



HAL
open science

Dispersion optimization using a genetic algorithm

Simón Tamayo, Simon Tamayo, Thibaud Monteiro, Nathalie Sauer

► **To cite this version:**

Simón Tamayo, Simon Tamayo, Thibaud Monteiro, Nathalie Sauer. Dispersion optimization using a genetic algorithm. *International Journal of Advanced Operations Management*, 2009, 1 (2/3), pp.267-285. 10.1504/IJAOM.2009.030676 . hal-00580612v2

HAL Id: hal-00580612

<https://hal.science/hal-00580612v2>

Submitted on 10 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dispersion optimisation using a genetic algorithm

Simon Tamayo*

LGIPM EA 3096 / INRIA-COSTEAM,
ENIM – University Paul Verlaine – Metz,
Île du Saulcy, 57012 Metz – Cedex, France
and
Adents High-Tech International,
1 Allée des Garays, 91120 Palaiseau, France
E-mail: stamayo@adents.com
*Corresponding author

Thibaud Monteiro

LASPI EA 3059,
University Jean Monnet – Saint Etienne,
20 avenue de Paris, 42334 Roanne – Cedex, France
and
INRIA-COSTEAM,
Île du Saulcy, 57012 Metz – Cedex, France
E-mail: thibaud.monteiro@univ-st-etienne.fr

Nathalie Sauer

LGIPM EA 3096 / INRIA-COSTEAM,
ENIM – University Paul Verlaine – Metz,
Île du Saulcy, 57012 Metz – Cedex, France
E-mail: nathalie.sauer@univ-metz.fr

Abstract: The presented works concern the traceability phenomenon implementation within the production organisation, particularly in the field of raw materials management in the food industry. This project aims to minimise the raw material's dispersion in the manufactured products. We seek to solve the problem of raw materials allocation into finished products, in order to minimise its dispersion and moreover, the products recall if needed. The dispersion criteria are afterwards used to determine production's criticality in terms of sanitary risk, from which it is possible to optimise the processes of picking and dispatching. The final objective is to reduce the number of recalls in case of a crisis using the traceability information and the decision making tools. To solve the first part of the problem, a genetic algorithm is proposed.

Keywords: meta-heuristics; evolutionary algorithms; decision making; raw materials dispersion; traceability; logistic choices optimisation.

Reference to this paper should be made as follows: Tamayo, S., Monteiro, T. and Sauer, N. (2009) 'Dispersion optimisation using a genetic algorithm', *Int. J. Advanced Operations Management*, Vol. 1, Nos. 2/3, pp.267–285.

Biographical notes: Simon Tamayo is a Mechanical Engineer from the university EAFIT (Colombia) and Generalist Engineer from the National Engineering School of Metz 'ENIM' (France). After receiving his research Masters degree in Design, Industrialisation and Innovation from the schools ENIM-UPV-ENSAM, he has been doing PhD at the University of Metz in association with the Laboratory of Industrial Management and Production of Metz 'LGIPM', in the fields of traceability, production optimisation, decision making aid tools and algorithmics. He also works as a Research Engineer at Adents High-Tech International in Paris since 2007.

Thibaud Monteiro received his Engineer degree in 1998 from the National Engineering School of Tarbes (ENIT). He also obtained his PhD in Industrial Engineering from Institut National Polytechnique of Grenoble 'INPG' in 2001. He is currently Maître de Conférences at the University of Saint-Étienne. He manages the Healthcare Logistic axis of the Laboratoire d'Analyse des Signaux et des Processus Industriels 'LASPI' and is also an Associate Member of the INRIA Costeam team. He was co-author of eight articles in international revues. His main research interests include operational supply chain management for industrial and healthcare system, performance measurement and collaboration modelling.

Nathalie Sauer is Professor at University Paul Verlaine-Metz since September 2005. After having obtained her PhD degree from the University of Metz in 1994, she became Associate Professor at Ecole des Mines of Nantes. She obtained the ability to supervise research work (HDR) in 2004 at University of Nantes. She is currently team vice-leader of the Laboratory of Industrial Management and Production of Metz 'LGIPM', leading the Performance Evaluation research team of this laboratory and a member of the INRIA Costeam team. Her research interests include performance evaluation and optimization of production systems and scheduling problems.

1 Introduction

1.1 Industrial problem

Traceability is defined as 'the ability to find the history, use, or location of an item or an activity by a recorded identification' (ISO 8402). The traceability system in the farming and food supply chain can be described as the *documented identification of the operations which leads to the production and sale of a product* (Bertolini et al., 2006). It involves a close relation between the information flow and the existent manufacture conditions. Traceability is the process of finding a product or service since its creation (production) to its destruction (consumption); it is now an inescapable challenge for producers, they must establish accurate systems for effectively removing defective products from the market if a crisis occurs (GS1, 2008).

The food traceability (or food security), is now a crucial challenge to all stakeholders in the food supply chain. In this industry, the consideration and implementation of traceability has become an imperative requirement due to its direct relation with the sanitary performance and risk management. More generally, traceability is a major asset for the industrial safety improvement, because it impacts the ability to effectively manage logistics processes. However, the implementation of traceability systems requires a large

financial and human investment, and it may involve certain side effects in the management methods. The economic dimension of traceability must be evaluated considering the perspective of the increasingly costs of products recalls, as well as the cost of the defective products withdrawal, and their impact on the companies reputation.

Thus, the two main issues of traceability are, in one hand, the health security issue, and in the other hand an economic issue, these two matters are easily enforceable in the short term; and only a strategic vision of the problem can show the duality and the synergy arising from a successful integration of traceability in the management methods. It is important to ensure that traceability is no longer seen as an additional cover, but as a real competitive advantage. Therefore traceability solutions have reached an important stage of development. This research is held within the project of designing a deliveries optimisation unit, to be adapted to traceability software. The aim of this work is the reduction of the products recall cost, in case of a certain crisis by minimising the quantity of products to be recalled. The presented work was carried out as a PhD research project in agreement with a software provider company working in the industrial traceability field.

