
On the simulated annealing adaptation for tasks transportation optimization

ANNA BURDUK*, *Mechanical Department, Wrocław University of Science and Technology, Wrocław, Poland.*

WOJCIECH BOŻEJKO** AND JAROSŁAW PEMPERA†, *Department of Control Systems and Mechatronics, Wrocław University of Science and Technology, Wrocław, Poland.*

KAMIL MUSIAŁ ††, *Mechanical Department, Wrocław University of Science and Technology, Wrocław, Poland.*

Abstract

The paper presents an optimization task of transportation–production solved with simulated annealing algorithms. For the network of processing plants (factories) and collection centres there will be a time-optimal transportation plan established. The plan concerns transportation of raw materials to the relevant factories. A task of transportation–production with regard to the milk transportation and processing will be investigated. Simulated annealing algorithms, their properties and capabilities in solving computational problems will be described and conclusions will be presented.

Keywords: Optimization, transportation, metaheuristic.

1 Introduction

Nowadays challenges posed to transportation problems are growing increasingly; thus, more and more advanced methods need to be adapted. To proceed transport processes fluently and affectively each company has to take under consideration the principles of logistics management and focus on system organization and synchronization of physical flow of materials from manufacturers or wholesalers to consumers through all phases of the process [3, 10, 12, 22].

Expenses spent on transport are a large part of total costs and one of the factors causing the difference between the cost of producing goods and the price paid by the consumer. There are many types of products that need various procedures [6, 7]. Fresh products—due to limited period of validity—are a part of the most sensible group. In this case, transport should take place as quickly as possible and both temperature and humidity conditions have to be met. These and many more other reasons draw our attention to the problem of transport optimizing [19]. To meet these requirements and calculate so complex and multi-threaded tasks the new methods that can be successfully used

*E-mail: anna.burduk@pwr.edu.pl

**E-mail: wojciech.bozejko@pwr.edu.pl

†E-mail: jaroslaw.pempera@pwr.edu.pl

††E-mail: kamil.musial@student.pwr.edu.pl

in solving transport problem should be deployed. It is possible because of continuous technological and information advances. Transport issues are most often used to [11, 16]

1. optimal products transport planning, taking into account the minimization of costs or execution time
2. optimization of production factors distribution in order to maximize production value, profit or income.

Optimization of the cost of goods transportation have constituted the object of research for decades. Due to the nondeterministic polynomial (NP)-hardness of even the simplest (in description) problems generated by the real transportation systems there were created plenty of heuristic algorithms based on advanced optimization methods, including artificial intelligence methods.

Vehicle routing problem (VRP) is usually used to model such a kind of the system—VRP is an NP-hard problem, whose solution constitutes a key element for effectiveness of improvement in transportation. The problem, introduced by Dantzig and Ramser [8] in 1959, appears in many variants with different level of complexity. Rego [14] proposed a parallel tabu search algorithm for the VRP which uses *ejection chain* type neighbourhood. Alba and Dorronsoro [1] proposed cellular genetic algorithm for VRP, wherein crossover and mutations operators use local search procedures. The algorithm allows operating on unfeasible solutions using the appropriately modified cost function with penalties. Wodecki *et al.* [20] described a genuine parallel cost function determination on Graphics Processing Unit (GPU) for the VRP. The authors of this paper with co-authors [21] proposed also the use of another parallel metaheuristic (memetic algorithm) for this class of problems, namely Capacitated Vehicle Routing Problem (CVRP). For the problem of routing with time windows (*VRP with Time Windows*) Bouthillier and Crainic [4] proposed an effective procedure of optimization based on parallel metaheuristics (two based on the tabu search method and two genetic-based approaches) and post-optimization based on ejection chains and local search procedures.

Unfortunately, application of an algorithm taken from the literature in case of transport of food products requires consideration of specific constraints and requirements connected with keeping of high-quality resources in the food production. In particular there are properly adjusted and equipped trucks and relatively short time that may elapse from the time of milk production until the start of production of finished products.

