MULTI-COORDINATION METHODS FOR PARALLEL SOLUTION OF BLOCK-ANGULAR PROGRAMS

By

Golbon Zakeri

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

(Computer Sciences and Mathematics)

at the

UNIVERSITY OF WISCONSIN – MADISON

1995

Abstract

This thesis is concerned with the parallel solution of smooth block-angular programs using multiple coordinators. The research herein extends the three phase method of Schultz and Meyer, who use barrier decomposition methods with complex coordinators which are less suited to parallel computation.

We start by surveying the existing literature for block-angular programs and reviewing barrier function methods and the Schultz-Meyer method. We then present our synchronous multi-coordination schemes and prove their convergence. We tested our algorithms on the Patient Distribution System problems, a class of large-scale real world multicommodity network flow problems, as well as on some randomly generated multicommodity network flow problems. Computational results on the CM-5 parallel supercomputer demonstrated that the method was significantly faster than its Schultz-Meyer predecessor. We also present multiple coordinator asynchronous schemes to solve block-angular programs and prove the convergence of those methods.

Acknowledgements

In the Name of God, the Most Gracious and the Most Merciful

First and foremost I would like to thank the best thing that has ever happened to me: my husband Mark Wilson. His continuous support and witty companionship has been the most enjoyable part of my life.

I would like to thank my grandmother Mrs. Esmat Sarmadi and my parents for all they have done for me throughout my life. My mother Mrs. Shahin Sabetghadam and my uncle Dr. Reza Sabet made my education possible and I am indebted to them forever.

I can not imagine a better job of advising than that done by my advisor Professor Bob Meyer. He has always been the perfect advisor for me and I thank him from the depth of my heart for any success in my career. I would also like to thank my many educators from Roshangar High School, Iowa State University and University of Wisconsin. I would like to specially thank Renato De Leone and Michael Ferris for their stimulating courses and valuable discussions. Professor Frank Forelli was a great friend, teacher and mentor, may he

rest in peace.

I thank Olvi Mangasarian and Richard Brualdi who read my thesis as well as Sigrun Andradottir and John Strikwerda for serving on my committee.

Many thanks to my many friends in Madison who made life more interesting. Special thanks to my good friends, colleagues and officemates Armand Zakarian and Geoff Pritchard. I learned a great deal from our discussions. Also special thanks to Armand for the use of his software NSM.

This research was partially funded by National Science Foundation grants CDA-9024618 and CCR-9306807 and Air Force Office of Scientific Research grant F 496 20-94-1-0036.

Contents

Abstract i

Acknowledgements ii

1 Introduction 1

1.1 Nonlinear Programming Problem 1

1.2 The Block-Angular Program 3

1.3 A Survey of Solution Methods for Block-Angular Programs . . . 4

1.3.1 Resource Directive Decomposition 4

1.3.2 Alternating Directions Method 8

1.3.3 Price Directive Decomposition 9

2 A Review of Barrier Function Methods and Schultz-Meyer Decomposition 16

2.1 General Barrier Functions 16

2.2 Convex Barrier Functions 20

2.3 Schultz-Meyer Decomposition Scheme 22

2.3.1 Relaxed Phase 23
2.3.2 Feasibility Phase 23
2.3.3 Optimality Phase 25
2.3.4 Schultz-Meyer Decomposition of the Barrier Problem . . 26

3 Synchronous Multi-Coordination Schemes 30

3.1 Subproblems and Decoupled Resource Allocation 32
3.2 Single-Variable Multi-Coordination 35
3.3 Group and Block-Plus-Group Multi-Coordination 40
3.4 Approximate Solution of Coordinators and Subproblems 42
3.4.1 Stabilization Algorithm for Single-Variable Coordination 42
3.4.2 Approximate Solutions for the Subproblems 45

4 Computational Results 52

4.1 The Implementation 53
4.1.1 Algorithm and Implementation 53
4.1.2 Parameters 54
4.1.3 The PDS Problems 55
4.2 Analysis of the Results 61
4.2.1 Single-variable multi-coordination 61
4.2.2 Group multi-coordination 62
4.2.3 Comparison 63
4.3 The MNETGEN problems 65

5 Asynchronous Decomposition Methods **67**

5.1 Discription of the Method 67

5.2 Single-Variable Coordination in the Asynchronous Case 69

5.3 Asynchronous Group Multi-Coordination 76

5.3.1 Description of the Algorithm 76

5.3.2 Lemmas 77

5.3.3 Convergence Theorem 82

6 Future Directions **87**

Bibliography **89**

Chapter 1

Introduction

In this chapter we start by describing the general nonlinear programming problem, then we introduce the block-angular problem and give some background information about it. We then discuss recent methods as well as classical methods that are used to solve this problem.

1.1 Nonlinear Programming Problem

The problem we consider in this thesis falls under the general category of smooth (at least once continuously differentiable) nonlinear programming and is denoted by (NLP):

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & g(x) \leq \mathbf{0} \\ & h(x) = \mathbf{0} \end{aligned}$$

where we have:

$$f : \Re^N \rightarrow \Re$$

$$g : \Re^N \rightarrow \Re^J$$

$$h : \Re^N \rightarrow \Re^M$$

The Lagrangian for NLP is defined as:

$$\mathcal{L}(x, w, v) = f(x) - wg(x) - vh(x)$$

where $w \in \Re_{-}^{1\times J}$ and $v \in \Re^{1\times M}$ are called the multipliers (Lagrange multipliers).

A KKT point for NLP is a tuple (x, w, v) that satisfies the Karush-Kuhn-Tucker conditions [11]:

$$\begin{aligned} & g(x) \leq \mathbf{0} && h(x) = \mathbf{0} \\ & w \leq \mathbf{0} && \nabla_x \mathcal{L}(x, w, v) = \mathbf{0} \\ & wg(x) = 0 \end{aligned}$$

A point x for which there exists (w, v) satisfying KKT conditions is also referred to as a KKT point for NLP. It is well known that if x is a local minimizer of NLP and some constraint qualifications hold then there must exist multipliers (w, v) such that (x, w, v) satisfy the KKT conditions. The algorithms developed in this thesis are guaranteed to converge to KKT points of the special NLP, the block-angular program we introduce in next section, that we consider.

When f and g are convex functions and h is affine then NLP is a convex program.

1.2 The Block-Angular Program

Suppose that the variables $x \in \Re^N$ can be partitioned into K disjoint blocks such that $x_{[k]} \in \Re^{N_{[k]}}$ and $N = \sum_k N_{[k]}$ (so x can be thought of as the concatenation of $x_{[k]}, \quad k = 1, \ldots, K$). Then the problem:

$$\begin{array}{lllll} \min_x & c(x) & & & \\ \text{s/t} & A_{[1]}x_{[1]} & & = & b_{[1]} \\ & & A_{[2]}x_{[2]} & = & b_{[2]} \\ & & \ddots & & \\ & & A_{[K]}x_{[K]} & = & b_{[K]} \\ & & D(x) & \leq & d \\ & \mathbf{0} \leq x \leq u & & & \end{array}$$

is called a **block-angular** programming problem, denoted (BAP). The theory can be extended to non-zero lower bounds on x, however we assume $\mathbf{0}$ lower bounds for convenience. We will sometimes refer to the block constraints as $Ax = b$ where A is the direct sum of the matrices $A_{[k]}$, x is the concatenation of the blocks $x_{[k]}$ and b is the concatenation of $b_{[k]}$'s. The upper bounds u are assumed to be finite. If the function D is separable (e.g. the BAP is linear) then we can rewrite the coupling constraints as:

$$D_{[1]}(x_{[1]}) + \cdots + D_{[K]}(x_{[k]}) \leq d$$

and if D is linear then:

$$D_{[1]}x_{[1]} + \cdots + D_{[K]}x_{[k]} \leq d$$

Also if c is separable (e.g. the BAP is linear) then we can rewrite the objective as:

$$c_{[1]}(x_{[1]}) + \cdots + c_{[K]}(x_{[K]})$$

and if c is linear then:

$$cx = c_{[1]}x_{[1]} + \cdots + c_{[K]}x_{[K]}$$

Problems with block-angular structure come up very often in large-scale optimization and have important practical applications. Two well-known types of BAP are multicommodity network flow problems and problems arising from multi-period financial modeling. In the case of the multicommodity network flow problems the matrices $A_{[k]}$ are node-arc incidence matrices for the underlying network structure. For more information on network flow problems see [8] and for more on multicommodity network problems see [12].

1.3 A Survey of Solution Methods for Block-Angular Programs

1.3.1 Resource Directive Decomposition

The BAP may be viewed as a problem of optimally allocating the scarce resource d to the blocks. Resource directed decomposition methods solve this problem by choosing a feasible allocation, testing it for optimality and then improving upon it if possible. This is done in an iterative, fork-join manner. At each

iteration, during the fork stage, subproblems of the following form (for block k) are considered:

$$w_{[k]}(y_{[k]}) := \min \left\{ c_{[k]}(x_{[k]}) | \quad A_{[k]}x_{[k]} = a_{[k]} \text{ and } D_{[k]}(x_{[k]}) \leq y_{[k]} \right\} \tag{1}$$

where $y_{[k]}$ is the resource allocation for block k. Now let $V_{[k]}$ be the set of all resource allocations $y_{[k]}$ for which subproblem k is feasible. Then the master problem is:

$$\begin{aligned} \min_{y} \quad & \textstyle\sum_{k=1}^{K} w_{[k]}(y_{[k]}) \\ \text{subject to} \quad & y_{[k]} \in V_{[k]} \quad (\forall k = 1, \cdots, K) \\ & \textstyle\sum_{k=1}^{K} y_{[k]} \leq d \end{aligned} \tag{2}$$

Note that the objective of the master problem is usually nondifferentiable.

Bundle Methods

De Leone et al. [4] apply a resource directed decomposition method to multicommodity network flow problems where c and D are linear. They use the bundle method to solve the nondifferentiable master problem (2). They do not solve the subproblems (1) to optimality; rather, they terminate with an approximately optimal solution. They start with $y_{[k]}$ that have the property $\sum_{k=1}^{K} y_{[k]} = d$. Let $x^k(y_{[k]})$ be any feasible solution to the subproblem for block k and χ_k the corresponding (LP) dual objective. Also λ_k are the duals associated with the constraint $D_{[k]}x_{[k]} \leq y_{[k]}$ and ϵ_k is the associated optimality gap (that is the difference between the primal and dual objective values: $\epsilon_k = c_{[k]}x_{[k]} - \chi_k$). Then

λ_k is an ϵ_k-subgradient of $w_{[k]}(y_{[k]})$. Now if $\epsilon = \sum_{k=1}^{K} \epsilon_k$ then $z = \sum_{k=1}^{K} c_{[k]} x_{[k]}$ differs not more than ϵ from the optimal objective of the master problem (2). Also $\Lambda = (\lambda^1, \cdots, \lambda^K)$ is an ϵ-subgradient for the objective of the master.

Their algorithm then proceeds as follows:

The current point is y^j and the old information triplet (y^i, z^i, Λ^i) is available for $i \in I$. Also define for each $i \in I$

$$\xi_i = 2\epsilon_i + z^j - z^i - \Lambda^i(y^j - y^i).$$

Then the following problem is solved to obtain a descent direction to improve the current iterate (t here is a positive scalar):

$$\begin{aligned} \min_{v,s} \quad & v + \tfrac{1}{2t}||s||^2 \\ \text{subject to} \quad & v \geq \Lambda^j s - \epsilon_j \\ & v \geq \Lambda^i s - \xi_i \quad (\forall i \in I) \end{aligned} \tag{3}$$

Problem (3) corresponds to finding the minimum of a piecewise linear approximation to the objective of the master (2). This approximation is obtained by taking the maximum of a finite number of affine functions which support the graph of the objective function of (2) from below. Once the optimal solution to (3) (say (v_j, s_j)) is found then s_j is used as a correction. If certain descent conditions are met then $\hat{y} = y^j + s_j$ is used as a new iterate, otherwise $\hat{y}$ is added to the bundle and the current iterate remains unchanged. The termination condition is of the type $|v_j| \leq \delta$.

Medhi [16] also uses bundle methods to solve BAP. However, he applies the

bundle method to the dual (Rockafellar dual [19]) of BAP which is a nonsmooth, concave problem. He solves the subproblems:

$$\partial f_k^*(D_k y) = \text{argmin}_{x_k} \{f_k(x_k) - < D_k y, x_k >\}$$

where

$$f_k(x_k) = \begin{cases} c_{[k]} x_{[k]} & \text{if } A_k x_{[k]} = b_k \\ +\infty & \text{otherwise} \end{cases}$$

for $k = 1, \cdots, K$ in parallel. He then uses the solutions of the subproblems to calculate a subgradient for the objective of the dual (Rockafellar dual). He also uses the optimal solution to the subproblems to find dual subgradients. Then a quadratic master problem is solved to obtain an ascent direction from which a new point with improved objective may be constructed. The following is the quadratic master problem's formulation:

$$\begin{array}{lll} \min_\lambda & \frac{1}{2}||\sum_{j=1}^{l} \lambda_j \pi^j||^2 & \\ & \sum_{j=1}^{l} \lambda_j & = \quad 1 \\ & \sum_{j=1}^{k} \alpha_{jl} \lambda_j & \leq \quad \epsilon_l \\ & \lambda_j & \geq \quad 0 \quad j = 1, \ldots, l \end{array}$$

In the above l is the current number of iterates, π^js are dual subgradients for dual iterates y^js, and $\alpha_{jl} = g(y^j) - g(y^l) + < \pi^J, y^l - y^j >$ where g is the dual objective. The algorithm continues in a loop in this fashion until the termination criteria are met.

1.3.2 Alternating Directions Method

De Leone, Meyer and Kontogiorgis in [5] use the "Alternating Directions" method to solve the BAP. They introduce for each block a vector of extra variables $\tilde{d}_{[k]}$ that is an upper bound for $D_{[k]}x_{[k]}$; $\tilde{d}_{[k]}$ indicates an allocation of scarce resources for block k (this is the resource directive component of their method). They define:

$$\mathbf{h}_{[k]}(x_{[k]}) = \begin{cases} c_{[k]}(x_{[k]}) & \text{if } x_{[k]} \text{ satisfies the network constraints of block } k \\ +\infty & \text{otherwise} \end{cases}$$

from which they form closed, proper, convex, extended real valued functions:

$$G_1(x_{[1]}, \cdots, x_{[K]}, \tilde{d}_{[1]}, \cdots, \tilde{d}_{[K]}) = \begin{cases} \sum_{k=1}^{K} \mathbf{h}_{[k]}(x_{[k]}) & \text{if } D_{[k]}x_{[k]} \leq \tilde{d}_{[k]} \quad (\forall k) \\ +\infty & \text{otherwise} \end{cases}$$

and

$$G_2(y_{[1]}, \cdots, y_{[K]}, d_{[1]}, \cdots, d_{[K]}) = \begin{cases} 0 & \text{if } \sum_{k=1}^{K} d_{[k]} = d \\ +\infty & \text{otherwise} \end{cases}$$

The above yields an equivalent form for the BAP:

$$\begin{aligned} \min_{x_{[k]}, \tilde{d}_{[k]}, y_{[k]}, d_{[k]}} \quad & G_1(x_{[1]}, \cdots, x_{[K]}, \tilde{d}_{[1]}, \cdots, \tilde{d}_{[K]}) + G_2(y_{[1]}, \cdots, y_{[K]}, d_{[1]}, \cdots, d_{[K]}) \\ \text{subject to} \quad & x_{[k]} = y_{[k]} \quad (\forall k = 1, \cdots, K) \\ & \tilde{d}_{[k]} = d_{[k]} \quad (\forall k = 1, \cdots, K) \end{aligned}$$

Now in order to solve the linearly-constrained convex problem:

$$\begin{aligned} \min_{w,z} \quad & F_1(w) + F_2(z) \\ \text{subject to} \quad & Aw + b = Bz \end{aligned}$$

one can find the saddle points of the augmented Lagrangian given below (multipliers p represent the price directive element of this approach).

$$L_\lambda(w,z,p) := F_1(w) + F_2(z) + p(Aw + b - Bz) + \frac{\lambda}{2}||Aw + b - Bz||_2^2$$

De Leone, Meyer and Kontogiorgis do this using the alternating directions method in a block Gauss-Seidel fashion:

$$w^{t+1} \in \text{argmin} L_\lambda(w, z^t, p^t) \quad (4)$$

$$z^{t+1} \in \text{argmin} L_\lambda(w^{t+1}, z, p^t) \quad (5)$$

$$p^{t+1} = \lambda \nabla_p L_\lambda(w^{t+1}, z^{t+1}, p^t) \quad (6)$$

They allow for different penalty factors $\lambda_{[k]}$ for each block and also let these factors vary with time. In the case of the reformulated BAP, step (4) amounts to solving a quadratic multicommodity network flow problem which is separable over blocks (hence this is done in parallel). Step (5) translates to adjusting the allocations to achieve feasibility and step (6) is updating the multipliers. They also investigate two other variants of the above scheme.

