Maximizing the Throughput of multimodal logistic platforms by simulation-optimization: the Duferco case study

Original
Maximizing the Throughput of multimodal logistic platforms by simulation-optimization: the Duferco case study / Ghirardi, Marco; Perboli, Guido; Sasia, D.. - ELETTRONICO. - (2011), pp. 563-568. (Intervento presentato al convegno Automation Science and Engineering (CASE), 2011 IEEE Conference on tenutosi a Trieste (Italy) nel August 24-27, 2011) [10.1109/CASE.2011.6042475].

Availability:
This version is available at: 11583/2438777 since:

Publisher:
IEEE/IET Electronic Library (IEL)

Published
DOI:10.1109/CASE.2011.6042475

(Article begins on next page)

Maximizing the Throughput of Multimodal Logistic Platforms by Simulation-Optimization: the Duferco Case Study

Marco Ghirardi [1], Guido Perboli [1 2], Daniele Sasia [1]
[1] DAUIN - Politecnico di Torino - Italy
[2] CIRRELT, Canada

***Abstract*— In this work a multimodal transport platform designed by Duferco near a steel factory in Belgium is considered. Aim of this paper is to present a simulation-optimization decision system for evaluating the maximum throughput of the platform in presence of uncertainties over the durations of the operations. A greedy algorithm and a local search phase has been implemented in order to create a feasible schedule of the planned operations over different time horizons (from a month to a year). Once a schedule has been computed using approximated input data, it is simulated, introducing stochasticity in the system parameters. From the simulation results it is possible to better evaluate some of the scheduling data, modify them accordingly, and re-schedule and simulate the system. This procedure is iteratively applied until an asymptotic state is reached. Extensive computational results show how the new solutions improve the company best practices.**

I. INTRODUCTION

Aim of this work is to present a simulation-optimizaton method to analyze the throughput of a multimodal system, when an intelligent scheduling system is applied and in presence of uncertainties on the durations of the operations (products movements, setup times, etc). The method is applied to a real case study: the Garocentre platform multimodal platform of Duferco Group.

Traditional methods for dealing with uncertainties in optimization are scenario optimization [2] and robust optimization [4]. These approaches, however, only consider a very small subset of possible scenarios and the size and complexity of models they can handle are very limited.

In this work, in order to determine the maximum throughput of the platform and the impact o of an efficient operations scheduling algorithm on the behaviour of the system, a simulation-optimization tool is introduced, i.e. a method integrating both simulation and combinatorial optimization methods. The need of optimizing the system while simulating and the underlying difficulty in solving the discrete optimization problems (they are usually NP-Hard even in their deterministic version) pushed in recent years the academic community toward solutions mixing simulation and heuristics in order to incorporate both the robustness of heuristics with the established methods for guaranteeing that performance is valuable ([5] [1] [3]).

Marco Ghirardi is Assistant Professor, Politecnico di Torino, Italy marco.ghirardi@polito.it
Guido Perboli is Assistant Professor, Politecnico di Torino, Italy, and Associate Member, CIRRELT, Montreal, Canada guido.perboli@polito.it

The complete optimization system is composed by a scheduling algorithm able to plan the activities of the platform and a simulation model of the system. The scheduling system is based on deterministic expected values of the problem data, while the simulation model is used in order to evaluate the schedule, and then to adjust the values of the scheduling data. The process is repeated until a convergence is met, obtaining as a result a schedule which is robust to data uncertainties.

Extensive computational results based on real data show how the developed method not only validates the scheduling algorithm, but let the company obtain a better insight on its impact on the overall system performances. Moreover, the computational tests show how the convergence is achieved (which happens in few iterations).

The paper is organized as follows: in Section II a detailed description of the platform (layout, flows, constraints) is presented. Sections III and IV describe the scheduling algorithm and the simulation model. Section V presents the computational results obtained by applying the simulation-optimization system to real data.

