A multi-agent approach for the capacity allocation problem

Armando Brandolese, Alessandro Brun, Alberto Portioli-Staudacher*

Dipartimento di Economia e Produzione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy

Received 5 February 1999; accepted 20 December 1999

Abstract

This paper proposes a new approach for the problem of allocating production capacity to multiple requirements, based on the Multi-Agents paradigm. The proposed system is a distributed, decentralised system that presents many advantages over the traditional centralised algorithms. Among the others it is more flexible, more reactive, and easily understandable. Moreover, a prototype of personal digital assistant is described. Such an assistant is able to support distributed decision-making in companies where several product lines compete for the same resources, in a contest of swiftly changing and scarcely foreseeable demand. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Capacity allocation; Multi-agent systems; Distributed systems; Negotiation

1. Overview

This paper focuses on organisational and managerial issues related to production capacity management in the middle term. For this work, capacity is regarded as given, being the result of long-run corporate decisions except for little adjustments as, for instance, overtime.

The low degree of flexibility characterising production capacity in the middle term leads almost inevitably to criticality when turbulent markets are concerned. Problems in managing turbulent markets are not due to the far-from-steady demand profiles, since in seasonal markets even stronger fluctuations are usually easy to manage; rather, what is troublesome is the fact that quite normally significant variations in the demand rate arise suddenly and in an unpredictable way.

In recent years the search for increased flexibility brought improvement in production systems in terms of set-up times and costs reductions, more accurate demand forecasts and more sophisticated capacity allocation algorithms. Nonetheless, the capacity allocation systems available so far seem not completely satisfactory.

This paper will introduce a new approach in the realm of capacity allocation methodologies, where no new algorithm is presented, but a managerial and organisational tool is proposed with the aim of supporting the human decision making, rather than substituting it.

The second and third sections of this paper are dedicated, respectively, to a deeper analysis of the problem, and to a better definition of the considered industrial environment. In Section 4 a brief survey of Multi-Agent Systems (MAS) is presented,
while Section 5 constitutes the proposal of a new architecture, which builds upon the MAS paradigm. Section 6 contains additional considerations regarding a perspective industrial implementation of the proposed architecture. Simulation results are commented upon in Section 7; while more general comments, arising looking at the architecture from a managerial perspective, are provided in Section 8. Conclusions are given in Section 9.

2. Problem relevance

The number of markets presenting a significant level of turbulence is increasing every year. Behind the concept of turbulence gather a wide set of characteristics typical of contemporary markets, among which speed and change are those more relevant. A high degree of turbulence implies demand unsteadiness – following from the very definition of market speed1 – and unpredictability of future trends on the basis of historical data – as a consequence of market structural changes.

Anyway, this definition of turbulence does not just mean unexpected threats. Turbulence means changes, unforeseen events, as well as opportunities, therefore a new strategy is emerging: the ability to exploit these opportunities.

There are several approaches to satisfy sudden variations in the end products’ demand, among which are the following most common:

- allowing an extra amount of production capacity. In this case, the gap between the production capacity and the average demand, a capacity buffer, will remain unused for most of the time, a waste of resource following. The buffer size will be the greater, the lower the degree of demand predictability and the higher the gap between its maximum and medium values.
- improving the production management system, in order to reduce as much as possible, or even to eliminate, each kind of inefficiency, and to improve resource allocation.
- recurring to stock. Production will therefore take place in advance with respect to the period in which demand is to be satisfied, thus reducing the probability of shortages even in case of sudden demand leaps.

Let us focus on the case of end-items showing high-variance, yet not-correlated demand patterns: in many cases it is reasonable to regard the total demand as somewhat steady. As a consequence, it would be definitely cheaper to keep some spare capacity related to the most critical resources, shared by the whole of product lines, rather than keeping in stock a certain amount of finite product for each product line. This disadvantage of the stock approach sharpens as the uncertainty increases, eventually bringing to unbearable inventory carrying costs. Moreover, in some cases recurring to stock is not even a feasible solution. Typical examples are those of products related with rapidly evolving technologies, with fashion industry, presenting too short a life cycle to create stocks, or with perishable products. Even more relevant is the case of services: the production is compelled to take place at the same time the demand does. As a consequence, demand satisfaction requires spare capacity.

Anyway extra capacity costs, so managers want to keep the extra capacity to a minimum. Improving the capacity allocation process is the main way to reduce the capacity buffer needed.

Therefore, the main aim of this paper is to develop a framework supporting the capacity allocation process, tailored to achieve both effectiveness – leading to allocations consistent with overall business goals – and efficiency, in terms of low response times – thus reducing to a minimum the amount of overcapacity necessary to cope with demand fluctuations and uncertainty.

The proposed architecture was tested in the case of several product lines, competing for the same resources with strongly variable mix quantities. In fact, in such a case a capacity allocation mechanism is needed, which is able to compensate unbalanced demands by mean of a dynamic allocation of the critical resource.

---

1 Marked speed could be regarded as the inverse of the time lag between significant variations in the demand profile parameters, as mean value, variance, trend, seasonality, and so on.
3. A sample scenario

It is at this point convenient to set a reference scenario for the rest of the paper, so to make it easier to focus problems, issues and solutions.

Let us consider the general case of a company producing several product lines, characterised by idiosyncratic demand profiles, and competing for the same resources. If the company is large enough it is usually organised in commercial divisions operating with a high degree of autonomy. We may consider, for example, the case of the textile industry for the fashion market: very often a big enterprise controls a few production units whose capacity is shared by several product lines (in the silk industry there are common fabrics that are then used for ties, foulards, women’s clothes, and so on). The production units have to satisfy the requirements of the product lines; yet in many cases the overall capacity is not enough to satisfy all requirements, therefore choices have to be made.

