RADOSŁAW BUŁAT, MICHAŁ RADZISZEWSKI*

Multi-criterion optimisation of transport orders with the innovative evolutionary approach

Key words: genetic algorithms, PDPTW, SPEA, logistic support system

Summary: One of the common problems encountered frequently in logistic issues is PDPTW (pickup and delivery problem with time windows) where a limited transport base is to be used to expedite goods in an efficient way from point A to point B. Every organisation, both business and non-profit is, for obvious reasons, unable to grasp the whole logistic process without the aid of automation, so it has to be equipped with a logistics support system.

A viable alternative to other analytical solutions can therefore come in the form of a system based on genetic algorithms, which takes into account the limitations of the infrastructure, the time frame and the resulting penalty for any delay. This platform should also allow for the transition from a mathematically defined solution to a problem (however little practical use it has) to the real logistical problems based on the actual needs of the industry. Such a system was implemented, and with the basic genetic operators (cloning, mutation and crossover) is able to plan a solution for any arbitrarily defined, solvable problem of transportation, with the help of any algorithm using those operators. After starting the program and entering the dataset, the pre-set number of simulated generations of the genetic algorithm is started with the default chosen SPEA algorithm (strength Pareto evolutionary algorithm). The results of the simulation in the form of the final set of solutions are being saved to a file. For the algorithm applied to the test problem, the optimal solution for each variable, or middle-ground solutions were found.

1. Introduction

The processes of physical flow of material goods in the company, as well as between companies, and the flow of information used in enforcing control over these processes is the basis of logistics. Under the conditions of the modern economy, the physical flow of

* Eng. Radosław Bułat, MA—Jesuit University of Philosophy and Education "Ignatianum" in Krakow; Eng. Michał Radziszewski, MA—AGH University of Science and Technology in Krakow.

material goods becomes more and more complex. Beier and Rutkowski (1, p. 16) state that logistics faces three tasks:

– coordination of the flow of raw minerals, materials and finished products for consumers;
– minimising the costs of this flow;
– subordinating the logistic activities to the requirements of customer service.

Control of this process requires proper information and tools and the methods of their processing to achieve optimum solutions for each of the three reasons.

Execution of these tasks according to the above concept of logistics requires technical infrastructure, that is means of transport, warehouse capacity and human support. Any organisation, both business and non-profit, for obvious reasons of the incapacity of grasping the entire logistic processes without automation, must be provided with the logistic support system (Polish abbr. swl). The logistic processes executed by the system which support the organisation consist of several components, including: planning of logistic support, support of the economy with human resources, services supporting functioning and maintenance of the organisation, the databases, the information systems, the technical documentation and maintenance of reliability, performance (2). The issue of logistic infrastructure is of special significance, and at the same time it is very difficult for the economy of a country. The logistic infrastructure includes: transport infrastructure, telecommunications, warehouse facilities, other facilities, human resources infrastructure (personnel potential).

The transport infrastructure is made up of the following branches of transport: railway, road, air, pipeline, inland and sea shipping transport. Due to the protection of nature and continuous growth of transport networks, the road transport is most significant, and it has the largest share in load transport since 1998. The issue of optimal use of available transport capacity is a complex problem not only in logistics, but also in mathematics.

The objective of this paper is thus to develop a methodologically correct simulation platform allowing solution of the PDPTW problem (pickup and delivery problem with time windows)—the transport issue which consists in finding the optimum route between many points, taking into account time windows and load capacity of the available vehicles with the simulation system and evolution algorithms. This platform should also enable transition from settling the problem defined mathematically (yet with little practical application) to logistic problems based on the actual industrial needs.

2. Definition of the problem

The matrix $M_{[2q+1]\mathrm{x}[2q+1]}$ represents time and optionally costs of connection between any two points. One of these points is the transport base, the others refer to the order. There are p trucks in the base, each with the capacity c_i, $i = 1 \dots p$. There are q transport orders to be executed.

Every order is defined as a set of five elements (A, B, m, $t1$, $t2$), where A and B are identification numbers (indexes in the matrix) of points of loading and unloading, m is mass of goods to be transported (we must direct the trucks with the total capacity larger than the mass of the goods $\Sigma c_i >= p$ to execute the order), and t1 and t2 form the time window in which the order must be executed (date/ time from ($t1$) to ($t2$) the order should be executed).

