in 1972, and received a 3rd Cycle Thesis from the University of Paris IX in 1977.

From September 1973 to September 1976 he worked at the Centre National de la Recherche Scientifique (C.N.R.S.), and he is currently a researcher at the Institut de Recherche d'Informatique et d'Automatique (IRIA). His area of main interest is stochastic control and system theory.

Jean Pierre Quadrat was born in St. Flour, Cantal, France, on June 7, 1946. He received the degree of Applied Mathematic Engineers from the Institut Polytechnique de Grenoble, Grenoble, France, in 1969, and the Engineer Doctor Thesis in 1973 from the University of Paris VI.

Since 1971 he has been a researcher in the Institut de Recherche en Informatique et Automatique, where he has done research in optimal stochastic control.

Optimization by Decomposition and Coordination: A Unified Approach

GUY COHEN

Abstract—**In the general framework of inifinite-dimensional convex programming, two fundamental principles are demonstrated and used to derive several basic algorithms to solve a so-called "master" (constrained optimization) problem. These algorithms consist in solving an infinite sequence of "auxiliary" problems whose solutions converge to the master's optimal one. By making particular choices for the auxiliary problems, one can recover either classical algorithms (gradient, Newton–Raphson, Uzawa) or decomposition-coordination (two-level) algorithms.**

The advantages of the theory are that it clearly sets the connection between classical and two-level algorithms, it provides a framework for classifying the two-level algorithms, and it gives a systematic way of deriving new algorithms.

I. Introduction

SINCE 1960–61 with Dantzig–Wolfe's algorithm [11] in the framework of linear programming, and the Arrow–Hurwicz paper [27], through Takahara's algorithm [26] in 1964 on optimal control, and Lasdon *et al.*'s papers [18], [4] in 1965 on "separable" mathematical programming, there has been an increasing interest in the so-called "two-level" optimization algorithms. One can say that this was the earliest starting point of the theory of large-scale systems (Mesarovic *et al.* [21], 1970).

Since this date, there has been an increasing number of such algorithms proposed in the literature for various optimization problems [28], [29]. Although most of these algorithms have been related to some "coordination principles" by their authors, it seems that there is no general agreement about the number and the classification of these principles. Moreover, it is possible to find examples of algorithms which differ from each other only by the way the "free variables" are chosen, but which fundamentally amount to the same principle (e.g., see Section VII-A).

Such a situation is not satisfactory from the theoretical viewpoint (even if it may be relevant to go more deeply into details for practical purposes). Moreover, the following question naturally arises when considering this new approach of optimization: is there any original principle in the iterative coordination process with respect to classical optimization techniques (gradient, Newton–Raphson...)? Of course, one already knows that "price algorithms" [18] are exactly Uzawa algorithms in the separable case. But the question has not yet been answered in general.

One purpose of this paper is to show that well-known classical algorithms and most of the two-level algorithms can be derived from two basic principles. These principles are not really new and have appeared in many places in the literature on numerical analysis. Consequently, we prove that the essential nature of two-level algorithms is the same as that of classical algorithms. This may place the hard debate between opponents and proponents of two-level algorithms on a new ground.

We shall distinguish between sequential and parallel

Manuscript received March 7, 1977.
The author is with the Centre d'Automatique et d'Informatique, ENSMP, Fontainebleau, Cedex, France.

0018-9286/78/0400-0222$00.75

decompositions. While the former can be considered as a more general way of describing optimization algorithms to be implemented on classical computers, the latter is perhaps the true original feature of the theory of two-level algorithms since it allows one to set relatively important tasks to be achieved in asynchronous parallelism. So, one must consider the usefulness of such a theory with regard to the new possibilities offered by microprocessors arranged into "parallel" computers.

The second purpose of the paper is to give a general framework in which almost every already existing or forthcoming algorithms can be situated. This will contribute to give a clearer, less intricate vision of the theory.

The third purpose, and the most important one, is to provide the user with a powerful tool for designing new algorithms and systematically exploring the various possibilities. On our way, we shall extend well-known methods to a larger class of situations (e.g., the price coordination will be extended to the nonseparable case).

In the process of adapting the following basic algorithms to practical situations, we do not, however, intend to deny the important part played by experience (or art) in making many choices and in bringing one's problem into a form amenable to the use of these algorithms. We shall call this process "problem manipulation" (a term that we borrow from Geoffrion [15]) and whose significance will become more precise hereafter.

The paper is organized as follows. We first present the two basic principles in an abstract framework of convex optimization on Banach or Hilbert spaces. Then we combine these two principles in various ways to build several abstract algorithms. However, the complete study of these basic algorithms (theorems of convergence and proofs) will be published elsewhere [6]. Here we shall pay more attention to showing by several examples how this abstract material can be used in practical situations. This does not mean at all that we intend to make a survey of the plentiful literature on two-level algorithms and to show that all of them can actually be derived from our formalism. We let the reader play the game by himself with any algorithm he is aware of and convince himself whether or not our objectives are met.

II. The Auxiliary Problem Principle and First Family of Basic Algorithms

A. The Auxiliary Problem Principle

Let us consider a functional J on a Banach space $\mathcal{U}$ that we assume l.s.c., convex, and differentiable, a functional J_1 that we assume only l.s.c. and convex, and a closed convex subset $\mathcal{U}^f$ of $\mathcal{U}$. We consider the so-called master problem (MP):

$$\min_{u\in\mathcal{U}^f} J(u)+J_1(u). \tag{1}$$

Now let K be another functional with the same assumptions as for J and consider the following functional depending on some $v\in\mathcal{U}$ and $\epsilon>0$:

$$G^v: u\to K(u)+\langle \epsilon J'(v)-K'(v),u\rangle+\epsilon J_1(u) \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes the duality product between $\mathcal{U}$ and its dual space $\mathcal{U}^*$ (or the scalar product in $\mathcal{U}$ when $\mathcal{U}$ is a Hilbert space canonically identified with $\mathcal{U}^*$). The accent denotes the derivative. Then we have the fundamental lemma.

Lemma 1: Assume that v is such that

$$G^v(v)=\min_{u\in\mathcal{U}^f} G^v(u); \tag{3}$$

then

$$J(v)+J_1(v)=\min_{u\in\mathcal{U}^f} J(u)+J_1(u). \tag{4}$$

Proof: Straightforward from the variational inequality characterizing every minimizing element u^* of a functional $J+J_1$, namely,

$$u^*\in\mathcal{U}^f \text{ and } \forall u\in\mathcal{U}^f:\langle J'(u^*),u-u^*\rangle + J_1(u)-J_1(u^*)\geqslant 0. \tag{5}$$

(See, e.g., Ekeland–Teman [12].) □

The above lemma says that if v happens to be a solution of the problem of minimizing G^v on $\mathcal{U}^f$ [the so-called auxiliary problem (AP)], then it is also a solution of the MP.[1] To find such a v we use a kind of fixed-point algorithm starting from some initial guess u^0, finding u^1, minimizing G^{u^0}, and so on. We can have a functional K as well and a scalar ϵ depending on the iteration index k. So we have the first basic algorithm for some family $\{K^k,\epsilon^k,k\in N\}$:

Algorithm 1:

i) Choose u^0 (preferably in $\mathcal{U}^f$). Set $k=0$.

ii) Solve the AP^k:

$$\min_{u\in\mathcal{U}^f} K^k(u)+\langle \epsilon^k J'(u^k)-(K^k)'(u^k),u\rangle+\epsilon^k J_1(u). \tag{6}$$

Let u^{k+1} be a solution.