In particular, the capacity allocation process has to deliberate about how much of the sales budget of each commercial line has to be produced in each production unit.

Traditional capacity allocation systems normally solve the capacity allocation problem by means of a process, which usually consists of two main steps:

1. a periodic meeting where managers of commercial divisions provide capacity requirement budgets, and managers of production units report the expected available capacity of the plants. Normally, some sort of rough cut capacity check is performed and, whenever evident inconsistencies are highlighted (this is most often the case), managers try to attenuate the conflict, planning overtime, and/or reallocating the requirements of the various divisions. It is a long process, where managers utter sentences like: “I’ll wait two more weeks for these 2,000 meters, provided that you deliver me those other 3,000 meters with no delay at all”; “My production unit could allow some overtime this month, so to satisfy your requirements, but next month overtime is up to another unit”. Moreover, divisions managers base their requirements on forecasts, and forecasts are uncertain. Managers tend to reserve capacity also for sales that are far from sure: nobody wants to miss a sale just because he or she did not reserve enough capacity. This increases the requirements and makes the capacity allocation more difficult.

2. a planning system (such as the Aucamp LP model [1], or the Karni-Roll algorithm [2]), which elaborates data assessed during last meeting and searches for a feasible answer.

The main problems of the above-described process are the following:

1. the decision-making process is a rather cumbersome procedure, involving subjective judgement, personal feelings, harsh negotiations. The personality of the managers and their relative position inside the company play a relevant role in this process. Changes in the agreed allocation are normally seen as harassment, and therefore avoided; this leads to inefficient resources allocation and rigidity to changes in the requirements;

2. in order to diminish the ineffectiveness and inefficiency of the above-described process, a branch of the research community is developing more and more powerful algorithms, trying to include as many elements as possible: the algorithm sets the overtime levels, allocates capacity to the divisions, decides what requirements cannot be satisfied, etc. Apparently, it is not possible to argue against the solution provided by such an algorithm. The problem is that these centralised algorithms cannot help overlooking most of the locally relevant information, because of their bounded computation capacity. Thus, for example, how can they take into consideration the different degrees of uncertainty in the sales (and therefore in the requirements)? Moreover, how can they take into consideration very specific issues as, for example, that (only for the next period) production unit A cannot work overtime if production unit C also has overtime?

On the contrary, the approach presented in this paper addresses directly the negotiation process ahead of the algorithms, so as to give them a good starting point.
4. Multi-agent systems

A Multi-Agent System (MAS) could be defined as “a loosely-coupled network of [asynchronous] problem solvers that work together to solve problems that are beyond their individual capabilities” [3]. The adjective asynchronous indicates that problem solvers operate in parallel; loosely coupled means that individual problem solvers spend a great percentage of their time in computation rather than in communication. The problem solving performed by agents in MAS is referred to as Distributed Problem Solving [4–6]. Distributed means that both control and data are logically and often geographically scattered among different units.

In order for a MAS to solve problems coherently, the agents must communicate with each other, co-ordinate their activities, and negotiate once they find themselves in conflict. [7]

A Multi-Agent System could be considered as an evolution of the object-oriented paradigm [8].

MAS provides a series of advantages, in that [7]:

- MAS constitutes a natural way to handle logically and physically distributed problems;
- they need neither global control nor global data storage;
- they enhance the overall reliability (capability to recover from the failure of individual components, with graceful degradation in performance) and robustness (tolerance to uncertainties in data and knowledge);
- the modularity of such systems allows their extensibility;
- even complicated problems can be solved rapidly;
- their architecture is simple and the way they work is transparent.

In production management, the relevance of these advantages is witnessed by the development of several multi-agent control architectures for manufacturing systems that took place in recent years. They have been referred to as co-operative-distributed control systems [9], fractal systems [10,11], heterarchies [12] and, lately, holarchies [13]. The co-operative nomenclature was firstly introduced in the distributed data processing field [14]; fractals constitute the kernel of Mandelbrot’s fractal geometry [15]; the idea of heterarchy comes from neurophysiology [16]; the word holon was proposed in the late 1960s as an attempt to model social organisations [17].

Basically, MAS architectures developed and tested in production management deal, on the one hand, with short-term real-time scheduling and control [18–23] and, on the other hand, with strategic decision making [24–26]. On the contrary, very little research work has been oriented toward the development of middle-term planning MAS. Nevertheless, we reckon that such issues as middle-term capacity allocation constitute a very favourable field of application for MAS, mainly due to MAS robustness, simplicity of design and capability to solve distributed problems. Therefore, in the next section, a middle-term capacity allocation system based on the Multi-Agent approach is proposed.

5. The proposed approach

5.1. The capacity allocation process

The proposed approach is based on the splitting of the firm into highly autonomous profit centres;² the autonomy of these profit centres concerns short-to-middle-term decisions.

A distinction will be made between commercial divisions acting in different market segments, usually with different product lines characterised by idiosyncratic demand profiles and production units, gathering under a common management various types of resources (raw materials, man-power, machines, transport and stocking facilities).

The main consequence of such a structure is the company organisation diagram being a network, with all nodes, the company divisions, situated at the same hierarchical level, unlike the classic multi-layered tree-shaped diagram characteristic of hierarchical structures.

²A profit centre is “a unit for which the manager has the authority to make decisions on sources of supply and choice of markets” [27].
Production capacity allocation takes place by means of negotiations, in a market-based auction framework, where price levels move toward the demand–supply equilibrium: managers can offer or require production capacity broadcasting their bids.

