Optimising supply chain networks by means of a hybridised simulation-based approach

Sven Schellenberg, Arvind Mohais, Neal Wagner, Maksud Ibrahimov and Zbigniew Michalewicz

***Abstract*— Supply chain management, in the scope of the described project, is about managing the flow of materials in a network of factories producing and transforming raw material into intermediate and final product, and the use of buffering instances such as storage tanks, silos and stockpiles. The system we present tries to reconcile the drivers of a supply chain: demand for final product and supply of raw material. In addition to balancing the material flow to honour physical constraints (i.e. storage capacities, minimum production rates, transportation bottlenecks, etc.), the system aims to maximise the overall output of the supply chain network. Other benefits from a business point of view are the reduction of time to generate a factory plan while providing better accuracy and visibility of the material flow. Reducing the costs for creating a plan allows for what-if-scenario analysis and strategic planning which would not have been possible otherwise.**

In order to optimise the material flow, an Evolutionary Algorithm (EA) was employed that incorporates operators handling business and general planning constraints. Furthermore, the EA utilises a discrete-event simulation (DES) with characteristics of continuous simulations as part of its fitness evaluation.

We present preliminary results obtained from a project carried out in cooperation with an Australian ASX listed company manufacturing agricultural chemicals.

I. Introduction

Understanding and managing a company's supply chain is one of the hardest tasks procurement planners and supply chain managers face in today's business environment. Ideally, a supply chain is driven by demand generated by customers placing orders. An order may include one or many order items which are composed of processed and unprocessed raw materials and components. In addition to the customers' orders which, generally speaking, draw final product out of the supply chain network (pull factors), supply chain networks are often subject to push factors which are caused by suppliers feeding raw material into the supply chain (push factors). In most cases, the supply of raw materials is not synchronisable with the demand generated at the other end of the supply chain, or an adjustment is delayed. Suppliers constantly, periodically or spontaneously deliver raw material or components into the supply chain network. An arbitrary imbalance is created by customers and suppliers which the supply chain network tries to even out or trade-off (see Figure 1).

In this paper, we describe ways to synchronise and reconcile the drivers of supply chain networks demonstrated on a real-world example of an Australian manufacturer of agriculture chemicals. The company employs a vertically integrated multi-echelon supply chain. As part of it, various plants convert raw materials obtained from mines, shipments and as by-product of external production processes into intermediate and eventually into a range of final products. Some of the ways to procure raw materials are subject to contractual obligations (hard constraints), some of them can be regulated as they are part of the company's own sourcing operations. Although the model presented in this paper aims to represent a specific supply chain model of a particular manufacturer, the components of the network represents supply chain entities as they can be found in many other businesses.

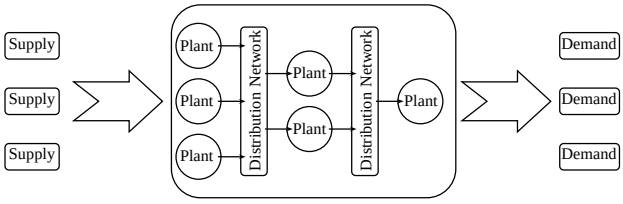

Fig. 1. Schematic view of multi-echelon supply chain network (3-echelon in this case)

Many attempts have been made to develop a unified supply chain model and terminology. Some of them like the Supply Chain Council [1] suggest a common terminology or Rabe et al. [2] underlines configurability and proposes event-discrete simulation as means to analyse supply chains, but this work emphasises on business processes rather than the definition of reusable component as they are desirable to create a programming model. Others like Mackulak [3], Ding [4] and Ganapathy et al. [5] use special modelling languages to express the complexity of business processes and automatically generate simulation models. Although all of them make use of simulations to analyse supply chain networks, they require expert knowledge of modelling languages like Rockwell Software's ARENA, one of the prevalent languages for model supply chain networks. The resulting models may be generated in a programming language that is incompatible to the rest of the system. We

Sven Schellenberg is with SolveIT Software Pty Ltd, Level 1, 99 Frome Street, Adelaide, SA 5000, Australia (email: ss@solveitsoftware.com).

Arvind Mohais is with SolveIT Software Pty Ltd, Level 1, 99 Frome Street, Adelaide, SA 5000, Australia (email: am@solveitsoftware.com).