II. THE MULTIMODAL PLATFORM

Duferco Group Garocentre multimodal platform is an area located near a steel factory located in the city of La Luovière in Belgium. All the required materials and manufactured products are handled in this area for transportation. The region is characterized by a dense network of ship-canals providing an efficient transport system by barges. The products of the factory have to be loaded on the barges, and raw materials arrive by barges. Trains are used in order to move the products from and to the steel factory. Moreover the platform is rented to other companies of the same geographical area that need to receive containers which arrive by barges and are carried to the customer site using trucks.

A. Handled materials

Cold rolled carbon steel (CRC). Cold rolling is a metallurgical process in which metal is passed through a pair of rollers at a lower temperature than the recrystallization temperature. This process hardens the metal, by compressing and stretching the crystals. Metals are rolled cold to make sheets which are then rolled forming a coil with a diameter of about 2 meter. Because of their physical properties CRC must be stored in a roofed place in order to be protected from atmospheric agents. CRCs are produced in the steel factory and then transported to a warehouse by train, waiting to be loaded on the barges.

Hot rolled carbon steel (HRC). The metallurgical process of hot rolling is similar to the cold one, except that the temperature of the rollers is generally higher than the metal recrystallization temperature. This permits large deformations to be achieved with a low number of rolling cycles. Also in this case the sheets are rolled forming coils. Hot coils have very strong physical characteristics, and do not need to be stored in a covered warehouse. Like CRCs, HRCs are transferred to the multimodal platform by train, then they are loaded on the barges.

Slabs. Steel slabs are very big block of metal (10.6 x 1.65 x 0.25 m). The factory uses them as raw materials to produce CRCs and HRCs. Slabs arrive by barges, and after being unloaded on the dedicated warehouse, they are forwarded to the factory by train.

Containers. Other companies will take advantages of the utilization of this structure. In fact they will rent the platform use for managing containers. They arrive in the area by barges. Differently from slabs, they leave the warehouses not by train, but using trucks.

B. Platform layout

Garocentre platform has to be able to manage both loading and unloading from the different types of transport systems and also to store the products that cannot be immediately moved and directed to the final destination. To exchange the products between warehouses and transports three cranes are used. The system layout is divided in five storing areas (see labels in figure 1):

- CRC warehouse (1): this area is roofed in order to protect the CRCs by atmospheric agents, and it is subdivided in four zones. Its capacity is 1188 coils.
- HRC warehouse (2): this is not a roofed area. Its capacity is 824 coils.
- Slabs warehouse (3): slabs can be stored in stack in this area. It can reach a capacity of 850 units.
- Containers warehouse (4): they can be put in stacks of 3 items each one. The store can support 600 containers.
- Intermediate zone (5): this area is used by slabs and containers in order to improve the barges unload speed, since their warehouses are far from barge dock. It is near dock 2 and HRC warehouse and can store up to 68 slabs and 80 containers.

C. Transports

There are four main areas in which products can arrive or leave from the multimodal platform:

1) 2 Barge Docks: these are barge loading and unloading platforms: Left platform Dock 1 is used to load CRCs from the warehouse to the barges. Right platform Dock 2 is used to load HRCs from stores to barges, and to unload Slabs and Containers from barges to the warehouses.
2) A single railway for the train used in order to unload trains with CRCs to warehouses; unload trains with HRCs to warehouses; load trains with slabs.
3) A truck load position: trucks are loaded on one at a time aside the container area.

In order to load, unload and move the items inside the system 3 cranes are used. Two of them are placed in the CRC roofed warehouse and they only serves this item type. The other crane serves all the other tasks of the platform and needs to have different tools on its terminal part (one for coils, one for slabs and one for containers): each time the crane switches product type, a tool change (*setup*) is needed.