According to [30], the term negotiation indicates “a process by which a joint decision is made by two or more parties. The parties first verbalise contradictory demands, and then move towards an agreement by a process of concession making or search for new alternatives.” This definition perfectly suits our model, in that the parties are the managers willing to sell or buy production capacity; contradictory demands means that they are not able to find a counterpart; and, finally, new alternatives are generated whenever managers modify the conditions of their bids. Once an agreement is achieved, the parties draw up a contract, in which the supplier, having received a sum called transfer price, pledges itself to make over, within a certain time span, a declared amount of production capacity to the customer.

From the basic situation in which production units sell capacity to commercial divisions, the scenario could evolve freely into a competitive market in which even commercial divisions can offer to each other the previously bought capacity. Commercial divisions can therefore play dynamically the role of resources purchaser or supplier, alternatively asking and offering production capacity to the other units.

5.2. Achieving co-ordination

As stated in Section 5.1, the core of the proposed architecture is a market-based auction framework in which the decision-making process is distributed and negotiation-driven.

Since the decision making is carried out by managers of autonomous divisions, one might well wonder how such a framework could result in local solutions being coherent both with each other and with the business overall objectives. In particular, there seems to be a need for an explicit co-ordination mechanism, since (i) there are dependencies between actions of different divisions; (ii) there is a need to meet global constraints and goals; and (iii) no one manager has sufficient competence, resource or information to solve the entire problem [31]. To answer such an objection we can refer to Coase’s Theorem. In fact, Coase’s Theorem states that the market-based negotiation mechanism, if regulated by a protocol allowing efficient bargaining, leads to global efficiency [32,33], and, according to the value maximisation principle, to the maximisation of business economic value [34]. The final decision does not depend on the bargaining power of the parties or on what assets each owned when the bargaining began: those factors can affect only decisions about how the costs and benefits are to be shared [32,33].

As a result, we could argue that a market-based auction framework in which the decision making process is distributed and negotiation-driven, could well be coherent with the business overall objectives: the search for local optimum leads to a globally optimal solution.

In other words, in case of a hierarchical enterprise, co-ordination is the result of the CEO’s efforts; while bringing the market paradigm into the firm, co-ordination emerges thanks to the market-like mechanism.

We should remember here one of the major contributions of the Transaction Cost Theory: market-like models are Pareto Efficient; yet they (normally) imply significant transaction costs. In other words, in a market economy, prices could evolve towards market-clearing values, and goods

---

3 See [28,29] for the first applications of contracting in distributed computing systems and [4–6] for the first negotiation protocol.

4 Type of resource, amount of production capacity, price, and so on.

5 See Appendix A for these and other theorems and principles related to efficiency.

6 A certain characteristic is said to be emergent whenever (i) none of the elementary particles constituting the system, if taken singularly, presents the same characteristic, and (ii) the characteristic is not added to the system from above, as a law ruling interactions among particles. It is said that the characteristic emerges from the bottom. Auto-organising complex systems built up by simple entities, and variable-sum-games are two among the most common examples of emergence [35–37].
could be allocated in an efficient way, only if the right information is exchanged. Such an exchange of information is costly.

It is not the objective of this paper to contribute to the never-ending debate about whether the cost of a central co-ordinator is higher or lower than the cost related with the identification of the relevant price set in a market-like framework. Rather to propose a new and feasible distributed architecture, allowing to exploit the many advantages of a distributed system.

At any rate, a communication mechanism should be designed, which allows to reduce the above-mentioned transaction costs to a minimum, without compromising the goodness of the final allocation. Auction is “a demand revealing mechanism” [23]: within the auction framework it is therefore possible to drive prices toward the relevant levels, with minimum exchange of information, provided that negotiations are ruled by an efficient negotiation protocol.

5.3. The negotiation protocol

The reader should be aware that the protocol which will be proposed in this section constitutes just an example, among the many architectures supporting a negotiation mechanism; in order to get a more versatile solution, several design parameters will be analysed separately, in the next section those parameters that cannot be set leaving out of consideration the actual environment in which the model is grounded.

We shall therefore proceed with a detailed description of the proposed negotiation protocol.

- Production capacity negotiations take place within restricted time-intervals, called negotiation sessions. We will split the calendar into even periods, each of them lasting (say) a fortnight to a month. There will be one, and only one, negotiation session in each period.
- Within a session it is possible to forbid those negotiation concerning production capacity available in the next period – or, more in general, in one of the next \( N \) periods. The set of these \( N \) periods will be referred to as the frozen period.
- It is possible to negotiate capacity available in one of the \( M \) periods immediately following the frozen period. The totality of these \( N + M \) periods is the planning horizon.
- Sessions are divided into rounds. In every round, managers can both sell their production capacity (putting in an offer), and ask for capacity (submitting a bid).
- An offer is the binding promise to sell a certain amount of a given critical resource’s capacity, in a given period, to the best bid, among those exceeding the stated reservation price.
- A bid is an offer to buy the stated amount of a given resource’s capacity, in a given period, from the cheapest offer, provided that the associated reservation price is lower than, or equal to, the stated reservation value.
- Besides these standard bids and offers, managers can submit exchange bids and offers, thus stating their willingness to pay, or to be paid, for a swap of available production capacity. The main advantage of swaps is the reduction of deadlocks risk (a deadlock is a situation in which, for instance, though having some spare capacity and no immediate need, the manager of a commercial division does not submit any offer and uses the available capacity in advance, just to avoid the risk of running short of that resource sometime in the future; in the same company there could be a manager delaying some deliveries due to a temporary lack of capacity, with no means to anticipate the availability date of the capacity his or her division owns).
- Bids and offers are submitted to a co-ordination entity (as, for instance, a mainframe linked via LAN with every division and production unit); after a correctness and completeness check, bids and offers are broadcast. When a certain time span passes without the submission of any offer or bid, say, two minutes, the co-ordination entity declares the end-of-the-round.
- At the end of every round the co-ordination entity runs a matching algorithm, aimed to find the counterpart to as many bids and offers as possible. In Appendix B the pseudo-code of a matching algorithm is presented, which finds the maximum number of matches allowable, and
minimises the variance of surplus\(^7\) among matchable bids and offers. At the end of the matching algorithm run, results are broadcast, so that every manager could reckon the amount of resources he or she managed to buy or sell. Every bid or offer was a binding promise. Therefore, managers can only acknowledge the results broadcast by the co-ordination entity.