Traceability is a real industrial problem of research for two basic reasons:

- there is a real industrial need; due to the recent growth of commercial and legal constraints imposing the need to track products
- the current state of products in the market does not offer formal approaches and tools in order to exploit and take advantage of traceability information.

Despite the growing importance of this matter, very little formal work has been presented in this field. In the past, the risk impact and the ontological requirements of a traceability system have been studied (Borst et al., 1997). The relevance of traceability in both the external supply chain and inside the production system has been considered by (Ramesh et al., 1997). The need for traceability computerisation introduced previously has been studied by numerous authors like Sahin et al. (2002), who deem information technology to be the fundamental tool in the traceability of manufactured products. Several models of managing traceability information have been considered (Jansen-Vullers et al., 2003) as well as a general framework and a statement of experimental evidence (Regattieri et al., 2007) which have considered the different technological solutions in the traceability market. One of the few works concerning the problem of dispersion optimisation was developed by (Dupuy, 2004), a PhD thesis opening a new path in the consideration of traceability in production systems, proposing a model to optimise the raw materials avoiding dispersion. Unfortunately, the proposed approach is based on a linear model that can not solve the raw materials dispersion minimisation problems due to its unreasonable time of resolution.

In this context, the presented work seeks to design designing decision making tools which may take into account the traceability's stakes within a production management system. Exploit the related information and solve industrial problems in reasonable time. The intention of this article is to propose an optimisation model for the raw materials dispersion problem and a tool capable of performing this optimisation, this by using a meta-heuristic method as a genetic algorithm. The dispersion is a measure telling how scattered are the raw materials batches within the different finished products. Thus, for each batch of raw material used, it is possible to calculate how dispersed it is in the production, storage and distribution chains. The dispersion has a direct influence on the

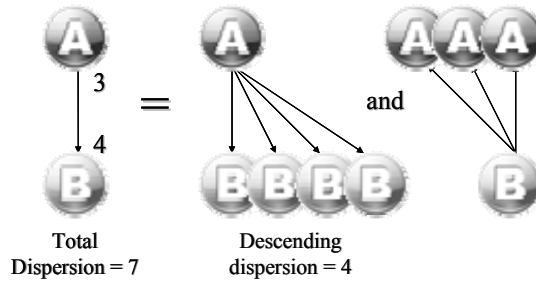
risks and costs of a possible sanitary crisis. However, the minimisation of this dispersion is a difficult problem which has not yet been solved nor integrated into any industrial tool.

We introduce a model of dispersion optimisation through the production chain, allowing us to determine the best allocation of raw materials batches to finished products in order to boost the production planning. A reduction of the batches physical dispersion represents a reduction in the batches mixtures, which will provoke smaller recalls in case of a defective raw material, as well as slighter recall costs and lower media impact. Initially a definition of the different types of dispersions will be introduced. Then the problem's model is presented, a heuristic for its optimisation is developed; this optimisation is based on a genetic algorithm. Finally, the given approach is summarised in a simple example and then conclusions and perspectives are remarked.

2 Definition of dispersion

Figure 1 illustrates the different types of dispersion defined as (Dupuy, 2004):

Figure 1 Example of dispersion



Note: total, descending and ascending

- The *descending dispersion* of a raw material batch is the number of finished products batches which contain a part of the raw material. For example, if a lot of ham is used in n lots of sausages, then the ham's descending dispersion is equal to n.
- The *ascending dispersion* of a finished product batch is the number of different raw material batches that it contains. As an example, a pork sausage product containing two different batches of shoulder and three different batches of fat has an ascending dispersion equal to five.
- The *total dispersion* of a system represents the summation of all batches ascending and descending dispersions, considering all raw materials and all finished products.

3 Problem statement

Minimising the raw materials dispersion directly provokes a reduction in the recalled products amount in case of crisis. To this end, one must determine the best allocation of raw materials batches throughout the production as well as the best way to assemble

the finished products from the semi finished products, in a multi-level production system. Therefore, it is necessary to distinguish the raw materials cutting conditions and the finished goods assembling information, which represent the problem's recipes.

In the food production field, two types of recipes must be taken into account:

- Disassembly or cutting recipes: Each type of raw material is cut in components of known and fixed quantities.
- Assembly recipes: Finished products are made of different materials which participate in the final product in known and fixed proportions.

The problem of minimising the dispersion consists on affecting the raw materials quantities in products, while respecting the cutting recipe, and in so affect the semi finished products quantities to finished products, considering the assembly constraints, thus it is a problem of assignment choice, in which one seeks to minimise the value of total dispersion. This quantities and recipes proportions are known inputs for the optimisation model as shown in Equation (1).

$$\min D_{TOT} = f(Q_{RM}, Q_{FP}, B_{RM}, B_{SP}, B_{FP}, T_{RM}, T_{SP}, T_{FP}, N_{CUT}, N_{ASS},) \quad (1)$$

in which:

Q_{RM}	raw materials quantities
Q_{FP}	finished products quantities
B_{RM}	number of raw material batches
B_{SP}	number of semi finished products batches
B_{FP}	number of finished products batches
T_{RM}	types of raw materials
T_{SP}	types of semi finished products
T_{FP}	types of finished products
N_{CUT}	cutting recipe
N_{ASS}	assembly recipe.