D. Material flows

As explained in previous paragraphs each material follows its own flow through the platform. The detailed operations for each material type are:

- CRC
  - CRC arrival: arrival of a CRCs train and its unloading through the two cranes (*portal 1*) on the roofed warehouse. Trains can travel only during the night (from 19.00 to 7.00).
  - CRC departure: it denotes the arrival of a barges and its loading performed by portal 1 on dock 1. This operation can be performed during the day only (from 7.00 to 19.00).
- HRC
  - HRC arrival: arrival of a HRCs train and its unloading through the external crane (*portal 2*). It can be executed during the night only.
  - HRC departure: arrival of a barges and its loading performed by the portal 2 on the dock 2. It can be performed during the night only.
- Slabs
  - Slabs arrival: arrival of a barge on dock 2 and its unloading in the intermediate area. It can be executed during the day.
  - Slabs shifting: slabs are moved by portal 2 from the intermediate zone to the slabs warehouse. It can be executed any time.
  - Slabs departure: arrival of a train and its loading performed by the external crane. It can be executed during the night only.
- Containers
  - Containers arrival: arrival of a barge on dock 2 and its unloading in the intermediate area. It can be executed during the day only.
  - Containers shifting: containers are moved from the intermediate zone to their warehouse. It can be executed any time.
  - Containers departure: arrival of the trucks and their loading performed by the external crane. It can be executed during the day only.

Note that slabs and containers are unloaded by the external crane, not directly on their proper warehouses, but in the

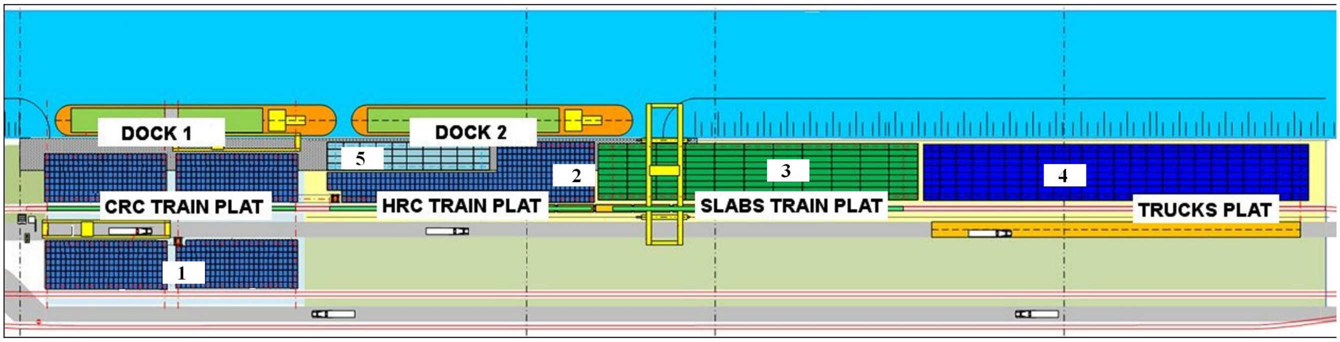

Fig. 1. Multimodal platform layout

intermediate area, in order to be transported in the final stores at a second time. The reason is to reduce the time of unloading, since slabs and container warehouses are quite far from dock 2. Hence, reducing the unloading time, barges can depart quickly from the docks, leaving them free for the following ones.

E. Additional Constraints and data

Some additional constraints come from the topological features of the platform and the transport system or from external factors like, for example, the rules that regulate the traffic of the barges over the canals. All these constraints has been provided by the Duferco company.

Trains: 1440 tonnes capacity, which means 68 coils or 68 slabs; trains can only access the area between 7.00 pm to 7.00 am (period defined as "night"); they must be served within 24 hours (otherwise a penalty can occurs); a train can only carry one product type each time (HRCs, CRCs or slabs): a mixed load is not permitted; they can only load or unload products each time they enters the system (so they cannot first unload material and then load another one before leaving the platform); only one train at a time is allowed in the system.

Barges: 1350 tonnes capacity, which means 68 coils or 68 slabs, or 80 containers; they can arrive from 7.00 am to 7.00 pm (period defined as "day"); they have to leave completely loaded or unloaded by 7:00 pm; a barge can only carry one material type at a time (HRC, CRC, slabs, or container); they can only perform one operation at a time, i.e. they can only load or unload material each time they enter the system, and then they have to leave the dock.

Trucks: 1 container capacity, they can arrive only during the day (7am-7pm), they have to be served in 1 hour; only one truck each time is allowed to enter the platform (the queued will wait on the access road, before the gate registration point).

