

## Hybrid Meta-Heuristics for Robust Scheduling

M. Surico, U. Kaymak, D. Naso and R. Dekker

ERIM REPORT SERIES <i>RESEARCH IN MANAGEMENT</i>	
ERIM Report Series reference number	ERS-2006-018-LIS
Publication	March 2006
Number of pages	20
Persistent paper URL	
Email address corresponding author	kaymak@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus Universiteit Rotterdam P.O.Box 1738 3000 DR Rotterdam, The Netherlands Phone: + 31 10 408 1182 Fax: + 31 10 408 9640 Email: <a href="mailto:info@erim.eur.nl">info@erim.eur.nl</a> Internet: <a href="http://www.erim.eur.nl">www.erim.eur.nl</a>

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:  
[www.erim.eur.nl](http://www.erim.eur.nl)

REPORT SERIES  
*RESEARCH IN MANAGEMENT*

ABSTRACT AND KEYWORDS	
Abstract	<p>The production and delivery of rapidly perishable goods in distributed supply networks involves a number of tightly coupled decision and optimization problems regarding the just-in-time production scheduling and the routing of the delivery vehicles in order to satisfy strict customer specified time-windows. Besides dealing with the typical combinatorial complexity related to activity assignment and synchronization, effective methods must also provide robust schedules, coping with the stochastic perturbations (typically transportation delays) affecting the distribution process. In this paper, we propose a novel metaheuristic approach for robust scheduling. Our approach integrates mathematical programming, multi-objective evolutionary computation, and problem-specific constructive heuristics. The optimization algorithm returns a set of solutions with different cost and risk tradeoffs, allowing the analyst to adapt the planning depending on the attitude to risk. The effectiveness of the approach is demonstrated by a real-world case concerning the production and distribution of ready-mixed concrete.</p>
Free Keywords	Supply Networks, Robust Scheduling, Meta-Heuristics, Multi-Objective Genetic Optimization
Availability	<p>The ERIM Report Series is distributed through the following platforms:</p> <p>Academic Repository at Erasmus University (DEAR), <a href="#">DEAR ERIM Series Portal</a></p> <p>Social Science Research Network (SSRN), <a href="#">SSRN ERIM Series Webpage</a></p> <p>Research Papers in Economics (REPEC), <a href="#">REPEC ERIM Series Webpage</a></p>
Classifications	<p>The electronic versions of the papers in the ERIM report Series contain bibliographic metadata by the following classification systems:</p> <p>Library of Congress Classification, (LCC) <a href="#">LCC Webpage</a></p> <p>Journal of Economic Literature, (JEL), <a href="#">JEL Webpage</a></p> <p>ACM Computing Classification System <a href="#">CCS Webpage</a></p> <p>Inspec Classification scheme (ICS), <a href="#">ICS Webpage</a></p>

# Hybrid Meta-Heuristics for Robust Scheduling

M. Surico <sup>a</sup>      U. Kaymak <sup>b</sup>      D. Naso <sup>a</sup>      R. Dekker <sup>b</sup>

<sup>a</sup> Politecnico di Bari, Dipartimento di Elettrotecnica ed Elettronica

Via Re David, 200 70125 Bari, Italy

Email: naso@poliba.it

<sup>b</sup> Econometric Institute, Erasmus School of Economics

P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

Email: u.kaymak@ieee.org

## Abstract

The production and delivery of rapidly perishable goods in distributed supply networks involves a number of tightly coupled decision and optimization problems regarding the just-in-time production scheduling and the routing of the delivery vehicles in order to satisfy strict customer specified time-windows. Besides dealing with the typical combinatorial complexity related to activity assignment and synchronization, effective methods must also provide robust schedules, coping with the stochastic perturbations (typically transportation delays) affecting the distribution process. In this paper, we propose a novel meta-heuristic approach for robust scheduling. Our approach integrates mathematical programming, multi-objective evolutionary computation, and problem-specific constructive heuristics. The optimization algorithm returns a set of solutions with different cost and risk tradeoffs, allowing the analyst to adapt the planning depending on the attitude to risk. The effectiveness of the approach is demonstrated by a real-world case concerning the production and distribution of ready-mixed concrete.

## Keywords

Supply networks, robust scheduling, meta-heuristics, multi-objective genetic optimization.

# 1 Introduction

Supply networks (SNs) are organizations of partially autonomous production and distribution centers through which goods are processed and delivered to customers. Optimizing the activities of a SN to improve production throughput and timeliness of the delivery requires dealing with a number of large-scale, interrelated assignment, scheduling and routing problems. The optimization is especially challenging for a SN that delivers rapidly perishable goods, such as ready-mixed concrete. Since the perishable good can only be used within a short period of time, it must be produced on demand and delivered strictly within the time window that the customer specifies. The optimization problem differs from production and distribution with time windows in that it is not possible to store the goods at strategic locations in the SN. Hence, it is not possible to build buffers to meet the demand at peak times, and so the optimization has to consider both the scheduling of the production and the routing of the delivery vehicles. Furthermore, in addition to the typical combinatorial complexity of such an optimization problem, there are also a large number of constraints because of specific customer or production process requirements.

Although finding a good solution in terms of costs is important, robustness of the solution is also required, in practice. A small delay of a local activity may trigger unpredictable avalanche effects on the other activities linked through precedence relationships or through sharing of common resources. Within the context of the distribution of perishable goods, this might lead to large losses, when a predefined “optimal” solution turns into an unfeasible one due to many timeliness constraints that may become violated. In particular, if the product lifetime is violated, it has to be disposed of, which implies that the costs are not even partially recoverable. Therefore, the decision makers face a tradeoff between minimizing the costs of the operations and safeguarding the robustness to disturbances.