- It could be the case that a manager is unsatisfied with the amount of resources he or she bought or sold within a round. He or she can therefore ask for another round, that will take place immediately. The negotiation session ends when, at the end of a round, each manager is satisfied, that is the case if the co-ordination entity does not receive any request for another round within a time of, say, two or three minutes, after the end of the last round. Note that there is a limit to the session duration (about one hour) that, in any case, cannot be exceeded.

The negotiation protocol presented in this section is derived from the Contract Net Protocol (CNP) originally proposed by [4]. Even if it is not possible to formally prove the optimality of CNP, its ability to support efficient information transfer is witnessed by scores of applications adopting protocols derived from the CNP itself.

### 5.4. Advantages of the proposed approach

After this brief overview of the model, several advantages could be highlighted.

1. Let us ponder the following points:
   - the manager of a division does not have to consider factors that pertain to other areas and does not know well, just because the algorithm logic considers them. On the contrary, he or she can consider all factors impacting on his or her division’s results.
   - negotiation-based decision-making allows the manager to consider with great accuracy local information.
   - managers’ efforts when negotiating are directed towards the achievement of the maximum valued alternative, from his or her division point of view.
   - in every moment of the session each manager can easily evaluate the capacity allocation achieved, and how to change it.

The decision process of a manager is oriented toward local efficiency. In the case of a company split into several small-sized divisions, so that every manager could have an almost complete knowledge regarding his or her division, this mechanism allows local efficiency for every division.

2. The presence of local agents makes the system highly sensitive and reactive to weak signals. It is therefore possible, when performing long-term planning, to reduce the amount of overcapacity.

3. No global information is required. No local information—i.e. information that relates to a single division—is publicly available; nonetheless, the bulk of the firm knowledge is mirrored, thanks to the work of the managers, in the only globally available information: the market prices of the critical resources.

4. Advantages arising from the model’s conceptual clarity and simplicity of design go far beyond effortless adoption and a trivial use:
   - the proposed mechanism provides strong motivation, in that it fosters a more active participation to the decision-making process, making actors aware that the final solution is strictly related to their efforts, and is not imposed by the executives;
   - the mechanism transparency, enforced by the fact that all actors play at the same hierarchical level, reduces the risk of unfairness (due, for instance, to different degrees of personal power within the organisation), thus eliminating the main source of complaints;

\(^7\)The surplus is the difference between the price set in a bid and the transfer price fixed for that specific bid by the matching algorithm. In particular, when bids are concerned the surplus equals the reservation value minus the transfer price, while when offers are concerned the surplus equals the transfer price minus the reservation price. Note that, for a capacity transfer to take place, the surpluses of the matched bid and offer both have to be non-negative.
the simplicity of the allocation process drastically reduces the inertia toward changes. This does not mean that the agreements are unstable, due to a diffuse euphoria toward changes. Rather, the actors will perform an objective evaluation of changes proposals, the element of prejudice toward changes being eliminated.

5. Capacity buffer has an evident cost. And the cost changes from period to period according to the utilisation level of the resource. Each manager can trade off between over requiring capacity for uncertain sales (taking the risk not to be capable of releasing it at a good price), and to wait and buy the capacity later, if the sale is less uncertain.

6. Personal influence in the meeting is less relevant, because the process is more formalised. Moreover, all transactions can be monitored and performances of the managers evaluated from this point of view also.

7. System modularity allows easily the extension of the mechanism to one or more new commercial divisions, production units and suppliers.

8. Within the proposed framework, it is straightforward to monitor several indicators, such as transfer-price levels, convergence-times in negotiations, or even classical indicators such as resources saturation, supporting short-term parameters tuning and providing insights even to long-run decision processes. For instance, the case of a critical resource price falling several times – or, even, being steady – outside the price range within which it is sensible to exchange capacity, could be regarded as a valid warning. The top management should – by mean of long-term planning – increase the available capacity, were the price too high, or reduce the capacity, were the price too low.

While designing the negotiation protocol, maximum attention was paid to the just-mentioned aspects, in order to exploit as much as possible distributed approach advantages.

5.5. PC’S APRON

In order to carry out the resources allocation process as it was described above, division managers are to perform quite a few actions and deal with several sources of information simultaneously. Even though each single action could be regarded as extremely simple, the whole process tends to be complex.

As for most business practices, better process performances could be achieved by providing managers with apt software. In our case, the following software functionalities appear most useful:

- user friendly interface,
- information filtering,
- information retrieval (e.g. BargainFinder, at \(<http://bf.cstar.ac.com/bf/><\), and Jango, at \(<http://jango.excite.com/cf/index.html/><\),
- personal digital assistant (representing buyers and sellers in a virtual market-place, as MAGMA [38], and KASBAH [39]).

For this reason, we developed a software architecture called PC’S APRON (Passive Configurable Screen, Active and PRO-active Negotiator). The following sub-sections describe PC’S APRON functionality.

5.5.1. User friendly interface

As soon as a negotiation session begins, PC’S APRON starts running, supporting managers work with its basic functionality: a friendly user interface.

Managers can easily put in offers and bids, filling an entry box.

Besides automatically performing a bid completeness and correctness check, PC’S APRON warns the user in case formally correct but “suspicious” data were entered (e.g. were an offer reservation price about one tenth of the past average, the software reckoned that the user dropped a digit, and printed a warning message, waiting for the user’s confirmation before submitting the offer).