1.3.3 Price Directive Decomposition

In price directed decomposition methods the coupling constraints of BAP are dealt with using some pricing mechanism. Examples of such pricing mechanisms are putting the binding constraints in the objective via Lagrangian relaxation, in the form of a barrier function or using other penalty terms.

In general, subproblems in the price directive decomposition scheme amount to minimization of a linear functional subject to the original block constraints. Subproblems are solved to gain search information used in the master and they provide the master with primal information.

We will discuss a few price directed decomposition schemes below-the most famous of which is perhaps Dantzig-Wolfe. The last such example discussed here is barrier decomposition methods on which the work of this thesis is based.

Dantzig-Wolfe Decomposition

The Dantzig-Wolfe decomposition method [3] solves the BAP when it is a linear program. Suppose the coupling constraints are put into the objective in the form of the Lagrangian relaxation:

$$\begin{array}{lll} \min & cx - \pi(Dx - d) & \equiv \quad \min \quad (c_{[1]} - \pi D_{[1]})x_{[1]} + \cdots + (c_{[K]} - \pi D_{[K]})x_{[K]} \\ \text{s/t} & \text{Block constraints} & \text{s/t} \quad \text{Block constraints} \end{array}$$

If the vector π is chosen as the vector of optimal dual variables then the objective value of the above is the same as the optimal objective for the original problem.

Let $S_k = \left\{x_{[k]} | A_{[k]}x_{[k]} = b_{[k]} \quad 0 \leq x_{[k]} \leq u_{[k]}\right\}$; then S_k is a compact set and we have:

$$x_{[k]} \in S_k \Leftrightarrow (x_{[k]} = \lambda_k \mathbf{X}_k \quad \lambda_k \geq 0, \quad e\lambda_k = 1)$$

where $\mathbf{X}_k$ is the matrix whose columns are the vertices of S_k. Then the original

BAP may be written as:

$$\begin{array}{ll} \min_\lambda & c_{[1]}\mathbf{X}_1\lambda_1 + \cdots + c_{[K]}\mathbf{X}_K\lambda_K \\ \text{s/t} & D_{[1]}\mathbf{X}_1\lambda_1 + \cdots + D_{[K]}\mathbf{X}_K\lambda_K \leq d \\ & \lambda_k \geq 0, \qquad e\lambda_k = 1 \quad (\forall k = 1, \ldots, K) \end{array}$$

Now, at the first iteration, for each block k the subproblem

$$\begin{array}{ll} \min & \tilde{c}_{[k]}x_{[k]} \\ \text{s/t} & A_{[k]}x_{[k]} = b_{[k]} \\ & 0 \leq x_{[k]} \leq u_{[k]} \end{array}$$

is solved in order to generate a single column of $\mathbf{X}_k$ (the cost $\tilde{c}$ is arbitrary at this point). Subsequently the restricted master

$$\begin{array}{ll} \min & (c_{[1]}\tilde{\mathbf{X}}_1)\lambda_1 + \cdots + (c_{[K]}\tilde{\mathbf{X}}_K)\lambda_K \\ \text{s/t} & D_{[1]}\tilde{\mathbf{X}}_1\lambda_1 + \cdots + D_{[K]}\tilde{\mathbf{X}}_1\lambda_K \leq d \\ & \lambda \geq 0, \qquad e\lambda_k = 1 \quad (\forall k = 1, \ldots, K) \end{array}$$

is solved where $\tilde{\mathbf{X}}_k$ is an approximation (in iteration one this is just the one column obtained so far,) of $\mathbf{X}_k$. The solution of the restricted master yields a set of duals $\tilde{\pi}_i$ corresponding to the linear inequality constraints and duals $\tilde{\sigma}_k$ to the constraints $e\lambda_k = 1$. After the initial iteration the subproblems take the form:

$$\begin{array}{ll} \min & (c_{[k]} - \tilde{\pi}D_{[k]})x_k \\ \text{s/t} & x_{[k]} \in S_k \end{array}$$

Now if the optimal objective to the subproblem (of the kth block) is less than $\tilde{\sigma}_k$ then a new column of $\mathbf{X}_k$ is generated which results in a new $\tilde{\mathbf{X}}_k$. If all the

subproblems have nonnegative optimal objectives then the current solution of the restricted master is an optimal solution for the the original BAP. For a more detailed discussion of the Dantzig-Wolfe decomposition method see [14].

Simplicial Decomposition

This method originally developed by Von Hohenbalken [22] considers the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s/t} \quad & x \in X \end{aligned}$$

where X is a compact, convex, nonempty set and f is differentiable. So in the context of BAP the set X is replaced by $\mathcal{B} = \{x| \quad Ax = b, \qquad 0 \leq x \leq u\}$ and the function f has "absorbed" the coupling constraints (e.g. in the form of a penalty function). The problem is decomposed into master and subproblem which are solved alternatively in a loop. At iteration t, first the subproblem:

$$\min_{y \in X} y \nabla f(x^t)$$

is solved where x^0 is arbitrary and x^t is the solution of the master from previous round if $t > 0$. For the case of $X = \mathcal{B}$ each subproblem is a linear programming problem decomposable over the blocks and the solution y^t is an extreme point of $\mathcal{B}$.

Let $\mathcal{Y} = [y^1|y^2|\ldots|y^t]$ then the master problem is:

$$\min_{w \in W} f(\mathcal{Y}w)$$

$$\text{where} \quad W = \left\{ w | \quad \sum_{i=1}^{t} w_i = 1, \quad w_i \geq 0 \quad (\forall i = 1, \ldots t \right\}$$

and $x^{t+1} = \mathcal{Y} w^*$ where w^* is the optimal solution to the master problem. Simplicial decomposition is a decomposition method based on Carathéodory's theorem which makes this method applicable for any compact, convex, nonempty set X. A modification of simplicial decomposition is the restricted simplicial decomposition in which there are only a fixed (predetermined) number of columns of $\mathcal{Y}$ are stored. Restricted simplicial decomposition uses less memory than the ordinary simplicial decomposition.

Smoothed Exact Penalty

Zenios and Pinar [18] introduce the following linear quadratic penalty function in the solution of BAP:

$$\tilde{p}(\epsilon, t) = \begin{cases} 0 & \text{if } t \leq 0 \\ \mu \frac{t^2}{2\epsilon} & \text{if } 0 \leq t \leq \epsilon \\ \mu(t - \frac{\epsilon}{2}) & \text{if } t \geq \epsilon \end{cases}$$

As the parameter $\epsilon \downarrow 0, \tilde{p}(\epsilon, t)$ approaches the piecewise linear penalty function

$$\tilde{p} = \mu \max \{0, t\} .$$

The linear quadratic penalty function is once continuously differentiable (for $\epsilon > 0$). Zenios and Pinar use the linear quadratic penalty function to put the mutual constraints in the objective and this yields:

$$\min_{x \in X} \Phi_{\mu,\epsilon}(x) = f(x) + \sum_{j=1}^{M} \tilde{p}(\epsilon, y_j) \tag{7}$$

where $y_j = D_j x - d_j$. They then solve the BAP using an iterative method. Initially a relaxed problem (the original BAP without the mutual constraints) is solved and they use the optimal solution $\hat{x}$ to set up $y = D\hat{x} - d$. At the start of each iteration they approximately solve the smooth penalty problem (7). This is done using linearization via simplicial decomposition which makes the process parallelizable. Once the approximate solution of (7) is determined they adjust the penalty parameters. The steps are repeated in a loop until the termination criteria is met. Mangasarian and Chen have since developed a class of smoothing functions one of which is the smoothed exact penalty function [2].

Augmented Lagrangian

Meyer and Zakarian [24] solve the BAP using an augmented Lagrangian approach. They put the binding constraints in the objective and obtain:

$$L_\lambda(x, p) = \sum_{k=1}^{K} c_k(x_{[k]}) + p(Dx - d) + \frac{\lambda}{2}||\hat{D}x - \hat{d}||_2^2$$

where $\hat{D}$, and $\hat{d}$ are subsets of the rows of D and d, corresponding to the current active set. The active set here refers to the constraints with positive multipliers, or ones that are satisfied at equality or the ones that are violated. The augmented Lagrangian algorithm then consists of minimizing $L_\lambda(x, p^t)$ over x subject to the linear equality and bound constraints, updating the multipliers, and updating the active set. These three steps are done in a loop until optimality is achieved. Meyer and Zakarian use nonlinear Jacobi algorithm with both simple and complex coordination for the first step of the augmented Lagrangian

algorithm. This approach allows use of parallelism on the time consuming portion of the algorithm (step 1 of the augmented Lagrangian algorithm) in an efficient manner. This is an application of the parallel variable distribution method developed by Ferris and Mangasarian [6] to the augmented Lagrangian formulation of the BAP.

Barrier Function Methods

We dedicate the next chapter to explanation of barrier function methods and Schultz-Meyer decomposition scheme for solution of barrier problems. The method developed here is a decomposition method related to Schultz-Meyer which uses parallel coordination. Our synchronous methods are presented in chapter 3. We present the numerical results for synchronous results in chapter 4 and asynchronous methods are discussed in chapter 5.

Chapter 2

A Review of Barrier Function Methods and Schultz-Meyer Decomposition

In this chapter we introduce the barrier function method and review some classical results for barrier functions. We then present a compact version of the Schultz-Meyer decomposition scheme for the solution of BAP which uses barrier functions.

2.1 General Barrier Functions

If the mutual constraints were not present in the formulation of BAP the constraints would be separable over the blocks (if we further had separability of the

objective function $c(x)$ then the problem would be completely separable). In order to remove the coupling constraints we may place them in the objective in the form of a barrier function. Barrier functions were first introduced by Frisch [9] and later analyzed in great detail by Fiacco and McCormick [7].

Definition 1 ***(Barrier function)*** *Let* $\rho : \Re^J \to \Re$ *be defined by:*

$$\rho = \begin{cases} \sum_{j=1}^{J} \rho_j(s_j) & \text{if } s > 0 \\ +\infty & \text{otherwise} \end{cases}$$

then ρ *is said to be a barrier function if and only if the three properties below hold for all* $j = 1, \ldots, J$.

- $\rho_j : \Re_{>0} \to \Re$ *is continuous.*
- $\lim_{\xi \downarrow 0} \rho_j(\xi) = +\infty$.
- $(\forall 0 < \xi < \zeta) \quad \rho_j(\xi) \geq \rho_j(\zeta)$, *that is* ρ_j *is decreasing.*

Note that ρ in effect introduces a barrier that separates $s > 0$ from other points $s \in \Re^J$. Now the BAP can be transformed to the barrier problem, denoted (BP), of the form:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c(x) + \tau\rho(d - D(x)) \\ \text{subject to} \quad & A_{[k]}x_{[k]} = b_{[k]} \qquad k = 1, \ldots, K \\ & \mathbf{0} \leq x \leq u \end{aligned} \tag{8}$$

In general, given the constrained problem:

$$\underset{x}{\text{minimize}} \qquad c(x) \tag{9}$$

$$\text{subject to} \quad D(x) \leq d$$

where there exists some point z with $D(z) < d$, the classical barrier function method would solve a sequence of unconstrained problems:

$$\min_x f_{\tau^i}(x) = c(x) + \tau^i \rho(d - D(x))$$

The hope is that as $\tau^i \downarrow 0$ the sequence of minimizers $\{x^i\}$ would converge to a minimizer of the original constrained problem. This hope in fact materializes through the following result obtained by Fiacco and McCormick (theorem 8 of [7]) for general constrained problems whose sets of local minima satisfy some specified properties. Before we present the theorem we have to make some definitions and introduce the concept of compact perturbation superset for a set of local minimizers.

Definition 2 ***(Isolated Set)*** *(from [7]) A set $X \subset Y$ is an isolated (sub)set of Y if there exists some set E such that $X \subset E^\circ$ and $\emptyset = (E \backslash X) \cap Y$.*

Theorem 1 ***(Existence of Compact Perturbation Set)*** *Suppose all functions f and g defining the generic nonlinear program:*

$$\min_x f(x) \quad \textit{subject to} \quad g(x) \leq \mathbf{0}$$

are continuous. If a set of local minima corresponding to a function value v^* is a nonempty, isolated, compact subset of

$$Y = \{g(y) \leq \mathbf{0} | f(y) = v^*\}$$

then there exists a compact set S such that $X^* \subset S^\circ$ and for any feasible $y \in S \backslash X^*, f(y) > v^*$.

The above result on the existence of compact perturbation set is due Fiacco and McCormick (theorem 7 of [7]). We now proceed with the main theorem:

Theorem 2 *Suppose that*

1. *c and D are continuous,*
2. *there exists some point z with $D(z) < d$,*
3. *the set of local minima of the constrained problem (9) X^* corresponding to the value $v^* = c(X^*)$ is a nonempty compact set,*
4. *X^* is isolated in*

$$Y = \{y | D(y) \leq d \quad and \quad c(y) = v^*\}$$

5. *at least one point in X^* is in the closure of $\{x | D(x) < d\}$, and*
6. *$\tau^i \downarrow 0$ monotonically.*

Then

1. *there exists a compact set S such that $X^* \subset S^\circ$ and for τ^i small enough the unconstrained minima of $f_{\tau^i}(.)$ in $\{x | D(x) < d\} \cap S^\circ$ exist and every limit point of any subsequence $\{x^i\}$ of such minimizers is in X^*,*
2. *$\lim_{i \to \infty} c(x^i) = v^*$,*

3. $\lim_{i\to\infty} \tau^i \rho(d - D(x^i)) = 0$,

4. $\lim_{i\to\infty} f_{\tau^i}(x^i) = v^*$,

5. $\{c(x^i)\}$ *is a monotonically decreasing sequence,*

6. $\{\rho(d - D(x^i))\}$ *is a monotonically increasing sequence.*

2.2 Convex Barrier Functions

If the functions c, D and ρ are convex then the function f described above would be convex. When convexity is assumed some of the assumptions for theorem (2) are unnecessary. We will assume convexity for our results.

A well known and widely used barrier function is the logarithmic barrier function given by:

$$\rho(d - D(x)) = -\sum_{j=1}^{J} \ln(d_j - D_{j.}(x))$$

The log barrier function has the very nice property of ϵ-optimality stated below:

Theorem 3 ***(ϵ-optimality of Logarithmic Barrier)*** *Suppose x^* is the optimal solution of the convex constrained problem (9) and x is the optimal solution of the logarithmic barrier problem formed from (9). Then*

$$c(x^*) \leq c(x) \leq c(x^*) + \epsilon$$

where $\epsilon = \tau.J$ (recall that J is the number of constraints in (9) and τ is the penalty parameter).

Schultz in his thesis presents a proof of this theorem in conjunction with other relevant theorems. Here we present a slightly shorter proof.

Proof First let us write the dual to problem (9):

$$\begin{aligned} \underset{\lambda,x}{\text{maximize}} \quad & c(x) + \lambda'(D(x) - d) \\ \text{subject to} \quad & c'(x) + \lambda' \nabla D(x) = \mathbf{0} \\ & \lambda \geq \mathbf{0} \end{aligned} \tag{10}$$

Now suppose $\overline{x}$ is feasible for (9) and $(\hat{x}, \hat{\lambda})$ is feasible for the (10) problem. Then we have:

$$\begin{aligned} c(\hat{x}) + \hat{\lambda}'(D(\hat{x}) - d) &= c(\hat{x}) + \hat{\lambda}'(D(\hat{x}) - d) + \left[c'(\hat{x}) + \hat{\lambda}' \nabla D(\hat{x})\right].(\overline{x} - \hat{x}) \\ &= [c(\hat{x}) + c'(\hat{x})(\overline{x} - \hat{x})] + \\ &\qquad < \hat{\lambda}, (D(\hat{x}) - d + \nabla D(\hat{x}).(\overline{x} - \hat{x})) > \\ &\leq c(\overline{x}) + \hat{\lambda}'(D(\overline{x}) - d) \\ &\leq c(\overline{x}) \end{aligned} \tag{11}$$

The first equality is due to feasibility of $(\hat{x}, \hat{\lambda})$ for (10), and the second equality is obtained by merely rearranging the terms. The first inequality is obtained using convexity of c and D and the fact that the gradient of a convex function underestimates it. The final inequality is due to $\overline{x}$ feasible for (9) and $\hat{\lambda} \geq \mathbf{0}$.

