|D|-r_w+1}}{\sum_{t=1}^{|D|} e^t}, \quad r_w = 1, \dots, |D| \quad (11)$$

The equity regulation of growers' beet collection is determined by the parameter θ, i.e., by the minimum percentage of the amount of beet that must be delivered by the collection plan on the day when the storage pile is formed. Two values for θ are considered, $\theta = 20\%$ and $\theta = 30\%$. Then, the parameter E_{pd} is calculated, as explained in Section 3.

For the small-sized instances, out of 5 storage piles in total, the 3 growers own 2, 2, and 1 storage piles, respectively. For large-sized instances and two values for the number of the storage piles (10 and 15) the 3 growers own 5, 2, and 3 storage piles, and 7, 3, and 5 storage piles, respectively. For each of the four defined daily production rates, we generated 10 instances, yielding 40 test examples. Including the two values for parameter θ (θ = 20% and θ = 30%), the total of 80 instances is obtained for the evaluation of the proposed approaches.

To consider both objective functions in the optimization process, we set the value of parameter λ to 0.5. In such a way, we equalize the impact of the number of used vehicles and the total level of collected sucrose during the planning period.

5.2 Parameter Tuning

The proposed VNS metaheuristic includes the three parameters: $niter$, r_{max_1}, and r_{max_2}. For providing the best performing values for these parameters, a subset of 16 instances is used, consisting of 2 (out of 10) instances for each pair defined by daily requirements and θ. Parameter values are tested on each of the selected instances 10 times with the time limit of 20 seconds and the best solution was recorded. To manually tune the values of three VNS parameters, we fix two of them and test the all possible values for the third one. The subset of fixed parameters is changed cyclically and the whole procedure is iterated as long as the best identified value of at least one parameter changes. The criterion for determining the best parameter value is the largest number of best solutions obtained on a set of 16 selected instances. The tested values of niter belong to the set $\{1, \dots, 10\}$. The values of r_{max_1} are defined by expression $k \cdot |N|/10$ for $k \in \{1, \dots, 10\}$, where $|N|$ represents the number of virtual nodes (the total number of tours). The last parameter r_{max_2} depends on r_{max_1}, according to the formula $r_{max_2} = k \cdot r_{max_1}/(5 \cdot |D|)$, for $k \in \{1, \dots, 10\}$, with $|D|$ representing the number of days in the planning period. Therefore, 10 different values are tested for each of the VNS parameters. The conducted experimental evaluation provided the following combination of parameter values: $niter$=1, $r_{max_1} = |N|/10$, and $r_{max_2} = 6 \cdot r_{max_1}/(5 \cdot |D|)$.

The proposed GRASP includes two parameters, $niter$ with the same set of values as in VNS and $\beta = k \cdot 0.05$ for $k \in \{1, \dots, 10\}$. Within two cycles, the parameter values converged to $niter$=1 and β=0.2.

5.3 Computational Results

Using the parameter values determinate in parameter tuning tests, the proposed approaches are tested on the described set of 80 instances. For CPLEX exact solver we provide an hour of execution time and used the default parameter settings on a single thread. Metaheuristic approaches are provided 120s for each execution as the termination criterion. Having in mind that VNS and GRASP are stochastic search algorithms, we run them 30 times on each instance. This provides a fair comparison between CPLEX and the proposed metaheuristics. To evaluate the performance of VNS and GRASP, for each run we record the best obtained objective function value and the time required to obtain this value. Based on the recorded values, we calculate the average time required to obtain the best objective function value (*atbest*) and average percentage gap (*agap*) with respect to the best obtained solution, calculated according to the formula $agap = \frac{\sum_{i=1}^{n}(f^* - f_i)}{f^* \cdot n} \cdot 100$, where n represents the number of repetitions (30 in this case), f^* is the best objective function value in all 30 runs, f_i is the best objective function value obtained in the i-th run. The values of *agap* are used to evaluate the stability of the proposed methods.

The computational results on small-sized instances are provided in Table 3.