Unfortunately, many general-purpose mathematical programming solvers based on exhaustive or partial enumeration methods do not provide adequate, practical support to decision-makers. They usually deal with single criterion optimization and have excessively long search times, which clash with the need of fast decision tools capable to cope with the dynamics of real-world production and distribution activities. Therefore, SN analysts and practitioners have usually used expert knowledge and heuristics to address these optimization problems. A considerable number of studies have proposed various heuristics to address logistic problems that

global SN scheduling involves, such as scheduling with earliness/tardiness penalty [1], and vehicle routing [11]. While a significant amount of research has also considered similar combinatorial problems with stochastic techniques [8], optimization combined with the robustness of the solutions has been studied to a much lesser extent.

In this paper, we address scheduling in a SN, where robustness of the solutions is also considered to be a goal in addition to the minimization of the costs. We propose to use a novel hybrid meta-heuristic approach for optimal scheduling in a SN for perishable goods. The core of the search strategy is based on evolutionary computation method, mainly to exploit its efficient exploration/exploitation capability in the large search space of the main decision variables characterizing our scheduling problem. Our approach integrates the following elements.

1. A detailed mathematical model of the logistic problem that unambiguously specifies the free decision variables.
2. A set of fast heuristics organized in a hierarchical structure that is able to construct a fully feasible solution starting from an initial assignment of a subset of decision variables.
3. A multi-objective Genetic Algorithm (GA) that searches for the set of best tradeoff solutions considering both the costs and the robustness of the corresponding schedules.

Our main approach is based on quantifying the robustness of a solution by using a risk index and minimizing the risk index as the second objective in a bi-objective optimization, which is solved by using a multi-objective GA. The optimization algorithm returns a set of solutions with different cost and risk tradeoffs, allowing the analyst to adapt the planning depending on the attitude to risk.

In the following, we use the extremely challenging problem of just-in-time production and distribution of ready-mixed concrete as a real-world example to illustrate the potential of our approach. Our analysis builds upon previous work reported in [14, 13]. In [13], the same problem is studied by using a single objective GA in order to minimize the costs. To avoid partial repetition of the material, we give only a brief description of the problem formulation, and concentrate on our novel bi-objective approach for obtaining robust solutions to the optimization problem that we consider. For more details on the description of decision variables and the constraints, we refer the reader to [13].

## 2 Production and Distribution of Ready-Mixed Concrete

A SN for ready-mixed concrete (RMC) consists of a consortium of independent and distributed production centers (PCs) serving a set of customers, which have their construction sites spread in a large geographical area. RMC is transported in special tank carriers (hereafter trucks) from the PCs to the customer sites, where the construction takes place. Some PCs in the SN own a fleet of trucks, but a few ones do not own trucks, and explicitly rely on the other PCs for transportation. Trucks have limited capacity, and so large demands require several truckloads (also called *jobs*) to transport all the concrete. These activities have to be properly synchronized, because the unloading at the customer site must be continuous in order to prevent compromising the mechanical properties of the material.

There are fully automated equipments at the loading docks of PCs, which can prepare any type of RMC by mixing raw components with water, while the product is being loaded on a truck. Therefore, production is simultaneous with loading. It is not possible to produce ahead of delivery and store the RMC temporarily. Each truck can deliver one job at a time, *i.e.* it is not possible to service multiple small requests with a single route. Therefore, all trucks must travel from a PC to a customer, and after unloading they must reach the next PC (which may be different from the previous one) for loading the job of its next task. In case none of the nodes of the SN is able to produce a certain (fraction of) demand at the specified time, the production of the exceeding amount can be outsourced to external suppliers with a consequent loss of revenue. Similarly, when none of the trucks from the internal fleet is available for delivering a given order, a truck from an external company may be hired with additional costs.

Each PC aims to increase resource utilization, decrease costs and ensure the timeliness of the deliveries. Hence, the PCs pursue multiple, contradictory goals. At present, many companies tend either to rely on skilled operators that work out production plans based on their experience [12], or to plan production operations on very short time horizons, sacrificing the optimization on longer horizon to achieve a reduced risk of delayed delivery [15]. Therefore, there is a need for innovative approaches capable of supporting the multi-objective decision processes involved in SN operations.

Several researchers have considered RMC delivery in the literature. A good overview of RMC delivery, including a discussion on how information and material flows can be improved by strategically placing materials and time-buffers, is available in [15]. Similarly, [12] sur-

veys the main characteristics of RMC SNs, and focuses on the routing of two types of vehicles (trucks and pumps) in the SN. In [7], the authors consider scheduling for a single PC owning an unlimited (large) fleet of vehicles. They propose using genetic algorithms for searching a production sequence that maximizes a performance index that is evaluated by discrete-event simulation of the operations of the fleet of vehicles. The working paper [13] reports the application of a single-objective GA for the same SN scheduling problem considered in this paper. The single-objective GA minimizes only the production costs. Robustness of the solution is not an explicit goal and only evaluated a posteriori by means of discrete event simulations.