Neal Wagner is with SolveIT Software Pty Ltd, Level 1, 99 Frome Street, Adelaide, SA 5000, Australia (email: ss@solveitsoftware.com).

Maksud Ibrahimov is with the School of Computer Science, University of Adelaide, South Australia 5005, Australia (email: maksud.ibrahimov@adelaide.edu.au).

Zbigniew Michalewicz is with the School of Computer Science, University of Adelaide, South Australia 5005, Australia. Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland, Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland (email: zbigniew.michalewicz@adelaide.edu.au.

978-1-4244-8126-2/10/$26.00

believe that a generic supply chain model can be developed that supports both, flexibility and ease of use when creating the model without the necessity learn simulation languages. The ideas presented in this paper are implemented in a framework which is employed to model the supply chain operations of a real-world business.

An obstacle when optimising supply chain networks is that various static and dynamic constraints disturb the constant flow of materials; the discontinuity and dynamics of the system make it challenging if not impossible to express the problem mathematically. In order to determine the correct amount of material being fed into and processed by manufacturing components of the supply chain network, simulation seems to be an adequate means. A common downside of simulations however is its computational expensiveness; the state of the system is usually sampled at a fixed predetermined rate (continuous simulation). Another disadvantage of this fixed sample rate is the oblivion of state changes that happen during two sample points. Either the event is totally ignored or it is considered to be occurring at the next sample point. In either way, the simulation gets skewed and the only way to regain fidelity is to increase the sample rate, which adds complexity slowing down the simulation again.

Since the supply chain network reacts on discrete events such as product changeovers, material delivery or plant outages, and parts of it continuously changes (e.g. liquids that are continuously pumped into storage tanks), a combination of discrete-event and continuous simulation seems to be most promising to tackle the problem. Other than most continuous simulations which take a sample of a given system in intervals of a fixed time base, the continuity of the simulation presented in this paper is solely inferred, that is, between any given two adjacent sample points the system's properties are calculated rather than measured. For instance, if we consider a simple system which comprises a factory feeding into a storage tank, the level of the tank will increase linearly with time. Given that discontinuous changes of the system (say, a factory stops its production) are points at which a sample of the system is taken, the continuous flow of material can be inferred between two of these sample points. This assumption excludes any kind of probabilistic change of the system as it is often used in discrete-event simulations to model uncertainty [6]. Modelling uncertainty, however, is not necessary, as the proposed solution serves as a planning tool. Part of the planning process is to run what-if-scenarios which constitutes already the simulation of unlikely events such as leakage of tanks or unplanned outages of factories. With the help of a human domain expert, the system can afford to neglect the simulation of uncertainty.

This paper is structured as follows. After this introduction, an elaborate description of the developed framework is given delving into the model of the supply chain employed as part of the framework, the simulation as well as the Evolutionary Algorithm performing the optimisation process. In addition, the main results are presented and discussed, finishing with the conclusion section.

II. Supply Chain Optimisation

A. The Model

A simulation of a real-world system should be a model that reflects all the important properties which are necessary to verify the simulation and, more importantly, to draw conclusions from the simulation results. While modelling as much business logic and system properties as possible contributes towards the fidelity of the simulation result, the computational expensiveness increases significantly. In order to enable a termination of the simulation within a reasonable amount of time, the model needs to be stripped down as much as possible. Especially, since the simulation is part of the fitness evaluation function and thus repeatedly executed, a simple and efficient way to simulate a model has to be found.

The supply chain network can be thought of as a directed graph $G = (V, E)$ with $V = \{0, ..., n\}$ as a set of vertices and $E = \{0, ..., m\}$ as the set of edges. Each vertex/node can be of any of the three following types:

- Plant
- Storage
- Switch

Although the type of the node determines the different way nodes process material, the three node types have common properties. Each node contains a set of predecessors which they depend on with respect to their source products. These predecessors are defined by the incoming edges of the vertex. Figure 2 illustrates a very simple supply chain network. *Plant A* produces two products, stores both of them in separate storages and *Plant B* converts *Product A* into *Product C* which is finally stored into *Storage C*. Note, that in this simplified version, *Plant A* has no predecessors, whereas *Plant B*'s predecessor is *Storage B*. The conversion from a raw material into an intermediate or final product can be thought of as a chemical reaction with arbitrary input and output products.