iii) Stop if $|J(u^k)+J_1(u^k)-J(u^{k+1})-J_1(u^{k+1})|$ or $\|u^k-u^{k+1}\|$ is below some threshold. Otherwise make $k\leftarrow k+1$ and return to step ii). □

At this point, some remarks are of interest. The above auxiliary problem principle is just a generalization of an idea that has already appeared elsewhere, e.g., in the following. Let us mention Céa [5] ("auxiliary operator method") (as pointed out to the author by Dr. Miellou) and Aubin [1] who himself refers to Lions [19] and Goldstein [17]. The main advantage of this principle, at least in the case when the nondifferentiable part J_1 is not present, rests in the fact that the main features of the AP^k are included in the functional K^k (*which we call for this reason the "core"*), since K^k is only affected by linear modifica-

[1] This property may be related to the concept of applicability in Mesarovic *et al.* [21].

tions (see (6) with $J_1 \equiv 0$) and that, when making the choice for the core, we shall be subjected to rather mild restrictions. Thus, we shall be able to ensure existence and unicity of the solution of (6), as well as good numerical conditioning, and moreover, we shall take care of structural properties of $\mathcal{U}$, $\mathcal{U}^f$ (and possibly J_1) to obtain a decomposed AP^k.

As far as convergence of Algorithm 1 is concerned, we only mention here that under standard convexity and coercivity assumptions on J and the K^k's and Lipschitz continuity of the derivatives, the algorithm is convergent for sufficiently small ϵ^k's. A precise statement and a proof can be found in [6]. Notice, moreover, that the sequence $\{J(u^k)+J_1(u^k)\}$ is monotonically decreasing (except if $u^k = u^*$ for some k).

B. Other Variants

We shall briefly describe here some other variants of Algorithm 1. First, notice that we can divide G^v [see (2)] by ϵ without changing the AP. This can be interpreted as taking $\epsilon = 1$ in (2), but changing the core K into K/ϵ. So, choosing ϵ small enough amounts to taking a core coercive[2] enough. Essentially the same effect can be achieved by changing K into $K+M$ where M is a quadratic functional, e.g.,

$$M : u \to \gamma \|u\|^2, \qquad \gamma > 0. \tag{7}$$

Let $\mathcal{U}$ be a Hilbert space. Then this modification introduces an additional term $\gamma\|u-u^k\|^2$ into (6). This kind of quadratic modification has often been used to ensure convergence. In particular, it is akin to the "proximal point algorithm" (Rockafellar [24]).

Another way of introducing the ϵ is by over-($\epsilon > 1$) or under-($\epsilon < 1$) relaxation schemes. It proceeds as follows. Take $\epsilon = 1$ in (6), but call $\hat{u}^{k+1}$ the solution of the corresponding AP^k and take

$$u^{k+1} = \epsilon^k \hat{u}^{k+1} + (1-\epsilon^k) u^k \tag{8}$$

as the next information to update the AP^{k+1} (make sure that u^{k+1} belongs to $\mathcal{U}^f$ by taking, if necessary, $\epsilon^k \leqslant 1$). This last algorithm converges under the same conditions on the ϵ^k's as Algorithm 1 does (see [6]), although the corresponding sequence $\{u^k | k \in N\}$ is different in general.

III. Relaxation and Relaxed Algorithms

The previous principle will hereafter be related to parallel decomposition (although, as we have seen, it is not intrinsically related to any idea of decomposition). The next principle we shall discuss now is related to the idea of sequential decomposition. Let a decomposition

$$\begin{aligned} &\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_N; \\ &\mathcal{U}_i^f \subset \mathcal{U}_i; \qquad i = 1, \cdots, N; \\ &\mathcal{U}^f = \mathcal{U}_1^f \times \cdots \times \mathcal{U}_N^f \end{aligned} \tag{9}$$

be given

Relaxation[3] *Algorithm:*

i) Choose $u^0 = (u_1^0, \cdots, u_N^0)$. Set $t = 0$, $i = 1$.

ii) Solve

$$\min_{u_i \in \mathcal{U}_i^f} J\left(u_1^{t+1}, \cdots, u_{i-1}^{t+1}, u_i, u_{i+1}^t, \cdots, u_N^t\right) \tag{10}$$

and let u_i^{t+1} be a solution.

iii) If $i = N$ go to step iv). Otherwise make $i \leftarrow i+1$ and return to step ii).

iv) Either stop if some desired level of accuracy is obtained or set $i = 1$, make $t \leftarrow t+1$, and return to step ii). □

Relaxation is a well-known idea in numerical analysis. We now combine the principles of relaxation and of the auxiliary problem into one algorithm. For the sake of notational simplicity, we drop the nondifferentiable part J_1 and do not mention the possible dependence of K and ϵ on the iteration index k. Moreover, we limit ourselves to a decomposition into only component spaces which we call $\mathcal{U}$ and $\mathcal{V}$ containing the feasible sets $\mathcal{U}^f$ and $\mathcal{V}^f$. All these restrictions are clearly not fundamental.

The MP now is

$$\min_{(u,v) \in \mathcal{U}^f \times \mathcal{V}^f} J(u,v). \tag{11}$$

Algorithm 2 (relaxed version of Algorithm 1):

i) Choose (u^0, v^0). Set $k = 0$.

ii) Solve the AP$_u^k$:

$$\min_{u \in \mathcal{U}^f} K(u, v^k) + \langle \epsilon_1 J_u'(u^k, v^k) - K_u'(u^k, v^k), u \rangle. \tag{12}$$

Let u^{k+1} be a solution.

iii) Solve the AP$_v^k$:

$$\min_{v \in \mathcal{V}^f} K(u^{k+1}, v) + \langle \epsilon_2 J_v'(u^{k+1}, v^k) - K_v'(u^{k+1}, v^k), v \rangle. \tag{13}$$

Let v^{k+1} be a solution.

iv) Either decide to stop or make $k \leftarrow k+1$ and go to step ii). □

Notice we can take different ϵ_1 and ϵ_2 since Lemma 1 can be generalized to this situation when (9) holds [and when J_1, if present, is additive w.r.t. to the decomposition (9)]. Notice also that combining the relaxation principle with other variants of Algorithm 1 (see Section II-B) yields some other algorithms, appearing in [6].

IV. First Examples of Decomposition-Coordination Algorithms

Before going to decomposed schemes, let us briefly show the connection of the previous principles with classical algorithms. If $\mathcal{U}$ is a Hilbert space and K is merely

[2]Coercive means that $\exists b > 0 : \forall u, v \langle K'(u) - K'(v), u - v \rangle \geqslant b\|u-v\|^2$.

[3]This is a classical but rather confusing terminology since it has a completely different meaning than in expressions such as over-or-under relaxation used previously.

$$K: u \to \tfrac{1}{2}\|u\|^2, \tag{14}$$

then the solution of (6) (with $J_1 \equiv 0$) is

$$u^{k+1} = \Pi\left(u^k - \epsilon^k J'(u^k)\right) \tag{15}$$

where Π denotes the projection onto $\mathcal{U}^f$. Hence, Algorithm 1 amounts to the classical projected gradient algorithm in this case. If

$$K^k: u \to \tfrac{1}{2}\langle u, J''(u^k)\cdot u\rangle \tag{16}$$

and $\mathcal{U}^f = \mathcal{U}$, then we recover the Newton–Raphson algorithm.