In the milk production and processing, milk holdings consist of many companies cooperating with milk producers to receive a certain quantity of milk during the contracting period. In case of milk products high-quality preservation, milk delivered by the producer should be transported to the processing plant in a short time. One of the challenges in milk processing is the uncertain quantity of milk delivered by producers, which depends on various factors that are not necessarily affected by the producer. The producer, having some information about quantities of milk produced by sub-producers, must decide on which product and in what quantities it will be then produced in each of the holding factories. Due to the big number of farms cooperating with holdings, management of the resource transportation to production generates big costs that can be minimized by optimization.

One of the earliest works considering milk transport optimization is the paper [17], in which there are single constructive optimization algorithms applied as well as simple local search procedures. However, in the work [13] a transportation aid planning system has been presented at some Tajjian milk processing company. Advanced optimization algorithms based on genetic algorithm method for the milk transportation problem have been proposed in work [18]. The authors show differences between VRP and milk transportation problems, e.g. disinfection costs, several routes during the

planning period, division of transport space into sections that allow transportation from several clients, etc.

In the vast majority of works devoted to milk transport the main optimization criterion is the transportation cost, which depends on the distance travelled by the truck. What is mostly not taken into consideration is the fact, that milk should be delivered as soon as possible, i.e. not exceeding 24 hours, due to necessity of keeping high quality of raw materials. In this paper we propose another optimization criterion that is minimization of all the transportation tasks completion time.

In the following sections we present the mathematical description of the milk transportation problem, the proposed optimization algorithm, experimental studies to evaluate its efficiency and a case study analysis.

2 Problem description

A dairy holding consists of m production plants from the set $Z = \{1, \dots, m\}$ and administrates a fleet consisting of f vehicles for milk transportation from the set $F = \{1, \dots, f\}$. A holding company cooperates with n dairy farms from the set $N = \{1, \dots, n\}$. In the planning period, each farm $i \in F$ produces a single unit of milk. The demand for raw materials $d_z, z \in Z$ is given from each production plant. The transport of single milk unit from a provider to production plant determines a transportation task. Additionally, factory demand fully covers the milk produced by farms, i.e.

$\sum_{z=1}^m d_z = n$. Let $t_{k,l}, k, l \in \{1, \dots, n, n+1, \dots, n+m\}$ denote the transportation time between k and l locations, where the locations $1, \dots, n$ represent dairy farms, while locations $n+1, \dots, n+m$ represent production plants. A truck $i \in F$ at the time of planning is in the factory b_i and it must return to it after finishing of the job. Filling and emptying time of the truck is constant and equals L .

Let $a = (a_1, \dots, a_n), a_i \in Z, i \in N$ be an assignment of factories to suppliers. It is easy to observe that the vector a determines a set of transportation tasks, i.e. a_i means that the raw material produced by the producer $i \in N$ needs to be delivered to the factory $a_i \in Z$. An assignment of transportation tasks and the order of transportation tasks proceeded by the vehicle's fleet will be described by $r = (r_1, \dots, r_f)$, where $r_i = (r_i(1), \dots, r_i(n_i))$ is a permutation representing the order of transportation tasks execution by a vehicle $i \in F$, while n_i represents the number of these tasks. An assignment and an order of execution of transportation tasks is feasible if the following conditions are fulfilled:

1. the allocation meets the needs of the raw material factories, i.e. $|\{a_s = i : s = 1, \dots, n\}| = b_i$, for each $i \in Z$,
2. all the transportation tasks are completed, i.e. $\{r_i(s) : s = 1, \dots, n_i, i \in F\} = N$,
3. each transport task is performed exactly one time.

For the feasible pair (a, r) the time of driver's work can be determined from the following equation:

$$C_i = \sum_{s=1}^{n_i} (t_{x_s, r_i(s)} + L + t_{r_i(s), a_{r_i(s)}+n} + L) + t_{a_{r_i(n_i)}+n, b_i+n} \quad (1)$$

where $x_i = b_i + n$ for $s = 1$ and $x_s = a_{r_i(s-1)} + n$ for $s = 2, \dots, n_i$. Elements of the sum denote (in the following order) the time of approaching to the milk producer, time of milk loading, transport time to the factory and time of unloading of milk. The last element of the equation (1) is the time of the truck approaching to the parking place.