The function D_{TOT} is the summation of all the individual dispersions for each raw material batch and each finished product batch, as it is presented in Equation (2).

$$\min D_{TOT} = \sum_{i=1}^{B_{RM}} \sum_{k=1}^{B_{FP}} D_{i,k} (Q_{RM}, Q_{FP}, B_{SP}, T_{RM}, T_{SP}, T_{FP}, N_{CUT}, N_{ASS},) \quad (2)$$

$D_{i,k}$ represents the binary value of dispersion for each individual couple 'raw material / finished product', which is zero if the material is not present in the finished product, or otherwise is one.

4 Complexity of the problem

The dispersion minimisation problem belongs to the NP-complete class. The resolution of this problem in polynomial time results in the resolution of any polynomial problem NP in polynomial time (Garey and Johnson, 1979). An estimation of the problem's complexity shows that the total number of possible combinations is:

$$C = E * \sum_{j=1}^{j=B_{RM}} \left((Q_{i,j})^n \Big|_{i=0}^{i=Q_{max}} \right) \quad (3)$$

In which:

- C number of combinations to explore
- E number of production levels
- Q_{max} largest assigning quantity
- $Q_{i,j}$ quantity of raw material to assign
- B_{RM} number of raw material batches
- n number of products to produce (in the actual production level).

The difficulty of the problem is revealed, the complexity equation (see Equation 3) is the same as the one for the 'graphs colouring problem' (Wegener, 2005), and since this is an NP-complete problem one can conclude that the problem of raw materials dispersion optimisation belongs to the same group. Since this is a NP-complete problem, the existence of a solution algorithm of polynomial complexity remains unknown (Palekar et al., 1990). Therefore, a meta-heuristic method, and more specifically a genetic algorithm was proposed in order to solve this problem. These algorithms are particularly suitable for combinatorial optimisation problems including the so called NP-complete problems (Barthelemy, 2001).

5 Proposed solution based on genetic algorithms

5.1 Genetic algorithms literature

Genetic algorithms are optimisation algorithms based on genetics and evolutionary theories inspired by nature and its mechanisms as reproduction, mutation, selection, etc. (Coeque et al., 1999). They are known as evolutionary algorithms.

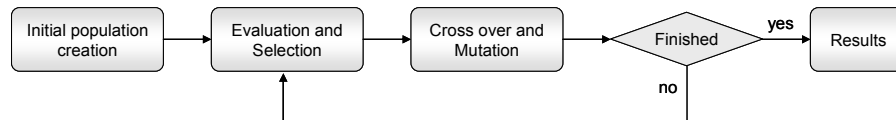
The application of genetic algorithms in combinatorial problems begun with the works carried out by Holland (1962) on adaptive systems in 1962. The research done by Goldberg (1994) has specially contributed to its diffusion. These algorithms can have a large field of application, some examples of these meta-heuristics and their hybrids can be found in the following works, Holger and Stützle (2004), Siarry and Michalewicz (2007) and Talbi (2006).

The genetic algorithm seeks to make evolve an initial set of solutions towards the optimum (or optima) of an optimisation (Bontemps, 1995). This set of solution points represents population containing characteristics subject to genetic operations as:

- *Evaluation:* To each solution an adaptation note is given, this note corresponds to the solution's level of adaptation to the problem. The evaluation will determine the 'fitness' of individuals (i.e., the value of the objective function) of each solution. In particular, the higher the fitness, the more the genes of a solution are likely to be propagated to the next iterations (Naso et al., 2007). In the presented case, this note is equal to the total dispersion associated to the allocation choices.
- *Selection:* To determine which individuals are closest to achieving the best results, a selection is made in the same way as natural selection occurs; the strongest individuals in terms of dispersion tend to win the reproduction competition, while weaker individuals tend to die before breeding. Similarly, some parents may be able to survive their next generation.
- *Cross over:* It aims to combine two parents in order to produce new individuals who inherit some characteristics from the parents.
- *Mutation:* Is a random alteration of the individual composition, which in some cases leads to its degeneration. Mutation may help the population escape from local optima.

Figure 2 shows the general functioning of genetic algorithms.

Figure 2 Working principle of genetic algorithms



The set of 'solution individuals' evolve gradually as generations advance, during each new generation the stages of evaluation, selection, reproduction and mutation are performed and renewed until a termination condition determines the end of the process. Common terminating conditions are presented (Drakos, 1994):

- Fixed number of generations reached. This criterion is interesting when the time is limited.
- The highest ranking solution's fitness is reaching or has reached a stage such that successive iterations no longer produce better results. Then one can suppose that the population has reached the optimum or its neighbourhood.

5.2 Application to the raw materials dispersion optimisation

Given the problem's configuration, the initial population individuals are represented by a set of matrices describing the allocation of raw materials to semi finished and finished products, according to the different levels of production and fabrication recipes. These individuals are randomly generated. When the initial population is created, it is subject to evolution as shown before.

The initial population

In order to create a population well adapted to our situation it is necessary to consider the following characteristics of the production to optimise:

- the different types and quantities of raw materials, semi finished products and finished products
- the cutting and assembly recipes.

From the problem data, one must initially create binary participation matrices, these contain the positions of assignment between two consecutive levels of the solution object (Figure 3A). Afterwards random quantities are assigned respecting the problem's boundary conditions (determined by the quantities and recipes defined above). The result is a matrix, called assignment matrix, connecting two production levels, and indicating for each assignment option the corresponding quantity amount (Figure 3B).

A matrix is created each time that products are cut or assembled, thus there will be a many matrices as production levels minus one. Since the matrices creation is done randomly it is important to check the validity of the solution. For example, if a material quantity is sent to a semi finished product in one stage of production (in the first matrix of the solution), then the solution part representing the next stage must consider the quantity of material already assigned before, so the matrices representing each level are coherent between them in order to ensure that the sum of total material quantities remains unchanged. If this is not the case, the solution has to be repaired.