A. General Remarks on the Use in the Decomposition-Coordination Approach

Consider the decomposition (9). Algorithm 2 allows a sequential decomposition of the MP. This scheme can be implemented on a classical computer (single processor). It has the advantage of reducing the amount of information present at the same time in the core memory of the computer. Consider now Algorithm 1 with cores K^k of the form

$$K^k(u) = \sum_{i=1}^{k} K_i^k(u_i). \tag{17}$$

If the nondifferentiable part J_1 is indeed present in (1), we must also assume that it is additive. However, no particular form of J is needed and (17) suffices to make the AP^k break up into N independent problems, yielding a structure of (asynchronous) *parallel* decomposition. This structure is well suited to computers made up of many smaller (micro?) processors.

It is worth mentioning a special technique to define the K_i^k's. Let us first introduce the symbolic notation $\mathcal{R}_i^k$ for some given u^k:

$$\mathcal{R}_i^k u \triangleq \left(u_1^k, \cdots, u_{i-1}^k, u_i, u_{i+1}^k, \cdots, u_N^k\right), \tag{18}$$

i.e., only u_i remains free. Then we may take

$$K_i^k: u_i \to J\left(\mathcal{R}_i^k u\right). \tag{19}$$

Notice in this case that taking $\epsilon^k = 1$ in (6) yields null linear modification terms (of course, convergence must be ensured by, say, quadratic modifications).

A way of comparing relaxed and nonrelaxed algorithms for some given core K is the following. In Algorithm 2, we use the core K and in Algorithm 1 we use an additive core (17) where we define K_i^k by

$$K_i^k: u_i \to K\left(\mathcal{R}_i^k u\right).$$

We shall refer to the former as a Gauss–Seidel scheme and to the latter as a Jacobi scheme, borrowing well-known terms from numerical analysis. In general, Gauss–Seidel algorithms converge faster (in number of iterations), but Jacobi algorithms allow time savings since parallel computations are possible (but, of course, one must have the ad hoc computing tool at one's disposal). Let us also mention more general schemes of computation, namely, "chaotic algorithms" with time lags as studied by Miellou [22].

B. Connection with the Interaction Prediction and Interaction Operator Principle

In [21], Mesarovic *et al.* proposed the so-called interaction prediction principle and showed a technique to modify the "natural" objective function of the subproblems by the so-called linearized interaction operator in order to coordinate these subproblems, i.e., to force their solutions towards the overall optimal. In [7], we have made use of this technique in the framework of optimal control theory.

We shall briefly imbed this work in the above theory, referring the reader to [7] for details.

Let the following optimal control problem be given:

$$\min_u \varphi(x(T)) + \int_{t_0}^{T} L(x,u)\,dt \tag{20}$$

with

$$\dot{x} = F(x,u); \qquad x(t_0) = \alpha \in \boldsymbol{R}^n \tag{21}$$

$$\text{s.t.} \quad u(t) \in \mathcal{U}^f(t) \subset \boldsymbol{R}^m \qquad \text{for a.e.} \quad t \in [t_0, T]. \tag{22}$$

Under mild assumptions, (22) defines a closed convex subset of $\mathcal{U} = L^2([t_0, T]; \boldsymbol{R}^m)$.

Expressions (20) and (21) define a functional J on $\mathcal{U}$, and the MP amounts to minimizing $J(u)$ submitted to (22). Assume that decompositions of $\boldsymbol{R}^m$ and $\boldsymbol{R}^n$ into N component subspaces $\boldsymbol{R}^{m_i}$ and $\boldsymbol{R}^{n_i}$ are given, and notice that the ith part of (21) can always be written as

$$\dot{x}_i = F_i(x,u) = f_i\left(x_i, u_i, h_i(x,u)\right) \tag{23}$$

at least by defining

$$h_i(x,u) \triangleq (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N, \times u_1, \cdots, u_{i-1}, u_{i+1}, \cdots, u_N). \tag{24}$$

Of course, in practical situations, more concise expressions of h_i may be chosen. Let us introduce the interaction variables

$$v_i \triangleq h_i(x,u); \qquad i = 1, \cdots, N. \tag{25}$$

Then (23) can be interpreted as the model of the ith subprocess and (25) as the interconnection equations.

We now define the core K^k in additive form [see (17)] as follows. For any given control history $u^k(\cdot)$ and the corresponding trajectories $x^k(\cdot)$ [through (21)] and v^k [through (25)], consider the expression

$$\varphi\left(\mathcal{R}_i^k x(T)\right) + \int_{t_0}^{T} L\left(\mathcal{R}_i^k x, \mathcal{R}_i^k u\right) dt \tag{26}$$

and define $K_i^k(u_i)$ as the value of that expression where x_i is derived from u_i through the differential equations

$$\dot{x}_i = f_i(x_i, u_i, v_i^k); \qquad x_i(t_0) = \alpha_i. \tag{27}$$

Then, applying the previous theory to compute the linear modifications terms (see below) to be added to the core (see (6) with $\epsilon^k = 1$), we finally find, after rather lengthy calculations (see [7])

$$\int_{t_0}^{T} \sum_{j=1}^{N} (\lambda_j^k)^T \left(f_j(x_j^k, u_j^k, v_j^k)\right)'_{v_j} \cdot \left[\left(h_j(x^k, u^k)\right)'_{u_i} u_i + \left(h_j(x^k, u^k)\right)'_{x_i} x_i\right] dt \tag{28}$$

in which $\lambda^k(\cdot)$ is given by

$$\dot{\lambda}^k = -\left[F'_x(x^k, u^k)\right]^T \lambda^k - \left[L'_x(x^k, u^k)\right]^T;$$
$$\lambda^k(T) = \left[\varphi'(x^k(T))\right]^T. \tag{29}$$

Hence, we define the ith subproblem as that of minimizing the sum of (26) and (28) submitted to (27) and $u_i(t) \in \mathcal{U}_i^f(t)$. Keeping v_i^k fixed during this minimization process [see (27)] is what Mesarovic *et al.* call the interaction prediction principle and (28) is the so-called linearized interaction operator, while computing (29) is the coordination task.

We shall later on derive a so-called generalized Takahara algorithm which looks like this one but with differences which make it more adapted for off-line computation.

However, we have shown in [7] that the above algorithm has nice properties, and in particular, robustness in case of subproblem failures, when used on-line, i.e., when each control history $u^k(\cdot)$ is actually implemented on the interconnected processes and the real state trajectory is fed back to the coordination level.

C. Example of a Relaxed Scheme on Two Levels

This example aims at showing that in some algorithms which can be considered as falling under the type of Algorithm 2, the AP_v^k [see (13)] in fact constitutes the coordination task while the AP_u^k breaks up into N parallel minimizations. This kind of coordination is called "primal coordination" by Findeisen [13]. Moreover, we shall see how the process of imbedding known algorithms in our framework provides a straightforward extension of these algorithms.

Let us consider the problem (11) where we assume that (9) holds and that J has the particular structure

$$J: (u, v) \to \Psi(J_1(u_1, v), \cdots, J_N(u_N, v)) \tag{30}$$

with Ψ being a strictly order-preserving function from $\boldsymbol{R}^N$ to $\boldsymbol{R}$ and J_i a functional on $\mathcal{U}_i \times \mathcal{V}$. This kind of problem has been considered in [13], but we here simplify the situation by assuming that the $\mathcal{U}_i^f$'s are independent from v. Such a restriction can be dropped (see Section VI-C), but this needs results that will be given later on.