Table 3
Computational results on small-sized instances

Inst.	CPLEX			VNS	GRASP	VNS	GRASP	VNS	GRASP
	best	*gap*	*t*	f^*		*agap*		*atbest*	
				$\theta = 20\%$					
1000_1	**0.856820**	0.00	58.07	**0.856820**	0.856616	0.00	1.39	0.21	62.13
1000_2	**0.870801**	0.00	39.71	**0.870801**	0.870536	0.00	1.39	0.12	68.80
1000_3	**0.796635**	0.00	1799.84	**0.796635**	0.796604	0.00	0.00	0.13	57.78
1000_4	**0.846125**	<0.005	3600.00	**0.846125**	0.833729	1.22	0.00	0.82	60.13
1000_5	**0.772602**	0.00	730.46	**0.772602**	0.772584	0.00	0.00	0.17	63.53
1000_6	**0.831879**	<0.005	3600.01	**0.831879**	0.819319	0.00	1.12	0.31	57.49
1000_7	**0.846342**	0.00	2975.78	**0.846342**	0.846320	0.00	0.00	0.17	64.50
1000_8	**0.831696**	0.00	603.63	**0.831696**	0.819234	0.00	0.00	2.46	49.45
1000_9	**0.846655**	0.00	49.01	0.834231	0.834216	0.00	0.00	0.34	48.67
1000_10	**0.806938**	0.01	3600.01	0.806933	0.781991	0.00	0.27	61.02	64.96
3000_1	0.862193	0.97	3600.09	**0.866354**	0.857947	0.00	0.15	78.91	45.58
3000_2	0.875899	0.95	3600.04	0.875897	**0.879906**	0.38	0.00	19.65	49.58
3000_3	**0.798460**	0.00	625.45	**0.798460**	0.756676	0.00	0.46	65.05	47.09
3000_4	**0.844826**	0.00	724.49	0.840690	0.844809	0.38	0.00	89.39	58.18
3000_5	0.824580	0.51	3600.08	**0.828730**	0.828662	0.35	0.00	105.28	63.28
3000_6	0.845378	1.48	3600.04	**0.853691**	0.845324	0.39	0.46	110.64	56.11
3000_7	0.847646	0.49	3600.06	**0.851776**	0.851496	0.02	0.35	45.82	69.40
3000_8	0.838322	0.99	3600.04	**0.842452**	0.842320	0.46	0.32	40.96	53.59
3000_9	**0.829025**	0.00	573.92	**0.829025**	0.828789	0.00	0.00	116.44	57.69
3000_10	**0.794613**	<0.005	3600.03	**0.794613**	0.794544	0.00	0.00	14.17	53.18
				$\theta = 30\%$					
1000_1	**0.856820**	0.00	109.47	**0.856820**	0.856647	0.00	1.35	0.52	67.16
1000_2	**0.870801**	0.00	20.38	**0.870801**	0.870536	0.00	1.38	0.08	71.31
1000_3	**0.796611**	0.00	622.69	**0.796611**	0.796603	0.00	0.00	0.04	61.15
1000_4	**0.846102**	<0.005	3600.01	**0.846102**	0.833712	1.22	0.00	0.16	67.02
1000_5	**0.772602**	0.00	71.13	**0.772602**	0.772584	0.00	0.00	0.11	60.48
1000_6	**0.831851**	<0.005	3600	**0.831851**	0.831786	0.00	1.40	1.20	53.06
1000_7	**0.846332**	0.00	1339.06	**0.846332**	0.846320	0.00	0.00	0.12	67.72
1000_8	**0.831669**	0.00	73.66	**0.831669**	0.819207	0.00	0.00	0.88	50.00
1000_9	**0.846655**	0.00	49.53	0.834231	0.834226	0.00	0.00	0.39	60.78
1000_10	**0.806938**	<0.005	3600.01	**0.806938**	0.794465	0.00	0.00	60.14	59.56
3000_1	0.866337	0.48	3600.07	**0.866339**	0.857946	0.00	0.24	68.62	52.06
3000_2	0.875859	0.95	3600.08	0.875859	**0.879887**	0.16	0.00	31.20	53.20
3000_3	**0.798437**	0.00	628.50	**0.798437**	0.769070	0.00	0.69	60.40	56.93
3000_4	**0.844826**	0.00	238.30	0.840690	0.844806	0.38	0.00	91.35	51.07
3000_5	0.824555	0.51	3600.04	**0.828717**	0.828663	0.30	0.00	86.50	56.17
3000_6	0.849505	0.98	3600.04	**0.853665**	0.841154	0.19	0.25	109.65	54.59
3000_7	0.847652	0.49	3600.07	**0.851776**	0.851550	0.03	0.26	37.57	64.07
3000_8	0.838326	1.00	3600.09	**0.842452**	0.842320	0.46	0.32	39.61	53.83
3000_9	**0.829005**	0.00	467.49	**0.829005**	0.828789	0.00	0.00	111.68	63.62
3000_10	**0.794613**	<0.005	3600.03	**0.794613**	0.794544	0.00	0.00	14.09	51.49