### 3 Mathematical Model

We consider a network of  $P$  PC's ( $p \in \{1, \dots, P\}$  is the PC index), which receive and process a set of  $R$  requests or orders from different customers ( $r \in \{1, \dots, R\}$  is request index). An order  $r$  has a customer-specified delivery time window  $[EDT_r, LDT_r]$  (earliest and latest delivery time), a required amount  $Q_r$ , and a delivery location. The SN is equipped with a fleet of  $K$  trucks ( $k \in \{1, \dots, K\}$ ) to deliver the product to customers. Each truck has a base location from which it starts every morning, and to which it returns every evening. If a demand exceeds the capacity of a single truck, we split it in a number of sub-demands or jobs ( $i \in \{1, \dots, N\}$  is the job index, and  $N$  is the total number of jobs of the SN), which the customers will receive one after the other. Each job is produced by mixing water with dry components directly when the product is being loaded on the truck, and each PC can load one truck at a time. We do not consider material constraints on the PC's, since raw materials are stored in sufficient quantities at the production centers, in practice. When a fraction of the requests cannot be produced by the PC's of the SN because of time constraints, it is possible to either refuse the unhandled requests, or outsource their production to external companies. Similarly, the SN can hire additional trucks to deliver jobs that cannot be handled by the internal fleet. Clearly, outsourcing production and hiring further vehicles involve additional costs, and are performed only when necessary. Moreover, the need of additional resources is not only related to the amount of requests, but also to the actual utilization of internal resources, determined by the effectiveness of the scheduling policy.

In [13], we have developed a detailed mathematical model to formulate a comprehensive

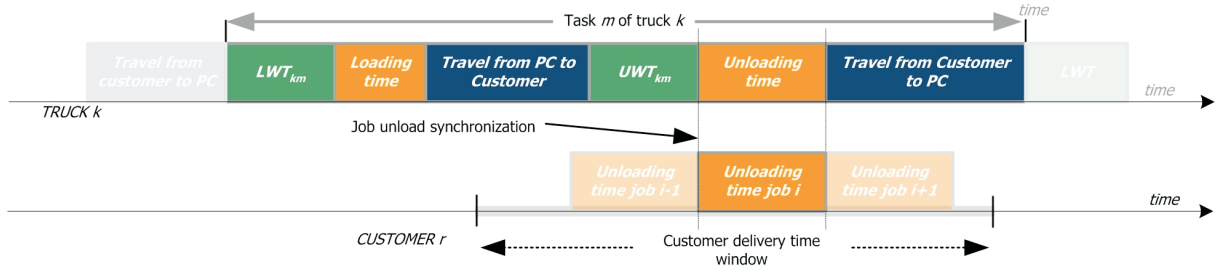


Figure 1: Gantt charts of truck activities for a task, and their synchronization with other tasks at the customer site.

formal description of the problem that could be used by automatic search techniques. Below, we give an overview of the key elements of this model. Let us define a task of a truck as the set of activities involved in picking up and delivering a job to its destination (see Fig. 1). We introduce the task index  $m \in \{1, \dots, M_k\}$ , where  $M_k$  is the last task of truck  $k$ . We assume that all time intervals and parameters in the model are deterministic and known a priori. We identify three groups of binary variables:

$X_{ikm} \in \{0, 1\}$  If job  $i$  is assigned to truck  $k$  as  $m$ -th task,  $X_{ikm} = 1$ , otherwise  $X_{ikm} = 0$ .

$Y_{ip} \in \{0, 1\}$  If job  $i$  is produced at the PC  $p$ ,  $Y_{ip} = 1$ , otherwise  $Y_{ip} = 0$ .

$Y_{oi} \in \{0, 1\}$  If the production of job  $i$  is outsourced,  $Y_{oi} = 1$ , otherwise  $Y_{oi} = 0$ .

The scheduling must take into account several types of constraints, related to

1. the logical assignment of decision variables (*e.g.* a job can be assigned only once to one truck),
2. overlap prevention (*e.g.* loading at a PC can only start when the previous one is finished),  
and
3. RMC lifetime (unloading must finish before the RMC sets).

A characteristic constraint of the problem that makes delays particularly dangerous is related to the continuity of the unloading operations. In fact, trucks must be synchronized (as shown in Fig. 1) such that the end of the unloading of a job coincides with the start of the unloading of the next job of the demand.



The waiting times, indicated in Fig. 1 as  $LWT_{km}$  and  $UWT_{km}$ , are the other key-variables of the model. They measure the interval between the time at which the truck is scheduled to be ready for a (un)loading operation and the time at which the operation is actually scheduled to start. Resource utilization is better for tight schedules with short waiting times. However, longer waiting times make the schedule more tolerant to delays occurring during job transportation. For example, if a waiting time of ten minutes is scheduled prior to any loading or unloading operation of a truck, any delay of a truck shorter than 10 minutes will not affect the subsequent parts of the schedule. For this reason, we have introduced a user-defined lower bound (*minimum waiting time*,  $MWT$ ) for all the waiting times in our model, together with additional constraints to ensure that all the waiting times are longer than the minimum allowed threshold, *i.e.* the condition

$$LWT_{km} \geq MWT, \text{ and } UWT_{km} \geq MWT$$

must hold for all the considered tasks. In other words, by specifying the  $MWT$ , the user defines the minimal length of the time buffer that the scheduling algorithm must place between truck operations. While increasing the  $MWT$  allows to achieve higher tolerance to stochastic delays, it must be also noted that the  $MWT$  cannot be chosen arbitrarily long because of the conflict with the perishable nature of the delivered goods.

The scheduling goals are related to production and delivery costs and timeliness of deliveries. Even assuming deterministic operation and transportation times, simultaneously achieving these two objectives is extremely difficult. Furthermore, the schedule must tolerate unpredictable stochastic perturbations (*e.g.* transportation delays due to traffic). We refer to this aspect as the *robustness* of a solution. In general, cost and robustness are conflicting objectives because tight schedules are also more sensitive to unexpected delays. In our problem, various operations are strongly interrelated by time and precedence constraints, and so even small delays can have unpredictable consequences on the successive operations. We can make a schedule more tolerant to stochastic perturbations by allowing longer time buffers between critical operations in order to absorb longer delays. However, the insertion of larger time buffers also involves a significant cost increase, since it reduces the actual resource utilization. Consequently, we view the SN scheduling problem as a bi-objective search problem in which ideal solutions are those that guarantee a good trade-off between low overall costs and a satisfactory robustness to unpredictable delays. Mathematically, we specify these two objectives as follows.