Cranes: in order to avoid to overload the work of the cranes, and allow maintenance, their utilization should be as close as possible to 70 %.

III. Scheduling Algorithms

In this section we present the the algorithm proposed to build a complete timetable containing the arriving time of the transports, the starting time and the completion time of all the operations is developed.

The following subsections describe the algorithm structure, which is divided into two main parts:

- A Greedy algorithm that computes a feasible schedule given the quantity of materials to serve (feasible means that it respects all the constraints of the system);
- A Local Search algorithm that improves the schedule.

The choice of a heuristic algorithm is justified by the large dimensions of the instances of the problem to be solved (about 700 loads). Note also that no mathematical model of the problem is provided due to the large number of constraints and particular behaviours of the system.

A. Greedy Algorithm

The main idea of a greedy algorithm is to insert the operations one at a time in the schedule, until all the input quantities have been scheduled. The types of products to manage are 4; for each of them we have a related sequence of operations and a number of loads to serve. At each iteration, the algorithm chooses which product type to serve, then it inserts in the schedule the operations according to the given sequence of that material. If the sequence is completed the counter of the relative number of loads to serve is decremented and the sequence is restarted.

Let us suppose, for example, that at the beginning the sequences for CRC, HRC, slabs and containers start respectively with the operation 0,2,4 and 7; the algorithm choose one material to serve HRC, so the operation inserted in the schedule will be 2 and the relative sequence advances with operation 3. In the next iteration the operations at the top of the sequences will be 0,3,4 and 7. Supposing that also this time the products chosen by the algorithm are again HRC, the operation inserted will be 3; now one sequence for HRC is completed (2-3), that means that one load is completely served, so the counter for HRC will be decremented and the sequence restarts. This procedure is repeated until all loads of all types of products are scheduled.

In order to choose the operation to be scheduled, the first available window of time where each candidate operation can be scheduled is computed. Hence, it is necessary to consider the characteristics of each operation and all the constraints of the system (involved warehouse capacity, period in which the operation must be executed, duration, including tool change if needed, cranes train and truck position availability). Once a possible schedule has been computed for the four candidate operations, two different versions of the algorithm have been tested:

1) *End time* version: chooses among the 4 candidate operations the one that ends earlier.
2) *Start time* version: inserts in the schedule the operation that starts earlier.

Once the operation has been inserted in the schedule three structures are created: a vector of the sequence of operations (which represent the solution which will be perturbed during the next improvement phase), and two vectors (*Sched 1* and *Sched 2*) containing the scheduling information for the two portals. The complete pseudo-code of the procedure is reported in algorithm 1.

Let us define O as the total number of operations of an instance. The computational complexity of the greedy procedure is then $o(O^2)$. In fact, all operations has to be scheduled, and the complexity of an insertion in the sequence requires in the worst case to try the insertion in every position.

Algorithm 1 Greedy algorithm

1) Inizialization
   - INIT: set the number of loads that must be served for each product type and the first operation of the sequence for each product type.
2) WHILE there are still operations to be scheduled
   - COMPUTE SCHEDULE: for each type of product, the first available scheduling window for the operation that stays on top of the sequence.
   - Choose the best among the 4 operations (the one that ends earlier or the one that starts earlier depending on the version (*end time* or *start time*) of the algorithm).
   - INSERT the selected operation in the correct portal schedule vector (*Sched 1* or *Sched 2*) and in the operations sequence vector *SeqVector*.
   - UPDATE the status of the involved warehouses.
   - SET the correct next operation in the sequence of the involved type of product
   - IF the sequence is complete THEN restart the sequence and decrement the relative load counter.