It is divided in two parts corresponding to the two values of the equity parameter θ. Instance identifications (consisting of daily requirements and the indices) are presented in the first column denoted by *Inst.* The next three columns correspond to the results of CPLEX solver and contain: the best obtained objective function value (*best*), percentage gap with respect to the upper bound (*gap*), and the total execution time (t) in seconds. The remaining six columns are divided in subset of two, containing: f^*, *agap,* and *atbest* for VNS and GRASP, respectively. In each row, the best among objective function values are bolded. The corresponding results for large-sized instances are presented in Table 4. From the results shown in Table 3 it can be seen that CPLEX solved 20 (14 with $R = 1000$ an 6 with $R = 3000$) out of 40 small-sized instances to optimality. For the remaining instances, near-optimal solutions ($gap < 1.5\%$) are provided. The best solutions generated by VNS coincide with CPLEX for 24 small-sized instances, out of which 16 are optimal. VNS provides better solution than CPLEX in the case of 10 instances, while for 6 instances, CPLEX outperforms VNS. For 2 out of 40 instances, GRASP outperformed both CPLEX and VNS. According to the values of *agap*, which do not exceed 1.22% and 1.40%, we can conclude that both VNS and GRASP are quite stable. Regarding the average time needed to provide best solution, VNS demonstrates superiority for the majority of small-sized instances. On the other hand, GRASP seems to be trapped in a local optimum in about one minute, which may be an indication that initial solutions do not diversify significantly.

According to the results from Table 4, CPLEX is not able to provide optimal solutions for large-sized problem instances within time limit of 1h. VNS outperforms CPLEX for all but one instances with respect to the solution quality. GRASP provides the best solutions for 8 instances, while VNS is the most successful in the remaining 31 instances. Both metaheuristics remain stable for large-size instances with *agap* values up to 0.37% and 0.88% for VNS and GRASP, respectively. From the

last two columns, it can be concluded that VNS requires almost all the available time to find the best solution, while GRASP performs similar as for small-sized instances.