The production cost of a solution is the sum of three terms:

$$C = C_{transport} + C_{waiting} + C_{extra}. \quad (1)$$

The transportation costs for a schedule account for the distances travelled by all the trucks of the fleet, including the initial and final trips from and to the base locations. It is obtained by multiplying the total distance travelled by all trucks with an average cost per kilometer. The waiting costs account for all the truck waiting times. They are obtained by multiplying the total waiting time of all scheduled trucks (waiting before loading and before unloading) with a penalty for each minute of idle waiting. The extra costs consider all the additional costs related to outsourced production, hired trucks, and payments for overtime of truck drivers.

We estimate the robustness of a schedule by using an index of risk defined as follows:

$$RF = 1 - \frac{Q}{Max(Delay)} \quad (2)$$

where

$$Q = avg(WT_i) - \alpha std(WT_i). \quad (3)$$

In (3),  $WT_i$  is the sum of the waiting times associated with job  $i$ ,  $\alpha$  is a weighting factor, and  $avg$  and  $std$  denote the average and the standard deviation, respectively. The index  $Q$  evaluates the way time buffers are distributed in a schedule. Ideally, waiting times are sufficiently long and evenly distributed across the whole activity schedule, and for this reason their average should be maximized and their standard deviation minimized. In our experiments, we have found  $[0.2, 0.25]$  to be a suitable range for the value of  $\alpha$ . In (2),  $Max(Delay)$  is the maximal expected delay of a travel, which can be easily set by plant managers according to historical data. It is used to bring the values of  $RF$  to a region around the interval  $[0, 1]$  to facilitate easy comparison of the index values. Clearly, to maximize  $Q$ ,  $RF$  has to be minimized. Note that  $RF$  is not a normalized index. Under exceptional circumstances, if the standard deviation of  $WT_i$  is very large,  $Q$  may become negative, forcing  $RF$  to be larger than 1. Such a schedule with a large standard deviation is undesirable, since the waiting times are not distributed evenly then, and the large values of  $RF$  reflect this property.

## 4 Hybrid Meta-Heuristic Solution

The key to our hybrid meta-heuristic solution is the application of a multi-objective search algorithm, with which we search for effective trade-off between costs and robustness. The area of multi-objective optimization has advanced considerably in the last decade. The survey [10] identifies two main reasons for this evolution: the rapid increase in computer power, and the development of more appropriate algorithms for coping with the various complicating factors usually neglected in traditional approaches. In the same survey, it is also acknowledged that the main advances in the context of multi-objective optimization have been in the area of GAs. Multi-objective GAs are variants of conventional (single-objective) GAs devised to simultaneously take into account two or more independent objective functions and return a population of solutions, each representing a different compromise between the considered objectives. Basically, in single-objective GAs the value of the scalar objective function associated to each solution is directly used as fitness of the solution. Clearly, all aggregative approaches combining multiple performance figures in a single objective function fall into this class of GAs. On the contrary, in multi-objective GAs the search goals are not aggregated, but considered separately, and using special ranking and selection mechanisms the population is progressively led toward the set of tradeoff solutions, technically known as *Pareto front* in multi-objective optimization literature. The Pareto front includes all the solutions for which an improvement for one of the considered objectives can only be achieved at the cost of worsening some others (note that when considering bi-objective problems, fronts can be easily viewed on the Cartesian plane of objective functions, as done in Fig. 4). Solutions belonging to a Pareto front are said to be *non-dominated*, since none of them is better than the other ones for all the considered objectives (analogously, if a solution is better than another one for all the objectives, it can be said that the first *dominates* the second). Multi-objective GAs, which have been extensively studied in recent years [3, 4, 5], are in general more complex and computationally more demanding than normal GAs, because they must perform a larger number of comparisons to rank individuals, and because they need specific mechanisms to prevent the concentration of the search on excessively narrow segments of the Pareto front [9]. Due to problem size and complexity, using a multi-objective GA to optimize all the free variables in the problem would involve an unsustainable computational cost. Thus, we use the multi-objective GA in conjunction with fast local heuristics that permit to reach optimized solutions with short execution times (a prototype of the

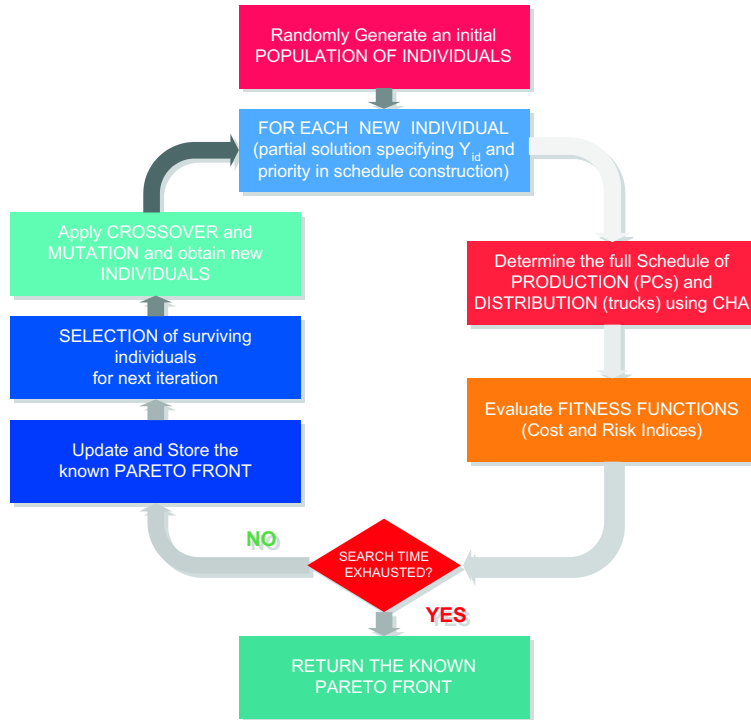


Figure 2: Schematic flow chart of the MOGM.