We want to find an assignment of suppliers to production facilities, assignment of transportation tasks to trucks and routes of trucks described by the pair (a^*, r^*) such that the time of execution of

TABLE 1. Transport times [min]

	Farms										Dairies	
	1	2	3	4	5	6	7	8	9	10	I	II
1	0	31	58	48	84	34	71	62	21	42	99	57
2	31	0	51	17	63	5	59	80	45	64	87	80
3	58	51	0	47	35	55	12	54	50	53	40	66
4	48	17	47	0	49	18	52	88	59	75	78	92
5	84	63	35	49	0	65	25	89	82	89	34	102
6	34	5	55	18	65	0	63	85	50	69	91	85
7	71	59	12	52	25	63	0	64	63	66	28	78
8	62	80	54	88	89	85	64	0	40	21	80	18
9	21	45	50	59	82	50	63	40	0	20	90	36
10	42	64	53	75	89	69	66	21	20	0	88	16
I	99	87	40	78	34	91	28	80	90	88	0	97
II	57	80	66	92	102	85	78	18	36	16	97	0

all the transportation tasks was the shortest, i.e.

$$C_{\max}(a^*, r^*) = \min_{a, r} C_{\max}(a, r) \quad (2)$$

where $C_{\max}(a, r) = \max_{i \in F} C_i$.

2.1 Example

A holding consisting of $m = 2$ factories cooperating with $n = 10$ production farms. For the better clarity of the example we will number the dairies with roman numerals. The transport fleet consists of $f = 3$ vehicles, two vehicles (1 and 2) are stationed at plant I, while one (3) at plant II. The demand for milk is 5 units in each plant i.e. $b_1 = b_2 = 5$. The transport times between locations are shown in Table 1. Loading and unloading time of raw material is $L = 30$ minutes.

For the pair (a, r) , $a = (2, 1, 1, 1, 1, 2, 1, 2, 2, 2)$, $r_1 = (1, 2, 7)$, $r_2 = (5, 3, 4)$, $r_3 = (10, 8, 9, 6)$ routes and working time for drivers are as follows:

1. I-1-II-2-I-7-I and $C_1 = 559$,
2. I-5-I-3-I-4-I and $C_2 = 484$,
3. II-10-II-8-II-9-II-6-II and $C_3 = 550$.

The completion time of all tasks is $C_{\max}(a, r) = 559$. Figure 1 illustrates truck routes. The red line indicates the route of driver 1, the blue of driver 2 and the black of driver 3. The dashed lines show the fragments routes, when milk is transported from 0 to dairies. It should be noted that trucks 2 and 3 supply raw material only to one plant (I and II, respectively), while truck 1 delivers the raw material to both plants.

3 Proposed algorithm SA

An algorithm based on the simulated annealing method [2], which is one of the most effective methods for constructing algorithms based on the search of solutions space, has been designed to minimize the transport time of the stock.

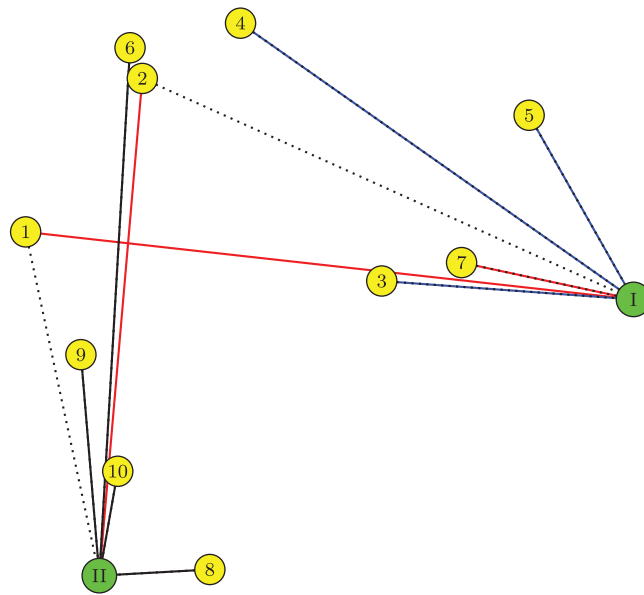


FIGURE 1. Paths of vehicles.