The problem considered here can be reformulated by introducing the functional

$$\varphi: v \to \min_{u \in \mathcal{U}^f} J(u, v) = J(\hat{u}(v), v). \tag{31}$$

Then, the upper level must minimize φ on $\mathcal{V}^f$ while the lower level is in charge of computing the minimizing $\hat{u}(v)$'s, a task which splits up into N independent problems thanks to (30). The upper-level problem can be solved by a projected gradient algorithm. Under various assumptions of continuity, continuous differentiability in v of the functionals involved, and mainly *unicity* of $\hat{u}(v)$ (see Danskin [10]), φ is Fréchet-differentiable with

$$\varphi'(v) = J'_v(\hat{u}(v), v). \tag{32}$$

Thus, the coordination task turns out to be

$$v^{k+1} = \Pi\left(v^k - \epsilon_2 J'_v(u^{k+1}, v^k)\right) \tag{33}$$

where Π is the projection on $\mathcal{V}^f$ and $u^{k+1} = \hat{u}(v^k)$.

We now drop the assumption on the structure of J [see (30)] and we consider a core of the form

$$K^k: (u, v) \to \sum_{i=1}^{N} K_i^k(u_i) + \tfrac{1}{2}\|v\|^2. \tag{34}$$

For example, one may choose (for (u^k, v^k) given)

$$K_i^k: u_i \to J(\mathcal{R}_i^k u, v^k). \tag{35}$$

Anyway, we can now apply Algorithm 2 which yields

$$\min_{u_i \in \mathcal{U}_i^f} K_i^k(u_i) + \langle \epsilon_1 J'_{u_i}(u^k, v^k) - (K_i^k)'(u_i^k), u_i \rangle \tag{36}$$

for $i = 1, \cdots, N$ at the lower level and exactly (33) at the upper level. With the particular choices (35) and $\epsilon_1 = 1$, the linear modification in (36) disappears and, moreover, when assuming (30), we again obtain Findeisen's algorithm (with the simplification previously underlined). Thus, we have shown that assumption (30) can be dropped while still obtaining a decomposition algorithm. However, we do not mean that (30) does not play a part in motivating such a coordination stucture and, maybe, in convergence of the algorithm.

Notice that in our approach, the "problem manipulation" which consists of introducing the functional φ (and consequently which needs to know Danskin's theorem) is avoided. This illustrates the fact that this approach provides systematic guidelines. The cleverness in manipulating the problem is here somewhat shifted onto the choice of a particular core, of a relaxed scheme, and so on.

V. The Auxiliary Problem Principle in Optimization Problems with Explicit Coupling Constraints

Until now, when dealing with implicit constraints such as $u \in \mathcal{U}^f$ and with decomposition, we have been assuming that these constraints were decomposed (i.e., (9) holds). We shall now drop this assumption, but this re-

quires that coupling constraints be explicitly formulated as equality or inequality constraints in Hilbert spaces. So we consider

$$\min_{u \in \mathcal{U}^f} J(u) \tag{37}$$

$$\text{s.t.} \quad \Theta(u)=0 \quad \text{or} \quad \Theta(u) \in -C \tag{38}$$

where Θ is a mapping from $\mathcal{U}$ into the Hilbert space $\mathcal{C}$ and C is a closed convex cone in $\mathcal{C}$ with nonempty interior $\mathring{C}$. Notice that, at least formally, we can, simultaneously, deal with both cases of equality and inequality constraints by setting $C=\{0\}$ in the former case. Then, from the definition of the positive conjugate cone C^* of C in $\mathcal{C}^*$ (see, e.g., Luenberger [20]), we find $\{0\}^*=\mathcal{C}^*$.

We shall keep on dealing with the convex case, i.e., J is convex, $\mathcal{U}^f$ is too, and Θ is convex in the following sense:

$$\forall u,w \in \mathcal{U}, \forall \alpha \in [0,1], \quad \alpha\Theta(u)+(1-\alpha)\Theta(w) - \Theta(\alpha u+(1-\alpha)w) \in C. \tag{39}$$

Notice that in the case $C=\{0\}$, this amounts to saying that Θ is affine. The classical so-called Lagrangian is

$$L(u,p)=J(u)+\langle p,\Theta(u)\rangle \tag{40}$$

with $p \in C^*$ (i.e., $p \in \mathcal{C}^*$ in the case $C=\{0\}$, which is well suited to the equality constrained case). We recall (see [20]) that solving (37)–(38) is equivalent to finding a saddle-point of L on $\mathcal{U}^f \times C^*$, provided that some so-called constraint qualification condition holds (in the inequality case a simple one is that there exists some $\bar{u} \in \mathcal{U}^f$ such that $\Theta(\bar{u}) \in -\mathring{C}$). We shall assume that this condition holds for the MP (37)–(38).

Thus, we focus our attention to searching for a saddle-point of L on $\mathcal{U}^f \times C^*$. With the convexity assumptions we have made and with obvious differentiability assumptions, this is equivalent to solving the two variational inequalities:

$$u^* \in \mathcal{U}^f \text{ and } \forall u \in \mathcal{U}^f, \quad \langle L'_u(u^*,p^*), u-u^*\rangle \geqslant 0 \tag{41}$$

$$p^* \in C^* \text{ and } \forall p \in C^*, \quad \langle L'_p(u^*,p^*), p-p^*\rangle \leqslant 0. \tag{42}$$

We can also replace (41) by

$$\langle J'_u(u^*), u-u^*\rangle + \langle p^*, \Theta(u)-\Theta(u^*)\rangle \geqslant 0 \tag{43}$$

when dealing with $\langle p^*,\Theta(u)\rangle$ as we did with the nondifferentiable part J_1 [see (5)].

Then, we use the same idea as previously, that is, we choose a core $\Psi(u,p)$ and we add to it linear modifications in order to have an extension of Algorithm 1 to this case. However, due to the presence of two groups of variables, we shall consider two different kinds of algorithms: one when the successive AP's consist of saddle-point problems (i.e., u and p are dealt with simultaneously) and the other when minimization and maximization problems are solved alternately [i.e., we have a relaxed scheme in (u,p)]. It is worth noticing that, corresponding to each case, we shall make particular choices of the form of the core Ψ in order to recover already known algorithms. Hence, we have the feeling that we are far from having exhausted all the possibilities offered by such a general way of thinking.

The two kinds of algorithms we have just mentioned will be referred to as one-level and two-level algorithms in the following for reasons that will become clearer later on.

A. One-Level Algorithms

We choose a core assuming a Lagrangian form, namely,

$$\Psi:(u,p) \to K(u)+\langle p,\Omega(u)\rangle \tag{44}$$

where K and Ω are mappings of the same kind as, respectively, J and Θ. Then, for given (u^k,p^k) the AP^k consists of finding the saddle-point on $\mathcal{U}^f \times C^*$ of the functional

$$K(u)+\langle \epsilon J'(u^k)-K'(u^k),u\rangle + \langle p^k,(\epsilon\Theta'(u^k)-\Omega'(u^k))\cdot u\rangle + \langle p,\Omega(u)+\rho\Theta(u^k)-\Omega(u^k)\rangle. \tag{45}$$

The positive real numbers ϵ and ρ correspond to the linear modifications, respectively, in u and p. From the sufficient part of the Lagrangian saddle-point theorem (see [20]), solving the above AP^k gives the solution of the constrained optimization problem:

$$\min_{u \in \mathcal{U}^f} K(u)+\langle \epsilon J'(u^k)-K'(u^k),u\rangle + \langle p^k,(\epsilon\Theta'(u^k)-\Omega'(u^k))\cdot u\rangle \tag{46}$$

$$\text{s.t.} \quad \Omega(u)+\rho\Theta(u^k)-\Omega(u^k) \in -C. \tag{47}$$

The problem of the existence of a saddle-point of (45) or of a strictly feasible solution of (47) for every u^k will not be considered in detail. From the practical viewpoint, notice that K and Ω must be chosen if possible for that purpose.