algorithm written in C++ converges in less than 10 minutes on a Pentium IV PC platform). Our Multi-Objective Genetic Meta-heuristic (MOGM) algorithm assigns demands to PCs (decision variables  $Y_{id}$ ) and defines the order of priority by which the demands are scheduled for production. Then, every time a new solution has to be evaluated, the MOGM launches a Constructive Heuristic Algorithm (CHA), which starts from the assignment given by the MOGM and

1. builds schedules satisfying all the described constraints,
2. assigns the non-outsourced jobs to the trucks.

During the schedule construction, the CHA uses several heuristic strategies to optimize the resource (both PCs and trucks) utilization, and therefore it can be seen as a local search procedure that finds a good solution starting from the solution passed by the MOGM. Figure 2 illustrates the general loop that is executed in our hybrid meta-heuristic approach with a schematic flow chart. The next subsections give a short overview of the basic mechanisms underlying the proposed approach.

Customer $s$ Request-to-PC Assignment						priority of request in schedule construction					
$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	Highest $\rightarrow$ lowest					
1	3	2	1	2	2	4	5	6	1	3	2

Figure 3: Outline of a chromosome (a different colour is used for each PC).

#### 4.1 The MOGM

The MOGM is adapted from Non-Dominated Sorting GA (NSGA-II) [6], an effective algorithm widely referred to in technical literature. A first important issue regards the solution encoding. In our approach, each chromosome consists of two separate parts, both containing  $R$  elements, as shown in Fig. 3 for  $R = 6$ .

The first part of the chromosome defines the initial values for the decision variables  $Y_{ip}$  that assign the demands to the PCs. For instance, gene  $r_2$  indicates that request 2 is assigned to PC 3. At this stage of the decision process, it is assumed that all the jobs composing a split request are produced at the same PC. The second part of the chromosome establishes the order in which the  $R$  requests are considered when building the schedule for the production chain (e.g. request  $r_4$  — scheduled on PC 1 — is allocated first, followed by  $r_5$  — on PC 2 —,  $r_6$  — on PC 2 — and so on). It thus contains a permutation of the numbers from 1 to  $R$ . The values of all decision variables not assigned in the chromosome are computed later by the CHA. This inner module is in charge of constructing a legal solution starting from the partial assignment of decision variables specified in the chromosome.

Since the chromosomes have a specific structure, we had to design new crossover and mutation operators. After an extensive comparative analysis of possible design options, we obtained a pair of operators which appear particularly effective when used together. Both operators randomly select a point in the chromosome. Depending on whether the selected point is in the first or the second part of the chromosome, we apply a different operator. Single-point crossover and simple mutation are used if the selected point is in the first part of the chromosome. Order-based crossover and inversion mutation are used if the selected point is in the second part of the chromosome. Technical details of these operators have been described in [13]. After new chromosomes are generated, CHA is called to construct a feasible schedule given the information in each chromosome. Subsequently, we compute two fitness functions, the cost  $C$  and the risk

index  $RF$ , for each individual. Then, the algorithm selects the best solutions for reproduction in the next population by using the same hybrid ranking/crowding methods of NSGA-II [6]. Using these sorting methods, a solution is always ranked higher than the solutions it dominates. For non-dominated solutions, preference is given to less crowded regions of the Pareto solutions, in order to explore a larger part of the Pareto front.

## 4.2 The Constructive Heuristic Algorithm

The CHA consists of two main modules. The first one (Production Schedule Builder or PSB) is in charge of scheduling the production of all the jobs. The second one (Truck Schedule Builder or TSB) deals with the organization of the transport operations, *i.e.* it assigns jobs and routes to trucks. It is important to remark that these modules are sequential, *i.e.* the TSB cannot modify the PC schedule built by the PSB. In principle, this decomposition may lead to sub-optimal solutions. Nevertheless, we have observed empirically that our approach always provides solutions that outperform those of other methods with which we have compared our meta-heuristics.

The PSB processes requests following the order of priority specified in the chromosome. For each demand, the PSB makes a preliminary set of feasibility checks (*e.g.* if the distance between the assigned PC and the customer permits the end of the unloading before the RMC setting time). If some constraints are violated, the assignment is modified in order to overcome the cause of the violation. Then, the PSB attempts to place the start of the loading window of the first job such that the unloading can start exactly at the customer-specified earliest delivery time. If this window overlaps with a previously assigned job, the PSB makes a series of attempts to overcome the overlaps by rearranging the job sequences without violating other constraints. As a result, among other possibilities, it may happen that some jobs are scheduled to be delivered earlier than the ideal unloading time, thus with a larger-than-desired time buffer. If no adjustment guarantees the feasible schedule of job delivery, the PSB marks the job as temporarily undeliverable and proceeds with the subsequent jobs for the specified PC until either one of the jobs is assigned to the PC, or none of the jobs of a request can be scheduled on the PC. In the latter case (none of the jobs of a request is scheduled on the PC specified in the chromosome), the chromosome is changed and the procedure re-examines the assignment of the request on other PC's in the order of shortest distance from the customer's site. In any case, the

PSB tries (but does not necessarily guarantee) to assign all the jobs of a split request to the same PC, always verifying that the unloading of each job can start exactly when the preceding one is expected to end. After examining all the demands, the PSB attempts to allocate the production of the undeliverable jobs in the idle time intervals of other PC's, starting from the one nearest to the customer's site. Several insertion procedures are examined for each undeliverable job. Finally, if none of these successfully places the job on a PC, the job has to be outsourced.

