Journal of Scientific Research & Reports

Implementation of a Computerized System Based on the Recursive Core Heuristic (RCH) Algorithm for Optimal Filling of Tankers

Kangiama Lwangi Richard[1*], Bokungu Efoto Patrick[1], Katambwa Madika Cedrick[1] and Djanga Ndjondjo Piere[1]

[1]University of Kinshasa, Kinshasa, Lemba, Democratic Republic of Congo.

Original Research Article

ABSTRACT

The paper focuses on the implementation of a computerized dispatching system for petroleum products in service stations for the company SEPCONGO. The objective is to achieve a software allowing to automate the optimal planning of filling of tanker trucks by using the tools proposed by Operational Research, more precisely of the combinatorial optimization resolution methods for the multiple backpack problem. In this contribution we propose the design and realization of a computerized system for optimal filling and dispatching of tank trucks through the service stations of the City of Kinshasa based on a mathematical optimization problem, with making an optimal good decision. Knowing the trajectory of vehicles is important for a company because it saves time and capital. In this article, we clearly show that with a backpack algorithm we can easily solve the dispatching problem. However, we are proposing a system that will allow SEPCONGO to optimize the delivery of products through the city of Kinshasa in particular and those of the Democratic Republic of Congo.

Keywords: RCH; MKP; multiple knapsack problems; heuristic; dynamic,programming; operational research.

1. INTRODUCTION

It is increasingly clear that companies can no longer operate efficiently without automating their IT infrastructure. Operational Research, through its tools, can help the decision-maker when he is confronted with a problem such as a combinatorial type, i.e. includes a large number of admissible solutions among which a solution is sought. optimal or close to the optimum. And this by using algorithmic processes making it possible to optimize the solution of the problem posed while respecting its constraints [1]. This is how the algorithms, like those of the backpack problem, which are part of these Operational Research tools make it possible to solve combinatorial type problems in practical areas of daily life within companies such as in logistics such as the loading of planes, boats or even trucks, Faced with the increase in data to be managed within companies, automation has become a key issue in them. Aware of this reality, SEP / CONGO, more particularly its Dispatching service, wishes to carry out the development of the planning of tanker truck loads in an automatic way using a computer application. Indeed, the difficulty of developing tanker truck loading planning lies in having to decide on: when faced with a set of controls and a tank truck set, which controls should be loaded

into a truck of a certain capacity. Given the tankers intended to deliver the products. Knowing about the different orders, what combination of products should be loaded into the trucks in order to maximize loading at 100%; and How to carry out this loading .while wanting to minimize the cost of transport? These are the questions the dispatching service seeks to answer, as it has been a difficult task to perform until then. Of course, we are stuck with a decision problem that may have a large number of admissible solutions among which we are looking for an optimal solution or close to the optimum, in order to solve it. We thus propose to design an application implementing a combinatorial optimization algorithm while modeling the problem in the form of a multiple backpack problem. This will allow the dispatching service: to considerably increase its potential for scalability and capacity by guaranteeing constant and controlled quality of services; reduce the arduousness of the tasks for the dispatching manager. By freeing the dispatching manager from tedious and iterative tasks, we will give them time to optimize the management of the production line: Anticipate, optimize, while making him able to opt for better solutions. And also, we will help him meet the customer's needs while controlling costs.

2. PRELIMINARIES

2.1 Mathematical Optimization

2.1.1 Definition

Please check yellow highlighted References

Optimization is a branch of mathematics seeking to model, analyze and solve analytically or numerically the problems which consist in minimizing or maximizing a function on a set. The function to be optimized is called economic function or criterion or objective-function or quite simply objective **[R. FAURE, 2009.]**. Optimization plays an important role in Operations Research in applied mathematics and in analysis and other areas of applied mathematics [2].

2.1.2 Methods of solving optimization problems

The resolution of different kinds of problems encountered in every day has prompted researchers to come up with resolution methods and to make great efforts to improve their performance in terms of the computing time required and / or the quality of the proposed solution. The exact methods, the approximate methods. metaheuristic methods. Metaheuristics can be classified into two categories: single solution-based methods and population-based methods of solutions. The single solution-based methods are generally based on a local search (search by neighborhood), unfortunately most of these methods suffer from the fact that they allow to escape the problem of the local optimum and to determine the overall optimum [3].

2.1.3 Fields of application

We have several fields of application, notably in optimization of a route, a sale price, a chemical reaction, air traffic control, the efficiency of a device, the operation of an engine, the management of railway lines. , the choice of the economy of the investments, the construction of a ship, in the development of databases, for the search engines, for the code etc.