In algorithms based on this method, the problem is identified with the energy state of the hot metal, while the temperature is a parameter that is reduced according to the adopted cooling scheme. There are two basic schemes: logarithmic ($T_{i+1} = T_i / (1 + \lambda T_i)$) and geometric ($T_{i+1} = \lambda T_i$) where T_0 is the initial temperature, λ is a parameter of the cooling scheme, while i is an algorithm iteration. In each iteration of the algorithm, K steps are executed. Each step generates a new solution. New solution (a', r') is accepted (replaces the current solution) unconditionally if $C_{max}(a', r') \leq C_{max}(a, r)$ and with probability $p = \exp(-\frac{\Delta}{T})$, where $\Delta = C_{max}(a', r') - C_{max}(a, r)$ otherwise.

Algorithm 1 shows the pseudocode of the proposed algorithm. The upper indexes in the form of * , $'$, 0 indicate respectively the best, newly generated and initial values.

In [2] a method that automatically determines the starting temperature T_0 and the coefficient λ for the logarithmic cooling scheme has been proposed. The operation of the SA algorithm with automatic tuning parameter consists of two phases: the initial and the proper phase. In the first phase there is a certain number of iterations in which all solutions are accepted and the maximum value of Δ is determined between successive solutions, denoted by Δ_{max} . The temperature T_0 is determined by the relation $T_0 = -\Delta_{max} / \ln(p)$, where $p = 0.9$ is the probability of accepting any solution. In the proper phase of each cooling step parameter λ is computed from the $\lambda = \ln(1 + \delta) / 3\sigma$, where δ (0.1–10.0) is the equilibrium parameter and σ is the standard deviation of the objective function value determined for all solutions generated at a given temperature. The parallel version of the algorithm also is described in the literature [5].

The basic element of the algorithm based on the simulated annealing method is the way of generating a new solution. In the proposed optimization algorithm, the solution is represented by a pair of (a, r) , which precisely identifies the allocation of farms to plants, transport tasks for trucks, truck paths and the timetable and working time of drivers. The new solution is generated in two ways: (i) by interchanging two elements in r and (ii) by moving a certain element in r (see Figure 2).

Algorithm 1 Simulated Annealing SA

```

 $T \leftarrow T_0, (a, r) \leftarrow (a^0, r^0)$ 
for  $iter = 1$  to  $MaxIter$  do
  for  $k = 1$  to  $K$  do
    generate new solution  $(a', r')$  form  $(a, r)$ 
    if  $C_{max}(a', r') < C_{max}(a^*, r^*)$  then  $(a^*, r^*) \leftarrow (a', r')$ 
    compute  $\Delta = C_{max}(a', r') - C_{max}(a, r)$ 
    if  $\Delta \leq 0$  then
       $(a, r) \leftarrow (a', r')$ 
    else
      generate  $p = random(0, 1)$ 
      if  $p \leq \exp(-\Delta/T)$  then  $(a, r) \leftarrow (a', r')$ 
    end if
  end for
   $T \leftarrow \lambda T$ 
end for

```

Each of the above mentioned modifications of the current solution can be described by the tuple $v = (k, x, l, y)$, where k and l correspond to vehicles, while x and y denote the positions in paths of these vehicles. In the case of an interchange modification, a new solution is created by replacing the x -th element of k -th path with the y -th element of l -th path and inversely. The insertion modification consists in removing x -th element of path k and inserting it into y position of path l . Note that based on the modification of the interchange type, we can generate three new solutions: modifying only r , modifying only a and modifying r and a .

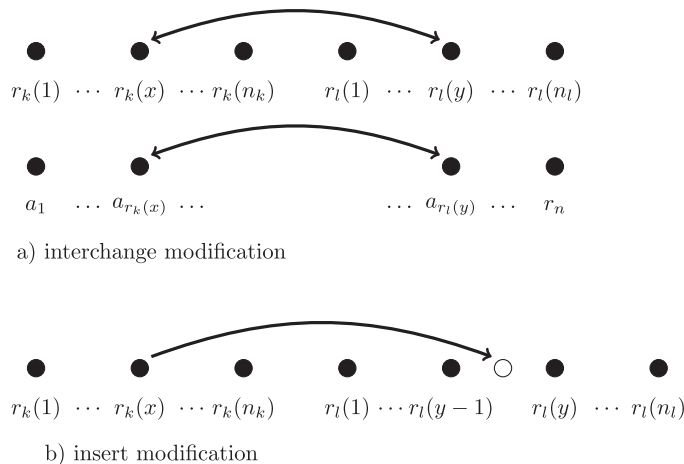


FIGURE 2. Acceptable modifications of the solution.