Once the PSB ends its task, the TSB processes the truck schedule so as to guarantee that a vehicle is available at the expected loading time of each non-outsourced job. The TSB is a nested sequence of various heuristic strategies devised to optimize truck utilization, attempting to simultaneously minimize the travelled distance and the idle times. Basically, jobs are assigned to trucks in the order of the starting time of their tasks. Trucks are considered in decreasing order of their Available Time (AT) defined as the time at which the truck can reach the PC after completing the preceding tasks. The assignment strategy is referred to as Shortest (truck) Idle Time (SIT), because it assigns higher preference to the latest truck that arrives to the PC. In this way, it attempts to improve the truck utilization, concentrating the jobs on the same trucks as much as possible, while leaving some other vehicles idle for longer times. PC's that do not have an internal fleet can use the latter trucks. The algorithm sorts the trucks with the same AT in the increasing order of the distance from the source PC, in order to account for distance-related cost criteria. The job is finally assigned to the truck that can be available at least  $MWT$  before the start of the loading of the scheduled job. If no trucks are available to fulfill this requirement, a request for a hired truck is issued for the delivery of the job.

Note that the CHA could be interpreted as a deterministic search strategy for a feasible solution, given the values of the decision variables specified in the chromosomes passed by the MOGM. In this sense, our approach is related to *memetic algorithms*, which combine genetic search with local search methods [2]. A main issue in memetic algorithms is finding a good balance between the genetic search and the local search [9]. The CHA can be considered a "least-effort" local search algorithm in that it follows a fixed sequence of heuristic procedures to determine variables. It could be noted that more extensive (or even exhaustive) local searches for all variables would be prohibitive in our problem because of the large number of decision variables and the large number of constraints that need to be satisfied. Hence, we have not chosen for a full local search in our approach.

## 5 Summary of Numerical Experiments

We base our case study on experimental data observed during a typical working day of a SN with 5 PC's located in the northern EU. The fleet of trucks consists of 49 vehicles housed in two PC's, and the customers are spread over the area surrounding the network of suppliers. The available data confirm the typical statistical distributions observed in similar cases, and documented in recent literature [12]. In all our experiments, we have set  $Max(Delay)$  to 90 minutes, and the weighting factor  $\alpha$  to 0.2. The value of  $\alpha$  has been determined experimentally. In all our tests, we found that the value of the index  $Q$  was always positive with this setting.

We consider three other scheduling algorithms as reference policies for performance comparison. The first one is a constructive procedure that incorporates the typical decision criteria used by expert plant schedulers. Basically, this algorithm (hereafter SD-SIT) attempts to assign the requests to the PC closest to customer's site (Shorter Distance, SD), and the jobs to the trucks with the Shortest (truck) Idle Time (SIT), mentioned previously. The algorithm also searches for local refinements of the solution by heuristic job insertion or exchange procedures. The second reference algorithm is a basic GA (BGA), in which the schedule corresponding to a generic chromosome is obtained directly using the priorities specified in the chromosome and assigning jobs to trucks with the SIT strategy. Moreover, the GA uses only the cost index as a scalar fitness function. The third algorithm is the Single-Objective GA (SOGA) with constructive stage recently proposed in [13]. Also this GA addresses only cost minimization, and the main difference with respect to the BGA is the adoption of the CHA in the schedule construction stage, which allows us to obtain an extremely efficient hybrid meta-heuristic. We have selected these three algorithms for comparison, because SD-SIT and SOGA were the best performing methods in a recent comparative study, where we had analyzed the performance of SOGA and several other heuristic strategies on problem instances of differing complexity [13]. The SD-SIT was the best non-evolutionary approach in all the considered cases, while the SOGA was always able to outperform non-evolutionary methods. The BGA is another interesting reference term, as it is a fairly common approach that relies only on evolutionary computation to solve the problem.

In this paper, we consider a reference instance describing a typical busy working day of the SN, with 71 requests split in more than 300 jobs. The main configuration and cost parameters used in the decision problem are summarized in Table 1.



Table 1: Cost parameters and the configuration of the algorithm

cost for each Km of travel of the trucks	10
penalty for idle time	15
loss of income for m <sup>3</sup> of concrete to outsource	2000
cost of an hired truck	10000
extra pay for truck drivers' overtime minute	5
population size (randomly generated)	100
termination condition (calls to CHA)	2500
crossover rate	33%
mutation rate	33%

We have run all the considered algorithms 10 times for each value of  $MWT$ . We have increased the  $MWT$  gradually in different experiments. In this way, we obtain different solutions with increasing costs and robustness to delays. Differently from SD-SIT, BGA and SOGA, which return only one solution, the MOGM provides a Pareto-front of non-dominated solutions describing different tradeoffs between cost and robustness. We have observed that the Pareto-fronts found in each of the 10 runs were similar to one another. Below, we report the union of the Pareto-fronts obtained in 10 runs for each value of the parameter  $MWT$ . For BGA and SOGA, we report the best solution found in the 10 runs, again for each value of  $MWT$ . For an immediate comparison, Table 2 and Table 3 report the values of the risk and cost objectives of the two extreme solutions in the front (those with the minimum cost and minimum risk, respectively), while Fig. 4 provides a graphical comparison using the two objectives as the Cartesian axes.