The problem is static, that is all the orders are known at the time when planning starts. No new call may appear during execution of the plan.

The solution of the problem consists in completing all transport orders within the available means and all the above limitations.

To guarantee existence of the solution, soft time windows may be introduced, i.e. the pre-set times $t1$ and $t2$ may be exceeded, but their exceeding makes the solution worse (by a function of penalty defined in the problem). For such statement of the problem, there is a solution if we have at least one truck with $c > 0$.

3. Optimisation and selection of the acceptable solution

The problem stated in this way may have a lot of solutions, some of which are better than others. Therefore, it is important to phrase the criteria for which the optimisation of the found solution is provided.

There are many possibilities, e.g. minimisation of the total execution time of all the orders, the number of the used trucks, etc. The client satisfaction index may also be entered, e.g. at the maximum value when the goods are delivered right away, and decreasing when delivery is delayed in time for longer than the acceptable value. When a soft time window is accepted, the function of penalty may be added to estimation of the satisfaction of the client.

With this number of variables, it is necessary to define some assumptions as regards the conditions of the task. In this case, the issue is solved with soft time windows which guarantee existence of a solution and better reflect the reality. With transport orders, the time of loading and unloading has to be included as the function of the quantity of the goods, and the cost and time of connection between each pair of points is stored in the matrix. The cost and time do not need to be proportional. Three functions were selected for optimisation (minimising):

– The maximum number of the used vehicles defined as $\max_{t\in[0,T]}\left(v(t)\right)$, where t stands for time, T stands for the time of execution of the last call, and $v(t)$ specifies how many vehicles are off the base at the given moment.
– The average cost (not time) of execution of all the orders defined as $\left(\sum_{j=1}^{v}\sum_{i=1}^{s_j-1} M_{[X_{s,v},X_{s+1,v}]}.cost\right)\Big/ n$, where v means the maximum number of vehi-

cles participating in execution of the order (defined as in the above function), s_j means the number of points visited by the j vehicle (along with the base at the beginning and the base at the end), and X means another point on the route of the given vehicle.

– The average index of dissatisfaction of the client with our services defined as $\left(\sum_{i=1}^{n} e^{\alpha \cdot \Delta t_i}\right) \Big/ n$, where Δt means the total delay in execution of the i call, α means dissatisfaction of our clients (their mood gets worse exponentially in time), and n is the number of calls. Delay is defined as a sum of exceeding the time window and the time added in execution of other orders, i.e. $\Delta t = \max((t_{arr}-t_1), 0) + (t_{real}-M_{[A, B]}.\text{time})$, where t_{arr} means the time of arrival, t_1 means the end of the time window, t_{real} means the time between the end of loading the goods and the arrival to the point B, and $M_{[A, B]}$ means the minimum time of transfer between the points A and B (read from the matrix).

Such a defined mathematical problem can also be extended to the dynamic task by introduction of time into the simulation. In the initial moment, we would only know some of the calls for which the plan would be executed according to the earlier assumptions. However, later during the execution of the plan, which takes some time, additional call may come up. The times of their appearance and the parameters may be entered by the user or may come from the random parameter generator with the distribution parameters selected so as to enable possibly the most faithful simulation of the actual situations of transport companies. The program, apart from generating the plan, should have the possibility of its modification with elements unknown before. However, adding new calls to the algorithm is not supported in the current version.

4. Introduction to genetic algorithms

Every problem may be defined as the environment in which there is some population of individuals: the possible solutions. Each one of the individuals has specific data assigned which constitute his/ her genotype, and which are the basis for developing the phenotype with the adjusting function. The phenotype is a set of features significant for the adjusting function modelling the environment and assessed by it. All in all, the genotype describes the proposed solution of a problem, and the adjusting function assesses how good this solution is.

Genotype is made up of chromosome—units of information collected by the adjusting function—in which phenotype is encoded and, possibly, some information auxiliary for the genetic algorithm. Chromosome consists of genes, the smallest indivisible units of information (single arguments of the assessing function).