3.1 Some properties of the problem

The computational efficiency of algorithms can usually be significantly increased by using the properties of problems. Referring to the well-known problem of Travelling Salesman Problem (TSP), for which a lot of different types of optimization algorithms have been developed, the most effective ones use the special features of some classes of the problem, in particular the symmetry of the distance matrix. In the further part of the section, we will present the property of the problem under consideration.

For solution (a, r) let $k \in F$ be such vehicle that $C_k = C_{\max}(a, r)$. The path r_k of vehicle k will be called a critical one.

PROPERTY 1.

Let (a', r') be a solution obtained from (a, r) such that $C_{\max}(a', r') < C_{\max}(a, r)$ then

- at least one element from the r_k path is assigned to a different path or
- elements of the path r_k occur in a different order or
- the factory allocation of at least one supplier from the r_k route has been changed.

It is easy to see that the move $v = (k, x, l, y)$ applied to r and/or a meets the conditions of property 1 if at least one of the paths r_k and r_l is a critical path. The conditions of property 1 are necessary conditions, unfortunately they are not sufficient. Therefore, performing this type of move does not guarantee a better solution.

3.2 Greedy heuristic GR

In the vast majority of dairies in which transport management is conducted manually, each facility has a specific transport fleet and cooperates with a specific group of milk suppliers. Of course, in this case, the assignment vector is fixed and unchangeable, while the paths are created using the greedy rule, “Assign the vehicle that delivered the stock to the plant at the earliest”.

3.3 Ant System AS

One of the most effective algorithms used in the optimization of vehicle paths in real transport systems are algorithms based on ant search (ant colony optimization, see [15]). For the first time, the ant algorithm was proposed for the TSP problem [9]. Unfortunately, the use of the ant search for other problems requires defining the concept of the path and methods for increasing the intensity of the pheromone on the paths generated by the ants so as to achieve the desired optimization effect.

The first problem to be faced is the form of the function. In VRP problems, vehicle paths are usually determined so that the constraints are met and the sum of the path lengths is as short as possible. In our problem one should find a solution in which the maximum length of the paths (driver working times) was as short as possible. In order to achieve such an optimization effect, it was assumed that the solution constructed by the ant will consist of paths for which the working time is not greater than the given value and only the last path may exceed this value. The method of constructing the solution is described in detail in the CONSTRUCTION OF SOLUTION section in the form pseudo-code of the ant algorithm.

The space on which the ants move can be described by the graph $G = (V, E)$. The set of nodes $V = V_f \cup V_s$ consists of nodes representing suppliers $V_s = \{1, \dots, n\}$ and the factory $V_f = \{n + 1, \dots, n + m\}$. The set of arcs $E = E_f \cup E_s$ consists of arcs representing conveyance from suppliers to factories $E_f = \{(i, j) : i \in \{1, \dots, n\}, j \in \{n + 1, \dots, n + m\}\}$ and transport from factories to suppliers $E_s = \{(i, j) : i \in \{n + 1, \dots, n + m\}, j \in \{1, \dots, n\}\}$.

When constructing the solution (a, r) , the supplier and factory are alternately chosen with a probability depending on the amount of pheromone left on the path leading directly to the supplier or factory, respectively. The probability of choosing a factory (supplier) j for the ant located at the supplier (in the factory) i is

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in \mathcal{A}} [\tau_{is}]^\alpha [\eta_{is}]^\beta} & \text{if } j \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where \mathcal{A} denotes a set of unvisited suppliers (factories with unmet needs), $\eta_{ij} = 1/t_{ij}$ denotes visibility, while α and β are the parameters of the algorithm. Pseudocode is presented in Algorithm 2.