B. Local Search

In order to improve the quality of the Greedy algorithm, a Local Search phase is applied to the given schedule. The algorithm works perturbing the vector which contains the sequence of the scheduled operations. For each operation, it tries to move it forward and backward in the vector: after each movements we try to schedule the operations exactly following insertion sequence of the new vector; if the makespan of the new schedule is lower than the current one, then the new vector is selected as current solution (*first improvement* strategy). It is possible that, moving an operation, an infeasible vector is obtained (for example it can be possible to obtain a sequence that tries to insert an operation of loading from a warehouse while this is empty). In this case the operation shifting on that direction is interrupted. At the end of the forward and backward shifting, the best solution is taken as current solution and in the next iteration the algorithm will work over it.
Hence, the dimension of the neighborhood is not fixed "a priori", but it depends on the current solution characteristics. The computation complexity of an operation sequence evaluation is $o(O)$.

Local Search pseudo-code is presented in algorithm 2.

Algorithm 2 Local Search

1) FOR each element in the current sequence vector
   - WHILE we can forward shift (or we reach the end of the current vector):
     – COMPUTE the Schedule of the new vector. The new vector becomes the new current one if its makespan is better than the current one.
   - WHILE we can backward shift:
     – COMPUTE the Schedule of the new vector. The new vector becomes the new current one if its makespan is better than the current one.
2) IF at least one improvement has been find in the previous cycle THEN goto 1.

IV. HANDLING UNCERTAINTIES THROUGH SIMULATION

One of the main objectives of this work is to simulate the platform in order to understand its behavior when the schedule computed through the developed algorithms is applied. In fact during the scheduling phase we cannot take in account some parameters that instead can be modeled in a simulation, in order to obtain results which are very near to the real case. A specific software called Flexsim has been used. Flexsim is a Windows-based, object-oriented simulation environment for modeling discrete-event flow processes. Apart from simulate the system, it allows also to build the 3D model of the system, in order to verify visually its evolution in time. Note that the simulation software choice is not unique: different tools (Arena, Extend, etc) could have been used in order to build the model.

All the involved transport systems are designed using their physical characteristics like dimensions, capacity, speed, acceleration. Cranes are the most important elements of the system for their impact on the stochastic solution. Thus, cranes and involved operations have been modeled with a high level of detail, indicating also speed and acceleration for the different movements they can perform like lifting, descending and trolley. Every time an operation is completely executed, the ending time is saved in a table in order to compare the simulation results with the expected values obtained from the scheduling algorithms. Every time a new scenario is mapped into the simulation, importing the scheduling information, multiple repeated simulations are executed in order to obtain more accurate results, eliminating some possible disturbances which can occurs running a single simulation, due to the stochastic nature of some parameters of the system.

The simulation phase allows to verify if the theoretical schedule will work as expected. Moreover through the sim-

TABLE I

MATERIAL FLOW VALUES FOR ANNUAL SCHEDULE

Material	Units/year	Loads/year
CRC	12,250	180
HRC	12,250	180
Slabs	10,500	154
Container	15,000	187

TABLE II

OPERATION DETAILS

ID	Description	Period	Duration	Trasporter	Dock
0	CRC Unloading	Night	126	Train	1
1	CRC Loading	Day	208	Barge	1
2	HRC Unloading	Night	261	Train	2
3	HRC Loading	Day	310	Barge	2
4	SLABS Unloading	Day	272	Barge	2
5	SLABS shift	Always	243	-	2
6	SLABS loading	Night	158	Train	2
7	Container Unload	Day	258	Barge	2
8	Container Shift	Always	690	-	2
9	Container load	Day	432	Trucks	2

TABLE III

GREEDY RESULTS - *end time* VERSION

Time Horizon	Makespan (days)		Idle Time (days)	
	Portal1	Portal2	Portal1	Portal2
Annual	180.1	521.3	137.6	197.7
Quarterly	45.1	131.3	34.1	49.8
Bimonthly	30.1	87.3	22.6	33.3
Monthly	15.1	44.3	11.1	16.7

TABLE IV

GREEDY RESULTS - *start time* VERSION

Time Horizon	Makespan (days)		Idle Time (days)	
	Portal1	Portal2	Portal1	Portal2
Annual	180.1	367.2	137.6	30.8
Quarterly	45.1	92.2	34.1	7.6
Bimonthly	30.1	61.2	22.6	5.1
Monthly	15.1	31.2	11.1	2.7

ulation tool, it is possible to compare schedules that are theoretically very similar in terms of end time and find which, in practice, works better.