Algorithm 3:

i) Choose (u^0,p^0). Set $k=0$.

ii) Solve the problem (46)–(47). Let (u^{k+1},p^{k+1}) be a solution.

iii) Either stop or make $k \leftarrow k+1$ and return to step ii). □

In [6] and also in [9] we have shown conditions of convergence in the restricted case where J and K are quadratic, Θ and Ω are linear, $C=\{0\}$, and $\mathcal{U}^f=\mathcal{U}$.[4] One feature which must be strongly underlined is the fact that, unlike previous or forthcoming algorithms, these conditions of convergence cannot be met merely by choosing a sufficiently small ϵ (and $\rho=\epsilon$). To have such a property, the following condition must be satisfied. Let A be the Hessian operator associated with J; then $\Theta A^{-1}\Omega^* + \Omega A^{-1}\Theta^*$ must be coercive. This latter operator is a kind of measure of resemblance between Θ and Ω. The above

[4]Notice that in this case we are able to guarantee existence and uniqueness of both (u^{k+1},p^{k+1}) and (u^*,p^*), the master solutions.

condition interferes with structural conditions on Ω (e.g., Ω must be block-diagonal) in the context of decomposition in a way which is not perfectly understood at the present stage.

B. Two-Level Algorithms

We now choose a core assuming a form

$$\Psi:(u,p)\rightarrow K(u)-\tfrac{1}{2}\|p\|^2 \tag{48}$$

and, as previously indicated, we use a relaxation principle between variables u and p. Moreover, we can deal with the part $\langle p,\Theta(u)\rangle$ as we did with the nondifferentiable part J_1 in (6) [see (43)]. This yields

Algorithm 4:

i) Choose (u^0,p^0). Set $k=0$.

ii) Solve the AP_u^k:

$$\min_{u\in\mathfrak{U}^f} K(u)+\langle \epsilon J'(u^k)-K'(u^k),u\rangle+\epsilon\langle p^k,\Theta(u)\rangle \tag{49}$$

Let u^{k+1} be a solution.

iii) Update p after

$$p^{k+1}=P\left(p^k+\rho\Theta(u^{k+1})\right) \tag{50}$$

where P is the projection onto C^*. ($P=I$ if $C=\{0\}$.)

iv) Either stop or make $k\leftarrow k+1$ and return to step ii). □

Notice that (50) gives the solution of the AP_p^k because of the particular dependence of Ψ on p. We have been able to prove the convergence of this algorithm under classical assumptions (see [6]) for sufficiently small ϵ and ρ.

It must be pointed out that, under these assumptions, u^{k+1} is unique, as is u^*. However, p^* may be nonunique. Consequently, we can only prove that the sequence $\{p^k\}$ is bounded, unless further assumptions are required.

We now consider a variant when Θ is assumed to be differentiable and when the part $\langle p,\Theta(u)\rangle$ is included in the differentiable part of the MP. However, in order to be able to prove the convergence in this case, we shall change the feasible set C^* of p into $C^*\cap B_R$ where B_R is the set $\{p\in\mathcal{C}^*\mid\|p\|\leqslant R\}$. This yields

Algorithm 5:

In Algorithm 4, change (49) into

$$\min_{u\in\mathfrak{U}^f} K(u)+\langle \epsilon J'(u^k)-K'(u^k),u\rangle+\epsilon\langle p^k,\Theta'(u^k)\cdot u\rangle \tag{51}$$

and (50) into

$$p^{k+1}=P_R\left(p^k+\rho\Theta(u^{k+1})\right) \tag{52}$$

where P_R is the projection onto $C^*\cap B_R$. □

Notice that step (52) can be implemented by first applying (50) and then reducing the norm of the resulting p^{k+1} to R by a normalizing factor if necessary. The convergence is proved under the assumption that R be chosen large enough so that there exists at least one optimal p^* in B_R. In this case, changing the feasible set C^* into $C^*\cap B_R$ does not modify the master solution. A kind of Lipschitz continuity of Θ' is also assumed (see [6] for a precise statement).

Notice, at last, that when Θ is affine, problem (51) is equivalent to problem (49) so that Algorithm 4 will be preferably used in this case.

VI. Using One-Level Algorithms in Decomposition

A. General Remarks on the Use in Decomposition

Before considering some decompositon of the AP, we must consider decompositions of the spaces and the implicit constraints involved, i.e., $\mathfrak{U}$, $\mathcal{C}^*$, $\mathfrak{U}^f$, and C^*. For the pair $(\mathfrak{U},\mathfrak{U}^f)$ we assume that a decomposition as (9) is given. We turn now towards the decomposition of $\mathcal{C}^*$. This, in fact, amounts to saying that the constraint space $\mathcal{C}$ is decomposed, that is, the constraints are distributed among the subproblems. Moreover, in the case when $C\neq\{0\}$, we must assume that the positive cone can be written as the product of cones C_i's in the $\mathcal{C}_i$'s.

Then, in order to obtain a parallel decomposition of the AP, we must choose additive cores [see (44)], namely,

$$\Psi^k(u,p)=\sum_{i=1}^{N}\left(K_i^k(u_i)+\langle p_i,\Omega_i^k(u_i)\rangle\right). \tag{53}$$

As previously, we may choose K_i^k by (19) and, in the same way, Ω_i^k by

$$\Omega_i^k:u_i\rightarrow\Theta_i\left(\mathcal{R}_i^k u\right) \tag{54}$$

where Θ_i maps $\mathfrak{U}$ onto $\mathcal{C}_i$.

Notice that $\mathcal{C}$ may be decomposed in less than N components. This amounts to saying that some of the subproblems will not receive (explicit) constraints. On the other hand, one must not allocate too many constraints to some of the subproblems. In the finite dimensional case, one obvious condition is that $\dim\mathfrak{U}_i\geqslant\dim\mathcal{C}_i$.

Sequential decomposition can, of course, be considered also, and is advisable for implementation on a single processor. It is obtained by introducing a relaxation scheme between the variables (u_i,p_i)'s, yielding an extension of Algorithm 2 to saddle-point problems. A numerical comparison of sequential and parallel decomposition is done in [9].

B. Generalized Takahara Algorithm as a Special Case of a Quasilinearization Algorithm

We again consider the optimal control problem (20)–(22) with the additional final state constraint

$$\theta(x(T))=0\in\boldsymbol{R}^r \quad (T\text{ given}). \tag{55}$$

However, as compared to Section IV-B, we here adopt a different viewpoint by considering $u(\cdot)$ and $x(\cdot)$ as "free" variables submitted to two constraints, namely, (21) and (55). The state trajectory $x(\cdot)$ is considered as an

element in the Sobolev space $H^1([t_0,T];\boldsymbol{R}^n)$ and the control history $u(\cdot)$ as belonging to $L^2([t_0,T];\boldsymbol{R}^m)$. Hence, the former constraint is formulated in $L^2([t_0,T];\boldsymbol{R}^n)$ and the latter in $\boldsymbol{R}^r$. Thus, $\mathcal{C}$ is the product of this two spaces and so is $\mathcal{C}^*$ which p belongs to. We have the following lemma.