So now let $\hat{x} = x$ where x is the optimal solution of the log barrier problem. Also define $\hat{\lambda}$ componentwise by:

$$(\hat{\lambda})_j = \frac{\tau}{d_j - D_{j.}(x)}$$

Then since x is the optimal solution of the barrier problem we have that

$$c'(\hat{x}) + \hat{\lambda}'\nabla D(\hat{x}) = 0$$

Hence by (11) and by definition of $\hat{\lambda}$ and $\hat{x}$ we get:

$$c(x) - \tau.J \leq c(x^*)$$

So by feasibility of x for (9) and the above we obtain:

$$c(x^*) \leq c(x) \leq c(x^*) + \epsilon \tag{12}$$

2.3 Schultz-Meyer Decomposition Scheme

Schultz and Meyer [20] use the barrier function techniques to transform the BAP to the BP. They then apply a three phase decomposition scheme to solve the BP. In order to solve the BP one needs to start with a point that satisfies the mutual constraints (as well as block and bound constraints). The starting point needs to satisfy the mutual constraints strictly. To obtain such a point they go through the feasibility phase. Once a feasible point is obtained they move toward optimality while maintaining feasibility. We will now proceed with the description of the Schultz-Meyer three-phase method on which our methods are based.

2.3.1 Relaxed Phase

The aim of this phase is to come up with an initial point. At this stage the relaxed problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c(x) \\ \text{subject to} \quad & A_{[k]}x_{[k]} = b_{[k]} \quad k = 1, \ldots, K \\ & \mathbf{0} \leq x \leq u \end{aligned}$$

is solved. If the objective function $c(x)$ is separable over the blocks (which in practice is often the case) this step can be done in parallel. That is, each processor $i \quad (i = 1, \ldots, K)$ would be responsible for solving the following problem:

$$\begin{aligned} \min_{x_{[i]}} \quad & c_{[i]}(x_{[i]}) \\ \text{subject to} \quad & A_{[i]}x_{[i]} = b_{[i]} \\ & \mathbf{0} \leq x_{[i]} \leq u_{[i]} \end{aligned}$$

The optimal solution of the relaxed phase is then the concatenation of the optimal $x_{[i]}$'s produced by each processor and the optimal objective the sum of the optimal objectives of the processors.

2.3.2 Feasibility Phase

At the start of this phase there is a point x^0 available (obtained in the relaxed phase) which satisfies the block and the bound constraints. The aim here is to produce $\hat{x}$ which satisfies the block and bound constraints and also satisfies the

coupling constraints strictly (that is $D(\hat{x}) < d$). Hence if x^0 happens to satisfy the coupling constraints strictly then we go on to the next phase. Otherwise the method of shifted barriers is applied to obtain $\hat{x}$:

Definition 3 ***(Shifted Barrier Problem)*** *We will denote the following problem (SBP):*

$$\mathcal{P}(\tau, \theta) = \left\{ \min_x \quad f_\tau(x, \theta) = c(x) + \tau\rho(\theta - D(x)) \quad s/t \quad x \in \mathcal{B} \right\}$$

(recall that $\mathcal{B} = \{x | \quad Ax = b, \quad \mathbf{0} \leq x \leq u\}$.) Schultz and Meyer solve a sequence of (SBP) in which the shifted barrier $\{\theta^i\}$ tends to the original barrier d. Let x^i be the approximate minimizer of $\mathcal{P}(\tau, \theta^i)$, then they choose θ^{i+1} such that the following hold:

$$D(x^i) < \theta^{i+1} \leq \theta^i \quad \text{and} \quad \theta^{i+1} \geq d \tag{13}$$

$$\text{and in the limit either } \theta^\infty = d \text{ or } (\exists j) \quad \text{ such that } \liminf_i \{\theta^i_j - D_{j.}(x^i)\} = 0 \tag{14}$$

They show (in theorem 2.7 of [20]) that one can obtain a strictly feasible point $\hat{x}$ using shifted barrier methods provided the original BAP has a strictly feasible point.

Theorem 4 ***(Finite Feasibility)*** *Suppose $c(.)$ is bounded from below on $\mathcal{B}$. Also let θ^{i+1} be chosen to satisfy (13) and (14). If $x^i \in \mathcal{B}$ are computed so that $\sup_i f_\tau(x^i, \theta^i)$ is finite, then a point $z \in \mathcal{B} \cap \mathcal{C}^\circ$ exists and will be found in a finite number of steps. On the other hand if no such point exists then $f_\tau(x^i, \theta^i) \to +\infty$.*

They propose the following algorithm for choosing θ^i which they prove (in theorem 2.9 of [20]) satisfies the conditions (13) and (14):

$$\theta_j^1 = \begin{cases} d_j & \text{if } D_{j.}(x^0) < d_j \\ D_{j.}(x^0) + \Theta & \text{if } D_{j.}(x^0) \geq d_j \end{cases} \tag{15}$$

where $\Theta > 0$ is a constant. After computing x^i set:

$$\theta_j^i = \begin{cases} d_j & \text{if } D_{j.}(x^i) < d_j \\ \lambda_\theta D_{j.}(x^i) + (1 - \lambda_\theta)\theta_j^i & \text{if } D_{j.}(x^0) \geq d_j \end{cases} \tag{16}$$

The parameter $\lambda_\theta \in (0, 1)$ is prechosen and fixed. Note here that an efficient algorithm for solving the shifted barrier problem (SBP) plays a key role in finding a feasible point.

2.3.3 Optimality Phase

We reach this phase if $\hat{x} = x^0$ (found in the relaxed phase) satisfies the coupling constraints strictly or if we've gone through the feasibility phase and have obtained a point $\hat{x}$ that not only satisfies the network and bound constraints but also satisfies the coupling constraints strictly. Schultz in his doctoral thesis refers to this stage of the algorithm as the "Refined Phase". Here the initial point $\hat{x}$ is "refined" through an iterative process and the algorithm terminates with an approximately optimal point x^*. Essentially the algorithm will proceed as follows:

Repeat until suitable x^i is achieved:
generate x^i, the approximate solution of $\mathcal{P}(\tau^i, d)$,
set $\tau^{i+1} = \max(\tau_{\text{inf}}, \lambda_\tau \tau^i)$

τ_{inf} and λ_τ are prechosen factors where $\lambda_\tau \in (0,1)$ and τ_{inf} is very small (e.g. 10^{-6}).

As observed from the above the major step in both feasibility and optimality phase is to approximately solve the barrier problem $\mathcal{P}$. This is the central point of Schultz-Meyer decomposition algorithm and also of this thesis. We will proceed by presenting decomposition scheme and the relevant theorems, discuss the improvements relative to Schultz-Meyer and present our methods (in chapters 3–5) which incorporate these improvements.

2.3.4 Schultz-Meyer Decomposition of the Barrier Problem

In order to generate an approximate solution of $\mathcal{P}(\tau, \theta)$, Schultz and Meyer start with an initial feasible point x (e.g. x^0 in the first iteration of the feasibility phase or $\hat{x}$ at the start of the optimality phase). At each iteration they first find search directions by solving subproblems and then use this information in a coordination step to come up with a new iterate with a better objective value.

Convex Quadratic Subproblems

At each iteration t, Schultz and Meyer use subproblems of the form:

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & g^t(y - x^t) + (y - x^t)G^t(y - x^t) \qquad (17) \\ \text{subject to} \quad & Ay = a \text{ and } 0 \leq y \leq \overline{\mathcal{R}}(x^t) \end{aligned}$$

$$(18)$$

where g^t satisfies:

$$\lim_{t \to \infty} ||g^t - \nabla f(x^t)|| = 0$$

and the symmetric matrices G^t satisfy:

$$(\exists \quad \beta > 0) \quad (\forall \quad t, x) \qquad 0 \leq x' G^t x \leq \beta ||x||^2$$

Note here that if $G^t = \mathbf{0}$ then the above quadratic subproblem turns into a linear subproblem.

The subproblem presented above is the minimization of a second (or in special cases first) order approximation of the barrier problem subject to the network and bound constraints. The upper bound $\overline{\mathcal{R}}(x^t)$ acts as a type of trust region to eliminate taking very short steps (this is small w variables or large values of y variables). We tried the algorithm without using the decoupled resource allocation and we got very poor computational results in that the convergence was much slower. Ordinarily there would be an equivalent of $\overline{\mathcal{R}}(x^t)$ for the lower bounds on the variables, however since the matrix of coupling constraints is non-negative the lower bounds remain at zero for the decoupled

resource allocation (see [20]).

If the matrix G^t is chosen appropriately (e.g. taking G^t to be a block diagonal approximation of the hessian of $f(x^t)$ or $G^t = \mathbf{0}$) then (17) is decomposable over the blocks. Hence the subproblem can be solved in parallel.

Coordinator Problem

Once the subproblem is solved and search directions $y^t_{[k]}$ for each block k are determined then we need to determine what stepsize is to be taken in each block. Consider the search matrix at iteration t:

$$Y^t = \begin{pmatrix} y^t_{[1]} - x^t_{[1]} & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & y^t_{[K]} - x^t_{[K]} \end{pmatrix}$$

the Schultz-Meyer decomposition method finds the relevant stepsizes by approximately solving the problem below (we will denote this problem SMC).

$$\min_{w \in \Re^K} f(x^t + Y^t w) \text{ subject to } \mathbf{0} \leq x^t + Y^t w \leq u.$$

Observe that (SMC) is a nonlinear problem with K variables, so as the size of the BAP grows, or as the number of blocks increase, the time to solve (SMC) gets very large. SMC is always feasible since the weight w can be chosen as 0. Also the coordination step here creates a serial bottleneck to an otherwise highly parallelized algorithm. That is, while the processor assigned to solve (SMC) is busy solving the coordinator, the rest of the processors are idle. This leads to the idea of parallelizing the coordination process. While it is possible

to parallelize the solution procedure for (SMC), this would not be an easy task. In view of this, we developed simple parallelizable approaches that proved to be effective.

Chapter 3

Synchronous Multi-Coordination Schemes

In this chapter we introduce ways of parallelizing the coordination step in the decomposition method for the solution of the barrier problem. As in the Schultz-Meyer, there will be K subproblems where K is the number of the blocks of BAP. However, instead of a single coordination problem there will be P coordinators.

The P coordinating problems are solved in parallel, each producing a proposed next iterate. The point produced by the coordinator that has the best (least) objective value is then chosen as the next iterate. So this process can be viewed as "two-level" coordination where the first level consists of producing possible iterates and the second is a simple pass through these candidates to pick the best as the next iterate (see figure 1 for illustration of the case $K = P$).

First we remind the reader of the form of the subproblems and discuss the

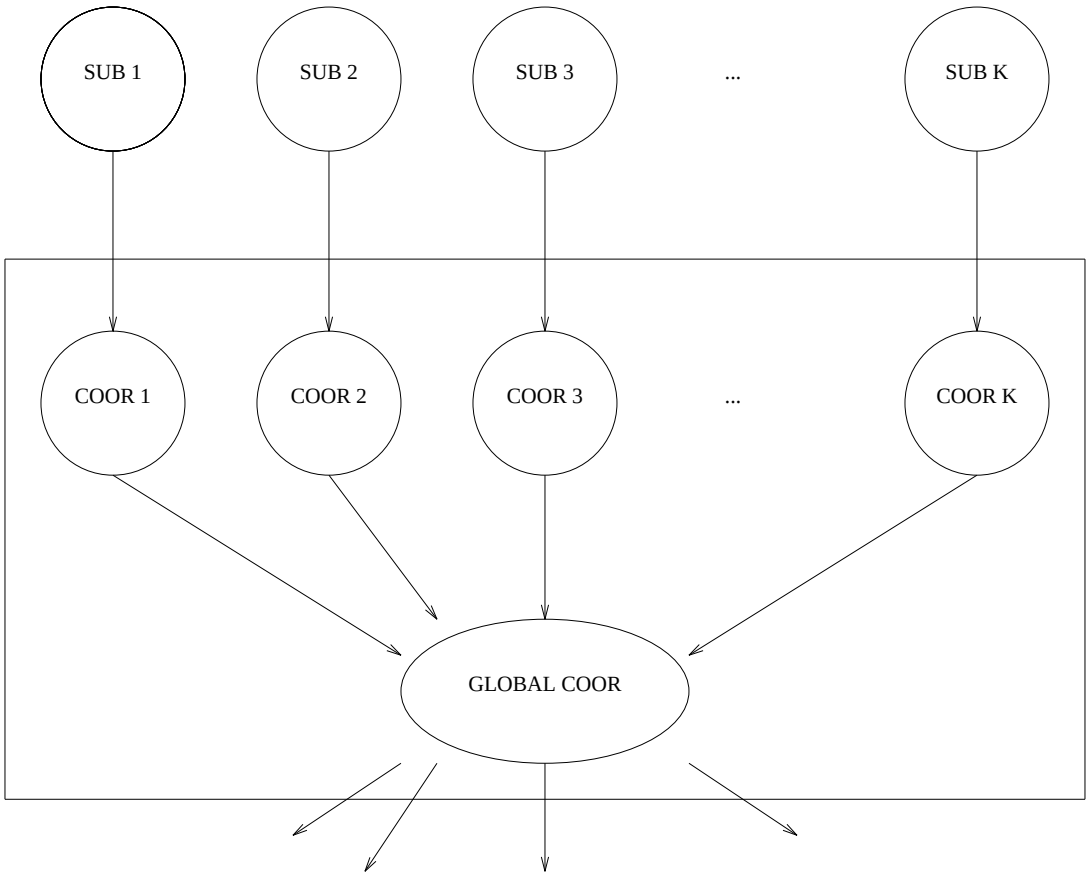

Figure 1: Single-variable multi-coordination

nature of decoupled resource allocations in more detail. Then we introduce the simplest parallel coordination scheme with $P = K$ and one coordinator for each subproblem. This scheme is referred to as "single-variable". We present convergence results for the single-variable coordination scheme. We also present "group" and "block-plus-group" multi-coordination schemes and give convergence results for them.

3.1 Subproblems and Decoupled Resource Allocation

Define a **decoupled resource allocation** for $\mathcal{B}$ to be an interval valued map R (essentially a box "centered" around the argument x^t) by:

$$\mathcal{R}(x^t) := \left\{ z \in \Re^N \quad | \quad \mathbf{0} \leq z \leq \overline{\mathcal{R}}(x^t) \right\}$$

where

$$\forall x \in \mathcal{B} \quad 0 \leq x \leq \overline{\mathcal{R}}(x^t) \leq u, \tag{19}$$

for any bounded sequence $\{x^t\} \subseteq \mathcal{B}$

$$\bigcup_{t=0}^{\infty} \mathcal{R}(x^t) \quad \text{is bounded} \tag{20}$$

and for any $z \in \mathcal{B}$ and any bounded sequence $\{x^t\} \subseteq \mathcal{B}$ with

$$\alpha^t := \max\{\alpha \mid 0 \leq \alpha \leq 1 \text{ and } x^t + \alpha(z - x^t) \in \mathcal{R}(x^t)\}, \tag{21}$$

we have $\liminf_{t\to\infty}\alpha^t > 0$.

Recall that the subproblem for determining the search direction generated by the l-th block then is of the form:

$$\begin{aligned} \min_{y_{[k]}} \quad & g^t_{[k]}(y_{[k]} - x^t_{[k]}) + (y_{[k]} - x^t_{[k]})G^t_{[k]}(y_{[k]} - x^t_{[k]}) \\ \text{subject to} \quad & A_{[k]}y_{[k]} = b_{[k]} \\ & \mathbf{0} \le y_{[k]} \le \overline{\mathcal{R}}(x^t)_{[k]}. \end{aligned} \tag{22}$$

Where it is assumed that if g^t is the concatenation of $g^t_{[k]}$:

$$\lim_{t\to\infty} ||g^t - \nabla f(x^t)|| = 0, \tag{23}$$

and if G^t is the block diagonal matrix with $G^t_{[k]}$ for the k-th block:

$$\exists \beta > 0 \text{ such that } (\forall t)(\forall x) \quad 0 \le xG^tx \le \beta||x||^2. \tag{24}$$

In our implementation we used the decoupled resource allocation provided by Schultz in his doctoral thesis. He proves that if $\mathcal{R}(x^t)$ is chosen as:

$$\begin{aligned} \mathcal{R}(x^t)_n \quad = \quad & \min\Big[u_n, x^t_n + \kappa, \\ & x_n + \min_j \left\{ \frac{\max\{\Omega, \omega(\theta_j - D_{j\cdot}(x^t))\}}{\nabla D(x^t)_{j,n}} \quad | \nabla D(x^t)_{j,n} > 0 \right\} \Big] \end{aligned}$$

then it satisfies the conditions (19) through (21). The parameters κ, Ω and ω are positive real numbers and we give their values in the computational results section. Let $j(t)$ be the index of the subproblem that produces the least subproblem objective, and $y^t_{[j(t)]}$ be the direction produced by that subproblem.