Note in Fig. 4 that the solutions found by the MOGM always dominate those found by SD-SIT. The solutions of the SD-SIT are not on the Pareto front. In some cases, the SD-SIT yields solutions that have the same overall cost as those found by the MOGM with shorter values of  $MWT$ . This indicates that our proposed multi-objective approach can find significantly better solutions. The BGA approach produces solutions that are generally better of those obtained with SD-SIT in terms of overall cost, but it can be noted that they always have a higher risk index. It is reasonable to consider this relatively unsatisfactory result a consequence of the lack of local adjustment of job production and delivery activities during the schedule construction, since it

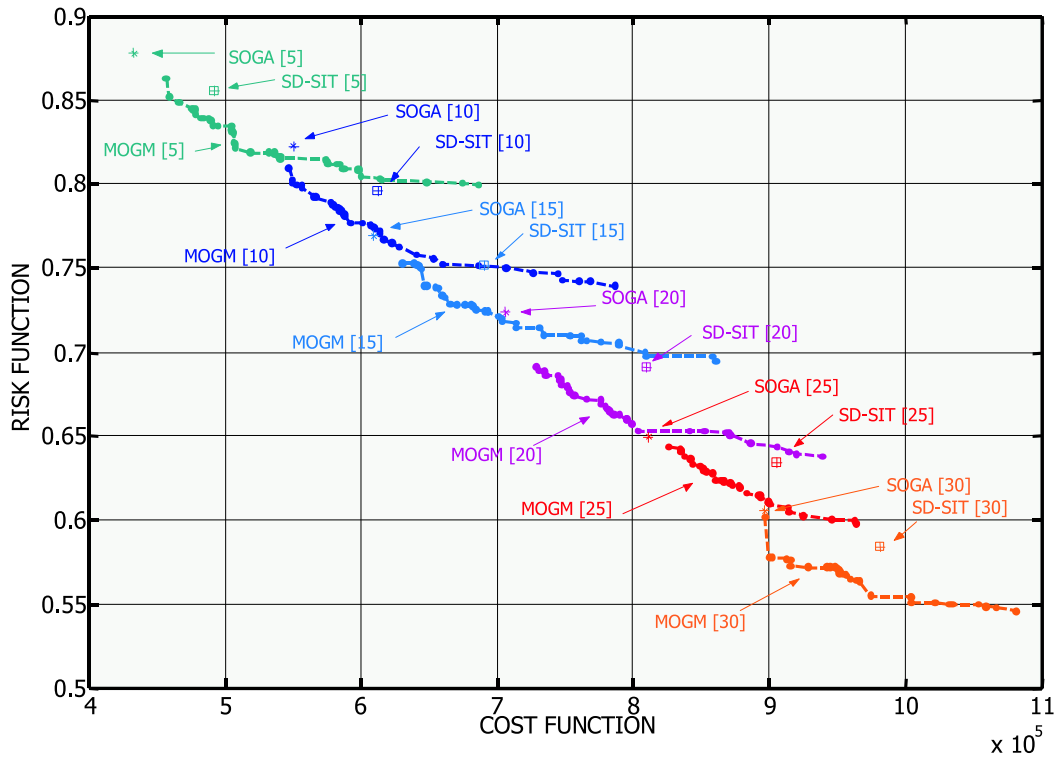


Figure 4: Comparative analysis of the algorithms (BGA results are omitted for the sake of figure clarity).

Table 2: Risk function for the case study ( $MWT$  are in minutes, risk index is a dimensionless number).

$MWT$	SD/SIT	BGA	SOGA	MOGM high cost	MOGM low cost
5	0.856	0.891	0.877	0.799	0.862
10	0.795	0.835	0.823	0.739	0.813
15	0.751	0.781	0.769	0.695	0.762
20	0.691	0.724	0.724	0.638	0.691
25	0.634	0.675	0.649	0.597	0.644
30	0.584	0.619	0.606	0.546	0.601

Table 3: Cost function for the case study ( $MWT$  are in minutes, costs are in normalized units reflecting actual costs)

$MWT$	SD/SIT	BGA	SOGA	MOGM high cost	MOGM low cost
5	492301	469945	432665	686512	457210
10	612462	550422	550178	786999	545475
15	690891	669283	609210	860848	627810
20	809216	759250	706759	938815	728630
25	905416	841689	811440	963676	826405
30	981239	935044	896618	1081105	897000

can be noted that the SOGA generally yields improved cost and risk indices. In particular, the use of the CHA makes SOGA solutions reach the known Pareto front. Since SOGA considers only costs, for a given value of the  $MWT$ , the SOGA solution lies near the upper-left side of the front. These solutions have in general a high value of the  $RF$ , and thus are potentially useful only for cases in which it can be guaranteed that a delay will not exceed the given value of the  $MWT$ .

Figure 4 also shows how the planning analyst can benefit from our proposed bi-objective optimization approach. The analyst can investigate the different tradeoffs between costs and the risks on the Pareto-front and select a particular solution based on his attitude to risk. In particular, one can see that  $MWT$  can be used as a single parameter to control the tradeoff between the robustness of the solutions and the costs involved. Increasing values of  $MWT$  increases the tolerance to unexpected delays, but at the expense of increased costs. Another interesting result is that since the single-objective GA (SOGA) solutions are on the Pareto-front, it is possible to replace the bi-objective optimization by a single objective one after the analysis. For example, the SOGA solution with an  $MWT$  of 15 minutes is on the Pareto-front of the bi-objective optimization with an  $MWT$  of 10 minutes. Hence, the analyst may decide to use a faster single-objective optimization for operational planning, provided he is satisfied with the risk-cost tradeoff offered by the single-objective solution.