Simulation results are also useful to correct some parameters of the scheduling algorithms, like the operations durations. Once the algorithms parameters are modified, a new schedule can be computed and simulated. In the next chapter computational results show that through an iterative application of the simulation procedure an asymptotic state can be found, where the parameters of the system are more accurately estimated and the computed schedule is more realistic than the first computed one.

V. COMPUTATIONAL EXPERIMENTS

In this section the results obtained using the algorithms and the techniques described before are illustrated. Duferco gave a set of values representing the expected material flows that the platform should serve in an year (see Table I). Using these data, a set of schedules differentiated by their duration (1 year, 4 months, 2 months, 1 month) are computed for each version of the algorithms. Duration and characteristics of the considered operations are summarized in Table II, where operation durations are expressed in minutes. Note that these data are approximated starting from mean values (considering, for instance, the maximum speed of the cranes, the mean acceleration, etc.) which are subjected to a high uncertainty. As previously stated, the external crane move three product types (HRC, slabs and containers), hence each time the current material changes, a new tool must be assembled over the crane. This task has a duration of 30 minutes.

A. Greedy Algorithm

In table III and IV the results of the different scenarios are reported.

In all the instances, the external crane (portal 2) is by far the most used one; this is what we expected, due to the fact that the cranes in portal 1 have only to serve CRC flows, while portal 2 has to manage HRCs, slabs and containers.

Looking at the different behaviors between the schedules time duration (annual, quarterly, bimonthly, and monthly), it is possible to see that the completion times ratio between each pair of scenarios, is almost equal to the ratios between their expected duration. This means that, in practice, there are no significant differences to schedule a long time period compared to schedule a set of short time periods. The main difference is that, since in the obtained solution the operations are not homogeneously mixed (some kinds of flows tends to be scheduled first with respect to the other), it could be better, in general, to schedule a reasonably short time period and repeat the result a sufficient number of times, in order to cover the entire time period that must be scheduled. This guarantees the homogeneity in the complete solution.

The *idle time* column reports the sum of the portal idle times between two consecutive operations. This gives a limit to the makespan reducing that it is theoretically possible to gain with the local search procedure.

Analyzing the results of the two versions of the algorithm it is clear that the *start time* version is significantly better than the *end time* version. In fact, looking at the annual schedule, the *start time* results is only 2 days longer than the threshold of 365 days, while the other version exceeds of about 5 months. This result is due to the fact that the *end time* algorithm has a very large quantity of downtime days (on portal 2) compared to the *start time* version. In fact inserting the operation in the schedule looking at the one that starts earlier means to reduce the downtime of the cranes between two successive operations.

The CPU time required for building the solution is not reported in the tables because it is negligible (few centiseconds).

B. Local search

In tables V the result obtained applying the local search to the solution found by the best of the greedy algorithm versions (*start time*) are shown. Note that in the annual schedule

TABLE V

LOCAL SEARCH IMPROVING THE *start time* GREEDY SOLUTION

Time Horizon	Greedy Makespan P 1	Greedy Makespan P 2	L.S. Makespan P 1	L.S. Makespan P 2	CPU time (s)
Annual	180.1	367.2	163.3	354.4	1342.9
Quarterly	45.1	92.2	41.4	89.4	7.9
Bimestrial	30.1	61.2	28.1	59.4	1.9
Monthly	15.1	31.2	14.3	30.4	0.84

the gain obtained with respect to the greedy solution over portal 2 is about 13 days. This means that from the 30 days of downtimes present in the initial solution (see the results in table IV) 13 days are gained. The values obtained by the other scenarios (quarterly, bimonthly, monthly schedule) are in practice proportional to the ratio between their duration and the annual one. This means that also in this case there are no significant differences, in terms of performance, changing the duration of the computed schedule.

For what concern the CPU time, it can be noticed that, as expected, the duration of execution in the annual cases is significantly larger than the time spent by the other cases. The testings has been run on a Centrino 2 notebook.