The following are common features for evolution algorithms which differentiate them from other, traditional methods of optimisation:

1. Using genetic operators which are adjusted to the form of solutions (the actions specific for the evolution process, fitted to the form of the input data).
2. Processing the population of solutions leading to parallel searching in the space of solutions from various points, which prevents “getting stuck” in the local extreme of the space of solutions.
3. Quality of the current solutions is the sufficient information for directing the search process.
4. Intentional introduction of random elements, similarly to the Monte Carlo calculation methods.

The genetic algorithm most often runs as follows (3, p. 33; 4, p. 38):

1. An initial population is drawn which creates the initial space of solutions of the problem according to statistical distribution.
2. The population is subjected to selection. The best adapted individuals in the population (the solutions closest to the optimal one) take part in the reproduction process, other are rejected as useless.
3. The genotypes of the selected individuals are subjected to evolutionary operators:
   - they are mutually matched way combination of the genotypes of their parents (crossing)—some arguments of the adjusting function are exchanged between the pair of solutions on the principle of complementarity, thus creating two new solutions maintaining some characteristics of the input solutions;
   - mutation is conducted, that is introduction of minor random changes in the solution to prevent stagnation of the algorithm in the local extreme of the adjusting function.
4. The second (successive) generation is born (the next population of solutions) and the algorithm returns to the second step if a satisfactory good solution was not found. Otherwise, the result is obtained.

5. Implementation details and substantial correctness of the platform

The software has been developed in which a clear user interface allows reading from a file, or directly from a user, the data necessary for the algorithm (the matrix of time and costs, the list of tasks and the list of trucks) processing these data (the evolutionary operators are mutation and crossover; the methodology of working with the data will be discussed in the following chapter), and returning the results. SPEA (Strength Pareto Evolutionary Algorithm) is the basic algorithm used by the software, although the program is ready for easy use of other algorithms implemented in a modular way by dll modular libraries. The descriptions of specific algorithms may be found in the literature (5, p. 126; 6, p. 47), and their tests and implementation form the subject of our next publication. Dissatisfaction of the client, the number of the trucks used and the cost of the operation are the optimised variables. Within the platform, individual solutions (proposals of solutions) are represented as single objects including several queues of the tasks executed by particular trucks (treated as the chromosomes of the crossing algorithm).

5.1. Format of data

The data, if they are not entered manually by the user, may be read from a text file including:

Matrix of time and costs

The first line of the file includes the number specifying the number of analysed places, the other lines include the square matrix where a pair of numbers in parentheses and separated with a coma is on the crossing of line n and column m. The first number is the time of travel between the points n and m, the second number is the cost of travel, and mathematical correctness of the entered data is guaranteed because the condition of the triangle in the metric space is tested for the entered data.

Table of tasks

The first line includes the number of tasks, further describe specific tasks as a series of numbers separated with commas, which mean (in the following order) the first point of the task (the point of loading) read from the matrix of time and costs, the beginning of the time window, the end of the time window, the second point (the point of unloading), the beginning of the time window, the end of the time window, the mass of transported goods.

Table of trucks

The first line specifies the number of the trucks, the further on specify load capacity of the trucks.

5.2. User interface for manual entering of data

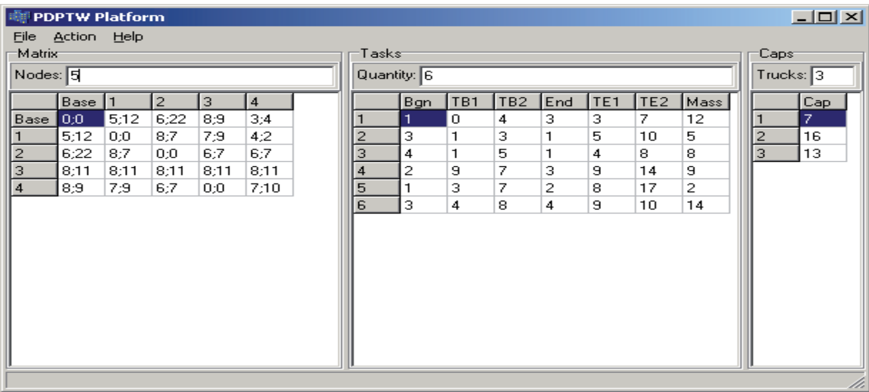

Figure 1. Interface design

Source: authors' own study.