2.1.4 Uses

The problems of the dynamics of indeformable solids (especially the dynamics of articulated rigid bodies) often need mathematical optimization techniques, since we can see the dynamics of rigid bodies as solving an ordinary differential equation on a constrained manifold, the constraints are various nonlinear geometric constraints such as "these two points must always coincide", or "this point must always be on this curve".

2.2 Backpack Problem

In algorithmic and optimization, the backpack problem, also noted KP (in English, Knapsack problem) is a combinatorial optimization problem whose statement is presented as follows: "Given several objects each having a weight and a value and given a maximum weight for the bag, what items should be put in the bag in order to maximize the total value without exceeding the maximum allowable weight for the bag?" The Multiple Knapsack Problem (MKP) is a generalisation of the standard 0-1 Knapsack Problem where instead of considering only one knapsack, one tries to fill m knapsacks of different capacities. Let N = {1,,... ,n} be the set of items where each item j has a corresponding profit pj and weight wj. We consider m knapsacks of capacity ci, i ∈{1,... ,m}, then the MKP consists in filling all knapsacks so that the total profit is maximised and the sum of weights in each knapsack i does not exceed the capacity ci. We denote the binary decision variables by xij which take value: 1 if item j is assigned to knapsack i and 0 otherwise. The MKP is

formulated as the following 0-1 integer programming problem:
maximise :,

$$\sum_{i=1}^{m}\sum_{j=1}^{n} p_j\, x_i \quad (1)$$

$$S.t\ \sum_{j=1}^{n} w_j x_{ij} \le c_i, i \in \{1 \dots m\} \quad (2)$$

$$\sum_{j=1}^{m} x_{ij} \le 1, j \in \{1 \dots m\} \quad (3)$$

$x_{ij} \in \{0,1\}$, $i \in \{1,\dots,m\}$, $j \in \{1,\dots,n\}$; where p_j, c_i and w_j are positive integers and constraints (2) and (3), respectively, ensure that the filling of knapsack i does not exceed its corresponding capacity c_i and every selected item is assigned only to one knapsack, respectively. In order to avoid any trivial case, we make the following assumptions.

- All items have a chance to be packed (at least in the largest knapsack):

$$\max_{j \in N} wj \le \max_{i \in \{1 \dots m\}} c_i \quad (4)$$

- The smallest knapsack can be filled at least by the smallest item:

$$\min_{i \in \{1 \dots m\}} c_i \le \min_{i \in N} w_j \quad (5)$$

- There is no knapsack which can be filled with all items of N:

$$\sum_{k=0}^{n} w_j \le c_{i,} \forall \forall_i \in \{1 \dots . m\} \quad (6)$$

We assume also that the items and the knapsacks are sorted as follows:

,

$$\frac{p_1}{w_1} \ge \frac{p_2}{w_2} \ge \dots \ge \frac{p_n}{w_n} \quad (7)$$

$$c1 \le c2 \le \dots \le cm. \quad (8)$$

Cargo loading is a real world application of the MKP, [see Eilon and Christofides (1971)]. The problem is to choose some containers in a set of n containers to be loaded in m vessels with different loading capacities for the shipment of the containers. Other industrial applications are the loading of n tanks with m liquids that cannot be mixed, [see Martello and Toth (1980),] vehicle loading, [see Hifi (2009)], task assignment and multiprocessor scheduling, [see Labbé et al. (2003)]. The MKP problem is strongly NP-complete and the need for algorithms that give a good heuristic solution is justified by the computational complexity of this problem. Reference is made to Hung and Fisk (1978), Martello and Toth (1990) and Kellerer et al. (2004) for contributions to this problem.In this paper, we present a heuristic which yields a feasible solution within a reasonable computing time, this is done by exploiting efficiently the core of each knapsack. To the best of our knowledge, this paper is the first paper to deal with a heuristic since Martello and Toth (1980). Section 2 deals with principle of the Martello and Toth heuristic. The proposed method is presented in Section 3. The Section 4 is devoted to the presentation and analysis of computational results on randomly generated instances. Finally, in Section 5, we give some conclusions and perspectives of our work.