A switch, the third kind of supply chain node, is not a physical entity, but it serves to route the material through the supply chain network. Plants and Storages are only allowed to source a material from one predecessor node. In case there are more than one predecessor nodes which output the same product, a switch has to be inserted which controls the material flow (see Figure 3). A switch can either implement a local heuristic to determine which successor to deliver to or it may be controlled by the optimiser which evolves the routing as part of its usual individual reproduction process (the switch becomes part of an individual's genotype). An example for the former may be that *Switch A* is implemented

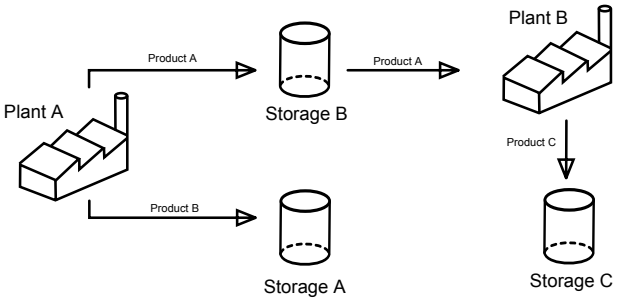

Fig. 2. Supply Chain model showing relationship between supply chain nodes

in a way such that it always exhausts the capacity of a tank before it starts filling up another tank (see Figure 3, *Switch A*). The local deterministic heuristic incorporated into *Switch B* may reduce the storage's level evenly draining all tanks at the same time.

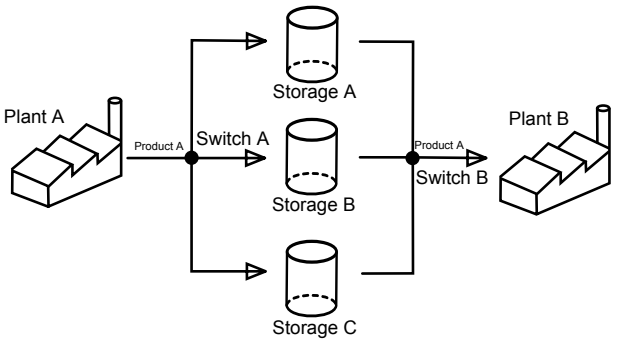

Fig. 3. Supply Chain illustrating material flow routed by Switches

In addition to nodes having predecessors, nodes also contain output buffers (one per product) in which the products they produce or store is kept. These output buffers temporarily store the produced material and its quantity. Switches do not contain output buffers, but delegate requests for material to their predecessors depending on their implemented routing logic.

The previously described node-predecessor relationship only works for continuous material flow such as liquids that are pumped from a producing plant into a storage tank. In many supply chain networks the distribution of material is done via transportation means with limited capacity or infrequent transportation times (trucks, ships, airplanes or railway). Therefore, an additional property of the edge between two nodes is a schedule defining the availability of the transportation means and its capacity. Only when the transportation means is available at the plant or storage, it can empty the node's buffers and store the material temporarily in its own buffer. Plants either have to shut down or store their products into adjacent tanks at times at which the transportation means is not available.

B. The Simulation

Before the simulation is started, the supply chain nodes have to be initialised and a list of the sample points (event queue) will be created. During the initialisation process of the nodes, the buffers of some storage entities are filled to simulate an opening stock. The next step is to determine the points of the planning horizon at which the system is sampled. These points are the set of all events that occur during the planning horizon, such as

- Change of availability of a plant (plants may run at reduced run rate due to partial maintenance or outages).
- Product changeovers.
- Arrival/departure of transportation means travelling between two nodes.

It is of paramount importance to the validity of the simulation to add all events that cause a discontinuous change of the supply chain network's state to this event queue. If desired, additional sample points can be added to verify the simulation's result.

Once the event queue is filled, the nodes are sorted by precedence. Nodes without predecessors occur first whereas terminal nodes that base their buffer's material and quantity on the result of all previous components' production are located last in this list. The next step is to iterate through the list of predecessors (beginning with the least dependant node) and process the node. Processing a node breaks down into three steps:

1) Pull Resources (raw material, intermediate material, final product, etc).
2) Apply conversion rule.
3) Push converted material into output buffer(s).