To obtain a further validation of the actual performance offered by the policies considered, we have also developed a discrete-event simulation model of the SN, in which truck routes

have been subject to stochastic delays of variable distribution. The investigation confirmed that the probability of critical events (*e.g.* gaps in the unloading of large demands) is significantly reduced in the solutions with low risk index.

## 6 Conclusions

A hybrid meta-heuristic approach based on a multi-objective genetic algorithm combined with constructive heuristics is a valuable decision support tool for planning operations in a supply network for rapidly perishable goods. Provided a detailed mathematical model of the supply network is available, our experimental investigation shows that such a hybrid approach is able to provide an effective scheduling algorithm. The hybrid meta-heuristic approach provides a unified framework within which both the cost aspects and the robustness of the solution are considered. The user is provided with a set of different schedules, each corresponding to a different ratio of production cost and tolerance to unexpected delays. Further, the multi-objective approach is able to obtain a satisfactory tradeoff front in an acceptable time, which is short enough to perform real-time rescheduling in case new orders are received during the working day.

## References

- [1] K. R. Baker and G. D. Scudder. Sequencing with earliness and tardiness penalties: a review. *Operations Research*, 38:22–36, 1990.
- [2] E. K. Burke and A. J. Smith. Hybrid evolutionary techniques for the maintenance scheduling problem. *IEEE Transactions on Power Systems*, 15(1):122–128, 2000.
- [3] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamour. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.
- [4] Y. Collette and P. Siarry. *Multiobjective Optimization: Principles and Case Studies*. Springer, Berlin, 2004.
- [5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, London, 2001.

- [6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [7] C. W. Feng, T. M. Cheng, and H. T. Wu. Optimizing the schedule of dispatching RMC trucks through genetic algorithms. *Automation in Construction*, 13(3):327–340, May 2004.
- [8] M. Gendreau, G. Laporte, and R. Séguin. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research*, 44(3):469–477, 1996.
- [9] H. Ishibuschi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2):204–223, 2003.
- [10] D. F. Jones, S. K. Mirrazavi, and M. Tamiz. Multi-objective meta-heuristics: an overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, 2002.
- [11] G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4–5):285–300, Sept. 2000.
- [12] N. F. Matsatsinis. Towards a decision support system for the ready concrete distribution system: a case of a greek company. *European Journal of Operational Research*, 152(2):487–499, Jan. 2004.
- [13] D. Naso, M. Surico, B. Turchiano, and U. Kaymak. Genetic algorithms in supply chain scheduling of ready-mixed concrete. ERIM report series Research in Management, Erasmus Research Institute of Management, Erasmus University Rotterdam, the Netherlands, Oct. 2004. ERS-2004-096-LIS.
- [14] D. Naso, M. Surico, B. Turchiano, and U. Kaymak. Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete. To appear in the *European Journal of Operational Research*, 2006.

- [15] I. D. Tommelein and A. Li. Just-in-time concrete delivery: mapping alternatives for vertical supply chain integration. In *Proceedings of the Seventh Annual Conference of the International Group for Lean Construction IGLC-7*, pages 97–108, University of California, Berkeley, California, July 1999.

## Publications in the Report Series Research \* in Management

### ERIM Research Program: "Business Processes, Logistics and Information Systems"

2006

*Smart Business Networks Design and Business Genetics*

L-F Pau

ERS-2006-002-LIS

<http://hdl.handle.net/1765/7319>

*Designing and Evaluating Sustainable Logistics Networks*

J. Quariguasi Frota Neto, J.M. Bloemhof-Ruwaard, J.A.E.E. van Nunen and H.W.G.M. van Heck

ERS-2006-003-LIS

<http://hdl.handle.net/1765/7320>

*Design and Control of Warehouse Order Picking: a literature review*

René de Koster, Tho Le-Duc and Kees Jan Roodbergen

ERS-2006-005-LIS

<http://hdl.handle.net/1765/7322>

*A Theoretical Analysis of Cooperative Behavior in Multi-Agent Q-learning*

Ludo Waltman and Uzay Kaymak

ERS-2006-006-LIS

<http://hdl.handle.net/1765/7323>

*Supply-Chain Culture Clashes in Europe. Pitfalls in Japanese Service Operations*

M.B.M. de Koster and M. Shinohara

ERS-2006-007-LIS

<http://hdl.handle.net/1765/7330>

*From Discrete-Time Models to Continuous-Time, Asynchronous Models of Financial Markets*

Katalin Boer, Uzay Kaymak and Jaap Spiering

ERS-2006-009-LIS

<http://hdl.handle.net/1765/7546>

*Mobile Payments in the Netherlands: Adoption Bottlenecks and Opportunities, or... Throw Out Your Wallets*

Farhat Shaista Waris, Fatma Maqsoom Mubarik and L-F Pau

ERS-2006-012-LIS

<http://hdl.handle.net/1765/7593>

*Hybrid Meta-Heuristics for Robust Scheduling*

M. Surico, U. Kaymak, D. Naso and R. Dekker

ERS-2006-018-LIS

---

\* A complete overview of the ERIM Report Series Research in Management:

<https://ep.eur.nl/handle/1765/1>

ERIM Research Programs:

LIS Business Processes, Logistics and Information Systems

ORG Organizing for Performance

MKT Marketing

F&A Finance and Accounting

STR Strategy and Entrepreneurship