Lemma 2: In the above-mentioned topology, the optimal multiplier p^* is $(\lambda^*(\cdot),\nu^*)$ given by Pontryagin's minimum principle, namely,

$$\dot{\lambda}^* = -\left[F_x'(x^*,u^*)\right]^T\lambda^* - \left[L_x'(x^*,u^*)\right]^T \quad (56a)$$

$$\lambda^*(T) = \left[\varphi'(x^*(T))\right]^T + \left[\theta'(x^*(T))\right]^T\nu^* \quad (56b)$$

where $(x^*(\cdot),u^*(\cdot))$ are the optimal trajectories. See, e.g., Bensoussan [2] for a proof of this result. □

We now make use of Algorithm 3 by identifying the pair (x,u) with the variable u in the general formalism. We choose the following K and Ω:

$$K(x,u) \triangleq \tfrac{1}{2}(x^TSx)_{t=T} + \tfrac{1}{2}\int_{t_0}^{T}(x^TQx + u^TRu)\,dt \quad (57)$$

$$\Omega(x,u)=0 \Leftrightarrow \begin{cases} \dot{x} = Ax + Bu; \qquad x(t_0)=0 & (58)\\ Cx(T)=0 \in \boldsymbol{R}^r & (59)\end{cases}$$

where A, B, C, Q, R, S are matrices of appropriate dimensions, with Q and S positive semi-definite and R positive definite.

Then, knowing x^k, u^k, λ^k, ν^k, one can compute the modifications given in (46)–(47). Taking $\epsilon=\rho=1$, this yields the following AP^k after straightforward calculations:

$$\begin{aligned}\min\ & K(x,u)\\ &+\left[\left(\varphi'(x^k)-(x^k)^TS+(\nu^k)^T\left(\theta'(x^k)-C\right)\right)x\right]_{t=T}\\ &+\int_{t_0}^{T}\left[L_x'(x^k,u^k)-(x^k)^TQ+(\lambda^k)^T\left(F_x'(x^k,u^k)-A\right)\right]x\,dt\\ &+\int_{t_0}^{T}\left[L_u'(x^k,u^k)-(u^k)^TR+(\lambda^k)^T\left(F_u'(x^k,u^k)-B\right)\right]u\,dt \end{aligned} \quad (60)$$

$$\text{s.t.}\quad \dot{x} = A(x-x^k)+B(u-u^k)+F(x^k,u^k); \quad x(t_0)=\alpha \quad (61)$$

and

$$\left[C(x-x^k)+\theta(x^k)\right]_{t=T}=0. \quad (62)$$

In the case when $\mathcal{U}^f(t)=\boldsymbol{R}^m$, $\forall t$, the solution of the AP is analytic (see, e.g., Bernhard [3]), yielding x^{k+1}, u^{k+1}, λ^{k+1}, ν^{k+1}.

One may think to take $A=F_x'(x^k,u^k)$, $B=F_u'(x^k,u^k)$ and so on at step k, which, in particular, simplifies expression (60). However, it is not advisable to do so, from the practical viewpoint at least, as pointed out in [3], since the resolution of the AP involves a Riccati equation which can be solved once and for all if the data A, B, $C\ldots$ are kept fixed.

This remark does not take into account the question of convergence. As briefly discussed previously, a part of the conditions of convergence can be satisfied by adding to (60) some sufficiently positive quadratic modification, namely,

$$\begin{aligned}&\tfrac{1}{2}\left[(x-x^k)^T\bar{S}(x-x^k)\right]_{t=T}+\tfrac{1}{2}\int_{t_0}^{T}(x-x^k)^T\bar{Q}(x-x^k)\,dt\\ &\qquad+\tfrac{1}{2}\int_{t_0}^{T}(u-u^k)^T\bar{R}(u-u^k)\,dt.\end{aligned} \quad (63)$$

As for the remaining part of the conditions of convergence, it tells us whether the auxiliary linear dynamics and linear final constraint approximate the master nonlinear ones sufficiently closely.

Notice that, once the convergence occurs, it can be proved to be not only in the topology L^2, but also in that of the uniform convergence over $[t_0,T]$.

This quasilinearization algorithm has been discussed somewhat in detail because it directly yields a decomposition algorithm by merely adding some further remarks. Assume that some decompositions of $\boldsymbol{R}^n$, $\boldsymbol{R}^m$, $\boldsymbol{R}^r$ into N (or less than N) parts are given. Choose block-diagonal A, B, C, Q, R, S, $\bar{Q}$, $\bar{R}$, $\bar{S}$ with respect to those decompositions. Then the AP breaks up into N independent problems. This decomposition algorithm appears as a generalization of the earlier algorithm proposed by Takahara in 1964 [26]. It was limited to linear–quadratic problems without coupling constraints (except through the dynamics). Moreover, a correct proof of convergence was missing.

The salient fact of our convergence analysis, although it is limited to the linear–quadratic case, is that the convergence cannot be systematically obtained by using a sufficiently large quadratic modification [see (63)], as sometimes admitted in the literature. There is some threshold in the coupling intensity by the dynamics and constraints. The time interval length is also involved in this analysis, as expected by Takahara.

Notice that the subproblems we have obtained are not only of lower dimensions than the MP, but can also be solved analytically. If some component of $\boldsymbol{R}^r$ is missing, this means that the corresponding subproblem is unconstrained. If this is the case for $\boldsymbol{R}^n$, some subproblem is reduced to static optimization (without state variables). Finally, if this occurs for $\boldsymbol{R}^m$, some subproblem is without control variables and reduces to computation of differential equations.

The fundamental feature that distinguishes this algorithm from that of Section IV-B is that x^k and λ^k involved in the latter are computed by global calculations (at the coordination level),[5] whereas they are exchanged

[5]Or recorded from the real process, as far as x^k is concerned.

between the subproblems in the former. This is more extensively explained in [9]. This feature justifies our terminology of one-level algorithm for the former since the coordination level is clearly missing in that kind of algorithm. The algorithm of Section IV-B has on-line properties, whereas the generalized Takahara algorithm is an off-line computational scheme. These differences illustrate nicely how we can obtain different algorithms by considering either u alone or u and x as the free variables, a choice which falls under the concept of problem manipulation.

C. A Resource Allocation Coordination Method or How One-Level Algorithms Can Split Up Into Two Levels

Let us consider the problem

$$\min_{u_i \in \mathcal{U}_i^f} \sum_{i=1}^{N} J_i(u_i) \tag{64}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \Theta_i(u_i) = \bar{v} \in \bar{\mathcal{V}} \tag{65}$$

which can be interpreted as the optimization of N independent units sharing a common resource $\bar{v}$. The problem manipulation step consists of introducing N new variables $v_i \in \mathcal{V}$, in setting

$$\mathcal{V} \triangleq (\mathcal{V}, \cdots, \mathcal{V}_N);$$

$$\mathcal{V} \triangleq \bar{\mathcal{V}}^N;$$

$$\mathcal{V}^f \triangleq \left\{ v \in \mathcal{V} \mid \sum_{i=1}^{N} v_i = \bar{v} \right\} \tag{66}$$

and in reformulating the problem (64)–(65) as

$$\min_{v \in \mathcal{V}^f} \min_{u_i \in \mathcal{U}_i^f} \sum_{i=1}^{N} J_i(u_i) \tag{67}$$

$$\text{s.t.} \quad \Theta_i(u_i) = v_i; \qquad i = 1, \cdots, N. \tag{68}$$

We intend to use a decomposition-coordination scheme looking like that of Section IV-C, but now we have coupling constraints (68). Thus, we need a relaxed version of Algorithm 3. We shall allocate the constraints (68) to the subproblems in $u \triangleq (u_1, \cdots, u_N)$ only. So let $p = (p_1, \cdots, p_N)$ be the multiplier associated with these constraints. The relaxation will take place between (u,p) on the one hand and v on the other hand. At last, we leave it to the reader as an exercise to derive the following, so-called resource-allocation coordination algorithm by using the core

$$K: (u,v) \to \sum_{i=1}^{N} J_i(u_i) + \tfrac{1}{2}\|v\|^2 \tag{69}$$

$$\Omega: (u,v) \to (\Theta_1(u_1) - v_1, \cdots, \Theta_N(u_N) - v_N) \in \mathcal{V}. \tag{70}$$