Figure 3 Participation and quantities assignment matrices

A) Participation matrix (raw materials - sub products)

	SP1	SP2	SP3	SP4	SP5	SP6
RM1	1	0	1	0	0	1
RM2	0	1	1	1	0	0
RM3	1	1	0	1	0	1
RM4	0	0	1	1	0	0

B) Quantities assignment (raw materials - sub products)

	SP1	SP2	SP3	SP4	SP5	SP6
RM1	Q_{11}	0	Q_{13}	0	0	Q_{16}
RM2	0	Q_{22}	Q_{23}	Q_{24}	0	0
RM3	Q_{31}	Q_{32}	0	Q_{34}	0	Q_{36}
RM4	0	0	Q_{43}	Q_{44}	0	0

In case of subcontracting a part of the production, other assignment matrices are created regarding the assignments of external providers. The algorithm prioritises internal production when allocating. All generated matrices are assembled and the entire set of coherent matrices is defined as an individual of the population to evolve.

Evaluation

In order to evaluate each individual, the total dispersion value is counted (Figure 1). For each raw material batch BRM_i one must check if it is present in each finished product BFP_k ; if it is, then the dispersion value Dik is 1. To achieve this we must count the number of times that the raw material participates in the semi finished products, then we count each time that these semi finished products participate in the products of the following production level, and so on until identifying all the possible paths that raw

materials are drawing to get to the finished products; then if one of these paths is used in the given solution, then the partial dispersion related to this path is equal to 1. Finally, all the partial dispersions must be added in order to obtain the total dispersion value associated to the individual (Equation 2). After the evaluation, a set of dispersion values is obtained. These values indicate the progress in the population's evolution, and are the main parameter to consider when the selection takes place.

Selection and elitism

In the proposed algorithm two methods of selection are presented:

- ranking selection
- random selection + elitism.

Ranking selection first sorts the population by dispersion value. Then adaptation values are assigned to the individuals. The adaptation value is inversely proportional to the individual's ranking and it is used to determine the number of reproductions that the concerned individual must perform. For a higher adaptation value, there will be a greater the number of reproductions. This is how more efficient people (in terms of raw materials dispersion) will arise in the next generation, and those less successful will have a greater probability of not surviving.

The random selection + elitism, randomly determines the reproduction probabilities, but applying an elitist generation mixture at the end of the reproduction process. The elitist mixture consists in copying one or more of the best 'parent individuals' in the new generation. The other individuals of the population are normally generated by reproduction.

When the reproduction probabilities are randomly assigned, elitism is necessary for assuring the algorithm's convergence. This form of selection performs a slower convergence, but it provides a more comprehensive exploration of the solutions space.

In all cases, during the creation of a new population, it is possible that the best chromosomes are lost (after the hybridisation and mutation operations). To avoid this, the elitism method can always be applied.

Cross over

The hybridisation probability (frequency of the hybridisation occurrence) is 100%, i.e., that the entire set of children is produced by hybridisation (elitist copies are not counted in this group). To accomplish the reproduction, hybridisation couples are randomly generated respecting the number of reproductions assigned to each individual. The choice of couples is non-elitist. Once the list of couples is established, the children are created. Each new individual inherits 50% of his genes from each parent (see Figure 4). The advantage of using a 50% crossing rate is that the inherent constraints of nomenclature are respected, therefore with this rate there is no need to repair the children (for possible degenerations) after each reproduction. However, the crossing rate may be changed if more effective crossing operators are implemented (Talbi, 1999; Holger and Stützle, 2004).

Figure 4 Children generation (see online version for colours)

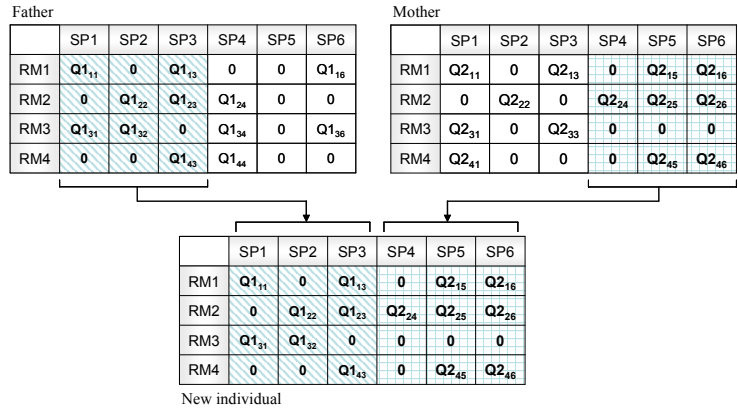
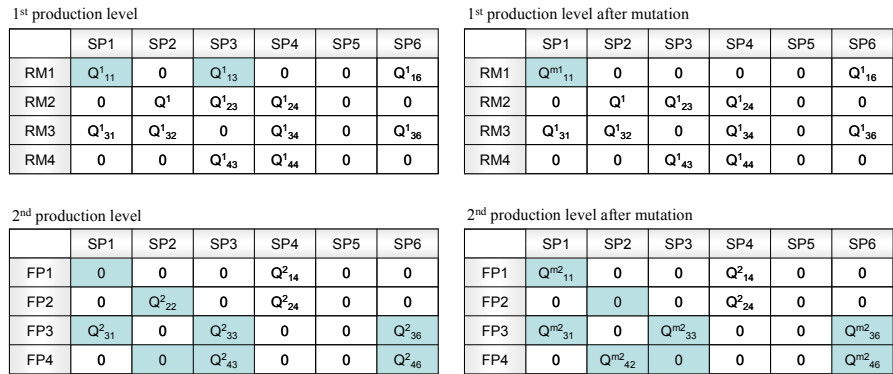


Figure 5 Mutation example (see online version for colours)



Mutation

Three different mutations have been conceived:

- a position change for a raw material quantity of assignment
- a systematic degeneration of assignments (increasing the selection choice in order to mount the dispersion value)
- a random alteration of quantities.