2.2.1 Martello and toth heuristic

In Martello and Toth (1980), the so-called MTHM heuristic is given for the MKP. This algorithm consists of three phases. The details are available on pages 179–181 in (Martello and Toth, 1990, pp. 179–181). The first phase of MTHM obtains an initial feasible solution by applying the Greedy algorithm (Procedure 1) to the first knapsack; a set of remaining items is obtained, then the same procedure is applied for the second knapsack; this is continued till the mth knapsack. The second phase tries to improve the initial solution by swapping every pairs of items assigned to different knapsacks and trying to insert a new item so that the total profit is increased. The last phase tries to exclude in turn each selected item if it is possible to replace it by one or more remaining items so that the total profit sum is increased. The advantage of the MTHM heuristic is that some items can be exchanged from a knapsack to another or excluded from the solution set so that total profit increases. This can lead to an efficient and fast solution when the solution given by the first phase is good. The main drawback of MTHM heuristic is that it considers only the exchanges between a pairs of items instead of combinations of items.

2.2.2 RCH, a recursive core heuristic for the MKP

The proposed heuristic called RCH, is a recursive method that performs computation on the core of knapsacks. The RCH heuristic is compared with the MTHM heuristic of Martello and Toth. Computational results on randomly generated instances show that the proposed approach gives better gap and smaller restitution times.The aim of the proposed approach consists in packing the maximum number of 'best' items with regards to relation (4) in a small computation time. For that purpose, an efficient

lower bound z of the MKP is built as follows. We consider the series of knapsack problems:

$$(KP_i)\left\{\{max\sum_{j\in N_i} px_i \mid \sum_{j\in N_i} w_j x_i \le c_i, x_j \in \{0,1\}\}, \in Ni, i=1\ldots..m\right. \quad (9)$$

With $N_1 = N$, $N_i = N_i - 1 - \{l \in N_i - 1 | x_l = 1\}$ and z_i the lower bound of (KP_i) such that $z = \sum x$

$_{ij} \in N_i p_{jj}$. Then, a lower bound of MKP is given as follows. $z = \sum_{i=1}^{m} z_i$

In order to simplify notation and without loss of generality, we consider that the elements of N_i are indexed from 1 to $Card(N_i) = n_i$, i.e., $N_i = \{1,\ldots,n_i\}$.

Procedure RCH
Entrée:N,$(p_j),(w_j)(x_j),c_i$
Sortie:Z;
Z=0;
//Traitement des m-1 premiers sac à dos//
For i :=1 to m-1 do ;
$Z_i = 0, c_i = c_i$
Procédure 1 sur c_i;
If $c_i > 0$ $then$
Définir le noyau C_i dec_i;
Procedure 2 sur SPP
End if
Mise à jours de N ;
Z=Z+Z_i ;
End for
//Traitement du dernier sac à doc//
Procédure 1 sur c_m;
If $C_i > 0$ $then$
Définir le noyau C_m de c_m ;
Procédure 3 sur KPC_m ;
End if
Z=Z+Z_m
end

3. MATERIALS AND METHODS

3.1 Presentation of the Tools Used

For our study, we used the following tools:

3.1.1 Google maps

This is an online mapping service created by Google allowing from the scale of a country, to zoom to the scale of a street.

3.1.2 Netbeans

Integrated development environment supporting various programming languages such as the one we used for the implementation of our application: Java [4].

3.1.3 Wampserveur

Windows-based web development platform for dynamic web applications using the Apache2 server, PHP scripting language, and a MySQL database. It also has PHPMyAdmin to more easily manage databases. We used it especially for its MySQL database [4].

.

3.1.4 EDraw max

EDraw Max is a software solution for making professional diagrams, graphics and drawings for many needs.

3.2 Methods

As far as we are concerned, we used the automatic processing method based on the creation of suitable software using the Java programming language and the MySQL DBMS. The following techniques were used: interview techniques which is verbal communication for gathering information through qualified people. The documentary technique which consists of consulting different works documentation, scientific publications, websites as well as related lecture notes. We present to you some orders from service stations across the city of Kinshasa to collect using Google Map software and GPS to process with a LENOVO DualCore Computer with a 500 Giga hard disk, 2 Giga Ram memory which also served us as implement our computerized system [4].

3.3 Collection of Filling Station Orders

We have orders for the product from petrol pump service stations located in the municipalities of Gombe and Lingwala:

3.4 Collection of Information from Tankers

We have harvested the following tankers:

4. DESCRIPTION OF RESULT AND DISCUSSION

4.1 Computer Modeling of the Application

For the modeling of our application, we opted for the Unified modeling language approach [5]. Indeed, the UML language makes it possible to realize a software system by representing it, by specifying it, by building it and by documenting it beforehand.

Please mention Ref no [6-16] after [5]

4.1.1 Identification of actors

With regard to the product delivery process, we were able to identify the following players: The dispatching manager; the customer; the marketer; the Operations Department and the Loading Order Service.