Define the notation $\overline{y}^t$ blockwise as:

$$\overline{y}^t_{[k]} = \begin{cases} y^t_{[k]} & \text{if } k = j(t) \\ x^t_{[k]} & \text{otherwise} \end{cases}$$

Theorem 5 (Sufficiency of Search Directions) *Suppose that* $\{x^t\} \subset \mathcal{B} \cap$ *dom* $f, x^t \to \tilde{x} \in \mathcal{B} \cap$ *dom* f, *and* $\overline{y}^t$ *is as above. Also suppose there exists* $\tilde{y} \in$ *lim pt*$\{\overline{y}^t\}$ *with*

$$\nabla f(\tilde{x})(\tilde{y} - \tilde{x}) \geq 0 \tag{25}$$

then

$$\forall z \in \mathcal{B} \quad \nabla f(\tilde{x})(z - \tilde{x}) \geq 0$$

Proof Without loss of generality thin the sequence so that $\overline{y}^t \to \tilde{y}$. Then from (23), (24), and (25) it follows that for any block l

$$(\forall \epsilon > 0) \quad \exists t_\epsilon \text{ such that } (\forall t > t_\epsilon)$$

$$g^t_{[l]}(\overline{y}^t_{[l]} - x^t_{[l]}) + (y^t_{[l]} - x^t_{[l]})^T G^t_{[l]}(y^t_{[l]} - x^t_{[l]}) > -\epsilon. \tag{26}$$

Now suppose for some $z \in \mathcal{B}$

$$\nabla f(\tilde{x})(z - \tilde{x}) = -\zeta < 0.$$

Still further thin the sequence so that $G^t \to \tilde{G}$. Define α^t as in (21) and $\tilde{\alpha} := \liminf_t \alpha^t > 0$. Then $\lambda \in (0, 1)$ independent of t may be picked such that:

$$\lambda \tilde{\alpha}(z - \tilde{x})^T \tilde{G}(z - \tilde{x}) < \zeta/2$$

then

$$\lambda\tilde{\alpha}\nabla f(\tilde{x})(z-\tilde{x}) + \lambda^2\tilde{\alpha}^2(z-\tilde{x})^T\tilde{G}(z-\tilde{x}) < -\frac{\lambda\zeta\tilde{\alpha}}{2} < 0.$$

Hence $(\exists\tilde{t} > 0)$ such that $(\forall t > \tilde{t})$

$$g^t[\lambda\tilde{\alpha}(z-x^t)] + [\lambda\alpha^t(z-x^t)]^T G^t[\lambda\alpha^t(z-x^t)] < -\frac{\lambda\zeta\tilde{\alpha}}{4} < 0.$$

This implies there exists a block m such that:

$$g^t_{[m]}[\lambda\tilde{\alpha}(z_{[m]}-x^t_{[m]})] + [\lambda\alpha^t(z_{[m]}-x^t_{[m]})]^T G^t_{[m]}[\lambda\alpha^t(z_{[m]}-x^t_{[m]})] < -\frac{\lambda\zeta\tilde{\alpha}}{4K} < 0$$

where K is the number of the blocks. The above contradicts (26) and hence the conclusion of the theorem is valid.

3.2 Single-Variable Multi-Coordination

In this section each coordination subproblem involves only a single variable and there is a $1-1$ correspondence between the coordinators and the subproblems. The subproblems generate descent directions for their block (or demonstrate that none exist) and then each subproblem communicates the descent direction to the coordinator responsible for that particular subproblem. The following describes the coordinator and the properties of the solution of the coordinator. Define blockwise the vector y_i^t corresponding to the ith subproblem:

$$(y_i^t)_{[k]} = \begin{cases} x^t_{[k]} & \text{if } k \neq i \\ y^t_{[i]} & \text{otherwise} \end{cases} \tag{27}$$

where $y^t_{[i]}$ is the direction produced by the ith subproblem and x^t is the current iterate. Then the ith coordinator problem is formed as:

$$\min_{w_i \in \Re} f(x^t + (y^t_i - x^t)w_i) \text{ subject to } \mathbf{0} \leq x^t + (y^t_i - x^t)w_i \leq u$$

It should be noted that the above constraints are equivalent to bound constraints on w_i. Also notice that the vector $y^t_i - x^t$ is non-zero only in the ith block. In this section we use the assumption that f is convex. Let

$$E(k,t) := \{x = (1-\lambda)x^t + \lambda y^t_k \quad | \quad 0 \leq \lambda \leq 1\}$$

Observe that $E(k,t)$ corresponds to a subset of the feasible region for the coordinator. Coordinator k computes $x^{t+1}_k = x^t + (y^t_k - x^t)w$ that

$$f(x^{t+1}_k) - f(x^t) \leq \mu \left[\min_{x \in E(k,t)} f(x) - f(x^t) \right] \tag{28}$$

where $\mu \in (0,1)$ is independent of t. To obtain a computationally checkable condition for (28) suppose $\mathbf{0}$ is not optimal for coordinator i, and let $f^t_k(\xi) = f(x^t_{[1]}, \ldots, x^t_{[k-1]}, x^t_{[k]} + (y^t_{[k]} - x^t_{[k]}, \ldots, x^t_{[K]})$, then:

$$\mu[\min_{x \in E(k,t)} f(x) - f(x^t)] = \mu[\min_{x \in E(k,t)} f(x) - f^t_k(\mathbf{0})]$$

and

$$\min_{x \in E(k,t)} f(x) = \min_{\mathbf{0} \leq \xi \leq \mathbf{1}} f^t_k(\xi)$$

Now f is convex hence $f^t_k(\xi)$ is convex (composing convex and linear functions,) therefore for every $\overline{\xi} \in \Re$:

$$f^t_k(\xi) \geq f^t_k(\overline{\xi}) + (\xi - \overline{\xi})\nabla f^t_k(\overline{\xi})$$

and

$$\min_{\mathbf{0}\leq\xi\leq\mathbf{1}} f_k^t(\xi) \geq f_i^t(\overline{\xi}) + \min\{\nabla f_k^t(\overline{\xi})(\mathbf{0}-\overline{\xi}), \nabla f_k^t(\overline{\xi})(\mathbf{1}-\overline{\xi})\}$$

since the linear functional takes its minimum over a line segment at one end. It will follow from above that:

$$\begin{aligned}
\mu[\min_{x\in E(k,t)} f(x) - f(x^t)] &= \mu[\min_{x\in E(k,t)} f(x) - f_k^t(\mathbf{0})] \\
&\geq \mu[f_k^t(\overline{\xi}) - f_k^t(\mathbf{0}) + \\
&\quad \min\{\nabla f_k^t(\overline{\xi})(\mathbf{0}-\overline{\xi}), \nabla f_k^t(\overline{\xi})(\mathbf{1}-\overline{\xi})\}]
\end{aligned}$$

Thus, if $\{\xi^i\}$ is a sequence of reals converging to a minimizer of f_k^t on $[0,1]$, then

$$\mu[f_k^t(\xi^i) - f_k^t(\mathbf{0}) + \min\{\nabla f_k^t(\xi^i)(\mathbf{0}-\xi^i), \nabla f_k^t(\xi^i)(\mathbf{1}-\xi^i)\}]$$

converges to

$$\mu[\min_{x\in E(k,t)} f(x) - f_k^t(\mathbf{0})]$$

Therefore, the sufficient decrease condition is satisfied by any point x_k^{t+1} corresponding to $w_k = \xi^i$ such that:

$$f(x_k^{t+1}) - f(x^t) \leq \mu[f_k^t(\xi^i) - f_k^t(\mathbf{0}) + \min\{\nabla f_k^t(\xi^i)(\mathbf{0}-\xi^i), \nabla f_k^t(\xi^i)(\mathbf{1}-\xi^i)\}]$$

As described in the introduction, at each iteration, K subproblems are solved each producing a search direction for the corresponding block. Within each group i the directions are communicated to the ith coordinator (recall that $P = K$ here and there is one coordinator per subproblem) and the coordinator

produces x_i^{t+1}. Then the point producing the best value amongst the groups is chosen as x^{t+1} (the new x iterate.)

Let $1 \leq j \leq K$ be the index of the subproblem that produces the minimum subproblem objective. Then:

$$f(x^{t+1}) \leq f(x_j^{t+1})$$

and therefore:

$$f(x^{t+1}) - f(x^t) \leq \mu[\min_{x \in E(j,t)} f(x) - f(x^t)] \tag{29}$$

where μ is the same as in (28).

Theorem 6 ***Convergence of Decomposition Method for Single-Variable Multi-Coordinators*** *Suppose*

1. *f is convex and once continuously differentiable,*
2. *$x^0 \in \mathcal{B} \cap dom\ f$,*
3. *$y_{[l]}^t$ is the solution to the l-th subproblem where (i) and (ii) hold for some decoupled resource allocation R,*
4. *$x^{t+1} = x^t + (y_i^t - x^t)w_i^t \in \mathcal{B}$ satisfies the coordinator condition (29) for all t.*

Then either $\lim_{t\to\infty} f(x^t) = -\infty$ *or* $\lim_{t\to\infty} f(x^t) = \tilde{f} > -\infty$ *, and in the latter case the limit points of x^t are optimal solution of the BP.*

Proof. From (29) we observe that the sequence $f(x^t)$ is decreasing. Therefore either the function values diverge to $-\infty$ or else they have a finite limiting value $\tilde{f}$. Suppose the latter is the case. Let $\tilde{x}$ be any limit point of $\{x^t\}$. Since f has the property of diverging to $+\infty$ at the barrier and the function values are decreasing it follows that $\tilde{x} \in \text{dom } f$. We see that $\lim_{t\to\infty} f(x^{t+1}) - f(x^t) = 0$. Consider a subsequence $\{x^{\sigma(t)}\} \subseteq \{x^t\}$ with $x^{\sigma(t)} \to \tilde{x}$. Define $\bar{y}^{\sigma(t)}$ as follows: Let $1 \leq j = j(t) \leq K$ be the index of the subproblem that produces the best descent direction at iteration t. Then let

$$\bar{y}^{\sigma(t)}_{[i]} = \begin{cases} x^t_{[i]} & \text{if } i \neq j \\ y^t_{[j]} & \text{otherwise} \end{cases}$$

and denote $E(j(t), t)$ by $E(t)$. Now by the boundedness of resource allocation (20) we have that $\{\bar{y}^{\sigma(t)}\}$ is bounded hence the sequence may further be thinned to obtain $\bar{y}^{\sigma(t)} \to \tilde{y}$. Now for all $\lambda \in (0, 1]$ the following holds:

$$\begin{aligned} f((1-\lambda)x^{\sigma(t)} + \lambda\bar{y}^{\sigma(t)}) &\geq [\min_{x\in E(t)} f(x) - f(x^{\sigma(t)})] \\ &\geq \frac{1}{\mu}[f(x^{\sigma(t)+1}) - f(x^{\sigma(t)})] \end{aligned}$$

(The first inequality is due to definition of $E(t)$ and the second follows from (29).)

Since the right hand side $\to 0$ as $t \to \infty$ we have $f((1-\lambda)\tilde{x} + \lambda\tilde{y}) - f(\tilde{x}) \geq 0$. Taking the limit as λ approaches 0 we get $\nabla f(\tilde{x})(\tilde{y} - \tilde{x}) \geq 0$. This together with (5) proves the conclusion.

3.3 Group and Block-Plus-Group Multi-Coordination

In this section there are two multiple coordination schemes presented as alternatives to single-variable coordinators: "group" coordinators and "block-plus-group" coordinators. There are no changes in the subproblems.

A group coordinator is used when instead of a $1-1$ relationship between subproblems and coordinators, each coordinator is responible for a group of subproblems. Let $\gamma \subseteq \{1, ..., K\}$ denote the group of subproblems coordinator p is responsible for. Also let Y_γ^t be the matrix described columnwise below:

$$(Y_\gamma^t)^l = \begin{cases} \mathbf{0} & \text{if } l \notin \gamma \\ (y_l^t - x^t) & \text{if } l \in \gamma \end{cases} \tag{30}$$

Then the p-th coordinator problem is formed as:

$$\min_w f(x^t + Y_\gamma^t w) \quad \text{subject to} \quad 0 \leq x^t + Y_\gamma^t w \leq u \ .$$

Let $i = i(\gamma) \in \gamma$ be the index of the subproblem that produced the best direction among those in group γ. Then coordinator p computes $x^{t+1}(p) = x^t + Y_\gamma^t w$ that satisfies:

$$(\forall t) \quad f(x^{t+1}(p)) - f(x^t) \leq \mu[\min_{x \in E(i,t)} f(x) - f(x^t)] \tag{31}$$

(Note that the above is certainly satisfied at optimality since $E(i,t)$ is a subset of the feasible region for the p-th coordinator.) The computationally checkable condition for (31) is the same as that for (28) in section 1.2.

Now x^{t+1} is chosen among $x^{t+1}(p)$'s so that $f(x^{t+1}) \leq f(x^{t+1}(p)) \quad (\forall p)$. Hence

if $1 \leq j \leq K$ is the index of the subproblem that produces the best direction it follows that:

$$f(x^{t+1}) - f(x^t) \leq \mu[\min_{x \in E(j,t)} f(x) - f(x^t)]$$

Therefore the crucial inequality for the proof of the decomposition method using multiple coordinators (6) is preserved. This yields convergence for the group coordinator case.

A "block-plus-group" coordinator handles a particular original block of variables i explicitly as well as stepsize variables for directions provided by a group γ of subproblems ($i \notin \gamma$). A block-plus-group coordinator problem is then formed as:

$$\begin{aligned}
\min_{\lambda, x_{[i]}} \quad & f(x_{[1]} + \lambda_1 (Y_\gamma^t)^1, ..., x_{[i-1]} + \lambda_{i-1} (Y_\gamma^t)^{i-1}, x_{[i]}, \\
& x_{[i+1]} + \lambda_{i+1} (Y_\gamma^t)^{i+1}, ..., x_{[K]} + \lambda_K (Y_\gamma^t)^K) \\
\text{subject to} \quad & A_{[i]} x_{[i]} = b_{[i]} \\
& 0 \leq x_{[l]} + \lambda_l (Y_\gamma^t)^l \leq u_{[l]} \quad (\forall l \in \gamma) \\
& 0 \leq x_{[i]} \leq u_{[i]}
\end{aligned}$$

Now let $\tilde{i} \in \{i\} \cup \gamma$ be the index of the subproblem producing the best direction among those in γ or the ith subproblem. Then the block-plus-group coordinator responsible for i and γ (say coordinator p) computes $x^{t+1}(p)$ such that:

$$(\forall t) \quad f(x^{t+1}(p)) - f(x^t) \leq \mu[\min_{x \in E(\tilde{i},t)} f(x) - f(x^t)] \tag{32}$$

(Again $E(\tilde{i}, t)$ is a subset of the feasible set for the block-plus-group coordinator therefore (32) hold at optimality.) The computationally checkable condition for

(32) is the same as that for (28).

Same as before, x^{t+1} is chosen among $x^{t+1}(p)$ so that $f(x^{t+1}) \leq f(x^{t+1}(p))(\forall p)$. So if $1 \leq j \leq K$ is the index of the subproblem that produces the best direction it follows that:

$$f(x^{t+1}) - f(x^t) \leq \mu[\min_{x \in E(j,t)} f(x) - f(x^t)]$$

Therefore the convergence proof carries through for block-plus-group coordinators as well.

3.4 Approximate Solution of Coordinators and Subproblems

3.4.1 Stabilization Algorithm for Single-Variable Coordination

We may use the stabilization algorithm outlined in figure 2 for approximately solving the single variable coordinator k:

Let $x^{t+1,k} = x^t + \alpha_k^t(x^t + y_k^t)$. Define $j(t)$ to be the index of the block with least subproblem objective. Also define $c(t)$ to be the index of the coordinator with least objective at time t. Then we choose $x^{t+1} = x^{t+1,c(t)}$, so we have

$$f(x^{t+1}) \leq f(x^t) + \gamma_1 \alpha_k^t \nabla f_{[k]}^t . (y_{[k]}^t - x_{[k]}^t) \qquad (\forall k)$$

We now proceed to prove the convergence of stabilization algorithm:

Theorem 7 (convergence of stabilization for SVC) *Suppose*

- *f is essentially smooth and $\mathcal{B}$ is closed and convex,*
- *$x^0 \in \mathcal{B}$ is given,*
- *the iterates x^t are bounded (one way of insuring this is to assume the upprebounds $u < \infty$),*
- *y_k^t defined as in (27),*
- *x^{t+1} and α^t chosen according to the stabilization algorithm.*

Then

- *The stabilization algorithm terminates in a finite number of steps,*
- *the svc algorithm produces a sequence $\{x^t\}$ of feasible points,*
- *if $\tilde{x}$ is a limit point of $\{x^t\}$ then $\tilde{x}$ is a KKT point for the barrier problem.*

Proof We thin the sequence so that:

$$x^t \to \tilde{x}$$

$$y_{j(t)}^t \to \tilde{y}$$

1. For each coordinator k, the stabilization algorithm terminates in a finite number of steps with $\alpha_k^t \in [0..1]$:

We have

$$f(x^t + \alpha(y_k^t - x^t)) - f(x^t) - \alpha \nabla f_{[k]}^t.(y_{[k]}^t - x_{[k]}^t) \in O(\alpha^2)$$

therefore for α small enough, the stabilization condition (33) is satisfied. We know $\gamma_1 \leq 1$ hence α can get as small as desired.