After testing the three types of mutation in different problems (see plan of experiences, Table 3), the third type of mutation was selected as the most convenient (altering quantities randomly). This mutation is certainly degenerative due to the very high complexity level of each solution. Altering a single gene possibly leads to change the

whole matrices of the solution in order to ensure its validity. The mutation consists on randomly changing the quantities distribution in the first level of production (the solution's first matrix, which concerns the raw materials and the first set of semi finished products) and afterwards checking the validity of the ulterior levels which eventually may be corrected so they keep respecting the problem's constraints (see Figure 5).

5.3 Setting the algorithm's parameters

Defining the parameters of a genetic algorithm has an impact on its performance and adaptability to a specific problem. According to the characteristics of the problem to be solved, sometimes it is necessary to modify these parameters. These variables are taken into account at the beginning of the problem's resolution in order to fix the criteria related to the population and its evolution (Bourda, 1997). These parameters are determined by a plan of experiences (see Tables 3 and 2) and they are specific to a given problem.

Population parameters

These parameters directly affect the population; they determine the operations that will repeat along the iterations:

- Number of population individuals: This parameter determines the number of individuals randomly generated to populate the first generation. It is also the number of individuals remaining at the end of each iteration. This number of population individuals remains constant: there are as many deaths as there are births in each step.
- Pre-selection population: There is a choice of creating a larger population at the beginning and perform a pre-selection procedure in order to start the algorithm with a set of more suitable individuals. This larger pre selection population is also randomly generated and ranked by fitness. The pre selection feature increases the algorithm's speed of convergence (Gao and Hu, 2006).
- Mutation rate: This parameter defines the number of generations separating two mutation phases. When a generation is mutated, the alterations are arbitrarily made. Due to the chosen mutation type, it is possible to obtain no changes in the individual after a mutation. All the individuals have the same probability of mutate.
- Elitism rate: Represents the quantity parents to be kept in the next generation in order to preserve the best individuals. It is imperative to have a rate of elitism different to zero when the selection is made at random, otherwise the concept of evolution is not guaranteed.

Setting the terminating conditions

There is a large variety of termination conditions. In our case, we use two as shown in Section 5.1:

- Limit the algorithm's iterations to a maximum number of generations in order to control the computation time.

- Stop the algorithm when successive generations no longer produce better individuals. For this the N last successive answers are considered, and the algorithm continue turning if one there is still a variation in them (N is a variable parameter directly proportional to the mutation rate).

6 Plan of experiences

6.1 Proposed approach

To evaluate the GA's applicability in different production configurations, several problems were considered (always seeking the raw materials dispersion optimisation). Initially a problem of hams manufacture was solved (furnished by the Aosta group), for which the algorithm has shown a great adaptation capacity, offering very convenient results in reasonable time. However, this is only one sample of a very large choice of configurations. There may be variations in the production sizes, the recipes and the types of fabrication. For this reason, a series of different problems was created. These problems are mainly categorised by two main factors:

- *The raw materials total quantity*, which is the amount of material to be produced. It can be quantified by adding all the quantities from the incoming raw materials batches that will be distributed according to the cutting recipe. This factor directly determines the computing time. If we have a constant minimal assignment unit, and the raw materials quantities increase, then the number of possible cases to assess increases in a direct relation. For example, if the minimal assignment unit of is 1kg and a quantity of a given raw material batch increased from 500 kg to 1,000 kg, then for this particular batch there will be twice as many possible distribution solutions.
- *The number of batches to assign*. In the case of a three levels production configuration, the amount of batches may vary in any of the three levels, raw materials, semi finished products, finished products or it may vary in all of them. Increasing in each case, the complexity level of the manufacturing recipes. In the presented model, an augmentation of the batches number provokes directly an increase in the number of constraints to be respected, and for the same reason, the proposed heuristic method will have more difficulty in its solutions generating process (the first and slower step in the algorithm).

Table 1 Classification of the 50 evaluation problems (see online version for colours)

		<i>Raw materials total quality</i>	
		<i>Large</i>	<i>Small</i>
Number of batches	Large	Type 1 15 problems	Type 2 5 problems
	Small	Type 3 5 problems	Type 4 25 problems

The raw materials total quantity has been classified in two groups of problems, those with large quantities and those with small quantities. The so-called 'small' are those whose

total amount of raw materials is lower or equal to 7 ton and the so-called ‘large’ are the ones that exceed this amount.

The number of batches to assign is also classified as ‘large’ or ‘small’. Small problems (in terms of number of batches) are the ones with less than 20 assignment batches, and the large problems are those with a higher amount.

According to these two criteria 50 problems were created and classified in types 1 to 4 (see Table 1).

In order to evaluate the performance of the proposed tool and its advantages towards the exact methods, these problems have been solved with an iterative method MILP, which stands for mixed integer linear programming (Wang et al., 2008), using the LpSolve software (version 5.5.0.13) and also using the developed genetic algorithm. The algorithm was tested with different parameters combinations in order to determine the best suited to each type of problem. The final goal of this approach is to obtain a configured algorithm capable of solving a problem representative of a real production in terms of raw material quantity and number of batches, as mentioned earlier.

To find the best algorithm’s configuration (the set of parameters that best suits to each problem), a simulation was performed, in which each problem was solved with different parameters combinations seeking to determine the optimal configuration as it is presented in the Table 2. In the simulation 1,728 different configurations of the algorithm were evaluated for each one of the 50 problems.