Algorithm 2 Ant System AS

```

for  $iter = 1$  to  $MaxIter$  do
  for edge  $(i, j) \in G$  do
    evaporate pheromone  $\tau_{i,j} = \rho \tau_{i,j}$ 
  end for
  for all ant do
    construct solution  $(a', r')$ 
    if  $C_{max}(a', r') < C_{max}(a^*, r^*)$  then  $(a^*, r^*) \leftarrow (a', r')$ 
    for each edge  $(i, k)$  in ant route do
      increase pheromone  $\tau_{i,j} = \tau_{i,j} + 1/C_{max}(a', r')$ 
    end for
  end for
end for

CONSTRUCTION OF SOLUTION  $(a, r)$ 
for  $k = 1, \dots, p$  do  $r_k = ()$ 
  Set  $k=1$ 
  for  $i = 1$  to  $n$  do
    Choose the unvisited supplier  $s$  to move to with probability (3)
    Choose the factory  $f$  to supply with probability (3)
    Add supplier  $s$  and factory  $f$  to  $r'_k$ 
    if  $k < p$  and  $C_k > C_{max}(a^*, r^*)$  then
      remove supplier  $s$  and factory  $f$  from  $r_k$  and add to  $r'_{k+1}$ , set  $k = k + 1$ 
    end if
  end for
end for

```

4 Experimental research

The purpose of the computer experiment was to compare the quality of the solutions generated by the GReedy (GR) algorithm and the proposed Simulated Annealing (SA) algorithm and to assess the usefulness of algorithms in the management of stock transport in real dairy farms.

For this purpose, the GR, SA and Ant System (AS) algorithms were implemented in C++. Tests of algorithms were executed on a computer with 2.4-GHz Pentium processor. The set of instances consists of 12 groups $n \times m$: 10×2 , 20×2 , 50×2 , 100×2 , 10×3 , 20×3 , 50×3 , 100×3 , 10×5 , 20×5 ,

50×5 and 100×5. Each group consisted of 5 instances. The number of vehicles was $p = n/5$. The demand for stock was evenly distributed among all mills. The distance between farm and factory locations was determined in two steps. In the first step each location coordinates were assigned to the plane while the second step was designating the Euclidean distance between the points. All coordinates (x, y) were generated randomly from the range $0 \leq x, y \leq 100$. The initial solution for the SA algorithm was generated by the GR algorithm. For each instance of the problem, the following values were set:

- (a^A, r^A) , $A \in \{GR, SA, AS\}$ - solutions generated by the GR, SA and AS algorithms,
- $CPU(A)$, $A \in \{GR, SA, AS\}$ - operating time of the GR, SA and AS algorithms.

Then, for each instance, a relative improvement was defined as follows:

$$PRD(A) = \frac{C_{\max}(r^{GR}) - C_{\max}(r^A)}{C_{\max}(r^A)} * 100\%, A \in \{SA, AS\}. \tag{4}$$

The following conclusions, basing on the collected data for each instance group, were designated:

- $ave(A)$ – average improvement obtained by $A \in \{SA, AS\}$ algorithm,
- $min(A)$ – minimal improvement obtained by $A \in \{SA, AS\}$ algorithm,
- $max(A)$ – maximal improvement obtained by $A \in \{SA, AS\}$ algorithm.

In Table 2, the average, minimum and maximum relative improvement of the initial solution generated by the algorithm GR and improved by the SA algorithm is given. The results of the studies compiled in the table show that the SA algorithm improves the solution generated by GR on average from 20% to 75%. The minimum improvement is 7.7% and the maximum is 102.3%. It can be easily seen that with the increase in the scale of activity (the increase in the number of cooperating farms and/or plants) there is also an improvement. For $m = 2$ the average improvement is in the range of 20.2–42.6% while for $m = 5$ it is in the range of 43.2–60.7%.

In Table 3, the average, minimum and maximum relative improvement of the AS algorithm is given. It is easy to see that only for a small number of n and m the value of PRD for both algorithms is comparable. With the increase in the number of suppliers and the number of factories, the advantage of the SA algorithm on the AS algorithm is growing. For $n = 100$, the average improvement of the SA algorithm is from 2 to nearly 3 times higher. The SA algorithm owes its high efficiency to generating solutions that meet the conditions of property 1.