Each company division is provided with its own data-base, that is up-dated in real-time, and which contains all the information relevant to the division itself (both static data, such as the bill of materials, and dynamic data, such as customers orders and available capacity, as well as data related to a previous negotiation session, such as average transfer price per resource per period). The software allows structured queries to the data-base, providing quickly and in an easy fashion even sophisticated pieces of information (such as the net requirement...
of a certain resource for the next month, given current customers orders and an additional unexpected demand of, say, 100 units).

5.5.2. Passive information filtering

In the default configuration, whenever a bid or an offer is broadcast by the co-ordination entity, PC’S APRON prints on the screen a dialog box. The dialog box contains the bid/offer relevant information (type of resource, quantity, period of availability, reservation value/price, name of the bidder).

To configure the filter, the user must set a boolean expression concerning bids/offers characteristics, from that moment on, bids and offers are printed, if and only if, the value of the expression is TRUE.

By setting the right expression, the user could avoid wasting time examining bids/offers not deemed interesting.

5.5.3. Active information retrieval

PC’S APRON filtering functionality could be switched to the active mode. Once set in the active mode, PC’S APRON autonomously elaborates available information. In the case of a commercial division, for instance, it computes, and keeps updated, the net requirements to face demand in the most likely sales scenarios. It can also forecast future demand trends on the basis of past data. These data serve two purposes: firstly, the software issues warnings when the gap between the capacity requirements and the available capacity turns out to be too big.

PC’S APRON also uses the data to decide which bids/offers to show to the user. It prints every bid concerning a resource for which the available capacity is greater than the requirements; and every offer concerning a resource for which there is a positive net requirement. It might also print offers concerning resources for which no net requirement exists, in case it deemed the reservation price to be so low that the purchase of overcapacity is likely to yield positive returns, also taking into account data such as the probability of unexpected sales, or the cost of commercial actions aimed to increase customer orders. On the other hand, it might print bids concerning resources for which no overcapacity is owned, deeming it safer to sell capacity rather than to keep it to face far-in-time or risky orders. It is important to note that, although being actually active, PC’S APRON keeps on being just a filter: it provides the user with a set of information, but it cannot take any decision.

Filters parameters are expert: the first time that the software is set in the active mode, the user is required to input the initial values. Thereinafter, PC’S APRON modifies the parameters exponentially smoothing the initial parameters and, period after period, the parameters which fit best the user’s habit.

5.5.4. Pro-active negotiation

It is possible to authorise PC’S APRON to accomplish a specific task. It will act autonomously, and print a report once it has completed the task. PC’S APRON can perform two main kinds of task: specific brokerage actions or generic bargaining. When allowing the software to perform specific brokerage actions, the manager must specify:

- which kind of action the software is to perform (purchase or sale);
- which quantity of which resource the software is to buy (to sell);
- which is the maximum (minimum) amount to pay (to receive);
- the tactic to adopt when negotiating and, where appropriate, the negotiation parameters.

---

8 If, for instance, the manager of a production unit had only two resources left – say, lathe and drill – and was not willing to sell the former for less than 20 $ per machine-hour and the latter for less than 15 $ per machine-hour, he or she should set the following boolean expression:

\[ ((\text{Type-of-message} \equiv \text{BID}) \land (\text{Reservation-value-price} \geq 20)) \lor ((\text{Type-of-resource} \equiv \text{DRILL}) \land (\text{Reservation-value-price} \geq 15)) \]

Note that a valid boolean expression could contain nested expressions; therefore within a single expression the manager can set as many conditions as he or she wants.

---

9 Two rather naive tactics are:

(1) ask for (offer) a fixed percentage of the total amount to buy (sell), at the maximum (minimum) reservation price (value). Once bought (sold) the quantity asked for (offered) wait for a specified time interval. Ask (offer) again, until the task is accomplished;

(2) ask for (offer) directly the total amount to buy (sell), but at a reservation price (value) lower (higher) than the maximum (minimum) allowed. After a while, raise (lower) the reservation price (value) by some percentage points, until the task is accomplished.

It is possible to implement scores of more elaborated tactics.
If the software is allowed to generic bargaining, it can issue exchange bids and offers. Exchange bids and offers are placed on a message board. The message board is the way in which several specimens of PC'S APRON, each of those running locally at one division's site, can exchange information. Each specimen of PC'S APRON executed a multi-sided-matching heuristic algorithm, aimed to find feasible capacity swaps. As soon as PC'S APRON spots a feasible set of exchanges, it contacts directly the counterparts, so that they may execute the swaps, without the user intervention. At the end of a session, PC'S APRON might warn the user that another session is desirable. It is anyway up to the user to decide whether to ask for another session.

6. Additional considerations for industry implementation

It is rather clear that, to yield good results, an adequate cultural background is required. Such a background could in some cases be even more important than technical skills about Production Planning and Control (PP&C). Oddly enough, the development of the management’s skills and culture depends, in turn, upon the protocol’s very innermost structure.

Thereby, it is not sensible to expect that the introduction of such a mechanism could find at once the right playground. Rather, the development of an adequate environment will follow a successful protocol implementation.

One could, on the other hand, find on the spot scores of cases in which the introduction of a sound PP&C system did not bring the expected results, because of a badly conducted transient, causing in some cases very long transient times, in some cases flaws in the systems steady state, and in some other cases the utter system rejection.

The proposed mechanism could be implemented gradually, while keeping on running the old PP&C system, thus strongly reducing the risk of failures. Managers could be educated on field about the system working, and the Accounting department could perform a fine parameters tuning without jeopardising the firm results during the whole transient. Thanks to the mechanism transparency, the system could be understood and, as a natural consequence, accepted.