Table 1. List of stations collected

N°	Localization	Title	Quantity	Distance in (m^3)
1.	Gombe: Proche de la maison communale	Engen1	2	6
2.	Gombe: Proche du rond-point Forescom	Engen 2	10	8
3.	Gombe: Proche de l'hôtel de ville	Engen 3	12	11
4	Gombe : Proche de la maison Schengen, Avenue de la libération	Engen 4	4	3
5.	Gombe : 1 Avenue de la Justice, réf : En face de la cours suprême de la justice	Engen 5	12	5
6.	Gombe : Crois Avenue du Tombalbaye et Bakongo. Rondpoint Kin Mazière	Engen 6	7	13
7.	Gombe : Avenue colonel EBEYA	Engen 7	15	10
8.	Gombe : Croisement Kasaï et Haut-Congo	Sudoil 1	11	14
9.	Gombe : Proche de l'avenue KASAVUBU Avenue Haut Congo	Cobil 1	12	12
10.	Gombe : Avenue colonel EBEYA, réf : DG Airtel	Cobil 2	12	9
11.	Gombe : Avenue LUKUSA, réf : Entre Vodasquaire et Rond Point Forescom	Cobil 3	11	7
12.	Gombe : Avenu de la liberation, ref : Alimentation JIJI	Total 1	15	4
13.	Lingwala : Avenue de la liberation, ref :RTNC	Total 2	8	2
14.	Lingwala : Crois. Avenue huillerie et Boulevard Triomphale	Cohydro 1	4	1

Table 2. Tank trucks

No	Tank trucks	Capacity (m^3)
1.	K02	20
2.	K03	25
3.	K04	25

4.1.2 Identification of use cases

The following use cases have been identified: Ordering; Generation of BLs and Delivery Program; BLs compliance test and Delivery program; Conversion of BLs and delivery program into OC; Consult the availability of vehicles; Elaborate the rounds of tankers and establish the roadmaps

4.1.3 Presentation of the use case diagram

Based on the use cases and actors listed above we can build the use case diagram as follows:

4.1.4 Class diagram

4.1.4.1 Identification of classes

Please write 2-4 lines here.

4.1.4.2 Identification of relationships / association

For our work, we have identified the following relationships:

4.1.4.3 Presentation of the class diagram

Starting from the elements listed above, we can represent our class diagram as follows:

4.1.4.4 Transition from the class diagram to the logical pseudo-data model

To enable us to design our application database, we propose to design a logical pseudo model of the data starting from our class diagram using the rules for passing from the conceptual model to the relational model.