Resource-Allocation Coordination Algorithm:

i) Choose $v^0 \in \mathcal{V}^f$ [see (66)]. Set $k = 0$.

ii) Solve the $AP^k_{u_i,p_i}$ for $i = 1, \cdots, N$:

$$\min_{u_i \in \mathcal{U}_i^f} J(u_i) \quad \text{s.t.} \quad \Theta_i(u_i) = v_i^k. \tag{71}$$

Let (u^{k+1}, p^{k+1}) be a solution.

iii) Solve the AP^k_v:

$$\min_{v \in \mathcal{V}^f} \tfrac{1}{2}\|v\|^2 - \langle v^k + \epsilon_2 p^{k+1}, v \rangle. \tag{72}$$

Let v^{k+1} be the solution.

iv) Either stop or make $k \leftarrow k+1$ and return to step ii). □

Let Π denote the projection onto $\mathcal{V}^f$; then (72) is solved by

$$v^{k+1} = \Pi(v^k + \epsilon_2 p^{k+1}) \Leftrightarrow$$

$$v_i^{k+1} = v_i^k + \epsilon_2 \left(p_i^{k+1} - \frac{1}{N} \left(\sum_{j=1}^{N} p_j^{k+1} \right) \right); \quad i = 1, \cdots, N. \tag{73}$$

This way of updating the resource allocation has a nice economic interpretation (see Bernhard [3]), and it has already been proposed as an heuristic in [8]. A proof of convergence is not available at the present time. Notice that the functional in (64) is not coercive when considered as depending on (u,v). Two other technical difficulties lie in the possible nonuniqueness of p^{k+1} (and p^*), except in some special situations (e.g., the linear–quadratic case), and in the possible nonexistence of a feasible solution to (71) for some values of v_i^k (unless we assume Θ_i from $\mathcal{U}_i^f$ to $\mathcal{V}$ to be *onto*).

However, our purpose here has been to show the two steps of problem manipulation and of imbedding a well-known coordination principle (see Geoffrion [16]) in our formalism. Notice also that we could have let the J_i's depend on v as we did in Section IV-C, and that we could have dropped the structural assumptions as in that paragraph.

VII. Price Coordination

Before going to considerations on decomposition, let us show how to recover well-known algorithms from Algorithms 4 and 5. Taking $K = J$ and $\epsilon = 1$ in Algorithm 4 yields the Uzawa algorithm. Considering Algorithm 5 with K given by (14) yields a solution u^{k+1} of (51) which verifies

$$u^{k+1} = \Pi(u^k - \epsilon L'_u(u^k, p^k)) \tag{74}$$

where Π is the projection onto $\mathcal{U}^f$.

A. Price Coordination in the Nonseparable Case

Probably the most famous decomposition principle is the so-called price coordination principle (also known as the "interaction balance principle" [21] or dual coordination as opposed to primal coordination). It fundamentally

rests on the assumptions of separability of the cost functional and of the coupling constraints. More specifically, let us consider again the problem (64)–(65) and let us set $\bar{v}=0$ without loss of generality. The Lagrangian (40) can be written as

$$L(u,p)=\sum_{i=1}^{N} L_i(u_i,p)=\sum_{i=1}^{N}\left[J_i(u_i)+\langle p,\Theta_i(u_i)\rangle\right] \quad (75)$$

so that, when assuming the existence of a saddle-point, and when using the Uzawa algorithm, the minimizing step splits up into N independent problems.

This principle has been used by many authors, e.g., Lasdon–Schoeffler [18] in static situations and Pearson [23] and Tamura [25] in dynamic ones. It is interesting to compare Pearson's and Tamura's approaches. Dealing with the control problem (20)–(22) and making the structural assumptions on the cost (φ,L) and the interconnection equations (25) [namely, $h_i(x,u)=\sum_j h_{ij}(x_j,u_j)$], Pearson takes (u,v) as the free variables [i.e., u in (75) is identified with the pair (u,v)] and identifies (25) with (65). As for Tamura, he does not introduce the interaction variables v_i explicitly, but he takes the (u_i,x_i)'s as the free variables and the dynamics as the dualized constraint, so that the corresponding multiplier is the co-state. In doing so, he obtained both space and time decompositions. We see, once again, the importance of problem manipulation.

However, our main purpose in this section is to point out that Algorithm 4 can be used in the case when J is not separable (or additive) by choosing an additive core K [e.g., after (17) and (19)]. As a matter of fact, problem (49) does break up into N independent problems provided that Θ is additive. If Θ is not additive (but differentiable), then one must use Algorithm 5 [see (51)]. We have thus obtained an extension of the price method to the nonseparable case, and all the previous algorithms related to this method could be revisited by dropping the structural assumptions.

This possibility of using price coordination in the nonseparable case appeared in some papers, e.g., Gabay–Mercier [14]. Moreover, it allows one to deal with so-called augmented Lagrangians, including a term which is the square of $\Theta(u)$ (in the equality case) and which is not additive even if Θ is so.

B. Mixing Price Coordination and the Interaction Prediction Principle

Our purpose in this section is to show that we could go on combining the two basic principles (auxiliary problem and relaxation) to obtain more and more abstract algorithms. However, one should have practical motivations to play this game. We just give an additional example.

Consider once again (20)–(22) with the final constraint (55). One may wish to obtain subproblems which will be optimal control problems, but without final constraints. So one must keep a (decomposed) dynamic constraint at the lower level, but dualize the final constraint. As we did in Section VI-B, let us identify the pair (u,x) with the general free variable u and consider the problem

$$\min_{u\in\mathcal{U}^f} J(u) \quad (76)$$

$$\text{s.t.}\quad \Theta_1(u)=0\in L^2([t_0,T];\boldsymbol{R}^n) \quad (77)$$

$$\Theta_2(u)=0\in\boldsymbol{R}^r \quad (78)$$

where (77) represents the dynamics (21) and (78) the final constraint (55). Then let p_1 and p_2 denote the multipliers respectively associated with (77) and (78); in order to find the saddle-point of the Lagrangian of problem (76)–(78), we choose a core assuming the form

$$\Psi(u,p_1,p_2)=K(u)+\langle p_1,\Omega_1(u)\rangle-\tfrac{1}{2}\|p_2\|^2 \quad (79)$$

[compare it to both (44) and (48)] and we used a relaxed scheme between the variables (u,p_1) on the one hand and p_2 on the other hand. This yields the following algorithm.

i) Choose u^0,p_1^0,p_2^0. Set $k=0$.

ii) Solve the $\mathrm{AP}^k_{(u,p_1)}$:

$$\min_{u\in\mathcal{U}^f} K(u)+\langle \epsilon J'(u^k)-K'(u^k),u\rangle$$

$$+\langle p_1^k,\left[\epsilon\Theta_1'(u^k)-\Omega_1'(u^k)\right]\cdot u\rangle+\epsilon\langle p_2^k,\Theta_2(u)\rangle \quad (80)$$

$$\text{s.t.}\quad \Omega_1(u)+\rho_1\Theta_1(u^k)-\Omega_1(u^k)=0. \quad (81)$$

Let (u^{k+1},p_1^{k+1}) be a solution.

iii) Solve the $\mathrm{AP}^k_{p_2}$ which amounts to making

$$p_2^{k+1}=p_2^k+\rho_2\Theta_2(u^{k+1}). \quad (82)$$

iv) Either stop or make $k\leftarrow k+1$ and return to step ii). □

We leave it as an exercise to the reader to write down the explicit algorithm and make choices for K, Ω_1, Ω_2 in the context of a decomposition of the control problem we have considered (following, in particular, the developments in Section VI-B).