Let us consider, as an example, the following approach to the introduction problem:¹⁰

- the management should keep on running the old capacity allocation process (e.g. quarterly meeting).
- the output of the traditional allocation process will constitute the actual plan. The newly introduced negotiation sessions will be regarded as training sessions and no change in the plan will be actually implemented.
- in the following months the traditional negotiation process will provide the new process with a rough cut plan. Shifts from the rough cut plan will be allowed to some extent. It could be possible to limit negotiation to, say, 5% of the total capacity, and to bound reservation prices/values within a range of, say, ± 15% of the standard transfer price.
- during this period the Accounting department will perform a parameter tuning, monitoring the dynamics of such constrained negotiations.
- the bounds will be incrementally loosened and, eventually, negotiation will become virtually boundless. The old allocation procedure will be hence discarded.

6.1. Design parameters

We deem useful to devote a paragraph to some considerations regarding the choice of those design parameters that are strongly context dependent. We shall therefore analyse the choice of broadcasting bids and offers during the negotiation sessions, and that of using transfer prices.

1. Regarding data publicity, the relevant choice concerns whether or not to broadcast every bid/offer. In the proposed protocol bids and

¹⁰The following example was designed to advocate the simplicity of introduction of the proposed allocation protocol. Anyway, it constitutes just one amongst the several ways in which it is possible to effectively introduce the mechanism.
offers are broadcast. The reason for this choice lies behind the boundedness of managers time; the broadcast of bids and offers significantly shortens negotiation session transient times.\textsuperscript{11}

Were data not broadcast, after putting a bid (offer) in, a manager had been waiting for the end of the session, to ascertain whether his or her bid (offer) matched with some other offers (bids) with higher (lower) reservation values/prices. Deadlocks are likely, with every manager waiting for some other manager to make the first move.

Data broadcasting allows real-time feedback; as a consequence, every manager could change his or her price policy many times within the very same session, and transients might be as short as one single round.

2. Transfer pricing is another critical issue of the proposed architecture. This choice allows the accounting department to evaluate each profit centre’s economic performance separately.

Anyway, one should take into account that transfer pricing is not always possible. In some big-size enterprises, the introduction of such a process for the allocation of some production units to some commercial lines might generate inconsistencies with the overall budget decided at the corporate level and, thus, no more negotiable. In that case, transactions among units could be settled with the payment of fictitious currency units. The manager performance evaluation process will therefore consider each division accounting units balance.

The specific case choice will depend upon additional context-related considerations and, in turn, upon the specific implementation site characteristics.

7. Simulations

From the beginning of our analysis, the need emerged clearly for a sound empirical tool to support theoretical considerations, in order to validate the general model and to discriminate among several likely architectures.

Traditional simulations approach was not apt to satisfy our requirements, not taking into account human behaviour. We deemed role playing approach to be the most suitable to simulate an enterprise decision process. In role playing games every player plays (acts) the part of a main character within the action to be represented. Every manager therefore, abiding by the canons of acting, behaves as if he or she truly were the impersonated character, getting script and reality mixed up.

We developed a role playing support software tool, called “Emul ’97”\textsuperscript{[40]}. Emul ’97 integrates the two main requirements of role playing simulations: firstly, the software simulates the behaviour of several customers of the commercial lines; secondly, supported by a friendly user interface, it performs the role of the co-ordination entity.

Simulations were run with the aid of several MEng students of the Politecnico di Milano. A fictitious enterprise manufacturing and selling guitars and bass-guitars constituted the scenario during the simulation. The case was designed to check, by means of simulation results analysis, whether the proposed model satisfied three basic conditions:

1. The allocation mechanism must work properly in steady-state situations: During simulations reservation prices and values converged toward an equilibrium level and all the resources were allocated, proving the mechanism’s ability to cope with routine situations.
2. The allocation mechanism must be flexible toward fluctuations in capacity requirements: Also, in this case the protocol worked properly. During the high demand sessions, prices tended to rise above standard ranges. Rounds become longer, but no session lasted more than a whole hour.
3. The proposed mechanism should not penalise a division facing unexpected demand fluctuations: In this case Residual Incomes\textsuperscript{12} (RI) of

\textsuperscript{11} Let the negotiation session transient be the period of time, starting at the beginning of the session, during which bids reservation prices and offers reservation values differ significatively.

\textsuperscript{12} Residual Income – i.e. a division’s net income before taxes reduced by a business capital charge – was first used in the Matsushita Corporation of Japan in the 30’s, to evaluate the work of the division managers [41].
different divisions have been compared. Relative differences among RI were acceptable.

We performed three kinds of comparisons:

- differences amongst commercial lines RI were proportionate to the differences in the demand; this result perfectly aligned with the requirement of having strong dependencies among commercial lines profits and market shares;
- differences among production units RI were almost insignificant, but in the case of a serious tactical error of one of the players (this was the case of inexperienced players); therefore, once managers are trained to the protocol’s working (the mechanism cannot work if managers are not sufficiently skilled), any difference in production units RI will not depend upon the negotiating ability of the managers;
- differences between the sum of the commercial lines RI and the sum of the production units RI offset in the middle term, but shown in the short-term peaks of about 10 to 20% of the enterprise RI. Any long-term disequilibrium should therefore constitute a warning signal indicating an uneven balance between requirements and available capacity.

In order to better explain the typical behaviour of players, a sample negotiation run is presented in Fig. 1.