2.We start with $x^0 \in \mathcal{B}$ and $\alpha \in [0..1]$ and y_k^t feasible for the coordinator, hence we maintain feasibility from x^t to x^{t+1}.

3.Here we want to prove that using stabilization algorithm, if $\tilde{x}$ is a limit point of the sequence x^t, then $\tilde{x}$ is a KKT point for the barrier problem:

Our algorithm ensures $f(x^{t+1}) \leq f(x^t)$, so either $f(x^t) \to -\infty$ in which case the problem is unbounded or $\lim_{t\to\infty} f(x^{t+1}) - f(x^t) = 0$. In the latter case (33) shows:

$$\liminf_{t\to\infty} \alpha_{j(t)}^t \nabla f_{[j(t)]}^t (y_{[j(t)]}^t - x_{[j(t)]}^t) \geq 0$$

but

$$\nabla f_{[j(t)]}^t.(y_{[j(t)]}^t - x_{[j(t)]}^t) = \nabla f^t.(y_{j(t)}^t - x^t) \quad \text{by def of } y_{j(t)}^t.$$

Hence $\liminf_{t\to\infty} \alpha_{j(t)}^t \nabla f^t.(y_{j(t)}^t - x^t) \geq 0$.

Now if $\alpha_{j(t)}^t$ is bounded away from zero we have:

$$\liminf_{t\to\infty} \nabla f^t.(y_{j(t)}^t - x^t) \geq 0$$

and the theorem is proved using sufficiency of search directions lemma. So consider $\alpha_{j(t)}^t$ for which a subsequence $\alpha_{j(\sigma(t))}^{\sigma(t)} \to 0$ (we'll drop the subscript σ

from here on).

Let $f(x^{t+1,k})$ be the objective of the kth coordinator at time $t+1$, then from the stabilization algorithm and the fact that $\alpha^t_{j(t)} \to 0$ we have:

$$f(x^t + \frac{\alpha^t_{j(t)}}{\gamma_2}(y^t_{j(t)} - x^t)) > f(x^t) + \frac{\gamma_1}{\gamma_2}\alpha^t_{j(t)}\nabla f^t(y^t_{j(t)} - x^t)$$

(otherwise we would have terminated with $\alpha = \frac{\alpha^t_{j(t)}}{\gamma_2}$.) Hence

$$\frac{f(x^t + \frac{\alpha^t_{j(t)}}{\gamma_2}(y^t_{j(t)} - x^t)) - f(x^t)}{\alpha^t_{j(t)}} > \frac{\gamma_1}{\gamma_2}\nabla f^t.(y^t_{j(t)} - x^t)$$

Since $\alpha^t_{j(t)} \to 0$ we get:

$$\nabla f(\tilde{x}).(\tilde{y} - \tilde{x}) \geq \gamma_1 \nabla f(\tilde{x}).(\tilde{y} - \tilde{x})$$

So we have $\nabla f(\tilde{x}).(\tilde{y} - \tilde{x}) \geq 0$ and hence the claim is proved. ∎

3.4.2 Approximate Solutions for the Subproblems

Here we present two theorems which establish convergence of the single-variable decomposition method when subproblems are only approximately solved.

Theorem 8 *Assume we solve the subproblems such that the optimality gap is less than ξ^i at each iteration i, and that $\{\xi^t\} \to 0$. Also assume:*

- $\{x^t\} \subset \mathcal{B} \cap dom\ f$,
- $x^t \to \tilde{x} \in \mathcal{B} \cap dom\ f$,
- *y^t are the search directions found from subproblems and $\overline{y}^t = y^t_{j(t)}$ is as in (27),*

- *there exists* $\tilde{y} \in$ *lim pt*$\{\overline{y}^t\}$ *with*

$$\nabla f(\tilde{x})(\tilde{y} - \tilde{x}) \geq 0 \tag{34}$$

are satisfied. Then we will prove that

$$\forall z \in \mathcal{B} \quad \nabla f(\tilde{x})(z - \tilde{x}) \geq 0.$$

Proof WLOG thin the sequence so that $\overline{y}^t \to \tilde{y}$. For any block l then

$$(\forall \epsilon > 0) \quad \exists t_\epsilon > 0 \text{ such that } (\forall t > t_\epsilon)$$

$$g^t_{[l]}(y^t_{[l]} - x^t_{[l]}) + (y^t_{[l]} - x^t_{[l]})^T G^t_{[l]}(y^t_{[l]} - x^t_{[l]}) > -\epsilon. \tag{35}$$

Now suppose for some $z \in \mathcal{B}$

$$\nabla f(\tilde{x})(z - \tilde{x}) = -\zeta < 0.$$

Still further thin the sequence so that $G^t \to \tilde{G}$. Define α^t as in the paper and $\tilde{\alpha} := \liminf_t \alpha^t > 0$. Then $\lambda \in (0,1)$ independent of t may be picked such that:

$$\lambda\tilde{\alpha}(z - \tilde{x})^T \tilde{G}(z - \tilde{x}) < \zeta/2$$

then

$$\lambda\tilde{\alpha}\nabla f(\tilde{x})(z - \tilde{x}) + \lambda^2\tilde{\alpha}^2(z - \tilde{x})^T \tilde{G}(z - \tilde{x}) < -\frac{\lambda\zeta\tilde{\alpha}}{2} < 0.$$

Hence $(\exists \tilde{t} > 0)$ such that $(\forall t > \tilde{t})$

$$g^t[\lambda\alpha^t(z - x^t)] + [\lambda\alpha^t(z - x^t)]^T G^t[\lambda\alpha^t(z - x^t)] < -\frac{\lambda\zeta\tilde{\alpha}}{4} < 0.$$

This implies there exists a block m such that:

$$g^t_{[m]}[\lambda\alpha^t(z_{[m]} - x^t_{[m]})] + [\lambda\alpha^t(z_{[m]} - x^t_{[m]})]^T G^t_{[m]}[\lambda\alpha^t(z_{[m]} - x^t_{[m]})] < -\frac{\lambda\zeta\tilde{\alpha}}{4K} < 0$$

where K is the number of the blocks.

Let $Q_{[m]}(x)$ represent the subproblem objective at x. Then

$$Q_{[m]}(\lambda\alpha^t(z - x^t)) < -\frac{\lambda\zeta\tilde{\alpha}}{4K} < 0$$

However $(\forall t > t_{\frac{\lambda\zeta\tilde{\alpha}}{8K}})$ we have

$$Q_{[m]}(y^t_{[m]} - x^t_{[m]}) > -\frac{\lambda\zeta\tilde{\alpha}}{8K}$$

However ξ^t, the duality gap, is decreasing to zero. So

$$\exists t^* \text{ such that } (\forall t > t^*) \quad \xi^t < \frac{\lambda\zeta\tilde{\alpha}}{16K}.$$

So now let $t > \max(\tilde{t}, t^*, t_{\frac{\lambda\zeta\tilde{\alpha}}{8K}})$ then:

$$|Q_{[m]}(\lambda\alpha^t(z - x^t)) - Q_{[m]}(y^t_{[m]} - x^t_{[m]})| > \frac{\lambda\zeta\tilde{\alpha}}{8K}$$

But the duality gap $\xi^t < \frac{\lambda\zeta\tilde{\alpha}}{16K}$, and this yields a contradiction. ∎

Another termination criteria for the subproblems can be formed using a bound on the optimality gap. We first develop a method for obtaining such bound and then use this for the subproblem termination criteria. Here we assume that the upper bounds u are finite. This works well since in practice it is often the case.

(P)	Minimize	$f(x)$	(D)	maximize	$f(x) + \lambda(Ax - a) +$
					$r(x - u) + sx$
	subject to	$Ax = a$		subject to	$\nabla f(x) + A\lambda + r - s = \mathbf{0}$
		$\mathbf{0} \le x \le u$			λ free, $\quad r, s \ge \mathbf{0}$

Now, for any point x that is feasible for (P) we may construct λ, r and s such that (x, λ, r, s) is feasible for (D) (one way of doing this is to take $\lambda = \mathbf{0}$ and write $\nabla f(x) = s - r$ with $s, r \geq \mathbf{0}$ and hence we have a bound on the optimality gap given by:

$$r(u - x) + sx.$$

Now the objective is to find the sharpest such bound, so we look for the optimal solution of the following:

$$\begin{aligned} \text{minimize} \quad & r(u - x) + sx \\ \text{subject to} \quad & s - r = \nabla f(x) + A\lambda \\ & s, r \geq \mathbf{0}. \end{aligned} \tag{36}$$

Let $g = \nabla f(x)$, then (36) is equivalent to

$$\begin{aligned} \text{minimize} \quad & (A\lambda)x + ru \\ \text{subject to} \quad & A\lambda + r \geq -g \\ & r \geq \mathbf{0} \end{aligned} \tag{37}$$

except the optimal objectives vary by a constant (objective of (36) is the objective of (37) plus gx). Now we write the dual to (37):

$$\begin{array}{llcllll} \text{(D37)} & \max & -g\gamma & \Leftrightarrow & \text{(D37')} & \min & g\gamma \\ & \text{s/t} & A\gamma = a & & & \text{s/t} & A\gamma = a \\ & & \mathbf{0} \leq \gamma \leq u & & & & \mathbf{0} \leq \gamma \leq u \end{array}$$

So the sharpest bound on the optimality gap obtained in this manner is $\beta =$ $-$the optimal solution of (D37') $+ gx$. Also the above demonstrates the process

of finding the bound on the gap is completely parallelizable and timewise equivalent to solving a subproblem. For our particular test problems however, this gap was a vast overestimate and did not work well as a convergence criteria. Now suppose that using β^t, the bound on the optimality gap at time t, we use the following scheme for solving subproblems approximately:

Theorem 9 *Suppose the above scheme is used to generate approximate solutions for the subproblems. Also suppose:*

- $\{x^t\} \subset \mathcal{B} \cap dom\ f$,
- $x^t \to \tilde{x} \in \mathcal{B} \cap dom\ f$,
- y^t *are the search directions found from subproblems and* $\overline{y}^t = y^t_{j(t)}$ *is as in (27),*
- *there exists* $\tilde{y} \in\ lim\ pt\{\overline{y}^t\}$ *with*

$$\nabla f(\tilde{x})(\tilde{y} - \tilde{x}) \geq 0 \tag{39}$$

- *The optimal objective value for barrier problem is* $f^* \in \Re$.

Then we will prove that

$$\forall z \in \mathcal{B} \quad \nabla f(\tilde{x})(z - \tilde{x}) \geq 0$$

and that any accumulation point of the sequence $\{x^t\}$ *is an optimal solution of the barrier problem.*

Proof Suppose there exists $z \in \mathcal{B}$ such that:

$$\nabla \tilde{f}(z - \tilde{x}) = -\delta \text{ and } \nabla \tilde{f}(\tilde{y} - \tilde{x}) = 0$$

Then for some block $m(t)$ and for some t_0 we have, $(\forall t > t_0)$:

$$\nabla f^t_{[m(t)]}(z_{[m(t)]} - x^t_{[m(t)]}) < \frac{-\delta}{4K}$$

however

$$\nabla f^t_{[m(t)]}(y^t_{[m(t)]} - x^t_{[m(t)]}) \geq \frac{-\delta}{4K}$$

So if there is a subsequence $\sigma(t)$ for which the above subproblems had been solved to optimality then the above presents a contradiction in view of theorem 5 and we obtain the conclusions using theorem theorem 6. Hence the approximate optimality criteria of (38) must have been satisfied in the blocks $m(t)$ for all $t > t_0$. Therefore:

$$0 \leq f(x^t) - f^* \leq \frac{1}{\mu}\beta^t \leq \frac{1}{\mu}\nabla f^t_{[m(t)]}(x^t_{[m(t)]} - y^t_{[m(t)]} \leq \frac{1}{\mu} opt^t$$

where opt^t is the optimal subproblem solution for block $m(t)$. Now we know $opt^t \downarrow 0$ therefore $f(\tilde{x}) = f^*$ so by minimum principle optimality condition we arrive at the conclusions. ∎

Given $0 < \gamma_1 < \gamma_2 \leq \gamma_3 < 1$

if $\quad \nabla f^t_{[k]}.(y^t_{[k]} - x^t_{[k]}) \geq 0$

set $\alpha^t_k = 0$

else

set $\alpha = 1$

repeat until explicitly stopped :

if $(f(x^t + \alpha(y^t_k - x^t)) \leq f(x^t) + \gamma_1 \alpha \nabla f^t_{[k]}.(y^t_{[k]} - x^t_{[k]})$ (33)

then $\alpha^t_k = \alpha$ and stop.

else $\alpha = \gamma_2 \alpha$ and continue.

Figure 2: Stabilization Algorithm for Single-Variable Coordination

Terminate with the subproblem for block k if:

$$\nabla f(x^t)_{[k]}(y^t_{[k]} - x^t_{[k]}) < 0 \text{ and } \nabla f(x^t)_{[k]}(x^t_{[k]} - y^t_{[k]}) \geq \mu \beta^t \tag{38}$$

for $\mu \in (0, 1)$, or if the above is not possible terminate with the optimal LP solution.

Figure 3: Approximate Optimality for Subproblems

Chapter 4

Computational Results

In this chapter we present computational results from the implementation of single-variable and group multi-coordination schemes. We start by describing the test problems and sketch a compact version of our solution algorithm. We will then present the values of the parameters in our code. In the last section of this chapter we present our computational results for each of the above mentioned schemes and compare them to those of De Leone [14], Zenios and Pinar [18], McBride and Mamer [15], Schultz and Meyer [20] and Grigoriadis and Khachiyan [10].

4.1 The Implementation

4.1.1 Algorithm and Implementation

As we discussed in chapter 2, our algorithm follows the basic three phase method of Schultz and Meyer. Figure (4) presents a sketch of the three phase method. Although we use the same structure as the three phase method, we generate approximate solutions of the shifted barrier problem using single-variable or group multi-coordination schemes. The algorithm for approximately solving (SBP) using single-variable multi-coordination is presented in (5) and the group multi-coordination scheme is presented in figure (6).

We implemented our code on the Thinking Machines Corporation's **Connection Machine CM-5**. The machine runs the **CMOST 7.3** operating system and we used the **CMMD** message passing library for inter-processor communication. In the single-variable coordination scheme very little inter-processor communication is necessary since each processor uses only the search direction it produced to solve the sv-coordinator problem. In fact the only inter-processor communication in this algorithm takes place when we look for the candidate with the least coordinator objective (see step 3 of (5)). This is done using the `reduce` command which very efficiently searches all processors for a minimum such value.

The group coordination scheme calls for more communication. Each processor, prior to solving its assigned coordinator problem needs to know the search

directions for the blocks in its group. We could have used the `scan` command available from the **CMMD** library so that each node would only get the information regarding the group assigned to it. However we found the `concat` command more flexible and efficient. Had we used the `scan` command, there would be $|\gamma| - 1$ passes (where γ is the group) through the processors to distribute the necessary information. We used `concat` to allow every processor to have information about the current search direction on every node (the entire vector y rather than $y_{[k]}$ for some blocks k).

The algorithm would terminate if the maximum subproblem objective norm was less than the parameter spobj-tol or passed a certain number of iterations. We ran the code until we achieved at least six digits of accuracy in the optimal objective (compared with the previous results obtained by Schultz and Meyer).

4.1.2 Parameters

In all instances we used the log barrier function to form the barrier problem. We also scaled the cost coefficients so that $||c||_\infty = 1$. The rest of the parameters were as follows:

- $\Theta = 10^{-2}$ is the original amount of perturbation for the shifted barrier.
- $\lambda_\theta = 0.95$ is the parameter used in (16) to produce a sequence of shifted barriers which converge to the original barrier.

- $\tau^0 = 100$ is the initial value of the penalty parameter. The objective costs are scaled so that $||c||_\infty = 1$.

- $\tau_{\text{inf}} = 10^{-6}$ is the minimum possible value for the penalty parameter.

- $\Omega = 10^{-8}$ is the parameter used in determining the value of the decoupled resource allocation. This small value of Ω allows for a close approach to the boundary of the trust region.

- $\kappa = 10^{10}$ (which turns out to be a redundant upper bound for decoupled resource allocation of our test problems).

- $\omega \in (0.5, 0.9)$ worked well. ω is the scaling factor used in determining the decoupled resource allocation.

- $\tau_{\text{fact}} \in (0.25, 0.4)$ the factor by which we dropped the penalty parameter (for the larger problems we used a greater value of τ_{fact} which corresponds to a more gradual decrease of the penalty parameter).

4.1.3 The PDS Problems

The class of test problems is referred to as the Patient Distribution System (or PDS) problems. We chose the PDS problems to test our algorithm because these are real world problems and quite a few recently developed methods (including the Schultz-Meyer) have used PDS problems as bench mark. The PDS model is a logistics model designed to help make decisions about patient evacuation.