The results of the research clearly show that the application of the proposed SA algorithm significantly reduces the maximum working time of drivers. Naturally, the question is, *does it also have a positive impact on the work of all drivers?*

TABLE 2. Average, minimal and maximal improvement obtained by the SA algorithm

n	$m = 2$			$m = 3$			$m = 5$		
	ave	min	max	ave	min	max	ave	min	max
10	20.2	7.7	29.9	24.0	4.9	36.4	43.0	32.5	60.1
20	34.0	17.3	53.2	75.3	40.1	102.3	58.0	35.4	88.7
50	42.6	36.2	46.3	46.6	32.1	62.5	60.7	40.6	83.8
100	34.6	27.5	45.3	55.2	42.9	62.8	57.9	43.5	67.8

TABLE 3. Average, minimal and maximal improvement obtained by AS algorithm

<i>n</i>	<i>m</i> = 2			<i>m</i> = 3			<i>m</i> = 5		
	ave	min	max	ave	min	max	ave	min	max
10	20.0	7.6	29.3	26.6	17.2	34.6	36.8	23.6	51.5
20	29.6	20.3	39.7	55.0	36.8	79.7	30.2	18.3	39.1
50	15.9	11.7	19.2	29.5	23.3	35.7	26.3	9.6	36.3
100	15.6	7.7	24.6	21.2	8.5	34.7	19.1	12.8	29.7

TABLE 4. Average working time of drivers

<i>n</i>	<i>m</i> = 2			<i>m</i> = 3			<i>m</i> = 5		
	GR	SA	[%]	GR	SA	[%]	GR	SA	[%]
10	810	743	9.0	510	452	12.9	338	300	12.5
20	820	681	20.3	788	649	21.3	696	494	40.8
50	849	703	20.7	762	657	16.1	850	620	37.0
100	842	737	14.3	770	632	21.8	830	649	27.8

TABLE 5. Standard deviation of working time of drivers

<i>n</i>	<i>m</i> = 2		<i>m</i> = 3		<i>m</i> = 5	
	GR	SA	GR	SA	GR	SA
10	117.4	0.3	83.0	42.3	98.5	36.9
20	80.8	7.2	325.3	1.8	98.6	12.3
50	97.9	5.8	128.9	8.7	109.5	3.7
100	86.6	4.2	128.7	7.6	106.0	6.2

In Table 4, the average working time of drivers generated by the GR and the SA algorithms was presented. In addition, the relative shortening of this time was presented. As a result of applying the SA algorithm, the average working time of drivers decreased from 9.0% to 40.8%. The smallest improvement can be observed for a small number of farms (*n*) 9.0–12.9%, while the largest for the large number (*m*) of production plants up to 40%. Table 5 shows the average standard deviation of the driver's time for paths generated by the GR and SA algorithm. It is much smaller in the case of the SA algorithm for instance $n \geq 20$. It varies in the range of 1.8–12.3%, while for the GR algorithm 80.8–325.3%. This means that the paths generated by the SA algorithm are balanced for the driver's working time (this time is roughly the same for all drivers). The execution time did not exceed 1 second for the GR algorithm and over a dozen in the case of SA algorithm. Because of its short duration, this algorithm can be used in practice.

5 Conclusions

The work is devoted to the problem of milk transport management in a dairy farm holding company. The paper presents a description of the problem and an optimization algorithm based on the simulated annealing method. The original path coding method was used in this design together with the possible ways allowing to modificate it. The experimental results of the proposed algorithm

clearly show that it generates solutions significantly better than the solutions generated by the greedy heuristic algorithm imitating manual management. Implementation of the proposed algorithm significantly reduces the maximum and the average working time of drivers. In addition, it helps to balance the working time of drivers.

Summarizing, it was shown a property of the problem, which was used in the construction of an algorithm based on the simulated annealing method. Experimental research has shown that the proposed algorithm generates much better solutions than the advanced algorithm based on swarm-intelligence methods (as ant search). Further, the short algorithm's runtime on desktop class computers allows its implementation in IT systems managing transport logistics in real-life dairy holdings.

Acknowledgements

The paper was partially supported by the National Science Centre of Poland, grant OPUS no. DEC 2017/25/B/ST7/02181.