5.3. Operations on solutions

The operators of mutation, crossover and cloning are defined on the solutions (individual agents). With these operations, more complex algorithms may be implemented.

Cloning

The result is the exact copy of the cloned solution, with breaking apart of all the memory dependencies: copying of all the data, bit by bit. Cloning is necessary to maintain the older solutions, as mutation and crossover modify the data of the solution.

Mutation

Mutation irreversibly modifies the proposed solution in a random way. If an incorrect solution results from mutation, the empty value is returned from the function. Our program in its original version executes three types of mutation (randomly selected for execution in a specific call):

– the permutation of the order of execution of the tasks by one truck within one solution;
– the exchange of the tasks between two trucks within one solution;
– handing over of the tasks by one truck to another within one solution.

The later tests proved that the second type of mutation runs worse than the third and is not necessary to ensure that the series of mutations could have a chance to generate any solution. In the final version, the first or third mutation is run with the probability of 50% each.

Crossover

Crossing between the solutions consists in selecting a random number of trucks from one solution and supplementing them with complementary trucks from the second solution, and then arranging the solutions so that every task is served exactly once. Our algorithm of the crossover:

1. Draw *n* trucks.
2. Create the list of tasks executed by these trucks in the second solution.
3. Remove all these tasks from the first solution.
4. For each drawn truck, at the end of its list of tasks in the first solution copy a list of the tasks executed by the corresponding truck in the second agent.

Crossover is defined in such a way that if the ancestors were correct it would give a correct descendant (each task should be served exactly once and the capacity of the truck will not be exceeded), which guarantees creation of only correct (not necessarily optimum) proposals of solutions.

After starting the program and entering the data, simulation is started for the preset number of generations of the genetic algorithm, by default executed according to the SPEA algorithm. The results of the simulation in the form of the final set of solutions are saved to a file.

6. Sample tests and results

Right at the very beginning it turned out that the exponential function counting dissatisfaction is not matched in terms of scale of the returned results with the rest of the optimised functions. Its values are several (even up to several dozen) orders of magnitude larger than for the other functions. This causes problems with the reliable implementation of the algorithm clustering method. Therefore, we replaced this function as follows: $\sqrt{\sum_{i=1}^{n}(\Delta t_i)^2}$, where Δt remained unchanged. After modification, the algorithm is correct and does not favour any of its optimised parameters.

Test problem

The following sample test problem was used, reproducible in the real world:

The freight operator has three vehicles in his transport fleet, capable of transporting 19, 27 and 36 units of goods, respectively. For the needs of the problem, it is not important whether it is road, sea or air transport, as we assume that all transport units are capable of identical performance in terms of speed and the possibility of reaching the points set forth in the task.

There are six clients in the address base of the forwarder, located in different places. The time and cost of transfer between any two of them is known (it is not calculated on the basis of geographical coordinates, although there is such a possibility with, e.g., paid motorways or detours instead of regular straight line distances).

The data used in the test are:

	X	A	B	C	D	E	F
X:	(0, 0)	(9,12)	(10, 14)	(8, 9)	(5, 9)	(7, 11)	(11, 13)
A:	(9, 12)	(0, 0)	(8, 7)	(7, 9)	(10, 13)	(8, 9)	(15, 16)
B:	(10, 14)	(8, 7)	(0, 0)	(15, 16)	(8, 11)	(16, 15)	(11, 12)
C:	(8, 9)	(7, 9)	(15, 16)	(0, 0)	(7, 10)	(6, 8)	(17, 18)
D:	(5, 9)	(10, 13)	(8, 11)	(7, 10)	(0, 0)	(8, 7)	(10, 11)
E:	(7, 11)	(8, 9)	(16, 15)	(6, 8)	(8, 7)	(0, 0)	(12,12)
F:	(11, 13)	(15, 16)	(11, 12)	(17, 18)	(10, 11)	(12, 12)	(0, 0),

where X is the transport base, and A–F mean consecutive clients. Thus it may be read from the above table, for example, that the time of transfer from the base to the point D takes five units of time (minutes, hours, days, depending on the scale of the enterprise) and is expressed in the cost of 9 (the costs of fuel, fees, etc.). In this case, the cost calculation is symmetric (the cost and time of transfer from X to D is identical in both directions), but it does not have to be the condition.