Table 4
Computational results on large-sized instances

Inst.	CPLEX			VNS	GRASP	VNS	GRASP	VNS	GRASP
	best	*gap*	*t*	f^*		*agap*		*atbest*	
				$\theta = 20\%$					
5000_1	0.836281	1.79	3600.09	**0.846256**	0.833624	0.25	0.00	106.86	66.61
5000_2	0.835153	1.80	3600.06	0.845134	**0.847301**	0.00	0.19	68.03	61.46
5000_3	0.860077	0.87	3600.18	**0.865070**	0.864637	0.05	0.00	110.34	66.79
5000_4	0.814160	2.14	3601.57	**0.826623**	0.818963	0.29	0.05	77.86	48.08
5000_5	0.852860	1.17	3600.10	**0.857791**	0.857737	0.27	0.01	109.86	61.44
5000_6	0.852048	0.59	3600.08	0.852051	**0.854219**	0.12	0.00	105.63	65.04
5000_7	0.832935	2.12	3600.43	**0.848070**	0.845245	0.13	0.01	92.79	61.47
5000_8	0.830832	0.91	3600.07	**0.835805**	0.795417	0.28	0.67	104.76	59.43
5000_9	0.852202	1.46	3600.18	**0.859690**	0.857104	0.00	0.00	100.73	58.16
5000_10	0.816358	2.14	3600.28	**0.826308**	0.815906	0.03	0.31	109.30	61.88
7000_1	0.824034	4.53	3600.72	**0.857934**	0.855888	0.00	0.00	100.94	50.39
7000_2	**0.802451**	3.09	3600.34	0.797192	0.796860	0.02	0.25	104.55	62.13
7000_3	0.835270	1.38	3600.28	**0.837125**	0.824487	0.09	0.12	106.93	54.20
7000_4	0.807414	2.71	3600.09	**0.816359**	0.812533	0.13	0.28	109.97	66.98
7000_5	0.810553	2.66	3600.37	**0.824806**	0.815697	0.07	0.72	107.59	69.91
7000_6	0.832498	1.53	3601.61	**0.837864**	0.823335	0.16	0.44	103.01	66.48
7000_7	0.838134	1.95	3603.61	0.841697	**0.843333**	0.01	0.37	114.64	68.21
7000_8	0.843847	1.34	3600.49	**0.852797**	0.848856	0.08	0.23	101.07	58.03
7000_9	0.840331	1.53	3600.12	**0.847473**	0.843514	0.19	0.12	95.22	76.44
7000_10	0.825322	2.01	3600.12	0.835966	**0.837310**	0.18	0.23	105.72	55.13
				$\theta = 30\%$					
5000_1	0.838764	1.49	3600.07	**0.846257**	0.833647	0.00	0.01	107.94	67.10
5000_2	0.837628	1.49	3600.18	0.845130	**0.847302**	0.00	0.11	67.06	58.66
5000_3	0.862515	0.69	3600.06	**0.865072**	0.864661	0.06	0.01	110.80	56.72
5000_4	0.811642	2.45	3600.10	**0.826628**	0.818958	0.13	0.04	67.05	56.89
5000_5	0.852866	1.17	3600.09	**0.857793**	0.857699	0.19	0.00	111.63	58.76
5000_6	0.849550	0.89	3600.12	0.852048	**0.854228**	0.12	0.00	105.41	52.40
5000_7	0.840488	1.20	3600.66	**0.848029**	0.845223	0.18	0.00	99.72	55.84
5000_8	0.830832	0.91	3600.18	**0.835794**	0.795407	0.23	0.68	104.54	50.15
5000_9	0.857154	0.88	3600.25	**0.859679**	0.857095	0.00	0.00	104.90	57.13
5000_10	0.821319	1.52	3601.24	**0.826308**	0.815881	0.03	0.32	109.33	54.04
7000_1	0.852577	1.03	3600.49	**0.857908**	0.855886	0.00	0.01	93.02	56.41
7000_2	0.795357	4.01	3600.32	**0.798957**	0.796904	0.22	0.02	109.78	52.59
7000_3	0.828253	2.23	3600.26	**0.837125**	0.824485	0.07	0.14	107.84	57.66
7000_4	0.805617	2.94	3600.45	**0.818130**	0.812532	0.37	0.24	108.64	69.22
7000_5	0.817695	1.77	3600.16	**0.824801**	0.817486	0.10	0.88	109.65	63.84
7000_6	0.834243	1.32	3600.06	**0.837864**	0.823331	0.16	0.50	105.03	69.64
7000_7	0.838135	1.95	3600.32	0.841694	**0.843333**	0.03	0.34	114.62	61.31
7000_8	0.847387	0.92	3600.69	**0.852797**	0.848859	0.08	0.22	101.10	53.59
7000_9	0.838573	1.75	3600.19	**0.847471**	0.843509	0.19	0.16	93.74	66.53
7000_10	0.809209	4.04	3601.18	0.835964	**0.837312**	0.18	0.21	108.74	54.87

6. Conclusion

Optimization of sugar beet transportation has a significant economic impact to all participants in sugar mills supply chains. Finding the smallest number of used vehicles, sufficient to meet the requirements of the collection process, is essential to reduce the transportation costs. Having in mind that harvested sugar beet deteriorates while waiting to be transported, the management of the sugar mill should create a transportation plan, which includes transporting beet with the highest possible percentage of sucrose to maximize profit. The collection plan is subject to the limitations of the sugar mill processing capacity, and therefore, the total amount of beet cannot be transported immediately. From the growers' point of view, it is important to define a feasible transportation plan that provides approximately equal treatment for all growers.