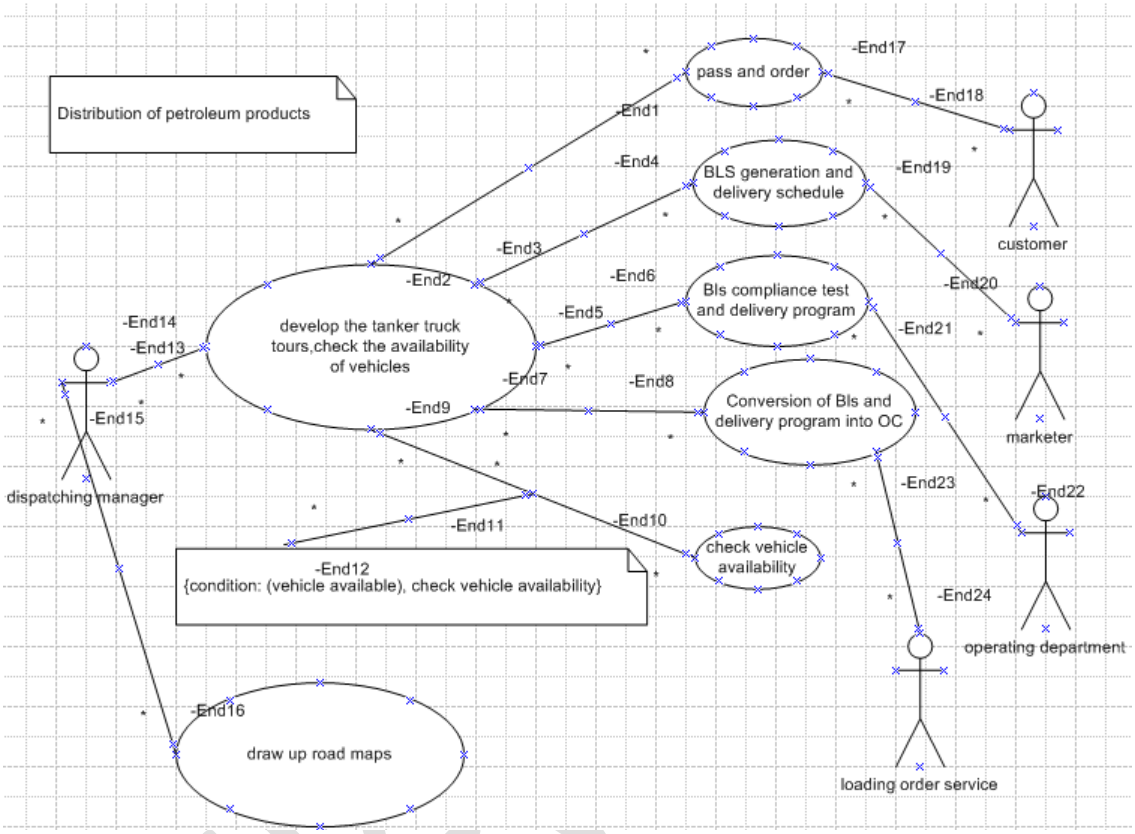

Fig. 1. Use case diagram

Table 3. Identification of classes

Classroom	Attributes	Attribute types	Methods
Station-service	- id_station - nom_stat - commune_stat - quartier_stat - avenue_stat - numero_stat	chaine chaine chaine chaine chaine chaine	Enregistrer () :void Supprimer () :void
Commande	- id_cmde - produit - quanté_cmde: - date_cmde	chaine chaine entier date	EnregistrerCmde () :void SupprimerCmde () : void
Camion-citerne	- id_cc: - num_plaque - capacité_cc	chaine chaine entier	EnregistrerCC() :void SupprimerCC() :void
ChargerCamionC	- id_chargerCC - quantite_chargeCC	chaine entier	EnregistrerCCC() :void AnnulerCCC() :void ModifierCCC() : void Pourcentage_charg() :void
Course	- id_course:chaine - libellé_course:chaine	chaine chaine	EnregistrerCourse () : void