The first steps is straight-forward: Since each node knows about its predecessors and the material it desires, it pulls the maximum amount of this material from each of its predecessor nodes. Switches forward the call to their pull method the appropriate predecessor. In a nutshell, the pull step basically boils down to copying the content of the predecessor's output buffers and passing it on to the conversion step.

In the conversion step, the actual business logic for each plant is implemented. If we consider a chemical formula like $1A + 2B \rightarrow 2C + 1D$ with hypothetical elements A, B, C and D, the factory that processes this formula would source all the material it could get for Product A and B and determine the quantities of resulting Product C and D (taking into account the ratio of A:B=1:2). The amount of input material, which has actual been used during the conversion process, is reduced from the predecessor's output buffers to account only for the actual consumption and to determine overproduction. While *Switches* do not implement a conversion routine, *Storages* pass on the incoming materials into their output buffers.

Finally in the last step, the produced material is stored into its output buffers to be available for the successor node's processing procedure. At the end of the sample cycle, that is, once all nodes have been processed, the simulator

captures the state of the system by storing the buffers which contain the quantity/products tuple for each node. The final buffer capacity allows to determine plants' utilisation or storages' remaining capacity.

Upon the completion of each sampling, a second stage which can be considered as part of the optimisation routine checks for any violation of capacity constraints of the storage tanks. Once it finds a constraint being violated, it kicks off the Production Equaliser. This module balances the quantity of produced material and the available storage capacity. It throttles production of plants to eliminate excess production which would overflow storage tanks otherwise. After the adjustment, another partial simulation starting from the previous to the current sample point has to be performed. The equalisation is repeated until all storages' levels are within their specified limits.
The simulation terminates once the planning horizon's end date is reached.

C. The Optimiser

The meta-heuristic used to optimise the supply chain network is a steady-state Evolutionary Algorithm (EA). A population of 50 individuals is randomly initialised and continuously evolved. The process terminates upon stagnation of the search (no improvement for 100 generations) or after 3000 generations. At each generation, two parent individuals are randomly selected to form the base for the offspring individual which replaces the less fitter of the two parents.

As already mentioned, the optimiser incorporates the simulator described in the previous section to evaluate the fitness of the individuals. Objectives of the optimisation process include but are not limited to (a comprehensive list of measures of supply chains can be found in Ganapathy et al. [7]):

- Maximisation of product yield.
- Minimisation of product changeovers.
- Honouring the specified product ratio for the remaining planning period.
- Satisfaction of demand.
- Minimising transportation, inventory, backlogging costs.

The fitness value is constituted by the sum of the weighted objective values (reduction of a multi-objective problem to single objective problem as described by Deb [8]).

In order to converge to a viable solution, the optimiser's individuals encode a list of events (tuples of a date and an action) which forms the input for the simulation. Special operators change the times of events, eliminate or create events and swap sequences of events with other individual's event list (similar to cross-over for Genetic Algorithms). An individual whose fitness value exceeds all other individuals' fitness is kept separately to prevent it from being recombined (keep-the-best elitism, [9]).

Over the course of the optimisation procedure, the optimiser constantly keeps track of its operator's performance. At predefined intervals, the optimiser reviews the operator's performance and applies weights. As part of the recombination step of the EA, those operators with higher weights, that is, those that successfully spawned superior individuals in previous generations, are more likely to be chosen to alter the individual's chromosomes. This is particularly important if the optimiser prematurely converges and the optimisation process gets stuck at a local optimum. The previously preferred operator would not contribute to improving the solution's quality which means other (potentially more explorative) operators come into play.