Table 2 Parameters variation in the configuration simulation

<i>Parameter</i>	<i>Min</i>	<i>Step</i>	<i>Max</i>
Nb of population individuals	30	30	250
Elitism rate	0%	5%	10%
Mutation rate	1	5	11
Pre-selection population	1%	1%	3%
Nb of generations	100	300	1,000
Type of selection	TS1 = Rank	---	TS2 = Random

Note: a total of 1,728 different combinations

6.1 Discussion and industrial statement

After performing the comparison between the MILP method and GA, we conclude that for the problems of types 1 and 2, the best configuration of the algorithm is presented as the heuristic #1 (see Table 3, the problems of type 3 can be solved better by the parameters composing the heuristic #2; and the problems of type 4 can be solved better using the heuristic #3. However, if the algorithm is used as an industrial tool, it is interesting to launch parallel heuristics; it is also advisable to make several releases of the same configuration. As several principles of the evolution are based on random, the progress in results may vary from one population to another even for the same set of parameters. Once a type of production is categorised by the two main factors (quantity and number of batches) its optimal dispersion can be obtained using the genetic algorithm with the corresponding heuristic configuration, afterwards this value of optimal dispersion compared to the real dispersion value embodies an important indicator of the production’s risk level, and becomes capital in future optimisations (Tamayo et al., 2009).

Table 3 The three best suited Heuristics to the different types of problems

	Pre-selection individuals	Number of individuals	Elitism rate	Mutation rate	Number of generations	Type of selection	Type of problems
Heuristic #1	2%	210	5%	1	1,000	Ranking	1 and 2
Heuristic #2	1%	120	0%	1	1,000	Random	3
Heuristic #3	No pre-selection	50	0%	1	No longer evolution	Ranking	4

Figure 6 Conversion graphs for two different problems

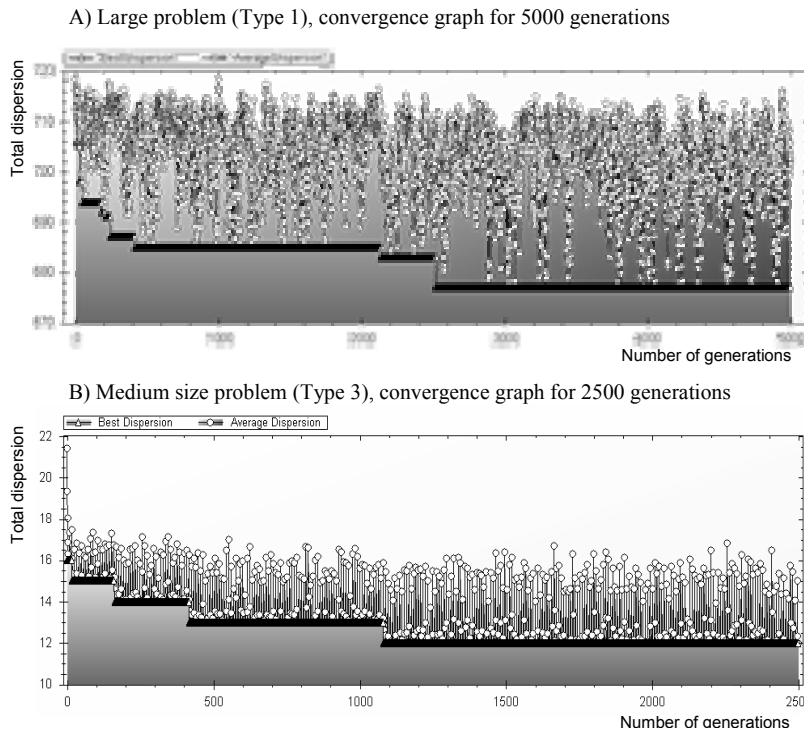


Figure 6 shows the algorithm’s behaviour for two different problems of type 1 and 4, for an optimal parameters configuration (obtained after the simulations). Figure 6A presents a final dispersion result equal to 677, found after 1,870 generations and 60.7 seconds of calculation time. For the same problem, the MILP method did not offer any better answer in 15 days of computation (this processing time corresponds to a dual core processor running at 3.01 Ghz, and with 2 Go of RAM capacity). Figure 6B presents a dispersion result equal to 12, found after 1,260 generations for a computing time of 2.3 seconds.

7 Numeric example

In order to illustrate the functioning of the algorithm and the influence of its parameters variation in the results, we propose a solution for a numerical example. This example comes from a real industrial situation and was proposed by the AOSTE group, which is a French meat manufacturer (Dupuy, 2004). Primarily, we tried to solve the problem using the MILP method (modelling the problem in the software ‘Lp_solve’), and after 15 days of processing, this method had not found the optimum yet, the best solution obtained at that moment was taken into account as a reference of what would be an acceptable solution value (in order to measure the performance of the genetic algorithm). For the algorithm’s solution, the parameters simulation was launched (as defined earlier in Table 2). The example considers a production of four raw material batches divided in two types, six batches of semi-finished products also divided in two types, two batches of subcontracted semi-finished products (one of each type), and four batches of finished products divided in two types as well.

Figure 7 Real industrial example of dispersion optimisation

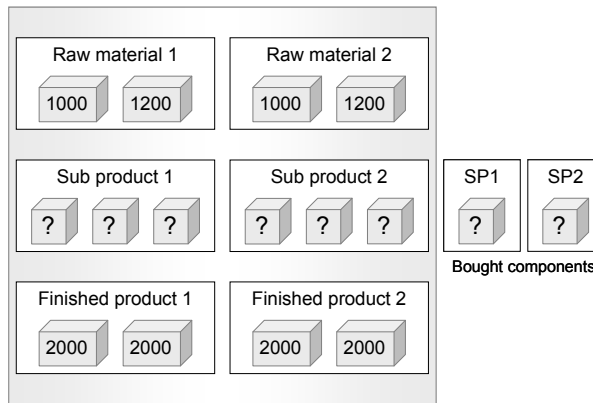


Figure 8 Cutting and assembling nomenclatures (recipes)

Cutting recipe			Assembling recipe		
	SP1	SP2		SP1	SP2
MP1	60%	40%	PF1	50%	50%
MP2	45%	55%	PF2	70%	30%

The quantities and names are given in Figure 7, the amounts to cut or assemble are specified inside each batch, and the cutting and assembling recipes are shown in Figure 8.