- date_course: date	date

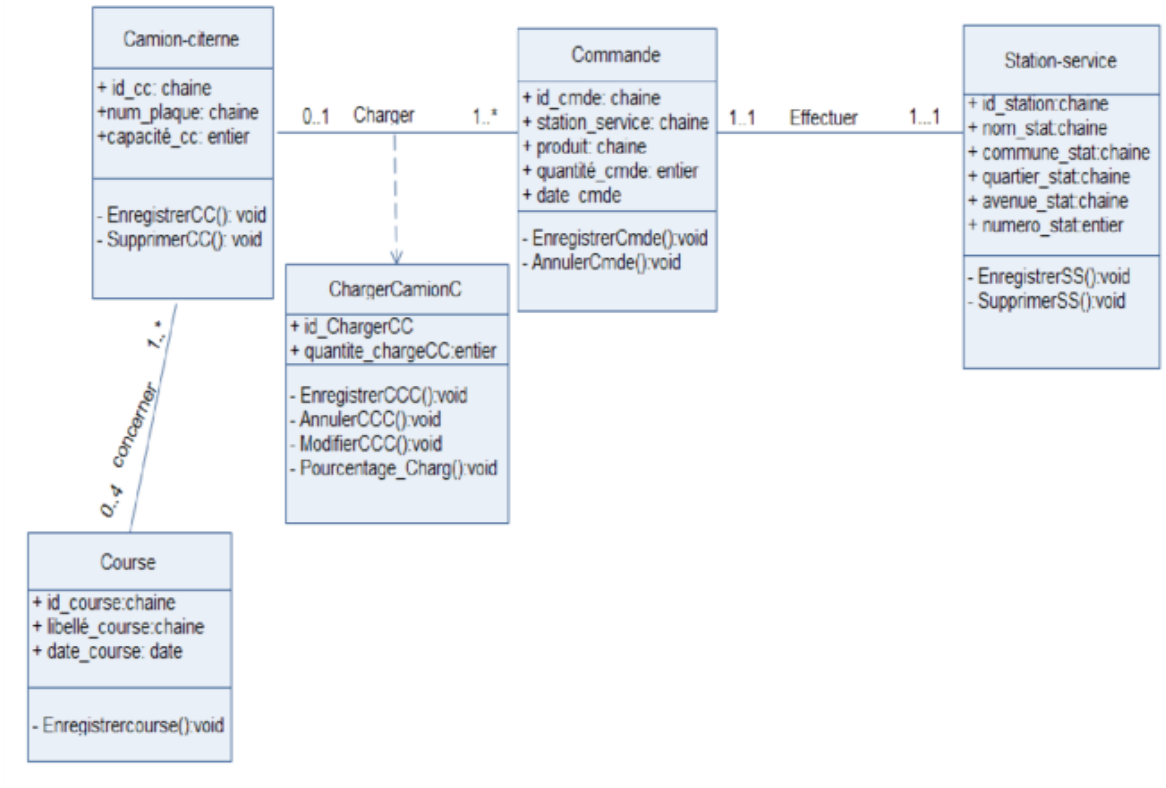

Fig. 2. Class diagram

4.1.4.5 Physical model of the data

The physical data model obtained from the logical data pseudo model looks like this:

4.1.5 Application interfaces

When we start our application, we have the following interface:

Table 4. Relationship identifier

Name of the relationship	Type of relationship	Related classes	Multiplicities
Effectuer	Association	Station-service et Commande	1…1 et 1…1
Charger	Association	Camion-citerne et Commande	1…* et 1…1
ChargerCamionC	Dépendance	Camion-citerne et Commande	
Concerne	Association	Course et Camion-citerne	1…* et 1…*

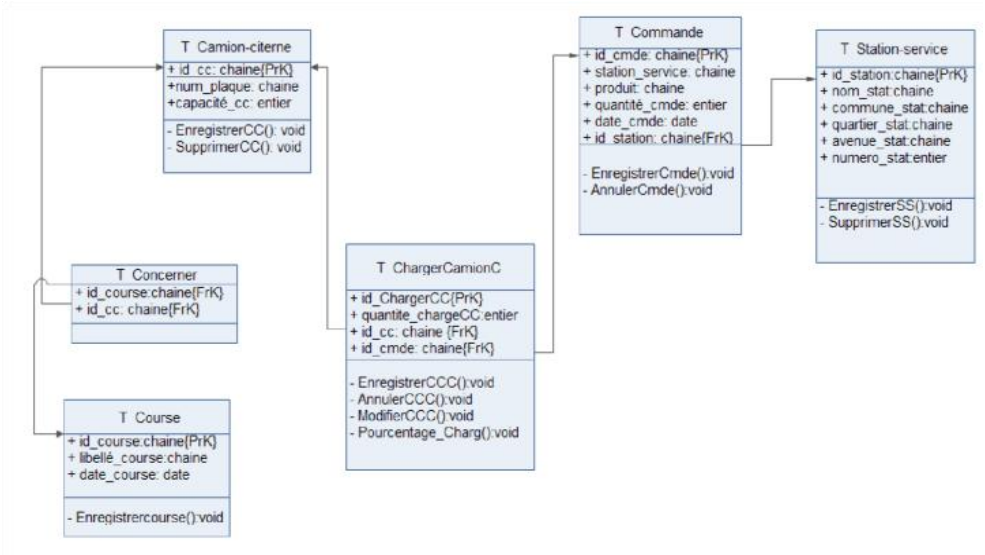

Fig. 3. Pseudo logical data model

Table 5. Physical model of the data

Class	Attributes	Types	Size
Station-service	- id_station	Varchar	4
	- nom_stat	Varchar	8
	- commune_stat	Varchar	20
	- quartier_stat	Varchar	20
	- avenue_stat	Varchar	20
	- numero_stat	Varchar	10
Commande	- id_cmde	Varchar	3
	- quantité_cmde:	Int	2
	- date_cmde	DateTime	15
	- produit	Varchar	6
	- id_station	Varchar	10
Camion-citerne	- id_cc:	Varchar	5
	- libelle_cc	Varchar	3
	- num_plaque	Varchar	10
	- capacité_cc	Int	2
ChargerCamionC	- id_chargerCC	Varchar	5
	- camion_Chrgt	Varchar	3
	- cmde_chrgt	Varchar	8
	- quantite_chargeCC	int	2
	- Date_chrgt	DateTime	15
Course	- id_course:chaine	Varchar	3
	- libellé_course:chaine	Varchar	9
	- date_course: date	DateTime	15
Concerner	-id_course	Varchar	3
	-id_cc	Varchar	5

Fig. 4. Authentication interface

It allows you to manage access to the application. Only dispatching officers are authorized to do so. After the user (the dispatching manager) has authenticated himself and clicked on the OK button, the Authentication window closes and then opens the main window of our application. It looks like this:

As you can see, this is a tabbed window. The default tab (Status tab) is displayed with two tables (order table and tanker table) initially empty. But after clicking on the "View" button, the stored data is loaded onto the tables. Indeed, this is the data received in the dispatching service to carry out the loading planning of the day. This planning is carried out in our second tab (Loading tab) which looks like this: Indeed, it is simply by clicking on the "Load" button that our application solves the loading problem. And display the results as follows:

4.2 Results Obtained

Let's take the results found above:

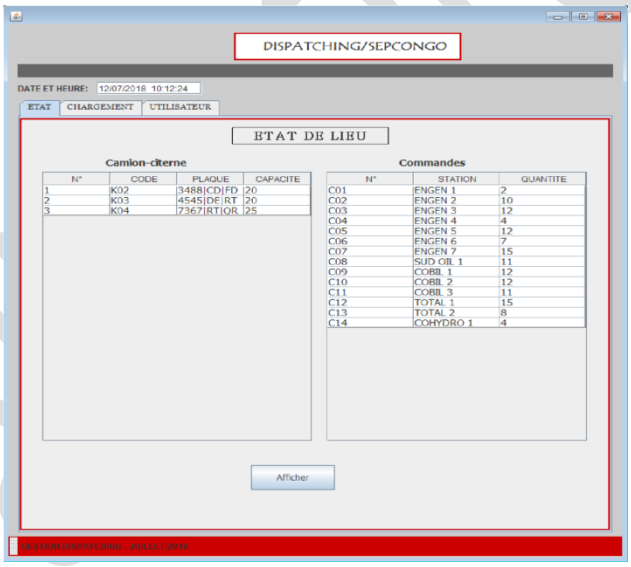

Fig. 5. Main window

Table 6. Table of results

No	Truck	Ordered	Amount	Date
1	K01	Engen 1	2	12/07/2018
2	K01	Engen 6	7	12/07/2018
3	K01	Sudoil	11	12/07/2018
4	K02	Cobil 1	12	12/07/2018
5	K02	Engen 4	4	12/07/2018
6	K02	Cohydro	4	12/07/2018
7	K03	Engen 3	12	12/07/2018
8	K03	Cobil 2	12	12/07/2018

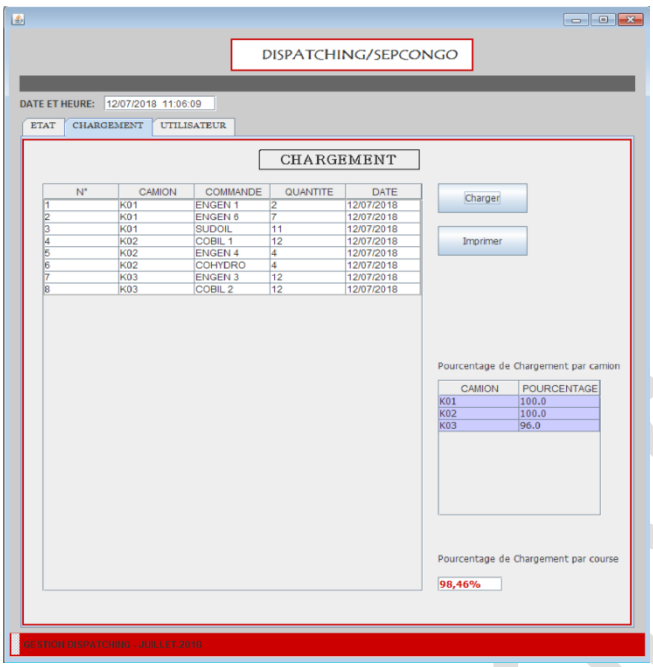

Fig. 6. Loading results

We obtained these results thanks to the RCH multiple backpack problem solving method having for: (w_j) the quantity ordered by the station and (p_j) the distance between the station and the SEP / CONGO depot at the head office such that the nearest station has the greatest (p_j) and (c_i) the capacities of tank trucks.Indeed, we can see that our planning using our app gave us a total load percentage of 98.46%. That is to say, for a total capacity of our 3 trucks of 65 m3, we managed to load the station orders for a total quantity equivalent to 64 m3. The orders not chosen will serve as the loading of the second race. These results are satisfactory for us and allow us to be reassured about the quality of our application, especially since they are obtained through an optimization process. We have limited ourselves to stations in the communes of Gombe and Lingwala, but in the real case, this is station for the whole city of Kinshasa. Certainly, for the dispatching manager it is difficult to carry out this planning and especially in an optimal way. Of course, he will be able to achieve even 100% schedules, but it will remain to be seen whether they are optimal. This is where our app comes in to optimize its decisions.