III. Main Results

As already stated in the introduction of this paper, the system presented is applied to a real-world problem and as such it is difficult to compare it to synthetic problems as they are usually used as a baseline in academia. The only plausible baseline can be obtained by comparing the factory planner's schedule and the expected product yield with the schedule generated by the system. Preliminary test results indicate a high degree of similarity between human and system generated factory schedule with respect to the length of product runs (or with other words the date of scheduled product changeovers) and accumulated product yield at the end of the planning horizon. Taking only these few measures into account, one can conclude that the model adequately represents the supply chain. The time it takes to generate a yearly plan by the planning personnel is tremendous. It takes at least 14 man days to come up with a plan that does not violate any capacity constraints. What-if-scenario analysis becomes virtually impossible. Unless we deal with strategic what-if-scenarios, the result of such a scenario would become obsolete by the time it is obtained. Using the proposed system, an optimal plan for an entire year could be create in less than 10 minutes (on a standard Dual Core 1.6GHz computer optimising a 5-echelon supply chain). This allows for what-if-scenario analysis especially for short term planning of a limited planning horizon which would reduce the

The length of the planning horizon for this test scenario is 1 year. With an objective of maximising the product yield, optimising the whole period at once yielded plans in which decisions were made that appear counter-productive or at least a waste of production capacity. However, after thoroughly analysing the generated plan, it becomes apparent that the optimiser made a decision that may seem to be inefficient in the short term, but is intrinsically judicious in the long term perspective. Trading-off in the short term to benefit from it in the longer term may be overall appropriate when striving for a plan that theoretically yields maximal final product. However, due the dynamics of real-world factory operations, the future advantage may never materialise. If the optimiser decided to wind up the utilisation of a plant to stock up a certain intermediate material and subsequently shut down

the factory entirely, in the context of the simulation and optimisation process this may have been a favourable decision, but in reality the tank storing the intermediate product has to undergo an unplanned maintenance procedure right at the time when it is fully filled. Unless we incorporate uncertainty into the model, the optimiser will always fail to deliver viable results taking into account unpredictable events. Another way to overcome the dilemma of uncertainty and decisions that only pay off in the future would be to optimise the plan for the near future only, say the next quarter, which maximises production and tries to meet demand, that is firm orders. Each period is initialised with the closing stock of the previous period (or the actual stock on hand for the first period). For the rest of the planning horizon the objective is to produce a steady product mix ratio which is based on historical sales data from past years. A downside of optimising each period after another is that the final result will be slightly worse than a result obtained by processing the entire planning period at once. In order to determine the loss of product yield, we created a 5-echelon supply chain network. The objective was to find a factory plan which adheres to all capacity constraints by maximising plant utilisation and thus total product yield. The EA terminates after 2000 generations or if the search stagnates for at least 200 generations. To obtain a normalised result that the total run-time of each period was limited to its share to the overall run. Optimising the whole period at once took about 8 minutes. For the test case with two periods, a maximum optimisation time of 4 minutes was allocated for each of the periods. Essentially, this means we allocated the processing time evenly, as opposed to allowing the EA to exhaust the maximum of 2000 generations for each period (in which case the result would be skewed as the search space is only half the size, but the same amount of computational resources are applied to optimise). The simulation was run four times over the whole period and the planning horizon split in 2, 4 and 8 periods. Figure 4 confirms our initial assumption. The total yields diminished by about 25% comparing an average optimisation run over the entire period to the averaged result obtained by subsequent optimisation over 8 periods. It is worth mentioning that the result improved significantly when each period was optimised for 2000 generations. This implies that the search space for an entire year is too vast for the EA to converge to an optimal solution. It seems to be better to abandon the advantages of a ”global optimisation” of the entire horizon by optimising only reduced areas of the search space eventually combining the solution. However, in such a case, the total run time of the EA will increase.

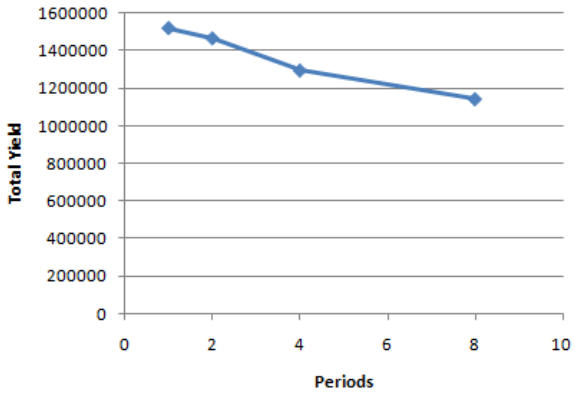

Fig. 4. Decline of total yield when optimising periods in isolation

IV. Conclusions

In this paper we looked at the application of an evolutionary algorithm to the problem of optimising the key decision points in the supply chain network of a major real-world agricultural chemicals company. Modelling the highly complex nature of that company’s operations was the first part of the challenge of successfully accomplishing this endeavour. A balanced mix of discrete and continuous event simulation had to be used. Furthermore, the model was designed in such a way as to be amenable to use within the framework of an evolutionary algorithm.