To consider all of the above mentioned aspects of sugar beet transportation, we introduced the Sugar Beet Transportation Problem under Growers' Equity Regulations (SBT-GER). To the best of our knowledge, we are among the first to consider equity regulations in sugar beet transportation. The problem is formulated as an Integer Linear Program that simultaneously maximizes the amount of collected sucrose during the planning period and minimizes the number of used homogeneous vehicles. Equity regulations are modeled by the requirement to collect the minimum at least the defined percentage of the quantity of sugar beet from each grower on the day of harvest. The developed model is used within CPLEX exact solver to provide optimal solutions of small-sized instances. To deal with large-sized instances, we develop two metaheuristic algorithms, Variable Neighborhood Search (VNS) and Greedy Randomized Adaptive Search Procedure (GRASP). For the experimental evaluation of the proposed approaches, we generated 80 random instances. Based on our previous experience

we selected single-solution metaheuristics that used local search to explore the solution space. The results of numerical experiments demonstrate the superiority of VNS over GRASP and CPLEX for the majority of test instances.

The avenues for future exploration should include the development of population-based metaheursitics, as well as the hybrids between different types of metaheuristics, as well as between metaheuristic and exact solver. An additional topic for future work is a generalization of the problem, e.g., by considering the heterogeneous fleet of vehicles. We strongly believe that the proposed algorithms based on GRASP and VNS can be adapted to similar transportation problems in various supply chains.

Acknowledgement

This work has been supported by the Serbian Ministry of Science, Technological Development, and Innovation, Agreement Nos. 451-03-137/2025-03/200128, 451-03-137/2025-03/200097, and 451-03-136/2025-03/ 200029.

References

Almeida, L. S., Goerlandt, F., Pelot, R., & Sörensen, K. (2022). A Greedy Randomized Adaptive Search Procedure (GRASP) for the multi-vehicle prize collecting arc routing for connectivity problem. *Computers & Operations Research*, *143*, 105804:1-14.

Anokić, A., Stanimirović, Z., Davidović, T., & Stakić, Đ. (2020).Variable neighborhood search based approaches to a vehicle scheduling problem in agriculture. *International Transactions in Operational Research, 27*(1): 26-56.

Anokić, A., Stanimirović, Z., Stakić, Đ., & Davidović, T. (2019). *General Variable Neighborhood Search for Scheduling Heterogeneous Vehicles in Agriculture*, in: Sifaleras, A., Salhi, S., & Brimberg, J. (Eds.), 6th International Conference on Variable Neighborhood Search. ICVNS 2018. Lecture Notes in Computer Science, 11328, Springer, Cham, pp. 125-140.

Anokić, A., Stanimirović, Z., Stakić, Đ., & Davidović, T. (2021). Metaheuristic approaches to a vehicle scheduling problem in sugar beet transportation. *Operational Research, 21*(3), 2021–2053.

Asadi, M. (2007). *Beet-sugar handbook*. Hoboken, New Jersey: John Wiley and Sons, Inc.

Baykasoğlu, A., & Madenoğlu, F. S. (2021). Greedy randomized adaptive search procedure for simultaneous scheduling of production and preventive maintenance activities in dynamic flexible job shops. *Soft Computing*, *25*(23), 14893-14932.

Brcanov, D., & Gvozdenović, N. (2024). Strategic decisions in logistic of sugar beet campaign. *XXIX International Scientific Conference on Strategic Management and Decision Support Systems in Strategic Management*, (pp. 302-306).

Brcanov, D., Dakić, S., Đokić, D., Gvozdenović, N., & Zekić, S. (2025). Optimization of transport activities in the sugar beet harvesting campaign. *Ekonomika Poljoprivrede*, *72*(1), 155-169.

Drenovac, D., Vidović, M., & Bjelić, N. (2020). Optimization and simulation approach to optimal scheduling of deteriorating goods collection vehicles respecting stochastic service and transport times. *Simulation Modelling Practice and Theory, 103*, 102097:1-23.

FAO. (2009). *Agribusiness Handbook - Sugar Beet. White Sugar*. Retrieved from https://www.fao.org/fileadmin/user_upload/tci/docs/AH1-(eng)Sugar%20beet%20white%20sugar.pdf

Feo, T. A., & Resende M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization, 6*, 109-133.

Festa, P., & Resende, M.G.C. (2018). *GRASP*, in: Martí, R., Pardalos, P., & Resende, M. (Eds.), Handbook of Heuristics. Springer, Cham, pp. 465-488.

Fikry, I., Gheith, M., & Eltawil, A. (2021). An integrated production-logistics-crop rotation planning model for sugar beet supply chains. *Computers & Industrial Engineering, 157*, 107300:1-13.