Assume that the parameters

$$\Theta > 0 \quad \lambda_\theta \in (0,1)$$
$$\tau^1 > 0 \quad \tau_{\inf} > 0 \quad \lambda_\tau \in (0,1)$$

are given. Also recall:

$$\mathcal{B} = \{x| \quad Ax = b, \qquad 0 \leq x \leq u\}$$

$$\mathcal{C} = \{x| \quad Dx \leq d\}$$

RELAXED PHASE
$i = 0$
Compute x^0 as the solution of the "relaxed" problem
If we determine that $\mathcal{B} = \emptyset$ then quit
Set θ^1 as in (15)
If $x^0 \in \mathcal{C}^\circ$
Then go to the refine phase
Otherwise go to the feasibility phase
FEASIBILITY PHASE
$i = i + 1$
Generate x^i as an approximate solution of $\mathcal{P}(\tau^i, \theta^i)$
Set θ^{i+1} as in (16)
$\tau^{i+1} = \tau^i$
If $x^i \in \mathcal{B} \cap \mathcal{C}^\circ$
Then go to the refine phase
Otherwise repeat the feasibility phase
REFINE PHASE $i = i + 1$
Set $\theta^i = d$
Generate x^i as an approximate solution of $\mathcal{P}(\tau^i, \theta^i)$
$\tau^{i+1} = \max(\tau_{\inf}, \lambda_\tau \tau^i)$
Repeat the refine phase until x^i is "suitable"

Figure 4: The three-phase method

To generate an approximate solution of $\mathcal{P}(\tau, \theta)$ using **single-variable** multi-coordination:

1. Solve the linear subproblem:

$$\begin{aligned} \min_{y_{[k]}} \quad & \nabla f(x^t)_{[k]}(y_{[k]} - x^t_{[k]}) \\ \text{such that} \quad & A_{[k]} y_{[k]} = b_{[k]} \\ & \mathbf{0} \leq y_{[k]} \leq \overline{\mathcal{R}}(x^t)_{[k]} \end{aligned}$$

for each block k

2. Solve the single-variable coordinator problem:

$$\min_{w_k \in \Re} f(x^t + (y^t_k - x^t)w_k) \text{ subject to } \mathbf{0} \leq x^t + (y^t_k - x^t)w_k \leq u$$

and set $x^{t,k} = x^t + (y^t_k - x^t)w^*_k$ where w^*_k is the optimal solution of the coordinator.

3. Choose $x^{t+1} = x^{t,c(t)}$ where $c(t)$ is the index of the block that produces the least objective for the barrier problem.

Figure 5: Inner iteration for single-variable multi-coordination

To generate an approximate solution of $\mathcal{P}(\tau, \theta)$ using **group** multi-coordination:

1. Solve the linear subproblem:

$$\begin{aligned} \min_{y_{[k]}} \quad & \nabla f(x^t)_{[k]}(y_{[k]} - x^t_{[k]}) \\ \text{such that} \quad & A_{[k]} y_{[k]} = b_{[k]} \\ & \mathbf{0} \leq y_{[k]} \leq \overline{\mathcal{R}}(x^t)_{[k]} \end{aligned}$$

for each block k

2. Let γ_k be the group of blocks node k is responsible for. We will drop the k subscript for ease of notation. Each group coordinator problem is of the form:

$$\min_w f(x^t + Y^t_\gamma w) \quad \text{subject to} \quad 0 \leq x^t + Y^t_\gamma w \leq u \ .$$

where γ is the group of subproblems coordinator k is responsible for and Y^t_γ is as in (30).
Set $x^{t,k} = x^t + Y^t_\gamma w^*$ where w^* is the optimal solution of above.

3. Choose $x^{t+1} = x^{t,c(t)}$ where $c(t)$ is the index of the block that produces the least objective for the barrier problem.

Figure 6: Inner iteration for group multi-coordination

PDS-x denotes the problem that models a scenario lasting x days. Table (7) lists the sizes of the problems we considered and the reader can observe that the size of PDS-x grows rapidly as x grows. PDS problems are linear multi-commodity network flow problems with 11 commodities. The column labeled *max node* gives the maximum number of nodes and the column labeled *max arc* presents the maximum number of arcs. The last two columns in the table present the size of the problem when considered as an LP. The column labeled *total constr.* contains the total number of node constraints plus the number of mutual constraints. The column labeled *total var.* contains the total number of arcs.

The block constraint matrices for these block-angular problems are node-arc incidence matrices. We take advantage of this fact in our code and use a very efficient network flow solver NSM [23] to solve the subproblems. NSM uses the network simplex method to solve the subproblems. Since the upper bounds on the subproblems are changed from each iteration to the next (we adjust the decoupled resource allocation) we can not use "hot starting". That is, we need to start with an all-artificial basis at every iteration.

We used the optimization package MINOS [17] in the form of a subroutine (MINOS 5.4) in order to solve the coordinator problems for both single-variable and group coordination. MINOS solves the above using a reduced-gradient algorithm in conjunction with a quasi-Newton algorithm. MINOS requires any domain constraint for the objective function to be specified explicitly. Therefore

problem	*max node*	*max arc*	*coupling*	*total constr.*	*total var.*
pds.1	126	339	87	1,473	3,729
pds.2	252	685	181	2,953	7,535
pds.3	390	1117	303	4,593	12,287
pds.5	686	2,149	553	8,099	23,639
pds.10	1,399	4,433	1,169	16,558	48,763
pds.20	2,857	10,116	2,447	33,874	105,728
pds.30	4,223	15,126	3,491	49,944	154,998
pds.40	5,652	20,698	4,672	66,844	212,859

Figure 7: Sizes of the PDS problems tested

we had to put in linear constraints that imposed lower bounds on the arguments of each log term involved in the barrier function. This procedure amounted to imposing upper and lower bounds on w when using single-variable multi-coordination. In the group multi-coordination case however we ended up with J (recall J is the number of coupling constraints) linear constraints for each coordinator problem which is quite significant for large problems. We used the default parameters except for the feasibility and optimality tolerance which were set to 10^{-11} in the single-variable case and to 10^{-6} in the group case.

We stopped at pds-40 mainly because the particular network solver we used does not work very efficiently for problems larger than those in pds-40. We also tested our code on another group of problems generated by MNETGEN [1] which is a derivative of NETGEN [13] discussed later in this section.

4.2 Analysis of the Results

4.2.1 Single-variable multi-coordination

In table (8) we present the solution results of the PDS problems we tested using single-variable multi-coordination method. In table (8) the first column, labeled *feas*, contains the number of outer iterations for the feasibility phase (there were 2 inner iterations per outer iteration). During the feasibility phase, from one outer iteration to the next the variable θ was changed as in (16). The next column contains the number of additional outer iterations to optimality. We changed the value of the penalty parameter τ by the relation $\tau^{i+1} = \min(\tau_{\text{fact}} * \tau^i, \tau_{\text{inf}})$. This column is followed by the *inner* column which contains the number of inner iterations per outer iteration in the optimality phase. Next is the column labeled *sub time*. Here we summed the time it took to solve the subproblems that took longest to solve in each iteration over the number of iterations. Column *coor time* contains the sum of the times it took to solve the coordinators that took longest to be solved. *rlx time* is the time it took to solve the most time consuming subproblem in the relaxed phase. The column labeled *total* contains the total time taken to solve the corresponding PDS problem and *comm.* is the percentage of time spent on communication. The communication percentage is calculated by taking the time it took to solve the relaxed problem plus subproblem and coordination time away from the total time and then dividing this by the total time and multiplying the result by 100.

problem	*feas*	*opt*	*inner*	*rlx time*	*coor time*	*sub time*	*total*	% *comm.*
pds.1	5	15	2	0.02	0.77	0.63	2.4	*
pds.2	5	15	2	0.04	0.90	1.49	3.7	*
pds.3	5	18	2	0.07	1.68	3.15	7.1	*
pds.5	5	23	2	0.23	3.19	9.14	16.9	*
pds.10	5	23	2	0.65	6.13	28.61	45.11	20
pds.20	5	23	2	4.6	12.56	191.91	252.84	17
pds.30	5	20	4	13.09	33.65	754.36	901.72	11
pds.40	5	20	4	36.57	45.75	1404.14	1806.44	17

Figure 8: Solution using single-variable multi-coordination

We used asterisks to indicate the times were too small to extract a meaningful communications percentage.

4.2.2 Group multi-coordination

Table (9) contains information about the solution of large PDS problems using group multi-coordination scheme. We also tried the group multi-coordination method on small PDS problems. The results for the small problems were not competitive with those found in the single- variable implementation. In table (9) the columns are labeled much the same as columns in table (8). The only new column is *group size*. This column contains the number of variables that we picked as the group size for each problem for the optimality phase.

In the group multi-coordination, the feasibility phase was still implemented using single-variable coordinators. As shown in table (8) a feasible point was obtained very efficiently using single-variable coordinators. Table (9) shows that

problem	*sub time*	*coor time*	*total*	*group size*	*% communication*
pds.10	29	6	45	1	20
pds.10	20	43	102	3	37
pds.20	192	13	253	1	17
pds.20	141	135	355	3	21
pds.30	754	34	902	1	11
pds.30	447	182	800	3	20
pds.30	308	191	654	5	22
pds.30	484	647	1273	7	11
pds.30	569	647	1395	11	12
pds.40	1404	46	1806	1	17
pds.40	1124	189	1704	3	21
pds.40	897	199	1434	5	21
pds.40	1009	666	1980	7	14
pds.40	1254	926	2370	11	7

Figure 9: Comparison of single-variable and group multi-coordination

for problems smaller than pds-30, solving subproblems is inexpensive and we don't gain anything from group multi-coordination. However for the larger PDS problems (e.g. PDS-30 and PDS-40) we can save time (26-27% speed up) if we choose the appropriate group size. Graph (10) plots the solution time for the large PDS problems using single-variable multi-coordination versus using group multi-coordination.

4.2.3 Comparison

In table (11) we compare the best of our results with other reported results on the same set of problems. We have reviewed the methods in chapter 1. The

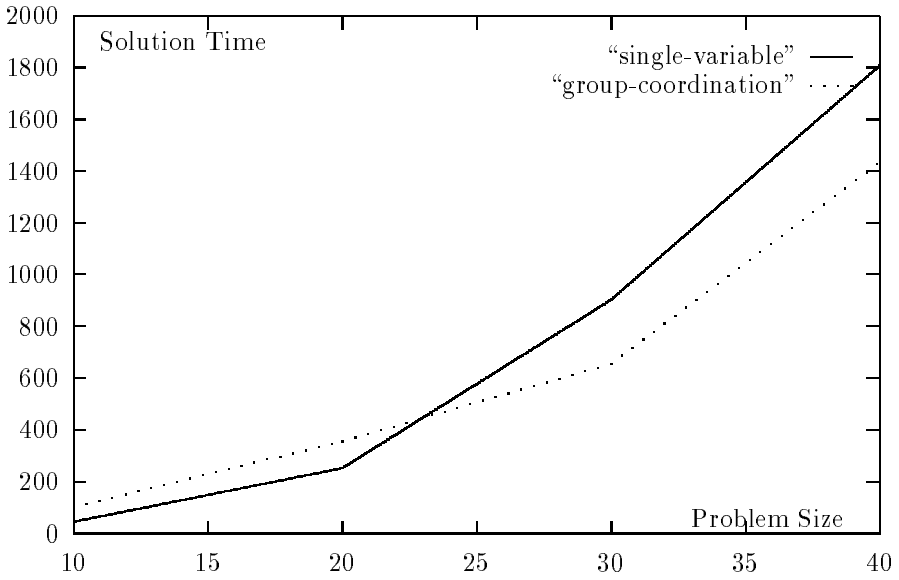

Figure 10: Single-variable multi-coordination vs. group multi-coordination

first column gives the timing results obtained by Shultz and Meyer. They implemented their method on the Sequent Symmetry machine on which each node is 5-10 times slower than the nodes of the connection machine CM5. Results obtained by Zenios and Pinar (in column labeled *ZP*) have been implemented on the Cray-YMP with 8 processors using the vector units hence the processors are 2-4 times faster than the nodes of CM5. Grigoriadis and Khachiyan [10] implemented their algorithm on the IBM RS 6000-550 which is 3 times faster than a node on the CM-5. McBride and Mamer [15] implemented their algorithm on the HP-730 work station which is also three times as fast as a node of the CM5. Another refrence for solution of the PDS problems (only up to pds.10) is [21].

problem	*SM*	*ZP*	*GK*	*MM*	*MC*
pds.10	1711	408	123	40	45
pds.20	7920	1947	372	427	253
pds.30	19380	7504	756	838	654
pds.40	37620	–	1448	4517	1434

Figure 11: Time comparison with other solution methods

4.3 The MNETGEN problems

Another set of problems we considered were those produced by MNETGEN. We discovered that the problems produced by this generator contain some mutual constraints that held as equations for any feasible point. Therefore there is no interior for the mutual constraints. Hence we perturbed the right hand side of the mutual constraints by 1 (this perturbation is 0.06-0.25 % of the original right hand side). We anticipated difficulties with these problems due to their random nature and initial lack of interior. The final solutions obtained were ran to a sufficient number of iterations (shown as col "opt") to achieve an accuracy of 4 digits (in comparison with previous results) and the timing results are reasonable. Table 12 contains the sizes and table 13 presents the results of the MNETGEN problems.

problem	*max node*	*# blocks*	*max arc*	*coupling*	*total constr.*	total var.
100.4	100	4	218	129	529	872
100.8	100	8	229	131	931	1048
200.4	200	4	423	252	1052	1692
200.8	200	8	449	277	1077	3592

Figure 12: Sizes of MNETGEN problems

problem	*# coor*	*opt*	*sub time*	*coor time*	*final objective*
100.4	1	30	6.3	8.9	0.668846e+06
100.4	3	30	6.3	16.2	0.668836e+06
100.8	1	30	6.7	9.1	0.131111e+07
100.8	3	30	6.7	15.5	0.131105e+07
100.8	5	30	6.8	26.7	0.131107e+07
100.8	7	30	6.7	33.1	0.131107e+07
200.4	1	40	11.0	20.9	0.164367+07
200.4	3	40	11.0	40.5	0.164154+07
200.8	1	40	16.2	24.0	0.320105+07
200.8	3	40	16.3	56.0	0.319489+07
200.8	5	40	16.3	80.6	0.319471+07
200.8	7	40	16.3	110.0	0.319413+07

Figure 13: Results for MNETGEN problems

Chapter 5

Asynchronous Decomposition Methods

Consider the case when the number of blocks in the BAP is larger then the number of processors available. This chapter is devoted to asynchronous solution methods, one application of which is when $K > P$ where K is the number of blocks and P is the number of available processors. We define a cycle length $n = \lceil K/P \rceil$. In this chapter we require the upper bounds u on the variable x to be finite.

5.1 Discription of the Method

We assume that $K > P$, and that information available on some blocks at time t is from previous iterations. This information has to be "recent" enough in

order to prove convergence. To formalize the above idea we use the "culling function" introduced by Schultz and Meyer.

Definition 4 *The function* $\eta : \{1, \ldots, K\} \times \mathbf{Z}_+ \to \mathbf{Z}$ *is called a* ***culling function*** *if* $y_{[k]}^{\eta(k,t)}$ *is a search direction used for block* k *at time* t *and*

$$(\forall k) \quad \lim_{t \to \infty} ||x_{[k]}^t - x_{[k]}^{\eta(k,t)}|| = 0. \tag{40}$$

(For a more detailed discussion of the above definition see Schultz's thesis [20]). We now construct a particular culling function for our algorithm. We divide the blocks into n disjoint groups, $n-1$ of which contain P blocks, while the last group contains the remaining blocks. At each iteration we concern ourselves with only one group. Hence at each iteration we have at least as many processors available as the number of blocks. Let us denote the group we process at iteration t by g_t. Then at iteration t we solve subproblems for every block in group g_t to obtain search directions for those blocks. We then solve single variable coordinators and choose the coordinator with the least objective value for producing the next iterate.

Let k be the index of a given block and define

$$i = \begin{cases} 0 & \text{if } t \leq n \\ \lfloor \frac{t - \lceil (k+1)/P \rceil}{n} \rfloor & \text{ow} \end{cases} \tag{41}$$

then the corresponding culling function for our asynchronous algorithm is:

$$\eta(k,t) = \lceil \frac{k+1}{P} \rceil + n.i$$

Notice that:

$$\frac{t - \lceil (k+1)/P \rceil}{n} - \lfloor \frac{t - \lceil \frac{k+1}{P} \rceil}{n} \rfloor \leq 1$$

therefore

$$t - \lceil \frac{k+1}{P} \rceil - n.\lfloor \frac{t - \lceil \frac{k+1}{P} \rceil}{n} \rfloor \leq n$$

so

$$t - \eta(k,t) \leq n.$$

5.2 Single-Variable Coordination in the Asynchronous Case

In this case, at each time t a batch of blocks (P of them) gets processed in the same manner as in the synchronous case. That is, each processor produces search directions for the block it is responsible for (at time t) and proceeds to solve the corresponding single variable coordinator. Then the next iterate is taken to be the one that produced the least coordinator objective (among the blocks considered at time t) concatenated by zeros in every other block.