5. PROSPECT

As an extension of our research work, we will consider a possibility to explore: the evolution of our work concerns the application aspect of the proposal detailed in this document. Indeed, this work has been proposed in the context of handling large volumes of data. We lacked data especially of delivery of petroleum products in all provinces of the Democratic Republic of Congo to broaden our research. Therefore, our wish is that research can continue in this direction.

6. CONCLUSION

At the end of this study, We have implemented the tanker truck filling and dispatching systemof tankers: Case of SEP / CONGO. The objective was to realize a software allowing to automate the planning of optimal filling of tanker trucks by using the tools proposed by the Operational Research, more precisely of the combinatorial optimization resolution methods for the backpack problem. multiple: Recursive core heuristic (RCH). To do this, we started from a base on mathematical optimization in general, and combinatorial optimization in particular, while presenting the different methods of solving optimization problems. Then backpack problems by presenting its variant of the multiple backpack and its different resolution methods and by highlighting the RCH heuristic to understand how it works, before implementing our application while taking into account the prior analysis. The implementation of our application was done with the Java programming language under the Netbeans environment with the MySQL database under the WampServeur platform.The

results obtained by our application being satisfactory, we are very happy to realize that our objectives set from the start have been achieved. However, as it is said in economics: "the satisfaction of one need creates another need", the automatic realization of the loading of products has created other automation needs which we classify in perspective under the improvement. of our application such as: after loading, manage to automate the rounds of these trucks for the delivery of its orders to service stations leaving from a depot while minimizing the cost of transport, why not leaving from several depots,…. This can lead to other optimization problems, in this case: the transport problem, the traveling salesman problem and the vehicle routing problem.

DISCLAIMER

The products used for this research are commonly and predominantly use products in our area of research and country. There is absolutely no conflict of interest between the authors and producers of the products because we do not intend to use these products as an avenue for any litigation but for the advancement of knowledge. Also, the research was not funded by the producing company rather it was funded by personal efforts of the authors.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Fukunaga. A branch-and-bound algorithm for hard multiple knapsack problems". Annals of Operations Research. 2011;184(1):97–119.
2. Martin Jones, and all.,An Introduction to Political Geography,Space, Place and Politics; 2015.
3. Noël Sébastien. hybrid meta heuristics for solving the car scheduling problem in an automobile assembly line, Université du Québec, memoire, Chicoutimi. 2007;7-32.
4. Kafunda PK. In-depth operational research, University of Kinshasa; 2016.
5. Kasoro NM. Object-oriented programming, University of Kinshasa; 2015.
6. Holland JH. Genetic algorithms and the optimal allocation of trials, SIAM Journal of Computing. Flight. 1973;2(2):88-105.
7. Padberg M, Rinaldi G. A branch and cut algorithm for the resolution of large- scale symmetric traveling salesman problem. SIAM Review. Flight. 1991;33(1):60-100.
8. Hung MS, Fisk JC. An algorithm for the 0-1 multiple knapsack problem, Multiobjectives: Application to the Multidimensional Multiobjective Backpack Problem. Doctoral thesis. Hassan 2 University Casablanca; 2008
9. Mladenovic N, Hansen P. Variable neighborhood arc routing problem. Computers and Operations Research . Flight. 1997;34:1097-1100.
10. Korf RE, Fukunaga A. Bin completion algorithms for multicontainer packing, knapsack, and covering problems. Journal of Artificial Intelligence Research. 2007;28:393–429, 2007.
11. Balbal S. Using artificial intelligence to solve the backpack problem. Magister's thesis, Oran Mohamed Boudiaf University of Science and Technology; 2015.
12. Mohamed Esseghir Lalami and all,A procedure-based heuristic for 0-1 Multiple Knapsack Problems. Int. J. Mathematics in Operational Research. 2012;4(3).
13. Barshandeh S, Piri F, Sangani SR. HMPA: an innovative hybrid multi-population algorithm based on artificial ecosystem-based and Harris Hawks optimization algorithms for engineering problems. Engineering with Computers; 2020. Available:https://doi.org/10.1007/s00366-020-01120-w
14. An Algorithm for 0–1 Multiple Knapsack Problems, Available:https://www.researchgate.net/publication/229710958,Ming S. Hung,John C. Fisk
15. Available:https://epf.pub/an-introduction-to-political-geography.html
16. Available:http://www.or.deis.unibo.it/kp/Chapter6.pdf+5ressources