With the known geographical situation of the region of operations, nine transport orders were entered into the system, each one with the target point, the end point, the load of the goods to be transported and the time window in which loading and unloading should be done.

One of the sample orders is as follows:

(A, 6, 10, C, 13, 17, 12)—which means that the vehicle must load 12 units of goods in the point A between the 6th and the 10th unit of time (e.g. within 8 days, counting from the start day) and unload them in the point C, where collection of the goods may only be done between the 13th and the 17th unit of time. If these conditions are not met, penalty is charged.

Results

For the test algorithm (SPEA) applied for the test problem, optimum solutions have been found for each of the variables or intermediate solutions. The following charts refer to the following configurable data of the algorithm (these are internal parameters of the program):

- population size 240—or the number of solutions analysed in one moment;
- elite size 24—or the number of the best solutions selected from the population for further tests (10%);
- number of iterations 512—the time of operation of the algorithm calculated in its repeated runs. The exact clock time is variable and dependent on performance of the computer hardware.

The number of binary tournaments per iteration 48—or the number of solutions rejected within one iteration.

The population chart after the end of the algorithm:

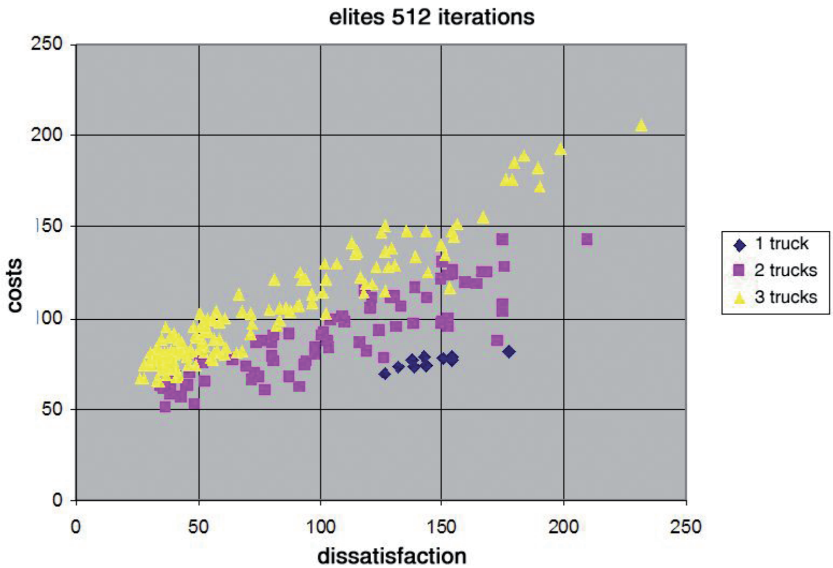

Figure 2. Initial test results

Source: authors' own study.

And the chart of elite populations in the Pareto fronts:

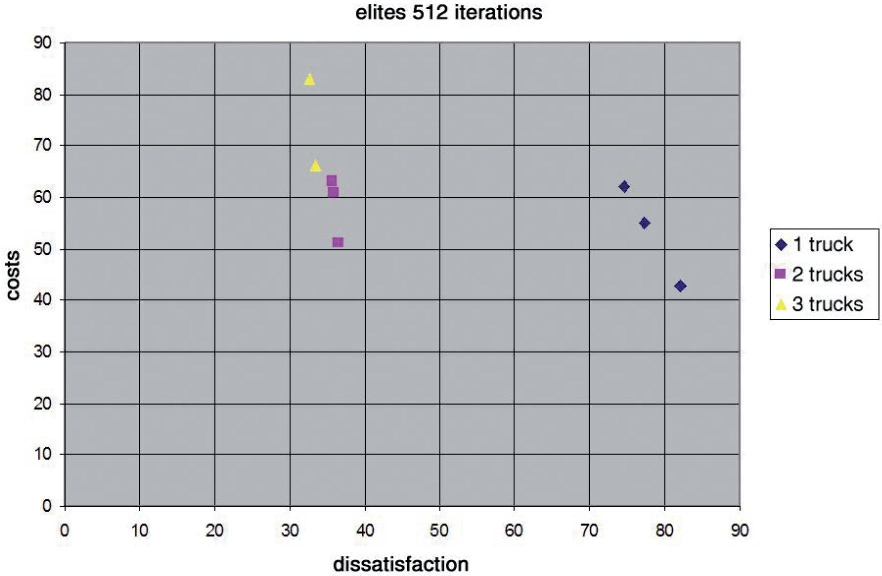

Figure 3. Optimum solutions found

Source: author's own study.

– elite solutions—or closest to the ideal—are:

32.4962	83	3
35.9166	61	2
82.0305	43	1
74.5185	62	1
35.6651	63	2
77.3499	55	1
36.4555	51	2
33.3617	66	3

For each of these solutions, it is possible to see the order of the places, where and when the vehicle should be sent to obtain the indicated result.

It is clear that after these corrections, the algorithm does what it is expected to do and finds the optimum solution for the pre-set three variables, which suggests its substantial correctness for the tested algorithm.

Bibliography

1. Beier F. J., Rutkowski K. 1995. *Logistyka*. Warszawa: Szkoła Główna Handlowa. ISBN 83-86689-06-4.
2. Chaberek M. 2001. "Modelowanie systemu wsparcia logistycznego organizacji". In: *Modelowanie procesów i systemów logistycznych*. Part One. Gdańsk: Wydawnictwo Uniwersytetu Gdańskiego, pp. 69–80.
3. Michalewicz Z. 1996. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. Transl. by Z. Nahorski. Warszawa: Wydawnictwa Naukowo-Techniczne.
4. Haupt R. L., Haupt S. E. 1998. *Practical genetic algorithms*. New York: John Wiley and Sons.
5. Kalyanmoy D. 2001. *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley and Sons. ISBN-10: 047187339X.
6. Goldberg D. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison-Wesley Professional. ISBN-10: 0201157675.

Wielokryterialna optymalizacja zleceń transportowych przy użyciu innowacyjnego podejścia ewolucyjnego

Streszczenie: Jednym ze standardowych problemów spotykanych często w zagadnieniach logistycznych jest PDPTW (Pickup and Delivery Problem with Time Windows), gdzie dysponując ograniczoną bazą transportową, należy w sposób efektywny transportować towary z punktu A do B. Każda organizacja, zarówno biznesowa, jak i o charakterze niekomercyjnym, z oczywistych powodów niemożności ogarnięcia całościowo procesów logistycznych bez pomocy automatyzacji musi być wyposażona w system wsparcia logistycznego.

Alternatywą dla innych rozwiązań analitycznych może być zatem system oparty na algorytmach genetycznych, biorący pod uwagę możliwości infrastruktury oraz ramy czasowe i wynikające z nich kary za opóźnienia. Platforma ta powinna też umożliwić przejście od rozwiązywania problematu zdefiniowanego matematycznie (jednak mającego nikłe zastosowanie praktyczne) do problemów logistycznych opartych na faktycznych potrzebach przemysłowych. System taki został zaimplementowany i przy użyciu podstawowych operatorów genetycznych – klonowania, mutacji i krzyżówki jest w stanie planować rozwiązania dla dowolnie zdefiniowanego rozwiązywalnego problemu transportowego oraz dowolnie zdefiniowanego algorytmu używającego tych operatorów. Po uruchomieniu programu i wprowadzeniu danych rozpoczynana jest symulacja zadanej ilości pokoleń algorytmu genetycznego, domyślnie wykonywanych według algorytmu SPEA (Strength Pareto Evolutionary Algorithm). Wyniki symulacji w postaci końcowego zbioru rozwiązań wypisywane są do pliku. Dla zastosowanego algorytmu dla problemu testowego znalezione zostały rozwiązania optymalne dla każdej ze zmiennych bądź rozwiązania pośrednie.

Słowa kluczowe: algorytmy genetyczne, PDPTW, SPEA, system wsparcia logistycznego