Define the following stabilization algorithm for solving coordinator k:

Let $x^{t+1,k} = x^t + \alpha_k^t(x^t + \overline{y}^{\eta(k,t),k})$. Define $j(t)$ to be the index of the block with the least subproblem objective (here we are using the $\eta(k,t)$ update for each block k.) Also define $c(t)$ to be the index of the coordinator with the least objective at time t. Then we choose $x^{t+1} = x^{t+1,c(t)}$, so we have

$$f(x^{t+1}) \leq f(x^t) + \gamma_1 \alpha_k^t \nabla f_{[k]}^t.(y_{[k]}^{\eta(k,t)} - x_{[k]}^t) \qquad (\forall k)$$

Given $0 < \gamma_1 < \gamma_2 < 1$

if $\quad \nabla f^t_{[k]}.(y^{\eta(k,t)}_{[k]} - x^t_{[k]}) \geq 0$

set $\alpha^t_k = 0$

else

set $\alpha = 1$

repeat until explicitly stopped :

if $(f(x^t + \alpha(\overline{y}^{\eta(k,t),k} - x^t)) - f(x^t)$

$\leq \gamma_1 \alpha \nabla f^t_{[k]}.(y^{\eta(k,t)}_{[k]} - x^t_{[k]}) - ||\alpha(y^{\eta(k,t)}_{[k]} - x^t_{[k]})||^2 (*)$

then $\alpha^t_k = \alpha$ and stop.

else $\alpha = \gamma_2 \alpha$ and continue.

Figure 14: Stabilization Algorithm for Asynchronous Single-Variable Coordination

The following lemma due to Schultz indicates that so long as $||x^t_{[k]} - x^{\eta(k,t)}_{[k]}|| \to 0$, the "old information" $y^{\eta(k,t)}_{[k]}$ may play the role of "new information" computed relative to $x^t_{[k]}$.

Lemma 5 *Suppose that η is a culling function as described above and that y^t solves the subproblems in (22) with base point x^t and G^t block diagonal (blocks corresponding to the original blocks) i.e. no coupling between the subproblems. Then for t sufficiently large the subproblem solutions with kth block given by $y^{\eta(k,t)}_{[k]}$ solve subproblems of the form (22).*

The proof for the above lemma is presented in Schultz's thesis [20]. Schultz also presents in his thesis two conditions that imply (40):

Proposition 6 *Each of the following implies (40):*

1. $\{x^t\}$ *converges and* $(\forall k) \quad \lim_{t\to\infty} \eta(k,t) = +\infty$

2. $\lim_{t\to\infty} ||x^{t+1} - x^t|| = 0$ *and* $(\exists T) \quad (\forall (k,t)) \quad t - \eta(k,t) \leq T$

The proof for the above is very short and simple, it may also be found in [20].

Lemma 7 *The stepsize* $\alpha^t = 0$ *if and only if* $\alpha_{j(t)} = 0$.

Proof Suppose $\alpha_{j(t)} = 0$. This only happens if

$$\nabla f^t_{[j(t)]}.(y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]}) \geq 0.$$

So by definition of $j(t)$ we have:

$$\nabla f^t_{[k]}.(y^{\eta(k,t)}_{[k]} - x^t_{[k]}) \geq 0 \qquad (\forall k)$$

hence $\alpha^t = 0$.

On the other hand if $\alpha^t = 0$ then

$$\begin{aligned} f(x^{t+1}) = f(x^t) &\leq f(x^{t+1,j(t)}) \\ &\leq f(x^t) + \alpha_{j(t)}.\gamma_1[\nabla f^t_{[j(t)]}.(y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]}) - \\ &\quad \alpha_{j(t)}\|(y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]})\|^2] \end{aligned}$$

If $\alpha_{j(t)} > 0$, then the right hand side is less than $f(x^t)$ and this yields a contradiction. ∎

We now proceed to prove the convergence of stabilization algorithm for the asynchronous case:

Theorem 10 *Suppose that*

1. *f is essentially smooth, B is closed and convex (and we assume the upper bonds u are finite so that B is bounded),*

2. *we start with $x^0 \in \mathcal{B}$, an initial feasible point,*

3. *$y_{[k]}^{\eta(k,t)}$ is the search direction for block k at time t using the culling function η from above,*

4. *the stepsizes α^t and the iterates x^t are chosen according to the algorithm outlined in figure 14.*

Then:

1. *the stabilization algorithm terminates in a finite number of steps with $\alpha_k^t \in [0..1]$,*

2. *the iterates x^t remain feasible,*

3. *either $f(x^t) \to -\infty$ or the accumulation points of the sequence $\{x^t\}$ are KKT points for the barrier problem.*

Proof 1. For each coordinator k, the stabilization algorithm terminates in a finite number of steps with $\alpha_k^t \in [0..1]$:

We have

$$\|\alpha(x_{[k]}^t - y_{[k]}^{\eta(k,t)})\| + f(x^t + \alpha(\overline{y}^{\eta(k,t),k} - x^t)) - f(x^t) - \alpha \nabla f_{[k]}^t . (y_{[k]}^{\eta(k,t)} - x_{[k]}^t) \in O(\alpha^2)$$

therefore since $\gamma_1 < 1$ the stabilization condition $(*)$ is satisfied for α small enough.

2.We start with $x^0 \in \mathcal{B}$ and $\alpha \in [0..1]$ and $\overline{y}^{\eta(k,t),k}$ feasible, hence we maintain feasibility from x^t to x^{t+1}.

3.Our algorithm ensures $f(x^{t+1}) \leq f(x^t)$, so either $f(x^t) \to -\infty$ in which case the problem is unbounded or $\lim_{t\to\infty} f(x^{t+1}) - f(x^t) = 0$. From here on we assume that the latter is the case. If we have a subsequence of $\alpha^{\sigma(t)} = 0$ (and we will drop the σ,) then :

$$\|x^{t+1} - x^t\| = 0$$

Therefore by proposition 6 we obtain property (40) and then using lemma 5 we observe that there are no descent directions left hence the theorem is proved for the subsequence with $\alpha^t = 0$. So now we consider the case where $\alpha^t > 0$. Using the above lemma observe that $\alpha^t > 0 \Leftrightarrow \alpha_{j(t)} > 0$. From the stabilization algorithm we have:

$$f(x^{t+1}) - f(x^t) \leq f(x^{t+1,j(t)}) - f(x^t)$$

$$\leq \gamma_1 \alpha_{j(t)} \nabla f^t_{[j(t)]} . (y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]}) - \|\alpha_{j(t)}(y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]})\|^2$$

which implies:

$$0 < \gamma_1 \leq \frac{f(x^{t+1}) - f(x^t)}{\alpha_{j(t)} \nabla f^t_{[j(t)]} . (y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]}) - \|\alpha_{j(t)}(y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]})\|^2}$$

Now the numerator $\to 0$ hence we must have the denominator approaching zero as well. The denominator consists of two negative terms hence each term has

to go to zero. This yields:

$$\alpha_{j(t)} \nabla f^t_{[j(t)]} . (y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]}) \to 0 \tag{42}$$

Also if we repeat the very same argument except use $\alpha^t = \alpha_{c(t)}$ instead of $\alpha_{j(t)}$ we will obtain:

$$\left\| \alpha^t (y^{\eta(c(t),t)}_{[c(t)]} - x^t_{[c(t)]}) \right\|^2 \to 0 \tag{43}$$

which is the same thing as:

$$x^{t+1} - x^t \to \mathbf{0}$$

4.Using proposition 6 and (43) from part 3 we conclude that η is a culling function.

5.Now suppose that $\tilde{x}$ is a limit point of $\{x^t\}$ (recall that the upper bounds u were assumed to be finite). Here we will proceed to show sufficiency of search directions:

$$\liminf_{t \to \infty} \nabla f^t_{[j(t)]} . (y^t_{[j(t)]} - x^t_{[j(t)]}) \geq 0$$

First we consider the subsequences with $\alpha^{\sigma(t)}_{j(t)}$ bounded away from zero (and again we'll drop the σ.) If $\liminf_{t \to \infty} \alpha^t_{j(t)} > 0$ then the stabilization algorithm together with (43) imply:

$$\liminf_{t \to \infty} \nabla f^t_{[j(t)]} . (y^{\eta(j(t),t)}_{[j(t)]} - x^t_{[j(t)]}) \geq 0$$

and using lemma 5 we have for t large enough, $y^{\eta(j(t),t)}_{[j(t)]} = y^t_{[j(t)]}$, therefore

$$\liminf_{t \to \infty} \nabla f^t_{[j(t)]} . (y^t_{[j(t)]} - x^t_{[j(t)]}) \geq 0$$

which combined with the sufficiency of search directions lemma proves the theorem.

If on the other hand we have a subsequence with $\liminf \alpha_{j(t)}^{\sigma(t)} = 0$ then by the stabilization algorithm we must have had:

$$\frac{f(x^t + \frac{\alpha_{j(t)}}{\gamma_2}(y_{[j(t)]}^{\eta(j(t),t)} - x_{[j(t)]}^t)) - f(x^t)}{\alpha_{j(t)}}$$

$$> \frac{\gamma_1}{\gamma_2}\Big[\nabla f_{[j(t)]}^t.(y_{[j(t)]}^{\eta(j(t),t)} - x_{[j(t)]}^t) - \frac{\alpha_{j(t)}}{\gamma_2}\|(y_{[j(t)]}^{\eta(j(t),t)} - x_{[j(t)]}^t\|^2\Big]$$

Now further thin the sequence so that:

$$x^t \to \tilde{x}$$

$$\text{and} \quad \overline{y}^{\eta(j(t),t),j(t)} \to \overline{y}^{t,j(t)} \to \tilde{y}_j$$

Then by taking limits of both sides we get:

$$\frac{1}{\gamma_2}\nabla\tilde{f}.(\tilde{y}_j - \tilde{x}) \geq \frac{\gamma_1}{\gamma_2}\nabla\tilde{f}.(\tilde{y}_j - \tilde{x}).$$

Therefore since $0 < \gamma_1 < 1$ and $\gamma_2 > 0$ we obtain

$$\nabla\tilde{f}.(\tilde{y}_j - \tilde{x}) \geq 0$$

which together with the sufficiency of search directions lemma proves the theorem. ∎

5.3 Asynchronous Group Multi-Coordination

5.3.1 Description of the Algorithm

Recall that $y_p^{\eta(t)}$ is the most recent search direction available for block p at time t. Here at each iteration t a different group of blocks is considered in the coordination step on processor p. For example if we had 12 blocks and 4 processors and we had decided that the size of coordinator 2 is two then processor number 2 would "consider" the blocks as follows:

at time $t = 1$ the group $\{1, 2\}$

at time $t = 2$ the group $\{5, 6\}$

at time $t = 3$ the group $\{9, 10\}$

at time $t = 4$ the group $\{1, 2\}$

So although the set of indices of blocks involved in coordination on node p is a function of the iteration, there are only n possibilities for this (as one can observe from the example). Hence we refer to $\Gamma_{p,q}$ as the group of blocks whose search direction is considered by the coordinator on node p at time t with $t \bmod n = q$. Define $Y^{\eta(t)}$ as:

$$Y^{\eta(t)} = \begin{pmatrix} y_{[1]}^{\eta(t)} - x_{[1]}^t & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & y_{[K]}^{\eta(t)} - x_{[K]}^t \end{pmatrix} \tag{44}$$

Let p be the index of our processor and $\hat{w}_p^t$ be the solution to the following:

$$\min \quad \nabla f(x^t) Y^{\eta(t)} w + w' H^t w$$

Given $0 < \gamma_1 < \gamma_2 < 1$, if $\nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t \geq 0$ then set $\alpha_p^t = 0$, else:

Set $\alpha = 1$

Repeat

if $f(x^t + \alpha Y^{\eta(t)} \hat{w}_p^t) - f(x^t)$
$\leq \gamma_1 [\alpha \nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t - ||\alpha Y^{\eta(t)} \hat{w}_p^t||^2]$

then choose $\alpha_p^t = \alpha$ and stop

else reset $\alpha = \gamma_2 . \alpha$ and continue.

On processor p, the new iterate $x^{t+1,p} = x^t + \alpha_p^t Y^{\eta(t)} \hat{w}_p^t$.

Figure 15: Stabilization Algorithm for Asynchronous Single-Variable Coordination

$$\text{subject to} \quad w_{[l]} = \mathbf{0} \quad \text{if } l \notin \Gamma_{p,q}$$

$$w \text{ feasible for the master problem on node } p$$

Here we require H^t to be symmetric positive definite and have:

$$(\exists \beta) \quad \text{such that} \quad (\forall w) \quad |w' H^t w| \leq \beta ||w||^2.$$

The following is the algorithm we propose for solving the group coordinator on processor p:

Let $j(t)$ be the index of the block with the best subproblem objective at time t and $c(t)$ the index of the block with best (least) coordinator objective. We choose the next iterate $x^{t+1} = x^{t+1,c(t)}$ and $\alpha^t = \alpha_{c(t)}^t$.

5.3.2 Lemmas

Lemma 8 *$\alpha^t = 0$ if and only if $\alpha_p^t = 0 \quad (\forall p)$.*

Proof Let p be an arbitrary but fixed processor index. If $\alpha^t = 0$ then

$$f(x^{t+1}) = f(x^t) \leq f(x^{t+1,p}) \leq f(x^t) + \gamma_1[\alpha_p^t \nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t - ||\alpha_p^t Y^{\eta(t)} \hat{w}_p^t||^2.$$

So if $\alpha_p^t > 0$ then the right hand side is less than $f(x^t)$ and this yields a contradiction. Proof for the other direction is clear since $\alpha^t = \alpha_p^t$ for some p. ∎

Lemma 9 *If $\alpha^t > 0$ then $\alpha_{j(t)}^t > 0$.*

Proof We will proceed by proving the contrapositive. Suppose $\alpha_{j(t)}^t = 0$, therefore we must have had $\nabla f(x^t) Y^{\eta(t)} \hat{w}_{j(t)}^t \geq 0$. Hence

$$\nabla f(x^t) Y^{\eta(t)} \hat{w}_{j(t)}^t + (\hat{w}_{j(t)}^t)' H^t \hat{w}_{j(t)}^t \geq 0. \tag{45}$$

From (45) we will conclude $\nabla f(x^t)_{[j(t)]} . (y_{[j(t)]}^{\eta(t)} - x_{[j(t)]}^t) \geq 0$: Assume on the contrary that $\nabla f(x^t)_{[j(t)]} . (y_{[j(t)]}^{\eta(t)} - x_{[j(t)]}^t) < 0$, then for γ small enough:

$$\gamma \nabla f(x^t) Y^{\eta(t)} e_{j(t)} + (\gamma e_{j(t)})' H^t \gamma e_{j(t)} < 0$$

where

$$(e_{j(t)})_{[l]} = \begin{cases} \mathbf{1} & \text{if } l = j(t) \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

Using the definition of $\hat{w}_{j(t)}^t$ we observe that the above contradicts (45) so we obtain:

$$\nabla f(x^t)_{[j(t)]} . (y_{[j(t)]}^{\eta(t)} - x_{[j(t)]}^t) \geq 0.$$

So, by definition of $j(t)$ we obtain:

$$\nabla f(x^t)_{[k]}.(y^{\eta(t)}_{[k]} - x^t_{[k]}) \geq 0 \quad (\forall k).$$

This yields $\alpha^t = 0$ which proves the lemma. ∎

Lemma 10 *Suppose* $(\forall k \in \{1..L\}) \quad z^t_k \leq 0$ *and we have* $\liminf_{t\to\infty} \sum_{k=1}^{L} z^t_k \geq 0$, *then* $(\forall k \in \{1..L\}) \liminf_{t\to\infty} z^t_k = 0$.

Proof Suppose that there exists r such that $\liminf_{t\to\infty} z^t_r < 0$. Now we know:

$$\sum_{k=1}^{L} z^t_l \leq z^t_r$$

(since all the terms in the sum are nonpositive). So taking lim inf of both sides we obtain:

$$\liminf_{t\to\infty} \sum_{k=1}^{L} z^t_l \leq \liminf_{t\to\infty} z^t_r < 0$$

which is a contradiction. ∎

Lemma 11 *Suppose we have*

- $\lim_{t\to\infty} x^{t+1} - x^t = \mathbf{0}$, *and*
- $\liminf_{t\to\infty} \nabla f(x^t) Y^{\eta(t)} \hat{w}^t_p \geq 0$ *for a given* p.

Then $\liminf_{t\to\infty} \sum_{k\in\Gamma_{p,q}} \nabla f(x^t)_{[k]}.(y^t_{[k]} - x^t_{[k]}) \geq 0 \quad (\forall q)$.