References

- [1] E. Alba and B. Dorronsoro. Cellular genetic algorithms. In *Operations Research/Computer Science Interfaces Series 42*. Springer US, 2008.
- [2] E. H. L. Aarts and P. J. M. van Laarhoven. Simulated annealing: a pedestrian review of theory and some applications. In *Pattern Recognition and Applications*, P. A. Devijver and J. Kittler, eds. Springer, Berlin, 1987.
- [3] A. Ayough, M. Zandieh and H. Farsijani. GA and ICA approaches to job rotation scheduling problem: considering employee's boredom. *International Journal of Advanced Manufacturing Technology*, **60**, 651–666, 2012.
- [4] A. Le Bouthillier and T. Crainic. *A cooperative parallel meta-heuristic for the vehicle routing problem with time windows*. *Computers & Operations Research*, **32**, 1685–1708, 2005.
- [5] W. Bożejko, J. Pempera and M. Wodecki. Parallel simulated annealing algorithm for cyclic flexible job shop scheduling problem. *Lecture Notes in Artificial Intelligence*. **9120**, 603–612, 2015.
- [6] A. Burduk. Artificial neural networks as tools for controlling production system and ensuring their stability. In *Computer Information Systems and Industrial Management: 12th IFIP TC8 International Conference, CISIM 2013, Lecture Notes in Computer Science*, vol. 8104, 487–498, 2013.
- [7] A. Burduk. The role of artificial neural network models in ensuring the stability of systems. *10th International Conference on Soft Computing Models in Industrial and Environmental Applications, Advances in Intelligent Systems and Computing*, **368**, 427–437, 2015.
- [8] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, **6**, 80–91, 1959.
- [9] M. Dorigo, V. Maniezzo and Colomi A. The Ant System: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, **26**, 1–13, 1996.
- [10] H. A. Guvenir and E. Erel. Multicriteria inventory classification using a genetic algorithm. *European Journal of Operational Research*, **105**, 29–37, 1998.

- [11] R. Jachimowski and M. Kodawski. Simulated annealing algorithm for the multi-level vehicle routing problem. *Logistyka*, **4**, 2013.
- [12] D. Krenczyk and B. Skolud. Transient states of cyclic production planning and control. *Applied Mechanics and Materials*, **657**, 961–965, 2014.
- [13] L. Lin and T. Kawaguchi. Study on reducing raw milk transportation cost in Taiwanese Dairy Industry. *Journal of the Faculty of Agriculture, Kyushu University*, **43**, 269–280, 1998.
- [14] C. Rego. Node-ejection chains for the vehicle routing problem: sequential and parallel algorithms. *Parallel Computing*, **27**, 201–222, 2001.
- [15] A. E. Rizzoli, R. Montemanni, Lucibello E. and Gambardella L.M.. Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, **1**, 135–151, 2007.
- [16] A. Sahu and R. Tapadar. Solving the assignment problem using genetic algorithm and simulated annealing. *International Journal of Applied Mathematics*, **36**, 2007.
- [17] K. Sethanan and R. Pitakaso. Differential evolution algorithms for scheduling raw milk transportation. *Computers and Electronics in Agriculture*, **121**, 245–259, 2016.
- [18] F. K. Timon Du Wang and Pu-Yun Lu. A real-time vehicle-dispatching system for consolidating milkruns. *Transportation Research Part E*, **43**, 565–577, 2007.
- [19] M. Yusoff, J. Ariffin and A. Mohamed. Solving vehicle assignment problem using evolutionary computation. *Lecture Notes in Computer Science*, **6145**, 523–532, 2010.
- [20] M. Wodecki, W. Bożejko, S. Jagiełło and J. Pempera. Parallel Cost Function Determination on GPU for the Vehicle Routing Problem. *Lecture Notes in Artificial Intelligence*, **9120**, 778–788, 2015.
- [21] M. Wodecki, W. Bożejko, M. Karpiński and M. Pacut. Multi-GPU Parallel Memetic Algorithm for Capacitated Vehicle Routing Problem. *Lecture Notes in Computer Science*, **8386**, 207–214, 2014.
- [22] S. H. Zegordi and M. A. Beheshti Nia. A multi-population genetic algorithm for transportation scheduling. *Transportation Research Part E: Logistics and Transportation Review*, **45**, 946–959, 2009.

Received 10 October 2017