Fig. 1 shows the reservation values and prices of the various bids and offers, related to a certain resource R, submitted within a typical negotiation run. The reader could notice that:

- at the beginning the reservation values are lower than the reservation prices – as a consequence, the capacity allocation cannot take place;
- recognising the above situation, managers of production lines submit offers with lower reservation prices; similarly, commercial managers increase their reservation values;
- moreover, the spread among the various reservation prices is big at the beginning, and tends to decrease as production lines managers receive the information regarding other offers; the same holds true for the reservation values;
- when a bid is submitted, whose reservation value is higher than the reservation price of at least one offer, the capacity allocation can take place; from that point on, reservation prices decrease monotonically, and reservation values increase monotonically, as in “traditional” auctions.

It should be also noted that the level of a certain demand is not always higher than the available capacity. As a consequence, the price at which the offer and the demand curve could be lower in low demand period, and higher in high demand period.

8. Control and management of the capacity allocation system

From this presentation, one could infer that the proposed mechanism was designed to work as a stand-alone system, without any external constraint. To avoid this serious mistake, it is necessary to place the model within the hierarchy of the enterprise decision processes. The units considered in the model are all at the same hierarchical level. Nonetheless, a control hierarchy does exist within every business. The medium-term planning lies below the strategic planning – also called long-range planning; the output of the strategic planning process constitutes an unshakeable constraint for the capacity allocation process.

Having constraints passed from a higher decisional level does not mean that division managers are less free to act within the medium term, nor
does it bound the models effectiveness. As a matter of fact, the proposed negotiation protocol could remarkably improve allocation efficiency in fairly uncertain periods; in steady state periods it bears the same results of the traditional allocation mechanisms, with the clear advantage of lowering high-level managers workloads; rather, high-level managers should operate in those periods in which the negotiation protocol cannot effectively deal with uncertainty.

More specifically, high-level managers could take action in several ways:

- bounding some commercial units to buy a minimum amount of resource. In case of manifest disequilibrium, the supervisor could force a capacity reallocation among two or more units. This allows the top management to keep strategically important, but economically weak divisions from dying out.
- deliberating about managers rewards. Normally, rewards are performance-related, though in some cases rewards falling below or rising above, the limit of fairness could not be applied;
- fixing maximum and minimum price levels for one or more resources. If the cost of one resource is always close to the maximum price allowed, either the upper threshold has been set to a too low value, or the resource is scarce, and capacity should be added to the system.

In order to have the whole system working properly, division managers should be aware of the exact arrangement of the capacity allocation process, within the control hierarchy; moreover, it is paramount the general acceptance of a basic rule: top-management intervention and high-level decisions, adding constraints to our model, constitutes the “natural” working of the enterprise.

9. Conclusions

In this paper, the authors present a medium-term capacity allocation mechanism, based on negotiations among the managers of different divisions. The proposed mechanism, which is not intended to be an algorithm, but a managerial decision support tool, allows to overcome disadvantages of traditional capacity allocation systems exploiting the advantages of Distributed Problem Solving techniques without losing efficiency.

In addition to the proposed architecture, a prototype of Personal Digital Assistant, able to support negotiations among divisions, has been proposed.

Appendix A. Efficiency

Each human being is provided with a certain measure of his or her welfare – economists call it utility function – and he or she likes one situation better than another if and only if the former gives greater utility than the latter. As a consequence, the economic goal of each individual is to maximise his or her measure of satisfaction, $U_i$, depending upon the set of decisions taken in the considered economic system, $D$, and upon the wealth of individual $i$, $E_i$, expressed in money units. In a formal notation $U_i = U_i(E_i, D)$. Since people often have conflicting interests, a universally accepted optimum solution does not exist. It is therefore necessary to define a criterion to distinguish “bad” and “good” options. An option $S^*$ is efficient if and only if there is no available alternative $S$ that is universally preferred in terms of the goals and preferences of the people involved.

Formally

$S^*$ efficient $\iff \not\exists S$ feasible: $(\exists i: U_i(S) < U_i(S^*)) \land ($

$(\exists j: U_j(S) > U_j(S^*))$)

On the other hand, a choice is inefficient when there is an alternative feasible choice for which at least an individual would be better off without harming any other.

Formally

$S^*$ inefficient $\iff \exists S$ feasible:

$(\exists i: U_i(S) > U_i(S^*)) \land (\forall j: U_j(S) \geq U_j(S^*))$

A.1. The efficiency principle

If people are able to bargain together effectively and can effectively implement their decisions, then
the outcomes of economic activity will tend to be efficient (at least for the parties to the bargain).

For an informal proof, see [34].

A.2. Local efficiency vs. global efficiency

Those options that are efficient for a sub-group of the decision makers, are said to be locally efficient; an option is globally efficient only if it is locally efficient, whichever the sub-group. Therefore, local efficiency is a necessary condition for global efficiency.

A.3. No wealth effect

There are no wealth effects for a certain decision maker with respect to a set of possible decisions whenever the following conditions hold:

1. given any two alternative decisions, $D_1$ and $D_2$, there is a definite amount of money, $C_i(D_1, D_2)$, that would be sufficient to compensate the decision maker for switching from $D_1$ to $D_2$.

   In formal notation
   \[
   U_i(E_i, D_1) = U_i(E_i + C_i(D_1, D_2), D_2).
   \]

2. were the decision maker given any amount of wealth, $C_i$ would be unaffected.

3. the decision maker’s wealth is greater than any $C_i(D_1, D_2)$.

A.4. The value maximisation principle

An allocation among a group of people whose preferences display no wealth effects is efficient only if it maximises the total value of the affected parties. Moreover, for any inefficient allocation, there exists another (total value maximising) allocation that all of the parties strictly prefer.

For a formal proof, see [34].

A.5. The Coase theorem

Linking the efficiency and the value maximisation principles, it follows that: if managers in a business are able to bargain together effectively and can effectively implement and enforce their decisions, if their preferences display no wealth effect, and if all the wealth generated by their decisions is distributed among the represented divisions, then the outcomes of the economic activity tend to maximize the firm total value.