In order to provide an idea of the problem’s size, its complexity is analysed by calculating the number of different solution combinations, as proposed in the Equation (3). The number of different solutions to this example is:

$$600^3 + 400^3 + 720^3 + 480^3 + 450^3 + 550^3 + 540^3 + 680^3 + 1000^4 + 1000^4 + 1400^4 + 600^4 = 5.973e^{12}$$

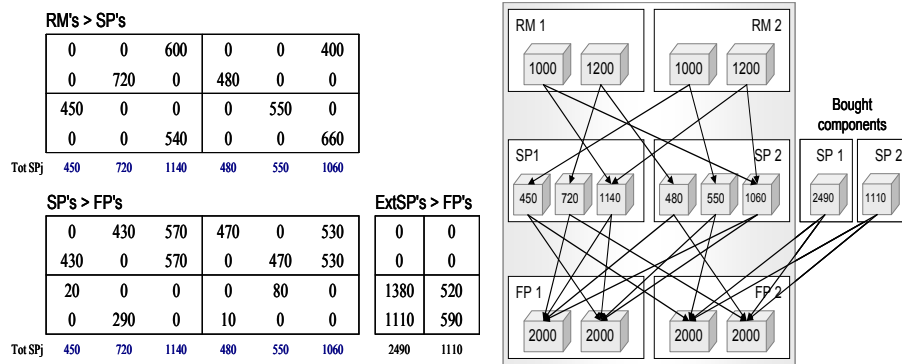
If a computer can obtain one hundred results each millisecond, then the number of years that it would take to calculate all the possibilities is given by:

$$5.973e^{12} * \frac{1}{100} * \frac{1}{1000} * \frac{1}{60} * \frac{1}{60} * \frac{1}{24} * \frac{1}{365} = 1.894_{years}$$

7.1 Results

The solution using the MILP method was stopped after 15 days of processing (under the same hardware conditions as the presented in Section 6), the result was a solution with a total dispersion equal to 12. The proposed genetic algorithm finds an answer of total dispersion equal to 12 in 1.3 seconds of processing (using the same machine). The obtained solution is detailed in the Figure 9. This result is more convenient than the one obtained with the truncated exact method, the processing time used by the GA is remarkably better. Thus the performance of this meta-heuristic is confirmed in the solution of this example, and the possibility of obtaining real time dispersion optimisation has been proved.

Figure 9 Solution to the evaluation example (see online version for colours)



Note: Dispersion = 12, time = 1.3sec

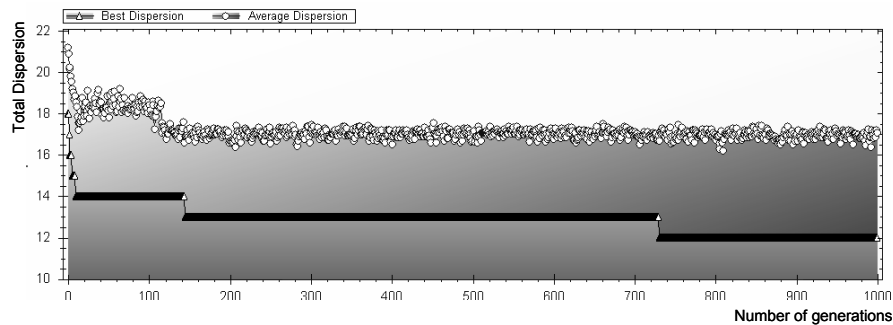
A rather fast convergence is observed (Figure 10); the algorithm tends towards an acceptable solution around after 730 generations. Figure 10 shows the evolution of the population towards the optimum. It is possible to have very fast convergence due to the fact that the population procreates a ‘great individual’. The disruptions caused by mutations can be identified, and also the fact that mutation can help a group of individuals to escape from a local optimum. As an example, in this particular case, the research was blocked in a local dispersion optimum equal to 14, and after being damaged by the action of mutation, a more interesting dispersion value was found.

The best set of parameters for this example are: An initial population of 50 individuals, with no pre selection population, a selection performed by ranking, a rate of elitism equal to 2%, a mutation rate equal to one (i.e., a mutation possible for each

generation), and a maximum number of generations equal to 1000 (as a termination condition).

The presented example is small; it has converged quickly compared to some examples of the genetic algorithms literature. We must not forget that this type of problem can easily become complicated when several manufacturing levels are taken into account, or when the problem size boosts (in terms of quantities or number of batches), and in those cases the convergence will be retarded. The example studied in this paper was chosen for its simplicity and because other authors have already worked on it.

Figure 10 Conversion graph, minimal dispersion and mean dispersion



Note: Numerical example

8 Conclusions and perspectives

In this work, we studied the problem of optimising the raw materials dispersion in finished products within the food manufacturing field. In terms of traceability, this dispersion has a fundamental impact on the producers' industrial performance, even though the subject has been rarely taken in consideration before.

A model for the dispersion minimisation has been proposed. This problem is very complex and it can not be solved using an exact method in a reasonable time. Thus a meta-heuristic method was suggested in order to obtain a solution in a convenient time, more specifically, an adaptation of genetic algorithms has been proposed to solve the problem. The solution process by genetic algorithms was tested on several examples (considering different production sizes and configurations), and the results showed great potential for applying the model.