Proof Using lemma 5 and proposition 6 coupled with the first assumption we get that η is a culling function and that for t sufficiently large $y^{\eta(t)} = y^t$. Now

assume on the contrary, then we thin the sequence to get:

$x^t \to \tilde{x}$	
$y^t \to \tilde{y}, y^{\eta(t)} \to \tilde{y}$	first implication is by boundedness of resource allocation, and the second by properties of the culling function η.
$Y^t \to \tilde{Y}, Y^{\eta(t)} \to \tilde{Y}$	by definition of $Y^{\eta(t)}$ and Y^t.
$H^t \to \tilde{H}$	by boundedness of eigenvalues requirement on H^t.

And we will have:

$$\liminf_{t\to\infty} \sum_{k\in\Gamma_{p,q}} \nabla f(x^t)_{[k]}.(y^t_{[k]} - x^t_{[k]}) = -\xi < 0$$

for some q. Now choose $\lambda_{p,q} \in (0..1]$ so that:

$$\sum_{k\in\Gamma_{p,q}} \lambda_{p,q} \nabla f(\tilde{x})_{[k]}.(\tilde{y}_{[k]} - \tilde{x}_{[k]}) + \lambda^2_{p,q} i_{p,q}{}' \tilde{H} i_{p,q} < \frac{-\lambda_{p,q}\xi}{2}$$

where

$$(i_{p,q})_{[k]} = \begin{cases} \mathbf{1} & \text{if } k \in \Gamma_{p,q} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

(Note that essentially we need $\lambda_{p,q}(i_{p,q}{}'\tilde{H} i_{p,q}) < \xi/2$.) Since $\lambda_{p,q} \in (0..1]$, $\lambda_{p,q} i_{p,q}$ is feasible for the master on node p, so from definition of $\hat{w}^t_p$, for t sufficiently large and $t \bmod n = q$ we'll have:

$$\nabla f(x^t) Y^{\eta(t)} \hat{w}^t_p \le \nabla f(x^t) Y^{\eta(t)} \lambda_{p,q} i_{p,q} + (\lambda_{p,q} i_{p,q})' H^t (\lambda_{p,q} i_{p,q}) < \frac{-\lambda_{p,q}\xi}{4} < 0.$$

And this contradicts the assumption so the lemma is proved. ∎

Lemma 12 *Suppose we have*

- $\lim_{t\to\infty} x^{t+1} - x^t = \mathbf{0}$, *and*

- $\liminf_{t\to\infty} \nabla f(x^t) Y^{\eta(t)} \hat{w}^t_{j(t)} \geq 0.$

Then $\liminf_{t\to\infty} \nabla f(x^t)_{[j(t)]}.(y^t_{[j(t)]} - x^t_{[j(t)]}) \geq 0.$

Note here that we can not merely combine lemma 11 and lemma 10 to get this since $j(t)$ is a time dependent index. Also here we take advantage of the assumption that $j(t) \in \Gamma_{j(t),q}$ from the way we design our algorithm.

Proof Using lemma 5 and proposition 6 coupled with the first assumption we get that η is a culling function and that for t sufficiently large $y^{\eta(t)} = y^t$. Assume on the contrary that:

$$\liminf_{t\to\infty} \nabla f(x^t)_{[j(t)]}.(y^t_{[j(t)]} - x^t_{[j(t)]}) = -\xi < 0.$$

Then

$$\liminf_{t\to\infty} \nabla f(x^t)_{[j(t)]}.(y^{\eta(t)}_{[j(t)]} - x^t_{[j(t)]}) = -\xi < 0.$$

So now define $i_{j(t)}$ blockwise by:

$$(i_{j(t)})_{[k]} = \begin{cases} \mathbf{1} & \text{if } k = j(t) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Then $\liminf_{t\to\infty} \nabla f(x^t) Y^{\eta(t)} i_{j(t)} = -\xi < 0$. Now define $||\mathcal{H}|| = \liminf_{t\to\infty} i'_{j(t)} H^t i_{j(t)}$ and choose λ such that:

$$(\lambda)(\nabla f(x^t) Y^{\eta(t)} i_{j(t)}) + \lambda^2 ||\mathcal{H}|| < -\lambda\xi/2$$

Note here that λ may be chosen independent of t so that $\lambda||\mathcal{H}|| < \xi/2$ and this choice of λ will yield the desired inequality. So then for t sufficiently large we have:

$$\nabla f(x^t)Y^{\eta(t)}\hat{w}^t_{j(t)} \leq \nabla f(x^t)Y^{\eta(t)}\lambda i_{j(t)} + (\lambda i_{j(t)})'H^t(\lambda i_{j(t)}) < -\lambda\xi/4.$$

Considering $\lambda \in (0..1]$, observe that $\lambda i_{j(t)}$ is feasible for the master on node $j(t)$ thus using the definition of $\hat{w}^t_{j(t)}$ we arrive at a contradiction. ∎

5.3.3 Convergence Theorem

Theorem 11 *Suppose that*

1. *f is essentially smooth and $\mathcal{B}$ is closed and convex (we also include large bounds on x so that $\mathcal{B}$ is bounded).*

2. *An initial feasible point $x^{(0)} \in \mathcal{B}$ is given.*

3. *$y_p^{\eta(t)}$ is the most recent search direction available for processor p.*

4. *$Y^{\eta(t)}$ is defined as in (44).*

5. *We use the stabilization algorithm in this section to obtain new x_p iterates from old ones on each processor.*

6. *The actual new iterate x^t is the x_p^t which resulted in the least coordinator objective amongst all processors used in iteration t.*

Then

1. *the stabilization algorithm terminates in a finite number of steps with* $\alpha_p^t \in [0..1]$.

2. *For the sequence* $\{x^t\}$ *either* $f(x^t) \to -\infty$ *or each* $\tilde{x}$ *that is a limit point of* $\{x^t\}$, $\tilde{x}$ *is a KKT point of the barrier problem we are minimizing.*

Proof Since the feasible region is bounded we must have a limit point for the $\{x^t\}$.

1. For each coordinator p the stabilization algorithm terminates in a finite number of steps with $\alpha_p^t \in [0..1]$: For any given processor p at time t if $\nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t \geq 0$ then we terminate with $\alpha_p^t = 0$ and we clearly have finite termination. If on the other hand $\nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t < 0$ we have:

$$f(x^{t+1,p}) - f(x^t) - \alpha \nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t + \left|\left|\alpha Y^{\eta(t)} \hat{w}_p^t\right|\right|^2 \in O(\alpha^2).$$

   Now since $0 < \gamma_1 < 1$ for α small enough the stabilization condition will be satisfied hence we exit the algorithm in finite number of steps and with $\alpha_p^t \in [0..1]$.

2. We maintain feasibility throughout the iterations since $x^{(0)}$ is feasible and each new iterate is a convex combination of the previous iterate and search direction both of which are feasible for the network constraints and weights are chosen so as not to violate any upper bound constraints.

3. The algorithm insures $f(x^{t+1}) \leq f(x^t)$ so either $f(x^t) \to -\infty$ in which case the problem is unbounded or $\lim_{t \to \infty} f(x^{t+1}) - f(x^t) = 0$.

From here on we assume that the latter is the case. If we have a subsequence $\alpha^{\sigma(t)} = 0$ (and we will drop the superscript σ for simplicity,) then

$$||x^{t+1} - x^t|| = 0 \tag{46}$$

and

$$\alpha_p^t = 0 \quad (\forall p) \quad \text{using lemma 8.} \tag{47}$$

Therefore by (47) we must have had:

$$\nabla f(x^t) Y^{\eta(t)} \hat{w}_p^t \geq 0 \quad (\forall p).$$

This together with (46) are the hypothesis of lemma 11 for any p, therefore we have the conclusion of lemma 11:

$$\liminf_{t\to\infty} \sum_{k\in\Gamma_{p,q}} \nabla f(x^t)_{[k]}.(y_{[k]}^t - x_{[k]}^t) \geq 0 \quad (\forall q)(\forall p).$$

So now we apply lemma 10 to the above to get:

$$\liminf_{t\to\infty} \nabla f(x^t)_{[k]}.(y_{[k]}^t - x_{[k]}^t) = 0 \quad (\forall k \in [1..K]).$$

Now using the sufficiency of search directions lemma it is clear that we have convergence for the case when a subsequence $\alpha^{\sigma(t)} = 0$.

So now suppose $\alpha^t > 0$. Then from lemma 9 we get $\alpha_{j(t)}^t > 0$. From the stabilization algorithm we have

$$f(x^{t+1}) - f(x^t) \leq f(x^{t+1,j(t)}) - f(x^t) \leq \gamma_1 \left[\alpha_{j(t)}^t \nabla f(x^t) Y^{\eta(t)} \hat{w}_{j(t)}^t - ||\alpha_{j(t)}^t Y^{\eta(t)} \hat{w}_{j(t)}^t||^2\right]$$

which implies:

$$0 < \gamma_1 \leq \frac{f(x^{t+1}) - f(x^t)}{\alpha_{j(t)}^t \nabla f(x^t) Y^{\eta(t)} \hat{w}_{j(t)}^t - ||\alpha_{j(t)}^t Y^{\eta(t)} \hat{w}_{j(t)}^t||^2}.$$

Now the numerator approaches 0 hence we must have the denominator approaching 0 as well. Since the denominator consists of two negative pieces we get:

$$\alpha^t_{j(t)} \nabla f(x^t) Y^{\eta(t)} \hat{w}^t_{j(t)} \to 0.$$

And if we repeat the same argument with α^t instead of $\alpha^t_{j(t)}$ then:

$$||\alpha^t Y^{\eta(t)} \hat{w}^t_{j(t)}||^2 \to 0 \tag{48}$$

which means

$$x^{t+1} - x^t \to 0. \tag{49}$$

So using (48) and taking lim inf of both sides of the required condition of convergence in the stabilization algorithm we get:

$$\liminf_{t\to\infty} \alpha^t_{j(t)} \nabla f(x^t) Y^{\eta(t)} \hat{w}^t_{j(t)} \geq 0.$$

Now if $\liminf_{t\to\infty} \alpha^t_{j(t)} > 0$ then we have

$$\liminf_{t\to\infty} \nabla f(x^t) Y^{\eta(t)} \hat{w}^t_{j(t)} \geq 0$$

and this together with (49) and lemma 12 proves the theorem.

Now the only case remaining is $\liminf_{t\to\infty} \alpha^t_{j(t)} = 0$. Suppose ${\alpha^t_{j(t)}}^{\sigma(t)}$ is the subsequence that converges to 0. (From here on we drop the superscript σ). Thus by the stabilization algorithm we must have had:

$$\frac{f(x^t + \frac{\alpha^t_{j(t)}}{\gamma_2} Y^{\eta(t)} \hat{w}^t_{j(t)}) - f(x^t)}{\alpha^t_{j(t)}} > \frac{\gamma_1}{\gamma_2} \left[\nabla f(x^t) Y^{\eta(t)} \hat{w}^t_{j(t)} - \frac{\alpha^t_{j(t)}}{\gamma_2} ||Y^{\eta(t)} \hat{w}^t_{j(t)}||^2 \right].$$

Now we further thin the sequence to get:

$$x^t \to \tilde{x},$$

$$Y^{\eta(t)} \to \tilde{Y},$$

and

$$\hat{w}^t_{j(t)} \to \tilde{w}_j.$$

Now by taking limits from both sides of the inequality we obtain:

$$\frac{1}{\gamma_2}\nabla \tilde{f}.(\tilde{Y}\tilde{w}_j) \geq \frac{\gamma_1}{\gamma_2}\nabla \tilde{f}.(\tilde{Y}\tilde{w}_j).$$

Hence:

$$\liminf_{t\to\infty} \nabla f(x^t) Y^{\eta(t)} \hat{w}^t_{j(t)} \geq 0.$$

So again we use (49) and lemma 12 to establish the conclusion. ∎

Chapter 6

Future Directions

There are a few related areas which remain to be explored. One is the generalization of asynchronous schemes to the block-plus-group coordinators and the implementation of all asynchronous methods. The asynchronous schemes presented in chapter 5 were designed to work for the case when the number of blocks is more than the number of processors. With some minor alterations they could be used to compensate and balance loads if there is any load unbalancing present (e.g. if a block is significantly larger than other blocks). Here we can go through several iterations using old information for the large blocks until the corresponding subproblems on those blocks are solved and new information is obtained.

Another area that needs more examination is formation of the group coordinators. As we discussed in chapter 4, we need to specify the domain constraints

explicitly. In the case of single-variable coordinators this translates to mere upper and lower bounds on the stepsize variable w. In the group coordinator case the logarithmic domain constraints are linear constraints. There are as many of these as the number of mutual constraints (that is a lot!) and this slows down the coordination process. Hence it is desirable to eliminate the redundant or linearly dependent constraints. Aggregation of constraints (used in bundle methods) may be a way of reducing the number of such constraints. Another approach may be to use an active set strategy.

We also pointed out in the chapter for computational results that we were not able to use hot starting for solution of the subproblems. This is due to the fact that the bounds on the variable y would change from one iteration to the next since the decoupled resource allocation would change. We could update the decoupled resource allocation every few iterations rather than every iteration to achieve this. Also, the decoupled resource allocation could be designed to have more of a centering property to it. That is, making a box trust region centered around the old iterate. That way the old iterate would still be feasible as a starting solution for the subproblems.

Bibliography

[1] Agha Iqbal Ali and J. Kennington. MNETGEN program documentation. Technical Report IEOR 77003, Department of Industrial Engineering and Operations Research, Southern Methodist University, Dallas, Texas 75275, 1977.

[2] Chunhui Chen and O.L.Mangasarian. A class of smoothing functions for nonlinear and mixed complimentarity problems. Technical Report TR94-11, Center for Parallel Optimization, Computer Sciences Department, University of Wisconsin, August 1994.

[3] G.B. Dantzig and P. Wolfe. A decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[4] Renato DeLeone, Manlio Gaudioso, and Maria Flavia Monaco. Nonsmooth optimization methods for parallel decomposition of multicommodity flow problems. Technical Report 1080, Centre for Parallel Optimization, Computer Sciences Department, University of Wisconsin, 1992.

[5] Renato DeLeone, Robert R. Meyer, and Spyridon Kontogiorgis. Alternating direction splittings for block-angular parallel optimization. Technical Report 1217, Centre for Parallel Optimization, Computer Sciences Department, University of Wisconsin, 1994.

[6] M.C. Ferris and O.L. Mangasarian. Parallel variable distribution. Technical Report 1175, Centre for Parallel Optimization, University of Wisconsin, Computer Sciences Dept., 1993.

[7] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley & Sons, New York, 1968.

[8] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princton University Press, Princton, NJ, 1962.

[9] K.R. Frisch. The logarithmic potential method of convex programming. Unpublished manuscript, University Institute of Economics, Oslo, 1955.

[10] M.D. Grigoriadis and L.G. Khachiyan. An exponential-function reduction method for block-angular convex programs. Technical Report 211, Laboratory for Computer Sciences Research, Rutgers, Computer Sciences Dept., May 1993.

[11] W. Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, University of Chicago, 1939.

[12] J.L. Kennington and R.V. Helgason. *Algorithms for Network Programming*. John Wiley and Sons, New York, 1980.

[13] D. Klingman, A. Napier, and J. Stutz. NETGEN—A program for generation of large-scale (un)capacitated assignment, transportation and minimum cost network problems. *Management Science*, 20:814–822, 1974.

[14] R. De Leone, R.R. Meyer, S. Kontogiorgis, A. Zakarian, and G. Zakeri. Coordination in coarse grained decomposition. *SIAM Journal on Optimization*, 4(4):777–793, 1994.

[15] Richard D. McBride and John W. Mamer. Solving multicommodity flow problems with a primal embedded network simplex algorithm. submitted.

[16] Deepankar Medhi. Parallel bundle-based decomposition for large-scale structured mathematical programming problems. *Annals of Operations Research*, 22:101–127, 1990.

[17] B.A. Murtagh and M.A. Saunders. MINOS 5.4 release notes, appendix to MINOS 5.1 user's guide. Technical report, Stanford University, 1992.

[18] M.C. Pınar and S.A. Zenios. Parallel decomposition of multicommodity network flows using a linear-quadratic penalty algorithm. *ORSA Journal on Computing*, 4:235–249, 1992.

[19] R.T. Rockafellar. *Conjugate Duality and Optimization*, volume 16 of *Conference Board of the Mathematical Sciences*. SIAM, Philadelphia, PA, 1974.

[20] G.L. Schultz. *Barrier Decomposition for the parallel optimization of block-angular programs*. PhD thesis, University of Wisconsin-Madison, 1991.

[21] Rudy Setiono. An interior dual proximal point algorithm for linear programs. Technical Report TR879, Center for Parallel Optimization, Computer Sciences Department, University of Wisconsin, September 1989.

[22] B. von Hohenbalken. Simplicial decomposition in nonlinear programming algorithms. *Mathematical Programming*, 13:49–68, 1977.

[23] Armand Zakarian. Private communication.

[24] Armand Zakarian. Parallel solution of multi-commodity network flow programs via nonlinear jacobi algorithm. Unpublished manuscript.