Convergence of the above abstract algorithm has not yet been considered.

VIII. Conclusions

Perhaps the most striking feature of the above theory is that the main principle, namely, the auxiliary problem principle, has no particular link with the idea of decomposition and with separability assumptions on the MP. The decomposition aspect results from special choices for the core. From this viewpoint, any gradient algorithm *is* a decomposition-coordination algorithm, but the lower level task is quite trivial because of the special form (14). By elaborating the principle and choosing more complicated cores, the lower level task can become more effective, while still being achieved by parallel computations. With regard to the new computers that may arise in the near future, this is probably the most important aspect of the theory, and a promising avenue for future research which must involve both mathematicians and computer scien-

tists. It may open the possibility of on-line computations in applications to automatic control, thanks to the time savings resulting from parallel computations.

However, one must not underestimate the usefulness of these algorithms when implemented on classical computers. Then, the sequential (Gauss–Seidel) decomposition is more adapted. Since the whole problem is broken up into a sequence of nontrivial smaller subproblems, one can solve either big problems with a small computer or huge problems with a big computer.

The question of knowing whether computational savings occur when using Gauss–Seidel decomposition algorithms instead of classical algorithms is irrelevant in the case of huge problems, since we mean by "huge" that these problems cannot be solved even with the biggest computers. In the case of big problems, the savings are questionable since the sequence of subproblems is theoretically infinite. However, it can be effective in the case when the amount of computations increases at an exponential rate with the problem size (e.g., dynamic programming). Moreover, one must take other considerations into account (e.g., numerical stability or "controllability"). Finally, one cannot give a definitive answer to this question of computational savings since, as we have seen, there is no difference of nature between classical and two-level algorithms, but only a difference of degree in the distribution of tasks between the two levels.

The possibility of solving bigger problems with smaller computers is actually the true advantage of sequential decomposition.

References

[1] J. P. Aubin, "Méthode de décentralisation par les subventions," European Inst. for Advanced Studies in Management (Brussels, Belgium), Working Paper 73-17, May 1973.

[2] A. Bensoussan, "Teoria moderna de control optimo," Cuadernos del Instituto de Mathematica Beppo Levi, 1974.

[3] P. Bernhard, *Commande, Optimale, Décentralisation et Jeux dynamiques*. Paris: Dunod, 1975.

[4] C. B. Brosilow, L. S. Lasdon, and J. D. Pearson, "Feasible optimization methods for interconnected systems," in *Proc. JACC*, Troy, NY, 1965.

[5] J. Céa, *Optimisation: Théorie et Algorithmes*. Paris: Dunod, 1971.

[6] G. Cohen, "Une approche unifiée des algorithmes d'optimisation par décomposition-coordination," Centre d'Automatique-Informatique ENSMP, Fontainebleau, France, Int. Rep. E/51, 1976.

[7] ——"On an algorithm of decentralized optimal control," *J. Math. Anal. Appl.*, vol. 59, pp. 242–259, June 1977. Also in Ph.D. thesis, Univ. Paris-Sud Orsay, France, Oct. 1975.

[8] ——"Coordination of constrained optimization problems by resource allocation," Centre d'Automatique-Informatique ENSMP, Fontainebleau, France, Int. Rep. A/49, 1972.

[9] G. Cohen and G. Joalland, "Coordination methods by the prediction principle in large dynamic constrained optimization problems," in *Proc. IFAC Symp. on Large Scale Systems*, Udine, Italy, June 1976.

[10] V. M. Danskin, *The Theory of Max-Min*. Berlin: Springer Verlag, 1967.

[11] G. B. Dantzig and P. Wolfe, "The decomposition algorithm for linear program," *Econometrica*, vol. 29, pp. 767–778, 1961.

[12] I. Ekeland and R. Teman, *Convex Analysis and Variational Problems*. Amsterdam: North Holland–American Elsevier, 1976 (English and expanded edition).

[13] W. Findeisen, "Parametric optimization by primal method in multilevel systems," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, July 1968.

[14] D. Gabay and B. Mercier, "A dual algorithm for the solution of a non-linear variational problem via finite element approximation," *Comp. Math. Appl.*, vol 2, pp. 17–40, 1976.

[15] A. M. Geoffrion, "Elements of large-scale mathematical programming, parts I&II," *Management Sci.*, vol., July 1970.

[16] ——"Primal resource-directive approaches for optimizing non linear-decomposable systems," *Oper. Res.*, pp. 375–403, May–June 1970.

[17] A. A. Goldstein, *Constructive Real Analysis*. New York: Harper and Row, 1967.

[18] L. S. Lasdon and J. D. Schoeffer, "A multilevel technique for optimization," in *Proc. JACC*, Troy, NY, 1965.

[19] J. L. Lions, *Contrôle Optimal de Systèmes Gouvernés par des Équations aux Dérivées Partielles*. Paris: Dunod–Gauthier–Villars, 1968.

[20] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[21] M. D. Mesarovic, D. Macko, and Y. Takahara, *Theory of Multilevel Hierarchical Systems*. New York: Academic, 1970.

[22] J. C. Miellou, "Algorithmes de relaxation chaotique à retards," *Revue RAIRO-AFCET*, Dunod, Paris, vol. R-1, pp. 55–82, 1975.

[23] J. D. Pearson, "Dynamic decomposition techniques," ch. 4 in *Optimization Methods for Large-Scale Systems with Applications*, D. A. Wismer Ed. New York: McGraw-Hill, 1971.

[24] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Contr. Optimiz.*, vol. 14, pp. 877–898, Aug. 1976.

[25] H. Tamura, "Application of duality and decomposition in high-order multistage decision problems," Eng. Dept., Cambridge Univ., Cambridge, England, Rep. CUED/B, Control TR49, 1973.

[26] Y. Takahara, "Multilevel approach to dynamic optimization," Syst. Res. Center, Case Western Reserve University, Cleveland, OH, Rep. SRC-50-C-64-18, 1964.

[27] K. J. Arrow and L. Hurwicz, "Decentralization and computation in resource allocation," in *Essays in Economics and Econometrics*, R. W. Pfouts, Ed.: Univ. North Carolina Press, 1960, pp. 34–104.

[28] A. Bensoussan, J. L. Lions, and R. Teman, "Sur les methodes de decomposition, de décentralisation et de coordination, et application," in *Methodes Numeriques D'Analyse De Systemes*, vol. II, IRIA Rep., 1972, pp. 5–190.

[29] D. Potier, "Algorithm de coordination, application a la gestion d'unités de production interdépendantes," in *Methodes Numeriques D'Analyse De Systemes*, vol. II, IRIA Rep., 1972, pp. 241–387.

Guy Cohen was born in 1947. In 1968 he graduated from Ecole Polytechnique, Paris, France, and he joined the Corps des Ponts et Chaussées (a corps of state engineers specialized in civil engineering and transportation problems).

During the two years (1970–1971) he spent in the corresponding application school, he was mainly involved in operations research applied to the study of transportation systems. In 1971 he joined the Centre d'Automatique et d'Informatique of Ecole des Mines de Paris (located at Fontainebleau) where he is currently an Associate Director. His main interest is in optimal control of large scale systems, both in its theoretical aspects and industrial applications. In 1975 he received the Docteur-Ingénieur degree from the University of Paris-Sud-Orsay.