In the field of dispersion optimisation, there is still a lot to be written; even if the proposed algorithm is capable of finding solutions in reasonable time, this kind of meta-heuristic can always be refined in several ways.

During the plan of experiences; some cases have shown a very fast convergence. This might be suggesting that the population some times engenders a 'great individual'. Since this individual is selected too often, the population tends to converge towards its genome. In this specific case, 'the diversity of the population's genetic pool is too small for the algorithm to progress' (Rennard, 2006). Thus, there is a perspective of developing methods in order to detect and avoid this scenario.

In the future, it seems pertinent to design a method for scattering the initial population seeking to reach all the areas of the solution space. Moreover, in the presented application the algorithm considers the optimisation of a simple dispersion function, it takes into account the discrete participation of the raw materials into the finished products, but without considering the quantities involved, in the case of minimising sanitary risk the actual function is perfectly adapted because if a dangerous raw material is present in a finished product, even in a very small proportion, the finished product will still be dangerous, but for future applications of dispersion optimisation in other fields than the food manufacturing, the consideration of quantity rates in the dispersion value may be crucial. One important perspective consists on reviewing this evaluation method.

In terms of logistics, producers often use multiple optimising objectives. Therefore, it can be interesting to develop the presented model applying multi-objectives evolutionary algorithms.

References

- Barthelemy S. (2001) 'Genetic algorithms and financial crises in emerging markets', *18th International Conference in Finance*, AFFI, Namur, Belgium.
- Bertolini, M., Bevilacqua, M. and Massinia, R. (2006) 'FMECA approach to product traceability in the food industry', *Food Control*, Vol. 17, pp.137–145.
- Bontemps C. (1995) *Principes Mathématiques et Utilisations des Algorithmes Génétiques*, INRA, Toulouse.
- Borst, P., Akkermans, H.M. and Top, J. (1997) 'Engineering ontologies', *International Journal of Human Computer Studies*, Vol. 46, Nos. 2–3, pp.365–406.
- Bourda, Y. (1997) 'Appréhender les algorithmes génétiques grâce à des exemples concrets', SUPELEC.
- Coueque, Y., Ohler, J. and Tollari, S. (1999) 'Algorithmes génétiques pour résoudre le problème du commis voyageur', Département de mathématiques appliquées, Université de Macquarie, Sydney.
- Drakos, N. (1994) 'Synthèse de comportements animaux individuels et collectifs par Algorithmes Génétiques', Computer Based Learning Unit, University of Leeds.
- Dupuy, C. (2004) 'Analyse et conception d'outils pour la traçabilité de produits agroalimentaires afin d'optimiser la dispersion des lots de fabrication', Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, France.
- Gao, L. and Hu, Y. (2006) 'Multi-target matching based on niching genetic algorithm', *IJCSNS International Journal of Computer Science and Network Security*, Vol. 6, No. 7A.
- Garey, M. and Johnson, D. (1979) *Computers and Intractability: a Guide to the Theory of NP-completeness*, W.H. Freeman & Company.
- Goldberg, D. (1994) *Algorithmes génétiques*, Addison-Wesley, France.
- GS1 France (2008) 'Solutions logistiques', www.gs1.fr.
- Holger, H. and Stützle, T. (2004) *Stochastic Local Search-Foundations and Applications*, Morgan Kaufmann, San Francisco, CA, USA.
- Holland, J.H. (1962) 'Outline for logical theory of adaptive systems', *Journal of the Association of Computing Machinery*, Vol. 3.
- Jansen-Vullers, M.H., Van Dorp, C.A. and Beulens, A.J.M. (2003) 'Managing traceability information in manufacture', *International Journal of Information Management*, Vol. 23, pp.395–413.

- Naso, D., Surico, M., Turchiano, B. and Kaymak, U. (2007) 'Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete', *European Journal of Operational Research*, Vol. 177, pp.2069–2099.
- Palekar, U.S., Karwan, M.H. and Zionts, S. (1990) 'A branch and bound method for the fixed charge transportation problem', *Management Science*, Vol. 36, pp.1092–1105.
- Ramesh, B., Powers, T., Stubbs, C. and Edwards, M. (1997) 'Requirements traceability: theory and practice', *Annals of Software Engineering*, Vol. 3, No. 1, pp.397–415(19).
- Regattieri, A., Gamberi, M. and Manzini, M. (2007) 'Traceability of food products: general framework and experimental evidence', *Journal of Food Engineering*, Vol. 81, No. 2, pp.347–356.
- Rennard, J.P. (2006) *Handbook of Research on Nature Inspired Computing for Economics and Management*, Section 1, Chapitre 4, Grenoble Graduate School of Business, Grenoble.
- Sahin, E., Dallery, Y. and Gershwin, S. (2002) 'Performance evaluation of a traceability system', in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, ISSN: 1062-922X, pp. 210–218.
- Siarry, P. and Michalewicz, Z. (2007) *Advances in Metaheuristics for Hard Optimization*, Springer-Natural Computing Series.
- Talbi, E.G. (1999) 'Metaheuristiques pour l'optimisation combinatoire multi-objectif. État de l'art', Rapport CNET (France Telecom).
- Talbi, E.G. (2006) *Parallel Combinatorial Optimization*, John Wiley & Sons, USA.
- Tamayo, S. et al. (2009) 'Deliveries optimization by exploiting production traceability information', *Engineering Applications of Artificial Intelligence*, Vol. 22, Nos. 4–5, June 2009, pp.557–568.
- Wang, X., Li, D. and O'Brien, C. (2008) 'Optimisation of traceability and operations planning: an integrated model for perishable food production', *International Journal of Production Research*, available online, pp.1–22.
- Wegener, I. (2005) *Complexity Theory: Exploring the Limits of Efficient Algorithms*, Chapters 5 and 6, pp.63–83, Springer Inc.