Florentino, H. de O., Jones, D. F., Irawan, C. A., Ouelhadj, D., Khosravi, B., & Cantane, D. R. (2022). An optimization model for combined selecting, planting and harvesting sugarcane varieties. *Annals of Operations Research, 314*(2), 451–469.

Greene, W. H. (2018). *Econometric Analysis*. 8th edition. New York: Pearson.

Grunow, M., Guntherb, H.-O., & Westinner, R. (2007). Supply optimization for the production of raw sugar. *International Journal of Production Economics, 110*(1-2), 224–239.

Gvozdenović, N., & Brcanov, D. (2018). Vehicle scheduling in a harvest season. *Ekonomika Poljoprivrede*, *65*(2), 633-642.

Hansen, P., & Mladenović, N. (2018). *Variable Neighborhood Search*, in: Martí, R., Pardalos, P., & Resende, M. (Eds.), Handbook of Heuristics. Springer, Cham, pp. 759-787.

Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2010). Variable neighbourhood search: methods and applications (invited survey). *Annals of Operations Research*, *175*, 367-407.

Hansen, P., Mladenović, N., Brimberg, J., & Pérez, J. A. M. (2019). *Variable neighborhood search*, in: Gendreau, M., & Potvin, J.-Y. (Eds.), Handbook of metaheuristics. Springer International Publishing, Cham, pp. 57-97.

Higgins, A. J. (1999). Optimizing cane supply decisions within a sugar mill region. *Journal of Scheduling, 2*(5), 229–244.

Jarumaneeroj, P., Laosareewatthanakul, N., & Akkerman, R. (2021). A multi-objective approach to sugarcane harvest planning in Thailand: Balancing output maximization, grower equity, and supply chain efficiency. *Computers & Industrial Engineering, 154*, 107129:1-13.

Jiao, Z., Higgins, A. J., & Prestwidge, D. B. (2005). An integrated statistical and optimisation approach to increasing sugar production within a mill region. *Computers and Electronics in Agriculture, 48*(2), 170–181.

Johnson, D. S. (1974). Fast algorithms for bin packing. *Journal of Computer and System Sciences, 8*(3), 272–314.
Matijević, L. (2023). General variable neighborhood search for electric vehicle routing problem with time-dependent speeds and soft time windows. *International Journal of Industrial Engineering Computations*, *14*(2), 275–292.
Melchiori, L., Nasini, G., Montagna, J. M., & Corsano, G. (2023). Resources synchronization in a full truckload pickup and delivery problem: An exact approach. *Computers & Operations Research*, *151*, 106118:1-13.
Mladenović, N. & Hansen, P. (1997). Variable Neighborhood Search. *Computers & Operations Research, 24*(11), 1097–1100.
Ren, M., & Wang, G. (2024). A hybrid genetic algorithm with variable neighborhood search for batch dispersion problem to improve traceability. *International Journal of Industrial Engineering Computations*, *15*(1), 41–58.
Resende, M. G., & Ribeiro, C. C. (2016). *Optimization by GRASP*. New York: Springer Science+ Business Media.
Resende, M. G., & Ribeiro, C. C. (2019). *Greedy randomized adaptive search procedures: advances and extensions*, in: Gendreau, M., & Potvin, J.-Y. (Eds.), Handbook of metaheuristics. Springer International Publishing, Cham, pp. 169-220.
Sethanan, K., Theerakulpisut, S., & Neungmatcha, W. (2014). *Sugarcane Harvest Scheduling to Maximize Total Sugar Yield with Consideration of Equity in Quality Among the Growers*, in: Golinska, P. (Eds.), Logistics Operations, Supply Chain Management and Sustainability. Springer, Cham, pp. 341-352.
Thuankaewsing, S., Khamjan, S., Piewthongngam, K., & Pathumnakul, S. (2015). Harvest scheduling algorithm to equalize supplier benefits: A case study from the Thai sugar cane industry. *Computers and Electronics in Agriculture, 110*, 42–55.
Wu, Y., Yang, S., Meng, L., Cheng, W., Zhang, B., & Duan, P. (2025). A novel hybrid algorithm of cooperative variable neighborhood search and constraint programming for flexible job shop scheduling problem with sequence dependent setup time. *International Journal of Industrial Engineering Computations*, *16*(1), 21–36.