The Coase Theorem expresses in a slightly different way the very same result: if the parties bargain to an efficient agreement (for themselves) and if their preferences display no wealth effects, then the value-creating activities that they will agree upon do not depend on the bargaining power of the parties or on what assets each owned when the bargaining began. Rather, efficiency alone determines the activity choice. The other factors can affect only decisions about how the costs and benefits are to be shared [33].

This theorem conciliates the value maximisation goal of every firm and the efficiency goal pursued by the negotiation mechanism proposed in this paper.

Appendix B. The matching algorithm

B.1. Objective

Let $O(r)$ be a set of offers, and $B(r)$ a set of bids, each concerning one unit of the $r$th critical resource. Let $O_i(r)$ be the $i$th offer in $O(r)$, and $B_j(r)$ be the $j$th bid in $B(r)$. Let $P[O_i(r)]$ and $V[B_j(r)]$, respectively, be the the reservation price of the $i$th offer, and the reservation value of the $j$th bid.

Let $C_{kl}$ be the contract drawn up matching the $k$th offer, and the $l$th bid. For $C_{kl}$ to be valid, $P[O_k(r)] \leq V[B_l(r)]$. Once committed in a contract, a bid is satisfied, and is therefore removed from $B(r)$. The same happens for offers.

The goal of the matching algorithm is to find the maximum possible number of matches, thus maximising the number of offers and bids satisfied.

B.2. Constraints

- An offer is not to be matched until all offers with a lower reservation price have been satisfied. In case of several offers with the same reservation price, the algorithm must operate according to a FIFO rule.
- Regarding bids, the rule is the dual: a bid is not to be matched until all bids with a higher reservation value have been satisfied. In case of
several bids with equal reservation value, the algorithm must operate according to a FIFO rule.

B.3. Input

The input is LIST_0, the set of bids and offers accepted by the co-ordination entity during a negotiation session, in the very order in which they were submitted.

B.4. Data Pre-processing

The algorithm starts splitting each bid into atomic bids, i.e. bids and offers related to a single unit of a critical resource. For instance, the offer \( O_i(r) \), concerning \( n \) units of the \( r \)th critical resource, will be substituted by \( n \) identical bids \( O_i,j(r) \) – with \( 1 \leq j \leq n \) – each of which concerns one unit of the \( r \)th critical resource, and with a reservation value of \( p\{O_i(r)\} \).

Algorithm steps
(1) If LIST_0 is empty, the algorithm ends.
(2) Set the counter COUNT to ZERO
(3) Repeat the following cycle, until COUNT reaches \( R \), where \( R \) is the total number of critical resources:
   (a) COUNT = COUNT +
   (b) Create the two lists O(COUNT) and B(COUNT).
   (c) Scan the first element of LIST_0.
   (d) Consider the last scanned element of LIST_0:
      - case END_OF_LIST: go to (h).
      - case \( O_i(r) \), with \( r = \) COUNT: go to (e).
      - case \( B_j(r) \), where \( r = \) COUNT: go to (f).
      - default: go to (g).
   (e) Append the last scanned element of LIST_0 to O(COUNT); then go to (g).
   (f) Append the last scanned element of LIST_0 to B(COUNT); then go to (g).
   (g) Scan the next element of LIST_0, then go to (d).
   (h) If either O(COUNT) is empty or B(COUNT) is empty, go to (q).
(j) Order all the bids in B(COUNT) with decreasing reservation values.
(k) Sub-procedure which counts how many offers cannot be matched. Returns the integer \#_OFF_NM.
(l) Sub-procedure which counts how many bids cannot be matched. Calculates the integer \#_BID_NM.\(^{13}\)

Sub-procedure (k)
(i) Create a pointer POINT_BID, and set it to the first element in B(COUNT).
(ii) Create a pointer POINT_OFF, and set it to the last element in O(COUNT).
(iii) Set the counter BID_CUM to zero.
(iv) Set the counter OFF_CUM to zero.
(v) Set \#_OFF_NM to zero.
(vi) VALUE = \text{max}\{P(POINT_OFF), V(POINT_BID)\}.
(vii) If \( P(POINT_OFF) \geq \) VALUE
   - move \( \text{POINT_OFF} \) to the preceeding element;
   - OFF_CUM = OFF_CUM +;
   - go to (vii).
(viii) If \( V(POINT_BID) \geq \) VALUE
   - move \( \text{POINT_BID} \) to the next element;
   - BID_CUM = BID_CUM +;
   - go to (viii).

Sub-procedure (l)

(i) Create a pointer POINT_BID, and set it to the first element in B(COUNT).
(ii) Create a pointer POINT_OFF, and set it to the last element in O(COUNT).
(iii) Set the counter BID_CUM to zero.
(iv) Set the counter OFF_CUM to zero.
(v) Set \#_BID_NM to zero.
(vi) VALUE = \text{max}\{P(POINT_OFF), V(POINT_BID)\}.
(vii) Order all the bids in B(COUNT) with decreasing reservation values.

Moreover, notice that the algorithm could perform the matching even without knowing \#_BID_NM.

\(^{13}\)The actually implemented algorithm does not execute step (l), since it computes \#_BID_NM solving the following equation:
\[
\text{\#_BID_TOT} - \text{\#_BID_NM} = \text{\#_OFF_TOT} - \text{\#_OFF_NM}
\] (matched bids = matched offers).
(ix) $\#_{\text{OFF,NM}} = \max(\#_{\text{OFF,NM}}, B\text{ID,CUM} - OFF,CUM)$.
(x) If either POINT_OFF is not set to the first element of O(COUNT), or POINT_BID is not set to the last element of B(COUNT), go to (vi).
(xi) Return $\#_{\text{OFF,NM}}$.

References