Of key importance, was developing the ability to represent the decision points in the supply chain network simulation as entities that could be manipulated by evolutionary operators. This was achieved in two way, first by allowing the timing of decision events to be determined by values within the candidate individual representation, and also the types of those decision events. Thus, for example, a candidate individual could specify that the operation of ”change from product A to product B” could be scheduled to happen in a particular part of the network on a given date and time. Secondly, the processing logic of ”switching” nodes of the network could be manipulated by evolutionary operators. For example, the production of a particular chemical could employ materials from a variety of sources, with there existing complex business contractual rules associated with each source. In some cases, it might be easy to determine a set of rules to spell out the correct routing logic to use. In other situation, it may not be obvious and thus it would be preferable to let the individual represent decision making logic, as part of the encoding, and then evaluate performance of evolved logic as a component of the overall fitness evaluation process.

The results presented in this paper also illustrate the trade-off that is frequently accepted by business managers in practice. By running the software in a global mode over a large timeframe, a relatively high optimal result could be achieved, but the validity of such a result could be called into question by human managers who would rightly point out that the further out into the future that assumptions are made about supply chain conditions, the less reliable those assumptions would be. Hence we chose to apply the simulation/evolution algorithm over the whole timeframe in phases, starting with a short term phase of a few months, and then looking further into the future. This gave our approach the benefit of seeking higher levels of optimisation

in the short term, wherein knowledge of conditions is quite firm, and then freezing those results and progressively expanding the scope of inclusion to seek out optimisation further into the future.

As constructed, the software application arising from the event simulation model and the evolutionary algorithm presented in this paper, was able to provide invaluable insight and speculative modelling (”what-if”) capabilities to managers of the client company, allowing them to find ways to optimise their supply chain network, and of course maximise production. This is the litmus test of the value of this application, and it is a rewarding application of the power of evolutionary computation to a real-world business.

ACKNOWLEDGMENT

This work was partially funded by the ARC Discovery Grant DP0985723 and by grant N516 384734 from the Polish Ministry of Science and Higher Education (MNiSW).

REFERENCES

[1] Supply Chain Council, “Supply-Chain Operations Reference-model Version 9.0,” 2008, http://www.supply-chain.org/ [2010-01-15].

[2] M. Rabe, F.-W. Jaekel, and H. Weinaug, “Reference models for supply chain design and configuration,” in *WSC ’06: Proceedings of the 38th conference on Winter simulation*. Winter Simulation Conference, 2006, pp. 1143–1150.

[3] G. T. Mackulak, F. P. Lawrence, and T. Colvin, “Effective simulation model reuse: a case study for amhs modeling,” in *WSC ’98: Proceedings of the 30th conference on Winter simulation*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1998, pp. 979–984.

[4] H. Ding, L. Benyoucef, X. Xie, C. Hans, and J. Schumacher, “”one” a new tool for supply chain network optimization and simulation,” in *WSC ’04: Proceedings of the 36th conference on Winter simulation*. Winter Simulation Conference, 2004, pp. 1404–1411.

[5] S. Ganapathy, S. Narayanan, and K. Srinivasan, “Logistics: simulation based decision support for supply chain logistics,” in *WSC ’03: Proceedings of the 35th conference on Winter simulation*. Winter Simulation Conference, 2003, pp. 1013–1020.

[6] A. Law, *Simulation Modeling and Analysis (McGraw-Hill Series in Industrial Engineering and Management)*. McGraw-Hill Science/Engineering/Math, 2006.

[7] A. Gunasekaran, C. Patel, and E. Tirtiroglu, “Performance measures and metrics in a supply chain environment,” *Internation Journal of Operations & Production Management*, vol. 21, no. 1, pp. 71–87, Feb. 2001.

[8] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, 1st ed. Wiley, 2001.

[9] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1996.