C. Simulation and feedback results

The goal of simulation is to understand if the system can actually apply the schedules computed through the algorithms. Already from the first simulations, we noticed that there can be differences between the expected operations durations and the simulated ones; this is due to the fact that in the computations of the durations, the use of the mean values of the distances and of the technical characteristics of the involved transport systems cannot give very precise estimations.

An effective way to solve this issues is to use the informations derived by the simulation results, in order to modify the operations durations in the scheduling algorithm, having a more precise estimation of those quantities. This procedure consists in computing the average values of the differences between theoretical durations and simulated ones and then modify the related quantities into the scheduling algorithm; at this point, the schedules have to be recomputed and simulated. This procedure must be repeated iteratively until the differences become null or irrelevant.

Since the behaviour of the simulated system is the same in all the scenarios that have been discussed, scheduled and simulated, table VI refers to the bimonthly schedule.

At each iteration, through the results of the simulation, the average values of the differences between the expected and simulated operation durations are derived (columns $Avg\Delta_i$), then the updated operation durations are computed (columns $IT{=}i$). After that, replacing the old values into the scheduling algorithm with the new ones, a new schedule is computed and simulated. Note that after the first iteration those differences are highly reduced. For completeness, a second iteration has been performed: the mean values of the differences are now practically null.

For what concern the CPU time, the simulation procedure without visual animations requires a few seconds to evaluate a 1-year operations schedule.

TABLE VI

ITERATIVE PROCEDURE RESULTS

Op. ID	IT=0	AvgΔ_0	IT=1	AvgΔ_1	IT=1	AvgΔ_2
0	126	14	140	-3	137	0
1	208	189	397	-2	395	0
2	261	-122	139	-1	138	-2
3	310	-125	185	-4	181	0
4	272	119	153	-1	152	0
5	243	0	243	0	243	0
6	158	9	167	-1	166	2
7	258	-30	228	2	230	1
8	690	0	690	0	690	0
9	432	-154	278	0	278	0

VI. CONCLUSIONS AND FUTURE RESEARCH

In this work the real case of the multimodal platform of Garocentre has been analyzed. In order to compute a schedule that satisfies all the required operations which have to be performed by the system, a heuristic based on a greedy algorithm followed by a local search phase has been developed. In order to validate the schedules, a simulation model of the multimodal platform has been built using the Flexsim environment. Adding the stochastic variables, that could not be taken in account during the scheduling phase (like delays, faults, etc.), it was possible to verify if the theoretical behaviour of the schedule could be actually implementable and, through a iterative procedure, better evaluate some parameters of the system (operations durations). The solutions found has been presented to the Duferco company staff, and a really good feedback about the quality of the solutions has been received. However, as future research, it could be interesting to test if those solutions could be improved by using more complex heuristic frameworks (Tabu Search, Genetic Algorithms, etc).

VII. ACKNOWLEDGMENTS

The authors want to thanks Duferco, Flexsim and Flexcon Srl for their support to this research.

Funding for this project has been provided by the Italian Ministry of University and Research, under the 2007 PRIN Project "Optimization of Distribution Logistics" and the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs.

REFERENCES

[1] M. Better, F. Glover, G. Kochenberger, and H. Wang, "Simulation optimization: Applications in risk management," *Int. J. Inform. Tech. Decis. Making*, vol. 7, no. 4, pp. 571–587, 2008.

[2] R. Dembo, "Scenario optimization," *Ann. Oper. Res.*, vol. 30, pp. 63–80, 1991.

[3] M. Fukushima, "How to deal with uncertainty in optimization some recent attempts," *Int. J. Inform. Tech. Decis. Making*, vol. 5, no. 4, pp. 623–637, 2006.

[4] P. Kouvelis and G. Yu, *Robust Discrete Optimization and Its Applications*. Dordrecht, Netherlands: Kluwer, 1997.

[5] S. Olafsson and J. Kim, "Simulation optimization," in *Proceedings of the 2002 Winter Simulation Conference*, E. Ycesan, C. Chen, J. L. Snowdon, and J. M. Charnes, Eds., 2002, pp. 479–485.